# LUKS with TPM in Ubuntu

This guide shows how to create a LUKS encrypted volume that uses TPM for key storage in Ubuntu. This guide is only for non-bootable volumes. Bootable volumes require some extra steps that are not mentioned in this guide.

Glen Tomkowiak · Follow

6 min read · Apr 8, 2016



## 1. Download the packages

**trousers** and **tpm-tools** provide the drivers and tools to work with a TPM under Linux. **pwgen** is a useful random password creation tool, you can substitute it with something else if it works for you. **cryptsetup** will allow you to create encrypted volumes. **sudo -i** makes you root so you can follow the steps with having to prefix every command as sudo.

```
sudo -i
apt-get install -y trousers tpm-tools pwgen cryptsetup
```

## 2. Partition your new disk

For all the examples I will use **/dev/sdb**. Be sure to use the correct disk when creating your volume to prevent data loss. The command below can help you find out what disks you have and what they are doing.

```
lsblk
```

### Optional hdparm security

*Be sure to read about the* **— user-master** *feature if you wish to try this:* http://man7.org/linux/man-pages/man8/hdparm.8.html

*Some hard drives have built in hardware based encryption and security-erase features that can be used in addition to LUKS encryption. You can check if your drive supports hardware encryption by running* `hdparm -I /dev/sdb`.

*Your drive should show not enabled with "supported: enhanced erase" security like in the example below.*

```
Security:
           Master password revision code = 65534
                   supported
           not     enabled
           not     locked
           not     frozen
           not     expired: security count
                   supported: enhanced erase
```

*You can set a user password for the hard drive with* `hdparm --user-master u --security-set-passwd "YOUR PASSWORD" /dev/sdb`

**DO NOT LOSE YOUR PASSWORD** *it is required to unlock the drive.*

*Then confirm that security is now enabled using* `hdparm -I /dev/sdb`. *See example below.*

```
Security:
        Master password revision code = 65534
                supported
                enabled
        not     locked
        not     frozen
        not     expired: security count
                supported: enhanced erase
        Security level high
        214min for SECURITY ERASE UNIT. 214min for ENHANCED
SECURITY ERASE UNIT.
```

I will not cover hdparm security in-depth but you can find more information here: http://www.admin-magazine.com/Archive/2014/19/Using-the-ATA-security-features-of-modern-hard-disks-and-SSDs

Use fdisk to partition your disk as shown below.

```
fdisk /dev/sdb
```

Choose **n** for a new partition, choose **p** for primary and then **w** to write it to the disk. You can safely accept the defaults in most scenarios.

### 3. Create your encrypted volume

Use the command below to create your encrypted disk.

```
cryptsetup luksFormat /dev/sdb1 -q --verify-passphrase
```

The passphrase will be needed to add the TPM key and it is good to have in case of a TPM failure.

Some versions of Ubuntu might require some extra steps: Disk Encryption Howto

### 4. Prep the TPM

An incompatible TPM might throw some errors, but most TPM chips will work. So far I have only had success with the following commands.

First ensure everything is running.

```
modprobe tpm_tis.ko
tcsd
```

Now clear the TPM to ensure it can be “owned”.

```
tpm_clear -z
```

You will now have to reboot and re-enable the TPM via the BIOS / UEFI after clearing it. Some machines might not let you clear their TPMs from Linux. This could happen because it is already owned and locked with a not well-known password. The workaround is to clear it from the BIOS. You can consult your vendor's documentation for specific instructions.

## 5. Own the TPM and store the key

The command **tpm_takeownership** takes ownership of the TPM with a default "well-known" TPM password. This avoids having to enter a TPM password. You could choose one if you require it, but then you will have to use it for unlocking.

Then you release the block of TPM space you need with **tpm_nvrelease**. If the TPM space is already free then you might get a harmless error. The 1 represents the TPM slot, you can choose another slot if 1 is occupied by something, just be sure to use the same slot in future commands, e.g. the key_script in step 7. Then you define a new slot with 32 bytes of storage using **tpm_nvdefine**. **pwgen** will generate a random password in memory. The device **/dev/shm/** is a RAM disk device that lets you read and write files in memory.

Then you write the key to the TPM with **tpm_write** and read it back with **tpm_nvread** to make sure the key can be retrieved once it is stored. You can open the file with `nano /dev/shm/tpm_temp.key` before and after writing / reading to ensure the key is exactly the same.

```
sudo -i
tpm_takeownership -y -z
tpm_nvrelease -i 1 -y
tpm_nvdefine -i 1 -s 32 -p "AUTHREAD|AUTHWRITE" -y -z
pwgen 32 1 > /dev/shm/tpm_temp.key
tpm_nvwrite -i 1 -s 32 -f /dev/shm/tpm_temp.key -z
tpm_nvread -i 1 -s 32 -f /dev/shm/tpm_temp.key -z
```

**Optional hdparm security** *Store your hdparm user password in another TPM slot (slot 2 in this example)*

```
tpm_nvrelease -i 2 -y
tpm_nvdefine -i 2 -s 32 -p "AUTHREAD|AUTHWRITE" -y -z
echo "Your hdparm user password" > /dev/shm/hdparm_temp.key
tpm_nvwrite -i 2 -s 32 -f /dev/shm/hdparm_temp.key -z
tpm_nvread -i 2 -s 32 -f /dev/shm/hdparm_temp.key -z
```

*View the password with* `cat /dev/shm/hdparm_temp.key` *to ensure it can be read correctly. Then safely delete it* `rm -f /dev/shm/hdparm_temp.key`.

## 6. Add the key to LUKS

Add the TPM key to the LUKS volume key slot. Then unlock the drive with your key to ensure it works. Leave the drive mapped and unlocked so you can format it in step 8. The passphrase you created in step 3 will be required to add the key. You can change **mapper_secure** to something else if you wish. Just be sure to use the same mapper name in your **crypttab** in step 7 (method 1). For the example I used key slot 2, but you are free to use another key slot if you want.

```
cryptsetup luksAddKey --key-slot 2 /dev/sdb1 /dev/shm/tpm_temp.key
cryptsetup luksOpen /dev/sdb1 mapper_secure --key-file /dev/shm
/tpm_temp.keyZero and remove the key file from the RAM disk.

dd if=/dev/zero of=/dev/shm/tpm_temp.key bs=1c count=32 >/dev/null
2>&1 rm -f /dev/shm/tpm_temp.key
```

Zero and remove the key file from the RAM disk.

```
dd if=/dev/zero of=/dev/shm/tpm_temp.key bs=1c count=32 >/dev/null
2>&1
rm -f /dev/shm/tpm_temp.key
```

## 7. Create a key script

Method 1 will work on Ubuntu 14 but Ubuntu 16 will require method 2 because systemd does not work with keyfiles currently.

**Optional hdparm security** *Add these line under* ***`service trousers start`*** *to unlock the hardware encryption before LUKS*

```
/usr/sbin/tpm_nvread -i 2 -s 32 -f /dev/shm/hdparm_temp.key -z
HDPARM_PASS=`cat /dev/shm/hdparm_temp.key` /sbin/hdparm --user-
master u --security-unlock $HDPARM_PASS /dev/sdb /sbin/partprobe
```

*This will unlock your hdparm encryption before LUKS using the password you stored in your other TPM slot.*

#### Method 1 Ubuntu 14

Open the nano text editor with `nano /root/key_script.sh` and add the following code, then save the script.

```
#!/usr/bin/env bash
# ensure TPM is setup and ready
modprobe tpm_tis.ko >/dev/null 2>&1
service trousers start >/dev/null 2>&1
# read key from TPM and copy to ram disk
tpm_nvread -i 1 -s 32 -f /dev/shm/tpm_temp.key -z >/dev/null 2>&1
# output key contents as if for crypt tab to read
cat /dev/shm/tpm_temp.key | tr -d "\n"
# zeros out the key in memory
dd if=/dev/zero of=/dev/shm/tpm_temp.key bs=1c count=32 >/dev/null
2>&1
# remove key from ram disk
rm -f /dev/shm/tpm_temp.key >/dev/null 2>&1
```

Ensure the script is executable and accessible by root.

```
chmod +x /root/key_script.sh
```

Edit `/etc/crypttab` and add the following

```
mapper_secure /dev/sdb1 /dev/shm/tpm_temp.key luks,keyscript=/root
/key_script.sh
```

Replace **/dev/sdb1** with your encrypted device. You can use the UUID for the drive by typing **blkid /dev/sd1** and using **UUID=** then the UUID output from **blkid** instead of the /dev/sdb1 device path.

Edit `/etc/fstab` and add the following line

```
/dev/mapper/mapper_secure /media/disk_secure ext4 defaults 0 2
```

Be sure to replace the mount point and device names with yours.

#### Method 2 Ubuntu 16 / systemd

Open the nano text editor with `nano /root/key_script.sh` and add the following code, then save the script.

```
#!/usr/bin/env bash
# ensure TPM service is ready
/etc/init.d/trousers start
# read key from TPM and copy to ram disk
/usr/sbin/tpm_nvread -i 1 -s 32 -f /dev/shm/tpm_temp.key -z
# decrypt drive
/sbin/cryptsetup luksOpen /dev/sdb1 mapper_secure --key-file
/dev/shm/tpm_temp.key
# mount drive
/bin/mount /dev/mapper/mapper_secure /media/disk_secure
# zeros out the key in memory
dd if=/dev/zero of=/dev/shm/tpm_temp.key bs=1c count=32
# remove key from ram disk
rm -f /dev/shm/tpm_temp.key
```

Ensure the script is executable and accessible by root.

```
chmod +x /root/key_script.sh
```

This script will load the key from the TPM to memory, decrypt the drive, and mount it.

Update crontab Run `crontab -e` as root then add the following line.

```
@reboot bash -x /root/key_script.sh > /root/tpm.log 2>&1
```

### 8. Format the volume

This step assumes you want ext4. You can technically chose any supported format you want. Just be sure to reference the correct format in your **fstab** in step 10.

```
mkfs.ext4 /dev/mapper/mapper_secure
```

### 9. Create a mount point

```
mkdir /media/disk_secure
chattr +i /media/disk_secure
```

**chattr +i** makes the mount point immutable. That way you can't accidently write data to this location if the encrypted drive is not mounted.

Then use the following command to mount your new volume without rebooting.

```
mount /dev/mapper/mapper_secure /media/disk_secure
```

You can test your volume by attempting to write to it. If it is unmounted the write will fail. Use the following command to test the drive.

```
touch /media/disk_secure/test_file
```

Linux Luks Tpm Encryption

Now you can reboot the system by typing `reboot`. Run `lsblk` after rebooting to verify **/media/disk_secure** is mounted. Then open your test file with `nano /media/disk_secure/test_file` to confirm everything is working.

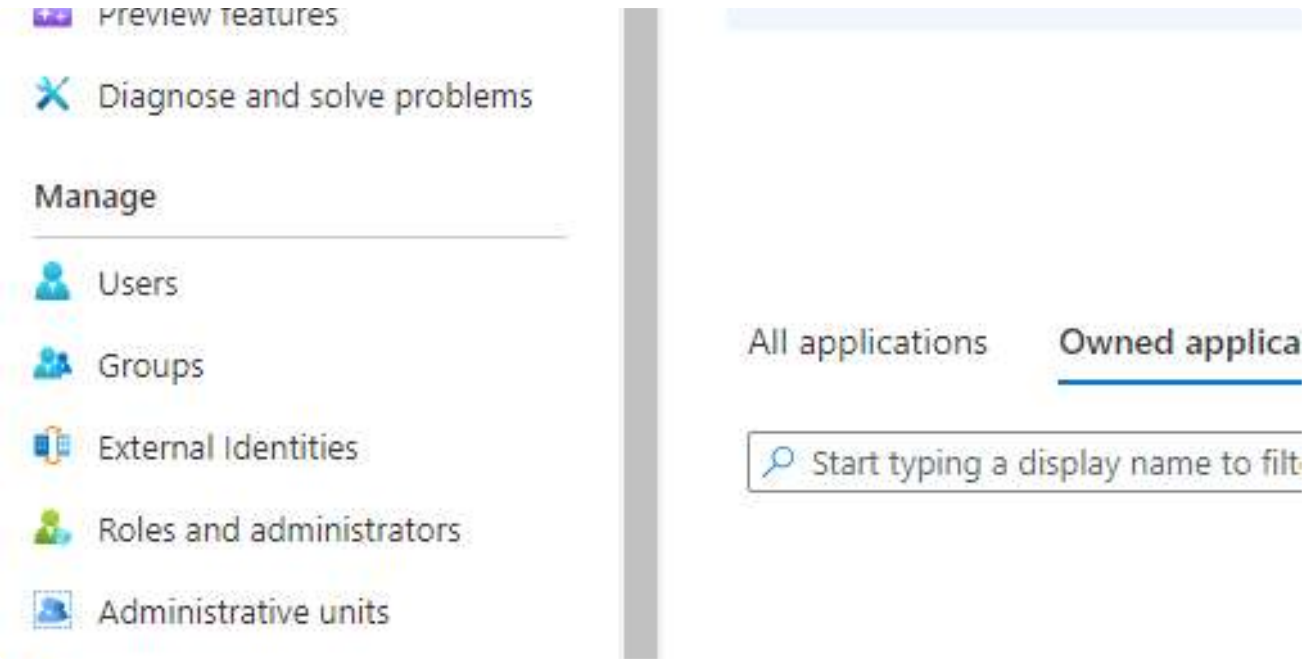
Follow

## Written by Glen Tomkowiak

8 Followers

Things that interest me: cloud computing, cyber security, DevOps, and mobile / web development.

### More from Glen Tomkowiak



Glen Tomkowiak in Towards AWS

## AWS Application Load Balancer with Azure AD oidc authentication

AWS Application Load Balancers can authenticate users with oidc. This is useful if you are using Azure AD and AWS within your organization.

3 min read · May 31, 2021

39 1

Glen Tomkowiak

## AWS SSM HTTPS/SSH Reverse Tunnel

The Amazon AWS Systems Manager Agent (SSM Agent) is a great way to manage systems in EC2 or on premises. It can run shell commands...

6 min read · Sep 22, 2018

1

Glen Tomkowiak

# ECR with on premises Kubernetes

Deploying Kubernetes on premises can be challenging enough.

2 min read · Sep 11, 2019

3 1

Glen Tomkowiak

## AWS EFS on Kubernetes with peered VPC

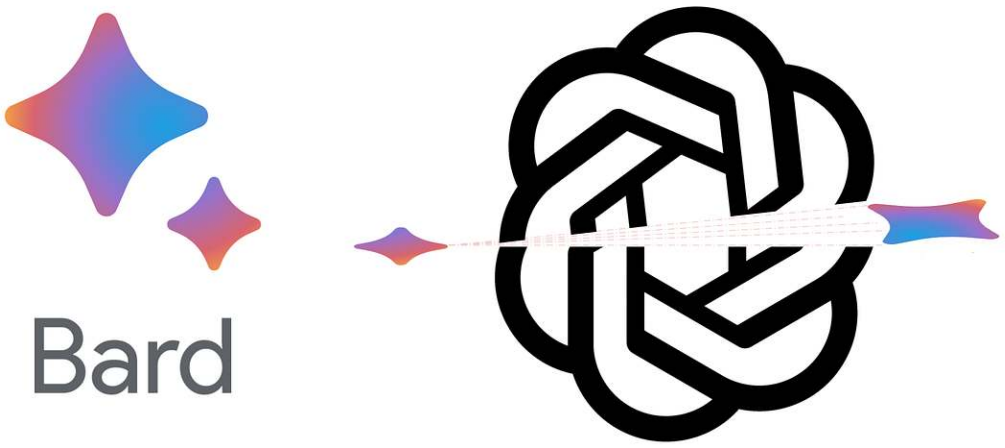
EFS storage “just works” with AWS EKS / Kubernetes. But there are a few quirks to be aware of when using a peered VPC.

3 min read · Apr 5, 2021

1

See all from Glen Tomkowiak

## Recommended from Medium




AL Anany

## The ChatGPT Hype Is Over — Now Watch How Google Will Kill ChatGPT.

It never happens instantly. The business game is longer than you know.

· 6 min read · Sep 1

7.8K 234


Nick Hilton

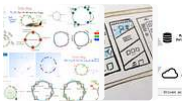
# The End of the Subscription Era is Coming

You’re overpaying for your porn (and journalism)

10 min read · Aug 30

10.6K 189

## Lists



**General Coding Knowledge**

20 stories · 351 saves

**Productivity**

230 stories · 103 saves

**Staff Picks**

447 stories · 283 saves

Unbecoming

## 10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my morning trail run. I'm wearing my stay-at-home mom...

· 4 min read · Feb 16, 2022

64K 937

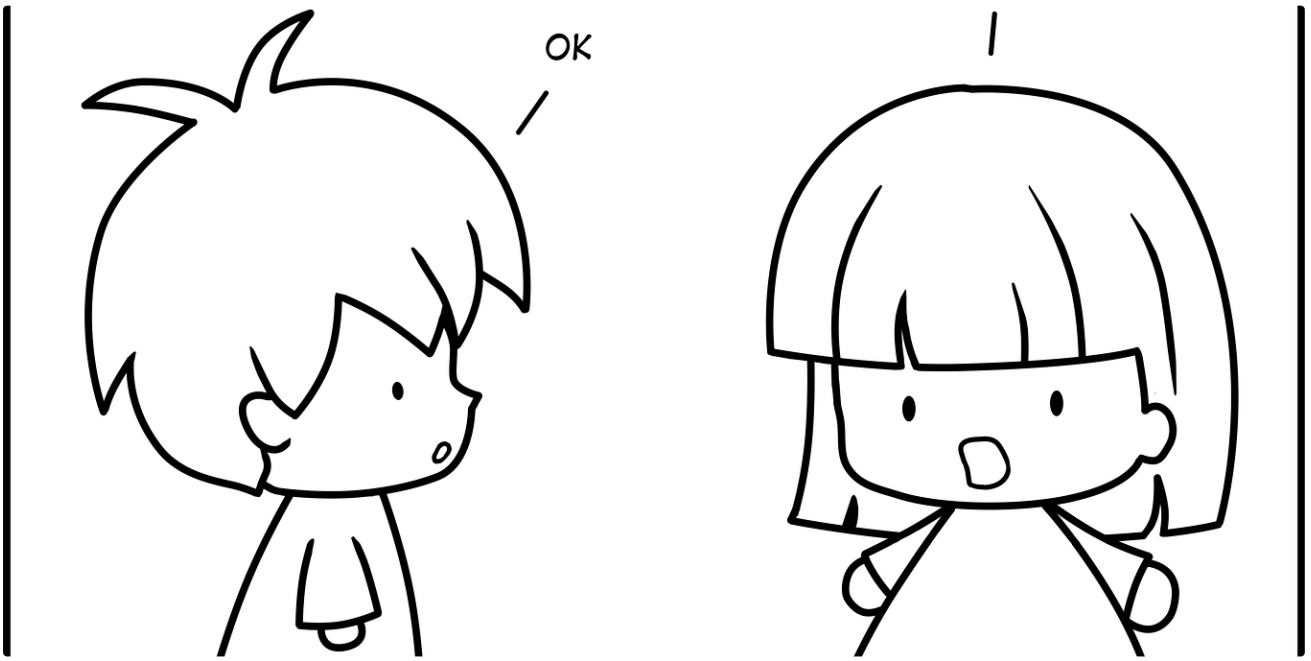
Benoit Ruiz in Better Programming

## Advice From a Software Engineer With 8 Years of Experience

Practical tips for those who want to advance in their careers

22 min read · Mar 20

8.5K 169




Julie Zhuo in The Year of the Looking Glass

## Average Manager vs. Great Manager

Explained in 10 sketches

2 min read · Aug 11, 2015

23K 210

Jerameel Delos Reyes

## Moving away from Raspberry Pi as my home server

I use a Raspberry Pi as my home server for a couple of years now. It was a cost-effective and energy-efficient way to host a variety of...

3 min read · Sep 1

135 2