ng_ipacct – подсчет трафика на интерфейсах

```
root@tst:/usr # portsnap fetch update
root@tst:/usr # portmaster net-mgmt/ng_ipacct
```

Для подсчета трафика с прокси Squid, нужно доставить calamaris:

```
root@tst:/usr # portmaster www/calamaris
```

Ядро дефолтное – все подгружается модулями.

Снимать статистику нужно на двух интерфейсах *rl0* – LAN, *re0* – WAN.

Правим /usr/local/etc/ng_ipacct.conf, строки начинающиеся на “ng_ipacct_xl0_” копируем в самый низ конфига (2 раза) и меняем *_xl0_* на *_rl0_* и *re0*.

```
svm@ring:/usr/local/etc# cat ng_ipacct.conf | grep "^[^#]"
ng_ipacct_enable="YES"
ng_ipacct_modules_load="YES"
ng_ipacct_modules_list="netgraph ng_ether ng_ipacct"
ng_ipacct_interfaces="rl0 re0"
ng_ipacct_default_ether_start='
        mkpeer %%iface%%: tee lower right
        name %%iface%%:lower %%iface%%_tee
        connect %%iface%%: lower upper left
        mkpeer %%iface%%_tee: ipacct right2left %%iface%%_in
        name %%iface%%_tee:right2left %%iface%%_ip_acct
         connect %%iface%%_tee: %%iface%%_ip_acct: left2right
%%iface%%_out
'
ng_ipacct_default_ether_stop='
        shutdown %%iface%%_ip_acct:
        shutdown %%iface%%_tee:
        shutdown %%iface%%:
```

```
'
ng_ipacct_bpf_ether_start='
        mkpeer %%iface%%: tee lower right
        name %%iface%%:lower %%iface%%_tee
        connect %%iface%%: lower upper left
        mkpeer %%iface%%_tee: bpf right2left %%iface%%_in
        name %%iface%%_tee:right2left %%iface%%_bpf
           connect %%iface%%_tee: right2left left2right
%%iface%%_out

mkpeer %%iface%%_bpf: ipacct %%iface%%_match_in %%iface%%_in
               name  %%iface%%_bpf:%%iface%%_match_in
%%iface%%_ip_acct
            connect %%iface%%_bpf: %%iface%%_ip_acct:
%%iface%%_match_out %%iface%%_out
'
ng_ipacct_bpf_ether_stop='
        shutdown %%iface%%_ip_acct:
        shutdown %%iface%%_bpf:
        shutdown %%iface%%_tee:
        shutdown %%iface%%:
'
ng_ipacct_xl0_dlt="EN10MB" # required line; see ipacctctl(8)
ng_ipacct_xl0_threshold="15000" # '5000' by default
ng_ipacct_xl0_verbose="yes" # 'yes' by default
ng_ipacct_xl0_saveuid="yes" # 'no' by default
ng_ipacct_xl0_savetime="no" # 'no' by default
ng_ipacct_xl0_start=${ng_ipacct_default_ether_start}
ng_ipacct_xl0_stop=${ng_ipacct_default_ether_stop}
ng_ipacct_xl0_checkpoint_script="path/to/your/script    --
checkpoint-and-save xl0"
                        # this script is called on "stop" (to
save accumulated
                          # data) or via "rc.d/ng_ipacct.sh
checkpoint"
ng_ipacct_cx0_dlt="RAW"
ng_ipacct_cx0_start='
        mkpeer %%iface%%: cisco rawdata downstream
        name %%iface%%:rawdata %%iface%%_hdlc
        mkpeer %%iface%%_hdlc: tee inet left
        name %%iface%%_hdlc:inet %%iface%%_tee
```

```
        mkpeer %%iface%%_tee: iface right inet
        mkpeer %%iface%%_tee: ipacct right2left %%iface%%_in
        name %%iface%%_tee:right2left %%iface%%_ip_acct
         connect %%iface%%_tee: %%iface%%_ip_acct: left2right
%%iface%%_out
'
ng_ipacct_cx0_stop='
        shutdown %%iface%%_ip_acct:
        shutdown %%iface%%_tee:
        shutdown %%iface%%_hdlc:
'
ng_ipacct_vpn0_dlt="RAW"
ng_ipacct_vpn0_start='
        mkpeer ipacct dummy dummy
        name .:dummy %%iface%%_ip_acct
          mkpeer %%iface%%_ip_acct: ksocket %%iface%%_in
inet/raw/divert
        name %%iface%%_ip_acct:%%iface%%_in ks_%%iface%%_in
        msg ks_%%iface%%_in: bind inet/0.0.0.0:4001
          mkpeer %%iface%%_ip_acct: ksocket %%iface%%_out
inet/raw/divert
        name %%iface%%_ip_acct:%%iface%%_out ks_%%iface%%_out
        msg ks_%%iface%%_out: bind inet/0.0.0.0:4002
        rmhook .:dummy
'
ng_ipacct_vpn0_stop='
        shutdown %%iface%%_ip_acct:
'
ng_ipacct_xl0_dlt="EN10MB" # required line; see ipacctctl(8)
ng_ipacct_xl0_threshold="15000" # '5000' by default
ng_ipacct_xl0_verbose="yes" # 'yes' by default
ng_ipacct_xl0_saveuid="yes" # 'no' by default
ng_ipacct_xl0_savetime="no" # 'no' by default
ng_ipacct_xl0_start=${ng_ipacct_bpf_ether_start}
ng_ipacct_xl0_stop=${ng_ipacct_bpf_ether_stop}
ng_ipacct_xl0_checkpoint_script="path/to/your/script    --
checkpoint-and-save xl0"
                        # this script is called on "stop" (to
save accumulated
                         # data) or via "rc.d/ng_ipacct.sh
checkpoint"
```

```
ng_ipacct_xl0_afterstart_script="path/to/your/script  --load-
bpf-filters xl0"
                        # this script is called just after
initialization
                         # of nodes to load filters into
xl0_bpf
ng_ipacct_ks_start='
        mkpeer ipacct dummy dummy
        name .:dummy %%iface%%_ip_acct
        mkpeer %%iface%%_ip_acct: tee %%iface%%_in left2right
        name %%iface%%_ip_acct:%%iface%%_in %%iface%%_tee_in
        mkpeer %%iface%%_ip_acct: tee %%iface%%_out left2right
        name %%iface%%_ip_acct:%%iface%%_out %%iface%%_tee_out
        mkpeer %%iface%%_tee_in: echo right in
        name %%iface%%_tee_in:right %%iface%%_echo_in
        mkpeer %%iface%%_tee_out: echo right out
        name %%iface%%_tee_out:right %%iface%%_echo_out

        mkpeer %%iface%%_tee_in: ksocket left inet/raw/divert
        name %%iface%%_tee_in:left %%iface%%_ks_in
        msg %%iface%%_ks_in: bind inet/0.0.0.0:4001
        mkpeer %%iface%%_tee_out: ksocket left inet/raw/divert
        name %%iface%%_tee_out:left %%iface%%_ks_out
        msg %%iface%%_ks_out: bind inet/0.0.0.0:4002
        rmhook .:dummy
'
ng_ipacct_ks_stop='
        shutdown %%iface%%_ks_in:
        shutdown %%iface%%_ks_out:
        shutdown %%iface%%_tee_in:
        shutdown %%iface%%_tee_out:
'
ng_ipacct_ks_dlt="RAW" # required line; see ipacctctl(8)
ng_ipacct_ks_threshold="15000" # '5000' by default
ng_ipacct_ks_verbose="yes" # 'yes' by default
ng_ipacct_ks_saveuid="yes" # 'no' by default
ng_ipacct_ks_savetime="no" # 'no' by default
ng_ipacct_ks_checkpoint_script="path/to/your/script    --
checkpoint-and-save ks"
                       # this script is called on "stop" (to
save accumulated
```

```
                        # data) or via "rc.d/ng_ipacct.sh
checkpoint"
ng_ipacct_rl0_dlt="EN10MB" # required line; see ipacctctl(8)
ng_ipacct_rl0_threshold="15000" # '5000' by default
ng_ipacct_rl0_verbose="yes" # 'yes' by default
ng_ipacct_rl0_saveuid="no" # 'no' by default
ng_ipacct_rl0_savetime="yes" # 'no' by default
ng_ipacct_rl0_start=${ng_ipacct_default_ether_start}
ng_ipacct_rl0_stop=${ng_ipacct_default_ether_stop}
ng_ipacct_rl0_checkpoint_script="/usr/sut/ipacct.sh rl0"
                        # this script is called on "stop" (to
save accumulated
                         # data) or via "rc.d/ng_ipacct.sh
checkpoint"
ng_ipacct_re0_dlt="EN10MB" # required line; see ipacctctl(8)
ng_ipacct_re0_threshold="15000" # '5000' by default
ng_ipacct_re0_verbose="yes" # 'yes' by default
ng_ipacct_re0_saveuid="no" # 'no' by default
ng_ipacct_re0_savetime="yes" # 'no' by default
ng_ipacct_re0_start=${ng_ipacct_default_ether_start}
ng_ipacct_re0_stop=${ng_ipacct_default_ether_stop}
ng_ipacct_re0_checkpoint_script="/usr/sut/ipacct.sh re0"
                        # this script is called on "stop" (to
save accumulated
                         # data) or via "rc.d/ng_ipacct.sh
checkpoint"
```

```
mkdir /usr/sut
```

Там же создаем скрипты и файлы для работы нашей системы

```
root@tst:/usr/sut  #  touch  daily_istat.pl  daily_pstat.pl
daily_sumnp.pl  daily_traf.sh  ipacct.sh  ipbas  ipbase  ipblan
ipbreal  ipmac.base
```

где *.pl и *.sh рабочие скрипты, а файлы вида ip* – список IP адресов сети.

Наполняем скрипты:

```
root@ring:/usr/sut # cat ipacct.sh
#!/bin/sh
# /usr/sut/ipacct.sh
```

```
IPACCTCTL="/usr/local/sbin/ipacctctl"
INTERFACES="rl0"
IFACE=$1
DIR=/usr/sut/ipacct

if [ ! -e "DIR" ]; then
 mkdir $DIR
fi

NAME="traf.log"
NAMEI=$NAME.$IFACE

for IFACE in $INTERFACES; do
$IPACCTCTL ${IFACE}_ip_acct:$IFACE checkpoint
$IPACCTCTL ${IFACE}_ip_acct:$IFACE show >> $DIR/$NAMEI
$IPACCTCTL ${IFACE}_ip_acct:$IFACE clear
done

dp=`/bin/date +%y%m%d-%H:%M`
messag=`/usr/bin/tail -1 $DIR/$NAMEI | /usr/bin/fgrep exceed`
if [ "$messag" ]; then
echo "$dp $messag" >> /usr/sut/ipacct/alarm.$IFACE
fi
```

```
root@ring:/usr/sut # cat daily_istat.pl
#!/usr/local/bin/perl
#
# /usr/sut/daily_istat.pl (daily_istat.pl interface)
# --interface "re0-inet, rl0-lan"

$iface1 = "re0";
$net1 = "194.44.";
$iface2 = "rl0";
$net2 = "192.168.";
#$net2 = /[0-9].[0-9]./;

$iface = shift(@ARGV);

if ($iface eq $iface1) {
        $ipi = "ipbreal";
        $net = $net1;
        }
```

```
if ($iface eq $iface2) {
        $ipi = "ipblan";
        $net = $net2;
        }
open (Fip, "/usr/sut/ipmac.base");
open (Fipi, ">/usr/sut/$ipi");
open (Fipii, ">/usr/sut/ipbas");
open (Fipiii, ">/usr/sut/ipbase");
$i = 0;
$j = 0;
while ($line = <Fip>) {
        ($p1,$xlam) = split(' ',$line,2);
        if ($p1 =~ /$net/) {
        print Fipi "$i $p1\n";
        print Fipii "$i $p1\n";
        $i = $i + 1;
        }
        $j = $j + 1;
        print Fipiii "$j       $p1\n";
}
$count = $i;
close (Fip);
close (Fipi);
close (Fipii);
close (Fipiii);


#goto Z1;
open (Fy, "/usr/sut/workyer");
chomp($y = <Fy>);
close (Fy);
open (Fm, "/usr/sut/workmon");
chomp($m = <Fm>);
close (Fm);
open (Fd, "/usr/sut/workday");
chomp($d = <Fd>);
close (Fd);
Z1:
goto Z2;
$y = "13";
$m = "08";
```

```
$d = "30";
Z2:

open (Fipii, "/usr/sut/ipbas");
@mip = <Fipii>;
close (Fipii);

$infile = "/usr/sut/$y.$m/$y$m$d.$iface";
open (InF, "$infile");

for ($i = 0; $i < $count; $i++) {
$ipin[$i] = 0;
$ipout[$i] = 0;
$ipsum[$i] = 0;
}

while ($line = <InF>) {
            ($p1,$p2,$p3,$p4,$p5,$p6,$p7,$p8) =  split(/
/,$line,8);
        foreach $str (@mip) {
        chomp($str);
        ($i,$ip) = split(/     /,$str,2);
                 if ($p1 eq $ip) {
                 $ipin[$i] = $ipin[$i] + $p7;
                 }
                 if ($p3 eq $ip) {
                 $ipout[$i] = $ipout[$i] + $p7;
                 }
         }
}

close (InF);
#---------------------------------
#goto ZZ;


open (TrF, ">>$infile.o");
$sumfile = "/usr/sut/$y.$m/tsum$y$m.$iface";
open (TrSum, ">>$sumfile");

$Mb = 1048576;
```

```
#$Mb = 10;

for ($i = 0; $i < $count; $i++) {
$str = @mip[$i];
chomp($str);
($xlam,$ip) = split(/ /,$str,2);
$ipin[$i] = int ($ipin[$i] / $Mb);
$ipout[$i] = int ($ipout[$i] / $Mb);
$ipsum[$i] = $ipin[$i] + $ipout[$i];

if ($ipsum[$i] > 0) {
printf TrF ("%-15s%15d%15d%15d\n", $ip, $ipin[$i], $ipout[$i],
$ipsum[$i]);

if ($d eq "01") {
printf  TrSum  ("%-15s%15d%15d%15d\n",  $ip,  $ipin[$i],
$ipout[$i], $ipsum[$i]);
}
}
}

close (TrF);
close (TrSum);

if ($d eq "01") {
goto ZZ;
}

#goto ZZ;

open (TekF, "/usr/sut/$y.$m/$y$m$d.$iface.o");
@mt = <TekF>;
close (TekF);

open (SumF, "/usr/sut/$y.$m/tsum$y$m.$iface");
@mts = <SumF>;
close (SumF);


open (SumF, "+>/usr/sut/$y.$m/tsum$y$m.$iface");
```

```
for ($k = 0; $k < $count; $k++) {
$str = @mip[$k];
chomp($str);
($k,$ip) = split(/     /,$str,2);
        for ($i = 0; $i < $count; $i++) {
        $line1 = @mts[$i];
#       $line1 = @m[$i];
        ($ip1,$in1,$out1,$sum1) = split(/ +/,$line1,4);
        $line2 = @mt[$i];
        ($ip2,$in2,$out2,$sum2) = split(/ +/,$line2,4);

        if ($ip eq $ip1) {
        $ins[$k] = $ins[$k] + $in1;
        $outs[$k] = $outs[$k] + $out1;
        $sums[$k] = $sums[$k] + $sum1;
        }

        if ($ip eq $ip2) {
        $ins[$k] = $ins[$k] + $in2;
        $outs[$k] = $outs[$k] + $out2;
        $sums[$k] = $sums[$k] + $sum2;
        }
        }
if ($sums[$k] > 0) {
printf SumF ("%-15s%15d%15d%15d\n", $ip, $ins[$k], $outs[$k],
$sums[$k]);
}
}

close (SumF);

ZZ:

#END
```

```
root@ring:/usr/sut # cat daily_pstat.pl
#!/usr/local/bin/perl
#
# /usr/sut/daily_pstat.pl
#

#goto Z1;
```

```
open (Fy, "/usr/sut/workyer");
chomp($y = <Fy>);
close (Fy);
open (Fm, "/usr/sut/workmon");
chomp($m = <Fm>);
close (Fm);
open (Fd, "/usr/sut/workday");
chomp($d = <Fd>);
close (Fd);

Z1:
goto Z2;
$y = "13";
$m = "08";
$d = "30";
Z2:

$file = "$y$m$d";

@temp = `cat /usr/sut/$y.$m/$file.log | \
 /usr/local/bin/calamaris -n -r -1 -S 14 -O -U M`;
$i = 0;
foreach $line (@temp) {
 ${ip[$i]} = substr($line,0,16);
 ${byte[$i]} = substr($line,51,8);
$i++;
}

$n = $i-7;

open (FH,">/usr/sut/$y.$m/$file.log.o");

print FH $temp[1];
print FH $temp[2];
print FH $temp[3];
print FH $temp[4];
print FH $temp[5];
print FH $temp[6];
for ($k = 7; $k<$n; $k++) {
print FH ${ip[$k]},${byte[$k]},"\n";
}
```

```
close (FH);

open (IpF, "/usr/sut/ipblan");
@mip = <IpF>;
close (IpF);

$infile = "/usr/sut/$y.$m/$y$m$d.log.o";
open (InF, "$infile");
@mp = <InF>;
close (InF);

$outfile = "/usr/sut/$y.$m/tsum$y$m.log";
open (OutF, "$outfile");
@msum = <OutF>;
close (OutF);

$outfile = "/usr/sut/$y.$m/tsum$y$m.log";
open (OutF, "+>$outfile");

#print "$y $m $d $infile $outfile\n";
#goto ZZ;


foreach $str (@mip) {
chomp($str);
($i,$ip) = split(/ /,$str,2);
 $ipsums[$i] = 0;
 foreach $strs (@msum) {
 ($ips,$ipsum) = split(/ +/,$strs,2);
 if ($ip eq $ips) {
 $ipsums[$i] = $ipsum;
 }
 }
 foreach $line (@mp) {
 ($p1,$p2) = split(/ /,$line,2);
 $p2 =~ tr/M\n/ /;
 if ($ip eq $p1 & $p2 > 0) {
 $ipsums[$i] = $ipsums[$i] + $p2;
 if ($d eq "01") {
 printf OutF ("%-15s%15d\n", $ip, $p2);
 }
```

```
 }
 }
 if ($d ne "01" & $ipsums[$i] > 0) {
 printf OutF ("%-15s%15d\n", $ip, $ipsums[$i]);
 }
}
close (OutF);

ZZ:

#END
```

```
root@ring:/usr/sut # cat daily_sumnp.pl
#!/usr/local/bin/perl
# Proxy + NAT
# /usr/sut/sum_n_p.pl

$iface = "rl0";

#goto Z1;
open (Fy, "/usr/sut/workyer");
chomp($y = <Fy>);
close (Fy);
open (Fm, "/usr/sut/workmon");
chomp($m = <Fm>);
close (Fm);
open (Fd, "/usr/sut/workday");
chomp($d = <Fd>);
close (Fd);
Z1:
goto Z2;
$y = "13";
$m = "08";
$d = "30";
Z2:
open (IpF, "/usr/sut/ipblan");
@mip = <IpF>;
close (IpF);

$infile1 = "/usr/sut/$y.$m/tsum$y$m.log";
open (InF1, "$infile1");
@mproxy = <InF1>;
```

```
close (InF1);

$infile2 = "/usr/sut/$y.$m/tsum$y$m.$iface";
open (InF2, "$infile2");
@mnat = <InF2>;
close (InF2);

$outfile = "/usr/sut/$y.$m/sumnp$y$m";
open (OutF, "+>$outfile");

#print "$y $m $d\n $infile1\n $infile2\n $outfile\n";
#goto ZZ;
#---------------
$trafproxy = 0;
$trafnat = 0;
$traflan = 0;
$trafproxyr = 0;
$trafnatr = 0;
$traflanr = 0;

foreach $str (@mip) {
chomp($str);
($i,$ip) = split(/ /,$str,2);
 $tsum[$i] = 0;
 $tproxyp[$i] = 0;
 $tsump[$i] = 0;
 foreach $linep (@mproxy) {
 ($ipp,$tproxy) = split(/ +/,$linep,2);
 if ($ip eq $ipp) {
 $tsum[$i] = $tproxy;
 $tproxyp[$i] = $tproxy;
 }
 }
 foreach $linen (@mnat) {
 ($ipn,$inn,$outn,$sumn) = split(/ +/,$linen,4);
 if ($ip eq $ipn) {
 $tsum[$i] = $tsum[$i] + $sumn;
 $tsump[$i] = $sumn;
 }
 }
if ($ip =~ /192.168./) {
```

```
$trafproxy = $trafproxy + $tproxyp[$i];
$trafnat = $trafnat + $tsump[$i];
$traflan = $traflan + $tsum[$i];
}
if ($ip =~ /194.***./) {
$trafproxyr = $trafproxyr + $tproxyp[$i];
$trafnatr = $trafnatr + $tsump[$i];
$traflanr = $traflanr + $tsum[$i];
}
$trafproxys = $trafproxy + $trafproxyr;
$trafnats = $trafnat + $trafnatr;
$traflans = $traflan + $trafnatr;

if ($tsum[$i] > 0) {
printf OutF ("%-15s%15d%15d%15d\n", $ip, $tproxyp[$i],
$tsump[$i], $tsum[$i]);
}
}
printf OutF ("%-15s%15d%15d%15d\n", "L Pr Nat Summ",
$trafproxy, $trafnat, $traflan);
printf OutF ("%-15s%15d%15d%15d\n", "R Pr Nat Summ",
$trafproxyr, $trafnatr, $traflanr);
printf OutF ("%-15s%15d%15d%15d\n", "LR Pr Nat Summ",
$trafproxys, $trafnats, $traflans);

close (OutF);

ZZ:

`cp $outfile /usr/sut/traffic`;

#END
```

```
root@ring:/usr/sut # cat daily_traf.sh
#!/bin/sh
# /usr/sut/daily_traf.sh
# Interfaces (out - WAN, in - LAN)
if_out='re0'
if_in='rl0'

y=`/bin/date +%y`
m=`/bin/date +%m`
```

```
d=`/bin/date +%d`

echo "$y" > /usr/sut/workyer
echo "$m" > /usr/sut/workmon
echo "$d" > /usr/sut/workday

DIR=/usr/sut/$y.$m
if [ ! -e "DIR" ]; then
 mkdir $DIR
fi

mv /usr/sut/ipacct/traf.log.$if_out $DIR/$y$m$d.$if_out
mv /usr/sut/ipacct/traf.log.$if_in $DIR/$y$m$d.$if_in
if [ -e /usr/sut/ipacct/alarm.$if_out ]; then
mv                        /usr/sut/ipacct/alarm.$if_out
/usr/sut/ipacct/a$y$m$d.$if_out
fi
if [ -e /usr/sut/ipacct/alarm.$if_in ]; then
mv /usr/sut/ipacct/alarm.$if_in /usr/sut/ipacct/a$y$m$d.$if_in
fi

cp /var/log/squid/access.log $DIR/$y$m$d.log
#cp /dev/null /var/log/squid/access.log
```

Теперь рабочие файлы:

```
root@ring:/usr/sut # cat ipbas
0     192.168.113.0
1     192.168.113.1
2     192.168.113.2
.....................
255   192.168.113.255

root@ring:/usr/sut # less ipbase
1     192.168.113.0
2     192.168.113.1
3     192.168.113.2
4     192.168.113.3
5     192.168.113.4
.........................

255   192.168.113.254
```

```
256   192.168.113.255
257   194.***.***.**1
258   194.***.***.**2
259   194.***.***.**3

root@ring:/usr/sut # cat ipblan
0     192.168.113.0
1     192.168.113.1
...................

253   192.168.113.253
254   192.168.113.254
255   192.168.113.255

root@ring:/usr/sut # cat ipbreal
0     194.***.***.***
1     194.***.***.***
2     194.***.***.***
3     194.***.***.***


root@ring:/usr/sut # cat ipmac.base
192.168.113.0     sn                        00   001   -----
    ----
192.168.113.1      mc  68:05:ca:02:06:62   00   002   g0309
 1496   roller
192.168.113.2     bl                        00
192.168.113.3     bl                        00
192.168.113.4     bl                        00
192.168.113.5     bl                        00
192.168.113.6     bl                        00
192.168.113.7      mc  00:07:e9:b8:ad:ee   00   024   g0310
----     pktn
.....................................
```

Добавляем в крон

```
root@ring:/usr/sut # cat /etc/crontab
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
MAILTO=""
#minute hour mday month wday who    command
#
```

```
*/4           *         *         *           *        root
/usr/local/etc/rc.d/ng_ipacct checkpoint > /dev/null
59      23   *    *     *    root   /usr/sut/daily_traf.sh
10       0    *    *     *    root   /usr/sut/daily_istat.pl
re0
20       0    *    *     *    root   /usr/sut/daily_istat.pl
rl0
30      0    *    *     *    root   /usr/sut/daily_pstat.pl
1       4    *    *     *    root   /usr/sut/daily_sumnp.pl
```

Проверка:

```
root@ring:/usr/sut # ngctl ls
There are 10 total nodes:
 Name: ngctl4605   Type: socket    ID: 000001e2    Num hooks:
0
 Name: mpd658-lso  Type: socket    ID: 00000003    Num hooks:
0
 Name: mpd658-cso  Type: socket    ID: 00000004    Num hooks:
0
 Name: mpd658-eso  Type: socket    ID: 00000005    Num hooks:
0
 Name: re0         Type: ether     ID: 00000006    Num hooks:
2
 Name: rl0         Type: ether     ID: 00000007    Num hooks:
2
 Name: rl0_tee     Type: tee       ID: 00000009    Num hooks:
4
 Name: rl0_ip_acct Type: ipacct    ID: 0000000a    Num hooks:
2
 Name: re0_tee     Type: tee       ID: 00000012    Num hooks:
4
 Name: re0_ip_acct Type: ipacct    ID: 00000013    Num hooks:
2
```

```
root@ring:/usr/sut # kldstat
Id Refs Address            Size      Name
 1 46   0xffffffff80200000 1f6e480   kernel
 2 1    0xffffffff82170000 316728    zfs.ko
 3 2    0xffffffff82487000 cb78      opensolaris.ko
 4 4    0xffffffff82494000 44bd8     ipfw.ko
```

```
 5 1    0xffffffff824d9000 9bd0     ipfw_nat.ko
 6 2    0xffffffff824e3000 17288    libalias.ko
 7 1    0xffffffff824fb000 28dd0    dummynet.ko
 8 1    0xffffffff82524000 8d40     ipdivert.ko
 9 1    0xffffffff82621000 2986     uhid.ko
10 1    0xffffffff82624000 39cc     ng_socket.ko
11 5    0xffffffff82628000 c57d     netgraph.ko
12 1    0xffffffff82635000 42ab     ng_mppc.ko
13 1    0xffffffff8263a000 81f      rc4.ko
14 1    0xffffffff8263b000 43da     ng_ether.ko
15 1    0xffffffff82640000 17db     ng_ipacct.ko
16 1    0xffffffff82642000 17ce     ng_tee.ko
```

Пример лога (файл вида /usr/sut/18.04/180408.re0):

```
Адр. источника   Порт    Адр. получателя  Порт   № прот.
Пакетов Байт
192.168.113.11   5230    192.168.113.1    3551   6       2
92    -1
192.168.113.11   5230    192.168.113.1    3551   6       4
168   0
192.168.113.11   5225    192.168.113.1    3551   6       2
92    -1
192.168.113.11   5225    192.168.113.1    3551   6       4
168   0
192.168.113.11   5223    192.168.113.1    3551   6       2
92    -1
192.168.113.11   5223    192.168.113.1    3551   6       4
168   0
192.168.113.17   50646   193.47.166.29    123    17      1
76    -1
192.168.113.11   5233    192.168.113.1    3551   6       2
92    -1
192.168.113.11   5233    192.168.113.1    3551   6       4
168   0
192.168.113.11   4427    92.157.81.29     28858  6       3
471   -1
10.90.90.90      0       239.255.255.100  0      2       2
184   0
192.168.113.11   5213    34.252.56.124    443    6       8
2664  100
```

где — номер протокола (описываются в /etc/protocols)