Lab Sheet — File-system Encryption

1 Overview

The learning objective of this coursework is for students to obtain working knowledge with different technologies for performing file-system encryption. After finishing the lab, students should be able to set-up and manage an encrypted file-system, USB stick or file, using the preferred technology chosen from the set of possible technologies.

2 Lab Environment

This lab is about practical experience of encrypting data on your own machine. Therefore, you should run these tasks on your own laptop, if necessary using a virtual machine to run a foreign operating system (eg. Linux inside Windows). Alternatively, if you don't have a laptop, you can run the tasks on the Linux lab and Windows lab machines. The task descriptions explain which system to use for each task.

2.1 Software

To check that you have all necessary software installed on your machine do the following tests.

Testing `cryptsetup`. In this lab, we will use `cryptsetup` commands and libraries, which are included in most recent Linux distributions. To test the command line shell do

```
   %  which cryptsetup
/sbin/cryptsetup
   % cryptsetup --version
cryptsetup 1.2.0
```

Testing `truecrypt`. `truecrypt` is a platform-independent package for encrypting whole (logical) disks or container files. It comes with a GUI that guides you through the steps needed to setup such a device. You will have to download the most recent version from http://www.truecrypt.org/downloads.

On a Linux system do the following to unpack and install the package:

```
  % tar xfz truecrypt-7.1a-linux-x64.tar.gz
  % sh truecrypt-7.1a-setup-x64
  # you need to interactively accept the Licence, and then select option (2)
  % cd /tmp
  % tar tfz truecrypt_7.1a_amd64.tar.gz
  # if you want to install it into a different place, just move the entire usr/ dir
  % export PATH=$PATH:/tmp/usr/bin
  # you should now have truecrypt in your path
  % which truecrypt
/tmp/usr/bin/truecrypt
```

To launch truecrypt just type

```
  %  truecrypt
```

From this point onward, follow the Beginner's Tutorial and do Task 2.

2.2 Setting up your USB stick

Use a USB memorystick to setup an encrypted partition or a container file (preferred). First make sure that the USB stick doesn't contain valuable data, or backup any data (it will be destroyed in the process).

If you want to encrypt an entire partition on your USB stick, you need to launch a partition manager, such as `fdisk` on Linux, delete the existing partitions and create a second partition (eg. with the name `/dev/sdd2`).

3 Lab Tasks

Tasks 1 and 2 aim to instill basic skills on using and managing encrypted devices.

Task 1 should be run on a Linux machine, eg. the Linux Lab machines. Task 2 can be run on either a Linux or Windows machine, but needs an installation of `TrueCrypt`. Task 3 should be run on a Windows machine, eg. in the Windows lab.

3.1 Task 1: LUKS-based file-system encryption

This task should exercise the usage of the LUKS interface for file-system encryption on Linux machines (any recent Linux distribution should do).

Look up the slides on http://www.macs.hw.ac.uk/ hwloidl/Courses/F21CN/OSsec.pdf and go to Slide 64, which describes the steps needed to encrypt a container file. Make sure that you have set up your USB stick as explained in Section 2.2.

Follow the steps on Slide 64ff, to turn eg. `/dev/sdd2` into an encrypted partition. Generate a file `secret.dat` and enter a one-line text that you want to keep secret.

For a more detailed discussion, including a Video, on basic LUKS usage see this web page.

Answer the following questions:

- Which cipher and which mode did you use to encrypt this partition?
- Which key size did you use?
- Can you open this encrypted partition on a Windows machine?

3.2 Task 2: Truecrypt file-system encryption

Download and install Truecrypt from http://www.truecrypt.org/ as discussed in Section 2.1.

Launch TrueCrypt like this: `truecrypt`.

Follow the instructions in the “Beginners Tutorial” of the TrueCrypt User's Guide to set up an encrypted partition on your USB stick. If you partitioned the stick as suggested in Task 1, you can use the first partition, eg. `/dev/sdd1`, without destroying the data in `/dev/sdd2`.

Generate a file `secret.dat` with the same one-line text as in Task 1.

Answer the following questions:

- Is it possible to open the container file on a Windows machine, using `truecrypt` there?
- How can you check that the two files are identical?

3.3 Task 3: Steganography

This task needs to be done on a Windows machine, eg. in the Windows Lab. This task is not directly connected to Task 1 or 2 and you can do it in your own time.

Steganography is the science of writing hidden message within unsuspicious information.

- Intuitively, steganography "hides messages in plain sight".
- A concrete example is to use invisible ink to add text to, e.g. a letter containing non-confidential information.
- In computer science, there are several *covert channels* that can be used to hide information.
- For example, meta-data of files might contain unused byte, which will not normally be displayed.
- Another example is to embedd information within a picture, by using the lowest bit in each RGB value for the hidden text.
- Although this seems to provide only a small amount of space for hidden data, in pictures or even movies it is enough to hide hidden text.

For a practical exercise, visit this web page with a steganography example. Follow the instructions on the web page, to retrieve text hidden inside the picture shown on that web page.

Answer the following questions:

- Have you been able to retrieve the text files?
- What is the contents of these text files?
- What is the amount of data that you can hide in an 800×480 pixel colour picture, using the lowest bit for each colour as described above?
- Can you identify an operating system data structure, amenable to hiding information?