

# verot.net - aMule and amulecmd remote monitoring

[amule\\_monitor.htm](#)

## What?

This is a simple little script to monitor a remote [aMule](#) setup. I need to do this since my aMule is running on another box, that I administrate via SSH or xVNC. But I want to be able to have some feedback on how the mule is doing...

## How?

On the machine where aMule is running, (and I guess that would work too with [aMuled](#)), first install [amulecmd](#)

```
sudo apt-get install amule-utils
```

Then in aMule, go into *Preferences*, then *Remote Controls*, and tick *Accept external connections*. Fill in a password (for instance *mypass*), eventually change the port number (for instance *5555*) and maybe the IP of the listening interface.

Still on the machine where aMule is running, you can test if the external connections are working:

```
amulecmd -h localhost -p 5555 -P mypass -c status
```

It should display some information about the aMule client.

When all of that is working, you can try to run the same command on the local machine (if the remote machine is outside your network, you might want to forward the port number, or even tunnel the whole thing through SSH)

```
amulecmd -h 1.1.0.25 -p 5555 -P mypass -c status
```

Here, *1.1.0.25* is the IP of the remote machine on which aMule runs. The command should display some information about the aMule client.

## The script

Now, on the local machine, you can run the following script. Don't forget to change the host, port number and password in the first lines. You can also download the script [here](#).

```
#!/bin/sh
#
# amule_remote_status.sh v 0.1
#
# Bash script to look display the status of a remote amule + amulecmd setup
#
# by Colin Verot    colin@verot.net
```

```

#
# this script is free to use, just keep the copyright notice
#
host=1.1.1.25
port=5555
pass=mypass
# we first get the general data
r=`amulecmd -h $host -p $port -P $pass -c status`
# we parse it
download=`echo $r | grep -o 'Download: .* Up' | sed -e 's/Download: //' -e 's/ >
Up//'`
upload=`echo $r | grep -o 'Upload: .* Cl' | sed -e 's/Upload: //' -e 's/ >
Cl//'`
ed2k=`echo $r | grep -o 'ED2K: .* Kad' | sed -e 's/ED2K: //' -e 's/ > Kad//'`
kad=`echo $r | grep -o 'Kad: .* Dow' | sed -e 's/Kad: //' -e 's/ > Dow//'`
# and then display it nicely
echo "amulecmd : listens on $host:$port"
echo "Download : $download"
echo "Upload    : $upload"
echo "ED2K      : $ed2k"
echo "Kad       : $kad"
# we now get the current list of files, in a temp file
amulecmd -h $host -p $port -P $pass -c 'show DL' > /tmp/amulecmd
touch /tmp/amulecmd2
# we parse this file line by line, reformatting it
cat /tmp/amulecmd | grep '>' | (
    while read line;
        do l=`echo $line | sed -e 's/> [A-Z0-9]\{32\} \\.*\) \[([0-9\.]*\)%\]
\[([0-9]\{1,\}\/[0-9]\{1,\}\) - \\.*\)\2%\3#\1#\4/'`;
        echo $l >> /tmp/amulecmd2 ;
    done
)
# we output the list of files, formatted in columns
echo
cat /tmp/amulecmd2 | sort -r -g | column -s '#' -t
# we delete the temp files
rm /tmp/amulecmd
rm /tmp/amulecmd2
exit 0;

```

Make the script executable, and you should be able to run it with a simple command such as:

```
watch ./amule_remote_status.sh
```

## Output

The script will output something like this:

```

amulecmd : listens on server:5555
Download : 5.04 kB/s
Upload    : 2.00 kB/s
ED2K      : Connected to Some Server [xx.xx.xx.xx:4242] with HighID

```

```
Kad      : Connected (ok)
69.9%   0/13   Shadows.avi           Waiting
55.1%   1/37   Aguirre, Der Zorn Gottes.avi  Downloading 2.27 kB/s
31.5%   1/18   Network.avi          Downloading 3.68 kB/s
25.4%   0/54   Dersu Uzala - CD1.avi    Waiting
20.4%   0/148  Jules Et Jim [VF].avi    Waiting
19.3%   0/45   Le Genou De Claire.avi   Waiting
0.0%    0/43   Song From The Second Floor.avi Paused
```

Et voila!

## **Disclaimer**

Running aMule and amulecmd with external commands leaves a port open in your computer. I shall not be held responsible for any problem related to the use of this script.