UPDATE: As of June 2, 2023, you must ensure that your SUP client has been updated with the latest version 8.13 (available below). You must do one of the following to replace their SUP client manually:

1. Follow the installation instructions for the SUP client.
   or
2. Run the following command directly on top of the old client, or to a new file:
   ssh sup.nas.nasa.gov mesh-update --file=mc>sup
   If downloading the client to a new location, use chmod 700 sup to make the client executable, then move it to a location in your $PATH.

The Secure Unattended Proxy (SUP) allows you to perform remote operations on specific hosts within the NAS enclave front-ends (currently, PFEs and LFEs) *without* needing to use your RSA SecurID token at the time of the operation.

To accomplish this, you must first obtain special "SUP keys" using RSA SecurID authentication, which you can then use to perform operations from unattended jobs and/or scripts. Each SUP key is valid for a period of *one week* from the time it is generated. You may have multiple SUP keys at the same time, which will expire asynchronously.

SUP keys are currently allowed to call scp, sftp, qstat, rsync, shiftc, ssh-balance, and test.

Note: The shiftc command is built into the sup command; therefore, when you download the SUP client, you will also get shiftc.

SUP Usage Summary

The steps in this section demonstrate how to quickly get up and running with the SUP. Each step is explained in more detail in subsequent sections.

In these steps, *host* refers to a PFE or LFE; the command lines shown are examples.

1. Download and install the SUP client in your personal bin directory (one time).

   your_localhost% wget -O sup https://www.nas.nasa.gov/hecc/support/kb/file/9
   your_localhost% chmod 700 sup
   your_localhost% mv sup ~/bin

2. Authorize host for SUP operations (one time per host):

   pfe% touch ~/.meshrc

3. Authorize directories for write operations (one or more times per host):

   pfe% echo /nobackupp2 >> ~/.meshrc

4. Execute a command (each time)

   your_localhost% sup scp *foobar* pfe*21*:/nobackupp2/*username*/c_*foobar*

5. Examine expected output (as needed)
6. Troubleshoot problems (as needed)

SUP Client

The SUP client performs all the steps needed to execute commands through the SUP as if the SUP itself did not exist. Commands that are allowed to pass through the SUP can be executed as if the remote host were directly connected by simply prepending the client command sup, as shown in example 4, above. Besides executing remote commands, the client also includes an operating-system-independent virtual filesystem that allows files across all SUP-connected resources to be accessed using standard filesystem commands.

Requirements

The SUP requires OpenSSH version 5.4 (or later) and Perl version 5.8.5 (or later) and has been tested successfully on Linux, OS X, and Windows under Cygwin. These requirements are typically installed by default on Linux and OS X, but may need to be manually installed on Cygwin through the package manager as the openssh and perl packages. For general Cygwin installation instructions, see Installing Cygwin (PDF).

Installation

Complete the following steps to install the client:

1. Download the client and save it to a file called *sup*
2. Make the client executable by using chmod 700 sup
3. Move the client to a location in your $PATH

SSH Configuration

If your local username differs from your NAS username, it is recommended that you add the following to your~/.ssh/config file, where *nas_username* is replaced with your NAS username:

```
Host sup.nas.nasa.gov sup-key.nas.nasa.gov
    User nas_username
```

Note: If you are using a config file based on the NAS config template, you do not have to do this step.

Alternatively, the client's -u option can be used as described in the next section. If your local username is the same as your NAS username, no additional configuration or command-line options are required.

SUP Command-Line Options

-b
: Disable user interaction; for use within scripts. Note that the client will fail if any interaction is required--normally only needed when your SUP key has expired or is otherwise unavailable.

-u *username*
: Specify NAS username. Note that this option is required if your local username differs from your NAS username and you have not modified your SSH configuration appropriately.

-v
: Enable verbose output for debugging purposes.

SUP Authorizations

The basic set of operations that may be performed using the SUP is specified by the administrator. To protect accounts from malicious use of SUP keys, users must grant execute and write permissions to SUP operations on each target system.

Authorizing Execute Operations

By default, even SUP operations permitted by site policy are not allowed to execute on a given host. To enable SUP operations to a given host (PFE or LFE), the file ~/.meshrc must exist on that host, which can be created by invoking the following:

```
% touch ~/.meshrc
```

Note that the PFEs share their home filesystems, so this must only be done on one of these nodes. Other systems must be authorized separately. Once this file exists on a host, all operations permitted by site policy are allowed to execute on that host.

Authorizing Write Operations

By default, SUP operations are not allowed to write to the filesystem on a given host. To enable writes to a given directory on a given host, that directory must be added (on a separate line) to the ~/.meshrc file on that host. For example, the following lines in ~/.meshrc indicate that writes should be permitted to /nobackupp2 and your home directory.

```
/nobackupp2
/u/username
```

Each directory is the root of allowed writes, so this configuration would allow writes to all files and directories rooted at /nobackupp2 and your home directory (for example, /nobackupp2/*some/dir*).

Note that the root directory cannot be authorized. Also note that dot files (that is,~/.*) in your home directory are never writable regardless of the contents of ~/.meshrc.

Executing Commands Through SUP

Usage examples of each command that may be executed through the SUP are given below.

Note: SUP commands must be authorized for execution on each target host, and that transfers to a given host must beauthorized for writes. Before a given operation is performed, the client may ask for certain information, including the existing or new passphrase for ~/.ssh/id_rsa, the password + passcode for sup.nas.nasa.gov, and/or the password + passcode for sup-key.nas.nasa.gov.

For more detailed information on each command, see their corresponding **man pages**.

File Transfer Commands

rsync

your_localhost% sup rsync *foobar* pfe*21*:/nobackupp2/*username*/c_*foobar*

If you intend to transfer files to your home directory, note that even if your home directory has beenauthorized for writes, rsync transfers to your home directory will fail unless the --inplace option is specified. This is because rsync uses temporary files starting with "." during transfers, which cannot be written in your home directory. You can avoid this problem by specifying --inplace as shown in the following example:

your_localhost% sup rsync --inplace *foobar* pfe*21*:

scp

your_localhost% sup scp *foobar* pfe*21*:/nobackupp2/*username*/c_*foobar*

sftp

your_localhost% sup sftp pfe*21*

shiftc

your_localhost% sup shiftc *foobar* pfe*21*:/nobackupp2/*username*/c_*foobar*

For more information, see Using Shift for Remote Transfers and Tar Operations.

File Monitoring Command

test

your_localhost% sup ssh pfe*21* test -f /u/*username*/c_*foobar*

Job Monitoring Command

qstat

your_localhost% sup ssh pfe*21* qstat @pbspl1

ssh-balance and bbFTP

ssh-balance

your_localhost% sup ssh-balance -l pfe

Note that this command allows the load balancer among the PFEs to be used with bbftp:

your_localhost% sup bbftp -e "put *foobar* /u/*username*/*c_foobar*" `sup ssh-balance -l pfe`

SUP Expected Output

The following sequence shows the expected output for the command:

your_localhost% sup scp *foobar* pfe*21*:/nobackupp2/*username*/c_*foobar*

for a user who has never used the SUP before.

The conditions under which each sub-sequence will be seen are indicated next to each header. Most of the items will only be seen once or during key generation. A second invocation will only show the command output portion.

1. **Identity initialization** (seen once per identity)

   Initializing identity on sup-key.nas.nasa.gov (provide login information)
   (username@sup-key.nas.nasa.gov) Password:
   (username@sup-key.nas.nasa.gov) Enter PASSCODE:
   key SHA256:p3VqOcfV46fJwhOpYs/2h2sztRkNZ3nFu6VQbl2pgfI uploaded successfully

2. **SUP key generation** (seen when no valid SUP keys available)

   Generating key on sup.nas.nasa.gov (provide login information)
   (username@sup.nas.nasa.gov) Password:
   (username@sup.nas.nasa.gov) Enter PASSCODE:

3. **Client upgrade** (seen during key generation when new client available)

   A newer version of the client is available (8.04 vs. 8.1)
   ...do you wish to replace the current version? (y/n) **y**

4. **Command output** (always seen)

   foobar 100% 5 0.0KB/s 00:00

SUP Troubleshooting

The following error messages may be encountered during your SUP client usage. Note that the-v option can be given to the SUP client to output additional debugging information.

- "WARNING: Your password has expired"

  This message indicates that your current password has expired and must be changed. To change your password, you must log in to an LDAP host (for example, Lou) through an SFE and change your LDAP password. This change will be automatically propagated to the SUP within a few minutes.

- "Permission denied (~/.meshrc not found)"

  This message indicates that you have not created a .meshrc file in your home directory on the target host. SUP commands must be authorized for execution on each target host.

- "Permission denied (unauthorized command)"

  This message indicates that you have attempted an operation that is not currently authorized by the SUP. Check that the command line is valid and that the attempted command is one of the authorized commands. Certain options to authorized commands may also be disallowed, but these should never be needed in standard usage scenarios.

- "Permission denied during file access" (various forms)

  These messages indicates that you attempted to read or write a file for which such access is not allowed. The most common cause is forgetting to authorize directories for writes. Reads and writes of ~/.* are never permitted.

- "Permission denied (publickey)"

  This message indicates that you may not have proper permissions on your ~/.ssh and/or home directory on the target host. Check to make sure that ~/.ssh is not readable/writable by other users/groups and that your home directory is not writable by other users/groups.

Advanced SUP Use

Using the SUP Virtual File System

The SUP virtual file system (VFS) capability can be used to issue file-related commands to all SUP-connected hosts from the command line of your local host.

Accessing Files Across Multiple Hosts

The SUP client includes a virtual file system capability that allows files across all SUP-connected resources to be accessed using standard file system commands. For example, once you have activated VFS as described below, the command:

% ls pfe*21*:/tmp

from your local host would list the files in /tmp on pfe*21*. The command:

% cp *filename* pfe*21*:/tmp

from your local host would copy the file *filename* from your current directory on your local host to /tmp on pfe*21*.

The set of supported commands includes cat, cd, chgrp, chmod, chown, cmp, cp, df, diff, du, file, grep, head, less, ln, ls, mkdir, more, mv, pwd, rm, rmdir, tail, tee, test, touch, and wc.

Note that this functionality is not a true file system since only these commands are supported and only when used from within a shell. Unlike more general approaches such as FUSE, however, the SUP capability is completely portable and can be enabled with no additional privileges or software.

Commands through the VFS functionality can act on any combination of local and remote files, where remote files are prefixed with *hostname*:. For example, the command:

% cat pfe*21*:/tmp/*rfile* ~/*lfile*

would print the file *rfile* in /tmp on pfe*21* as well as the file *lfile* in the user's home directory on the local host to the terminal. Any number of hosts can be included in any command. For example, the command:

% diff lfe2:/tmp/*lfe_file* pfe*21*:/tmp/*pfe_file*

would show the differences between the file *lfe_file* in /tmp on lfe2 and the file *pfe_file* in /tmp on pfe*21*. The client determines if any remote access is needed based on the path(s) given. If not, it will execute the command locally as given as rapidly as possible. Fully local commands also support all options with the exception of options of the form -f *value* (that is, single-dash options that take values).

VFS Activation

Requirements

Currently, SUP VFS functionality is only supported for bash, but csh support is planned for the future. This functionality requires Perl version 5.8.5 (note that this is more recent than version 5.6.1 required by the basic client functionality). It also requires the standard Unix utilities cat, column, false, sort, and true; and has been tested successfully on Linux, OS X, and Windows underCygwin. Note that users of Windows under Cygwin may need to install the coreutils and util-linux packages to obtain these utilities.

Activation and Deactivation on Your Local Host

Complete these steps.

1. Install the SUP client if you have not already done so.
2. Activate VFS functionality in a bash shell.
    - For an interactive bash shell, run:

      ```
      eval `sup -s bash`
      ```

    - For a non-interactive bash shell, begin the script with the following:

      ```
      #!/bin/bash
      shopt -s expand_aliases
      eval `sup -s bash`
      ```

    The instructions will load aliases and functions that will intercept specific commands and replace them with commands that will perform the requested actions through the SUP client.

To deactivate VFS functionality in a bash shell or bash script at any time, run:

% eval `sup -r bash`

Command-Line Options

The behavior of the virtual file system can be modified using various options at the time it is activated.

-ocmd=opts
: Specify default options for a given command since the VFS functionality overrides any existing aliases for its supported set of commands.

-t *transport*
: Change the file transport from its sftp default to transport. Currently, the only additional transport available is bbftp. Note that using bbftp as the transport may slow down certain operations on small files as bbftp has higher startup overhead.

-u *user*
: Specify NAS user name. Note that this option is *required* if your local user name differs from your NAS user name.

For example, the following invocation activates the client virtual file system using bbftp as the transport mechanism, *nasuser* as the user and adds colorization of local file listings using the Linux ls --color=always option.

```
% eval `sup -s bash -t bbftp -u nasuser -ols=--color=always`
```

VFS Caveats

The VFS functionality is still somewhat experimental. In general, it works for the most common usage scenarios with some caveats. In particular:

- "Whole file" commands (that is, commands that must process the entire file), including cat, cmp, diff, grep, wc (and currently more/less due to implementation) retrieve files first before processing for efficiency. Thus, these commands should not be executed on very large files.
- There is a conflict between commands that take piped input and the custom globbing of the client, thus these commands have portions of globbing support disabled. These commands are grep, head, less, more, tail, tee, and wc. In these cases, globbing will work for absolute prefixes, but not relative. For example, grep *filename* pfe*21:*/tmp/* will work, but cd pfe*21:*/tmp; grep *filename* * will not.
- Redirection to/from remote files doesn't work. The same effect can be achieved using cat and tee (for example, grep localhost < pfe*21:*/etc/hosts > pfe*21:*/tmp/a could be done with cat pfe*21:*/etc/hosts | grep localhost | tee pfe*21:*/tmp/a > /dev/null). Redirection still works normally for local files.
- The first time a command is run involving a particular host, a SFTP connection is created to that host. When runningps, it may appear as if a zombie client process is running.

VFS Commands

Currently supported commands and their currently supported options are below. Unsupported options will simply be ignored except where noted. All commands are still subject to SUP authorizations, thus something that cannot be executed or written normally through the SUP cannot be executed or written through this functionality either.

- cat (no options)
- cd (no options)
  Note that when changing to remote directories, cd only changes $PWD, so to make changes visible the working directory (that is, \w in bash) must be in your prompt. For example, the following prompt:

  ```
  export PS1="\h[\w]> "
  ```

  would display the current host name followed by the current working directory.
- chgrp (no options)
  Groups may be specified either by number or by name. Names will be resolved on the remote host.
- chmod (no options)
  Modes must be specified numerically (for example, 0700). Symbolic modes, such as a+rX, are not currently supported.
- chown (no options)
  Users and groups may be specified either by number or by name. Names will be resolved on the remote host.
- cmp (all options)
- cp [-r]
  Note that copies between two remote hosts transfer files to the local host first since the SUP does not allow third party transfers. Thus, very large file transfers between remote systems should be achieved using an alternate approach.
- df [-i]
  Note that 1024-byte blocks are used.
- diff (all options)
- du [-a] [-b] [-s]
  Note that 1024-byte blocks are used.
- file (all options)
- grep (all options)
- head [-*number*]
  Note that head does not support the form -n *number*, so, for example, to display the first 5 lines of a file, use -5 and not -n 5.
- less (all options)
- ln [-s]
  Note that hard links are not supported. Links from remote files to local files (for example, ln -s pfe*21:*/*filename* /*filename*) will be dereferenced during certain operations (for example, cat /*filename* will cat pfe*21:*/*filename*).
- ls [-1] [-d] [-l]

For efficiency purposes, ls behaves slightly differently for remote commands than for local commands. In particular ls -l will not show links by default and will show what is actually linked instead of the link itself. Link details can be obtained using the -d option (for example, ls -ld *).

Also for efficiency, ls processes remote files before local files, so output ordering may be changed when remote and local files are interleaved on the ls command line. For example, ls /*file1* pfe*21*: /*file2* would show pfe*21*: first, then /*file1*, then /*file2*.

- mkdir (no options)
- more (all options)
- mv (no options)
- pwd (no options)
- rm [-r]
- rmdir (no options)
- tail [-*number*]
  Note that tail does not support the form -n *number*, so, for example, to display the last 5 lines of a file, use -5 and not -n 5.
- tee [-a]
- test [-b] [-c] [-d] [-e] [-f] [-g] [-h] [-k] [-L] [-p] [-r] [-s] [-S] [-u] [-w]
  Note that compound and string tests are not supported. Compound and string tests can be achieved using multiple test commands separated by shell compound operators. For example, instead of

  % test -f pfe*21*:/*filename* -a "*abc*" != "*123*"

  do

  % test -f pfe*21*:/*filename* && test "*abc*" != "*123*"

  Alternatively, the actual test command can be executed through the SUP:

  % sup ssh pfe*21* test -f /*filename* -a "*abc*" != "*123*"

- touch (no options)
- wc (all options)