開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
⠕⠏⠑⠝
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

open

USE IMPROVE EVANGELIZE

Data at rest: ZFS & lofi crypto

Darren J Moffat

Senior Staff Engineer, Solaris Security

Topics

- Raw block device crypto – lofi(7D)
- ZFS terminology review
- ZFS Crypto

Lofi Encryption

- http://opensolaris.org/os/project/loficc
- lofi(7D)
  - File as a block device
  - Originally created for mounting ISO CD images
- Extend lofi(7D) and lofiadm(1M)
  - Specify crypto algorithm & provide key
  - Includes support for encrypted swap space
  - Targeting snv_87

Lofi issues

- Current implementation uses AES_CBC
- No integrity protection
  - Considering other AES modes to help

ZFS Terminology

- Pool
  - Collection of disks in a RAID layout
- Data set
  - File system or ZVOL
- ZVOL
  - Reserved part of a pool acting as block device
- COW
  - All of ZFS is Copy on Write
- All data & metadata checksumed/hashed

ZFS Crypto high level goals

- Support software only solution
- Support keys & crypto ops in hardware
- Support local (HSM, TPM, smart card, password)
  - or remote key manager
- Don't break COW semantics
- Support secure delete – by “key destruction”
- Need ability for delegation of key management to a Solaris Zone
- Need ability to keep data set keys away from a Solaris Zone

Decisions

- Set encryption policy at the ZFS data set
  - Most systems have only one pool
  - This allows zones/TX labels to have different keys and algorithms, eg AES-128 vs AES-256
- Will support encrypted zvol as well
  - Gives encrypted swap and raw database
- Ultimately support for encrypted root file system
  - /var/tmp could be a separate file system
  - /tmp is backed by swap

Decisions

- Data set encryption set at create time
  - Avoids encrypt later problem
  - Avoids old clear text due to COW
  - In future
- may have “scrub behind” - early discussions
- Rekey – deadline?
  - Rekey could take a VERY long time for a large pool/dataset and WILL hurt performance
- send & receive
  - In clear text only

The Crypto bit

- Integrity protection of data & metadata
  - Fletcher
  - SHA256
- Data and file system metadata confidentiality
  - AES 128,192,256 using CCM
- No direct use of asymmetric crypto in file system
  - Maybe used in future remote key manager protocols

What is encrypted ?

Yes

All “application” data

POSIX layer data

Permissions, owner etc

Directory structure

All ZVOL data

Snapshots

Clones

No

Pool metadata

Disks, mount time, raid, etc.

Deployment Issues

Data set names

Data set properties

Where do we store things ?

- Every dnode has compress/checksum/encrypt alg
- Never write unwrapped keys to disk
  - Issues with suspend/resume to disk

Delivery

- Phased delivery of key management
- Phase 1 targeting snv_92
  - Per file system keys encrypted with per pool key
  - Key management is per pool and/or per dataset
- Scope of later phases TBD

Status

- In development
- PSARC approval for phase 1 features
- http://opensolaris.org/os/project/zfs-crypto/
- zfs-crypto-discuss@opensolaris.org

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
⠕⠏⠑⠝
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Data at rest: ZFS & lofi crypto

Darren.Moffat@Sun.COM
http://blogs.sun.com/darren/

http://opensolaris.org/os/project/zfs-crypto/
http://opensolaris.org/os/project/loficc/