

Call Loop Prevention

SIP has loop detection and prevention built-in. However, when you are interfacing with legacy technology through a gateway there is a very real risk that a loop can be setup between the two systems. As you add users to the SIPX install, SIPX first looks at current registrations, then will route calls through the dial-plan. If through an accounting error, your TDM systems sends calls to SIPX and there is no matching extension in SIPX then the call is routed back out to the TDM system, which routes it back into SIPX...you get the idea. This can result in calls that go above 40,000/hour! It will bring your proxies to their knees. Below is a simple script you can run from cron to block calls.

callloop.sh

```
#!/bin/bash
maxcall=75
email='khaefner@colostate.edu'
subject='Potential Call Loop'
date="$(date +%s)"

#create a white list
declare -a white_list=(15500 0 2)

#parse iptables looking to see if there is a block already
iptables_output=$(iptables -L INPUT -n --line-numbers | grep BLOCKED)

#get what is blocked into arrays from comments
ID1='\n' read -a blocked_numbers <<<$(sed -e 's/^.*BLOCKED:\([0-9]*\)*/\1/' <<<"$iptables_output")
ID2='\n' read -a blocked_dates <<<$(sed -e 's/^.*WHEN:\([a-zA-Z0-9]*\)|\s*\)/\1/' <<<"$iptables_output")
ID3='\n' read -a blocked_lines <<<$(sed -e 's/^([0-9]*)\s*/\1/' <<<"$iptables_output")

#Look through CDR to see if there are any calls
output=$(echo "select callee_aor, count(*) from view cdrs where start_time > CURRENT_TIMESTAMP AT TIME ZONE
'UTC' - interval '1' minute GROUP BY callee_aor order by count DESC LIMIT 1" | psql -t -q --user provision
SIPXCDR)
caller=$(sed -e 's/^.*sip:\([0-9]*\)*/\1/' <<<"$output")
callsperminute=$(sed -e 's/^.*|\s*\([0-9]*\)/\1/' <<<"$output")
message="The number $caller has been blocked because it has made $callsperminute calls in the last minute.
Calls for this number will be blocked for the next 10 minutes."

#Check if we have blocked the number preVIOUSLY if we have check timestamp if greater than 600 seconds remove
rule
for i in ${!blocked_numbers[*]}; do
    timediff=$((date-${blocked_dates[$i]})
    if [ "$timediff" -gt "600" ]
    then
        iptables -D INPUT -i eth0 -p udp -m udp --dport 5060 -m string --string "INVITE
sip:${blocked_numbers[$i]}" --algo bm --to 65 -j DROP -m comment --comment "BLOCKED:${blocked_numbers[$i]}
WHEN:${blocked_dates[$i]}"
    fi
done

#check if we have already block the number so we don't block again
for i in ${!blocked_numbers[*]}; do
    if [ "$caller" = "${blocked_numbers[$i]}" ]
    then
        exit
    fi
    for w in ${white_list[@]}; do
        if [ "$w" = "${blocked_numbers[$i]}" ]
        then
            echo $w
            exit
        fi
    done
done
```

```
done

if [ -n "$callsperminute" ]
then
    if [ -n "$caller" ]
    then
        if [ "$callsperminute" -gt "$maxcall" ]
        then
            echo $message
            #echo $message | /bin/mail -s "$subject" "$email"
            iptables -I INPUT -i eth0 -p udp -m udp --dport 5060 -m string --string "INVITE sip:$caller" --
        algo bm --to 65 -j DROP -m comment --comment "BLOCKED:$caller WHEN:$date"
        fi
    fi
fi
```

You will also need a file in the root directory of the user running this (likely root for iptables) file: `.pgpass`

```
~/.pgpass
```

```
localhost:*:SIPXCDR:sipxcdruser:secret
```