Oak Ridge National Laboratory

HPC Cybersecurity Best Practices

And OLCF Security Awareness Information

Ryan Adamson

@weezel@hachyderm.io

@hackgoestheweez

github.com/rmadamson

* Supplemented with wisdom from the movie *Hackers*

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Roadmap

1. **HPC systems and their security model**
2. OLCF security ‘rules of thumb’
3. Software best practices
4. Interconnections and workflow security
5. OLCF computing policy refresher

What is Unique About HPC Security?

Differences between HPC and Industry

- Performance is justifiably important in HPC, but can be sacrificed in enterprise settings
- Clustered computation vs single-node workloads
- Industry standard tools that work well for enterprise do not map well to HPC
- Systems administrators of HPC systems are very knowledgeable
- Supercomputers @ OLCF are first-in-class and very prestigious

What these differences mean

- Balancing performance and security is more difficult than in enterprise
- Single OS security tools like SELinux can’t solve distributed system security problems
- Security benchmarking must be customized to fit needs of HPC systems
- Systems staff are a great resource for security engineers to partner with
- Vendors will work with you to solve security issues

Simply put, security teams want to ensure that **each activity on a system is ‘intended’** and can be **associated to a human** or team that vouches for the activity with some level of assurance.

Example Attack Scenarios

Credential Harvesting

- Attacker compromises a user's credential at an external facility
- That credential is used to gain access to our system
- Attacker lies in wait for privilege escalation opportunity
- Attacker establishes a persistent foothold
- Attacker harvests credentials of others
- Attacker carries out intended work

Remote Exploitation

- Attacker finds a vulnerable externally facing service
- Service is compromised, attacker can run commands in the context of the service
- Attacker leverages service to 'pivot' into internal systems
- A persistent foothold is established
- Attacker carries out intended work

Supply Chain

- Attacker inserts code into 'software supply chain' such as a piece of research software or open source library
- GitLab runner or other automated workflow tool executes code automatically
- A shell is shoveled to attacker and a foothold is established
- Attacker carries out intended work

What could an attacker's 'intended work' be?

Typical Attacker Motivations

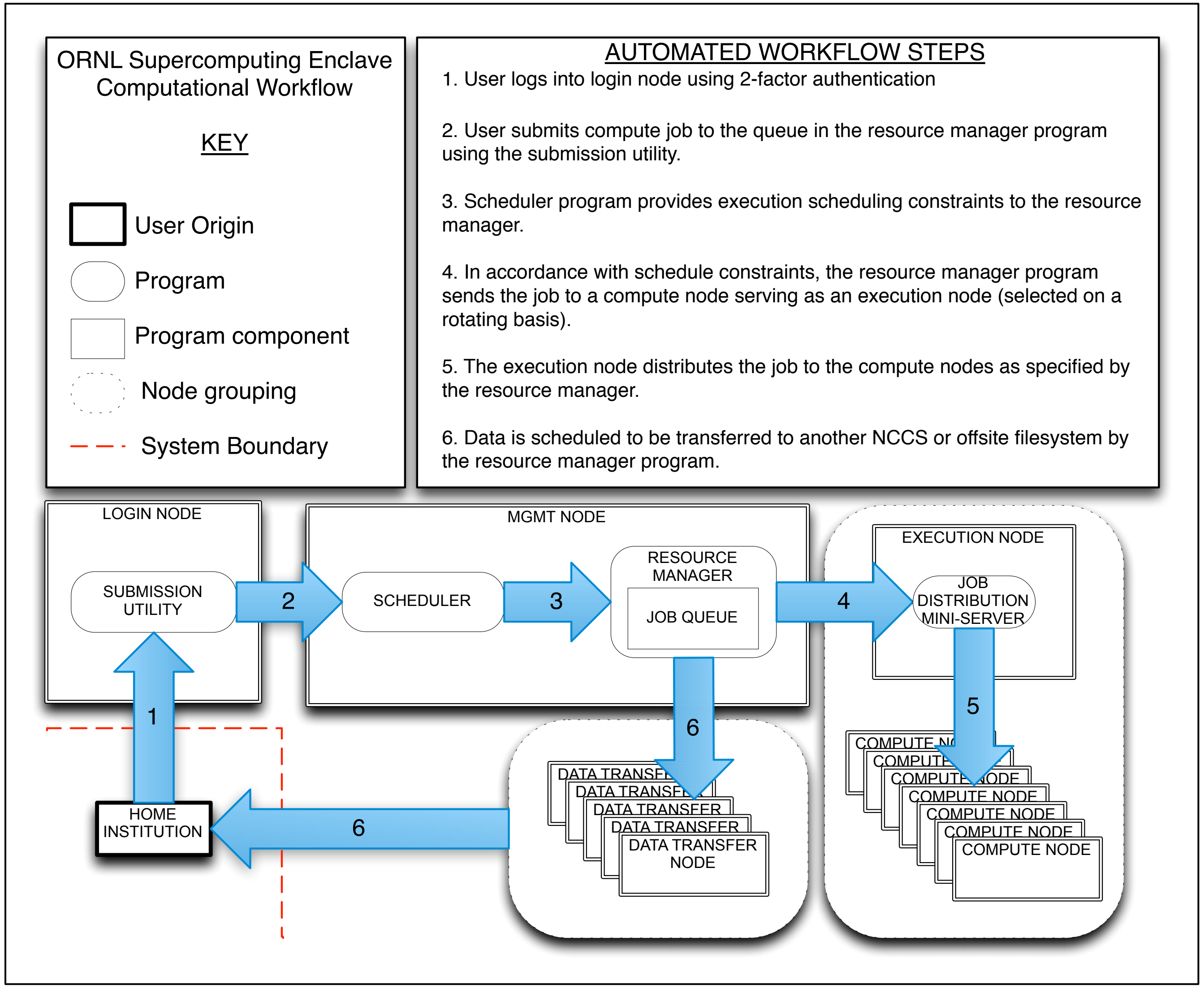
- Cryptocurrency mining
- Ransomware / holding data hostage
- Espionage and secret stealing
- Defacement / political motivations
- System sabotage
- Nuclear materials / weapons simulation
- Sowing mistrust in scientific discovery
- Planting evidence

Image by David Whelan, Licensed under Creative Commons CC0 1.0

HPC Security Model

- We can think of this in terms of ***sheriffs*** and ***deputies***
- **Sheriffs** represent facility-managed enforcement
  - Authorization, Firewall Rules, Scheduler policy
- **Deputies** are user-managed enforcement
  - Input sanitization
  - Correctness of code
  - Access to project

General user Workflow on HPC Batch Systems



ORNL Supercomputing Enclave Computational Workflow

KEY

- □ User Origin
- ○ Program
- □ Program component
- ◌ Node grouping
- – – System Boundary

AUTOMATED WORKFLOW STEPS

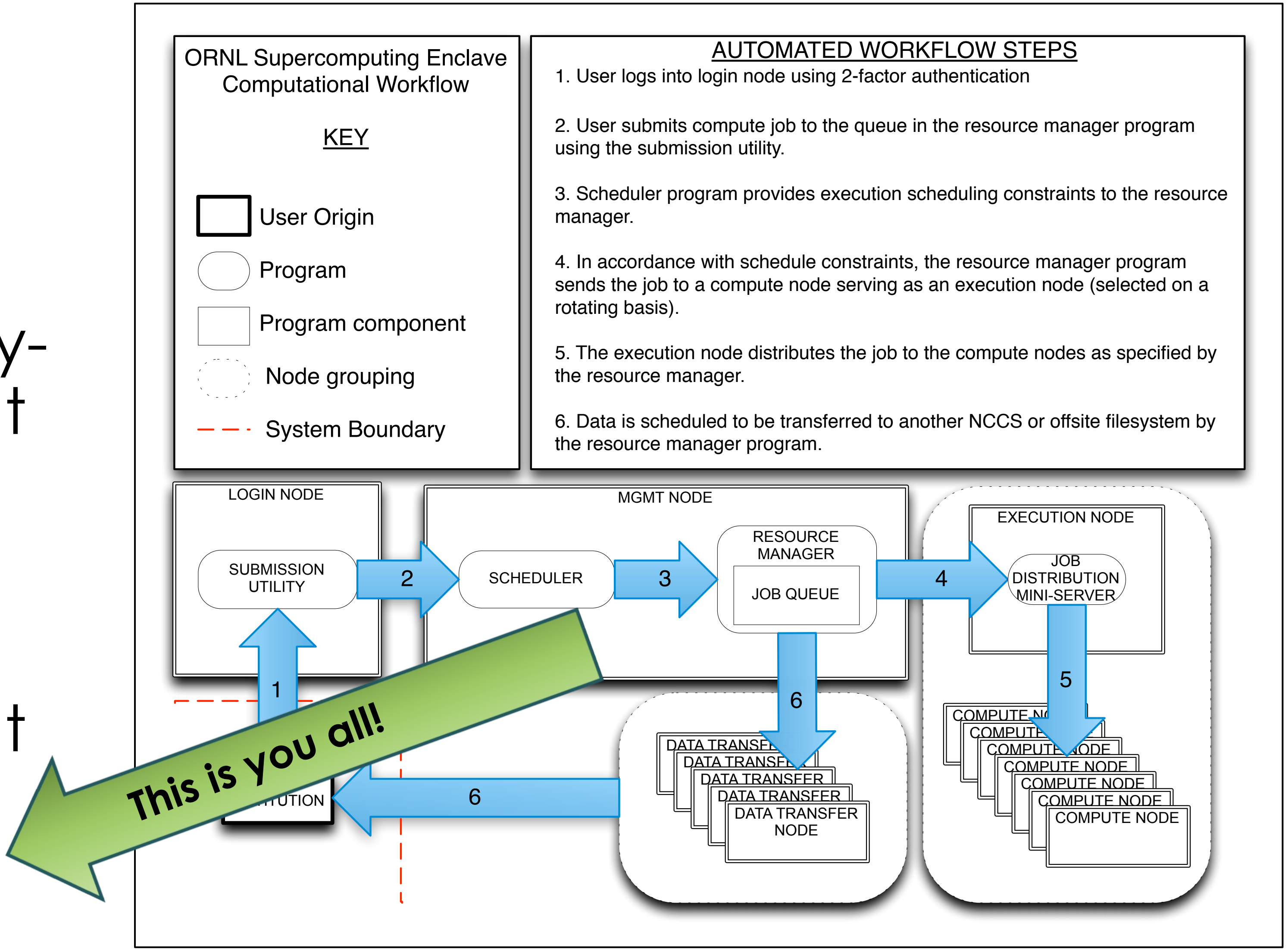
1. User logs into login node using 2-factor authentication
2. User submits compute job to the queue in the resource manager program using the submission utility.
3. Scheduler program provides execution scheduling constraints to the resource manager.
4. In accordance with schedule constraints, the resource manager program sends the job to a compute node serving as an execution node (selected on a rotating basis).
5. The execution node distributes the job to the compute nodes as specified by the resource manager.
6. Data is scheduled to be transferred to another NCCS or offsite filesystem by the resource manager program.

NIST SP 800-53 helps identify controls from an enterprise perspective but not all map well!

HPC Security Model

- We can think of this in terms of ***sheriffs*** and ***deputies***
- **Sheriffs** represent facility-managed enforcement
  - Authorization, Firewall Rules, Scheduler policy
- **Deputies** are user-managed enforcement
  - Input sanitization
  - Correctness of code
  - Access to project

General user Workflow on HPC Batch Systems



ORNL Supercomputing Enclave Computational Workflow

KEY

- □ User Origin
- ○ Program
- □ Program component
- ◌ Node grouping
- – – System Boundary

AUTOMATED WORKFLOW STEPS

1. User logs into login node using 2-factor authentication

2. User submits compute job to the queue in the resource manager program using the submission utility.

3. Scheduler program provides execution scheduling constraints to the resource manager.

4. In accordance with schedule constraints, the resource manager program sends the job to a compute node serving as an execution node (selected on a rotating basis).

5. The execution node distributes the job to the compute nodes as specified by the resource manager.

6. Data is scheduled to be transferred to another NCCS or offsite filesystem by the resource manager program.

NIST SP 800-53 helps identify controls from an enterprise perspective but not all map well!

Security Problem: Confused Deputy Attacks

- A **confused deputy** is a legitimate, more privileged computer program that is tricked by another program into misusing its authority on the system. It is a specific type of privilege escalation.[1]

From Wikipedia: https://en.wikipedia.org/wiki/Confused_deputy_problem

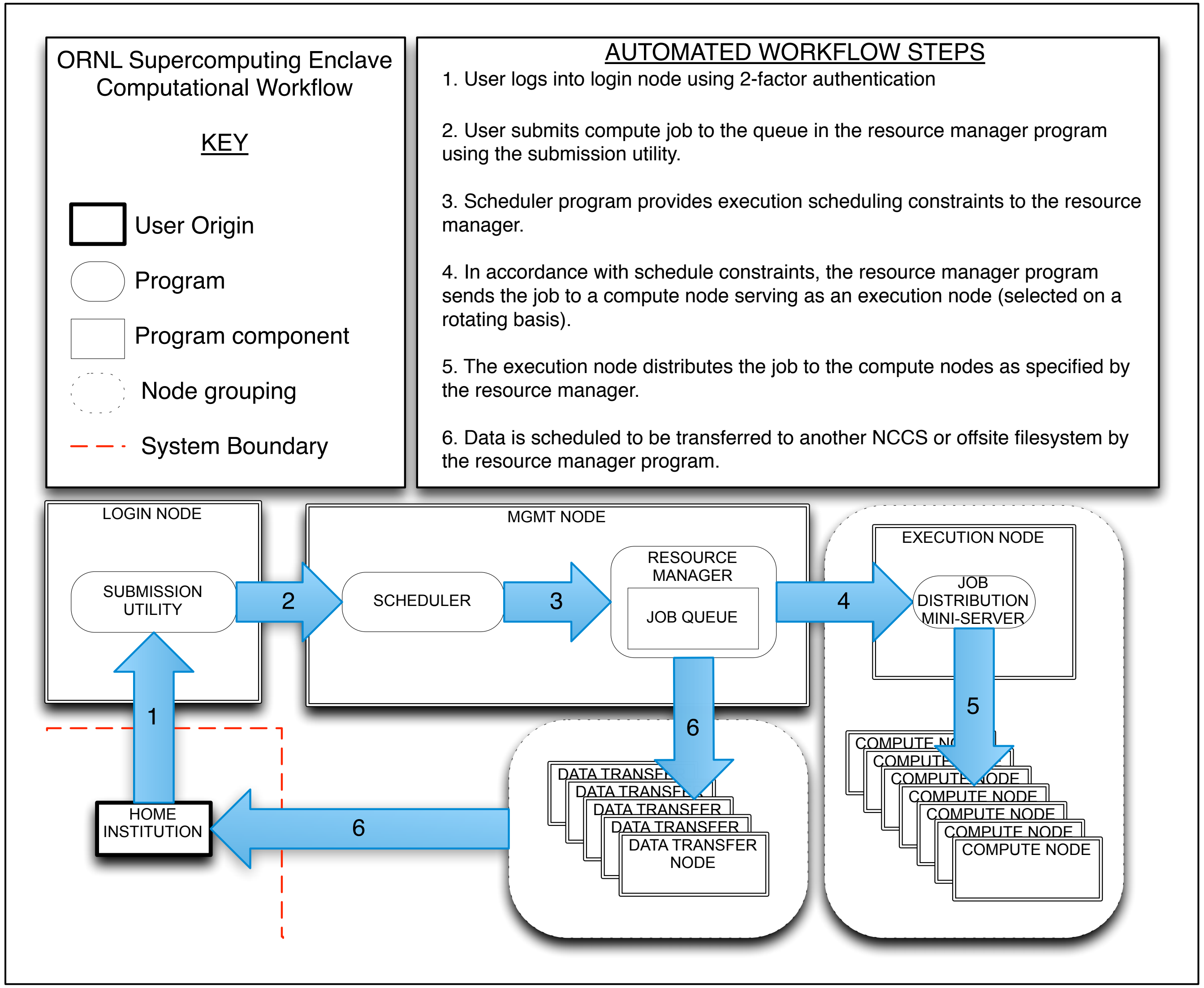
In the original example of a confused deputy,[3] there is a compiler program provided on a commercial timesharing service. Users could run the compiler and optionally specify a filename where it would write debugging output, and the compiler would be able to write to that file if the user had permission to write there.

A cross-site request forgery (CSRF) is an example of a confused deputy attack that uses the web browser to perform sensitive actions against a web application. A common form of this attack occurs when a web application uses a cookie to authenticate all requests transmitted by a browser. Using JavaScript, an attacker can force a browser into transmitting authenticated HTTP requests.

HPC Security Model

- We can think of this in terms of sheriffs and deputies
- Sheriffs represent facility-managed enforcement
  - Authorization, Firewall Rules, Scheduler policy
- Deputies are user-managed enforcement
  - Input sanitization
  - Correctness of code
  - Access to project

General user Workflow on HPC Batch Systems



AUTOMATED WORKFLOW STEPS

1. User logs into login node using 2-factor authentication

2. User submits compute job to the queue in the resource manager program using the submission utility.

3. Scheduler program provides execution scheduling constraints to the resource manager.

4. In accordance with schedule constraints, the resource manager program sends the job to a compute node serving as an execution node (selected on a rotating basis).

5. The execution node distributes the job to the compute nodes as specified by the resource manager.

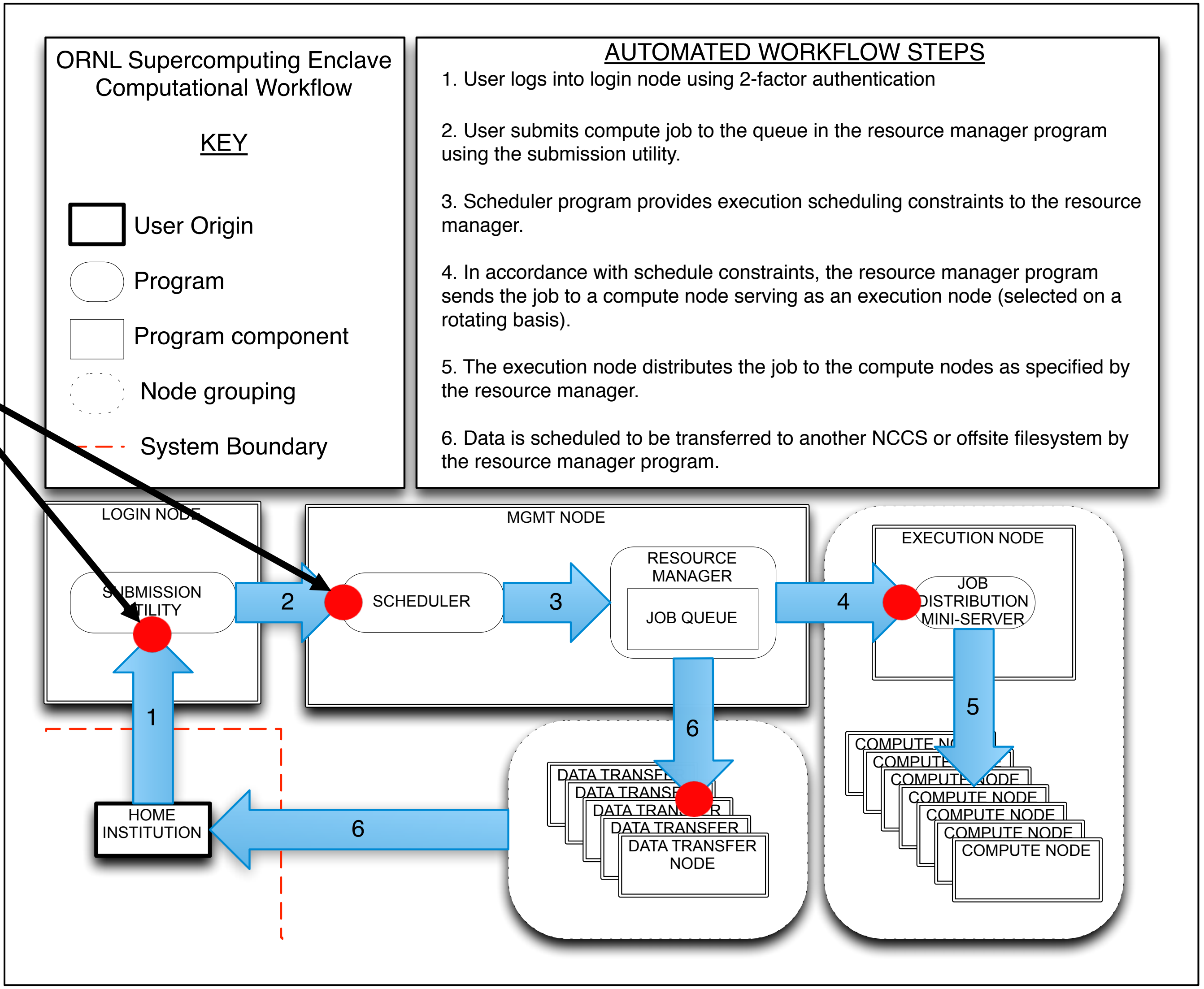
6. Data is scheduled to be transferred to another NCCS or offsite filesystem by the resource manager program.

NIST SP 800-53 helps identify controls from an enterprise perspective but not all map well!

HPC Security Model

- We can think of this in terms of sheriffs and deputies
- Sheriffs represent facility-managed enforcement
  - Authorization, Firewall Rules, Scheduler policy
- Deputies are user-managed enforcement
  - Input sanitization
  - Correctness of code
  - Access to project

General user Workflow on HPC Batch Systems



ORNL Supercomputing Enclave Computational Workflow

KEY

□	User Origin
▢	Program
□	Program component
◌	Node grouping
- -	System Boundary

AUTOMATED WORKFLOW STEPS

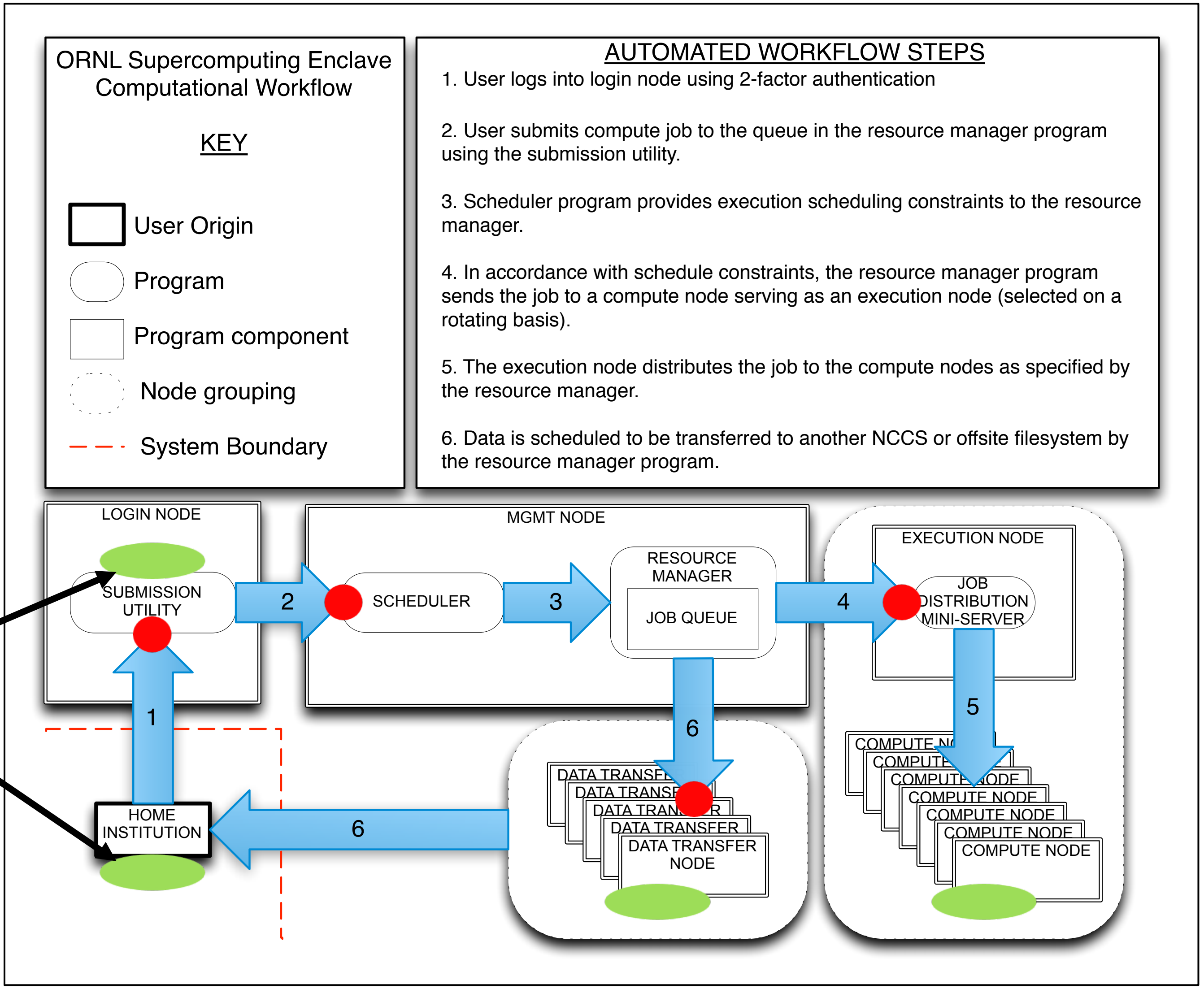
1. User logs into login node using 2-factor authentication
2. User submits compute job to the queue in the resource manager program using the submission utility.
3. Scheduler program provides execution scheduling constraints to the resource manager.
4. In accordance with schedule constraints, the resource manager program sends the job to a compute node serving as an execution node (selected on a rotating basis).
5. The execution node distributes the job to the compute nodes as specified by the resource manager.
6. Data is scheduled to be transferred to another NCCS or offsite filesystem by the resource manager program.

NIST SP 800-53 helps identify controls from an enterprise perspective but not all map well!

HPC Security Model

- We can think of this in terms of sheriffs and deputies
- Sheriffs represent facility-managed enforcement
  - Authorization, Firewall Rules, Scheduler policy
- Deputies are user-managed enforcement
  - Input sanitization
  - Correctness of code
  - Access to project

General user Workflow on HPC Batch Systems



ORNL Supercomputing Enclave Computational Workflow

KEY

□ User Origin

○ Program

□ Program component

◌ Node grouping

– – System Boundary

AUTOMATED WORKFLOW STEPS

1. User logs into login node using 2-factor authentication

2. User submits compute job to the queue in the resource manager program using the submission utility.

3. Scheduler program provides execution scheduling constraints to the resource manager.

4. In accordance with schedule constraints, the resource manager program sends the job to a compute node serving as an execution node (selected on a rotating basis).

5. The execution node distributes the job to the compute nodes as specified by the resource manager.

6. Data is scheduled to be transferred to another NCCS or offsite filesystem by the resource manager program.

NIST SP 800-53 helps identify controls from an enterprise perspective but not all map well!

Best Practices:

- Know that you and your project members have a significant role to play in securing our systems!

- Know what activities your project should or shouldn’t be performing, and ensure they are included in the statement of work within your project proposal.

Roadmap

1. HPC systems and their security model
2. **OLCF security ‘rules of thumb’**
3. Software best practices
4. Interconnections and workflow security
5. OLCF computing policy refresher

Rule of thumb (1): Identity

- **For every computational event or activity, there should be an identifiable person who intended for that action to happen**
  - Activities are pretty broad and include reads and writes, commands executed, jobs run, data transferred, etc
  - Identity, Authenticator, and Federation Assurance Level (IAL, AAL, FAL) dictate the strength of assurance that we have of this fact
- We make decisions based on who a person really is, not what accounts or credentials they have

Rule of thumb (2): Authorization

- **Activities should tie back to a scope of work for a particular project approved by the RUC**
  - The RUC authorizes scopes of work and delegates some authority to individual project PIs to request additional team members be involved in performing their project’s scope of work
  - Individual users are often on multiple projects, meaning that ‘permission creep’ is common
- If we know who users are, we can revoke access appropriately
  - Finer grained permissions are always better, if the system is scalable

Rule of thumb (3): Authentication

- **Strength of authentication is dictated by the access that it grants**
  - Access or potential access to Moderate data (export controlled information, proprietary information, and protected health info) requires 2 factor authentication (AAL2)
- The Facility must authenticate or delegate authentication to a trusted system
  - Individual users and user-managed programs should not be performing authentication on behalf of the facility
- Sharing or ‘delegation’ of authenticators is not allowed
  - This means sharing of tokens or passwords is forbidden
  - Globus Online breaks this rule in some sense, so we mitigate with shorter lifetime certificates

ORNL Follows US Department of Energy Policies

- The US Government ***requires*** departments to follow Federal Information Processing System policies, layed out in several publications.
  - FIPS 199
  - NIST SP 800-53 and 800-63
- Additionally, the US Department of Energy ***requires*** ORNL to follow other policies including
  - US Export Control Regulations
  - US Presidential Executive Orders
  - Orders regarding Foreign Visits and Assignments to Department of Energy laboratories

NIST Special Publication 800-53
Revision 5

Security and Privacy Controls for Information Systems and Organizations

NIST Special Publication 800-63-3

Digital Identity Guidelines

Paul A. Grassi
Michael E. Garcia
James L. Fenton

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-63-3

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

FIPS PUB 199

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

Standards for Security Categorization of Federal Information and Information Systems

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

February 2004

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Best Practices:

- Tokens ***must not*** be shared with anyone else, including graduate students.
  - Please have them sign up for a proper account!
- You should not travel with your OLCF token unless you will be performing OLCF work.
  - ORNL-issued equipment, including RSA tokens, are not allowed to be taken to certain countries that the US prevents exportation to.
- Let us know as soon as possible when project members no longer need access.

Roadmap

1. HPC systems and their security model
2. OLCF security ‘rules of thumb’
3. **Software best practices**
4. Interconnections and workflow security
5. OLCF computing policy refresher

Software Security Concerns

Has anyone encountered a segfault or caused a node to crash while executing code?

It is quite possible that sophisticated attackers can run code under your account!

Software Security Concerns

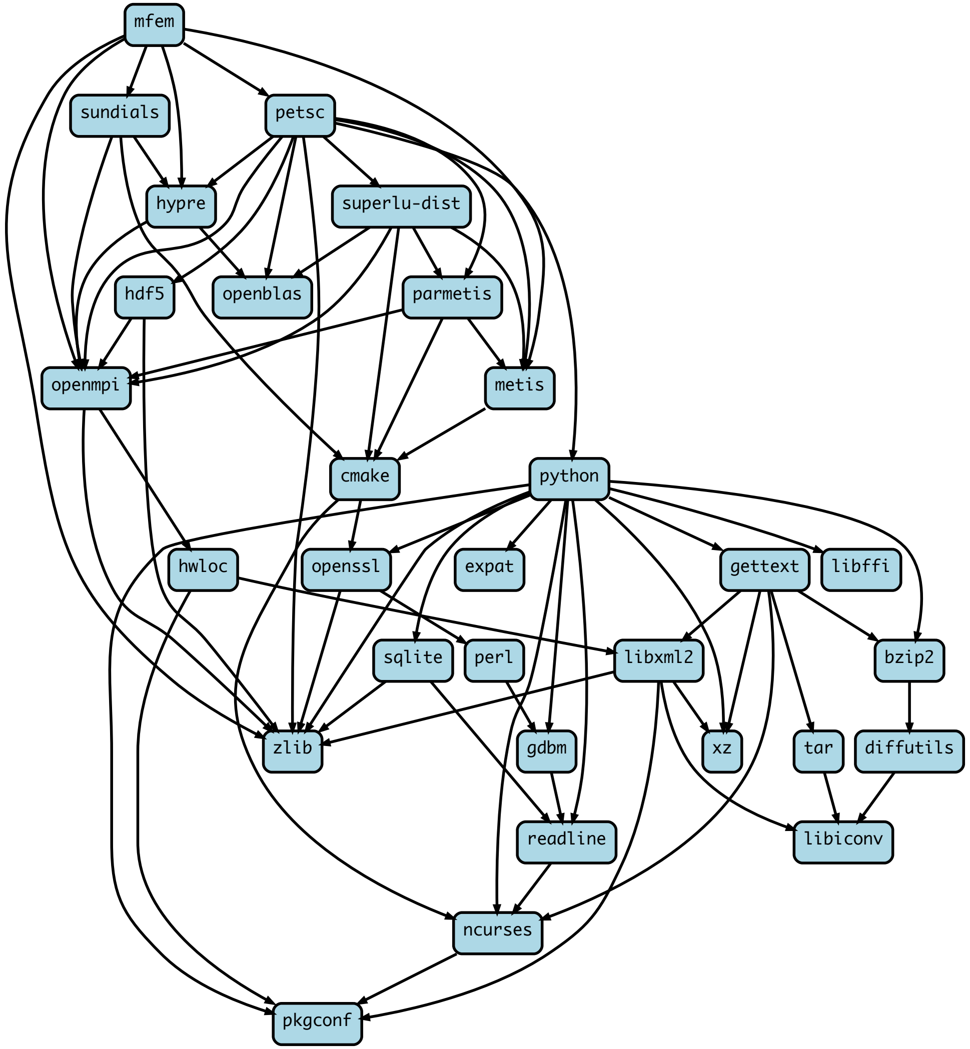
Has anyone had open source software libraries ‘update’ underneath your app which caused issues without you knowing?

The software supply chain conversation (both closed and open source)
is something that is gaining attention at all levels

Security Problem: Software Supply Chain Attacks

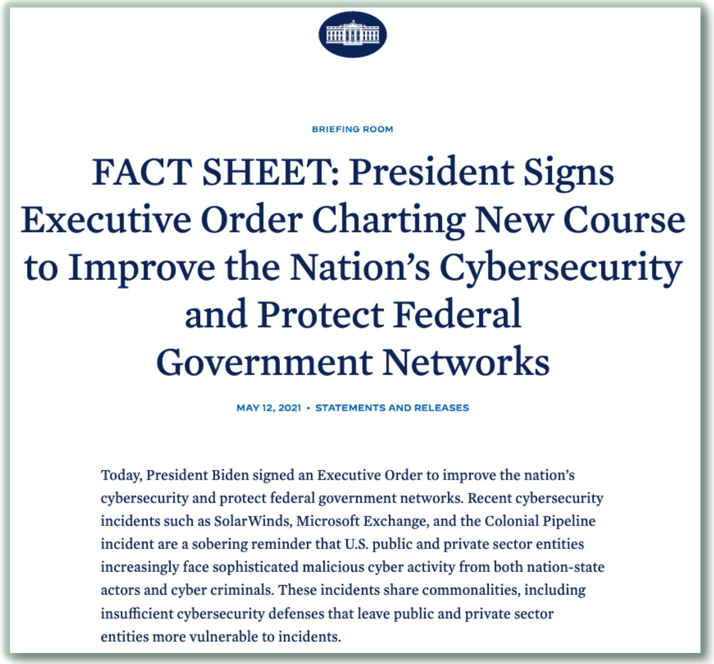
Exascale software stack characteristics

- Today's software stacks are **incredibly complex**
  - Tools like Spack are used to manage this complexity.
  - The graph to the right shows a single version of mfem and its dependencies.
  - Imagine how many individuals have contributed code… are they all benevolent?
  - Hundreds of additional packages are built to support the ~70 installed specs at an exascale facility.
- ECP has led several efforts to build HPC-specific Continuous Integration and Testing Frameworks
  - CI Pipelines are a prime target for attackers
  - Have we built enough security controls into these frameworks?



Where should pull requests to public repositories that can be created by anyone be tested?

Improving Software Supply Chain Security



BRIEFING ROOM

FACT SHEET: President Signs Executive Order Charting New Course to Improve the Nation's Cybersecurity and Protect Federal Government Networks

MAY 12, 2021 • STATEMENTS AND RELEASES

Today, President Biden signed an Executive Order to improve the nation's cybersecurity and protect federal government networks. Recent cybersecurity incidents such as SolarWinds, Microsoft Exchange, and the Colonial Pipeline incident are a sobering reminder that U.S. public and private sector entities increasingly face sophisticated malicious cyber activity from both nation-state actors and cyber criminals. These incidents share commonalities, including insufficient cybersecurity defenses that leave public and private sector entities more vulnerable to incidents.

*“**Improve Software Supply Chain Security.** The Executive Order will improve the security of software by establishing baseline security standards for development of software sold to the government, including requiring developers to maintain greater visibility into their software and making security data publicly available. It stands up a concurrent public-private process to develop new and innovative approaches to secure software development and uses the power of Federal procurement to incentivize the market. Finally, it creates a pilot program to create an “energy star” type of label so the government – and the public at large – can quickly determine whether software was developed securely. Too much of our software, including critical software, is shipped with significant vulnerabilities that our adversaries exploit. This is a long-standing, well-known problem, but for too long we have kicked the can down the road. We need to use the purchasing power of the Federal Government to drive the market to build security into all software from the ground up.”*

Necessary conditions for secure software stacks

- Vendors must build-in security from the start
  - **Especially** where security boundaries are crossed
- Long-term effort is required to maintain software
  - Without consistent attention, vulnerabilities and bugs cannot be fixed when they are discovered
- Software builds and tests need to be automated
  - CI pipelines exist at vendor, facility, and ecosystem levels
- Software Bills of Materials (SBOM) are necessary
  - Software inventories that are available and accurate will be used by security incident response
- We need focused vulnerability research
  - Vulnerabilities in software and the ecosystem exist. They are not frequently discovered, identified, reported, and fixed

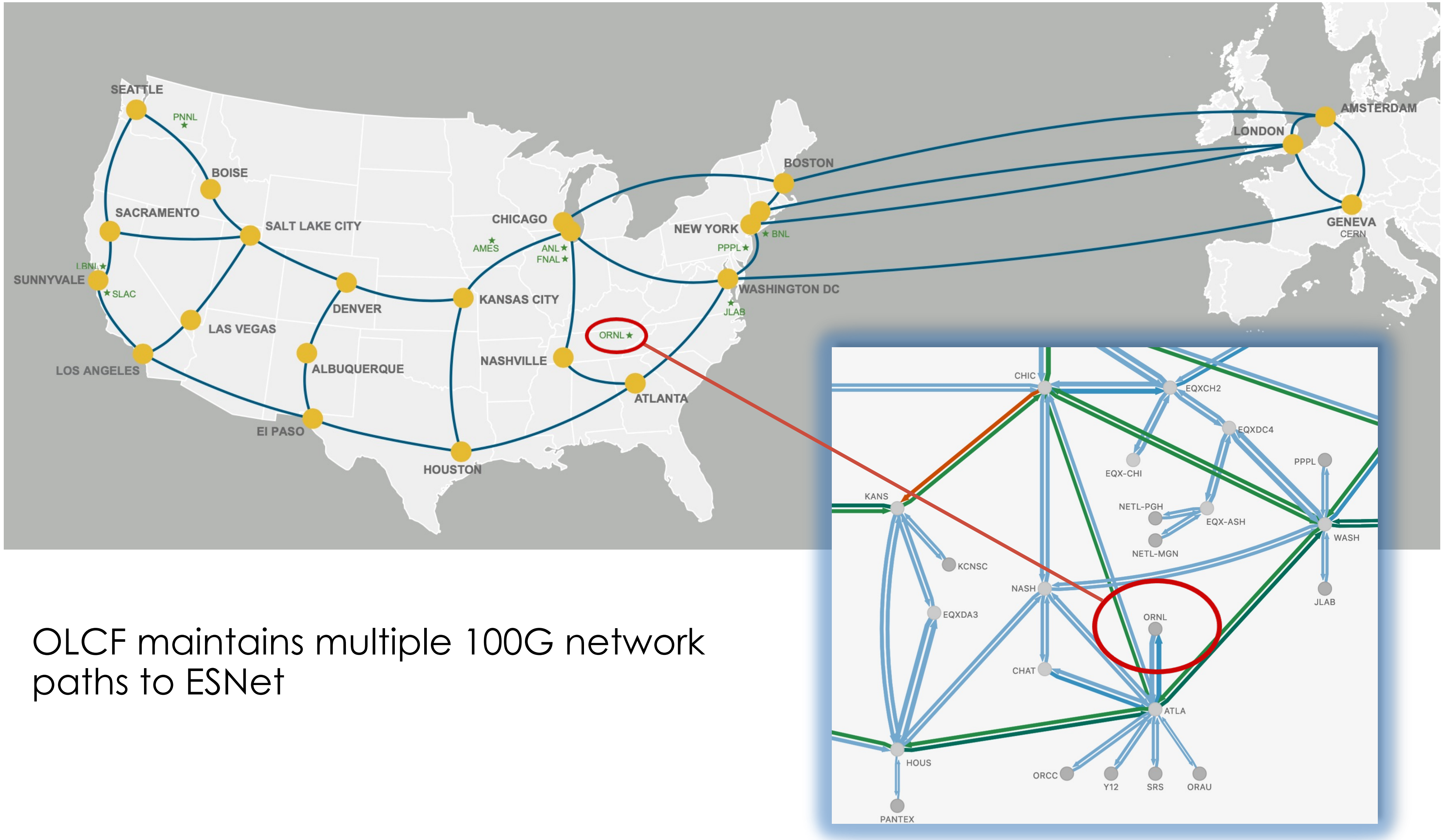
Best Practices:

- Utilize project and scratch areas with strong POSIX file permissions – is reading/writing to /tmp safe?
- Know where your software is coming from and who is contributing to it.
- Know where your input data is coming from and who has been able to modify it.
  - This could be considered ***Scientific Phishing***
- Perform ‘defensive programming’ and sanitize/validate all inputs.
- Utilize validation tests and continuous integration in order to write robust research software.

Roadmap

1. HPC systems and their security model
2. OLCF security ‘rules of thumb’
3. Software best practices
4. **Interconnections and workflow security**
5. OLCF computing policy refresher

ESNet Network is OLCF’s Front Door



OLCF maintains multiple 100G network paths to ESNet

What is a scientific workflow?

From Wikipedia: A **scientific workflow system** is a specialized form of a workflow management system designed specifically to compose and execute a series of computational or data manipulation steps, or workflow, in a scientific application.[1]

What is a scientific workflow?

- There are over 300 workflow systems!

https://github.com/common-workflow-language/common-workflow-language/wiki/Existing-Workflow-systems

  - Scientific domains usually have a preferred workflow tool
  - Some workflow tools operate specifically in a local mode called **Internal Orchestration** but many also enable remote actions via **External Orchestration**
  - External Orchestration usually requires caching credentials

- Workflows are almost always automated and typically expect the privileges and permissions that a user would have if they had logged in manually
  - Workflows are somewhat more **declarative** in contrast to interactive sessions which are **imperative**
  - Workflows require an understanding of available resources across different domains

Security Problem: Attack Chain / Kill Chain

- Defenders think in Lists. Attackers think in Graphs!
  - Security teams implement security policy controls and check boxes based on best practices and organizational mission
  - Attackers only need to find a single path of exploitation through a network to accomplish their goal
- Imagine a connected graph of 'state' nodes, with edges being actions that a particular account is allowed to take within a domain
  - Log in -> Compile Software -> Submit job -> Job Run -> Data Reduction -> Publish Data Set
  - A path that results in an undesired state is an **'attack chain'**

Barik et. Al examined at attack chains within a single organizational domain

M. Barik, A. Sengupta, and C. Mazumdar, "Attack graph generation and analysis techniques," Defence Science Journal, vol. 66, p. 559, 10 2016.

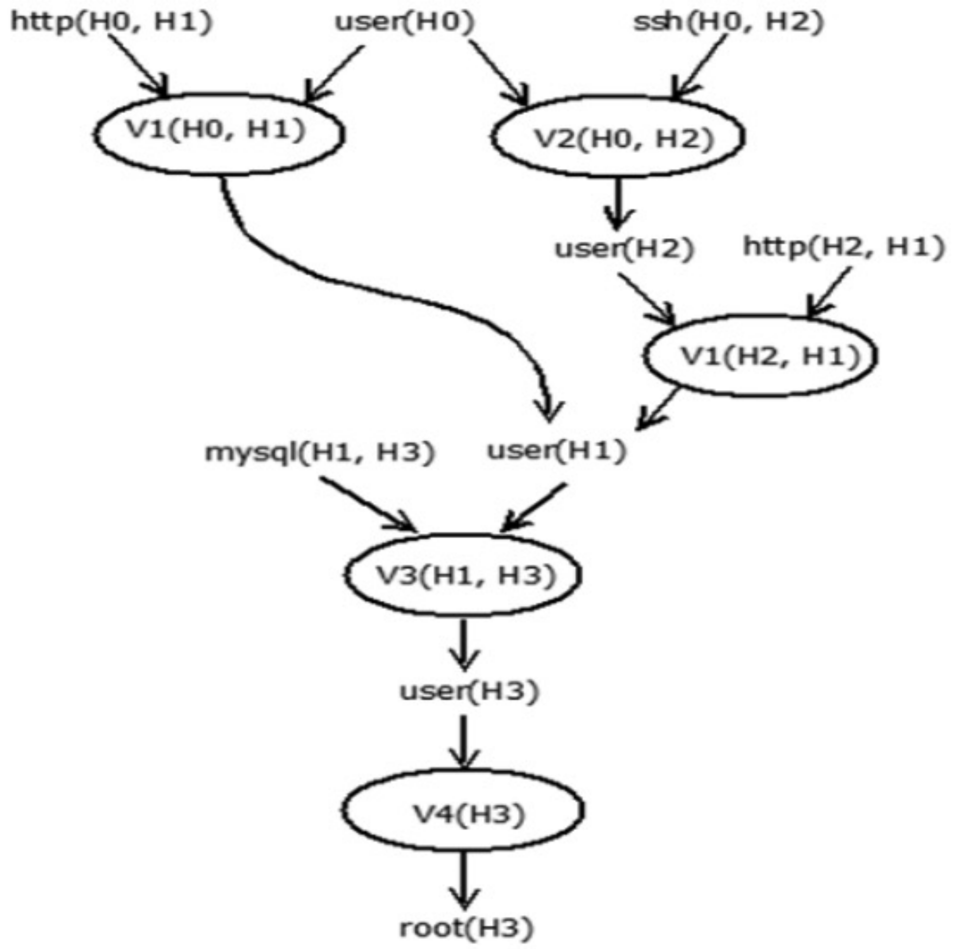
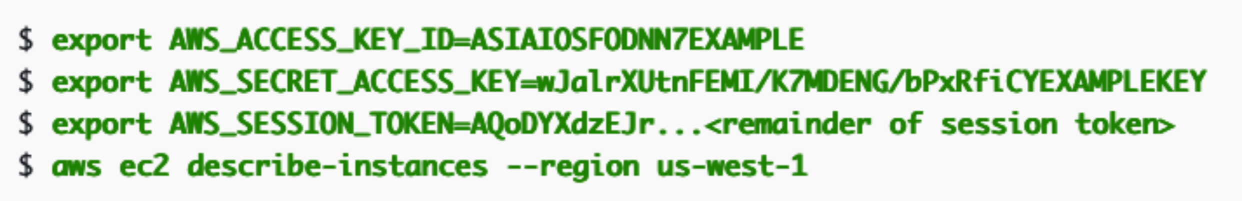


Figure 2. Exploit dependency attack graph.

OAK RIDGE National Laboratory

Scientific Workflow Best Practices:

- Know what ports and protocols your applications need to speak over the HSN and WAN.
- Prevent processes you don't control from communicating with the ones you do control.
- Utilize facility-managed authentication services
- Minimize the use of stored credentials and tokens
  - Things like:
    - Cloud based tokens
    - SSH keys
    - Environment Variables
    - Command Line Arguments
  - Could be read or used by:
    - Users on your project, now and in the future
    - Developers of code you run
    - Administrators at workflow sites



```
$ export AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
$ aws ec2 describe-instances --region us-west-1
```

Roadmap

1. HPC systems and their security model
2. OLCF security 'rules of thumb'
3. Software best practices
4. Interconnections and workflow security
5. **OLCF computing policy refresher**

OLCF Security Policy Reminders

OLCF computing resources are for business use only.

- Installation or use of software for personal use is not allowed.
- Incidents of abuse will result in account termination.
- Inappropriate uses include, but are not limited to:
  - Sexually oriented information
  - Downloading, copying, or distributing copyrighted materials without prior permission from the owner
  - Downloading or storing large files or utilizing streaming media for personal use (e.g., music files, graphic files, internet radio, video streams, etc.)
  - Advertising, soliciting, or selling

Sensitive information is protected at a higher level

- The following guidelines apply to sensitive data:
  - Only store sensitive data in designated locations. Do not store sensitive data in your User Home directory.
  - Never allow access to your sensitive data to anyone outside of your group.
  - Transfer of sensitive data must be through the use encrypted methods (scp, sftp, etc).
  - All sensitive data must be removed from all OLCF resources when your project has concluded.

https://docs.olcf.ornl.gov/accounts/olcf_policy_guide.html

OLCF Systems are Research Systems – Users and PIs are accountable for data used and activities performed

Forbidden Information and Activities

- Classified information
- Unclassified controlled nuclear information (UCNI)
- Naval nuclear propulsion information (NNPI)
- The design or development of nuclear, biological, or chemical weapons or any weapons of mass destruction

User Responsibilities

- Authors/generators/owners of information are responsible for its correct categorization as sensitive or non-sensitive as well as securing information during
  - Handling
  - Transmission
  - Processing
  - Storage
  - Disposal

PIs Responsibilities

- Principal investigators, users, or project delegates that use OLCF resources, or are responsible for overseeing projects that use OLCF resources, are strictly responsible for knowing whether their project generates any of these prohibited data types or information that falls under Export Control

Computers, software, and communications systems provided by the OLCF are monitored by the security team and are expected to be used for work associated with and within the scope of the approved project

Discussion

Acknowledgements:

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.