TRAFFIC MONITORING - ARGUS

One of the first rules of looking after any network is knowing what is happening on that network, becuase only then can see what should not be there. The first step on this road is to know what is leaving or entering your network and, of course, the best place to start looking is your firewall. now me being the cheapskate I am, I like to use free software. I have already spoken about using *rrdtool* for graphing (see here), but how do we get the details about our traffic to make the graphs? Enter *argus* (homepage here), an exceptionally useful tool, which is capable of very much more then protocol counts, but we will use it for that here.

What is needed?

A linux box (of course), *rrdtool* running somewhere and *argus* - both the server and the client. For ease of use, we will set both of those up on the same server. Lets first get and install the *argus* server..

```
wget ftp://qosient.com/dev/argus-3.0/argus-3.0.0.tar.gz
tar -xzvf argus-3.0.0.tar.gz
-cd into argus directory-
configure ; make ; make install
```

now we need to do some basic setup to get *argus* running..

```
cd support/
cd Config/
mkdir /etc/argus
cp ./argus.conf /etc/argus
chmod 600 /etc/argus/argus.conf
cd ..
cd Startup/
cp ./argus /etc/init.d
chmod 755 /etc/init.d/argus
chkconfig --add argus
```

lets get a little more detailed, we go into the *argus* configuration file, now make sure the following fields match your setup and needs..

```
cd /etc/argus
mkdir /var/log/argus
vi ./argus.conf

ARGUS_MONITOR_ID
ARGUS_INTERFACE
ARGUS_OUTPUT_FILE
ARGUS_GENERATE_RESPONSE_TIME_DATA
ARGUS_GENERATE_MAC_DATA
```

Almost done. Lets make sure that *argus* starts up ok..

```
cd /etc/init.d
vi ./argus
-make sure the setup reflects where you have the configuration file-
```

Finally, start the *argus* server..

```
/etc/init.d/argus start
```

Now if all has gone well, you will have your *argus* server busy collecting information about the traffic flowing in/out/through your system. But we still need to get that data into a meaningful format so we can graph it. For that we will need the *argus* client package. So lets get and install that..

```
wget ftp://qosient.com/dev/argus-3.0/argus-clients-3.0.0.tar.gz
tar -xzvf argus-clients-3.0.0.tar.gz
cd argus-clients-3.0.0
configure ; make ; make install
```

Now you have all the client tools (*racluster , ra* , etc). So lets take a look at using these tools. Lets first find out what type of packets are being seen..

```
/usr/local/bin/racluster -r /var/log/argus/argus.out -m proto -t `date +%T
--date '5 minutes ago'`-`date +%T` -s proto pkts bytes load rate loss
```

This will tell us the breakdown between tcp/udp/icmp/etc for the last 5 minutes. Nice but nowhere near a protocol based breakdown. To get that into a nice format we can use for our graphs, we will need a bit of scripting and as always I take no responsibilty if you use my script, I make no claim that it is pretty, optimized, or a work of art. But it works for me and make any changes you want. I put this in a scipt called *port_cnt.sh*..

```
##SETUP VARIABLES
MIN=5
RAC=/usr/local/bin/racluster
RAS=/usr/local/bin/rasort
LOG=/var/log/argus/argus.out
TMP1=/tmp/argus.port.in
TMP2=/tmp/argus.oth.in
DT1=`date +%H:%M`
DT2=`date +%H:%M -d '-5 min'`

##GET STATS INTO A TEMP FILE
$RAC -m proto dport -n -r $LOG -s saddr daddr sport dport bytes -t $DT2-$DT1
-w - | $RAS -n -m bytes -s dport bytes > $TMP1

##A FOR LOOP THROUGH ALL THE PORTS WE ARE INTERESTED IN WHICH DISPLAYS
<PORT> <BYTES>
for x in 21 22 25 53 80 110 143 443 444 445 1194 1433 1863 3306 5432
 do
  RSLT=`cat $TMP1 | grep -w $x 1> /dev/null ; echo $?`
  echo $x >> $TMP2
  if [ $RSLT == 1 ]
   then
    echo "$x      0" | gawk '{ print $1, $2 }'
   else
```

```
    cat $TMP1 | grep -w $x | gawk '{ print $1, $2 }'
  fi
 done

##NOW WE COUNT EVERYTHING ELSE
SUM=0
for y in `cat $TMP1 | grep -v -f $TMP2 | gawk '{ print $2 }'`
 do
  SUM=`expr $SUM \+ $y`
 done
echo "other $SUM"

#CLEANUP
rm -rf $TMP1
rm -rf $TMP2
```

That will give you a list of ports with the amount of bytes seen for that port in the last 5 minutes, which makes it easy to dump into *rrdtool*.

Last Words
This is a good example of how a couple of open source tools can really complement each other, gotta love it. Also please remember, *argus* is capable of a lot more then this and can used to gather all sorts of funky netflow data. So play around a bit more with it. As always have fun and learn.