

# Integrity Measurement on NFS

[chucklever@gmail.com](mailto:chucklever@gmail.com)

# In Brief

- Backgrounder for draft-ietf-nfsv4-integrity-measurement
- New protocol mechanism fits WG charter:
  - “Useful when using NFS in large-scale virtualization environments”
  - “More effective NFS response to security challenges”

# What Is IMA?

- Integrity Measurement Architecture (IMA)
  - Transparent end-to-end integrity checking of file content and attributes
  - HMAC hashes stored separately, can be cryptographically signed

# What Is IMA Used For?

- Cryptographic verification of the provenance and integrity of executables
- On mobile devices: detect intrusion and replacement of applications
- In multi-tenant virtualization environments: detect tampering by other tenants or host
  - NFS is a good fit for guest software distribution

# How Does IMA Work?

- HMACs for executables are computed and signed by software distributors or hardware vendors, then stored in Linux extended attributes
- Signatures and HMACs are verified by a special kernel module before each execution (keys stored in a TCM)
- Signatures and HMACs are not exposed to applications via the POSIX API – main use case is executables
- Only a select few attributes are verified

# IMA on NFS

- NFSv4.2 extension has been proposed
  - Initial support only for immutable files
  - How to get HMACs to and from clients
  - How to identify which file attributes are verified
- Security considerations

# IMA on NFS: Open Issues

- Performance will be a concern
- Some verified attributes are Linux extended attributes that are not exposed by NFS
- Some NFS-related security-sensitive attributes are not verified (*e.g.*, NFSv4 ACLs)
- There is no published standard defining IMA

# IMA: Future

- Mutable files
- Protection of directories (file names)
- Additional extended attributes



# File Capabilities

- A capability set is the super-user privilege split up
  - E.g. CAP\_CHOWN, CAP\_FOWNER, CAP\_KILL
- Inherited by a process on fork(2)
  - Permitted / inheritable / effective
- Application can use capset(2) to adjust set
- File capabilities determine the capability set after an execve(2)

# File Capabilities, cont.

- Capabilities can be retained when process transitions from super-user to a non-privileged UID
- Capabilities can be bounded to prevent unwanted escalation
- File capability sets may be in effect only for certain user namespaces (Linux containers)

# Capabilities on NFS

- Would be another NFSv4.2 extension
  - How to get capsets to and from clients
  - Who enforces them
  - How to flatten/rehydrate binary representation
- Security considerations

# Capabilities: Open Issues

- Interoperability may be problematic
  - No standards: Withdrawn POSIX.1e draft, man capabilities(7)
  - Independent implementations of capabilities can have different semantics from each other
- How to represent user namespace information