SOTERIA: Empowering Scientific Research with Secure Software Environments.

Farnaz Golnaraghi, University of Chicago, on behalf of

Brian Aydemir (Morgridge)
Cannon Lock (Morgridge)
Brian Bockelman (Morgridge)
Fengping Hu (UChicago)
Lincoln Bryant (UChicago)
Rob Gardner (UChicago)

Throughput Computing 23
Madison, WI
07/11/2023
The Rising Threat of Supply Chain Attacks

Supply chain attacks have been making headlines with increasing frequency.

Traditionally, they have been associated with sectors such as finance and industry.

Our scientific community is not immune to their repercussions.

The repercussions can be severe, leading to data breaches, intellectual property theft, and even wide distribution of compromised research results.

How do we bring industry advancements into the Open Science ecosystem?
Characteristics of Scientific Software

→ Scientific software exhibits some unique characteristics that differentiate it from other domains.
  ◆ **Disadvantage:** Insufficient software engineering training and lack of emphasis on software quality.
  ◆ **Disadvantage:** Minimal incentive for robust software functionality due to publication-centric credit systems.
  ◆ **Advantage:** Benefit from an established trust and reputation network through publication records.
  ◆ **Neutral:** Limited use of cloud infrastructure and OCI (‘Docker’) containers not yet dominant.

→ Despite these differences, scientific software has some similarities with other domains.
  ◆ Large dependency trees and reliance on common building blocks like Python and Conda.
  ◆ Growing adoption of containers and related "cloud-native" technologies.
Adoption of Containers:

➔ Containers streamline the execution of complex software.
  
  ◆ Simplified access through Docker Hub
    
    ● Caution is necessary regarding reputable sources and up-to-date images.
  
  ◆ Containers isolate the scientific software stack from the system software stack, enhancing portability.

➔ Container images offer immutability, but vulnerabilities can persist.
  
  ◆ Historical software bugs remain within immutable container images.
  
  ◆ Awareness and mitigation of security vulnerabilities are crucial to ensure robust containerized workflows
How does SOTERIA contributes to filling the gaps and why is it necessary?

- Registry for researchers and collaborations:
  - A service exclusively focused on meeting the needs of secure scientific software environment management.
- Traceability:
  - Where did the container come from? Was it signed by the user?
- Reproducibility:
  - Making the software more reusable and accessible
- Container visibility:
  - Provide information on the systems RPM, vulnerabilities, etc.
- Discoverability:
  - Make it easy to cite and discover images.
- Training and education on the importance of secure software environments
Architecture

- Exists today
  - Image creation
  - Image introspection
  - Image distribution

- Work For Tomorrow

Diagram elements:
- SOTERIA Container Registry
- HTTP Proxies
- Content Distribution (StashCache) (OSDF)
- Content Distribution (CVMFS)
- GitLab
- S3
- DB

Flow directions:
- Scientific Collaborations and Users
- CI Provider
- Users
- Image Push
- Image Transform
- Image Inspection
SOTERIA Container Registry

- We adapt and utilize open source tools to meet the scientific community’s needs
  - **OSG Hub** powered by **Harbor**
    - Authentication via **CILogon**.
      - Use federated identity, not tied to the project.
    - Operated on the **PATH Kubernetes platform**
    - Database has an on-site active-standby setup. Incremental snapshots are sent to alternate site and backed up offsite.
    - S3 bucket is similarly replicated.
    - Set to run at the Wisconsin and Chicago sites.

- **OSG Hub statistics:**
  - Project Count: 52
  - Repository Count: 430
  - Unique images: 28,261
  - Image pulls: ~78.4 million
  - Storage allocation: 3.2 TB
Identity and user management, project onboarding

- **COmanage** integrates with **CILogon** to provide a user registry for SOTERIA
  - Roles identify the degree of vetting a user has gone through
  - Group membership controls access to projects on OSG Hub
- The initial registration flow allows us
  - To link a OSG Hub user with their **ORCID iD**
  - Provision that user a private project so that they can explore the system
- Users can apply for **Researcher status**
  - Grants the ability to create **five projects** (three public, two private)
  - Grants the ability to add other SOTERIA users as members of those projects

**Key concept:**
Each image is tied to the researcher’s “social identity” (ORCID).
Vulnerability Scanning

- Vulnerability scanning is an obvious starting point for **image visibility goals**.
- Harbor comes with Trivy.
- We started with the Crowdstrike Falcon - readily-available tool that works with Harbor but isn’t a part of Harbor.
- Long-term vision:
  - Each artifact uploaded gets registered as a document in an ElasticSearch DB.
  - This triggers a suite of analysis tools that run against the artifact. Some security related and some visibility.
    - Can we enumerate and advertise all the software installed in a Conda environment?
  - Results of analysis tools are sent to the DB, advertised as part of the public webpage of the artifact.

Image Visibility
What Comes Next? Image Distribution

‘docker pull’ isn’t the end of the story. For each public image,

- Convert to singularity and upload to the OSDF for distribution.
  - Could even be done for non-public images.
- Convert to flat directory and publish into CVMFS.
  - Planning the successor to the singularity.opensciencegrid.org repo!

Beyond that, there’s a need for:

- **Auditing**: given an ‘interesting’ image, what was used where?
- **Selective replication**: Providing endpoints that only mirror images which are signed or without known critical-CVEs.
What comes next? Long-term

SOTERIA is a research project with a beginning, middle – and end.

- OSG Hub, as a part of the OSG Consortium, has a much longer lifetime.

There’s no universally-accepted way to capture, archiving, and assign a persistent identifier to a software environment.

- Minimally, we’d like to build tools to provide a smooth path to archive these containers to Zenodo.
  - Much, much to do in metadata - we’re not experts!

Bodleian Library, in Oxford, was founded in 1602. It is believed that 400 years is longer than the typical NSF project.
Thank you!!
Questions?

soteria.osg-htc.org

This work was supported in part by NSF Grant #2115148 CICI:UCSS:Securing an Open and Trustworthy Ecosystem for Research Infrastructure and Applications (SOTERIA)