Adding APFS Support to The Sleuthkit Framework

Presented by:
Joe T. Sylve,
Ph.D.
Director of R&D
Introduction
Overview

• We’ve got *pretty much* full support for APFS in TSK!
  • ... but I can’t give it to you just yet 😞
  • ... it will be released soon™

• We will be immediately releasing our pooled storage implementation
  • Will work with Brian, Jan-Niclas, Martin, et. al to convert their ZFS and BTRFS implementations and push them upstream
Supported Features

• Fully Parse APFS Containers (Pools)
• Fully Parse Filesystem Data/Metadata
• Full Support for Compressed and Sparse files
• Supports Decryption
  • Native APFS
  • Core Storage Upgraded
• Parse Snapshots
Work in Progress

• Support for Analysis of new iMac Pro / 2018 Macbook Pro
  • Comes with hardware T2 chip for encryption

• Support for Fusion Drives
  • Apple’s implementation of this hasn’t seem to stabilize yet
  • For now just image the logical container
Framework Changes
Pooled Storage Layer

• Sits between the VS and FS layers

extern const TSK_POOL_INFO *tsk_pool_open_*

extern void tsk_pool_close(const TSK_POOL_INFO *);

extern ssize_t tsk_pool_read(TSK_POOL_INFO *a_fs, TSK_OFF_T a_off, char *a_buf, size_t a_len);

extern TSK_FS_ATTR_RUN *tsk_pool_unallocated_runs(const TSK_POOL_INFO *);

extern TSK_POOL_TYPE_ENUM tsk_pool_type_toid(const TSK_TCHAR *str);

extern TSK_POOL_TYPE_ENUM tsk_pool_type_toid_utf8(const char *str);

extern void tsk_pool_type_print(FILE *hFile);

extern const char *tsk_pool_type_toname(TSK_POOL_TYPE_ENUM ptype);
File System Layer

• Pooled storage calls are optional
• Minor additions to the FS layer API

extern TSK_FS_INFO *tsk_fs_open_pool(const TSK_POOL_INFO *, TSK_DADDR_T, TSK_FS_TYPE_ENUM);

extern TSK_FS_INFO *tsk_fs_open_pool_decrypt(const TSK_POOL_INFO *, TSK_DADDR_T, TSK_FS_TYPE_ENUM, const char * password);
New Dependencies

• C++14
  • Implementation is in “modern” C++ with an exposed C API
  • Potential issues with pyTSK and VS 2008 for python 2.7

• OpenSSL
Future Work

• Java and Python bindings need to be updated
• Visual Studio Compilation
• Port the existing ZFS and BTRFS implementations to the pool storage layer
• Push everything upstream
DEMO TIME
Thank you for attending!

Questions?

Follow us on social for more webinars, blogs, product releases, tips and tricks and giveaways!