WORKING DRAFT

An Action Plan for High Performance Computing Security

Authors TBD

NIST Working Draft

Nov. 2016

1. Note to Reviewers and Coauthors

This is a working document and all content is tentative. NIST expects a period of iterative writing. Some text is intended only for coauthors and reviewers, to help the writing team agree on goals, status, etc. Such commentary text should use **bold** formatting, and will be removed prior to publication.

2. Executive Summary

TBD.

3. Acknowledgments

The NIST approach to collaborative writing is to acknowledge all who contribute unless they don't wish to be acknowledged. NIST has a guideline on when a contribution is so significant that the contributor should be an author; authorship decisions will be made when the document is nearly complete.

The authors **TBD** wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge John Russel of the National Science Foundation, Lee Beausoleil of the National Security Agency, Scott Sakai of the San Diego Supercomputer Center, Lei Ding of Intelligent Automation, Inc., Alex Malin of Los Alamos National Laboratory, Erik Deumens of U. Florida, Tim Polk, Office of Science and Technology Policy, **TBD**.

4. Introduction

In July of 2015, the President of the United States issued Executive Order 13702 to create a National Strategic Computing Initiative (NSCI). The NSCI was established to promote U.S. leadership in High Performance Computing (HPC) for the advancement of economic competitiveness and scientific discovery. Cybersecurity is a critical element of the NSCI mission. HPC security is necessary to ensure that machines are available for use, that HPC resources are not misused, that data produced is valid and trustworthy, and that sensitive information stored on HPC will not lose confidentiality.

In Sep. 2016, the National Institute of Standards and Technology (NIST) hosted a workshop to:

1. describe current HPC use cases and security practices,
2. identify HPC security gaps,
3. identify strategies for closing the gaps, and
4. make recommendations for next steps.

After the workshop, a number of workshop participants collaborated to document, confirm, and extend the workshop's results, producing this report which is organized by the four goals enumerated above.

5. Current HPC Use Cases and Security Practices

Many different academic, industrial, and government organizations use HPC systems, with a broad diversity of use cases across all technological readiness levels, from basic research in fundamental physics all the way to operational capabilities including weather prediction. Design features and specifications are tailored to mission-specific problems, frequently described on a spectrum from compute-intensive to data-intensive. HPC for compute-intensive problems, such as simulations and modelling, relies on fast calculation to develop a *causal* understanding of mechanisms as modelled systems evolve in time. In contrast, HPC for data-intensive problems, such as data analytics, image processing, and visualization, uses statistical *correlation* to identify and predict trends. Differences in the problem space have resulted in a very diverse HPC hardware and software ecosystem with a variety of architectures, including GPU accelerators and high memory machines. HPC is a tool used in common by different organizations, with different missions, for different problems, with different machines. Because it is so generally useful, cybersecurity for HPC is critically important and presents a unique challenge for the NSCI.

Stakeholders in the HPC ecosystem, as shown in Fig. 1, can include academic computing, national laboratories, industry, defense R&D, and HPC for operational problems, such as weather prediction, air traffic control, and transportation. The restrictions on operator use may depend in part on the regulatory environment, where there are legal obligations to provide greater security for sensitive information. Likewise, HPC may support critical infrastructure with requirements for continuous availability, and therefore operator restrictions may be greater. As a general rule of thumb, productivity and ease of use directly trades off with security, with the idea being that security typically benefits from user restrictions.

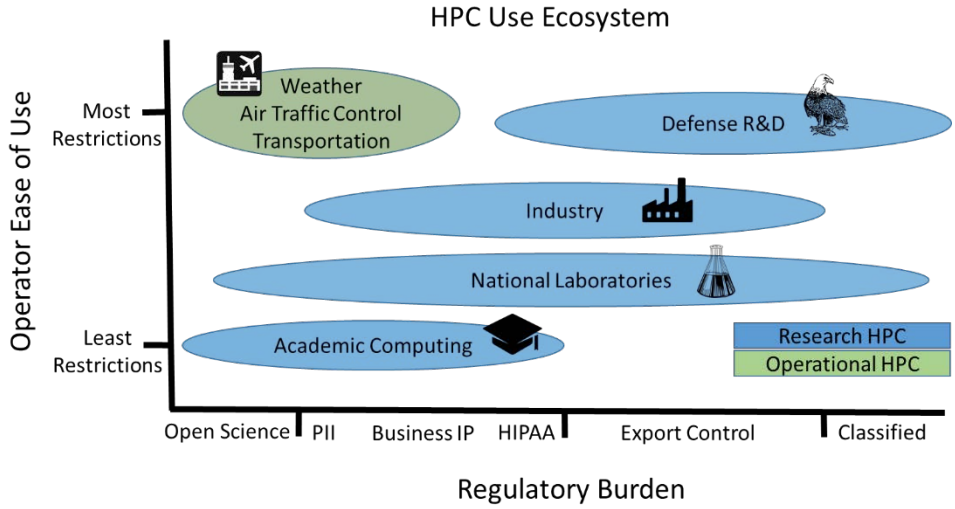

Figure 1. The regulatory burden of sensitive information imposes restrictions on operator use. HPC capabilities used for operational purposes may also have highly restricted operating environments where the information itself is not sensitive.

Traditionally, HPC systems in academic environments are *big iron* computers with domain scientist programmers writing code on bare-metal. A typical academic HPC system runs a Linux operating system on an exotic hardware and software stack with a connection to a high-performance Wide Area Network.

These systems are used to transfer data and to perform fast, parallel, and repetitive mathematical calculations for scientific problems.

The most powerful HPCs (see www.top500.org/statistics/sublist) have substantial physical footprints (e.g., similar to a basketball court) and are large distributed systems, containing millions of CPU cores, and consuming substantial amounts of power. In Dec. 2016, the most powerful, in terms of TFlops/s, is the Sunway TaihuLight at the national Supercomputing Center in Wuxi, China. The Sunway TaihuLight contains 10,649,600 CPU cores and achieves 93,014.6 TFlops/s. Programming such a system to fully apply its resources to a problem is a challenge in and of itself. Many considerations come into play, e.g., the sizes of different kinds of working memory and the ability to move data within the supercomputer as a computation proceeds.

Although characterizing a supercomputer by its cores suggests a compute-centric focus, many supercomputer jobs are more data intensive. In both cases, result data sets can be large, and supercomputers need to have the ability to store and then communicate very large result data sets at speed. Supercomputers often use Data Storage Nodes (DSNs), which are hosts that are optimized for high-performance data storage and transmission over Wide Area Networks [Dart] to perform this function.

Although many use cases use HPC in a batch processing style, some use cases, such as weather forecasting, may be on-line.

Large jobs can use all of an HPC.

Portal access sometimes is used for finding/searching-for results of calculations, and for limiting the actions of users.

Communities prepare code that prioritizes work.

Data from simulations published to a community to analyze.

Sometimes a system needs to persist the data for reference.

One approach to speeding up research is to run many simulations in parallel.

OpenStack is becoming more popular.

“bring your own VM” sometimes must also be supported.

Sometimes: dev, test, production all on the same system.

5.1. Commercial & Industry Use Cases

The increased adoption rate within the industrial and commercial sectors are becoming a significant driving force behind the evolution of HPC. Commercial applications require the processing and analysis of large amounts of data in sophisticated ways.

- "Big Data", databases, data mining
- Oil exploration
- Web search engines, web based business services
- Medical imaging and diagnosis
- Pharmaceutical design
- Financial and economic modeling

- Management of national and multi-national corporations
- Advanced graphics and virtual reality and networked video and multi-media technologies

[Source: https://computing.llnl.gov/tutorials/parallel_comp/#Overview]

5.2. Science and Engineering Use Cases

HPC's are used to model complex problems in many areas that need massively parallel computation and throughput.

- Atmosphere, Earth, Environment
- Physics - applied, nuclear, particle, condensed matter, high pressure, fusion, photonics
- Bioscience, Biotechnology, Genetics
- Chemistry, Molecular Sciences
- Geology, Seismology
- Mechanical Engineering
- Electrical Engineering, Circuit Design, Microelectronics
- Computer Science, Mathematics
- Defense, Weapons

[Source: https://computing.llnl.gov/tutorials/parallel_comp/#Overview]

Perhaps we should insert the top500.org usage application table here.

5.3. Current HPC Security Practices

The ScienceDMZ approach [Dart] provides four design patterns for organizing a data-intensive HPC for security. The first pattern is to locate the science DMZ at the high-performance WAN interface, thus preventing latencies that can be caused by traffic transiting conventional LANs. As pointed out in [Dart], even small delays can cause substantial backoff by TCP, causing extreme losses of performance. The second pattern is the use of dedicated DSNs, typically PC-based Linux servers, to provide data transfer onto a WAN, either singly, or in clusters. The third pattern is to use lightly-supervised performance monitoring to assist with the detection of misconfigurations or other imperfections (like aging components or loose cables) that cause dropped packets, and the tuning of a system's communications ability. The fourth pattern is to use "appropriate" security, which focuses on maximizing the use of high-performance router access controls (ACLs) rather than much slower and possibly buffer-limited firewalls. As pointed out in [Dart], if the set of applications performing data transfer is known and limited, the benefits of a deep-packet-inspecting firewall may be limited.

A science DMZ can also funnels scientific computing through its own enclave where data transfers can be monitored by tools, such as Bro IDS.

Some HPCs use facilities such as CILogon and how they manage incidents.

Use of web portals with constrained interfaces.

Risk management.

5.3.1. General HPC System Characteristics

Because HPC security practices must work within HPC system architectures and configurations, it is important to characterize those elements. For the purposes of this report, HPC systems exhibit the following characteristics:

Massively parallel systems - Millions of simultaneous thread execution on advanced processors.

Advanced processors - many-core and heterogeneous (GPU, CPU, FPGA).

High speed interconnects - not Ethernet (OmniPath™, InfiniBand, Aries™)

Maximum job configurations - execution of large jobs that can use every processor on the system that demand extreme reliability & scalability

Job scheduling - typical user develops and launches either batch or interactive jobs of varying sizes

Node types – login-nodes admit end users and allow them to prepare their jobs; compute-nodes perform calculations; data-storage-nodes allow users to transfer data into or out of a HPC facility.

Large, fast data stores -

- Fast non-volatile memory
- Parallel file systems
- Archival storage

[Source: mostly NIST HPC Workshop ORNL presentation]

6. HPC Security Gaps

Academic users of HPC systems are widely dispersed and frequently include international collaborations; these systems essentially cannot be "air gapped." The potential to do background checks on users is extremely limited. Many users are not citizens, and are not eligible for HSPD-12 cards. As a result, authentication of users is limited.

There are limits to use of machine learning and pattern matching for the detection of unauthorized applications because academic users often write their own code from scratch. E.g., the use of HPC to model hot plasma could be indistinguishable from a model of nuclear detonation. Because academic users can write their own code, it would be extremely challenging to develop markers for inappropriate research although some clearly inappropriate uses, such as bitcoin mining, probably can be identified The key security concept for academic HPC systems is to focus monitoring efforts on what is most important, such as critical data and the limited hardware and software stack.

Sensitive information presents challenges for HPC systems in academic environments. Research guidelines for appropriate handling of personally identifiable information (PII) and information governed by the Health Insurance Portability and Accountability Act of 1996 (HIPAA) require special handling for HPC systems, such as encryption and additional access controls to prevent loss of confidentiality.

Department of Energy (DOE) facilities at some national laboratories support academic research which can handle more sensitive information appropriately. E.g, at DOE open science HPC facilities such as at Oak Ridge National Laboratory, export controlled material can be identified and sandboxed properly; however, classified information is segregated from open science facilities.

At DOE weapons laboratories HPC systems for basic research are not available for general use by academic researchers.

In general, the security posture of NSF HPC systems are relatively weak compared to those of national security systems, such as those supported by the DOE National Nuclear Security Agency (NNSA) or Department of Defense (DoD). The difference in security posture correlated with the significance of the

national security mission, but also with the technology readiness level of the research performed. In other words, HPC systems focused on basic research systems have potentially low impact with respect to confidentiality, integrity, and availability (CIA). In contrast HPC systems focused on engineering or operational prediction, such as military vehicle design or weather prediction, have potentially high consequence. The requirements for validation and verification of models and code are much greater for research with high technology readiness levels.

Maliciously altered data could impact important research with policy implications, such as climate. Confidentiality is an issue for “open science” because even open science may include embargoed information due to export control or require special handling, such as medical data governed by HIPAA. Integrity and availability are issues, especially with respect to the misuse of computing cycles. For academic HPC, major risks are misuse of cycles by insiders (i.e. bitcoin mining), loss of integrity for important research data sets, and loss of confidentiality with respect to intellectual property such as source code. These risks can be mitigated by monitoring logs for bitcoin behavior by systems administrators and maintaining access controls to prevent damage/theft of data by unauthorized users. Some of these topics were addressed at the 2015 Cybersecurity for Scientific Computing Integrity - Research Pathways and Ideas Workshop organized by Robinson Pino (DOE). Requirements vary greatly based on the potential impacts of loss of confidentiality, integrity, and availability of information on HPC systems.

Management and technical experts responsible for implementing cybersecurity plans need a common language for risk decisions.

No clearly articulated security policy, which would express the harms that security features are intended to prevent. For example, some harms are:

1. scale of consequences is greater – more processing power to bring down,
2. more places to hide,
3. theft or misuse of HPC resources,
4. risk to application and data integrity,
5. traditional “government” concern with intellectual property confidentiality, and
6. insider Threat – everyone agrees on basic definition but specifics vary by sector.

Challenges:

1. Shared HPC Threat Intelligence and Best Practices
2. Wider selection Mandatory Access Control technologies that can work within parallel architectures
3. Minimally impactful HPC Anomaly Detection techniques
4. HPC Software Attestation – Application, Open Source, system
5. Ubiquitous and inexpensive multi-factor authentication
6. Inappropriate use
7. Ip theft
8. Denial of service

Unique aspects:

1. Scale of consequences
2. Paramount importance of integrity
3. Users are expected to be writing code, so malware has a way in
4. Security may be relaxed on compute nodes for performance

5. Rapidly-changing environment; it's hard to keep up
6. Bad behavior can be distributed among many parts of a system
7. There are potentially very many files to check
8. Can't afford both dev and test systems at the same scale

6.1. Insider Threat Challenge

The insider threat problem refers to scenarios where a person, who has legitimate, authorized access to a system, misuses that authorized access for nefarious purposes. Insider threats are particularly pernicious because the access rights given to some insiders may be very powerful and dangerous if intentionally misused.

Those individuals may be staff or users; we chose to exclude the user of stolen credentials even though they appear the same to the system.

It is important to baseline normal behavior so you can identify anomalies. It is critical to have a well-defined policy - so we know what is unacceptable, and to enable consequences for bad behavior.

Two (partial) countermeasures for the insider threat problem include 1) limiting the privileges of insiders perhaps using role based access control, and 2) developing behavior profiles that constitute "normal" activity, and detecting anomalous actions. Given the overriding specific context of scientific computing integrity, open research questions exist. For example, how we can better understand behaviors in HPC systems? How we can better understand the effect of those behaviors on security and scientific computing integrity? Where monitoring data can be collected in hardware and software? How custom hardware and software stacks can provide opportunities for enhanced security monitoring for HPC systems?

Locking down systems for open science conflicts with pursuit of science - Wild West is the key to innovation.

Some inside threats: unauthorized disclosure, deception, disruption, usurpation, use of HPC to run inappropriate code, escalation of privileges to gain additional resources or to grant to others, physical damage, DDos.

6.2. Supply Chain Challenge

China has identified HPC as a high priority area for investment, with a goal to build the entire supply chain from semiconductor to system integration. The Chinese government uses 5-year plans for social and economic development for which 2016-2020 includes HPC development. "Made in China 2025" is a Chinese program to develop a world class manufacturing sector, to convert from "Made in China" to "Made by China." The fastest Chinese HPC system, TaihuLight, is listed as #1 on the Top500 list running at 93.0 petaflop capability, compared to Titan, the US top machine, listed as #3 on the Top500 list running at 17.6 petaflop capability. TaihuLight is also reported to have 3 domestic simulation packages developed running at 30-40 Petaflops. The accomplishment is significant with respect to both hardware and software development. There is an emphasis on buying anything but US chips, and the Department of Commerce implemented a ban on Intel exports of select processors to China in 2015. In one year, the market share of US companies on the domestic-only China Top 100 HPC list (2013/2014) dropped to nearly zero. Additionally, for the first time, the US no longer has the most computers on the international Top500 list of most powerful supercomputers. **TBD: get reference for David Kahaner (ATIP), who provided this info.**

7. Strategies for Closing HPC Security Gaps

Ideas:

- Containerized operation
- Conduct advanced hunting operations
- Develop a common-sense security control baseline
- A cybersecurity framework for HPC
- HPC-compatible intrusion detection methods (light-weight, working with complex/massive systems)

Because the primary goal of HPC systems is to ensure large-scale computing applications to be executed in an efficient manner, the security solutions for HPC systems must be lightweight such that the overhead imposed by security solutions will not substantially degrade the computing performance. In other words, security solutions for HPC systems must ensure both computing performance and security requirements.

Areas of opportunity for new research include instrumentation and tools for HPC security, data integrity through the scientific lifecycle for large scale data sets, methods for integrated hardware/software security, and introduction of security into GPU-based computation, exascale proxies, neuromorphic chips, and quantum computing.

HPC has greatly advanced with numerous models for sharing memory, distributed architectures, growing numbers of processor cores, and special-purpose hardware (e.g., FPGA and GPU). Possible futures include optical computing, quantum computing, and neuromorphic computing. Although hardware has advanced greatly, the software running on HPC systems has a great deal in common with commodity system software, including common programming languages, applications, and even scripting. As such, software for HPC systems could experience many of the same security problems as conventional software, e.g., bugs, malware, and vulnerabilities. Automated static/runtime analysis tools might be developed to check HPC codes for insure behaviors. Scientific software security will become more fundamental. Besides, a large number of HPC application codes are written by divers set of participants. Training for software developers to follow secure coding practices can help reduce the impact of buggy and insecure code, such that application code vulnerabilities are more likely to be discovered and remediated before deployment.

In some cases, hardware features can significantly help to protect running software. Virtualization provides an abstraction over a system. Hardware virtualization features, for example, if used correctly, can implement protective boundaries around vulnerable software, and prevent some vulnerabilities from being exploited. Security tends to benefit from more constrained operation such as system-level virtualization and process-level virtualization. Trend toward containerized operation and limited interfaces in HPC may also help for improved user and process isolation.

Trusted platform modules can provide strong identity attestation and trustworthy bootstrapping. Tagged architectures can in principle maintain provenance information for processed data, and support very fine-grained access control. Hardware-supported monitoring feature can provide system-level introspection and enable close supervision of running programs for signs of compromise.

Software attacks involve corrupt processes or the operating system itself. The operating system must either be trusted and considered part of the trusted computing base or dealt with in another way. This is handled in different ways, for example, 1) adding a secure and trusted kernel to the system, or 2) using only hardware in the trusted computing base while treating the operating system as any other untrusted

process, or 3) a hybrid approach that include some portion of a trusted kernel or a trusted hypervisor along with hardware support.

Memory encryption and memory integrity checking can be used for securing computing hardware architectures. The overhead of decryption operations that precede loads requiring access to off-chip memory can be very high and hence can lead to significant performance degradation. Hardware counter-based encryption techniques are desired to reduce such overhead.

7.1. Science Gateways

The design patterns of the Science DMZ idea are not guarantees of security, but they constitute a set of tools with which data-intensive systems can be designed that achieve good performance with attention to security also.

7.2. Threat Analysis and Information Sharing

Traditional intrusion detection methods, such as looking at every byte crossing the wire, every file access, and scanning every file for malware is not feasible in HPC environments where the scale of the monitored variables are orders of magnitude larger. Having a (constantly updated) toolbox of HPC-compatible intrusion detection methods for each HPC installation to pick and choose from would help cut down overlapping development efforts. * "Threat" is actually misused in this context. A more correct term would be "threat actor" or "attack".

It might be good to develop an enumeration of threats (or actually, threat consequences) from individual participants of all applications of HPC, along with the participant's use of HPC. This could help frame otherwise conflicting priorities and provide a path forward based on application (or similar risk analysis), rather than the lowest-common-denominator of what every faction can agree on.

Work up a threat profile.

In the Open Science example, we care primarily about contamination of code or data. Protect the libraries. Assume competitive or hostile users on a hot button issue.

From an industrial perspective - the research data on the HPC is relatively low compared with final manufacturing designs, etc. The integrity of that data is important, but the CAD CAM design files for the final product are much more important.

Improper use like bitcoin mining . Practical worries on data - like HIPAA. Sometimes the data set is hard to replicate, so it is valuable. But the code is often what is really valuable. All of them want to game system to get to the head of queue, though.

NSF researchers - write their own code, compile their own code, doesn't look like anything else. I really don't want anyone else to get my source code, less worried about the data set. So, worried about users trying to access other users' directories or the shared scratch space. Not really worried about reverse engineering the code - assume I will have my paper published before you get the code reverse engineered.

7.3. Critical Baseline of Security Controls for HPC

Identify security controls that can reduce HPC security risk.

However, a one-size-fits-all template seems unlikely given the diversity of HPC systems.

7.4. Recommended Practices and Lessons Learned

This section would contain practices that are often called “best practices,” but without the implication that they are required.

Formulate the guidance with enough flexibility that the tolerance for risk can be set mostly by organizations using HPCs, rather than imposing a single approach.

Rules of thumb; concise accounts of trials and errors.

7.4.1. Establish usage policies

If you don't have policy, you are stuck and can't punish bad behavior. Example is bitcoin mining; if the policy doesn't say you are restricted to your Research proposal then no consequences. Perform anomaly detection to the extent feasible. HPC workflows may be more predictable than non-HPC.

7.4.2. Understand users as much as feasible

Identifying insider threat means you have to know what those users are doing when they behave legitimately, data for forensics, and then have software to analyze.

There are limitations with anonymous users.

7.4.3. Multi-factor authentication

You either will use it now, or implement it after you get popped. Often means you use only for front end, then find the gushy center.

7.4.4. Access control and restricted execution environments

Containers? Containers sound good, but container sprawl has proven to be a problem.

Enforcing policy from a central location. Even better if it is modular.

Better tool chains to control workflow and allocate resources.

Consider why tools like SELinux don’t get used.

Identify the sensitivity of data (e.g., PII, PHI, ITAR) and rules governing access to it.

7.4.5. Attestation for software

For example, using TPMs to implement trusted launch. Could be added to Lustre or something else but customers need to prioritize... would need a customer who was willing to pilot. Note that HPCs with millions of processors might benefit from such integrity checks.

Possibly statically analyze code when it’s in the queue.

7.4.6. collect and retain logs in a methodical fashion

It is best current practice to establish a centralized syslog server on an isolated system to collect logs in near real-time. This creates a base data set for forensics, etc. even if an attacker deletes your logs at the host. Time synchronization is a challenge, but is needed so that you can get ordering and causality. An alternate server is required for permanent (off site) record keeping. Note that the logs are also a key tool in analyzing system and application failures, so this security comes at low incremental cost.

The technical challenges associated with this BCP include volume of data, and obtaining/developing data analysis tools that can cope with the volume. One cluster with 2k compute nodes produces 6 Gbytes of syslog data, which tends to overwhelm Splunk or other data analysis tools. Tools are expensive, but a number of open source analysis tools are available including Apache Metron, Gigamon, etc...

7.4.7. share information with peer organizations

Coordinate and share information between HPC centers across organizations and sectors (gov't, industry, academia), and between leadership, budget, HPC operations, and cybersecurity within an organization.

Sharing best practices and threat information across sectors can potentially improve security for all, but current venues are not HPC focused. A common baseline would support security enhancements, since a principle of due diligence would emerge. There is a vicious cycle where budget won't be approved without leadership from management, cybersecurity can't be improved without budget, and management will not lead until cybersecurity explains the need.

Unfortunately, you probably need a translation layer.

Management challenge is cylinders of excellence in enterprise leadership, budget, and cybersecurity personnel. Best practices are not widely promulgated or available. Could really use an HPC ISAC. Need an HPC specific feed. Need to know key dependencies so we can understand.

7.4.8. Balance user needs with security requirements

Be skeptical of security requirements imposed across the board by outside players.

Review compliance challenges carefully. Verify that the compliance rules actually apply to your environment before investing time and resources. Few environments are uniform, so hierarchical requirements may need to be developed. But being too lax has its own issues...

The main Security Challenge is translating security requirements to HPC environments. A risk management approach is probably the best strategy.

A common Management Challenge is justifying the exceptions to an official that may not be as knowledgeable (about HPC or security).

7.5. Usage Reports

An important harm is misuse of HPC resources by users. While it is in general un-decidable from a computer science perspective to determine if processing is legitimate or illegitimate, one practical idea for addressing this issue was suggested. The representative from Oakridge said that they send out quarterly reports to their PIs that show the processing used by each user charging to the PI's project during the quarter. The PI is asked whether the results they are getting are consistent with the utilization shown for the quarter. Blue Horizon at the University Illinois also does something similar.

Perform anomaly detection to the extent feasible. HPC workflows may be more predictable than non-HPC.

7.5.1. Lessons Learned: Using FISMA

In 2014, decided that UF needed to have a FISMA NIST 800-53 moderate computing environment for research. We built one between March when they made that decision and July 1st, when the contract UF signed said it would be in place. Then we found that the security consultants, that we hired at significant expense, know how to build FISMA compliant infrastructure that include webservers and data bases to run a service for a mission like the Affordable Care Act. However, they do not know how to build infrastructure for the uncertain, short-notice, ever-changing, budget-constrained research mission of a university. The security people present have a more enterprise focused background and have some difficulty seeing how the research world operates.

HPC people have a strong focus on open science, trying to do the best they can to keep the system secure in a generic abstract sense.

The security people have a strong focus in enterprise IT security and see the open-ended research as plain chaos.

After we built this systems so fast, we found it is orders of magnitude too expensive to operate for even the best funded medical research project. It is too inflexible to accommodate the needs of rapidly evolving research needs, like ITAR. So we started, after our expensive lesson, to build a second system. We took the time to plan and we had a better idea of the needs and constraints. We now know we have to make a system that is usable and scalable and secure and compliant and affordable. It is not complete yet, but we are getting there. I have now learned that the answers exist and how to translate them into the "other" world's" language.

A lot of work has been done over the past decade, since FISMA 2002, that can be applied with some guidance and training and translation. I worked through the NIST documents and worked with Ron Ross, the people at FedRAMP, and some third-party accreditors. It is possible to apply the FISMA and NIST framework to research and get to an achievable and affordable place.

7.6. Risk Management

Build on the NIST risk management framework.

7.7. Common Terminology

Perhaps based on RFC 4949, "Internet Security Glossary, Version 2," https://tools.ietf.org/html/rfc4949. Official documentation should also be consistent with the language of RFC 2119, which establishes consistency for otherwise ambiguous phrases such as "should", "must", "must not", etc.

Code validation

Data set validation

Monitoring

Avoid "by rote" compliance – implement context specific compliance mechanisms.

Robust/comprehensive intrusion detection & centralized and isolated audit logging

Know the system, know the users before designing security features

Develop a threat information sharing approach, and an ISAC for HPC.

For some use cases, further development of portals where a reduced set of tools can be sufficient.

Situational awareness is extremely valuable. Automation is required to help reduce the amount of information that needs analysis by an operator. One approach is to improve categorization of risks. Acceptable risks can be defined as activity by known users. Hostile risks can be defined as data flows involving known bad actors. Unknown risks can be defined as flows where more investigation was needed. The important objective would be to improve detection of *actionable* security events, so that attention could be focused on situations where operator intervention is needed. This requires more research to help identify what those actionable malicious security events might be and to develop more innovative tools such as sinkholes or Nessus.

8. Recommendations for Next Steps

Some cloud providers verge on providing HPC services. In those cases, guidance on how to safely use outsourced HPC services may be useful.

8.1. Establish a HPC security critical control baseline

Security challenge - translating to HPC environments.

8.2. HPC Red Teaming

could we get someone to donate some cycles on a system and let the community red team it?

8.3. Organize a second HPC-Security workshop

1. Organize a workshop that starts with some training about the NIST framework for risk management, compliance, and controls. Providing both worlds with the same concepts and language.
2. Next have a scoping session: What does current and near-future research look like?
    a. The frontier of research involves big data and complexity. The goal and dream is to assemble teams that can address problems with big data input, using models and simulation, making inferences with deep learning, to make real-life products and services.
    b. Example: Medical research wants to be predictive using past history data, gene-sequence analysis, chemical modeling of drugs, models of the cell and the brain, all combined, to guide doctors, nurses, and surgeons in the clinical setting.
3. Then discuss the threats that HPC systems and work faces and that we need to protect against. There are two kinds, from the bad guys and from the good guys. From the example, it is clear that the algorithms and the data in the research projects will contain protected health information and intellectual property. That means it is “restricted data”, i.e. governed by laws, regulations, and contracts.
    a. The bad guys will want to steal the protected data and the intellectual property. Theft of that data has serious consequences for the operator of HPC systems and the research organizations and institutions. We want to avoid theft. This was clearly recognized at the meeting, but people were hedging how important it is to do so.
    b. The above research data is CUI (controlled unclassified information). The good guys will audit HPC systems and organizations to make sure they are operated in complaint ways. If problems are found, serious fines and costly remediation plans will be imposed on the organizations. We want to avoid fines.
    c. The HPC organizations that choose to stay away from such data, and stay with purely open data, will not be chosen as partners for the most competitive researchers working at the bleeding edge of science and of transitioning that science into real-life products and services. <u>The world of “pure open science HPC” is shrinking. The problems that the World needs us to solve require innovative collaborations that will be governed by contracts and that involve data governed by laws and regulation, even if, in the end, some of the findings will be published in the open scientific literature.</u>
4. Then discuss the design of security from the ground up.
    a. UF has chosen to build an environment on normal hardware and to put a security layer on top to provide security. The design starts with the assumption that the network is compromised, TLS certificates have been stolen, system admins have turned to the dark side, yet the users can collaborate on their data without anyone getting access to their data

unless they authorize it. This is just an example of what is possible to do today in a cost effective way.

9. Acronyms

10. Glossary

11. References

[EO13702] Executive Order 13702, National Strategic Computing Initiative (NSCI).

[Dart] R. Dart, L. Rotman, B. Tierney, M. Hester, J. Juraski, “The Science DMZ: A Network Design Pattern for Data-Intensive Science”, SC13 November 17-21, 2013, Denver, CO.

Public domain images used in figures:

Fig. 1:
https://openclipart.org/detail/237855/airport-terminal
https://openclipart.org/detail/770/bald-eagle-2
https://openclipart.org/detail/237859/factory
https://openclipart.org/detail/261899/flask-3
https://openclipart.org/detail/190967/graduation-cap