Security: How do I configure Bright to manage a Self Encrypted Drive (SED)?

This article explains Bright Cluster Manager can be configured to manage a regular node with a Self Encrypted Drive (SED), so that the regular node is able to boot via PXE and be provisioned.

WARNING: Please note that an update of certain Bright packages can remove some of the changes that were done manually here. If the package node-installer-nfsroot is updated/reinstalled on the head node, then step 1 of this article has to be done again.

This procedure was tested on Bright Cluster Manager 8.1 and 8.2 on CentOS 7. The drives in which we have tested it include Intel SSD DC P4610 and Samsung SSD 983 DCT M.2 1 (for the latter it is necessary to boot in legacy mode)

The requirements are:

- The SED must implement OPAL 2.0.
- The node must boot via PXE. That is, it won't have a local boot record or be booted locally. With this simplification taken into account, this means that for this procedure no PBA (Pre-boot authentication) is installed to the disk, as unlocking is always done by the node-installer.

To keep it simple, the article assumes that:

- The nodes with a SED are in category "sed-category"
- That they use the software image "sed-image"
- That the only SED disk is /dev/sda

Security: How do I configure Bright to manage a Self Encrypted Drive (SED)?

The procedure is the same whether /dev/sda is the disk with the root filesystem or not.

The steps given here should be taken as a suggestion. But, if the administrator has a good understanding of what the sedutil-cli commands do, then the logic of the initialize script can be customized according to the particular needs of the administrator.

In this procedure the password used to decrypt the MEK is stored in an internal key value store of CMDaemon (on the head node). The node-installer---a process that runs on the compute node during provisioning---communicates with the head node to retrieve this value via HTTPS.

.

The procedure is based mainly on the documentation of sedutil available at https://github.com/Drive-Trust-Alliance/sedutil/wiki , as well as other SED-related documentation and forums.

1 - Install the sedutil package on the node-installer environment

Run the following commands on the head node:

```
# yumdownloader sedutil

# rpm -ivh sedutil-1.15.1-1.el7.x86_64.rpm --root=/cm/node-installer
```

2 - Configure the libata.allow_tpm kernel parameter for the compute nodes

This is necessary to be able to run the sedutil-cli tool.

Run the following commands on the head node:

```
# cmsh
% softwareimage use sed-image
% append kernelparameters " libata.allow_tpm=1"
% commit
```

The space before libata is necessary.

3 - Store the password for the disks in the internal CMDaemon key value store

Run the following commands on the head node to store the value of the password, test, in a key called sed.password. This is just an example---the administrator may prefer to store the password for each disk/node in a different key.

```
# cmsh
% profile use node
% append services cmkeyvalue
% append tokens key_value_store_get_pairs_token
% commit
% keyvaluestore
```

```
% set sed.password "test"

% commit
```

4 - Configure the initialize script of the category

Define the initialize script for the category:

```
# cmsh

% category use sed-category

% set initializescript

…
```

An editor session opens up. Edit the initialize script to include the following lines:

```
#!/bin/bash

# Get the SED password from the CMDaemon key value store

URL="https://master:8081/json/"


MAC=$(cat /proc/cmdline | tr ' ' '\n' | grep BOOTIF | cut -d= -f2)

if [ "$(echo "$MAC" | tr -cd '-' | wc -c)" == "6" ]

  then

    MAC=${MAC:3}

fi
```

```
KEY="/certificates/$MAC/key"

CERT="/certificates/$MAC/cert"

key="sed.password"

request='{"service":"cmkeyvalue","call":"keyValueStoreGetPairs","args":[0,["'$key'"]],"minify":true}'

result=$(curl -sq --cert "$CERT" --key "$KEY" -k -X POST -d "$request" $URL | \

        python -c 'import json,sys;print json.load(sys.stdin)["result"][0]["value"]' 2>/dev/null)

if [ $? -ne 0 ]; then

  echo "Unable to retrieve SED password"

  exit 1

else

  PASSWORD=$result

fi



DEVICE=/dev/sda

if sedutil-cli --query ${DEVICE} | grep "LockingEnabled = N" > /dev/null

  then

    echo "Locking is disabled. Setting up OPAL configuration"

    sedutil-cli --initialsetup ${PASSWORD} ${DEVICE}

    sedutil-cli --enablelockingrange 0 ${PASSWORD} ${DEVICE}

  elif sedutil-cli --query ${DEVICE} | grep "Locked = Y" > /dev/null

    then
```

```
    echo "Disk ${DEVICE} is locked. Unlocking using sedutil-cli"

    sedutil-cli --setlockingrange 0 rw ${PASSWORD} ${DEVICE}

    sedutil-cli --setmbrdone on ${PASSWORD} ${DEVICE}

    partprobe ${DEVICE}

fi
```

When done with editing the initialize script, commit the changes:

```
% commit
```

5 - Applying the configuration to the nodes

The first time a compute node boots and runs the initialize script of the previous step, then, if locking is not enabled for the node, the initialize script runs the commands necessary to set up the initial configuration. The node then continues provisioning normally. The next time that the node is powered off, the disk will be locked.

6 - How to disable locking if it is not needed anymore (optional)

If the administrator wishes to disable locking of the disk, or wishes to remove the OPAL configuration completely, then there are some procedures described at https://github.com/Drive-Trust-Alliance/sedutil/wiki/Encrypting-your-drive

However, before attempting one of those procedures, the administrator should remove the initialize script that was configured earlier on in this article, otherwise locking will be reconfigured when the node boots. The removal can be carried out by assigning the compute node to a different category that doesn't have this initialize script configured, or by removing the initialize script from the category completely.

This removal guarantees that Bright will no longer attempt to modify the OPAL configuration of the disk. However the administrator still has to remove the configuration manually from the disk. This removal can be carried out by first configuring the node to boot in maintenance mode:

```
# cmsh

% device use node001

% set nextinstallmode main

% commit

% reboot
```

Maintenance mode runs in an environment identical to the one used by the node-installer. Once the node has booted in maintenance mode, the administrator can log in to the node via SSH and perform one of the following procedures:

Manual removal from disk procedure 1: As described at https://github.com/Drive-Trust-Alliance/sedutil/wiki/Encrypting-your-drive, the easiest way of doing it is just to disable locking, by running the following commands:

```
# sedutil-cli -–disableLockingRange 0 <password> <drive>
# sedutil-cli –-setMBREnable off <password> <drive>
```

Manual removal from disk procedure 2: The other method is removing the OPAL configuration. The note at https://github.com/Drive-Trust-Alliance/sedutil/wiki/Encrypting-your-drive should be read carefully, as for some particular disk models this procedure has been known to delete all data!

```
# sedutil-cli --revertnoerase <password> <drive>

# sedutil-cli --reverttper <password> <drive>
```

Unique solution ID: #1444

Security: How do I configure Bright to manage a Self Encrypted Drive (SED)?

Author: Carlos Pintado
Last update: 2019-07-29 12:07