Requirements for Tape Drive based Encryption

Greg Wheeless & Roger Cummings
SYMANTEC
greg_wheeless AT symantec DOT com
roger_cummings AT symantec DOT com

Topics

- Overview
- Approach
- Non-requirements
- Requirements
- Futures
- Questions

Overview

- What's needed is a SIMPLE way to tunnel keying material through SCSI to a tape drive
  - And we need it defined SOON
- Its OK if initial proposal only mitigates against threat of data privacy loss due to loss of media (i.e. the "tapes fell off the back of a truck" problem)
  - May well need to address wider threat mitigation later

Symantec Background

- We rely on Reservations & Persistent Reservations, and will always transfer keying material within a reservation or PR
  - But we don't think that tape drive should enforce this
- We'd like to recommend that a tape drive supporting encryption also support the "Only If Reserved" bit in the Device Configuration mode page
  - Will ensure apps get earliest detection of reservation loss that might allow key changes that we don't know about!
- We see no reason to express a preference for a particular key length, algorithm or mode
  - That's not up to us – but to the drive vendors & marketplace
  - Any solution should accommodate multiples – and make allowance for transfer of related information
- We generally don't get to explicitly control the contents of CDBs
  - We can access new mode pages, respond to new ASC/ASCQs and often generate complete new CDBs if the don't transfer data

Non requirements

- More than 1 key active at a time in a tape drive
- Key caching across media changes and key directories on tape
- Reading back the active key itself
- Bits in read and write commands to control encryption
  - Because on most platforms apps cannot control bits in CDB
- Compliance with FIPS 140-2, Common Criteria profiles etc…….
  - But this needs to be addressed in the near future (see later)

Requirements

- No enforced relationship to tape format
  - If app wants to change keys in mid media, it can synchronize and do that (and better be able to handle reading correctly!)
- No change to logical block numbering on the media because of encryption
- Key implicitly deleted on media unload, loss of reservation, & on hard reset if there is no reservation
- Method to provide notification of key changes
  - Based on Generation code, or some form of key index
- Method to retrieve information written when the key was established to aid in key and algorithm identification 20 years later (probably will use attributes)
- Method to allow the app to cause the key in the tape drive to be explicitly deleted
  - May have to do an immediate (application level) verify to ensure key was delivered to the drive correctly in the 1st place

Requirements

- Method to allow app to determine the following with respect to the encryption support in the drive:
  - All methods supported
  - Subset of methods supported with the media currently loaded
  - Method to be used for the next block to be read or written
- Also need to define how this relates to density codes

Recommendations

- Implement the key transport using the Trusted In/Out commands and a Trusted Protocol field value of 07h
  - Assumes command definitions in 05-157r7 will be modified so that they conflict with reservations
- Assume future security protocols will use other Trusted Protocol values
  - i.e. that they will be separate from the definitions being created today
  - Command definition already has mechanism to list supported protocols
  - Define all related mode pages to support separate sub pages for each trusted protocol
- Above is best guarantee of future extensibility IMO

Futures

- Security compliance to FIPS 140-2, Common Criteria taken very seriously by some major storage consumers
  - Need to determine level of interest in this among T10 members and how it impacts us
    - But for a follow-on activity once this initial work is completed
  - May be that getting TCG to specifically address removable media is the best way to go, and will be sufficient

Questions

1. What’s our expectation when encrypted media is mounted in an existing drive in the field that knows nothing of encryption? Will it even recognize the media? It had better not leak any security information!
2. Can we assume going forwards that new tape technologies will recognize encrypted media and be able to report what it is, even if they cannot actually decrypt it?