CallerVerify Docs
v.1.0
10/5/92

1. What is CallerVerify?

CallerVerify is a fully-featured callback system for Second Sight BBS sytems. This program allows your bbs to call new users back automatically, allowing new users instant access to your bbs. As you can imagine, this allows you, the sysop, to spend more time working on your bbs rather than calling users back on the phone.

2. How does it work?

CallerVerify is setup to work in two ways, as part of a logon script, and, as a menu command type 51 application. Let me explain how I have it setup on my Second Sight 2.2 (b35) bbs:

When a user logs on, he fills out the information requested by the bbs (CR/LF, size of screen, "Hot Menus", system messages, etc.) and then goes to the main menu. There, he may execute a survey (that doesn't give him access automatically) or use CallerVerify. He then (it is assumed) tries the CallerVerify option. CallerVerify then takes over. It explains to the user what a callback verifier is, how it works and then asks for his home phone number. He then enters his phone number. CallerVerify then hangs up the phone, waits a few seconds, and calls the user's phone number. Upon a successful connect, the user is asked his password. If the password checks out, he is given access to the BBS.

Note: if this fails, I've got a backup newuser survey to serve as a 'catch-all'. It doesn't validate the user, but it does allow me to call those manually that couldn't use CallerVerify. Typically, these are long distance users.

Pretty simple stuff. There are a few stops along the way, but, generally, that is how it works on Fort Mac.

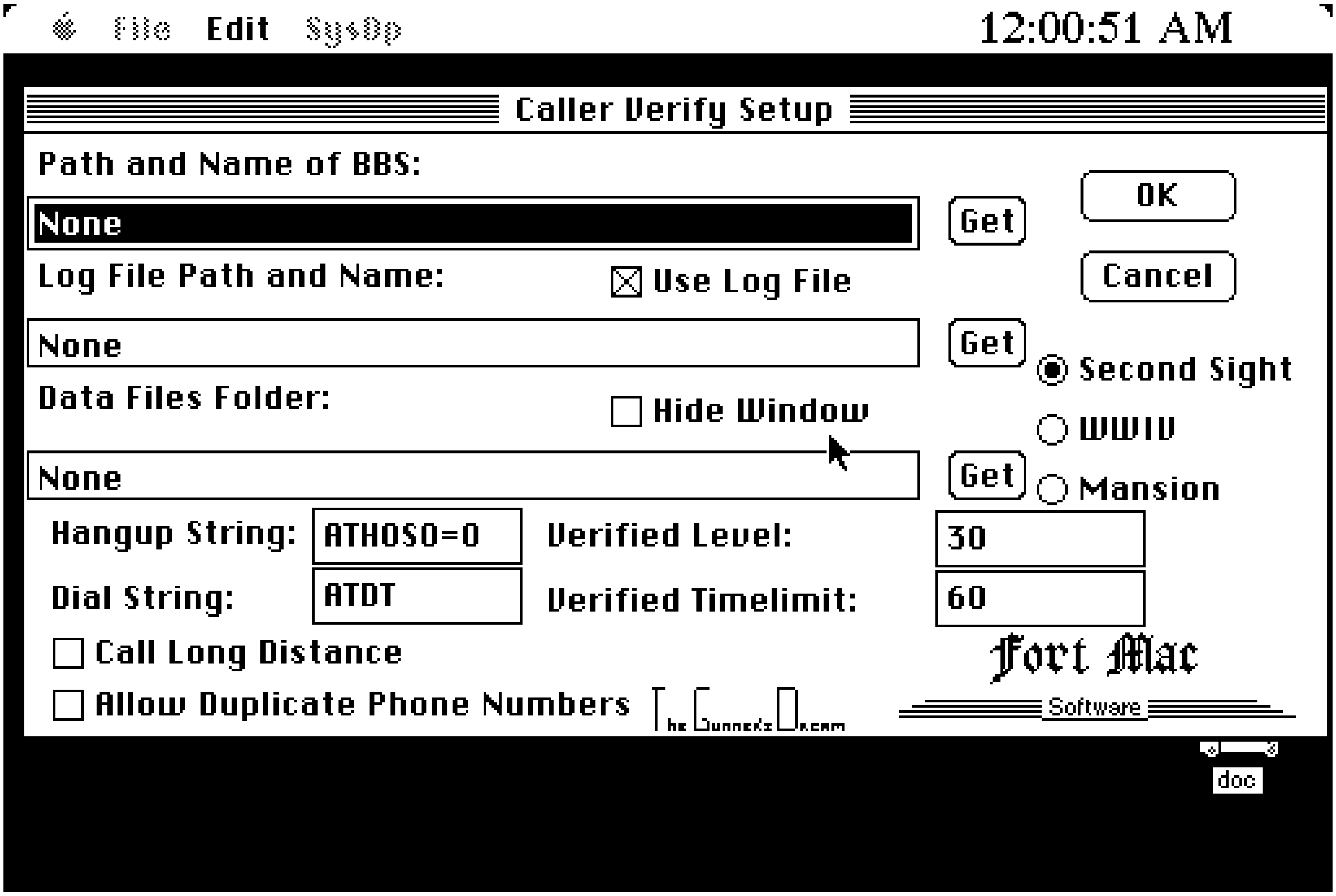
3. Setup

CallerVerify is quite simple to setup. Simply unpack the archive into your BBS folder. Within that folder should be the following heirarchy:

BBS Folder:
Second Sight (or Red Ryder Host)
CallerVerify
CV Folder

CV Folder:	What
CallerVerify Reg	(Furnished by me when you register CV <grin>)
CVLog	A sample log file. Should be empty.
Used Phone Numbers	The storage area for used phone numbers. Should also be empty.
CallerVerify Welcome	A simple text file with a short description of CallerVerify.
Duplicate Number Instructions	The text file that is displayed when a previously used phone number has been used by the current user.
Long Distance Instructions	The text file that is displayed when a user is calling from long distance and you do not support long distance callbacks.

When you have it setup as above, double click on CallerVerify. CV will then display the setup dialog.



Caller Verify Setup

Path and Name of BBS: None

Log File Path and Name: None ☒ Use Log File

Data Files Folder: None □ Hide Window

Hangup String: ATHOSO=0

Dial String: ATDT

Verified Level: 30

Verified Timelimit: 60

□ Call Long Distance

□ Allow Duplicate Phone Numbers

● Second Sight
○ WWIV
○ Mansion

In the first text entry item, you will need to enter the BBS folder path along with the name of the BBS. Click on the 'Get' button to select the BBS and path.

In the second text entry item, you will need to enter the log file path. Again, click on 'Get' to select the path. Note: You may use your callerlog as the log file. If you do not want a log file (I HIGHLY recommend it -- all users that are verified are logged, and, all errors are logged also), de-highlight the Use Log File check box.

In the third text entry item, you will need to tell CV where it's data files folder is located. Select the folder via 'Get'.

Further down, you see the modem hangup string. Generally, I recommend ATH0S0=0. This allows the modem to hang up and avoid people calling in during the verification process. (NOTE: I am assuming that +++ hangs up your modem. If it does not (e.g. you've changed your modem hangup character), call me and I will build a version for you with the hangup string.

Keeping going, you see the modem dial string. This is tacked onto the start of the user's phone number to dial. Generally, I'd recommend ATDT, but if you've got to dial out through a PBX or such, you can add whatever you wish.

Next, you will see two check boxes: *Allow Duplicate Phone Numbers* and *Call Long Distance*. If you wish to allow duplicate phone numbers (e.g. the user will be given access if his phone number is one that has already been used to gain access), check the check box. If you wish to allow long distance callbacks, check the *Call Long Distance* check box.

Moving to the right, you will need to select your bbs type. Note that Second Sight is the only supported BBS at this time. (Mansion should be following based on demand) DO NOT check anything other than SS as unpredictible results will follow. (e.g. system crashes)

Further down from there you see the *Verified User Level* text entry area. On my system, users log on with access level 25. When they are verified, they move to access level 30. Consequently, I type in 30 in this text entry.

Below that, you see *Verified User Timelimit*. On Fort Mac, I don't use different timelimits, but, if you wish your temporary users to have only, say, 10 minutes, you'd enter your verified timelimit here. (e.g. 10 minutes for a non-user and 60 minutes for a verified user).

And that's it! Click on OK and you should be ready to run!

4. <u>Registration Information</u>

You may register CallerVerify by sending $40.00 in US currency to:

Fort Mac Software
15651 E. Caspian Circle #203
Aurora, CO 80013

Please fill out the registration form and send it to me. If you do not register CallerVerify within the 19 day trial period, nothing will happen. Meaning, CallerVerify will fail to validate users. It will give a 'I'm closed' message and exit to the BBS.

5. In Detail

Let me explain, in detail, what happens during a typical CallerVerify session.

a. A user logs onto the bbs. You get his name, number, etc.

b. After reading the system messages and such, he is shot out to CallerVerify. He is shown the intro as well as a short description of the program.

c. He is asked to enter his phone number and told what to do when the program calls him back. (e.g. ATA from his keyboard)

d. The program hangs up the modem.

e. The program waits a few seconds and calls the user back using his phone number.

f. If a connection is made, he is then asked to enter his password. If he types it in correctly (he gets 3 tries), he is validated and returned to the BBS.

g. If no connection is made, the CV program tries 2 more times (for a total of 3) and gives up. It returns to the BBS at that point.

6. Security

CallerVerify is a very security aware product. It tries its best to not allow any loopholes that a user may use to gain unauthorized access to the BBS. This includes the phone number lookup as well as the 'ATS0=0' string that is part of the configuration. Some other security issues follow:

a. A user must not be able to call back quickly to 'fake out' the callback verifier into thinking it got a successful connect. This is the purpose of the S0=0 addition to the hangup string.

b. The user shouldn't be able to enter a '1' when the sysop does not allow long distance callbacks. This is checked for.

c. When the no duplicate phone numbers is enabled, phone numbers are reduced to a no-hypen listing. This allows the
check vs. the used phone numbers log to be complete. (e.g. 5551212 vs. 555-1212)

Ways Around CallerVerify:

There are a number of ways to subvert CallerVerify. One, a user may simply call back by another phone. After he has been verified, he has 'full' access. He could switch the phone line back and be a 'new' user. He may have a friend type in his password as a callback bypass.

That's all I can think of offhand. If you find some other loophole, please let me know.

7. Why I wrote CallerVerify

I run a pretty files-oriented BBS. About 60% of my users are interested in files. Previous to CallerVerify, I used to have an auto-validation script. You'd type in your name, your phone number and a short "Why do you want access?" sentence and you'd have access. Since files are a primary part of the BBS, I had a serious leech problem. I finally got sick enough of it to write CallerVerify. Since I've installed CallerVerify (about two weeks ago), I'd had 4 be discovered by the program. It's very efficient about discovering falsifications.

Well, I believe that is it. I think you'll really love CallerVerify. I know it really takes a load off of my mind -- it's like having your own automatic sub-op!

Greg Shaw
10/5/92