Linux iptables 速查手冊

簡介

Linux 核心裡，負責處理網路封包的子系統稱為 Netfilter，而 iptables 是用於設定這個子系統的 user-space 命令。本手冊涵蓋 iptables 1.2.7a 版，此版本可操作 Linux 2.4 版核心的 Netfilter 子系統，以及 2.6 版核心裡的大部份等效機制。由於 Netfilter 與 iptables 有密不可分的關係，所以本手冊一律以「iptables」一詞來統稱這兩者。

iptables 以「**篩表**」(table) 的觀念來組織各種網路封包的處理**規則** (rule)，不同用途的處理規則（封包過濾、網路位址轉譯、修改封包內容），分別歸納於不同的篩表。「**規則**」(rule) 是以**篩選條件** (match) 與**處置目標** (target) 構成，前者挑出要被處理的封包，後者決定如何處置符合條件的封包。

iptables 是 OSI Layer 3（網路層）的工具，加上擴充模組後，它也可以處理 OSI Layer 4（傳輸層）的協定，但是對於 OSI Layer 2（連結層），就不是 iptables 能夠掌握的領域了，在這方面，你需要改用其它技術，例如 ebtables（Ethernet Bridge Tables），詳情請參閱 http://ebtables.sourceforge.net/。

操作實例

這裡有個簡單的 iptables 命令：

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80
    -j DNAT --to-destination 192.168.1.3:8080
```

《表 1》解釋這個 iptables 命令的意義。

表 1：拆解 iptables 命令

參數	說明
-t nat	操作nat 篩表（table）。
-A PREROUTING	添加（append）規則到指定表的 PREROUTING 串鏈。
-i eth1	篩選條件：從 eth1 介面進來（in）的封包。
-p tcp	篩選條件：封包格式符合 tcp 協定（protocol）的規範。
--dport 80	篩選條件：封包的目的埠（destination port）欄位為 80。
-j DNAT	處置目標：跳到（jump）DNAT 目標。
--to-destination 192.168.1.3:8080	處置目標：將封包的目的地位址與目的埠分別改成 192.168.1.3 和 8080。

觀念

Linux 核心的封包處理流程中，共設置了五個**攔截點**（hook points），分別是 PREROUTING、INPUT、FORWARD、POSTROUTING 以及 OUTPUT。內建串鏈只能作用在這些攔截點；你可以針對個別攔截點設置一系列處理規則，每條規則各代表一次影響（或監測）封包處理流程的機會。

訣竅

很多說明文件常有『... nat 篩表的 PREROUTING 串鏈 ...』這樣的說法，隱喻著串鏈是屬於篩表。然而，串鏈與篩表兩者之間並沒有統屬關係，頂多只有隱諱的關聯性而已。串鏈（chains）的真正含意是「封包徑路上的攔截點」，而篩表（tables）則是象徵「處理效果」。然而，為了措詞上的方便，本手冊仍免不了出現『... 某篩表的某串鏈 ...』之類的說法，請讀者注意。

圖 1、2、3 分別展示了篩表與串鏈的三種有效組合，以及各種組合所象徵的封包處理流程。其中《圖 1》是封包經過「網址轉譯系統」（NAT）的流程，相關串鏈是作用於 nat 篩表。

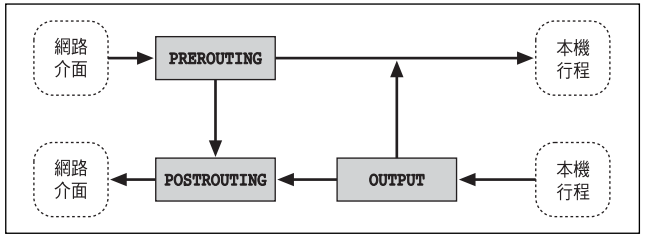

圖 1：nat 篩表與相關攔截點

《圖 2》是封包流經「封包過濾系統」的流程，相關串鏈是作用於 filter 篩表。

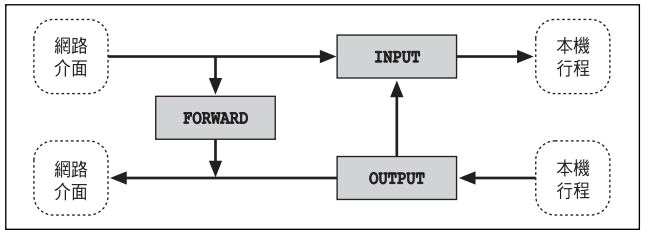

圖 2：filter 篩表與相關攔截點

《圖 3》是封包通過「封包內容調整系統」的流程，相關串鏈是作用於 mangle 篩表。

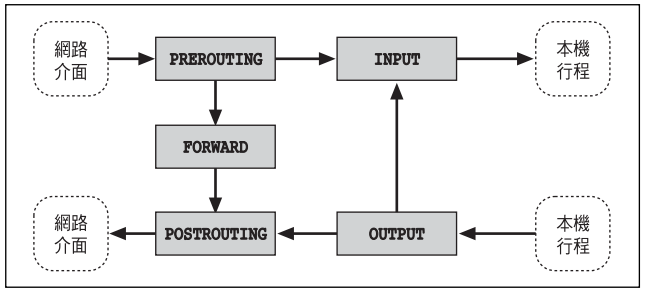

圖 3：mangle 篩表與相關攔截點

《表 2》說明五種攔截點（串鏈）的作用，以及各攔截適合處理的封包類型。

表 2：攔截點（串鏈）

攔截點	適合處理的封包類型
FORWARD	要通過閘道器的封包：從某一個介面進來，然後立刻從另一個介面出去。
INPUT	即將交付給本機行程的封包。
OUTPUT	本機行程剛剛產生的封包。
POSTROUTING	即將從網路介面出發的封包。
PREROUTING	剛剛抵達網路介面的有效封包（丟棄非混雜模式【譯註】下不應該收下的封包，或是應該收下、但是 checksum 有誤的封包）。

譯註　混雜模式（promiscuous mode）：Ethernet 網卡的一種特殊作業模式，在此模式下，即使目的地 MAC 位址不是指向自己的 Ethernet 封包，也會被收下來。Ethernet 網卡通常是在非混雜模式下作業，也就是只接收 MAC 指向自己的 Ethernet 封包。

訣竅

好奇的讀者，可從核心原始程式的 /usr/include/linux/netfilter_ipv4.h 標頭檔查出各攔截點的定義；它們的名稱類似 NF_IP_FORWARD、NF_IP_LOCAL_{IN,OUT}、和 NF_IP_{PRE,POST}_ROUTING。

規則應該設置於哪個篩表的哪個串鏈，取決於規則本身的功能性，以及封包的性質。比方說，若你要設置的一條用於過濾離境封包的規則，則應該將該規則設置於「filter 篩表」（因為功能性是「過濾」）的「OUTPUT 串鏈」（因為封包性質是「離境封包」）。雖然要離境的封包，其最後一關應該是 at 篩表的 POSTROUTING 串鏈（參閱《表 4》與《表 6》），但由於 nat 篩表管不到 POSTROUTING 串鏈，所以你不能將出境封包過濾規則設置在那裡。

篩表（Tables）

iptables 內建三個篩表：filter、mangle 以及 nat。每個篩表都被預先設置了一或多個代表各攔截點的串鏈（參閱《表 2》、《圖 1》、《圖2》、《圖3》）。這三個內建篩表的作用，請參閱《表 3》。

表 3：內建的篩表

名稱	用途說明
nat	依據「連線追蹤」（connection tracking）機制提供的線索來修改封包的來源位址或目的地位址，以達成「連線轉接」（redirect）的效果。nat 篩表內建的串鏈分別是 OUTPUT、POSTROUTING 與 PREROUTING。
filter	用於設定「入境、穿越、出境」三種性質的封包的處理原則。內建於 filter 篩表的串鏈包括 FORWARD、INPUT 與 OUTPUT。使用 iptables 時，如果沒刻意以 -t 指出要操作的篩表，iptables 就假設你要操作的是 filiter 篩表。
mangle	用於執行特殊的封包內容修改，例如裁掉 IP options（需搭配 IPV4OPTSSTRIP 擴充模組）。mangle 篩表內建的串鏈包括 FORWARD、INPUT、OUTPUT、POSTROUTING 和 PREROUTING。

iptables 將適當的串鏈設置於上述三個內建篩表，當系統收到封包時，便依據封包的來源（本機行程或網路）、來源主機的位址、目的地位址來判斷封包的性質（穿越、輸入、輸出、繞回），然後依照封包的性質，分別以《表 4》到《表 7》所列的程序之一來處理該類封包。

訣竅

圖 1 ~ 3 只是個別相關功能的示意流程，而非封包的實際旅程。表 4 ~ 7 所描述的順序，才是封包真正的旅程。

串鏈（Chains）

每個篩表都預設了相關攔截點的串鏈，當系統剛開機時，所有

串鏈都是空的；為此，iptables 套件另外提供了兩個輔助工具 — iptables-save 與 iptables-restore — 可供你儲存、回復所有規則（參閱《輔助工具》）。每個串鏈各代表一個攔截點，《表 2》是各攔截點的說明，《表 3》列出各篩表預設的串鏈。

除了內建串鏈之外，你也可以建立自己的串鏈來組織你的規則。

若封包能通過串鏈裡的每一條規則而不受影響，最後將由串鏈的**政策**（policy）決定封包的命運。內建串鏈只能以「內建目標」（參閱《表 8》）— ACCEPT 與 DROP — 為政策；預設政策為 ACCEPT。所有自訂串鏈的政策都固定是 RETURN，不能改變。

如果要為內建串鏈制定更複雜的政策，或是希望自訂串鏈改用 ETURN 之外的其它政策，唯一辦法是在串鏈末端添加一條通適規則（任何封包都符合條件的規則），並將你要的任何目標設定於該規則。

你不能直接修改已經設置到串鏈裡的現有規則，如果發現某規則有誤，唯一辦法是刪除掉舊規則（使用 -D）、然後將正確的新規則加回去（使用 -A 或 -I）。

封包流程

當封包流經串鏈時，必須依序通過該串鏈裡每一條規則的檢驗。若封包符合某條規則的「篩選條件」（match），則將封包交給該規則的「目標」（target）來處理，否則，就繼續由同串鏈裡的下一條規則予以檢驗。倘若封包順利通過串鏈裡的所有規則（不符合任何規則的篩選條件），則以串鏈的「政策」（policy，參閱《串鏈（Chains）》來決定其去向。

封包實際會經過哪些串鏈，取決於封包本身的性質（轉交、輸入、輸出、繞回），《表 4》到《表 7》分別列出各種性質的封包的旅程順序。《圖 1》、《圖 2》和《圖 3》是單看特定篩表時，封包如何通過該篩表各串鏈的詳細流程。

表4：從一個網路介面到另一個介面（forwarding，轉交）

篩表	串鏈
mangle	PREROUTING
nat	PREROUTING
mangle	FORWARD
filter	FORWARD
mangle	POSTROUTING
nat	POSTROUTING

表五：從網路介面進入本機行程（輸入）

篩表	串鏈
mangle	PREROUTING
nat	PREROUTING
mangle	INPUT
filter	INPUT

表六：從本機行程到網路介面（輸出）

篩表	串鏈
mangle	OUTPUT
nat	OUTPUT
filter	OUTPUT
mangle	POSTROUTING
nat	POSTROUTING

表七：從本機行程到另一個本機行程（本地）

篩表	串鏈
mangle	OUTPUT
nat	OUTPUT
filter	OUTPUT
filter	INPUT
mangle	INPUT

《附錄》描繪 filiter 篩表的詳細處理流程。

規則（Rules）

iptables 的每一條**規則**（rule），都是由兩部份組成，第一部份包含一或多個「篩選條件」，其作用是檢查封包是否符合處理條件（所有條件都必須成立才算數）；第二部份稱為「目標」，用於決定如何處置符合條件的封包。

對於每一條規則，iptables 各維護兩個計數器：一個計算符合條件的封包數，稱為 packet counter；另一個計算該規則所處理的總資料量，稱為 byte counter。每當有封包符合特定規則的篩選條件，該規則的 packet counter 便會被累加一，並將該封包的大小累加到該規則的 byte counter。

規則可以只有「篩選條件」或「目標」（處置方式）的其中之一；沒指定篩選條件時，則所有封包都算符合條件；沒指定處置方式時，則放任讓封包繼續其流程，也就是說，封包本身不會有任何改變，只有該規則的兩個 counters 會累增而已。使用下列命令可將這種只有計數作用的空規則加入 filiter 篩表的 FORWARD 串鏈：

```
iptables -t filter -A FORWARD
```

篩選條件（Matches）

iptables 可讓你設置多種篩選條件，但是某些條件需要核心有提供相關功能才行。iptables 本身內建一般性的 Internet Protocol（IP）篩選條件，也就是說，即時沒載入任何擴充模組，你也可以用 IP 封包標頭的「傳輸協定類型」、「來源位址」、「目的地位址」等欄位為篩選條件。關於一般性的 IP 篩選條件，請參閱《IPv4 篩選條件》。

除了 IP 之外的其它協定，諸如 ICMP、TCP、UDP 等等，必須載入相關的擴充模組，才可以作為篩選條件。使用 iptables 的 -m 或 --match 選項，指出載入特定協定的擴充模組。

所有關於篩選條件的擴充模組，幾乎都是針對網路層（IP、ICMP）或傳輸層（TCP、UDP），唯一例外是 mac 模組，它能讓你以 Ethernet 網卡的 Media Access Controller（MAC）位址為篩選條件【譯註：Ethernet 屬於網路層之下的「資料連結層」（OSI Model）或「實體層」（IP Model）】。

目標（Target）

「目標」（targets）決定如何處理符合篩選條件的封包，或是當成串鏈的政策。iptables 共內建四種目標（參閱《表 8》），除此之外的其它目標，必須透過擴充模組來提供。

表 8：iptables 內建的目標

目標	說明
ACCEPT	放任封包進入下一階段的處理程序。封包會停止目前串鏈的旅程，而開始圖 1～3（或表 4～7）所描述的下一階段流程。
DROP	終結封包的後續旅程。不再繼續檢查其它規則、串鏈、或篩表。如果你想提供一些回饋給封包來源，請改用 REJECT 目標擴充模組。
QUEUE	將封包傳入 userspace（核心之外的其它程式）。詳情請參閱 ibipq manpage。
RETURN	當封包流程是走到使用者自訂串鏈中的規則，則停止該串鏈的後續處理，回到先前呼叫該串鏈的規則的下一條規則。若封包流程是走到內建串鏈中的規則，則以該串鏈的政策處理該封包。關於串鏈政策，請參閱前一小節《串鏈（Chains）》。