Scripts/ssh failed

Version: 1737

Table of Contents

RichardJune/Scripts/ssh failed.....1

RichardJune/Scripts/ssh failed

```
#!/bin/bash
# This script is meant to be run once an hour. It will index failed login attempts from
#  the previous hour. After BADCOUNT attempts, it will drop all traffic from that IP
#  address via the ssh_failed chain. To make your router actually filter traffic based on
#  this script, use "iptables --append INPUT --jump ssh_failed" in your firewall script.
# BLOCKPOINTS is the score (or failed attempts) required before an address is blocked.
# POINTLOSS is how many points are forgiven each time this script runs.
# INSTALLATION:
# Add this to the cron configuration.
# 0 * * * *       /root/bin/ssh_failed
#
# Add this to the firewall configuration
# /root/bin/ssh_failed
# iptables --append INPUT --jump ssh_failed

BLOCKPOINTS="10"
POINTLOSS="2"

# This stuff is not really user modifiable.
#  It is used to specify search information, etc.
LOGFILE="/var/log/syslog*"
CHAINNAME="ssh_failed"
BLOCKDIR="/tmp/${CHAINNAME}"
MONTH="$(date +%b)"
DAY="$(date +%e)"
HOUR="$(date +%H)"
DAYSEARCH="$MONTH $DAY"
HOURSEARCH="$MONTH $DAY $(dc $HOUR 1 -)"
mkdir -p ${BLOCKDIR}
iptables --new ${CHAINNAME} >/dev/null 2>&1
#iptables --flush ${CHAINNAME}

# First remove two points from the score of each IP each time this script runs.
#  If the score reaches zero, unblock the addres.
CWD="$(pwd)"
cd ${BLOCKDIR}
ls * > /dev/null 2>&1
if [ $? -eq 0 ] ; then
  for ip in * ; do
    count="$(cat ${ip})"
    if [ $count -gt 0 ] ; then
      let count=count-${POINTLOSS}
      if [ ${count} -lt 0 ] ; then
        count=0
      fi
      if [ ${count} -eq 0 ] ; then
        echo "traffic now allowed from sshd: "$ip
       iptables --delete ${CHAINNAME} --jump DROP --source ${ip} >/dev/null 2>&1
      fi
      echo ${count} > ${ip}
    fi
  done
fi
# Next get a list of IP addresses that attempted to login with an invalid username
```

```
INVALID_USER_IP="$(grep "${HOURSEARCH}" ${LOGFILE} | grep "Failed password for invalid user" |awk '{prin
# Next get a list of IP addresses that failed to login as root
ROOT_IP="$(grep "${HOURSEARCH}" ${LOGFILE} | grep "Failed password for root" |awk '{print $11}')"

# Count the number of addresses and keep score. If the score gets above BLOCKPOINTS, shut it down.
for ip in ${ROOT_IP} ${INVALID_USER_IP} ; do
  if [ ! -e "${BLOCKDIR}/${ip}" ] ; then
    echo 0 > ${BLOCKDIR}/${ip}
  fi
  count="$(cat ${BLOCKDIR}/${ip})"
  let count++
  echo ${count} > ${BLOCKDIR}/${ip}
  if [ ${count} -eq ${BLOCKPOINTS} ] ; then
    echo "banned from sshd: "$ip
    iptables --append ${CHAINNAME} --jump DROP --source ${ip} > /dev/null 2>&1
  fi
done
```