

Traffic counting on the CCGX

There are situation in which it is interesting to know the traffic usage of the color control. This can be done on the CCGX using [iptables](#). To do so requires a number of changes on the CCGX, which are described here.

The rules

The following commands clear the existing rules from iptables and add rules for counting specific traffic.

```
#Flush all
iptables -F
#Delete all user defined chains
iptables -X

#ACCEPT traffic to local host
iptables -I INPUT -s 127.0.0.0/8 -j ACCEPT
iptables -I OUTPUT -d 127.0.0.0/8 -j ACCEPT
#ACCEPT traffic to the local LAN
iptables -I INPUT -s 10.0.0.0/8 -j ACCEPT
iptables -I OUTPUT -d 10.0.0.0/8 -j ACCEPT
iptables -I INPUT -s 172.16.0.0/12 -j ACCEPT
iptables -I OUTPUT -d 172.16.0.0/12 -j ACCEPT
iptables -I INPUT -s 192.168.0.0/16 -j ACCEPT
iptables -I OUTPUT -d 192.168.0.0/16 -j ACCEPT

#ACCEPT multicast traffic
iptables -I INPUT -d 224.0.0.0/4 -j ACCEPT
iptables -I OUTPUT -d 224.0.0.0/4 -j ACCEPT

#ACCEPT broadcast traffic (dhcp)
iptables -I INPUT -d 255.255.255.255 -j ACCEPT
iptables -I OUTPUT -d 255.255.255.255 -j ACCEPT

#Send all remaining traffic to OTHER_IN/OUT and log
iptables -N OTHER_IN
#The following line can oly be used when the xt_limit kernel module is
available
#iptables -A OTHER_IN -j LOG -m limit --limit 10/hour --log-prefix
"IPTables-OTHER_IN: " --log-level 7
iptables -A OTHER_IN -j ACCEPT
iptables -N OTHER_OUT
#The following line can oly be used when the xt_limit kernel module is
available
#iptables -A OTHER_OUT -j LOG -m limit --limit 10/hour --log-prefix
"IPTables-OTHER_IN: " --log-level 7
```

```
iptables -A OTHER_OUT -j ACCEPT
iptables -A INPUT -g OTHER_IN
iptables -A OUTPUT -g OTHER_OUT

iptables -N REMOTE_SUPPORT_IN
iptables -I REMOTE_SUPPORT_IN -j ACCEPT
iptables -N REMOTE_SUPPORT_OUT
iptables -I REMOTE_SUPPORT_OUT -j ACCEPT
iptables -I OTHER_IN -s suppothost.victronenergy.com -g REMOTE_SUPPORT_IN
iptables -I OTHER_OUT -d suppothost.victronenergy.com -g REMOTE_SUPPORT_OUT

iptables -N VRM_IN
iptables -I VRM_IN -j ACCEPT
iptables -N VRM_OUT
iptables -I VRM_OUT -j ACCEPT
iptables -I OTHER_IN -s ccgxlogging.victronenergy.com -g VRM_IN
iptables -I OTHER_OUT -d ccgxlogging.victronenergy.com -g VRM_OUT

iptables -N UPDATE_IN
iptables -I UPDATE_IN -j ACCEPT
iptables -N UPDATE_OUT
iptables -I UPDATE_OUT -j ACCEPT
iptables -I OTHER_IN -s updates.victronenergy.com -g UPDATE_IN
iptables -I OTHER_OUT -d updates.victronenergy.com -g UPDATE_OUT

iptables -N PUBNUB_IN
iptables -I PUBNUB_IN -j ACCEPT
iptables -N PUBNUB_OUT
iptables -I PUBNUB_OUT -j ACCEPT
iptables -I OTHER_IN -s 54.246.196.128/26 -g PUBNUB_IN
iptables -I OTHER_OUT -d 54.246.196.128/26 -g PUBNUB_OUT
iptables -I OTHER_IN -s 54.93.127.192/26 -g PUBNUB_IN
iptables -I OTHER_OUT -d 54.93.127.192/26 -g PUBNUB_OUT

iptables -N NTP_IN
iptables -I NTP_IN -j ACCEPT
iptables -N NTP_OUT
iptables -I NTP_OUT -j ACCEPT
iptables -I OTHER_IN -p udp --sport 123 -g NTP_IN
iptables -I OTHER_OUT -p udp --dport 123 -g NTP_OUT

iptables -N DNS_IN
iptables -I DNS_IN -j ACCEPT
iptables -N DNS_OUT
iptables -I DNS_OUT -j ACCEPT
iptables -I OTHER_IN -p tcp --sport domain -g DNS_IN
iptables -I OTHER_IN -p udp --sport domain -g DNS_IN
iptables -I OTHER_OUT -p tcp --dport domain -g DNS_OUT
iptables -I OTHER_OUT -p udp --dport domain -g DNS_OUT
```

```
#Connman online check
iptables -N CONNMAN_IN
iptables -I CONNMAN_IN -j ACCEPT
iptables -N CONNMAN_OUT
iptables -I CONNMAN_OUT -j ACCEPT
iptables -I OTHER_IN -s ipv4.connman.net -g CONNMAN_IN
iptables -I OTHER_OUT -d ipv4.connman.net -g CONNMAN_OUT
```

Scripts

The configuration of iptables is normally lost between reboots. The following code should be placed in the file `/etc/init.d/iptables.sh`

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          iptables
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Set up iptables
### END INIT INFO

PATH=/sbin:/bin:/usr/sbin:/usr/bin
NAME=iptables.sh
DESC="iptables"

case "$1" in
  start)
    echo -n "Starting $DESC: "
    if [ -e /var/run/iptables ]; then
      echo "iptables is already started!"
      exit 1
    else
      touch /var/run/iptables
    fi

    # Load saved rules
    if [ -f /etc/iptables/rules ]; then
      iptables-restore -c < /etc/iptables/rules
    fi
    echo "$NAME."
    ;;
  stop)
    echo -n "Stopping $DESC: "
    if [ ! -e /var/run/iptables ]; then
      echo "iptables is already stopped!"
      exit 1
    else
```

```
        rm /var/run/iptables
    fi

    mkdir -p /etc/iptables
    # Backup old rules
    if [ -f /etc/iptables/rules ]; then
        cp /etc/iptables/rules /etc/iptables/rules.bak
    fi
    # Save new rules
    iptables-save -c > /etc/iptables/rules

    # Revert to Default Policy
    iptables -P INPUT ACCEPT
    iptables -P OUTPUT ACCEPT
    iptables -P FORWARD ACCEPT

    # Flush all rules and delete all custom chains
    iptables -F
    iptables -X
    echo "$NAME."
    ;;
restart|force-reload)
    echo -n "Restarting $DESC: "
    $0 stop
    $0 start
    echo "$NAME."
    ;;
backup)
    mkdir -p /etc/iptables
    # Backup old rules
    if [ -f /etc/iptables/rules ]; then
        cp /etc/iptables/rules /etc/iptables/rules.bak
    fi
    # Save new rules
    iptables-save -c > /etc/iptables/rules
    ;;
log-reset)
    mkdir -p /log/iptables/
    iptables -L -v -x -n -Z > /log/iptables/iptables-"$2"
    ;;
*)
    N=/etc/init.d/$NAME
    echo "Usage: $N {start|stop|restart|force-reload|backup|log-reset}"
>&2
    exit 1
    ;;
esac

exit 0
```

Then also execute the following command to make the script executable:

```
chmod +x /etc/init.d/iptables.sh
```

This allows shutting down the traffic counting using

```
/etc/init.d/iptables.sh stop
```

and start it using

```
/etc/init.d/iptables.sh start
```

.

When the traffic counting is stopped in this way, the current count values are stored together with the rules, allowing it to continue in the state it was stopped.

When the traffic counting should be started and stopped on startup and reboot of the CCGX, the following commands should also be executed:

```
ln -s /etc/init.d/iptables.sh /etc/rc5.d/S04iptables
ln -s /etc/init.d/iptables.sh /etc/rc6.d/K50iptables
```

By adding the following line to /etc/crontab, the current rules and counts will be backed up every 10 minutes, protecting them more or less from hard resets:

```
*/10 * * * * root /etc/init.d/iptables.sh backup
```

Viewing the data

After executing these commands, the traffic is counted by iptables. The data can be viewed by giving the command

```
iptables -L -x -v -n
```

The following is an example output:

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts      bytes target     prot opt in      out     source
destination
    78      25590 ACCEPT     all  --  *      *      0.0.0.0/0
255.255.255.255
   1033    141226 ACCEPT     all  --  *      *      0.0.0.0/0
224.0.0.0/4
      0         0 ACCEPT     all  --  *      *      192.168.0.0/16
0.0.0.0/0
   15848  1778034 ACCEPT     all  --  *      *      172.16.0.0/12
0.0.0.0/0
```

0	0	ACCEPT	all	--	*	*	10.0.0.0/8
0.0.0.0/0	8438	660949	ACCEPT	all	--	*	127.0.0.0/8
0.0.0.0/0	13432	2313739	OTHER_IN	all	--	*	0.0.0.0/0
0.0.0.0/0			[goto]				

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source
 destination

0	0	ACCEPT	all	--	*	*	0.0.0.0/0
255.255.255.255	173	33763	ACCEPT	all	--	*	0.0.0.0/0
224.0.0.0/4	0	0	ACCEPT	all	--	*	0.0.0.0/0
192.168.0.0/16	16646	4024183	ACCEPT	all	--	*	0.0.0.0/0
172.16.0.0/12	0	0	ACCEPT	all	--	*	0.0.0.0/0
10.0.0.0/8	8438	660949	ACCEPT	all	--	*	0.0.0.0/0
127.0.0.0/8	24853	7624844	OTHER_OUT	all	--	*	0.0.0.0/0
0.0.0.0/0			[goto]				

Chain CONNMAN_IN (1 references)							
	pkts	bytes	target	prot	opt	in	out source
destination							
0.0.0.0/0	11	960	ACCEPT	all	--	*	* 0.0.0.0/0

Chain CONNMAN_OUT (1 references)							
	pkts	bytes	target	prot	opt	in	out source
destination							
0.0.0.0/0	10	758	ACCEPT	all	--	*	* 0.0.0.0/0

Chain DNS_IN (2 references)							
	pkts	bytes	target	prot	opt	in	out source
destination							
0.0.0.0/0	0	0	ACCEPT	all	--	*	* 0.0.0.0/0

Chain DNS_OUT (2 references)							
	pkts	bytes	target	prot	opt	in	out source

```

destination
  0      0 ACCEPT    all  --  *    *    0.0.0.0/0
0.0.0.0/0

Chain NTP_IN (1 references)
  pkts    bytes target    prot opt in    out    source
destination
  3      228 ACCEPT    all  --  *    *    0.0.0.0/0
0.0.0.0/0

Chain NTP_OUT (1 references)
  pkts    bytes target    prot opt in    out    source
destination
  39     2964 ACCEPT    all  --  *    *    0.0.0.0/0
0.0.0.0/0

Chain OTHER_IN (1 references)
  pkts    bytes target    prot opt in    out    source
destination
  11     960 CONNMAN_IN all  --  *    *    87.106.208.187
0.0.0.0/0      [goto]
  0      0 DNS_IN    udp  --  *    *    0.0.0.0/0
0.0.0.0/0      [goto] udp spt:53
  0      0 DNS_IN    tcp  --  *    *    0.0.0.0/0
0.0.0.0/0      [goto] tcp spt:53
  3      228 NTP_IN    udp  --  *    *    0.0.0.0/0
0.0.0.0/0      [goto] udp spt:123
  0      0 PUBNUB_IN all  --  *    *    54.93.127.192/26
0.0.0.0/0      [goto]
  0      0 PUBNUB_IN all  --  *    *    54.246.196.128/26
0.0.0.0/0      [goto]
  0      0 UPDATE_IN all  --  *    *    185.24.223.128
0.0.0.0/0      [goto]
  11808  2161730 VRM_IN    all  --  *    *    46.19.32.79
0.0.0.0/0      [goto]
  1610   150821 REMOTE_SUPPORT_IN all  --  *    *    77.72.145.194
0.0.0.0/0      [goto]
  0      0 LOG      all  --  *    *    0.0.0.0/0
0.0.0.0/0      limit: avg 10/hour burst 5 LOG flags 0 level 7 prefix
"IPTables-OTHER_IN: "
  0      0 ACCEPT    all  --  *    *    0.0.0.0/0
0.0.0.0/0

Chain OTHER_OUT (1 references)
  pkts    bytes target    prot opt in    out    source
destination
  10     758 CONNMAN_OUT all  --  *    *    0.0.0.0/0
87.106.208.187 [goto]
  0      0 DNS_OUT    udp  --  *    *    0.0.0.0/0
0.0.0.0/0      [goto] udp dpt:53
  0      0 DNS_OUT    tcp  --  *    *    0.0.0.0/0

```

```

0.0.0.0/0          [goto] tcp dpt:53
    39      2964 NTP_OUT   udp  --  *      *      0.0.0.0/0
0.0.0.0/0          [goto] udp dpt:123
    0        0 PUBNUB_OUT all  --  *      *      0.0.0.0/0
54.93.127.192/26  [goto]
    0        0 PUBNUB_OUT all  --  *      *      0.0.0.0/0
54.246.196.128/26 [goto]
    0        0 UPDATE_OUT all  --  *      *      0.0.0.0/0
185.24.223.128   [goto]
    21431   7309572 VRM_OUT   all  --  *      *      0.0.0.0/0
46.19.32.79      [goto]
    3373    311550 REMOTE_SUPPORT_OUT all -- *      *      0.0.0.0/0
77.72.145.194    [goto]
    0        0 LOG      all  --  *      *      0.0.0.0/0
0.0.0.0/0          limit: avg 10/hour burst 5 LOG flags 0 level 7 prefix
"IPTables-OTHER_IN: "
    0        0 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain PUBNUB_IN (2 references)
  pkts      bytes target      prot opt in      out      source
destination
    0        0 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain PUBNUB_OUT (2 references)
  pkts      bytes target      prot opt in      out      source
destination
    0        0 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain REMOTE_SUPPORT_IN (1 references)
  pkts      bytes target      prot opt in      out      source
destination
    1610    150821 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain REMOTE_SUPPORT_OUT (1 references)
  pkts      bytes target      prot opt in      out      source
destination
    3373    311550 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain UPDATE_IN (1 references)
  pkts      bytes target      prot opt in      out      source
destination
    0        0 ACCEPT   all  --  *      *      0.0.0.0/0
0.0.0.0/0

```

```
Chain UPDATE_OUT (1 references)
  pkts      bytes target    prot opt in      out     source
destination
   0         0 ACCEPT    all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain VRM_IN (1 references)
  pkts      bytes target    prot opt in      out     source
destination
 11808 2161730 ACCEPT    all  --  *      *      0.0.0.0/0
0.0.0.0/0

Chain VRM_OUT (1 references)
  pkts      bytes target    prot opt in      out     source
destination
 21431 7309572 ACCEPT    all  --  *      *      0.0.0.0/0
0.0.0.0/0
```

The second column of the lines

```
 13432 2313739 OTHER_IN  all  --  *      *      0.0.0.0/0
0.0.0.0/0          [goto]
```

and

```
 24853 7624844 OTHER_OUT all  --  *      *      0.0.0.0/0
0.0.0.0/0          [goto]
```

show the incoming and outgoing traffic (in bytes) that is not for the local LAN. This is roughly the traffic that would normally be counted by the provider.

The lines under

```
Chain OTHER_IN
```

and

```
Chain OTHER_OUT
```

show the amount of traffic used by the different services running on the CCGX.

The following line also resets the counters (besides showing the current values):

```
iptables -L -x -v -n -Z
```

Warning: The current values are not stored anywhere and thus are lost!

By adding the following line to /etc/crontab, the overview is stored per day in /log/iptables/iptables-YYYY-MM-DD-HH-MM-SS and the counters reset, giving a traffic counting per day:

```
@daily                                root    /etc/init.d/iptables.sh log-
```

Last update: 2015-04-14 07:46 open_source:ccgx:ccgx_traffic_counting https://www.victronenergy.com/live/open_source:ccgx:ccgx_traffic_counting?rev=1428990401

reset \$(date +%F-%H-%M-%S)

From:
<https://www.victronenergy.com/live/> - **Victron Energy**

Permanent link:
https://www.victronenergy.com/live/open_source:ccgx:ccgx_traffic_counting?rev=1428990401

Last update: **2015-04-14 07:46**

