

# **GT 5.0.0 GSI-OpenSSH: System Administrator's Guide**

---

# GT 5.0.0 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 5.0.0](#). 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

---

## 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 5.0.0 installation. For basic installation instructions, see [Installing GT 5.0.0](#). 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 5.0.0 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 5.0.0](#) 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)

# **GT 5.0.0 GSI-OpenSSH: User's Guide**

---

# GT 5.0.0 GSI-OpenSSH: User's Guide

## Introduction

This is a guide for using the GSI-enabled OpenSSH client. It assumes that you (or your system administrator) have already installed the GSI OpenSSH and that you have also acquired a *user certificate* from an appropriate *Certificate Authority*.

---

---

# Table of Contents

1. Using GSI-OpenSSH .....	1
1. Creating a proxy .....	1
2. Deleting a proxy .....	1
3. Getting authorized to connect to a site .....	1
I. GSI-OpenSSH Commands .....	?
gsssh .....	3
gsscp .....	4
gssftp .....	5
2. Troubleshooting .....	6
1. Errors .....	7
2. The gsssh command prompts you for a pass phrase when you run it .....	8
3. Debugging .....	9
1. Specifying verbose output .....	9
Glossary .....	10

---

## List of Tables

2.1. GSI-OpenSSH Errors .....	7
-------------------------------	---

---

# Chapter 1. Using GSI-OpenSSH

## 1. Creating a proxy

First, set the `GLOBUS_LOCATION` environment variable to the location of your GSI-enabled OpenSSH installation. It may already be set for you by your system administrator.

Then, create a *proxy credential* for GSI authentication by running the **grid-proxy-init** program. This is your single sign-on to the Grid. By default, **grid-proxy-init** will create a proxy credential good for 12 hours.

To create a proxy credential with a different lifetime, use the **-hours** option.

For example:

```
% grid-proxy-init -hours 8
```

## 2. Deleting a proxy

To delete a proxy that was previously create with `grid-proxy-init`, run:

```
% grid-proxy-destroy
```

## 3. Getting authorized to connect to a site

Before you can connect to a site, the site needs to know the identity in your certificate so that it can map that identity to your local account. At a minimum, the site will need to know your subject name from your certificate. You can get your subject name by running **grid-cert-info** with the **-subject** argument. For example:

```
% grid-cert-info -subject
```

Email your subject name to the administrator of the system you wish to connect to so that they can add your entry to the appropriate authorization files.

Once you have your proxy credential, all you should have to do is run `gsissh`, providing it with the hostname of the host you want to connect to. For example:

```
% gsissh myhost.somedomain.edu
```

You should then find yourself automatically logged into your account on the remote system. If something goes wrong, please see [Troubleshooting](#) for assistance.

---

# GSI-OpenSSH Commands

The `gssissh(1)`, `gssiscp(1)`, and `gssisftp(1)` commands provide the same interfaces as the standard OpenSSH `ssh`, `scp`, and `sftp` commands, respectively, with the added ability to perform X.509 *proxy credential* authentication and delegation.

---

## Name

`gsissh` -- Secure remote login

`gsissh`

## Tool description

Use the `gsissh` command to securely login to a remote machine.

## Command syntax

`gsissh` [-l login\_name] hostname | user@hostname [command]

---

## Name

`gsiscp` -- Secure remote file copy

`gsiscp`

## Tool description

Use the `gsiscp` command to securely copy files to or from a remote machine.

## Command syntax

`gsiscp [-P port] [[user@]host1:]file1 [...] [[user@]host2:]destfile`

---

# Name

`gsisftp` -- Secure file transfer

`gsisftp`

# Tool description

The *gsisftp* command provides an interactive interface for transferring files to and from remote machines.

# Command syntax

`gsisftp` [[user@]host[:dir[/]]]

---

# Chapter 2. Troubleshooting

Some common errors are listed below. If you need additional assistance, please run `gsssh` with the `'-vvv'` argument (specifying verbose output) and send the output to your system administrator for assistance.

For a list of common errors in GT, see [Error Codes](#).

# 1. Errors

**Table 2.1. GSI-OpenSSH Errors**

<b>Error Code</b>	<b>Definition</b>	<b>Possible Solutions</b>
GSS-API error Failure acquiring GSSAPI credentials: GSS_S_CREDENTIALS_EXPIRED	This means that your proxy certificate has expired.	Run <code>grid-proxy-init</code> to acquire a new proxy certificate, then run <code>gsissh</code> again.
...no proxy credentials...	Failing to run <code>grid-proxy-init</code> to create a user proxy with which to connect will result in the client notifying you that no local credentials exist. Any attempt to authenticate using GSI will fail in this case.	Verify that your GSI proxy has been properly initialized via <code>grid-proxy-info</code> . If you need to initialize the proxy, use the command <code>grid-proxy-init</code> .
...bad file system permissions on private key; key must only be readable by the user...	The host key that the SSH server is using for GSI authentication must only be readable by the user which owns it. Any other permissions will cause this error.	Make sure that the host key's UNIX permissions are mode 400 (that is, it should only have mode readable for the user that owns the file, and no other mode bits should be set).
...gssapi received empty username; failed to set username from gssapi context; Failed external-keyx for <user> from <host> <port>...	If the server was passed an "implicit username" (i.e. requested to map the incoming connection to a username based on some contextual clues such as the certificate's subject), and no entry exists in the <code>grid-mapfile</code> for the incoming connection's certificate subject, the server should output a clue that states it is unable to set the username against which to authenticate.	Add an entry for the user to the <a href="#">Section 1.2, "Gridmap file"</a> .
...INTERNAL ERROR: authenticated invalid user xxx...	If the subject name given in the system's <code>grid-mapfile</code> points to a non-existent user, the server will give an internal error which is best caught when it is running in debugging mode.	Add a new account to the system matching the username pointed at by the user's subject in the <code>grid-mapfile</code> .
...gssapi received empty username; no suitable client data; failed to set username from gssapi context; Failed external-keyx for <user> from <host> <port>...	Should the user attempt to connect without first creating a proxy certificate, or if the user is connecting via a SSH client that does not support GSI authentication, the server will note that no GSSAPI data was sent to it. Verify that the client is able to connect through another GSI service (such as the gatekeeper) to make sure that the user's proxy has been created correctly.	Verify that you are using a GSI-enabled SSH client and that your GSI proxy has been properly initialized via <code>grid-proxy-info</code> . If you need to initialize this proxy, use the command <code>grid-proxy-init</code> .

## 2. The `gsissh` command prompts you for a pass phrase when you run it

This could mean that you don't have a proxy certificate; try running **`grid-proxy-init`** and then running `gsissh` again. It could also mean that the GSI authentication is failing for some reason and `gsissh` is falling back to a different authentication mechanism. Reasons that it might fail include:

- The host you are connecting to does not have a GSI-enabled OpenSSH server.
- You are not authorized to use GSI authentication to the host. Contact the administrator.

---

# Chapter 3. Debugging

For information about sys admin debugging, see [Debugging](#).

## 1. Specifying verbose output

If you need additional assistance, please run `gsissh` with the `'-vvv'` argument (specifying verbose output) and send the output to your system administrator for assistance.

---

# Glossary

## C

Certificate Authority ( CA )      An entity that issues certificates.

## P

proxy credentials      The combination of a proxy certificate and its corresponding private key. GSI typically stores proxy credentials in `/tmp/x509up_u<uid>` , where `<uid>` is the user id of the proxy owner.

## U

user certificate      A EEC belonging to a user. When using GSI, this certificate is typically stored in `$HOME/.globus/usercert.pem`. For more information on possible user certificate locations, see [this](#).

# **GT 5.0.0 GSI-OpenSSH: Developer's Guide**

---

# GT 5.0.0 GSI-OpenSSH: Developer's Guide

## Introduction

This document provides information for GSI-OpenSSH developers.

The changes to OpenSSH<sup>1</sup> to add GSI support are limited, because we build on the existing GSSAPI support in OpenSSH for Kerberos. See the GSI-OpenSSH Internals<sup>2</sup> page for more details.

---

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

<sup>2</sup> <http://dev.globus.org/wiki/GSI-OpenSSH/Internals>

---

---

# Table of Contents

1. Before you begin .....	1
1. Feature summary .....	1
2. Tested platforms .....	1
3. Backward compatibility summary .....	1
4. Technology dependencies .....	1
5. GSI-OpenSSH Security Considerations .....	2
2. Usage scenarios .....	3
3. Tutorials .....	4
4. Architecture and design overview .....	5
I. GSI-OpenSSH Commands .....	?
gsssh .....	7
gsscp .....	8
gssftp .....	9
5. Configuring .....	10
6. Environment variable interface .....	11
1. Environmental variables for GSI-OpenSSH .....	11
7. Troubleshooting .....	12
1. Common GT Errors .....	12
2. Errors .....	13
3. Additional GSI-OpenSSH Troubleshooting .....	13
8. Related Documentation .....	14
Glossary .....	15

---

## List of Tables

7.1. GSI-OpenSSH Errors .....	13
-------------------------------	----

---

# Chapter 1. Before you begin

## 1. Feature summary

Features new in GT 5.0.0

- None.

Other Supported Features

- The **gsissh** command provides a secure remote login service with forwarding of X.509 *proxy credentials*.
- The **gsiscp** and **gsisftp** commands provide a secure file transfer service authenticated with X.509 proxy credentials, mimicking the **rcp/scp** and **ftp/sftp** commands.
- All standard OpenSSH features are supported, excluding Kerberos authentication. Kerberos authentication is *not* compatible with GSI-enabled OpenSSH.
- The GSI-OpenSSH server can replace the standard system SSH server in typical environments.
- If no username is given on the command-line, GSI-OpenSSH automatically determines the username that corresponds to the X.509 proxy *certificate subject* in the server's `grid-mapfile`.

Deprecated Features

- None

## 2. Tested platforms

Tested Platforms for GSI-OpenSSH

- Mac OS X 10.5
- x86/x86\_64 GNU/Linux
- PPC AIX 5.3
- Sun4u Solaris 5.10

## 3. Backward compatibility summary

GSI-OpenSSH is backward compatible.

## 4. Technology dependencies

GSI-OpenSSH depends on the following GT components:

- GSLC

GSI-OpenSSH depends on the following 3rd party software:

- [OpenSSH](#)<sup>1</sup>

## 5. GSI-OpenSSH Security Considerations

GSI-OpenSSH is a modified version of [OpenSSH](#)<sup>2</sup> and includes full OpenSSH functionality. For more information on OpenSSH security, see the [OpenSSH Security](#)<sup>3</sup> page.

---

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

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

<sup>3</sup> <http://www.openssh.org/security.html>

---

# Chapter 2. Usage scenarios

The GSI-OpenSSH interface is through command-line tools only.

---

# Chapter 3. Tutorials

There are no tutorials available at this time

---

# Chapter 4. Architecture and design overview

See the [GSI-OpenSSH Internals](http://dev.globus.org/wiki/GSI-OpenSSH/Internals)<sup>1</sup> page.

---

<sup>1</sup> <http://dev.globus.org/wiki/GSI-OpenSSH/Internals>

---

# GSI-OpenSSH Commands

The `gssish(1)`, `gssscp(1)`, and `gssisftp(1)` commands provide the same interfaces as the standard OpenSSH `ssh`, `scp`, and `sftp` commands, respectively, with the added ability to perform X.509 *proxy credential* authentication and delegation.

---

## Name

`gsissh` -- Secure remote login

`gsissh`

## Tool description

Use the `gsissh` command to securely login to a remote machine.

## Command syntax

`gsissh` [-l login\_name] hostname | user@hostname [command]

---

## Name

`gsiscp` -- Secure remote file copy

`gsiscp`

## Tool description

Use the `gsiscp` command to securely copy files to or from a remote machine.

## Command syntax

`gsiscp [-P port] [[user@]host1:]file1 [...] [[user@]host2:]destfile`

---

## Name

`gsisftp` -- Secure file transfer

`gsisftp`

## Tool description

The *gsisftp* command provides an interactive interface for transferring files to and from remote machines.

## Command syntax

`gsisftp` [[user@]host[:dir[/]]]

---

# Chapter 5. 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 6. Environment variable interface

## 1. Environmental variables for GSI-OpenSSH

The GSI-enabled OpenSSH needs to be able to find certain files and directories in order to properly function.

The items that OpenSSH needs to be able to locate, their default location and the environment variable to override the default location are:

- *Host key*
  - Default location: /etc/grid-security/hostkey.pem
  - Override with X509\_USER\_KEY environment variable
- *Host certificate*
  - Default location: /etc/grid-security/hostcert.pem
  - Override with X509\_USER\_CERT environment variable
- *Grid map file*
  - Default location: /etc/grid-security/grid-mapfile
  - Override with GRIDMAP environment variable
- *Certificate directory*
  - Default location: /etc/grid-security/certificates
  - Override with X509\_CERT\_DIR environment variable

---

# Chapter 7. Troubleshooting

## 1. Common GT Errors

For a list of common errors in GT, see [Error Codes](#).

## 2. Errors

**Table 7.1. GSI-OpenSSH Errors**

Error Code	Definition	Possible Solutions
GSS-API error Failure acquiring GSSAPI credentials: GSS_S_CREDENTIALS_EXPIRED	This means that your proxy certificate has expired.	Run <code>grid-proxy-init</code> to acquire a new proxy certificate, then run <code>gsissh</code> again.
...no proxy credentials...	Failing to run <code>grid-proxy-init</code> to create a user proxy with which to connect will result in the client notifying you that no local credentials exist. Any attempt to authenticate using GSI will fail in this case.	Verify that your GSI proxy has been properly initialized via <code>grid-proxy-info</code> . If you need to initialize the proxy, use the command <code>grid-proxy-init</code> .
...bad file system permissions on private key; key must only be readable by the user...	The host key that the SSH server is using for GSI authentication must only be readable by the user which owns it. Any other permissions will cause this error.	Make sure that the host key's UNIX permissions are mode 400 (that is, it should only have mode readable for the user that owns the file, and no other mode bits should be set).
...gssapi received empty username; failed to set username from gssapi context; Failed external-keyx for <user> from <host> <port>...	If the server was passed an "implicit username" (i.e. requested to map the incoming connection to a username based on some contextual clues such as the certificate's subject), and no entry exists in the <code>grid-mapfile</code> for the incoming connection's certificate subject, the server should output a clue that states it is unable to set the username against which to authenticate.	Add an entry for the user to the <a href="#">Section 1.2, "Gridmap file"</a> .
...INTERNAL ERROR: authenticated invalid user xxx...	If the subject name given in the system's <code>grid-mapfile</code> points to a non-existent user, the server will give an internal error which is best caught when it is running in debugging mode.	Add a new account to the system matching the username pointed at by the user's subject in the <code>grid-mapfile</code> .
...gssapi received empty username; no suitable client data; failed to set username from gssapi context; Failed external-keyx for <user> from <host> <port>...	Should the user attempt to connect without first creating a proxy certificate, or if the user is connecting via a SSH client that does not support GSI authentication, the server will note that no GSSAPI data was sent to it. Verify that the client is able to connect through another GSI service (such as the gatekeeper) to make sure that the user's proxy has been created correctly.	Verify that you are using a GSI-enabled SSH client and that your GSI proxy has been properly initialized via <code>grid-proxy-info</code> . If you need to initialize this proxy, use the command <code>grid-proxy-init</code> .

## 3. Additional GSI-OpenSSH Troubleshooting

Please see the [GSI-OpenSSH Troubleshooting Page](#)<sup>1</sup> at NCSA.

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

---

# Chapter 8. Related Documentation

Please see the [GSI-OpenSSH Home Page](http://grid.ncsa.uiuc.edu/ssh/)<sup>1</sup> at NCSA for more information.

---

<sup>1</sup> <http://grid.ncsa.uiuc.edu/ssh/>

---

# Glossary

## C

certificate subject

An identifier for the certificate owner, e.g. `"/DC=org/DC=doegrids/OU=People/CN=John Doe 123456"`. The subject is part of the information the CA binds to a public key when creating a certificate.

## G

grid map file

A file containing entries mapping certificate subjects to local user names. This file can also serve as a access control list for GSI enabled services and is typically found in `/etc/grid-security/grid-mapfile`. For more information see the Gridmap section [here](#).

## H

host certificate

An EEC belonging to a host. When using GSI this certificate is typically stored in `/etc/grid-security/hostcert.pem`. For more information on possible host certificate locations see the [GSI C Developer's Guide](#).

## P

proxy credentials

The combination of a proxy certificate and its corresponding private key. GSI typically stores proxy credentials in `/tmp/x509up_u<uid>`, where `<uid>` is the user id of the proxy owner.

---

# GT 5.0.0 Migrating Guide for GSI-OpenSSH

<titleabbrev>Migrating Guide</titleabbrev>

No special procedures are required for upgrading GSI-OpenSSH installations. GSI-OpenSSH is backward compatible.

---

# GT 5.0.0 GSI-OpenSSH: Quality Profile

## Table of Contents

1. Test coverage reports .....	1
2. Code analysis reports .....	1
3. Outstanding bugs .....	1
4. Bug Fixes .....	1
5. Performance reports .....	1

<titleabbrev>Quality Profile</titleabbrev>

## 1. Test coverage reports

Not yet available.

## 2. Code analysis reports

Not yet available.

## 3. Outstanding bugs

No bugs are known to exist for GSI-OpenSSH.

## 4. Bug Fixes

See the [GSI-OpenSSH Release History](#)<sup>1</sup> for details.

## 5. Performance reports

None.

---

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

---

# GT 5.0.0 Release Notes: GSI-OpenSSH

## Table of Contents

1. Component Overview .....	1
2. Feature summary .....	1
3. Summary of Changes in GSI-OpenSSH .....	2
4. Bug Fixes .....	2
5. Known Problems .....	2
6. Technology dependencies .....	2
7. Tested platforms .....	2
8. Backward compatibility summary .....	3
9. Associated Standards .....	3
10. For More Information .....	3
Glossary .....	3

<titleabbrev>Release Notes</titleabbrev>

## 1. Component Overview

GSI-OpenSSH is a modified version of OpenSSH that adds support for X.509 *proxy certificate* authentication and delegation, providing a single sign-on remote login and file transfer service. GSI-OpenSSH can be used to login to remote systems and transfer files between systems without entering a password, relying instead on a valid *proxy credential* for authentication. GSI-OpenSSH forwards proxy credentials to the remote system on login, so commands requiring proxy credentials (including GSI-OpenSSH commands) can be used on the remote system without the need to manually create a new proxy credential on that system. For more information about GSI-OpenSSH, see the [GSI-OpenSSH Home Page](#)<sup>1</sup>.

## 2. Feature summary

Features new in GT 5.0.0

- None.

Other Supported Features

- The **gsissh** command provides a secure remote login service with forwarding of X.509 *proxy credentials*.
- The **gsiscp** and **gsisftp** commands provide a secure file transfer service authenticated with X.509 proxy credentials, mimicking the **rcp/scp** and **ftp/sftp** commands.
- All standard OpenSSH features are supported, excluding Kerberos authentication. Kerberos authentication is *not* compatible with GSI-enabled OpenSSH.
- The GSI-OpenSSH server can replace the standard system SSH server in typical environments.

---

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

- If no username is given on the command-line, GSI-OpenSSH automatically determines the username that corresponds to the X.509 proxy *certificate subject* in the server's `grid-mapfile`.

#### Deprecated Features

- None

## 3. Summary of Changes in GSI-OpenSSH

GT 5.0.0 contains GSI-OpenSSH version 4.7. GSI-OpenSSH clients now attempt only GSI authentication by default, rather than the confusing behavior of attempting other SSH authentication methods when GSI authentication fails. (The GSI-OpenSSH server still supports both GSI and non-GSI authentication methods by default, for compatibility with both GSI and non-GSI clients.) See the GSI-OpenSSH Release History<sup>2</sup> for more details on this and other GSI-OpenSSH versions.

## 4. Bug Fixes

See the GSI-OpenSSH Release History<sup>3</sup> for details.

## 5. Known Problems

The following problems and limitations are known to exist for GSI-OpenSSH at the time of the 5.0.0 release:

### 5.1. Limitations

- No known limitations exist.

### 5.2. Outstanding bugs

No bugs are known to exist for GSI-OpenSSH.

## 6. Technology dependencies

GSI-OpenSSH depends on the following GT components:

- GSI C

GSI-OpenSSH depends on the following 3rd party software:

- OpenSSH<sup>4</sup>

## 7. Tested platforms

Tested Platforms for GSI-OpenSSH

- Mac OS X 10.5

---

<sup>2</sup> <http://grid.ncsa.illinois.edu/ssh/history.html>

<sup>3</sup> <http://grid.ncsa.illinois.edu/ssh/history.html>

<sup>4</sup> <http://www.openssh.org/>

- x86/x86\_64 GNU/Linux
- PPC AIX 5.3
- Sun4u Solaris 5.10

## 8. Backward compatibility summary

GSI-OpenSSH is backward compatible.

## 9. Associated Standards

Associated standards for GSI-OpenSSH:

- [RFC 2743](http://www.ietf.org/rfc/rfc2743.txt)<sup>5</sup> GSSAPI
- [RFC 2744](http://www.ietf.org/rfc/rfc2744.txt)<sup>6</sup> GSSAPI: C-bindings
- [RFC 4462](http://www.ietf.org/rfc/rfc4462.txt)<sup>7</sup> GSSAPI Authentication and Key Exchange for the SSH Protocol

## 10. For More Information

See [GSI-OpenSSH](#) more information about this component.

# Glossary

## C

certificate subject

An identifier for the certificate owner, e.g. `"/DC=org/DC=doegrids/OU=People/CN=John Doe 123456"`. The subject is part of the information the CA binds to a public key when creating a certificate.

## P

proxy certificate

A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.

For more information about types of proxy certificates and their compatibility in different versions of GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

proxy credentials

The combination of a proxy certificate and its corresponding private key. GSI typically stores proxy credentials in `/tmp/x509up_u<uid>`, where `<uid>` is the user id of the proxy owner.

---

<sup>5</sup> <http://www.ietf.org/rfc/rfc2743.txt>

<sup>6</sup> <http://www.ietf.org/rfc/rfc2744.txt>

<sup>7</sup> <http://www.ietf.org/rfc/rfc4462.txt>