Documentation - Bug #1867

Warn users about arbitrary execution of code with full disk encryption through fast physical access

2018-07-03 12:43 AM - GNUtoo

Status:	open	**% Done:**	0%
Priority:	bug		
Assignee:			
Category:			

Description

Users using full disk encryption without /boot in clear typically expects that it's harder to gain arbitrary execution of code inside the distribution that resides in it.

An attacker would then need to temper with the non-encrypted code that runs before or during the opening of the encrypted partition. For instance:

- If the user uses GRUB_ENABLE_CRYPTODISK=y the attacker would need to temper with the tiny GRUB code that is embedded on the internal disk.

However there are some cases where the attacker might need to reflash the boot software (BIOS, UEFI, etc):

- If the user uses an external USB key to boot and the internal computer storage is fully encrypted
- If users are using Libreboot or Coreboot with GRUB to open the encrypted partition with the internal storage fully encrypted
  This can be mitigated by adding seals on the laptop screws (such as with nail polish or glue with glider)

An other way for an attacker would be to try to temper with the storage device content and/or firmware: Authenticated encryption is pretty new in cryptsetup, and the commonly used encryption algorithms are not authenticated. So there may be ways to gain arbitrary execution of code either by injecting content by manipulating encryption parameters or by trying to implement some way to recover the key by using an oracle (as fsck may correct the corrupted data) but it's probably far from trivial to attempt any of that.

However there is an easier way with Parabola: if the attacker can guess the root= kernel parameter for instance root=/dev/laptop-rootfs, the attacker could stick an SD card with the same vg and lv.

I can reproduce it with:

- A thinkpad under Coreboot that has an SD card slot
- The same VG/LV than the rootfs on a SD card
- The encryption key being inside the initramfs

I'll try to gather more information on the conditions necessary to trigger that problem (I had the issue several weeks ago).

This probably affects Libreboot too as there is documentation about such setup there too.

History

#1 - 2018-07-03 06:34 PM - GNUtoo

So in practice the issue is that if there is more than one block device that satisfies the root= it could pick the wrong device
Possible solutions:

- tell the user to use the UUID for root= and to keep that UUID secret
- swiftgeek from #libreboot IRC channel had a better idea: (1) make the initramfs refuse to boot on non-encrypted rootfs and (2) teach users to use /dev/disk/by-path/ in cryptdevice. This way all the encrypted partition would come from the right devices, and an attacker could not try to make the initramfs decrypt an SD card for instance (which would have the same root= UUID).