Table Of Contents

nixCraft: Linux Tips, Hacks, Tutorials, And Ideas In Blog Format
http://www.cyberciti.biz/

OpenVZ Iptables: Allow Traffic To Pass Via venet0 To All VPS

Posted by Vivek Gite <vivek@nixcraft.com>

This entry is part 4 of 5 in the series RHEL / CentOS OpenVZ Virtualization [1]

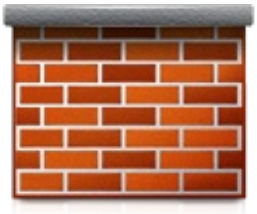
RHEL / CentOS OpenVZ Virtualization [1]

- How To Setup OpenVZ under RHEL / CentOS Linux [2]
- CentOS Linux Install OpenVZ Virtualization Software [3]
- How To Create OpenVZ Virtual Machines (VPS) [4]
- OpenVZ Iptables: Allow Traffic To Pass Via venet0 To All VPS
- OpenVZ Virtual Machine (VPS) Management [5]

How do I configure IPTABLES to pass all traffic to my VPS (container) under hardware node?

venet0 is recommend networking for security and performance under OpenVZ Virtualization. Protecting hardware node is important from unauthorized access. venet0 is used to communicate between VPS and the LAN / Internet.

[7]



[6]

```
Router
   \\
     \\
Hardware Node - eth0
            //
           //
        venet0
+----------+------------+
|          |            |
vps1      vps2          vps3
```

Allow All Traffic To VPS

Following iptables rules allows to pass all traffic between hardware node and all vps / containers. Services running on hardware node such as ssh, http, webmin can only accessed within our LAN and not over the Internet.

[8]

```
#!/bin/bash
# Explains how to setup iptables on the hardware node to allow selective access,
# but allow all traffic into the containers (VPS) so they may define their own iptables ru
# therefore manage their own firewall.
# Author: Vivek Gite < http://www.cyberciti.biz/ >
# See tutorial : http://www.cyberciti.biz/faq/series/rhel-centos-openvz-virtualization/
# This script is under GPL v2.0 or above.
# ----------------------------------------------------------------------------------------
IPT="/sbin/iptables"
MOP="/sbin/modprobe"
SYST="/sbin/sysctl"

### ********************************************************************** ###
### Part 1 - Protect Hardware Node          ###
### ********************************************************************** ###

### HW Node Main IP ranges ###
SRVIP="123.xx.xx.yy"
ADMIN_RANGES="192.168.1.0/24"
```

```
SPOOFIP="127.0.0.0/8 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 169.254.0.0/16 0.0.0.0/8 240.0

### Path to other scripts ###
[ -f /root/fw/blocked.ip.txt ] && BADIPS=$(egrep -v -E "^#|^$" /root/fw/blocked.ip.txt)

### Interfaces ###
PUB_IF="eth0"   # public interface
LO_IF="lo"      # loopback
VE_IF="venet0"

### start firewall ###
echo "Starting Firewall..."
$IPT -F
$IPT -X
$IPT -t nat -F
$IPT -t nat -X
$IPT -t mangle -F
$IPT -t mangle -X
$IPT -P INPUT ACCEPT
$IPT -P OUTPUT ACCEPT
$IPT -P FORWARD ACCEPT

# Enable ip_conntrack
$MOP ip_conntrack

# DROP and close everything all incomming traffic
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

# Unlimited lo access
$IPT -A INPUT -i ${LO_IF} -j ACCEPT
$IPT -A OUTPUT -o ${LO_IF} -j ACCEPT

# Allow Full Outgoing connection but no incomming stuff by default
$IPT -A OUTPUT -o ${PUB_IF} -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -i ${PUB_IF} -m state --state ESTABLISHED,RELATED -j ACCEPT

# Drop bad stuff
# get all bad spam / scrap ips
if [ -f /root/fw/blocked.ip.txt ];
then
 $IPT -N spamlist
 for ipblock in $BADIPS
 do
   $IPT -A spamlist -i ${PUB_IF} -s $ipblock -j LOG --log-prefix "SPAM List Block"
   $IPT -A spamlist -i ${PUB_IF} -s $ipblock -j DROP
 done
 $IPT -I INPUT -j spamlist
 $IPT -I OUTPUT -j spamlist
 $IPT -I FORWARD -j spamlist
done

$IPT -N spooflist
for ipblock in $SPOOFIP
do
 $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j LOG --log-prefix "SPOOF List Block"
 $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j DROP
done
$IPT -I INPUT -j spooflist
$IPT -I OUTPUT -j spooflist
$IPT -I FORWARD -j spooflist

# Stop sync
$IPT -A INPUT -i ${PUB_IF} -p tcp ! --syn -m state --state NEW -j DROP

# Stop Fragments
$IPT -A INPUT -i ${PUB_IF} -f -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
```

```
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL ALL -j DROP

# Stop NULL packets
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -m limit --limit 5/m --limit-burst
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

# Stop XMAS
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 5/m --limit
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Stop FIN packet scans
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -m limit --limit 5/m --limit-bur
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

# Get rid of broadcast
$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type broadcast -j DROP
$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type multicast -j DROP
$IPT  -A INPUT -i ${PUB_IF} -m state --state INVALID -j DROP

# allow SSH, HTTP, HTTPD and webmin ONlY from $ADMIN_RANGES
$IPT -A INPUT -i ${PUB_IF} -s ${ADMIN_RANGES} -d ${SRVIP} -p tcp --destination-port 22 -j A
$IPT -A INPUT -i ${PUB_IF} -s ${ADMIN_RANGES} -d ${SRVIP} -p tcp --destination-port 10000 -
$IPT -A INPUT -i ${PUB_IF} -s ${ADMIN_RANGES} -d ${SRVIP} -p tcp --destination-port 80 -j A
$IPT -A INPUT -i ${PUB_IF} -s ${ADMIN_RANGES} -d ${SRVIP} -p tcp --destination-port 443 -j

# Allow incomming ICMP ping pong stuff
$IPT -A INPUT -i ${PUB_IF} -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED -
$IPT -A INPUT -i ${PUB_IF}  -p icmp -m icmp --icmp-type 3 -m limit --limit 30/sec -j ACCEPT
$IPT -A INPUT -i ${PUB_IF}  -p icmp -m icmp --icmp-type 5 -m limit --limit 30/sec -j ACCEPT
$IPT -A INPUT -i ${PUB_IF}  -p icmp -m icmp --icmp-type 11 -m limit --limit 30/sec -j ACCEP

### ********************************************************************** ###
### Part 1 - Protect Hardware Node END          ###
### ********************************************************************** ###

### ********************************************************************** ###
### Part 2 - ALL VPS Specifc Config          ###
### ********************************************************************** ###

# Allow all ports for all VPS i.e. full access
# user can set their own firewall inside vps
$IPT -P FORWARD ACCEPT
$IPT -F FORWARD

### ********************************************************************** ###
### Part 2 - ALL VPS Specifc Config END          ###
### ********************************************************************** ###

# drop and log everything else
$IPT -A INPUT -m limit --limit 5/m --limit-burst 7 -j LOG
$IPT -A INPUT -j REJECT --reject-with icmp-port-unreachable

exit 0
```

Install this script at /root/fw/firewall:

```
# chmod +x /root/fw/firewall
```

Call it from /etc/rc.local

```
# echo '/root/fw/firewall' >> /etc/rc.local
```

Series Navigation

«How To Create OpenVZ Virtual Machines (VPS) [4] OpenVZ Virtual Machine (VPS) Management» [5]

Article printed from Frequently Asked Questions About Linux / UNIX: **http://www.cyberciti.biz/faq/**

URL to article: **http://www.cyberciti.biz/faq/centos-rhel-linux-openvz-hardware-node-iptables-firewall/**

URLs in this post:

[1] RHEL / CentOS OpenVZ Virtualization: **http://www.cyberciti.biz/faq/series/rhel-centos-openvz-virtualization/**
[2] How To Setup OpenVZ under RHEL / CentOS Linux: **http://www.cyberciti.biz/faq/openvz-rhel-centos-linux-tutorial/**
[3] CentOS Linux Install OpenVZ Virtualization Software: **http://www.cyberciti.biz/faq/rhel-redhat-centos-setup-openvz-virtualization/**
[4] How To Create OpenVZ Virtual Machines (VPS): **http://www.cyberciti.biz/faq/centos-rhel-redhat-create-openvz-virtual-machines-vps/**
[5] OpenVZ Virtual Machine (VPS) Management: **http://www.cyberciti.biz/faq/openvz-virtual-machine-vps-management/**
[6] Image: **http://www.cyberciti.biz/faq/category/iptables/**
[7] Image: **http://www.cyberciti.biz/faq/category/centos/**
[8] Image: **http://www.cyberciti.biz/faq/category/redhat-and-friends/**