AUTOSSH(1) BSD General Commands Manual AUTOSSH(1)

NAME

autossh — monitor and restart ssh sessions

SYNOPSIS

autossh [-V] [-M port[:echo\_port]] [-f] [SSH\_OPTIONS]

DESCRIPTION

autossh is a program to start a copy of ssh and monitor it, restarting it as necessary should it die or stop passing traffic.

The original idea and the mechanism were from rstunnel (Reliable SSH Tunnel). With version 1.2 of autossh the method changed: autossh uses ssh to construct a loop of ssh forwardings (one from local to remote, one from remote to local), and then sends test data that it expects to get back. (The idea is thanks to Terrence Martin.)

With version 1.3, a new method is added (thanks to Ron Yorston): a port may be specified for a remote echo service that will echo back the test data. This avoids the congestion and the aggravation of making sure all the port numbers on the remote machine do not collide. The loop-of-forwardings method remains available for situations where using an echo service may not be possible.

CONTROLLING SSH

SSH exits

autossh tries to distinguish the manner of death of the ssh process it is monitoring and act appropriately. The rules are:

1. If the ssh process exited normally (for example, someone typed "exit" in an interactive session), autossh exits rather than restarting;

2. If autossh itself receives a SIGTERM, SIGINT, or a SIGKILL signal, it assumes that it was deliberately signalled, and exits after killing the child ssh process;

3. If autossh itself receives a SIGUSR1 signal, it kills the child ssh process and starts a new one;

4. Periodically (by default every 10 minutes), autossh attempts to pass traffic on the monitor forwarded port. If this fails, autossh will kill the child ssh process (if it is still running) and start a new one;

5. If the child ssh process dies for any other reason, autossh will attempt to start a new one.

Startup behaviour

If the ssh session fails with an exit status of 1 on the very first try, autossh

1. will assume that there is some problem with syntax or the connection setup, and will exit rather than retrying;

2. There is a "starting gate" time. If the first ssh process fails within the first few seconds of being started, autossh assumes that it never made it "out of the starting gate",

and exits. This is to handle initial failed authentication, connection, etc. This time is 30 seconds by default, and can be adjusted (see the AUTOSSH\_GATETIME environment variable

below). If AUTOSSH\_GATETIME is set to 0, then both behaviours are disabled: there is no "starting gate", and autossh will restart even if ssh fails on the first run with an exit

status of 1. The "starting gate" time is also set to 0 when the -f flag to autossh is used.

Continued failures

If the ssh connection fails and attempts to restart it fail in quick succession, autossh will start delaying its attempts to restart, gradually backing farther and farther off up to a

maximum interval of the autossh poll time (usually 10 minutes). autossh can be "prodded" to retry by signalling it, perhaps with SIGHUP ("kill -HUP").

Connection setup

As connections must be established unattended, the use of autossh requires that some form of automatic authentication be set up. The use of RSAAuthentication with ssh-agent is the recommended method. The example wrapper script attempts to check if there is an agent running for the current environment, and to start one if there isn't.

It cannot be stressed enough that you must make sure ssh works on its own, that you can set up the session you want before you try to run it under autossh

If you are tunnelling and using an older version of ssh that does not support the -N flag, you should upgrade (your version has security flaws). If you can't upgrade, you may wish to do

as rstunnel does, and give ssh a command to run, such as "sleep 99999999999".

OPTIONS

-M port[:echo\_port]

specifies the base monitoring port to use. Without the echo port, this port and the port immediately above it ( port + 1) should be something nothing else is using. autossh will

send test data on the base monitoring port, and receive it back on the port above. For example, if you specify “-M 20000”, autossh will set up forwards so that it can send data on port 20000 and receive it back on 20001.

Alternatively, a port for a remote echo service may be specified. This should be port 7 if you wish to use the standard inetd echo service. When an echo port is specified, only

the specified monitor port is used, and it carries the monitor message in both directions.

Many people disable the echo service, or even disable inetd, so check that this service is available on the remote machine. Some operating systems allow one to specify that the

service only listen on the localhost (loopback interface), which would suffice for this use.

The echo service may also be something more complicated: perhaps a daemon that monitors a group of ssh tunnels.

Setting the monitor port to 0 turns the monitoring function off, and autossh will only restart ssh upon ssh's exit. For example, if you are using a recent version of OpenSSH, you

may wish to explore using the ServerAliveInterval and ServerAliveCountMax options to have the SSH client exit if it finds itself no longer connected to the server. In many ways

this may be a better solution than the monitoring port.

-f causes autossh to drop to the background before running ssh. The -f flag is stripped from arguments passed to ssh. Note that there is a crucial a difference between -f with

autossh, and -f with ssh: when used with autossh ssh will be unable to ask for passwords or passphrases. When -f is used, the "starting gate" time (see AUTOSSH\_GATETIME) is set to 0.

-V causes autossh to display its version number and exit.

ENVIRONMENT

Other than the flag to set the connection monitoring port, autossh uses environment variables to control features. ssh seems to be still collecting letters for options, and this seems the

easiest way to avoid collisions.

AUTOSSH\_DEBUG

If this variable is set, the logging level is set to to LOG\_DEBUG, and if the operating system supports it, syslog is set to duplicate log entries to stderr.

AUTOSSH\_FIRST\_POLL

Specifies the time to wait before the first connection test. Thereafter the general poll time is used (see AUTOSSH\_POLL below).

AUTOSSH\_GATETIME

Specifies how long ssh must be up before we consider it a successful connection. The default is 30 seconds. Note that if AUTOSSH\_GATETIME is set to 0, then not only is the gate‐

time behaviour turned off, but autossh also ignores the first run failure of ssh. This may be useful when running autossh at boot.

AUTOSSH\_LOGLEVEL

Specifies the log level, corresponding to the levels used by syslog; so 0-7 with 7 being the chattiest.

AUTOSSH\_LOGFILE

Specifies that autossh should use the named log file, rather than syslog.

AUTOSSH\_MAXLIFETIME

Sets the maximum number of seconds that the program should run. Once the number of seconds has been passed, the ssh child will be killed and the program will exit.

AUTOSSH\_MAXSTART

Specifies how many times ssh should be started. A negative number means no limit on the number of times ssh is started. The default value is -1.

AUTOSSH\_MESSAGE

Append message to echo message sent when testing connections.

AUTOSSH\_NTSERVICE

(Cygwin only.) When set to "yes" , autossh sets up to run as an NT service under cygrunsrv. This adds the -N flag for ssh if not already set, sets the log output to stdout, and

changes the behaviour on ssh exit so that it will restart even on a normal exit.

AUTOSSH\_PATH

Specifies the path to the ssh executable, in case it is different than the path compiled in.

AUTOSSH\_PIDFILE

Write autossh pid to specified file.

AUTOSSH\_POLL

Specifies the connection poll time in seconds; default is 600 seconds. Unless AUTOSSH\_FIRST\_POLL is used, the first poll time will set to match the poll time. If the poll time is

less than twice the network timeouts (default 15 seconds) the network timeouts will be adjusted downward to 1/2 the poll time.

AUTOSSH\_PORT

Sets the connection monitoring port. Mostly in case ssh appropriates -M at some time. But because of this possible use, AUTOSSH\_PORT overrides the -M flag. A value of 0 turns the

monitoring function off.

DEBIAN-SPECIFIC BEHAVIOR

The debian version of autossh uses a wrapper to automatically select a free monitoring port and -M overrides AUTOSSH\_PORT, see /usr/share/doc/autossh/README.Debian for further informa‐

tion.

AUTHOR

autossh was written by Carson Harding.

SEE ALSO

ssh(1), ssh-add(1), ssh-agent(1), ssh-keygen(1), cygrunsrv(1).

BSD Jul 20, 2004 BSD