Common Criteria®

Title: Full Disk Encryption Essential Security Requirements
Maintained by: CCDB Work Group for Software Full Disk Encryptor
Version: 0.2
Date of issue: *2014-May-1*
Supersedes: N/A

Status

The CCDB Working Group has been requested to develop an Essential Security Requirements (ESR) for Software Full Disk Encryptor products. Given feedback from developers going against the NIAP Software Full Disk Encryption PP and the Trusted Computing Group, it was determined that the scope should be expanded beyond software only technology.

Background and Purpose

This document describes the high-level security requirements for Full Disk Encryption. It provides a minimal, baseline set of requirements targeted at mitigating well defined and described threats. In addition to stating what properties the FDE will minimally exhibit, the ESR also expresses functionality that vendors could optionally consider as an extension, but goes beyond the expected baseline. Furthermore, the ESR identifies aspects that are definitely outside the desired scope so as to limit the final set of security functional requirements specified in the cPP, as well as the evaluation activities performed during the course of evaluation. The reason behind this scoping of the product's capabilities to be specified in the cPP is to ensure that objective and repeatable evaluation activities can be captured in the cPP while still delivering a cPP in a timely manner. Another factor is to ensure the security functionality prescribed is not well beyond the current best practices and is achievable by multiple developer products.

Full Disk Encryption applies to software and hardware solutions that encrypt data on a hard drive. Full Disk Encryption (FDE), also known as whole drive encryption, is the process of encrypting all the data on the hard drive, including the computer's OS, and permitting access to the data only after successful authentication to the FDE product. FDE products may leave a portion of the drive or firmware unencrypted for the initial bootable partition. This ESR allows software drive encryption products to leave of the portion of the drive unencrypted to accommodate a bootable partition (e.g. MBR, GPT, etc.) as long as it contains no user data.

The ESR expects products to employ approved cryptographic algorithms to protect data on the drive from unauthorized disclosures. The ESR allows the product to invoke cryptographic functionality in the Operational Environment (OE) (for example, using a cryptographic library resident on the platform on which the product runs), or this cryptographic functionality can be implemented by the product itself. Therefore, the ESR are divided into two sections; those requirements that the FDE must implement and those that either the product or the operational environment can implement. Any functionality the product is relying on the OE for will have to be validated against the requirements. Since the ESR covers multiple implementations involving hardware and software, certain requirements and Assurance Activities may only apply to certain implementations.

The following discussion of terminology will aid in understanding the ESR text.

The hard disk is encrypted using a data encryption key (DEK). The DEK is masked using one or more key encryption keys (KEKs). The KEK can be derived from multiple components (referred to as submasks, which are derived from authorization factors) or obtained from a single submask. The foremost security objective of encrypting the storage devices is to force an adversary to perform a cryptographic exhaust against a prohibitively large key space.

Authorization factors can include a variety of mechanisms, some of which are mandatory, and some of which are optional. Authorization factors can be combined as long as the strength of the individual authorization factors is not diminished by this operation.

Use Case(s)

- The product will fully encrypt drives such that if an adversary obtains the device in a powered-down state (whether through a complete power-off by the user, or by a defined set of power-saving modes, such as “hibernation”), they will not be able to recover the data, authorization factors, submasks, or keys without performing a complete cryptographic exhaust against the keyspace of the DEK..

Resources to be protected

- All data stored on the protected devices.
- All data used to access the data stored on the protected devices, such as authorization factors, keys, and key material..

Attacker access

- An attacker gains physical access to a powered-down disk (or a powered-down system containing the disk) and can attempt to physically and logically access the disk/system, including booting the system off of an external drive or installing the drive in a system under the attacker’s control.
- An attacker gains remote access to a locked drive (an encrypted drive for which no one has successfully presented authorization factors for access). They can either attempt to unlock the drive on-line, or remotely image the encrypted data from the drive to brute force attack it off-line.

Attacker Resources

- Arbitrary amount of time to examine the disk
- A copy of the product and the skills to reverse-engineer the code for the product, run it in a debugger, etc..
- Commercially and/or publically available software/knowledge/equipment (including electron microscopes and focused ion beam instruments). The tools available to attackers are expected to be sophisticated enough such that cryptographic will need to be used to protect the data, since direct examination of the physical disk is possible.

Boundary of Device

- The software/firmware/hardware, including any libraries that are directly linked in or supplied by the product as part of product installation form the logical boundary..
- It is allowable for the product to invoke certain functionality supplied by the platform on which it executes, or on external devices such as USB tokens. However, the extent to which this can be done and comply with the ESR is defined by the particular ESR requirement.

Essential Security Requirements

Requirements to be met by the SWFDE product:

- The product shall be able to generate the DEK, change an existing DEK, zeroize/cryptographic erase the DEK, and ensure the DEK is protected.
- The product shall be able to derive KEKs from authorization factors, appropriately conditioning any authorization factors so that they can be used as a KEK or combined to form a KEK(s)
- A user must be authorized by the product before being able to fully boot up the system on which the product resides, or before accessing any data on a protected device
  - Minimum strength (entropy of authorization factor) for authorizing the user access to the data contained within the device subject of approval by appropriate cryptographic approval authorities of endorsing nations.
  - The entropy of the authorization factor shall not be weakened by choices of algorithms or any conditioning that is used in the key derivation process.
  - Authorization factor shall not be persistently stored on the device.
  - The product shall determine that the authorization factors are valid prior to decrypting the protected disk, while ensuring that this process does not expose or reduce the effective strength of any key or key material
- The product shall zeroize/erase all authorization factors, plaintext secret, private cryptographic keys and cryptographic security parameters when no longer required
- The product shall provide a cryptographic means to validate the source of updates to the product
- The product shall wrap the DEK with one or more KEKs
- The product shall provide self-tests to ensure the functions it implements are operating correctly.
- The product shall provide a lock-out mechanism for failed authentication attempts.

Requirements to be met by a hardware solution:

- The product shall provide firmware integrity.

Requirements to be met by either the SWFDE product or the Operational Environment:

- The product or environment will perform low level symmetric encryption and decryption, digital signature verification, hashing, and random bit generation using appropriate entropy sources
- The product or environment will perform the actual bit manipulations to protect the DEK with the KEK (or a set of intermediate keys)

- The product or environment will be capable of disabling power-saving states that leaves power to volatile memory (e.g. S3/sleep mode)
- The product or environment shall provide audit functionality (e.g. such as authorization failure, etc.)

Assumptions

- The data is only guaranteed to be protected when the drive is in an unauthenticated or powered down state.
- When authorization factors are generated outside of the device, it is assumed that the strength and entropy is commensurate with the key size.
- The drive is not left in an unprotected power-saving state (e.g., S3/sleep)
-

Optional Extensions

- The product may generate authorization factors (e.g., passphrases, bit strings) to be used in protecting the disk.
- The product may provide protection in low-power modes that is compliant with the requirements listed in the ESR, and must ensure (in these cases) that authorization occurs when the device returns from the allowed mode(s)
- The product may support key recovery; however, the product must have the capability to disable key export.

Outside the Scope of Evaluation

- Protections that counter attacks that alter the underlying platform or the code of an instantiation of the product that count on the altered item being returned to a legitimate user (aka "Evil Maid Attack")
- Protections that counter attacks on the platform in operation (for instance, over a network to which the platform is connected)