Lab 10.1: Disk Encryption

In this exercise, you will encrypt a partition on the disk in order to provide a measure of security in the event that the hard drive or laptop is stolen. Reviewing the **cryptsetup** documentation first would be a good idea (`man cryptsetup` and `cryptsetup --help`).

1. Create a new partition for the encrypted block device with **fdisk**. Make sure the kernel is aware of the new partition table. A reboot will do this but there are other methods.
2. Format the partition with **cryptsetup** using **LUKS** for the crypto layer.
3. Create the un-encrypted pass through device by opening the crypted block device, i.e., `secret-disk`.
4. Add an entry to /etc/crypttab so that the system prompts for the passphrase on reboot.
5. Format the filesystem as an **ext4** filesystem.
6. Create a mount point for the new filesystem, ie. /secret.
7. Add an entry to /etc/fstab so that the filesystem is mounted on boot.
8. Try and mount the encrypted filesystem.
9. Validate the entire configuration by rebooting.

Solution 10.1

1. `$ sudo fdisk /dev/sda`

   Create a new partition (in the below /dev/sda4 to be concrete) and then either issue:

   ```
   $ sudo partprobe -s
   ```

   to have the system re-read the modified partition table, or reboot (which is far safer).

   Note: If you can't use a real partition, use the technique in the previous chapter to use a loop device or image file for the same purpose.

2. `$ sudo cryptsetup luksFormat /dev/sda4`

3. `$ sudo cryptsetup luksOpen /dev/sda4 secret-disk`

4. Add the following to /etc/crypttab:

   ```
   secret-disk    /dev/sda4
   ```

5. `$ sudo mkfs -t ext4 /dev/mapper/secret-disk`

6. `$ sudo mkdir -p /secret`

7. Add the following to /etc/fstab:

   ```
   /dev/mapper/secret-disk    /secret    ext4    defaults    1 2
   ```

8. Mount just the one filesystem:

   ```
   $ sudo mount /secret
   ```

   or mount all filesystems mentioned in /etc/fstab:

   ```
   $ sudo mount -a
   ```

9. Reboot.