Building a Secure Data Environment

Maureen Dougherty
Center for High-Performance Computing
University of Southern California

Secure Data Environment

USC Center for High-Performance Computing (HPC) created an environment to provide access to compute resources in a seamless and transparent manner to USC researchers for the analytical work of their protected data (HIPAA, PHI, PII), without separating compute nodes from the HPC general resource pool.

- Must meet all Federal, State, and USC Office of Compliance regulations and requirements
- Must be easy to use – can not be too cumbersome or researchers will not use it
- Data must be encrypted at rest
- Data restricted to PI-authorized researchers only
- Auditing must inform of potential access violations, with a plan for addressing per regulations
- PI must still be accountable for following the secure data source's agreement
- No additional funds provided

Initial Configuration

After reading and interpreting the regulations and requirements from all parties, HPC leveraged open source tools, home grown code, and insight from the Office of the Chief Information Security Officer (OCISO), and licensed applications to build our HPC Secure Data Account (HSDA) environment.

- Docker container

  Silo researcher runtime environment on head node and compute nodes – integrated with PBS

- EncFS

  Used on existing file systems for home directories, project directories and staging directories. Launched within container startup

- Encryption key management administration system

  Home grown code to address encryption/decryption behind the scenes

- Audit log analysis tool
- Private head node
- Two-factor authentication

Initial Implementation

- Developed Docker container image for login node and compute nodes
  - Establish communications with file system and separate cluster VLANs
  - Interact with key management system to identify and mount EncFS for home directory, project directory and staging directory
  - Ensure functionality of GPUs
  - Tear down after jobs complete, log out, or time out on head node

- Scheduler hooks developed to launch Docker containers for HSDA jobs
  - Ability to identify HSDA job and launch compute node Docker container
  - Ensure multi-node compute jobs functional (Docker network)
  - Build /tmp or /scratch file system with EncFS depending job node count

- Deployed open source EncFS for encryption on existing file system structures
  - Interwoven with key management system and Docker containers
  - Enabled only for HSDA accounts and jobs

Initial Implementation (cont)

- Developed private encryption key management system
  - Behind-the-scenes process to address encryption/decryption with EncFS
  - Isolates home directory access to associated owner
  - Allows for sharing data between authorized members of HSDA for project and staging directories
- Audit logs gathered locally, consolidated and transferred live to security for filtered feed into QRadar (OCISO licensed application) for access authorization
- Private head node
  - Restricted access
  - Two-factor authentication
  - Standard idle timeout
- Implemented ssh+DUO integration
  - Leveraged OCISO licensed application

Configuration Updates

After initial deployment, continue to evolve environment to address upgrades to our cluster kernel, migration to new job scheduler/resource manager, security monitoring and security updates.

- New job scheduler/resource manager deployment
  - Required new HSDA job runtime workflow development
  - Extensive changes to job scheduler integration for HSDA job launching and teardown scripts
- Monitoring and updating of audit log configuration after kernel updates to ensure proper security workflow
- As part of kernel upgrade test environment, test HSDA workflow and auditing and make appropriate updates
- Address changes by regulatory groups (internal)
- Test everything, then test again

Lessons Learned

- Test and production environments must be as identical as possible and testing environment should include a sample of all hardware architectures in production
- Test all software applications to ensure functionality
- Scalability issues with Docker's console service with initial deployment
  - 200 compute node limit for all of HSDA while looking for alternative, open-source, console service
- Performance hit with EncFS instead of encryption on storage array/disk hardware
- Ensure all researchers are aware of limitations with secure data agreement, required as part of account request.
  - PI is still responsible managing data appropriately
- Docker integration into the framework of ssh, DUO two-factor authentication, and parallel mpi jobs (multi host networking) was challenging
- Integration with job scheduler requiring in-depth testing and re-testing
- DUO data transfer issues
  - Currently limited to scp

Questions