



Contents

1. Building GDB for Darwin
2. Giving gdb permission to control other processes
 1. Method for Mac OS X 10.5 (Leopard) and later
 1. Create a certificate
 2. Refresh the system's certificates and code-signing data
 3. Code-sign the gdb binary
 2. Old method
 3. Troubleshooting code-signing

1. Building GDB for Darwin

Creating the binary for Darwin isn't very difficult.  Download a release snapshot or  get the current source via git/CVS/FTP, then configure and make as usual.

Building the 7.0 release unfortunately results in many "warning: format not a string literal and no format arguments" warnings. This problem has been fixed in CVS. To avoid such warnings building 7.0, configure with `--disable-intl`. Later releases are fine and do not require any special build options.

2. Giving gdb permission to control other processes

If you try to use your freshly built gdb, you will get an error message such as:

```
Starting program: /x/y/foo
Unable to find Mach task port for process-id 28885: (os/kern) failure
(0x5).
(please check gdb is codesigned - see taskgated(8))
```

This is because the Darwin kernel will refuse to allow gdb to debug another process if you don't have special rights, since debugging a process means having full control over that process, and that isn't allowed by default since it would be exploitable by malware. (The kernel won't refuse if you are root, but of course you don't want to be root to debug.)

2.1. Method for Mac OS X 10.5 (Leopard) and later

The most up to date and secure method to allow gdb to control another process is to sign it with any system-trusted code signing authority.

2.1.1. Create a certificate

Start Keychain Access application
(`/Applications/Utilities/Keychain Access.app`)

Open the menu item
`/Keychain Access/Certificate Assistant/Create a Certificate...`

Choose a name (`gdb-cert` in the example), set `Identity Type` to

Self Signed Root, set Certificate Type to Code Signing and select the Let me override defaults. Click several times on Continue until you get to the Specify a Location For The Certificate screen, then set Keychain to System.

If you cannot store the certificate in the System keychain, create it in the login keychain, then exported it. You can then import it into the System keychain.

Using the contextual menu for the certificate, select Get Info, open the Trust item, and set Code Signing to Always Trust.

Finally, quit Keychain Access application to refresh the certificate store.

2.1.2. Refresh the system's certificates and code-signing data

The most reliable way is to reboot your system.

A less invasive way is to and restart taskgated service by killing the current running taskgated process (so before using gdb). However, sometimes the taskgated service will not restart successfully after killing it, so ensure that it is alive after this step by checking e.g. ps. Or just reboot your system, as mentioned above.

2.1.3. Code-sign the gdb binary

If the certificate is known as gdb-cert, use:

```
$ codesign -s gdb-cert gdb
```

Make sure to specify the full path to the gdb binary. You may also have to prepend this command with sudo if the gdb binary is located in a place that is not writable by regular users.

If debugging still does not work, check the troubleshooting section.

2.2. Old method

Warning: do *not* combine (part of) these instructions with the code signing method, because they interfere with each other. See the troubleshooting section in case you (possibly) did mix them to fix up any potential issues. In general, following the instructions in this section is *strongly unrecommended* if you are using Mac OS X 10.6 (Snow Leopard) or later.

In Mac OS X 10.4 (Tiger), the kernel would accept processes whose primary effective group is procmod or procview. That meant that making gdb setgid procmod worked.

Later versions of Darwin still accept this convention provided that taskgated (the daemon that controls the access) is invoked with option '-p'. This daemon is configured by /System/Library/LaunchDaemons/com.apple.taskgated.plist, which is where you can add this '-p' option. On OS X 10.11 (El Capitan) and later, this requires (temporarily) disabling rootless mode, otherwise that file cannot be modified.

On OS X 10.10 (Yosemite) and later, you also need to be a member of the Unix group `_developer` for this method to work. In case your use name is Jack, you can do this with the following command:

```
$ sudo dscl . merge /Groups/_developer GroupMembership Jack
```

Follow the instructions above to refresh all subsystems that need to know about this group change.

3. Troubleshooting code-signing

If you used the code-signing method and you still get the `(os/kern) failure (0x5)` error, one of the following items may pinpoint you to the cause.

- Ensure that the certificate is in the `System` keychain
- Ensure that the certificate is marked as `Always Trust`
- Ensure that the `gdb` binary is correctly signed with the certificate. You can check this with the following command:

```
$ codesign -v gdb
```

(use the full path to the `gdb` binary)

If this command does not print anything, it means the binary is correctly signed. If there is a problem with the code signature, it will print an error message. In the latter case, re-sign it.

- You may have also (partly) followed the instructions from the old method section. E.g., if the `gdb` binary is `setgid procmod`, the code-signing method will *not* work anymore. In that case, restore the `gdb` binary to the original settings with the following commands:

```
$ chmod 755 gdb  
$ chgrp admin gdb
```

(possibly prepended with `sudo` if the `gdb` binary is located in a place that cannot be modified by regular users or if you do not belong to the `admin` group, and replacing `gdb` with the full path to the `gdb` binary)

- Ensure that you code-signed the actual `gdb` binary and not just a script that starts this binary (you can code-sign shell scripts too). Check using e.g. the `file` command that the file you code-signed is a binary and not a text file or shell script.

None: BuildingOnDarwin (last edited 2016-06-13 14:14:48 by JonasMaebe)

All content (C) 2008 Free Software Foundation. For terms of use, redistribution, and modification, please see the [WikiLicense](#) page.