Sending Messages to Samba Clients

Once a Samba/CIFS network is created, it can be quite helpful to be able to send 'Winpopup' messages to these clients. Some firewalls do block these messages though, so they are not a guaranteed method of communication.

To send a message to a host either

```
smbclient -M HOSTNAME
Type Message Here then Press Ctrl-D to send
```

or

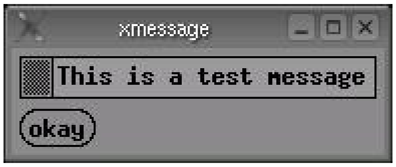
```
cat textfile | smbclient -M HOSTNAME
```

will suffice.

In order to receive such messages on a Linux/Unix nod, you must have the messages option set in smb.conf. By default this option is not set at all, so insert a line similar to the one below;

```
message command = bash -c 'DISPLAY=:0.0 xmessage -file %s & rm %s'
```

This will pop-up a messagebox containing the message received;



```
This is a test message
```

Broadcasting to Multiple Hosts with smbclient -m

Currently there is no easy way to broadcast to more than one host, however a _very_ basic set of scripts is attached overleaf. All reside within the same directory and are called using Smbbroadcast from a console.

The Final script can be edited and put into your PATH to allow the script to be called from any directory, and will also give you the option to specify hostnames/I.P. Addresses to broadcast to. It will also allow you to specify the message to send

Project Files are available from http://benscomputer.no-ip.org/2009/Jan/smbbroadcast/

Smbbroadcast

```
#!/bin/bash
# Smbclient Broadcast Script
# Allows the broadcasting of a message to all detected clients
#
# Released by Ben Tasker under the GPL


echo "Hope you've entered a message in MyMessage, otherwise there's gonna be a few blanks boxes!"

if [ "$HOST_SPECIFIED" == "1" ]
then
echo "Host Autodetect Disabled by HOST_SPECIFIED"
else
./findsmb_mod > smbusers
fi
for i in `cat /tmp/smbusers`
do
cat MyMessage | smbclient -M $i &
done
```

findsmb_mod

```
#!/usr/bin/perl
#
# Prints info on all smb responding machines on a subnet.
# This script needs to be run on a machine without nmbd running and be
# run as root to get correct info from WIN95 clients.
#
# syntax:
#    findsmb [-d|-D] [-r] [subnet broadcast address]
#
# with no agrument it will list machines on the current subnet
#
# There will be a "+" in front of the workgroup name for machines that are
# local master browsers for that workgroup. There will be an "*" in front
# of the workgroup name for machines that are the domain master browser for
# that workgroup.
#
# Options:
#
# -d|-D         enable debug
# -r            add -r option to nmblookup when finding netbios name
#

# Reporting Modified by Ben Tasker for use in smbclient broadcast function.
# Adjustments could be far cleaner than they are, but it's a proof of concept
# So I'll clean it later.

$SAMBABIN = "/usr/bin";

for ($i = 0; $i < 2; $i++) {    # test for -d and -r options
        $_ = shift;
        if (m/-d|-D/) {
                $DEBUG = 1;
        } elsif (m/-r/) {
                $R_OPTION = "-r";
        }
}

if ($_) {                       # set broadcast address if it was specified
        $BCAST = "-B $_";
}


######################################################################
# do numeric sort on last field of IP address
sub ipsort
{
        @t1 = split(/\./,$a);
        @t2 = split(/\./,$b);
        @t1[3] <=> @t2[3];
}
######################################################################

# look for all machines that respond to a name lookup

open(NMBLOOKUP,"$SAMBABIN/nmblookup $BCAST '*' --debuglevel=0|") ||
        die("Can't run nmblookup '*'.\n");

# get rid of all lines that are not a response IP address,
# strip everything but IP address and sort by last field in address

@ipaddrs = sort ipsort grep(s/ \*<00>.*$//,<NMBLOOKUP>);

# print header info

# Don't need this !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


#print "\n                                *=DMB\n";
#print "                                +=LMB\n";
#print "IP ADDR         NETBIOS NAME     WORKGROUP/OS/VERSION $BCAST\n";
#print "---------------------------------------------------------------------\n";
# Commented out by Ben Tasker
```

```perl
foreach $ip (@ipaddrs)          # loop through each IP address found
{
        $ip =~ s/\n//;          # strip newline from IP address

        # find the netbios names registered by each machine

        open(NMBLOOKUP,"$SAMBABIN/nmblookup $R_OPTION -A $ip --debuglevel=0|") ||
                die("Can't get nmb name list.\n");
        @nmblookup = <NMBLOOKUP>;
        close NMBLOOKUP;

        # get the first <00> name

        @name = grep(/<00>/,@nmblookup);
        $_ = @name[0];

        if ($_) {                       # we have a netbios name
                if (/GROUP/) {          # is it a group name
                        ($name, $aliases, $type, $length, @addresses) =
                        gethostbyaddr(pack('C4',split('\.',$ip)),2);
                        if (! $name) {                  # could not get name
                                $name = "unknown nis name";
                        }
                } else {
                        # The Netbios name can contain lot of characters also '<' '>'
                        # and spaces. The follwing cure inside name space but not
                        # names starting or ending with spaces
                        /(.{1,15})\s+<00>\s+/;
                        $name = $1;
                        $name =~ s/^\s+//g;
                }

                # do an smbclient command on the netbios name.

                if ( "$name" ) {
                        open(SMB,"$SAMBABIN/smbclient -L $name -I $ip -N --debuglevel=1 2>&1 |") ||
                                die("Can't do smbclient command.\n");
                        @smb = <SMB>;
                        close SMB;

                        if ($DEBUG) {           # if -d flag print results of nmblookup and smbclient
                         print "===============================================================\n";
                                print @nmblookup;
                                print @smb;
                        }

                        # look for the OS= string

                        @info = grep(/OS=/,@smb);
                        $_ = @info[0];
                        if ($_) {                               # we found response
                     s/Domain=|OS=|Server=|\n//g;    # strip out descriptions to make line shorter

               } else {                                # no OS= string in response (WIN95 client)

                                # for WIN95 clients get workgroup name from nmblookup response
                                @name = grep(/<00> - <GROUP>/,@nmblookup);
                                $_ = @name[0];
                                if ($_) {
                                        # Same as before for space and characters
                                        /(.{1,15})\s+<00>\s+/;
                                        $_ = "[$1]";
                                } else {
                                        $_ = "Unknown Workgroup";
                                }
                        }
                }

                # see if machine registered a local master browser name
                if (grep(/<1d>/,@nmblookup)) {
                        $master = '+';                  # indicate local master browser
                        if (grep(/<1b>/,@nmblookup)) {  # how about domain master browser?
                                $master = '*';                  # indicate domain master browser
                        }
                } else {
                        $master = ' ';                  # not a browse master
                }

                # line up info in 3 columns


# If you have problems with Name Resolution, change this variable to $ip
                print "$name \n";
# Changed by Ben Tasker


        } else {                                # no netbios name found
                # try getting the host name
                ($name, $aliases, $type, $length, @addresses) =
                gethostbyaddr(pack('C4',split('\.',$ip)),2);
                if (! $name) {                  # could not get name
                        $name = "unknown nis name";
                }
                if ($DEBUG) {                   # if -d flag print results of nmblookup
                        print "===============================================================\n";
                        print @nmblookup;
                }
                print "$ip".' 'x(16-length($ip))."$name\n";
        }
}
```

smbbroadcast-bin

```
#!/bin/bash


#
# Wrapper Script for Smbbroadcast
#
# Copyright Ben Tasker
# Released under the GNU GPL
# Please see http://benscomputer.no-ip.org/LICENSE for more details
# Make this script executable and put it into your PATH

# Variables

# Set this variable to point to the directory containing Smbbroadcast
PROGDIR="/home/ben/programs/Smbbroadcast"
# Set this to use your chosen text editor
EDITOR="nano"

clear

# We are going to change Directory, so make a note of where the script was called from so we can
# cd back to it
CURRENTDIR=$( pwd )

cd $PROGDIR
echo "Welcome to Smbbroadcast"
echo ""
echo "Press Return and then enter your message, remember to save it"
read tmp1
$EDITOR MyMessage
clear
echo "Please select from the following"
echo ""
echo "1) Broadcast to all known hosts"
echo "2) Broadcast to a user specified list"
echo ""
read BROADCASTSET
if [ "$BROADCASTSET" == "1" ]
then
./Smbbroadcast
echo ""
echo "Message Sent"
cd $CURRENTDIR
exit
fi

if [ "$BROADCASTSET" == "2" ]
then
clear
echo "OK Press Enter to Enter Hostnames/I.P. Addresses, please put each entry on a seperate line and remember to save!"
$EDITOR smbusers
echo ""
HOST_SPECIFIED=1 ./Smbbroadcast
echo ""
echo "Message Sent"
cd $CURRENTDIR
exit
fi

echo "Invalid choice, Your Message has been saved, please call this script again"
exit
```