

# **GT 5.0.2 GSI-OpenSSH: System Administrator's Guide**

---

## GT 5.0.2 GSI-OpenSSH: System Administrator's Guide

### Introduction

This guide contains advanced configuration information for system administrators working with GSI-OpenSSH. It provides references to information on procedures typically performed by system administrators, including installation, configuring, deploying, and testing the installation.

### Important

This information is in addition to the basic Globus Toolkit prerequisite, overview, installation, security configuration instructions in [Installing GT 5.0.2](#). Read through this guide before continuing!

This guide is meant solely to cover the GSI aspects of GSI-OpenSSH, and is not meant to be a full manual for OpenSSH itself. Please refer to the [OpenSSH Home Page](#)<sup>1</sup> for general documentation for OpenSSH.

---

<sup>1</sup> <http://www.openssh.org/>

---

---

# Table of Contents

1. Building and Installing .....	1
1. Optional Build-Time Configuration .....	1
2. Building and Installing only GSI-OpenSSH .....	1
2. Configuring .....	3
3. Deploying .....	4
4. Testing .....	6
5. Security Considerations .....	7
1. GSI-OpenSSH Security Considerations .....	7
6. Debugging .....	8
1. Logging .....	8
7. Troubleshooting .....	9
8. Usage statistics collection .....	10
1. GSI-OpenSSH usage statistics collection .....	10

---

## List of Tables

1.1. GSI-OpenSSH build arguments .....	1
--	---

---

# Chapter 1. Building and Installing

GSI-OpenSSH is built and installed as part of a default GT 5.0.2 installation. For basic installation instructions, see [Installing GT 5.0.2](#). No extra installation steps are required for this component.

## 1. Optional Build-Time Configuration

You can optionally pass build-time configure options to the GSI-OpenSSH package using the `--with-gsiopensshargs` option when running `configure` for your GT 5.0.2 installation. For example:

```
./configure --prefix=$HOME/globus
            --with-gsiopensshargs="--with-pam"
```

No options are typically needed for client-only installations, but options are often needed for full server functionality. The following table lists suggested options for different platforms.

**Table 1.1. GSI-OpenSSH build arguments**

Platform	Configuration
Linux	<code>--with-pam --with-md5-passwords --with-tcp-wrappers</code>
Solaris	<code>--with-pam --with-md5-passwords --with-tcp-wrappers</code>
Irix	<code>--with-tcp-wrappers</code>
AIX	<code>--with-tcp-wrappers</code>

Note: If you enable PAM support with the `--with-pam` configuration option, be sure to also set "UsePAM yes" in `$GLOBUS_LOCATION/etc/ssh/sshd_config` after installation.

If you have an already configured and installed system-wide SSHD and you would like your build of GSI-OpenSSH to behave similarly, investigate the `configure` options available in GSI-OpenSSH and select those options that would add the functionality that your current SSHD possesses. Be aware that since GSI-OpenSSH is based on OpenSSH, the standard set of functionality is turned on by default.

*Please do not attempt to override the following options:*

```
--prefix
--sysconfdir
--with-globus
--with-globus-flavor
--with-ssl-dir
```

## 2. Building and Installing only GSI-OpenSSH

If you wish to install GSI-OpenSSH without installing the rest of the Globus Toolkit, follow the instructions in [Installing GT 5.0.2](#) with the following changes. First, you do not need Ant, a JDK, or a JDBC database to build only GSI-OpenSSH. Second, instead of running "make", run:

```
globus$ make gsi-openssh
```

This will install the GSI-OpenSSH client and server programs. For client-only installations, simply do not configure or use the installed server.

---

# Chapter 2. Configuring

The GSI-enabled OpenSSH software is installed with a default set of configuration files, described below. You may want to modify the `ssh_config` file before using the clients and the `sshd_config` file before using the server.

If the GSI-enabled OpenSSH install script finds existing SSH key pairs, it will create symbolic links to them rather than generating new key pairs. The SSH key pairs are not required for GSI authentication. However, if you wish to support other SSH authentication methods, make sure the `sshd` (running as root) can read the key pair files (i.e., beware of NFS mounts with `root_squash`). If running multiple `sshd`s on a system, we recommend configuring them so they all use the same key pairs (i.e., use symbolic links) to avoid client-side confusion.

- `$GLOBUS_LOCATION/etc/ssh/moduli`  
`moduli` is a crypto parameter for generating keys.
- `$GLOBUS_LOCATION/etc/ssh/ssh_config`  
`ssh_config` contains options that are read by `ssh`, `scp`, and `sftp` at run-time. The installed version is the default provided by OpenSSH, with GSI authentication and `X11Forwarding` enabled. You may need to customize this file for compatibility with your system SSH installation (i.e., compare it with `/etc/ssh/ssh_config`).
- `$GLOBUS_LOCATION/etc/ssh/ssh_host_key[.pub]`  
Your system's RSA public-/private-key pair for SSH protocol 1 communications.
- `$GLOBUS_LOCATION/etc/ssh/ssh_host_dsa[.pub]`  
Your system's DSA public-/private-key pair for SSH protocol 2 communications.
- `$GLOBUS_LOCATION/etc/ssh/ssh_host_rsa[.pub]`  
Your system's RSA public-/private-key pair for SSH protocol 2 communications.
- `$GLOBUS_LOCATION/etc/ssh/ssh_prng_cmds`  
`ssh_prng_cmds` contains paths to a number of files that `ssh-keygen` may need to use if your system does not have a built-in entropy pool (like `/dev/random`).
- `$GLOBUS_LOCATION/etc/ssh/sshd_config`  
`sshd_config` contains options that are read by `sshd` when it starts up. The installed version is the default provided by OpenSSH, with `X11Forwarding` enabled. You may need to customize this file for compatibility with your system SSH installation (i.e., compare it with `/etc/ssh/sshd_config`). For example, to enable PAM authentication, you may need to set "UsePAM yes" in this file.

---

# Chapter 3. Deploying

1. To install the GSI-Enabled OpenSSH Server on most systems, you must be a privileged user, such as root.

```
sh$ /bin/su - root
```

Note: If your system functions like this and you attempt to run these commands as a user other than root, these commands should fail.

2. (optional) Start a copy of your system's currently running SSH server on an alternate port by running, eg.

```
sh# /usr/sbin/sshd -p 2000 &
```

You may then choose to log in to this server and continue the rest of these steps from that shell. We recommend doing this since some `sshd` shutdown scripts do particularly nasty things like killing *all* of the running SSH servers on a system, not just the parent server that may be listening on port 22. Roughly translated, this step is about guaranteeing that an alternate method of access is available should the main SSH server be shutdown and your connection via that server be terminated.

3. Locate your server's startup/shutdown script directory. On some systems this directory may be located at `/etc/rc.d/init.d`, but since this location is not constant across operating systems, for the purposes of this document we will refer to this directory as `INITDIR`. Consult your operating system's documentation for your system's location.
4. Run the following command.

```
sh# mv $INITDIR/sshd $INITDIR/sshd.bak
```

5. Either copy or link the new `sshd` script to your system's startup/shutdown script directory.

```
sh# cp $GLOBUS_LOCATION/sbin/SXXsshd $INITDIR/sshd
```

6. Shutdown the currently running main SSH server.

```
sh# $INITDIR/sshd.bak stop
```

7. Provided you still have a connection to the machine, start the new SSH server.

```
sh# $INITDIR/sshd start
```

8. Test the new server by connecting to the standard SSH port (22) and authenticating via multiple methods. Especially test that GSI authentication works correctly.
9. If you are performing a new install, or if the old server was not configured to be started at run-time and shutdown automatically at system halt or reboot, either use a system utility such as RedHat's `chkconfig` to configure the system for the correct run-levels, or manually link up the correct run-levels.

```
sh# /sbin/chkconfig sshd reset
```

The recommended run-levels are listed in a set of comments within the SXXsshd startup script. For example, on standard Unix systems we recommend running the GSI-Enabled OpenSSH server in run-levels two, three, four, and five.

10. Finally, if, as a precautionary measure, you started a SSH server on an alternate port in order to complete the install process, you can now safely stop all instances of that server.

---

# Chapter 4. Testing

1. Edit the file `$GLOBUS_LOCATION/sbin/SXXsshd` so that the GSI-Enabled OpenSSH server starts up on an alternate port.
2. Run the command

```
sh# $GLOBUS_LOCATION/sbin/SXXsshd start
```

and verify that the server is running by checking that it both shows up in a process listing and creates a file named `$GLOBUS_LOCATION/var/sshd.pid`.

3. From a remote machine attempt to connect to the local server on the modified test port using the standard SSH authentication methods plus authenticating via your GSI credentials. This may require you to authorize these users via an appropriate entry in the `grid-mapfile`.
4. Stop the SSH server by running the command

```
sh# $GLOBUS_LOCATION/sbin/SXXsshd stop
```

and reverse any changes you made that altered the port on which the server resided upon startup. After this step, running `SXXsshd start` should start the server on the default port (22).

---

# Chapter 5. Security Considerations

## 1. GSI-OpenSSH Security Considerations

GSI-OpenSSH is a modified version of [OpenSSH](http://www.openssh.org/)<sup>1</sup> and includes full OpenSSH functionality. For more information on OpenSSH security, see the [OpenSSH Security](http://www.openssh.org/security.html)<sup>2</sup> page.

---

<sup>1</sup> <http://www.openssh.org/>

<sup>2</sup> <http://www.openssh.org/security.html>

---

# Chapter 6. Debugging

## 1. Logging

The GSI-OpenSSH server logs to the system logger (syslog). Log messages from OpenSSH are unmodified in GSI-OpenSSH. GSI-specific log messages include:

```
failed to map GSI user DN
GSI user DN mapped to target user username
GSI user DN is authorized as target user username
GSI user DN is not authorized as target user username
```

In the messages above, DN is replaced with the client certificate subject distinguished name, and username is replaced with the local Unix username.

---

# Chapter 7. Troubleshooting

Please see the [GSI-OpenSSH Troubleshooting Page](#)<sup>1</sup>. For a list of common errors in GT, see [Error Codes](#).

---

<sup>1</sup> [http://grid.ncsa.illinois.edu/ssh/ts\\_common.html](http://grid.ncsa.illinois.edu/ssh/ts_common.html)

---

# Chapter 8. Usage statistics collection

## 1. GSI-OpenSSH usage statistics collection

By default, the GSI-OpenSSH server (sshd) will send a UDP packet containing usage information to `usage-stats.cilogon.org:4810` after serving each request. See the [Privacy Policy](#)<sup>1</sup> for details. You can see the Globus Usage Statistics policy [here](#)<sup>2</sup>.

---

<sup>1</sup> <http://grid.ncsa.illinois.edu/ssh/privacy.html>

<sup>2</sup> [/toolkit/docs/latest-stable/Usage\\_Stats.html](/toolkit/docs/latest-stable/Usage_Stats.html)