Subject: iptables auf hardware node
Posted by faithless on Mon, 03 Mar 2008 17:44:49 GMT
View Forum Message <> Reply to Message

sehr geehrte openvz-community

habe folgendes iptables script auf meiner hw-node laufen, welche an eth0 eine öffentliche und an eth1 eine interne ip adresse gebunden hat.

```
#!/bin/bash

case "$1" in
  start)
    echo "Starte IP-Paketfilter"

    # iptables-Modul
    modprobe ip_tables
    # Connection-Tracking-Module
    modprobe ip_conntrack
    # Das Modul ip_conntrack_irc ist erst bei Kerneln >= 2.4.19 verfuegbar
    modprobe ip_conntrack_irc
    modprobe ip_conntrack_ftp

    # Tabelle flushen
    iptables -F
    iptables -t nat -F
    iptables -t mangle -F
    iptables -X
    iptables -t nat -X
    iptables -t mangle -X

    # Default-Policies setzen
    iptables -P INPUT DROP
    iptables -P OUTPUT DROP
    iptables -P FORWARD DROP

    # MY_REJECT-Chain
    iptables -N MY_REJECT

    # MY_REJECT fuellen
    iptables -A MY_REJECT -p tcp -j REJECT --reject-with tcp-reset
    iptables -A MY_REJECT -p udp -j REJECT --reject-with icmp-port-unreachable
    iptables -A MY_REJECT -p icmp -j DROP
    iptables -A MY_REJECT -j REJECT --reject-with icmp-proto-unreachable

    # MY_DROP-Chain
    iptables -N MY_DROP
    iptables -A MY_DROP -j DROP
```

```
# Korrupte Pakete zurueckweisen
iptables -A INPUT -m state --state INVALID -j DROP
iptables -A OUTPUT -m state --state INVALID -j DROP

# Stealth Scans etc. DROPpen
# Keine Flags gesetzt
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j MY_DROP

# SYN und FIN gesetzt
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j MY_DROP

# SYN und RST gleichzeitig gesetzt
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j MY_DROP

# FIN und RST gleichzeitig gesetzt
iptables -A INPUT -p tcp --tcp-flags FIN,RST FIN,RST -j MY_DROP

# FIN ohne ACK
iptables -A INPUT -p tcp --tcp-flags ACK,FIN FIN -j MY_DROP

# PSH ohne ACK
iptables -A INPUT -p tcp --tcp-flags ACK,PSH PSH -j MY_DROP

# URG ohne ACK
iptables -A INPUT -p tcp --tcp-flags ACK,URG URG -j MY_DROP

# Loopback-Netzwerk-Kommunikation zulassen
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Connection-Tracking aktivieren
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# ICMP Echo-Request (ping) zulassen und beantworten
iptables -A INPUT -m state --state NEW -p icmp --icmp-type echo-request -j ACCEPT

# LAN-Zugriff auf eth1
iptables -A INPUT -m state --state NEW -i eth1 -j ACCEPT

# Default-Policies mit REJECT
iptables -A INPUT -j MY_REJECT
iptables -A OUTPUT -j MY_REJECT

# SYN-Cookies
echo 1 > /proc/sys/net/ipv4/tcp_syncookies 2> /dev/null
```

```
  # Stop Source-Routing
  for i in /proc/sys/net/ipv4/conf/*; do echo 0 > $i/accept_source_route 2> /dev/null; done

  # Stop Redirecting
 for i in /proc/sys/net/ipv4/conf/*; do echo 0 > $i/accept_redirects 2> /dev/null; done

  # Reverse-Path-Filter
  for i in /proc/sys/net/ipv4/conf/*; do echo 1 > $i/rp_filter 2> /dev/null; done

  # Log Martians
  for i in /proc/sys/net/ipv4/conf/*; do echo 1 > $i/log_martians 2> /dev/null; done

  # BOOTP-Relaying ausschalten
  for i in /proc/sys/net/ipv4/conf/*; do echo 0 > $i/bootp_relay 2> /dev/null; done

  # Proxy-ARP ausschalten
  for i in /proc/sys/net/ipv4/conf/*; do echo 0 > $i/proxy_arp 2> /dev/null; done

  # UngÃ¼ ICMP-Antworten ignorieren
  echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses 2> /dev/null

  # ICMP Echo-Broadcasts ignorieren
  echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts 2> /dev/null

  # Max. 500/Sekunde (5/Jiffie) senden
  echo 5 > /proc/sys/net/ipv4/icmp_ratelimit

  # Speicherallozierung und -timing fÃ¼De/-Fragmentierung
  echo 262144 > /proc/sys/net/ipv4/ipfrag_high_thresh
  echo 196608 > /proc/sys/net/ipv4/ipfrag_low_thresh
  echo 30 > /proc/sys/net/ipv4/ipfrag_time

  # TCP-FIN-Timeout zum Schutz vor DoS-Attacken setzen
  echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout

  # Maximal 3 Antworten auf ein TCP-SYN
  echo 3 > /proc/sys/net/ipv4/tcp_retries1

  # TCP-Pakete maximal 15x wiederholen
  echo 15 > /proc/sys/net/ipv4/tcp_retries2

  ;;

stop)
  echo "Stoppe IP-Paketfilter"
  # Tabelle flushen
  iptables -F
  iptables -t nat -F
```

```
    iptables -t mangle -F
    iptables -X
    iptables -t nat -X
    iptables -t mangle -X
    # Default-Policies setzen
    iptables -P INPUT ACCEPT
    iptables -P OUTPUT ACCEPT
    iptables -P FORWARD ACCEPT
    ;;

  status)
    echo "Tabelle filter"
    iptables -L -vn
    echo "Tabelle nat"
    iptables -t nat -L -vn
    echo "Tabelle mangle"
    iptables -t mangle -L -vn
    ;;

  *)
    echo "Fehlerhafter Aufruf"
    echo "Syntax: $0 {start|stop|status}"
    exit 1
    ;;

esac
```

sobald die iptables regeln aktiv sind, kann meine virtuelle maschine (die ebenfalls eine öffentliche ip adresse besitzt) nicht mehr mit dem netz kommunizieren. lediglich die kommunikation zwischen hw-node und virtueller maschine ist dann noch möglich.

eine ahnung welche regeln ich noch hinzufügen muss, um meiner virtuellen maschine uneingeschränkten zugriff aufs netz zu erlauben (absicherung würde dann per iptables auf der vm selber konfiguriert werden).

vielen dank,
joseph h.

ps: handelt sich um selbes problem wie bereits hier beschrieben.

Subject: Re: iptables auf hardware node
Posted by faithless on Wed, 12 Mar 2008 16:06:27 GMT
View Forum Message <> Reply to Message

hm... weiß denn keiner die lösung?

unter http://wiki.openvz.org/Setting_up_an_iptables_firewall ist folgendes vermerkt:

"Setting up a firewall that allows per-container configuration

This setup configures iptables on the HN to disallow access to all hosts, including the containers. However, it allows all traffic into the containers so they may define their own iptables rules and therefore manage their own firewall.

This content is missing. You are invited to fill it in, if you get to it before I do. "

irgendjemand muss doch wissen, wie das funktioniert?!
liebe grüße,
joseph
