32 iptables examples to manage firewall

What is NAT "Network Address Translation" **STEPS**

1. PREROUTING
    1. ROUTING
        1. POSTROUTING
            1. OUT
    2. LOCALPROCESS
        1. POSTROUTING
            1. OUT

[crayon-669e20796d63d791322154/]
Input Rules — INPUT ACCEPT or Reject
[crayon-669e20796d649076315868/]

[crayon-669e20796d64b078496937/]

Listing

[crayon-669e20796d64e181672732/]

Delete Rules

[crayon-669e20796d650643072709/]

Forwarding

[crayon-669e20796d652843325670/]

Postrouting Masquareade

[crayon-669e20796d655589984181/]

Pre Routing

[crayon-669e20796d659662950747/]

1. Deny User
2. Allow port for a user.

Great
http://www.thegeekstuff.com/2011/06/iptables-rules-examples/
Kutay Zorlu   Introduction Iptables is a firewall that plays
an essential role in network security for most Linux systems.
While many iptables tutorials will teach you how to create
firewall rules to secure your server, this one will focus on a
different aspect of firewall management: listing and deleting
rules. In this tutorial, we will cover how to do the following
iptables tasks: List rules Clear Packet and Byte Counters
Delete rules Flush chains (delete all rules in a chain) Flush
all chains and tables, delete all chains, and accept all
traffic Note: When working with firewalls, take care not to
lock yourself out of your own server by blocking SSH traffic
(port 22, by default). If you lose access due to your firewall
settings, you may need to connect to it via the console to fix
your access. Once you are connected via the console, you can
change your firewall rules to allow SSH access (or allow all
traffic). If your saved firewall rules allow SSH access,
another method is to reboot your server. Prerequisites Before
you start using this tutorial, you should have a separate,
non-root superuser account—a user with sudo privileges—set up
on your server. If you need to set this up, follow the
appropriate guide: Initial Server Setup with Ubuntu 14.04
Initial Server Setup with CentOS 6 Let's look at how to list
rules first. There are two different ways to view your active
iptables rules: in a table or as a list of rule
specifications. Both methods provide roughly the same

information in different formats. List Rules by Specification To list out all of the active iptables rules by specification, run the iptables command with the -S option: sudo iptables -S Example: Rule Specification Listing

[crayon-669e20796d65c072597148/]

As you can see, the output looks just like the commands that were used to create them, without the preceding iptables command. This will also look similar to the iptables rules configuration files, if you've ever used iptables-persistent or iptables save. List Specific Chain If you want to limit the output to a specific chain (INPUT, OUTPUT, TCP, etc.), you can specify the chain name directly after the -S option. For example, to show all of the rule specifications in the TCP chain, you would run this command: sudo iptables -S TCP Example: TCP Chain Rule Specification Listing -N TCP -A TCP -p tcp -m tcp —dport 22 -j ACCEPT Let's take a look at the alternative way to view the active iptables rules, as a table of rules. List Rules as Tables Listing the iptables rules in the table view can be useful for comparing different rules against each other, To output all of the active iptables rules in a table, run the iptables command with the -L option: sudo iptables -L This will output all of current rules sorted by chain. If you want to limit the output to a specific chain (INPUT, OUTPUT, TCP, etc.), you can specify the chain name directly after the -L option. Let's take a look at an example INPUT chain: sudo iptables -L INPUT Example: Input Chain Rule Table Listing Chain INPUT (policy DROP) target prot opt source destination ACCEPT all — anywhere anywhere ctstate RELATED,ESTABLISHED ACCEPT all — anywhere anywhere DROP all — anywhere anywhere ctstate INVALID UDP udp — anywhere anywhere ctstate NEW TCP tcp — anywhere anywhere tcp flags:FIN,SYN,RST,ACK/SYN ctstate NEW ICMP icmp — anywhere anywhere ctstate NEW REJECT udp — anywhere anywhere reject-with icmp-port-unreachable REJECT tcp — anywhere anywhere reject-with tcp-reset REJECT all — anywhere anywhere reject-with icmp-proto-unreachable The first line of output indicates the chain name (INPUT, in this case), followed by its default

policy (DROP). The next line consists of the headers of each column in the table, and is followed by the chain's rules. Let's go over what each header indicates: target: If a packet matches the rule, the target specifies what should be done with it. For example, a packet can be accepted, dropped, logged, or sent to another chain to be compared against more rules prot: The protocol, such as tcp, udp, icmp, or all opt: Rarely used, this column indicates IP options source: The source IP address or subnet of the traffic, or anywhere destination: The destination IP address or subnet of the traffic, or anywhere The last column, which is not labeled, indicates the options of a rule. That is, any part of the rule that isn't indicated by the previous columns. This could be anything from source and destination ports, to the connection state of the packet. Show Packet Counts and Aggregate Size When listing iptables rules, it is also possible to show the number of packets, and the aggregate size of the packets in bytes, that matched each particular rule. This is often useful when trying to get a rough idea of which rules are matching against packets. To do so, simply use the -L and -v option together. For example, let's look at the INPUT chain again, with the -v option: sudo iptables -L INPUT -v Example: Verbose Listing Chain INPUT (policy DROP 0 packets, 0 bytes) pkts bytes target prot opt in out source destination 284K 42M ACCEPT all — any any anywhere anywhere ctstate RELATED,ESTABLISHED 0 0 ACCEPT all — lo any anywhere anywhere 0 0 DROP all — any any anywhere anywhere ctstate INVALID 396 63275 UDP udp — any any anywhere anywhere ctstate NEW 17067 1005K TCP tcp — any any anywhere anywhere tcp flags:FIN,SYN,RST,ACK/SYN ctstate NEW 2410 154K ICMP icmp — any any anywhere anywhere ctstate NEW 396 63275 REJECT udp — any any anywhere anywhere reject-with icmp-port-unreachable 2916 179K REJECT all — any any anywhere anywhere reject-with icmp-proto-unreachable 0 0 ACCEPT tcp — any any anywhere anywhere tcp dpt:ssh ctstate NEW,ESTABLISHED Note that the listing now has two additional columns, pkts and bytes. Now that you know how to list the active firewall rules in a

variety of ways, let's look at how you can reset the packet and byte counters. Reset Packet Counts and Aggregate Size If you want to clear, or zero, the packet and byte counters for your rules, use the -Z option. They also reset if a reboot occurs. This is useful if you want to see if your server is receiving new traffic that matches your existing rules. To clear the counters for all chains and rules, use the -Z option by itself: sudo iptables -Z To clear the counters for all rules in a specific chain, use the -Z option and specify the chain. For example, to clear the INPUT chain counters run this command: sudo iptables -Z INPUT If you want to clear the counters for a specific rule, specify the chain name and the rule number. For example, to zero the counters for the 1st rule in the INPUT chain, run this: sudo iptables -Z INPUT 1 Now that you know how to reset the iptables packet and byte counters, let's look at the two methods that can be used to delete them. Delete Rule by Specification One of the ways to delete iptables rules is by rule specification. To do so, you can run the iptables command with the -D option followed by the rule specification. If you want to delete rules using this method, you can use the output of the rules list, iptables -S, for some help. For example, if you want to delete the rule that drops invalid incoming packets (-A INPUT -m conntrack —ctstate INVALID -j DROP), you could run this command: sudo iptables -D INPUT -m conntrack —ctstate INVALID -j DROP Note that the -A option, which is used to indicate the rule position at creation time, should be excluded here. Delete Rule by Chain and Number The other way to delete iptables rules is by its chain and line number. To determine a rule's line number, list the rules in the table format and add the —line-numbers option: sudo iptables -L —line-numbers [secondary_output Example Output: Rules with Line Numbers] Chain INPUT (policy DROP) num target prot opt source destination 1   ACCEPT all — anywhere anywhere ctstate RELATED,ESTABLISHED 2   ACCEPT all — anywhere anywhere 3 DROP all — anywhere anywhere ctstate INVALID 4   UDP udp — anywhere anywhere ctstate NEW 5   TCP tcp — anywhere anywhere

tcp flags:FIN,SYN,RST,ACK/SYN ctstate NEW 6   ICMP icmp — anywhere anywhere ctstate NEW 7   REJECT udp — anywhere anywhere reject-with icmp-port-unreachable 8   REJECT tcp — anywhere anywhere reject-with tcp-reset 9   REJECT all — anywhere anywhere reject-with icmp-proto-unreachable 10   ACCEPT tcp — anywhere anywhere tcp dpt:ssh ctstate NEW,ESTABLISHED … This adds the line number to each rule row, indicated by the num header. Once you know which rule you want to delete, note the chain and line number of the rule. Then run the iptables -D command followed by the chain and rule number. For example, if we want to delete the input rule that drops invalid packets, we can see that it's rule 3 of the INPUT chain. So we should run this command: sudo iptables -D INPUT 3 Now that you know how to delete individual firewall rules, let's go over how you can flush chains of rules. Flush Chains Iptables offers a way to delete all rules in a chain, or flush a chain. This section will cover the variety of ways to do this. Note: Be careful to not lock yourself out of your server, via SSH, by flushing a chain with a default policy of drop or deny. If you do, you may need to connect to it via the console to fix your access. Flush a Single Chain To flush a specific chain, which will delete all of the rules in the chain, you may use the -F, or the equivalent —flush, option and the name of the chain to flush. For example, to delete all of the rules in the INPUT chain, run this command: [crayon-669e20796d662770308530/]
Flush All Chains To flush all chains, which will delete all of the firewall rules, you may use the -F, or the equivalent —flush, option by itself: sudo iptables -F Flush All Rules, Delete All Chains, and Accept All This section will show you how to flush all of your firewall rules, tables, and chains, and allow all network traffic. Note: This will effectively disable your firewall. You should only follow this section if you want to start over the configuration of your firewall. First, set the default policies for each of the built-in chains to ACCEPT. The main reason to do this is to ensure that you won't be locked out from your server via SSH:

[crayon-669e20796d664672055105/]
Then flush the nat and mangle tables, flush all chains (-F),
and delete all non-default chains (-X):
[crayon-669e20796d667434507664/]
Your firewall will now allow all network traffic. If you list
your rules now, you will will see there are none, and only the
three default chains (INPUT, FORWARD, and OUTPUT) remain.
[crayon-669e20796d669255086488/]

PRE ROutING and OTHERS
[crayon-669e20796d66c820358587/]