Configuring: How do I install LUKS/NBDE onto Bright nodes?

How do I use LUKS + NBDE "Network-Bound Disk Encryption" to encrypt data on regular Bright nodes?

Introduction:

This FAQ describes how to use Network-Bound Disk Encryption (NBDE) to encrypt non-root volumes or partitions on regular Bright nodes. Support for root volumes or root partitions for regular Bright nodes is on the roadmap for after Bright 8.4

NDBE is available on RHEL/CentOS, starting from version 7.5. The procedure described here was tested on Bright 8.2 + CentOS 7.6.

Using LUKS on its own to encrypt block devices, partitions, LVM volumes, and so on, would be unwieldy on a large cluster because it requires manual intervention to enter the passphrase needed for LUKS, so that the encrypted volume can be decrypted.

NBDE deals with the problem by automatically handling decryption over the network, without the need for a manual intervention by the cluster administrator when the system is restarted.

Clevis, Tang, And Clevis Pin

Clevis and Tang are generic client and server components that provide network-bound encryption. In Red Hat Enterprise Linux 7.5+, they can be used to encrypt and decrypt root and non-root volumes of hard drives, to carry out NBDE. The description that follows is largely based on the Red Hat NBDE documentation at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Network-Bound_Disk_Encryption.html

Tang is a server to bind data to a network presence. It makes a system containing your data available when the system is bound to a certain secure network. Tang is stateless, and does not require TLS or authentication. Unlike escrow-based solutions, where the server stores all encryption keys and has knowledge of every key that has ever been used, Tang never sees a single client key. Tang thus never gains any identifying information from the client. This reduces the risk of information leakage.

Clevis is a pluggable framework for automated decryption. In NBDE, Clevis automates the unlocking of LUKS volumes. The clevis package provides the client side of the feature.

Clevis Pins are plug-ins into the Clevis framework. One of such Pins is a plug-in that implements interactions with the Tang NBDE server, and is unimaginatively called “Clevis Pin for Tang server.”

An aside about the software terms used: The terms clevis, clevis pin, and tang, originate from a kind of mechanical fastener (https://en.wikipedia.org/wiki/Clevis_fastener), but are just convenient labels for the software.

The Clevis Pin for Tang server gets a list of the Tang server's advertised asymmetric keys. Alternatively, since the keys are asymmetric, a list of the public keys for Tang can be distributed out-of-band, so that clients can operate without access to the Tang server. This mode of distribution is called offline provisioning.

The Clevis Pin for Tang uses one of the public keys to generate a unique, cryptographically-strong encryption key. Once the data are encrypted using this key, the encryption key is discarded. The client should store the state produced by this provisioning operation in a convenient location. This process of encrypting data is the provisioning step. The provisioning state for NBDE is stored in the LUKS header leveraging the luksmeta package.

When the client is ready to access its data, it loads the metadata produced in the provisioning step and it responds to recover the encryption key. This process is the recovery step.

In NBDE, Clevis binds a LUKS volume using a PIN so that it can be automatically unlocked. After successful completion of the binding process, the disk can be unlocked using the provided Dracut unlocker.

NOTE: this is a basic NBDE deployment. This doesn't include steps to setup HA for Tang server, rotating Tang keys, and other more advanced features.

For further details, the RedHat security guide can be consulted:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Network-Bound_Disk_Encryption.html

Configuring: How do I install LUKS/NBDE onto Bright nodes?

Installation Of Tang Server On The Head Node:

- Install Tang:

  [root@head ~]# yum -y install tang

- Modify the port of the tangd.socket, at

  /usr/lib/systemd/system/tangd.socket

  to be 8089. This means that

  ListenStream=80

  should be changed to:

  ListenStream=8089

  because otherwise it could be in conflict with the cmd port. The service can then be enabled and started with:

  [root@head ~]# systemctl enable tangd.socket

  [root@head ~]# systemctl start tangd.socket

- The Tang service should be added in cmsh
  [root@head ~]# cmsh
  % device use master
  % [head->device[head]]% services
  % [head->device[head]->services]% add tangd.socket
  % [head->device*[head*]->services*[tangd.socket*]]% set monitored yes
  % [head->device*[head*]->services*[tangd.socket*]]% set autostart yes
  % [head->device*[head*]->services*[tangd.socket*]]% commit

Installation Of Clevis client On The Software Image:

On the software image:

- Install clevis and cryptsetup:
  [root@head ~]# yum install clevis clevis-luks clevis-dracut --installroot=/cm/images/default-image/

- Create the mount point where the decrypted disk/volume will be mounted
  [root@head /]# mkdir /secret

- Reboot the compute node(s)

- Enable clevis-luks-askpass.path in the software image. It is needed early during boot before cmd starts:
  [root@head ~]# chroot /cm/images/default-image/
  [root@head /]# systemctl enable clevis-luks-askpass.path

Configuring: How do I install LUKS/NBDE onto Bright nodes?

- Update /etc/crypttab

Note that spaces and tabs should be maintained, otherwise cryptsetup will not identify _netdev option during boot.

Note: this test was done using a Bright COD cluster, which is a virtual cluster. Hence the vdc block device. The device /dev/vdc1 must be replaced with the correct block device name:

```
[root@head /]# echo secret  /dev/vdc1 none _netdev >> /etc/crypttab
```

- Re-create initramfs for the software image in cmsh using createramdisk:

```
[root@head ~]# cmsh
[head]% softwareimage use default-image
[head->softwareimage[default-image]]% createramdisk
```

Configuration Of The Compute Node(s):

On the compute node:

- Add the disk that will hold the encrypted volume. Here it will be /dev/vdc
- Using fdisk or other preferred tool, create the partition/volume that will be encrypted
- Create the encrypted LUKS volume with a passphrase
  [root@node001 ~]# cryptsetup luksFormat /dev/vdc1
- Open the LUKS volume
  [root@node001 ~]# cryptsetup luksOpen /dev/vdc1 secret
  [root@node001 ~]# ls /dev/mapper/secret
  /dev/mapper/secret
- Create a filesystem, e.g. ext4
  [root@node001 ~]# mkfs.ext4 /dev/mapper/secret
- Mount it
  [root@node001 ~]# mount /dev/mapper/secret /secret/
- Make sure you can reach Tang on the head node (replace the IP address with your head node's internal IP address)
  [root@node001 ~]# curl -f http://10.141.255.254:8089/adv/
- Associate our LUKS volume with the Tang server. It is better to use IP addresses instead of hostnames, to avoid any name resolution issues:
  [root@node001 ~]# clevis bind luks -d /dev/vdc1 tang '{"url":"http://10.141.255.254:8089"}'
- Add a new mount point, with the details shown next. Note that the _netdev option must be added:
  [root@head ~]# cmsh

```
[head]% device use node001
[head->device[node001]]% fsmounts
[head->device[node001]->fsmounts]% add /secret
[head->device*[node001*]->fsmounts*[/secret*]]% set filesystem ext4
[head->device*[node001*]->fsmounts*[/secret*]]% set device /dev/mapper/secret
[head->device*[node001*]->fsmounts*[/secret*]]% set mountoptions _netdev
[head->device*[node001*]->fsmounts*[/secret*]]% commit
[head->device[node001]->fsmounts[/secret]]% show

Parameter                      Value
------------------------------ --------------------------
Device                         /dev/mapper/secret
Dump                           no
Filesystem                     ext4
Filesystem Check               NONE
Mount options                  _netdev
Mountpoint                     /secret
RDMA                           no
Revision

[head->device[node001]->fsmounts]%
```

- Reboot the compute node
- Use tail -f or grep on /var/log/messages on the head node to check that tangd is logging the requests:
  [root@head ~]# grep tang /var/log/messages
  [...]
  Feb 26 14:29:53 head tangd: 10.141.0.1 GET /adv => 200 (src/tangd.c:85)
  Feb 26 15:29:50 head tangd: 10.141.0.1 GET /adv/ => 200 (src/tangd.c:85)
  Feb 26 16:05:56 head tangd: 10.141.0.1 POST /rec/dPAw2nwUthXYe57xo_tBs6QlytI => 200 (src/tangd.c:168)

- Check the compute nodes. If all is well, then the encrypted volume was unlocked during boot and mounted on /secret
  [root@node001 ~]# mount|grep secret
  /dev/mapper/secret on /secret type ext4 (rw,relatime,data=ordered,_netdev)

Unique solution ID: #1460
Author: Frank Furter
Last update: 2019-04-26 16:48