How to install LUKS encrypted Ubuntu 18.04.x Server and enable remote unlocking

Hamy

Last updated on Sep 24, 2019 · 9 min read · 111 Comments · Tutorial

Much has been changed since my last post about LUKS remote unlock workaround (Particularly, The bug is finally fixed in cryptsetup 2:2.0.2-1ubuntu1.1 and no more workaround is needed). This, is the updated version on how to set things up properly.

UPDATE: Well, it turned out that while the previous bug is fixed, another one still exists. You can find the required workaround for it at the end of this article

In this post, I'm going to show you the required steps and downfalls on running a LUKS encrypted Ubuntu Server setup and how it can be extended to allow remote unlocking.

Prerequisites

1. A server to install on
2. Static public IP address
3. The so called *Alternative Ubuntu Server installer*[1]
4. Some patience 🙂

Installing and Setting up encrypted LVM

It is assumed that you already know your way around ISO files and how to boot them on your server.

We will also use the simplest possible setup: A server with a single disk

These steps would completely remove any leftover partitions and their associated data on the drive without the possibility to recover. Consider yourself warned!

We are going to use LVM inside the LUKS container, it is not only supported, but the recommended way as we could also make use of advanced LVM functionalities later on.

Follow the installation until you reach the disk partitioning section:

1. The disk might need to be unmounted first:

```
    <Go Back>                                                    <Yes>    <No>
```

2. choose *Guided – use entire disk and setup encrypted LVM* option:

```
                              [!!] Partition disks

The installer can guide you through partitioning a disk (using different standard
schemes) or, if you prefer, you can do it manually. With guided partitioning you will
still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk
should be used.

Partitioning method:

                  Guided - use entire disk
                  Guided - use entire disk and set up LVM
                  Guided - use entire disk and set up encrypted LVM
                  Manual

    <Go Back>
```

3. In the next window, take extreme care to select the right HDD in case you have multiple ones:

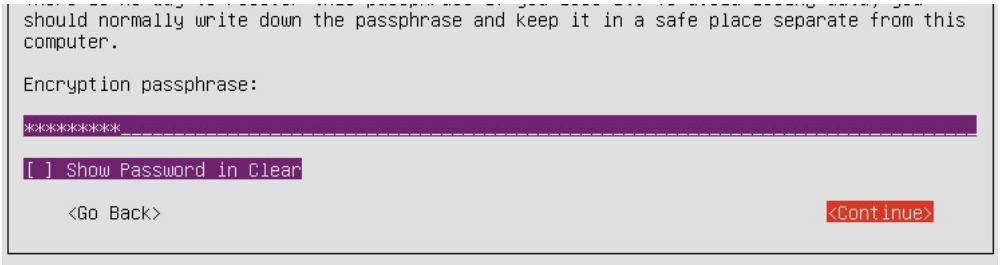
```
                              [!!] Partition disks

Note that all data on the disk you select will be erased, but not before you have
confirmed that you really want to make the changes.

Select disk to partition:

                  SCSI3 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK

    <Go Back>
```

4. Confirm changing the partition scheme to LVM if it was asked for:

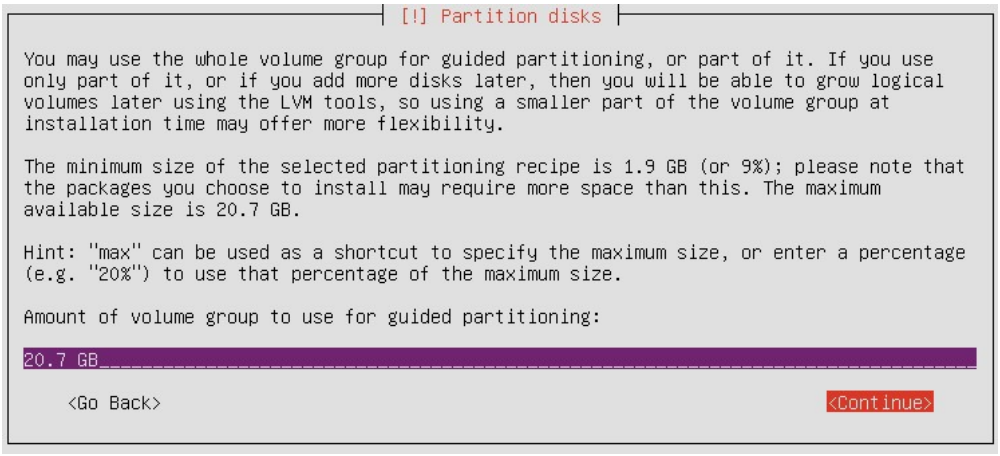
```
                              [!!] Partition disks

Before the Logical Volume Manager can be configured, the current partitioning scheme has
to be written to disk. These changes cannot be undone.

After the Logical Volume Manager is configured, no additional changes to the partitioning
scheme of disks containing physical volumes are allowed during the installation. Please
decide if you are satisfied with the current partitioning scheme before continuing.

The partition tables of the following devices are changed:
   SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
   partition #1 of SCSI3 (0,0,0) (sda) as ext4

Write the changes to disks and configure LVM?

    <Yes>                                                        <No>
```

5. Setup a strong passphrase for LUKS and confirm it:

```
                              [!!] Partition disks

You need to choose a passphrase to encrypt SCSI3 (0,0,0), partition #5 (sda).

The overall strength of the encryption depends strongly on this passphrase, so you should
take care to choose a passphrase that is not easy to guess. It should not be a word or
sentence found in dictionaries, or a phrase that could be easily associated with you.

A good passphrase will contain a mixture of letters, numbers and punctuation. Passphrases
are recommended to have a length of 20 or more characters.

There is no way to recover this passphrase if you lose it. To avoid losing data, you
```

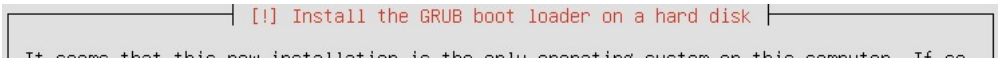


6. Decide on how much of the disk space you want to dedicate to the root partition. For advanced setups, you can use only a percentage of the available space and create more partitions later on (And all of them would be automatically encrypted). The default is to make a single partition the size of the HDD:



7. Confirm the partition setup and continue with the installation:

8. Since There is no other OS on this system, it's safe to install GRUB boot loader:



```
It seems that this new installation is the only operating system on this computer. If so,
it should be safe to install the GRUB boot loader to the master boot record of your first
hard drive.

Warning: If the installer failed to detect another operating system that is present on
your computer, modifying the master boot record will make that operating system
temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

    <Go Back>                                                    <Yes>    <No>
```

9. You will be prompted for the LUKS password after reboot to unlock the disk:

```
Begin: Loading essential drivers ... [    3.836065] raid6: sse2x1   gen()  6547
MB/s
[    3.884137] raid6: sse2x1   xor()  4724 MB/s
[    3.932080] raid6: sse2x2   gen()  7979 MB/s
[    3.980078] raid6: sse2x2   xor()  5363 MB/s
[    4.028072] raid6: sse2x4   gen()  9089 MB/s
[    4.076129] raid6: sse2x4   xor()  6245 MB/s
[    4.076698] raid6: using algorithm sse2x4 gen() 9089 MB/s
[    4.077284] raid6: .... xor() 6245 MB/s, rmw enabled
[    4.077899] raid6: using ssse3x2 recovery algorithm
[    4.082204] xor: measuring software checksum speed
[    4.120074]    prefetch64-sse: 11835.000 MB/sec
[    4.160064]    generic_sse: 10647.000 MB/sec
[    4.160324] xor: using function: prefetch64-sse (11835.000 MB/sec)
[    4.162354] async_tx: api initialized (async)
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ...   WAR
NING: Failed to connect to lvmetad. Falling back to device scanning.
  Volume group "ubuntu-vg" not found
  Cannot process volume group ubuntu-vg
  WARNING: Failed to connect to lvmetad. Falling back to device scanning.
  Volume group "ubuntu-vg" not found
  Cannot process volume group ubuntu-vg
Please unlock disk sda5_crypt:
```

Our overall setup would be something like this:

```
NAME                    FSTYPE
sda
├─sda1                  ext4
├─sda2
└─sda5                  crypto_LUKS
  └─sda5_crypt          LVM2_member
    ├─ubuntu--vg-root   ext4
    └─ubuntu--vg-swap_1 swap
```

- `sda1` is our boot partition. It is NOT ENCRYPTED[2]
- `sda2` marks the start of the logical partitions
- `sda5` is our encrypted LUKS partition
- `sda5_crypt` is the virtual *crypt* partition after unlocking (which uses LVM)
- `ubuntu--vg-root` is our root partition
- `ubuntu--vg-swap_1` is the swap partition

Remote unlocking overview

The process behind this fairly simple. The kernel loads initramfs image, inside this image are the required files/modules/scripts for decrypting/mounting root.

Now if we could somehow run a SSH server in initramfs and make it accessible via network, one could remotely connect to it to unlock root partition.

As initramfs runs in memory, we are somewhat limited in the size and complexity of the running programs. This is the main reason why Dropbear is being used as the SSH server combined with

BusyBox to provide the shell and basic utilities.

> All provided steps require root access. So you might want to `sudo -i` to root before continuing.

Update packages

Make sure your system (specially *cryptsetup* package) is up-to-date:

```
apt-get update && apt-get --assume-yes upgrade
```

Install Dropbear package for initramfs

As i said earlier, Ubuntu uses special Dropbear package to provide SSH server functionalities in initramfs environment with all the required hooks and scripts. Install it by issuing:

```
apt-get --assume-yes install dropbear-initramfs
```

If you get this warning while installing it, just ignore it for now, we will fix it soon enough:

> dropbear: WARNING: Invalid authorized_keys file, remote unlocking of cryptroot via SSH won't work!

Dropbear SSH keys

When you install the package for the first time, it also generates `dss`,`rsa` and `ecdsa` host keys[3] placed in `/etc/dropbear-initramfs/`.

Although possible, It is not wise to share your real `OpenSSH` host keys with the `dropbear-initramfs` ones. This is because for the keys to be accessible by the SSH server, they must not be encrypted (The same also applies to the initramfs itself as the kernel needs to be able to load it).

This means that even on a fully encrypted root system, physical access would be enough to retrieve the `dropbear-initramfs` private keys (unless boot partition is also encrypted. That unfortunately however, would also render our remote unlocking approach useless)

The downside of using a different private key for the Dropbear server, is that it will likely result in the client getting a scary warning about the possibility of a man-in-the-middle attack. This is because the server keys would be different before/after unlocking the root partition. The simplest (and possibly the best) way to avoid this issue, is to run the Dropbear SSH instance on another port. We will cover this shortly.

Dropbear config file

Dropbear's config file for the special `dropbear-initramfs` package, are placed in `/etc/dropbear-initramfs/config`

Changing default port

For the reason discussed above, we're better off using a custom port to listen on. This also would have the advantage of reducing attacks on the server, as no firewall is running in initramfs environment.

To make it listen on port `4748`, edit the said config file and add the following line:

```
DROPBEAR_OPTIONS="-p 4748"
```

Alternativ method:

A user reported that for some reason, the above method did not work for him and he added the same line to `/etc/initramfs-tools/initramfs.conf` instead.

Please note that this would result in these messages when rebuilding initramfs later on:

> dropbear: WARNING: Setting DROPBEAR_* or PKGOPTION_dropbear_* in /etc/initramfs-tools/initramfs.conf is deprecated and will be ignored in a future release dropbear: WARNING: Use /etc/dropbear-initramfs/config instead

Further adjustments

I have also added `-s -j -k -I 60` to `DROPBEAR_OPTIONS` just for the peace of mind. See `man dropbear` for details.

SSH Authentication

Password login has been disabled for `dropbear-initramfs` and only publickey authentication is allowed. Public keys should be placed in `/etc/dropbear-initramfs/authorized_keys`, one entry at a line. rsa based authentication is advised over ecdsa and dss.

You can also limit shell access to unlocking encrypted root partition only, by adding further per-user limitations in `authorized_keys` file like this:
`no-port-forwarding,no-agent-forwarding,no-x11-forwarding,command="/bin/cryptroot-unlock" ssh-rsa ...`

> In case it is not clear, you need to import your client public key to `/etc/dropbear-initramfs/authorized_keys`
>
> For rsa, it's default location on the client is `~/.ssh/id_rsa.pub`
>
> You might need to issue `ssh-keygen` on the client first if this hasn't been done already.

After changing Dropbear's settings, do not forget to regenerate initramfs:
`update-initramfs -u`
This time, no error should appear.

Kernel IP parameters

> Make sure the "`.yaml`" file(s) in your "`/etc/netplan/`" folder, is not named after your physical NIC's (e.g., if your NIC is "*ens5*", you should not use a file named "*ens5.yaml*" as your netplan configuration), that's just asking for trouble. By default this file is called something like "*01-netcfg.yaml*", which is fine.
>
> Thanks to the user Achim for finding that out.

Connecting remotely to the SSH server, would require the kernel to be able to setup network interfaces properly.

This would require that the kernel to first recognize the network interface (which is usually the case[4]), and also be able to setup IP parameters correctly.

Setting up static IP

The default kernel's behavior is getting the IP address via dhcp (`ip=dhcp`). If your network lacks a DHCP server, special kernel boot `IP` parameter is needed. This would usually be in the form of:
`ip=<client-ip>::<gw-ip>:<netmask>`[5]

Append that to the `GRUB_CMDLINE_LINUX_DEFAULT` parameter in `/etc/default/grub` and regenerate GRUB config file:

`update-grub`

Alternativ method:

A user reported that for some reason, the above method did not work for him and instead he added this line to `/etc/initramfs-tools/initramfs.conf`:
`IP="<client-ip>::<gw-ip>:<netmask>::eth0:off"`

The above line, uses the same structure for most part. Except that its location is different and it uses the capital `IP` variable.

And then of course rebuild initramfs: `update-initramfs -u`

Fixing dorpbear-initramfs

This is a long story, but basically, there is an inconsistency between the *dropbear-initramfs* and *busybox-initramfs* packages.

> While this problem has been fixed in recent versions of the *dropbear-initramfs* package, it still exists in *Ubuntu 18.04.x* .

After a complete setup, this problem manifests itself in these forms:

- Remote SSH sessions might not get closed automatically after a successful remote LUKS unlocking.
- You will get a couple of `ps` errors in your terminal right after LUKS unlocking.
- You might get this scary message in your terminal after boot up:

  > Aiee, segfault! You should probably report this as a bug to the developer

- Your network interfaces *might* refuse to automatically come up after a successful boot.

PS: I have yet to confirm whether the last issue in this list, does in fact have something to do with this bug. So if you believe that after applying the workaround, it fixes your specific issue, please let me know in the comments section below.

To overcome this, I have written a small script to address the issue in the most non-aggressive way I could think of. More information on how to download and apply it, could be found here: **dropbear-init-fix**

You are advised to apply this workaround, but you may as well skip it and settle with the buggy version of the *dropbear-initramfs* package.

The result

At this point if you have set up everything correctly, after a restart and right after the kernel loads initramfs, network's IP settings would be applied. Dropbear would start shortly after, listening for new connections:

```
[    4.440455] xor: using function: prefetch64-sse (11946.000 MB/sec)
[    4.442579] async_tx: api initialized (async)
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ...   WAR
NING: Failed to connect to lvmetad. Falling back to device scanning.
IP-Config: enp0s3 hardware address 08:00:27:92:cf:b5 mtu 1500 DHCP RARP
  Volume group "ubuntu-vg" not found
  Cannot process volume group ubuntu-vg
[    4.545857] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[    4.548714] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
 RX
```

```
[    4.549747] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
  WARNING: Failed to connect to lvmetad. Falling back to device scanning.
  Volume group "ubuntu-vg" not found
  Cannot process volume group ubuntu-vg
Please unlock disk sda5_crypt: IP-Config: no response after 2 secs - giving up
IP-Config: enp0s3 hardware address 08:00:27:92:cf:b5 mtu 1500 DHCP RARP
IP-Config: enp0s3 guessed broadcast address 10.0.2.255
IP-Config: enp0s3 complete (dhcp from 10.0.2.2):
 address: 10.0.2.15        broadcast: 10.0.2.255       netmask: 255.255.255.0
 gateway: 10.0.2.2         dns0     : 10.0.2.3         dns1   : 0.0.0.0
 rootserver: 10.0.2.4 rootpath:
 filename   : Ubuntu Server 18.04.pxe
Begin: Starting dropbear ... _
```

And now we could connect to the Dropbear server remotely:

`ssh -o "HostKeyAlgorithms ssh-rsa" -p 4748 root@client-ip`

Which brings us to the BusyBox built-in shell:

> To unlock root partition, and maybe others like swap, run `cryptroot-unlock`
>
> BusyBox v1.27.2 (Ubuntu 1:1.27.2-2ubuntu3) built-in shell (ash) Enter 'help' for a list of built-in commands.
>
> #

`cryptroot-unlock` command is all that stands now between you and booting up your server!

1. The *live version* lacks the required steps necessary for setting up LVM and encryption. At the time of writing, the latest version is `18.04.3` and can be downloaded from This Link. You'd probably want to download the `ubuntu-18.04.3-server-amd64.iso`. ^
2. While GRUB2 now supports unlocking an encrypted boot partition, I do not believe that it could be setup to do so remotely. ^
3. Dropbear does not seem to be supporting `ed25519` ^
4. If not, the module must be included in initramfs. Refer to `/usr/share/doc/dropbear-initramfs/README.initramfs` for details ^
5. Take a look at nfsroot.txt ^

LUKS Linux Ubuntu Networking

Hamy

a sysadmin in the wind

Related

- Remote unlocking of LUKS-encrypted root in Ubuntu/Debian
- How to Fully Uninstall Kaspersky's NDIS Filter
- OpenDKIM-OpenDMARC and a Chrooted Postfix Instance
- OpenVPN TAP adapter MTU in Windows
- Optimizing OpenVPN Throughput

ALSO ON **HAMY.IO**

How to compile OpenWrt and still …

a year ago • 4 comments

In this post, I discuss why you would want to build OpenWrt yourself, and …

Setting up an encrypted SOCKS …

a year ago • 9 comments

In this post, we are going to combine the power of Dante and stunnel together, to …

How to hide (obfuscate) SSH …

2 years ago • 2 comments

In this post I'm going to give you a real example of obfs4proxy being used to …

How to hide (obfuscate) a

2 years ago • 18

In this post, I'm you the informa necessary to be

111 Comments hamy.io Disqus' Privacy Policy Login

Recommend 1 Tweet Share Sort by Best

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Muammar El Khatib • 2 years ago

Great. This worked like a charm :)

5 • Reply • Share ›

Brad Bondurant • 2 months ago • edited

Worked perfectly in Ubuntu 20.04!

One thing to note: if you have multiple encrypted disks configured to mount during boot (for my setup, it's an SSD with LVM on LUKS for the root partition, and an HDD with LVM on LUKS mounted at /data), then by default dropbear will close after you unlock just the root partition. For me, this resulted in the second disk (/data) remaining locked and the boot was stuck at a password prompt on the server console, and with dropbear no longer running I was unable to reconnect to unlock it.

The solution I found was a simple modification to my crypttab, adding the "initramfs" option to both disks. So, instead of:

```
dm_crypt-0 UUID=<uuid_0> none luks
dm_crypt-1 UUID=<uuid_1> none luks
```

it would be:

```
dm_crypt-0 UUID=<uuid_0> none luks,initramfs
dm_crypt-1 UUID=<uuid_1> none luks,initramfs
```

After making that change, I'm now prompted for both passwords when I run `cryptroot-unlock`, although from what I've read you may have to run the command again for each disk you want to unlock.

1 • Reply • Share ›

Hamy Mod → Brad Bondurant • 8 days ago

Hello Brad,

Thank you very much for your comment and workaround. I've been rather busy in the last few months but hopefully I'll create a dedicated post for Ubuntu 20.04 soon and your input will definitely become handy.

Thanks again

• Reply • Share ›

Privacy Policy