Processing Sensitive Data on HPC Systems

Clemens Thölken

Contents

Requirements & Preparations

You will need access to MaRC3 from your terminal via `ssh`/`scp`.

Note: *Under MacOS search for `terminal` in Spotlight (`Command + Space`). In Windows you start a command line by searching `cmd` in your Start Menu.*

Creating a SSH Key

If you have not done so, create a new SSH key so you do not have to login to MaRC3 with your password each time:

```
pc$ ssh-keygen -t ed25519
```

and use the default suggestions.

Configure SSH

Configure MaRC3 as a shorthand with login credentials to access it quicker, if you have not done so.

```
pc$ nano .ssh/config
```

In Windows use notepad instead. **Note:** *Make sure that you save the file by selecting `File->Save as (Ctrl + Shift + s)`, change type to `All files (*.*)` and the file name to `config` **without** the `.txt` extension.*

```
pc$ notepad .ssh/config
```

Where you paste the following config with your HRZ user name instead of `USER`:

```
IdentityFile ~/.ssh/id_ed25519

Host marc3
  HostName marc3a.hrz.uni-marburg.de
  Port 223
  User USER
  Proxyjump jumphost # if you connect from outside the UMR network

Host jumphost
    Hostname ssh.staff.uni-marburg.de
    User USER
```

Copy the SSH Key to MaRC3

Add the SSH key to MaRC3 `authorized_keys` in UNIX:

```
pc$ ssh-copy-id -i .ssh/id_ed25519 marc3
```

or in Windows alternatively with:

```
pc$ type .ssh\id_ed25519.pub | ssh marc3 "cat >> .ssh/authorized_keys"
```

Preparing Encryption

Connect to MaRC3 in a seperate terminal:

```
pc$ ssh marc3
```

On the MaRC3 cluster any sensitive data must be kept in a folder with access permissions only for your user in your home. For this purpose create the folder `~/secure` in your `/home` directory with read and write permissions (`600`) only for your user:

```
hnode$ cd ~
hnode$ mkdir -p secure
hnode$ chmod 700 secure
hnode$ cd secure
```

Windows does not offer GPG encryption software by default. Install GPG4win via `winget`:

```
pc$ winget install GnuPG.Gpg4win
```

GPG4win offers `gpg` in the command line and a graphical user interface called `Kleopatra` where you can encrypt/decrypt files also.

Creating a Secret Key Passphrase File

On your local machine create a keyfile for your project (in this case called `example`):

```
pc$ gpg --gen-random 2 32 | base64 > example.key
```

Copy the key to your `secure` MaRC3 folder:

```
pc$ scp example.key marc3:~/secure/
```

On the MaRC3 headnode make sure that the key file is also only accessible by you:

```
hnode$ chmod 600 example.key
```

Encrypting Data Locally

For the purpose of this workshop we will create some test data into a file `input/patient.txt`:

```
pc$ mkdir -p input
pc$ printf 'header\ntest\nsensitive data\nother data\nend' > input/patient.txt
```

On your local machine compress your data (folder) into a single `zip` file and encrypt it with your secret key file:

```
pc$ zip -r -o input.zip input/
pc$ gpg -c --batch --passphrase-file example.key input.zip
pc$ scp input.zip.gpg marc3:~/secure/
```

Interactive HPC Processing

For the purpose of this workshop we showing how secure data processing is possible in an interactive session on a computation node.

On the head node request a node in interacive mode (outside the workshop you would request exlusive use of an entire node by adding the `--exclusive` parameter so that nobody else could possibly access your senstitive data while it is decrypted during processing):

```
hnode$ srun --nodes=1 --ntasks-per-node=1 --time=01:00:00 --pty bash -i
```

Wait until a node is available and you are logged into this node. Create a temporary directory where all potentially senstive unencrypted data (input and output) is stored during processing.

```
node$ TMPDIR=$(mktemp -d /dev/shm/tmp.XXXXXX)
noee$ trap "rm -rf $TMPDIR" EXIT
```

The `trap` command will delete the directory with any data in it when you exit your current session, effectively cleaning up after you.

Decrypt the senstive input data from your `~/secure` folder and unzip it to this temporary directory:

```
node$ cd $TMPDIR
node$ gpg --batch --passphrase-file ~/secure/example.key -o input.zip -d ~/secure/input.zip.
node$ unzip input.zip
```

Check that the data is there and decrypted:

```
node$ ls input
node$ cat input/patient.txt
```

Normally, you would now do all necessary computation on your input data in this tempory directory. Make sure that none of the processing steps leaks sensitive information (e.g. partial data in log file in your home directory by default), or delete these files manually in your future job script.

As an example processing step, extract only the lines containing the word `sensitive` and save the result in `processed.txt`:

```
node$ grep 'sensitive' input/patient.txt > processed.txt
```

After you are done with all of the data processing, compress either only the selective results or the entire temporary directory into a `zip` file and encrypt it to into your `~/secure` folder:

```
node$ zip -o output.zip -r .
node$ gpg -c --batch --passphrase-file ~/secure/example.key -o ~/secure/output.zip.gpg outpu
```

Exit the compute node and transfer the results to your local computer:

```
node$ exit
```

Decrypting the Results

```
pc$ scp marc3:~/secure/output.zip.gpg .
```

Decrypt, unzip and check the results:

```
pc$ gpg --batch --passphrase-file example.key -o output.zip -d output.zip.gpg
pc$ unzip output.zip
pc$ cat processed.txt
```