

```

#!/usr/bin/perl
#
# K.Creason 10.02.2012, with thanks to Ben Rockwood's toasterview which got me started in
the right direction

# When used in teamquest automated User Table Agent this will run as nobody:root,
# so read/write/exec permissions need to be set accordingly

my $DEBUG=0;
if($ARGV[0] eq "" ){ printf("Usage: $0 hostname,hostname,hostname [options: -t=interval
(teamquest report)]\n"); exit; }

#### Configuration Variables that need to be set for your environment
my $SNMPGET = "/opt/csw/bin/snmpget";           # Location of snmpget command
my $MIB = "/opt/teamquest/scripts/data/netapp.mib";      # Location of the NetApp MIB
my $COMMUNITY = "yourcommunity";                # Filer Community Name
my $SNMPuser="youruser";                        # username on NetApp with the login-snm
ability/role
my $SNMPup="thepassword";                       # password for username on NetApp with
the login-snm ability/role
my $netstatdir="/opt/teamquest/scripts/data";        # data directory to write logs to
my $extralogging=1;                               # extra logging of netstats, maybe more

my $SNMPOPTIONS="-v3 -n \"\" -u $SNMPuser -l authNoPriv -A $SNMPup -a Md5";
my $tqrep=0; my $interval=0;

#### Var setup
my @HOSTS= split (/,/, @ARGV[0]);

use Sys::Hostname;
    $host = hostname;
chomp(my $DATE = `date "+%m/%d/%Y %H:%M:00"`);

# Which header to print?
if (" $ARGV[1]" =~ /\^-t/)
{
    $tqrep=1;
    ($nothing,$interval)= split /=/, $ARGV[1];
    if($DEBUG>0) { print STDERR "Teamquest report on at interval $interval.\n";}
    # Print TQ head
    print "\"$DATE\" $interval $host\n";
    #print "\$time_interval $interval \$system\n";
}
else
{
    print "Hostname OnTapVersion Model Firmware CPUbusyPercent batteryStat Disks Spares
FailedDisks FailedFans FailedPS NetKbRecvd NetKbSent\n"; }

foreach $HOST (@HOSTS)
{
if ($extralogging>0){ open ELOG, ">>$netstatdir/$HOST.log" or die $!;}

my $SNMPCMD = "$SNMPGET $SNMPOPTIONS -O qv -c $COMMUNITY -m $MIB $HOST ";
if ($DEBUG>0){print STDERR "My SNMPCD: \"$SNMPCMD\"\n"; }

chomp($OnTapVersion = `$SNMPCMD NETWORK-APPLIANCE-MIB::productVersion.0`);
    my @OTVsplit = split ( /[ :]+/, $OnTapVersion);
    $OnTapVersion = $OTVsplit[2];
chomp($Model = `$SNMPCMD NETWORK-APPLIANCE-MIB::productModel.0`);
    $Model =~ s//g;
chomp($Firmware = `$SNMPCMD NETWORK-APPLIANCE-MIB::productFirmwareVersion.0`);
    $Firmware =~ s//g;

```

```

chomp($CPUbusyPercent = `$$SNMPCMD NETWORK-APPLIANCE-MIB::cpuBusyTimePerCent.0`);
chomp($FailedFans = `$$SNMPCMD NETWORK-APPLIANCE-MIB::envFailedFanCount.0`);
chomp($FailedPS = `$$SNMPCMD NETWORK-APPLIANCE-MIB::envFailedPowerSupplyCount.0`);
chomp($batteryStat = `$$SNMPCMD NETWORK-APPLIANCE-MIB::nvramBatteryStatus.0`);
chomp($Disks = `$$SNMPCMD NETWORK-APPLIANCE-MIB::diskTotalCount.0`);
chomp($FailedDisks = `$$SNMPCMD NETWORK-APPLIANCE-MIB::diskFailedCount.0`);
chomp($Spares = `$$SNMPCMD NETWORK-APPLIANCE-MIB::diskSpareCount.0`);

chomp($NetRcvd = `$$SNMPCMD NETWORK-APPLIANCE-MIB::misc64NetRcvdBytes.0`);
chomp($NetSent = `$$SNMPCMD NETWORK-APPLIANCE-MIB::misc64NetSentBytes.0`);
if ($DEBUG>0){ print STDERR "\tData raw bytes\t\treceived=$NetRcvd,\tsent=$NetSent.
\n";}
# Dumb down the bytes to kilobytes
$NetRcvd=sprintf("%0.f", $NetRcvd/1024);
$NetSent=sprintf("%0.f", $NetSent/1024);
if ($DEBUG>0){ print STDERR "\tData raw kilobytes\t\treceived=$NetRcvd,\tsent=$NetSent.
\n";}
if ($extralogging>0){print ELOG "$DATE: CPU=$CPUbusyPercent, ";}

if ( -f "$netstatdir/$HOST.rec" )
{
open NETSTAT, "<$netstatdir/$HOST.rec" or die $!;
my $oldNetRec=<NETSTAT>; close (NETSTAT);
open NETSTAT, ">$netstatdir/$HOST.rec" or die $!;
print NETSTAT "$NetRcvd"; close (NETSTAT); close (NETSTAT);
if ($extralogging>0){print ELOG "NRec=$NetRcvd, Prev=$oldNetRec, ";}
if ($DEBUG>0){ print STDERR "\t\tCalculating starting with New KB received
$NetRcvd and subtracting $oldNetRec";}
$NetRcvd= $NetRcvd-$oldNetRec;
if ($DEBUG>0){ print STDERR "and new value is $NetRcvd.\n";}
if ($extralogging>0){print ELOG "NRec=$NetRcvd, ";}
}
else
{
if ($DEBUG>0){print STDERR "Creating the netstat rcvd file for this the host $HOST.
\n";}
open NETSTAT, ">$netstatdir/$HOST.rec" or die $!;
print NETSTAT $NetRcvd; close (NETSTAT);
$NetRcvd=0;
}

if ( -f "$netstatdir/$HOST.sent" )
{
open NETSTAT, "<$netstatdir/$HOST.sent" or die $!;
my $oldNetSent=<NETSTAT>; close (NETSTAT);
open NETSTAT, ">$netstatdir/$HOST.sent" or die $!;
print NETSTAT "$NetSent"; close (NETSTAT);
if ($extralogging>0){print ELOG "NSent=$NetSent, Prev=$oldNetSent, ";}
if ($DEBUG>0){ print STDERR "\t\tCalculating starting with New KB sent $NetSent
and subtracting $oldNetSent";}
$NetSent= $NetSent-$oldNetSent;
if ($DEBUG>0){ print STDERR "and new value is $NetSent.\n";}
if ($extralogging>0){print ELOG "NSent=$NetSent\n";}
}
else

```

```
{
    if ($DEBUG>0){print STDERR "Creating the netstat sent file for this the host
$HOST.\n";}
    open NETSTAT, ">$netstatdir/$HOST.sent" or die $!;
    print NETSTAT $NetSent; close (NETSTAT);
    $NetSent=0;
}

    if ($DEBUG>0){ print STDERR "\tcalculated kilobytes\treceived=$NetRecvd,\tsent=
$NetSent.\n";}

print "$HOST $OnTapVersion $Model $Firmware $CPUbusyPercent $batteryStat $Disks $Spares
$FailedDisks $FailedFans $FailedPS $NetRecvd $NetSent\n";

if ($extralogging>0){ close (ELOG);}

}
print "\n";
```