

## Using IPTables edit rules with Dynamic IP hostnames like dyndns.org

Written by BiRU

Tuesday, 11 July 2017 14:08 - Last Updated Tuesday, 11 July 2017 15:45

---

Whenever IPTables has a hostname in a rule it looks up the hostname's IP address and uses that instead of the actual hostname – so it's stuck with the IP until the next time IPTables is flushed/restarted. Here's a quick little python script to stick in a crontab which checks the IP of your dynamic IP hostname (free ones provided by dyndns.org) and will restart iptables if it catches a change in your hostname. The script was made for CentOS so should work on Red Hat based distributions – if you don't have an /etc/init.d/iptables file you'll have to modify the reload iptables command in the source. [Viewable Source After Jump](#)

I just set this up as root and in root's crontab.

[Download Source](#)

### Source:

```
#!/usr/bin/python                                     import                                     os
```

### Output looks like:

```
Log: example.dyndns.org has address 114.76.31.112. With Current IP: 114.76.31.112. Rebuilding iptables as updates Flt
```

Be Sociable, Share!

So I couldn't get the dns name in the IP tables to work so I took your idea and wrote a bash script that looks up the host name adds it to the iptables and if the IP changes removes the old

## Using IPTables edit rules with Dynamic IP hostnames like dyndns.org

Written by BiRU

Tuesday, 11 July 2017 14:08 - Last Updated Tuesday, 11 July 2017 15:45

---

rule and adds a new one for the new ip.

```
#!/bin/bash
```

```
#allow a dyndns name
```

```
HOSTNAME=HOST_NAME_HERE
```

```
LOGFILE=LOGFILE_NAME_HERE
```

```
Current_IP=$(host $HOSTNAME | cut -f4 -d' ')
```

```
if [ $LOGFILE = "" ] ; then
```

```
iptables -I INPUT -i eth1 -s $Current_IP -j ACCEPT
```

```
echo $Current_IP > $LOGFILE
```

```
else
```

```
Old_IP=$(cat $LOGFILE)
```

```
if [ "$Current_IP" = "$Old_IP" ] ; then
```

```
echo IP address has not changed
```

```
else
```

```
iptables -D INPUT -i eth1 -s $Old_IP -j ACCEPT
```

```
iptables -I INPUT -i eth1 -s $Current_IP -j ACCEPT
```

```
echo $Current_IP > $LOGFILE
```

```
echo iptables have been updated
```

```
fi
```

```
fi
```

then just add this line to your crontab and it will check every 5 mins and keep your iptables up-to-date.

```
*/5 * * * * /root/NAME_OF_SCRIPT.sh > /dev/null 2>&1
```