XRootD 5.0.0
Encryption and Beyond

CHEP 2019
Adelaide, Australia
November 4 - 8, 2019

Andrew Hanushevsky, SLAC

http://xrootd.org
XRootD 5.0.0 The Next Big One

- Introduces many new features
  - Breaks plug-in ABI in some cases
    - Some external plug-ins will need to recompile
      - No source changes are needed
- It’s very ambitious & planned for 4Q19
  - Realistically, for all practical purposes, 1Q20
- This talk presents the highlights
  - And introduces what’s ahead
Transport Layer Security (TLS)

Why do it?
- Allow for authorization token handling
  - E.g. SciToken
- Improves security and data integrity

What are the obstacles?
- Backward compatibility
- Forward migration path
The **XRootD** approach?

- **Flexible TLS**
  - Not every client has TLS
    - We need to supply backward compatibility
  - Not everything needs TLS
    - We need to account for operational context
  - So, a connection may or may not require TLS
    - At the discretion of the client, or
    - The insistence of the server
Flexible TLS

- Initial connection is non-TLS
  - Client may request TLS
  - Server may request TLS
  - If either side requests TLS
    - The connection is converted to TLS

- The heart of flexible TLS is negotiation
  - Ability to go from non-TLS to TLS at any time
  - Provides backward compatibility & migration
    - Plus, no special ports are needed (but you can have one)
Flexible TLS Is Super-flexible

Client’s connections may be mixed

- Requests may use TLS but not data responses
  - Similar to what gridFTP does for data transfer
- TLS only when and where it’s needed

Request (i.e. login) connection
This may or may not be TLS
Additional data-only connections
What triggers TLS?

- Client URL that uses **roots** or **xroots**
  - `xrdcp xroots://server//mydata /tmp`
  - Implicit for authorization token usage

- Server configuration
  - TLS may be required for certain contexts
    - Third Party Copy
    - All TLS-capable clients
    - For all data
XRootD TLS Implementation

- Based on OpenSSL
  - All typically deployed versions are supported
    - Version 1.0.0 and above
      - Though should work with the old 0.9 series
    - Hostname verification added to cover all versions

- All TLS actions are logged
  - When a connection switches to TLS
  - What version of TLS the client is using
XRootD 5.0.0 has more than TLS

- Internal improvements & geeky features
  - Plug-in stacking
  - New general monitoring stream
  - Better containerization coexistence
  - Xcache improvements
  - See IN2P3 XRootD Workshop presentations
    - https://indico.cern.ch/event/727208

- And…
User settable file extended attrs

- Allows adding metadata to a file
  - Client can only play in the user namespace
    - System name space is fully hidden
- Done via binary API or xrdfs command
  - xrdcp extended to copy attributes as well (soon)
- Requires underlying file system support
  - Most file systems have it but not all
    - Some require special mount options
Beyond XRootD 5.0.0

5.0.0 lays the groundwork for...

- End-to-end data verification
  - On-the-fly disk \textit{and} network verification
- Server-side appends to a zip archive
- uid/gid tracking for files/directories
- Apply/Map operation for data pipelining
- RDMA support for better HPC integration
- Multi-protocol third party copy

Some of these will appear in 5.1.0!
In The End

5.0.0 significantly extends usability

- Important because **XRootD** is now embedded in many HEP data delivery system
  - EOS, DPM, CTA, dCache (Java version), QSERV, etc
  - New experiments are also relying on **XRootD**
    - E.g. Dune, LCLS II, LSST

5.0.0 addresses new and evolving needs

- Not only for HEP but other fields as well
  - Via it’s **Xcache** component