IP Chains Quick Reference

Scott Bronson <bronson@soda.berkeley.edu>
ipchains v. 1.3.9 26 May 1999

Built-In Chains

ipchains provides three built-in chains. All packets start on one of these chains and, depending on the rules, may end up passing through all three.

input	Packets arriving over a network device on your machine from an external source. Locally generated packets come from the loopback device.
forward	Packets that are just passing through your machine. To be specific, if both the source and destination of this packet are remote, the packet is sent to the forwarding firewall.
output	Packets leaving your machine over a network device to an external source. Or, if sent to the loopback device, the packet reappears on the input chain.

Targets

Each rule has a target that tells what to with the packet upon a match. If a packet reaches the end of a chain without matching any rules, the chain's default target policy is invoked.

ACCEPT	Let the packet pass through the firewall
REJECT	Do not accept the packet. If it's not an ICMP packet, send an ICMP Host Unreachable reply to the sender.
DENY	Do not accept the packet, do not reply. Basically, ignore the sender.
MASQ	(forward chain only): Let the packet pass, but masquerade it. In other words, replace the sender of the packet with the local address and replace the source port in the header with a locally-generated temporary number.
REDIRECT	(input chain only): Send the packet to a local socket or process instead. Can be applied to TCP and UDP packets in input and user chains only.
RETURN	equivalent to falling off the end of the chain, either invoking a built-in chain's target policy or returning to a user chain's caller. Like a return statement in C.
(no target specified)	The rule's byte and packet counters will be incremented, but the packet will be passed on to the next rule in the chain even if this rule maches. Normally used for accounting.

If a target is used incorrectly (MASQ from an input chain for instance), the packet will be dropped and a message will be sent to the syslog.

Creating and Destroying Chains

chain specifies the name of a chain: input, output, forward, or the name of a user-defined chain. It may be up to 8 letters long and, to prevent conflicting with possible future built-in target names, should contain at least one lower-case letter.

-N *chain*	Create a new user-defined chain
-X *[chain]*	Delete an empty, user-defined chain (or all user-defined chains). All rules referring to a particular user chain must be deleted before the chain can be deleted. Built-in chains cannot be deleted.

Adding and Removing Rules

-v (verbose): when used with the following group of commands will cause the rule(s) to be echoed to stdout.

A *rule*, as specified on the command line, consists of zero or more predicates followed by zero or more actions. Predicates and actions are listed on the next page.

index is used to specify a rule by its position in the chain. The first rule in the chain has an index of 1.

-A *[-v] chain rule*	Append a rule to a chain
-D *[-v] chain index* -D *[-v] chain rule*	Delete a rule from a chain by position Delete first rule matching *rule* from *chain*.
-F *[chain]*	Flush all rules from *chain* (or all chains). Equivalent to deleting all rules one by one.
-I *[-v] chain index rule*	Insert *rule* into chain in front of the rule at *index*. So, if index is 1, *rule* becomes the first rule in the chain.
-R *[-v] chain index rule*	Replace the rule at *index* in *chain* with *rule*

Masquerading Commands

-M L *[-v]*	List currently masqueraded connections. *-v* will cause information related to delta sequence numbers to be dumped as well.
-M -S *tcp tcpfin udp*	Set masquerading timeout values. The parameters specify the timeout in seconds for TCP sessions, TCP sessions after receiving a FIN packet, and UDP packets respectively. A timeout of 0 will retain the previous timeout. Defaults are 15, 2, and 5 minutes.

Information

-h -h icmp	Prints a quick command summary. Lists ICMP names known to ipchains
--version	Prints the version of the ipchains command.

Chain Utilities

-P *chain target*	Change the policy for a built-in chain to *target*. *target* must be one of ACCEPT, DENY, REJECT, or MASQ (forward chain only). No other targets can be used as a built-in chain's policy.
-Z *[-L] [chain]*	Zero the packet and byte counters in *chain* (or all chains). -Z can be used in combination with -L to read the counters and then erase them, ensuring that no packets will go unaccounted. However, when doing this, you must zero all chains; you cannot atomically zero and list a single chain.
-L *[-vxn] [chain]*	List the rules in *chain* (or all chains). *-v* specifies verbose mode, *-x* expands numbers (displays exact values rather than using K, M, and G), *-n* causes ipchains to print numeric IP addresses rather than doing all the name lookups. Can be used with -Z above.
-C *[-v] chain pkt*	Test a packet on a *chain. pkt* is formed exactly like a *rule* except that it must contain at least the -s, -d, -p (must be a protocol, not ALL), and -i flags (unless, of course, -f is used to create a fragment). *-v* causes a trace of the path the packet took to be printed to stdout.

Actions

-j *target [port]*	Jump-to, specifies the target to be invoked when this rule matches. *target* may be one of the targets listed above, or the name of a user chain. You cannot jump to a built-in chain. If the packet is traversing the input chain, specify *port* for REDIRECT targets. If -j is omitted, the rule's packet counters will be incremented and processing will continue with the next rule in the chain.
-l	Specifies that matching packets should be logged in the syslog (careful—this could get out of hand quickly). Can be used alone or with -j.
-m *val*	Number to mark on packet. Used with the Quality of Service implementation in 2.1 kernels. If *val* begins with a + or -, then val will be added to or subtracted from the packet's mark value, rather than replacing it.
-o *[maxsize]*	Copies the packet to the userspace device. *maxsize* limits the number of bytes to be copied.
-t *and xor*	Changes the packet's TOS (Type of Service) bits. The packet's TOS is first ANDed by *and*, then XORed with *xor*. Both parameters are specified as 8-bit hexadecimal values. RFC 1349 specifies the LSB of the TOS field must be unaltered, so ipchains will refuse any rules that would violate that. RFC 1349 also specifies that only one bit in the TOS field may be set at a time. Rules that would cause more than one bit to be set are accepted with a warning sent to stdout.

Predicates

-p [!] *protocol*	Specifies a protocol. May be either the names, TCP, UDP, ICMP, or ALL (case doesn't matter), or the numeric IP protocol number (0 means ALL).
-s [!] *address* -d [!] *address*	The source address to match The destination address to match The address can be specified either by name (localhost or www.linuxhq.com) or IP address: (204.209.212.113). For an IP address, you can specify how much of the address is significant by specifying either the netmask (192.168.1.31/255.255.255.0) or the start of the host part (192.168.1.31/24). These two rules are equivalent and will match any IP address in the range 192.168.1.0 to 192.168.1.255 inclusive. When used with ICMP packets, you can specify the ICMP name after the -s option, or the numeric ICMP type after the -s option and code after the -d option. "ipchains -h icmp" to list the known ICMP names. ICMP names cannot currently be preceeded by a !.
[[!] *port*]	When used with the TCP or UDP protocols (-p tcp, -p udp), a port or range of ports can be specified. Simply specify the port number or name (i.e. www). A range can be specified using a colon to separate the low bound (0) from the high bound (65535).
[!] -y	When used with the TCP protocol, specifies TCP SYN packets only. This will match TCP packets requesting a TCP connection.
[!] -f	Matches the second through the last fragment of a fragmented packet. No ports may be specified when the -f flag is used.
-i [!] *name*	Specifies a network interface by name (as listed by the ifconfig command). The interface for input packets is the device they came in on, the interface for forward and output packets is the device they will leave on. An interface name ending with a + will match all interfaces of that type (ppp+ will match all ppp interfaces). It is legal to specify an interface name that does not currently exist--no packets will match until the interface comes up.
-b	Bidirectional, this will allow the rule to trigger no matter the packet's direction. It's equivalent to repeating the rule with the source and destination reversed (it actually inserts 2 rules in the chain). It's probably better just to declare the 2 rules explicitly.

[!] signifies that a rule may be inverted by preceding it with a bang. "! -s 10.64.127.128" will match any packet that does *not* originate from 10.64.127.128.