# Assessing Risk in High Performance Computing Attacks

Erika Leal[1], Cimone Wright-Hamor[2], Joseph Manzano[2], Nicholas Multari[2], Kevin Barker[2], David Manz[2] and Jiang Ming[1]
[1]*Tulane Univerity, U.S.A.*
[2]*Pacific Northwestern National Lab, U.S.A.*

Keywords: High Performance Computing, Risk.

Abstract: High-Performance Computing (HPC) systems are used to push the frontier of science. However, the security of these systems remains a significant concern as the number of cyber-attacks on HPC systems have increased. Attacks on HPC systems can threaten data confidentiality, integrity, and system availability. Thus, if left unaddressed, these threats could decrease the ability to push the frontier of science. While HPC and enterprise systems are found to have similar threats, traditional security solutions are insufficient for HPC systems. This research examines HPC attacks by using NIST Special Publication 800-30r1: Guide to Conducting Risk Assessments to create a generalized threat profile. A threat profile characterizes the threat sources and adversarial outsiders and is used to identify traditional security solutions that could mitigate risks. Results demonstrated that attacks originated at the login nodes, followed by coordinated campaigns that propagated the attacks across organizational systems. The traditional security solutions that could be used to protect the login nodes negatively impact HPC performance. These performance impacts impede the ability to push the frontier of science. As a result, these security solutions are unlikely to be deployed in HPC systems.

## 1 INTRODUCTION

High-Performance Computing (HPC) systems, such as supercomputers, have been used to push the frontier of science via modeling, simulation, and analysis. There is a considerable body of HPC research that focuses on increasing performance (Camier et al., 2021; Chen, 2017; Valeria Barra et al., 2020; Kaiser et al., 2014), but relatively little is understood about the security of HPC systems (Peisert, 2017). The security of these systems remains a significant concern as cyber-attacks on HPC systems have increased to impact the data confidentiality and integrity, and availability of HPC systems. Thus, if left unaddressed, these threats could decrease the ability to push the frontier of science. While HPCs and enterprise systems have similar threats, traditional security solutions are insufficient for HPC systems. This work aims to understand threats that have occurred to HPC systems and shed some light on why traditional enterprise-grade security solutions are insufficient for HPC systems. The NIST 800-30r1: Guide to Conducting Risk Assessments is used to conduct a qualitative assessment of six publicly documented attacks. Risk assessments are used to recognize significant trends, solutions, and decide where effort should be applied to eliminate or reduce threat capabilities (Blank and Gallagher, 2012). The Guide to Conducting Risk Assessments is a framework for federal agencies, and HPCs are typically funded and governed by the National Science Foundation, an independent federal agency. To our knowledge, no publicly available risk assessment has been performed on HPCs using NIST 800-30r1. We then use this information to generate a threat profile for HPC systems (Blank and Gallagher, 2012). A threat profile is a characterization of threat sources and adversarial outsiders. The profile is used to identify enterprise-grade security defenses capable of mitigating threats. However, these security defenses may counter the performance of the HPC system, which renders them insufficient. This paper is not intended to be a comprehensive survey of HPC attacks as the number of publicly documented attacks is limited. Although these cases may not provide the full picture of attacks on HPC, this may be the first risk assessment on such attacks of its kind.



Leal, E., Wright-Hamor, C., Manzano, J., Multari, N., Barker, K., Manz, D. and Ming, J.
Assessing Risk in High Performance Computing Attacks.
DOI: 10.5220/0011900500003405
In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, pages 793-803
ISBN: 978-989-758-624-8; ISSN: 2184-4356

## 2 HPC BACKGROUND

High Performance Clusters have exotic architectures that require high-performance networks, filesystems for production I/O, high floating-point performance, large bandwidth, and large storage. These requirements are needed to create high fidelity simulations of the physics of astronomical objects (Peres, 2003), variable resolutions of analysis of molecular interactions in both classical and quantum levels (Valeria Barra et al., 2020), simulation of future energy technologies, such as fusion reactors (Madduri et al., 2011), disease spreading (Minutoli et al., 2020), understanding how climate evolves (Bougeault, 2008), planning for resource allocation at the nationwide level (Huang and Nieplocha, 2008), among others. All these applications have massive computational requirements and / or intensive Input / Output exchanges with expected "reasonable" execution times[1]. Examples of HPC architectural designs are represented by the Oak Ridge National Laboratory's Summit (Melesse Vergara et al., 2019) and Frontier (Oakridge Leadership Computing Facility, ) Supercomputers, the RIKEN's Fugaku system (Fujitsu, ), the National Energy Research Scientific Computing Center's (NERSC) Perlmutter computer (Gerber, 2019), the Argonne Leadership Computing Facility's Aurora supercomputer (Argonne Leadership Computing Facility, ), the Livermore Computing Center's Sierra computer (Livermore Computing Center, ), among others. Supercomputers are represented by arrays of powerful computational components composed of Commercial-Off-The-Shelf (COTS) devices (e.g., CPUs, such as Intel, AMD, or PowerPC nodes, and GPUs, such as Nvidia A100 or V100 cards) with efficient network-on-chips fabrics (e.g., PCI express (Wilen et al., 2003)), NVLink (Foley and Danskin, 2017), QPI (Corp, 2009), etc.) and node interconnects (such as Cray's Slingshot (Hewlett Packard Enterprise, 2020) or Mellanox InfiniBand (Shanley, 2002)) that create high bandwidth capacities and low latency. Inside these components, one of the most important aspects of efficient computation is memory. Several memory types (Stocksdale et al., 2017; Pelley et al., 2014; Pandey, 2019; Wan et al., 2019) and low latency locality based storage (i.e., computer caches) might coexist inside these nodes to provide high-performance and low latency access to the data. Besides the components described above, a supercomputer facility might have a set of dedicated login/authentication nodes (for accepting users, preparing program and data, etc.), scheduling facilities, Data Transfer Nodes and/or Storage components. A typical user logs into the external facing nodes and then schedules their jobs into the computational components of the supercomputer (see (Argonne Leadership Computing Facility, ; Livermore Computing Center, ) for examples of this procedure). Moreover, as the computation evolves, produced data (this being simulation results, profiling/visualization information and others) can be saved to more permanent locations residing in the storage components or the data transfer nodes. The grouping of software components used to support the application in a given hardware architecture is called a software toolchain. This set is composed of compilers (tools to translate from higher languages to machine code and apply optimizations along the way, such as GCC (Stallman et al., 2009) and LLVM (Lattner and Adve, 2004)), interpreters (used to run high level code directly into the hardware, such as the Python interpreter (Rossum, 1995)), runtime systems (dedicated to organizing the computational flow in the hardware, such as OpenMP runtime(Dagum and Menon, 1998)), specialized libraries (designed to provide efficient solutions to domain problems such as the Intel GraphBuilder library (Intel Corp, ) or NVIDIA cuDNN (Chetlur et al., 2014)) and other tools to understand the behavior of the workload (such as debuggers and profilers, such as Intel VTune (Tsymbal and Kurylev, 2021)). Vendor supplied software tool chain, such as compilers and libraries, are repurposed and used to create scientific workflows. Experimental libraries (e.g., (The Trilinos Project Team, ; Dongarra et al., 2015)) and highly optimized[2] runtime systems (e.g., (Hammond et al., 2019; Kaiser et al., 2014; Khronos OpenCL Working Group, 2011; Vasilache et al., 2014; Hayashi et al., 2017)) are used to accelerate the application workloads even further via specialized scheduling algorithms, more powerful concurrency constructs, data marshalling techniques, among other techniques (such as the ones presented in (Blumofe et al., 1995). The various components coexist together to provide a fertile ground to efficiently implement applications. For example, taking advantage of the near memory storages (like methods introduced in (Patwardhan and Upadrasta, 2019) for accelerator's caches) and fully utilizing the available concurrency (by fully parallelizing the workload among nodes, cores, and hardware threads, as pre-

[1] Some of these applications deal with very complex problems that might take years to run if not correctly optimized and parallelized

[2] Optimization here refers to specialized data structures with optimized memory aware layout and access patterns, and methods that use architectural specific features to accelerate the computation that trades portability for performance

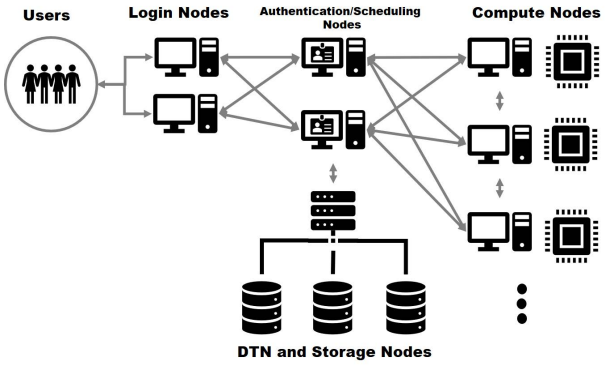


Figure 1: Overview of a typical HPC Architecture Design.

sented in (Sergeev and Del Balso, 2018) for machine learning workloads) is of utmost important to take advantage of the substrate. However this is not an easy task and, as a result of these complexities and the size of the HPC market, some of these optimizations are absent in commercial IT software environments. This leaves some of these workflows fully dependant on academic software that might be in an experimental state or not well supported (e.g., Combustion application S3D implemented in the novel Legion Programming Model (Chen, 2017) and its dependencies).

# 3 RELATED WORK

The need for HPC security-focused research has been acknowledged by experts (Hamlet and Keliiaa, 2010; Peisert, 2017; Blankenship, 2019). A workshop was created by NIST that gathered stakeholders from industry, academia, and the government to identify gaps in HPC-Security (NIST, 2018; NIST, 2016). As HPCs have specific purposes some traditional security measures could not be adapted for HPC. As such, those specific purposes could be leveraged for security (Peisert, 2017). Researchers have attempted to create a threat model for a wide range of clusters, including High-Performance Computing (Pourzandi et al., 2005). Integrating several security solutions due to a heterogeneous nature was nontrivial. Therefore, suggested security controls focused on distributed authentication, access control, monitoring, and secure communications.

# 4 RISK ASSESSMENT

The National Institute of Standards and Technology is a part of the U.S. department of Commerce. NIST Special Publication 800-30r1 is an American federal tool for measuring risk within information systems. The idea behind the Guide for Conducting Risk Assessments is to begin establishing a context of risk to to address the needs of an organization so that a broader risk management process begins. What we found is that HPCs face the same risks other systems face. However, due to the nature of HPC using traditional security solutions may not be enough.

## 4.1 Risk Assessment Methodology

The NIST Risk Assessment includes four steps: prepare for assessment, conduct assessment, communicate results, and maintain assessment. The scope of this assessment is limited to steps 1 through 3. Preparing for the assessment is necessary to establish a context. The objectives are to identify the purpose, scope, assumptions, and constraints of the assessment, sources of threat, vulnerability, and impact information to be used, and define the risk model, assessment approach, and analysis approach. While conducting a risk assessment, organizations will identify threat sources, threat events, vulnerabilities, predisposing conditions, the likelihood that threat events will result in adverse impacts, the likelihood of threat event occurrence, the level of adverse impact, and finally risk. Risk is a measure of the extent to which an entity is threatened by a potential circumstance or event, and is typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence. Finally, we chose to communicate results by documenting and sharing our risk assessment results by the way of this paper.

### 4.1.1 Prepare for Assessment

The risk assessment guide states that historical data on successful cyberattacks such as our cases can be used to perform a risk assessment. The purpose of performing the risk assessment is to identify common vulnerabilities and threat events that have occurred in

Table 1: Summary of Likelihood, Impact, and Risk for HPC Attack.

| Case | Likelihood | Impact | Risk |
|---|---|---|---|
| 1 | High | High | High |
| 2 | High | High | High |
| 3 | High | Moderate | Moderate |
| 4 | High | High | High |
| 5 | Very High | Very High | Very High |
| 6 | High | Very High | High |

past HPC attacks while also establishing a baseline of risk due to the harm resulting from the consequences of unauthorized access. Then we can apply that information to build a general threat profile for HPC. The threat profile will identify how risks of this type could be mitigated using traditional security solutions that exist. Finally, we can then discuss how these solutions fail HPC. The scope of this assessment includes publicly documented attacks that occurred in government-funded HPC systems. The information sources used in this assessment are a combination of reports, articles, keynotes, and legal documents (National Vulnerability Database, 2019; U.S. Department of Justice, 2013; Nixon, 2006; Barr et al., 2002; U.S. Department of Justice, 2009; Mclaughlin et al., 2015; Stoll, 1988). The assumptions and constraints are first, that all threat events that are mentioned were documented in our provided citations. Table E-2 from the NIST guide 800-30r1 was used to help identify threat events within the documentation, and actually, there were no different threat events in the case studies than what was stated on Table E-2. Our assumption is also that the vulnerabilities and predisposing conditions are the only vulnerabilities and predisposing conditions based on the case studies provided in our citations. Vulnerabilities and Predisposing Conditions are placed on a severity scale and a pervasiveness scale provided by the NIST guide. The process to conduct likelihood determinations is based on the assessment scales from the NIST guide. A likelihood score is given based on available evidence and judgement (Blank and Gallagher, 2012) assuming the information of the evidence is accurate. Impact determination is based on the evidence or actual events that transpired after the attack occurred. Risk tolerance is determined individually based on each attack and, again, will be based upon actual events. Uncertainty (Blank and Gallagher, 2012), unfortunately, will always exist and is a part of the risk assessment as it is inherent. Predictions on the future can only be made based upon the analysis of past attacks and cataloging of the similarities found. However, there is no guarantee that future similar attacks will have the same outcome or impacts. The publicly documented attacks will provide the sources of threat, vulnerabilities, and impact. The risk model used to define key risk factors was the risk model provided by the risk assessment which we were easily able to implement for HPC attacks. A qualitative approach is used to assess the attacks and analyze the attacks using a threat-oriented approach. The threat-oriented approach focuses on identifying the threat sources and threat events and the development of threat scenarios. Vulnerabilities are identified in the context of the threats, and impacts are identified based on malicious intent.

#### 4.1.2 Conduct Assessment

**Identify Threat Source.** Threat Sources for the cases assessed are considered to be Adversarial Outsiders. The characteristics of Threat Sources as defined by the risk assessment are Capability, Intent, and Targeting. The Capability of an adversarial outsider, in all cases, was rated moderate to very high. According to the NIST guide, an adversary would have at least a moderate amount of resources, opportunities, and expertise to conduct multiple successful attacks. In the cases where capabilities were rated as moderate the adversary supported multiple attacks with a moderate amount of resources. When giving a high score we concluded that the attacker had more sophistication in attacking due to multiple successful coordinated attacks, where multiple locations were invaded simultaneously. Finally, for a very high capability score, the adversary had a sophisticated level of expertise to support multiple, continuous, and coordinated attacks. All but one case are rated high to very high in Intent. Case 3 was rated moderate as there were no real clear intentions. However, the adversarial outsiders entered the system and continued in the system without detection for 6 months. They gained access to accounts through corrupted SSH shells and then focused selectively on higher accounts (OakRidge National Laboratory, 2019). A high intent score was given to an adversary who impeded systems while setting up ways to maintain a presence with minimal detection, as seen in cases 1, 2, and 6. In cases 1, 2, and 6, information was also disclosed on the internet with the Intent to sell (U.S. Department of Justice, 2013), espionage (Archer Support, 2020), and sensitive data scans (Stoll, 1988). A very high intent, as seen in case 5, was rated as the attacker only pursued confidential information and hacked over 97 computers in 13 months searching for it, as well as disrupted the United States Army's Military District of Washington network by deleting files (Department of Justice, 2002). In Targeting, half of the cases were rated as high. Case 1, case 4, and case 6. A high score was given as the attacker used information obtained

Table 2: Summary of Threat Events.

| Threat Event | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | | | | |
| Deliver modified malware to internal organizational information systems | x | x | x | x | x | x |
| Conduct brute force login attempts/password guessing attacks | x | x | x | | x | |
| Deliver malware for control of internal systems and exfiltration of data | | x | x | | | |
| Obfuscate adversary actions | x | | | x | | x |
| Coordinated campaign propagates attack across organizational systems | x | x | x | x | x | x |
| Adapt cyber attacks based on details surveillance | | | | x | | x |
| Perform network sniffing | | | | x | x | |
| Collect publicly accessible information on organization | | | | x | x | |
| Data Integrity loss on publicly accessible information systems | | | | x | | |

Table 3: Summary of Threat Sources.

| Case | Capability | Intent | Targeting |
|---|---|---|---|
| 1 | High | High | High |
| 2 | Moderate | High | Moderate |
| 3 | Moderate | Moderate | Moderate |
| 4 | High | High | High |
| 5 | Very High | Very High | Very High |
| 6 | Moderate | High | High |

via reconnaissance to continue targeting an organization while also focusing on high-value profiles or information. The critical difference between moderate and high is that in moderate, an attacker will only use publicly available information to target an organization, like in case 2, while in a high score, the attacker uses information obtained via reconnaissance. Finally, a very high score is extreme targeting. A very high score describes the attack as obtaining information through reconnaissance and then using that information to only target high-value positions within the organization as in case 5.

**Identify Threat Events.** Two threat events appeared in all cases. Table 2, s summary of all events per case, displays the two most common threat events. In all cases the common threat events were, the delivery of modified malware to internal organizational information systems and a coordinated campaign that propagates across organizational systems. This is most likely due to the fact that high-performance computers have a wide global network of users. The second most prevalent threat event was brute force login attempts. Then the third most prevalent events were obfuscating the actions of the adversary, performing network sniffing, collecting publicly accessibly information, delivering malware to extract data, and adapting the attack based on surveillance.

**Identify Vulnerabilities and Predisposing Conditions.** Vulnerabilities found in the HPC system cases were rated from at least moderate to very high. In the cases of vulnerability severity, moderate meant that some security control was implemented but not practical. A High vulnerability score implies that security control may have existed but was not implemented. A very high score purported that no security control existed at all. In all cases, the Impact of what happened due to the vulnerability reflects in the severity score. Predisposing Conditions affect the likelihood that threat events will result in adverse impacts. The type of predisposing condition found in all attack cases was Technical Architectural as systems were made vulnerable due to the predisposing condition of having various architectures and software which is a common practice in HPC. In cases where the attack spread to other users and systems, cases 2, 4, and 6 were given the predisposing condition Technical Function Networked Multi-User. It is made clear by our assessment that due to the HPC's distinctiveness of being technical, functional networked multi-user, results in the increased likelihood that an attack will occur. These two predisposing conditions occur in all attacks is no accident. As all High-Performance Computing Systems have multiple users and various hardware and software, these specific predisposing conditions make HPCs susceptible to attack, as proven by our study. Other predisposing conditions that HPCs may have based on predisposing conditions represented by our cases, is Information related and Technical Architectural OS. The predisposing conditions were then rated with Pervasiveness, as in who in the organization was affected by the attack. A moderate score is suggested that many users were affected, a high score is suggested that most were affected, and very high score in pervasiveness meant all users were affected.

**Determine Likelihood of Occurrence.** The likelihood that a threat can be initiated is a combination of the adversary's capability, intent, and targeting. According to the guide and the cases, an adversary would need at least a moderate amount of capability, intent, or targeting in order to have a moderate amount of likelihood the adversary would be able to initiate a threat. A moderately skilled attacker would then be

able to initiate a threat on HPC. It is then at least highly likely those threats would have adverse impacts as seen in Table IV. In all cases, the HPC systems had to be completely shut down and some never recovered. The overall likelihood that a threat would be initiated by an attacker and then have an adverse impact is at least high. A moderate threat initiation is considered to have a high adverse impact.

**Determine Magnitude of Impact.** The Impacts shown in Table 4, were Harm to Operations, Harm to Assets, Harm to Other Organizations, and Harm to the Nation. In all cases, there were adverse impacts to operations. The systems were taken down and not functional, as seen in all cases. In all cases, another prevalent adverse impact was harm to assets, where there was damage to informational assets, the network, and loss of intellectual property. Another prevalent type of Impact was Harm to Other Organizations, where the attack spread past the affected system into other institutions. Finally, one case had a direct Harm to the Nation which was case 5, which directly affected the United States of America. After identifying the type of Impact a Threat Event could affect, the overall impact of Threat Events on HPC systems ranged from moderate to very high as seen in Table 1. The cases were rated based on the evidence and what happened afterward. Case 1 Impact was rated as high not only due to the attack at hand but also because there exists an entirely new system (Archer Support, 2021). In case 2, the Impact was rated as high. According to (U.S. Department of Justice, 2013), the attacker impaired the integrity of his victim's network as well as damaged it. In response, NERSC began using an intrusion detection system named Bro that analyzes user command activity. In case 3, Impact was rated as moderate due to the system being taken offline and rebuilt to the impact of events, but nothing more was documented. In case 4, it was rated Impact as high. The Impact of case 4, resulted in global cooperation, damage to system integrity, damage of at least 5,000 dollars, and even the loss of a person (U.S. Department of Justice, 2009). In case 5, the Impact was rated as very high. The attacker shut down the United States Army District of Washington Network and deleted highly sensitive files. The attacker is suspected of crashing networks at the Naval Air Station where he was accused of deleting weapon logs as well as six counts of damage to The United States Army, The United States Navy, NASA, The United States Department of Defense, and the United States Air Force aggregating more than 5000 dollars (U.S. Department of Justice, 2009). In case 6, Impact of this attack was rated as very high for its' time. The attack caused damages of up to 100,000 dollars, risked the life of a person due to access violation of a real-time system, accessed confidential information, modified systems, read thru emails, and on-top of this, all systems had to be rebuilt (Stoll, 1988).

**Determine Risk.** Risk which is a combination of Likelihood and Impact, for all cases, were at least high as the effects in all cases were considered severe.

# 5 DISCUSSION

The results from this work provide evidence that HPC systems are high risk and have similar threats as enterprises systems. Instinctively, one would deploy enterprise-grade security solutions to mitigate HPC system threats. However, these security solutions may impede collaboration and reduce the usability of HPC systems. Threat profiles provide organizations with a characterization of common threat sources and known adversarial outsiders to guide decision-making for choosing cyber security solutions that minimize risk. Detailed below is a generalized threat profile created from the risk assessment results. The threat profile was used to identify enterprise-grade security solutions that could mitigate threats. In addition, a brief discussion for each security solution was identified to examine why these solutions may be insufficient for HPC systems. The general threat profile for HPC systems is an adversarial outsider with a high impact given a moderate to a high level of capability, intent, and targeting. All publicly documented attacks compromised the login nodes and SSH keys. In an enterprise system, the network and all devices within the network are managed by one organization. However, in an HPC system, the level of security is difficult to manage as there are multiple entities individually managing security. For example, the user of an HPC system could be a student from a university. In this scenario, the user is responsible for managing the security of the device they are using to connect to the HPC system. The university is responsible for managing the network's security the user is leveraging to access the HPC system remotely, and the HPC center is responsible for managing the security of their system. Unfortunately, each entity involved (the user, university, and HPC center) rarely coordinate to ensure a minimum level of security is met. The lack of a shared level of protection makes it difficult to provide secure authentication. In addition, many of these systems must be available 24/7 to accommodate international users, making it difficult to take offline to install updates. Thus, the systems tend to lack the latest updates and patches. These situations can invite malicious actors with moderate capability to compromise the clus-

Table 4: Summary of Adverse Impacts.

| Case | Harm to Operations | Assets | Other Orgs | Nation |
|---|---|---|---|---|
| 1 | x | x | x | |
| 2 | x | x | x | |
| 3 | x | x | | |
| 4 | x | x | x | |
| 5 | x | x | x | x |
| 6 | x | x | x | |

ter. These actors' attacks can range from espionage, scientific sabotage, financial gain, or disruption. Once attackers pass the user authentication processes, the computational nodes and File I/O can be exploited. Thus, updates, fixes, and patches on the login nodes, DTN's software stack, and hardware must take priority. Yet, this might not be entirely realistic in an HPC setting. The threat actor has a moderately high capability to attack an HPC system. This capability level implies that the threat actor is experienced, has a moderate amount of resources available, and has various opportunities to complete coordinated attacks. These coordinated attacks are exploited using a combination of zero-day attacks and known vulnerabilities. There are two reoccurring threat events, i)delivery of malware and ii)spread of attack to other organizations. When forming solutions to protect HPCs against these two threat events, one must be able to detect the malicious programs being delivered and enforce stronger boundaries between organizations. In an enterprise system, known vulnerabilities may be mitigated for example by strategically deploying firewalls. However, it is known that common defensive mechanisms such as firewalls are not practical as they cannot maintain the throughput required for high-speed data transfers, which is needed for HPCs to share data between compute nodes. Alternatively, the threat actor may disrupt mission-critical business functions by maintaining a presence in infrastructure. Given the advanced capabilities of the threat actor, they will conceal their presence to minimize attack disclosure. Reconnaissance is performed to gain information about the target organization and this information is used to target mission-critical business functions. Publicly available information about the organization might be used to target employees that support those mission-critical business functions. Unfortunately, it is challenging to discern attacker behavior from a legitimate user because HPC systems are designed for experimental purposes. As a result, all software running on the system is provisional and could be mistaken as malicious software. To further complicate the situation, the experimental software may not be designed using a secure development methodology. Most of this software is fragile, misused, and lacks the hardening necessary to increase its resilience and security. Moreover, the software is usually optimized for a specific HPC architecture, making maintenance difficult. Efforts are underway to update experimental software (e.g., high-performance libraries developed by research groups such as OpenMPI, Kokkos, Trilinos, and OpenBLAS). However, it might require a few years to update the large amount of legacy code spanning several decades of research. As shown in Table 2, the threat actor will perform actions that coordinate campaigns across multiple organizations to perform reconnaissance for a specific attack. The threat actor will use a sophisticated mechanism to distribute malware to internal systems. Threat actors establish a foothold via vulnerable software, OS, and networks. The threat actor will cause a high adverse impact. Expect a severe degradation of organizational operations, unavailability of assets, and business continuity disruption. A severe degradation may cause this in one or more mission-critical functions for a prolonged duration. The degradation could result in significant damage to organizational assets or financial loss. The same vulnerabilities that exist in the average enterprise system also exist in HPC. These vulnerabilities can at least moderately impact an HPC system and the validity of the data. For acceptance by the HPC community, a security defense employed in an HPC system must decrease risk while maintaining an acceptable level of performance. Traditional security tools search for known vulnerabilities and are designed to traverse networks similar to enterprise systems. HPC systems have exotic architectures and vulnerabilities may not always be easily identifiable. Additionally, HPCs host experimental software, thus the process of searching for CVEs will not identify threats because the software has not been released to the general public. Therefore, many security tools may not capture the critical threats to HPC systems. The experimental software is not always production ready and may contain unintended functionality that could be used to jeopardize the data integrity and confidentiality or disrupt the availability of the HPC system.

### 5.1 Moving Toward HPC Specific Security

In recent years, researchers have become increasingly interested in cybersecurity solutions for open science HPC systems. In 2013 researchers at Los Alamos described the challenges of continuous monitoring (Malin and Van Heule, 2013) in HPC and explored the development of such a tool. Then in a technical report from 2016, researchers at Lawrence Livermore National Lab deployed a continuous monitoring tool (Garcia-Lomeli et al., 2015) for system configuration information and security patch levels. At the National Center for Supercomputing Applications at the University of Illinois, due to an incident that resulted in the compromise of the cluster iforge, an automated tool that identified and excluded hosts that were experiencing an SSH brute force attack was created. CAUDIT (Cao et al., 2019) was the first of its kind that could handle large-scale networks and workloads without the interference of security experts and did not hamper the performance of the network. Due to the incorporation of this tool, they decreased their attacks from 30 security incidents a year to about 2. Another example of a low-latency and high-throughput tool is research from MIT Lincoln Laboratory Supercomputing Center (Andrew Prout et al., 2012) in 2012 and 2019 (Prout et al., 2019). In (Andrew Prout et al., 2012), the researchers created and implemented a cryptographic library used for user authentication on HPCs. Though this direction did not solve the escalation of privileges or offer protection from the compromise of a cluster or node, it did protect against the export of the cryptographic keys for malicious intent. In (Prout et al., 2019), MIT researchers continued their efforts in HPC authentication by incorporating federated authentication and scaling it to their HPC system. The Texas Advanced Computing Center (Proctor et al., 2017) described their experience switching their system over to MFA. In (Al-Mamun et al., 2019) (Al-Mamun et al., 2018), and most recently, (Al-Mamun and Zhao, 2020), and (Al-Mamun and Zhao, 2020), researchers explore trustworthy data provenance by enabling blockchain as a service as well as implementing the ledger of a blockchain into the HPC architecture to ensure trustworthy scientific data. Then in (Scheerman et al., 2021), suggest their novel platform that is not proprietary, as a service for protecting sensitive data, At MIT Lincoln Library, research surfaced for a system-level application that also had a low overhead to the network speed (Prout et al., 2016) that recognized user processes. In (Lee et al., 2021), researchers applied ACL filtering to existing firewalls of HPC systems in order to reduce CPU loads. In, (Pourzandi et al., 2005) created an example threat model for a wide range of clusters, including High-Performance Computing. The authors' mentioned that a challenge for cluster security was integrating several security solutions due to their heterogeneous nature. The authors suggested distributed authentication, distributed, distributed access control, distributed monitoring, and distributed secure communications. In (Peisert, 2017), HPC's have distinctive purposes and those purposes could be leveraged for security. It was explained how some traditional security measures could not be adapted for HPC. Fingerprinting workloads for classifying standard computations were also explored, an example of how security could move forward with machine learning.

## 6 CONCLUSION

In Open science environments (e.g., universities and national laboratories) there has been increased support for domestic and international research groups. Due to the collaborative nature of HPC systems, access to large quantities of data, computational power, and internet connectivity, these systems are prime targets for malicious activity. High Performance Computers need their own cybersecurity standards and methods in order to continue evolving with the ever changing cybersecurity landscape. We have taken some of the first steps in identifying risk in High Performance Computing. Though not exhaustive, it aids in HPC specific security. The NIST Special Publication 800-30r1: Guide to Conducting Risk assessment was used to evaluate successful attacks on HPC systems. Assessing these real world cases in our research can provide information to the community that may not have been available before. A threat profile based on the risk assessment provides researchers with cybersecurity context-specific to HPC systems and sheds light on the types of threats HPCs encounter. Then we explained why usual solutions cant work and gave examples of HPC specific solutions and research. The results from the risk assessment indicate that HPC systems attacks are coordinated campaigns across various organizations and have malware delivered that spreads to multiple parts. The threat profile indicated that a threat actor might target critical mission and business functions and information systems. The general threat profile for HPC systems is an adversarial outsider with a high impact given a moderate to a high level of capability, intent, and targeting. The threat profile provides a detailed record of the threat actor and can be used as the basis for future

HPC security research.

## REFERENCES


Al-Mamun, A., Li, T., Sadoghi, M., Jiang, L., Shen, H., and Zhao, D. (2019). Hpchain: An mpi-based blockchain framework for data fidelity in high-performance computing systems. In *Supercomputing*.

Al-Mamun, A., Li, T., Sadoghi, M., and Zhao, D. (2018). In-memory Blockchain: Toward Efficient and Trustworthy Data Provenance for HPC Systems. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3808–3813.

Al-Mamun, A. and Zhao, D. (2020). BAASH: enabling blockchain-as-a-service on high-performance computing systems. *CoRR*, abs/2001.07022.

Al-Mamun, A. and Zhao, D. (2020). SciChain: Trustworthy Scientific Data Provenance.

Andrew Prout, A., Arcand, W., Bestor, D., Byun, C., Bergeron, B., Hubbell, M., Kepner, J., Michaleas, P., Mullen, J., Reuther, A., and Rosa, A. (2012). Scalable Cryptographic Authentication for High Performance Computing. In *2012 IEEE Conference on High Performance Extreme Computing*.

Archer Support (2020). Archer Service Status.

Archer Support (2021). Archer Service Status.

Argonne Leadership Computing Facility. Aurora Introduction.

Barr, T., Langfeldt, N., Vidal, S., and McNeal, T. (2002). Linux NFS-HOWTO.

Blank, R. and Gallagher, P. (2012). Guide for Conducting Risk Assessments.

Blankenship, G. (2019). Keeping High-Performance Computers Cybersecure.

Blumofe, R. D., Joerg, C. F., Kuszmaul, B. C., Leiserson, C. E., Randall, K. H., and Zhou, Y. (1995). Cilk: An Efficient Multithreaded Runtime System. *SIGPLAN Not.*, 30(8):207–216.

Bougeault, P. (2008). High performance computing and the progress of weather and climate forecasting. In *High Performance Computing for Computational Science - VECPAR 2008*, pages 349–349, Berlin, Heidelberg. Springer Berlin Heidelberg.

Camier, J. et al. (2021). High-order lagrangian hydrodynamics miniapp (laghos).

Cao, P. M., Wu, Y., Banerjee, S. S., Azoff, J., Withers, A., Kalbarczyk, Z. T., and Iyer, R. K. (2019). CAUDIT: Continuous Auditing of SSH Servers To Mitigate Brute-Force Attacks. In *16th USENIX Symposium on Networked Systems Design and Implementation*.

Chen, J. (2017). S3d-legion: An exascale software for direct numerical simulation of turbulent combustion with complex multicomponent chemistry. In *Exascale Scientific Applications*, pages 257–278. Chapman and Hall/CRC.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., and Shelhamer, E. (2014). cuDNN: Efficient Primitives for Deep Learning.

Corp, I. (2009). An Introduction to the Intel QuickPath Interconnect.

Dagum, L. and Menon, R. (1998). OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.*, 5(1):46–55.

Department of Justice, U. (2002). London, England Hacker Indicted Under Computer Fraud and Abuse Act For Accessing Military Computers.

Dongarra, J., Gates, M., Haidar, A., Jia, Y., Kabir, K., Luszczek, P., and Tomov, S. (2015). HPC Programming on Intel Many-Integrated-Core Hardware with MAGMA Port to Xeon Phi. *Sci. Program.*, 2015.

Foley, D. and Danskin, J. (2017). Ultra-Performance Pascal GPU and NVLink Interconnect. *IEEE Micro*, 37(2):7–17.

Fujitsu. Fugaku Supercomputer: Specification.

Garcia-Lomeli, H. D., Bertsch, A., and Fox, D. (2015). Continuous security and configuration monitoring of hpc clusters. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Gerber, R. (2019). Perlmutter A 2020 Pre-Exascale GPUaccelerated System for Simulation, Data and Learning. In *International Computing for the Atmospheric Sciences Symposium (iCAS)*.

Hamlet, J. R. and Keliiaa, C. M. (2010). National cyber defense high performance computing and analysis: concepts, planning and roadmap. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ....

Hammond, J. R., Kinsner, M., and Brodman, J. (2019). A Comparative Analysis of Kokkos and SYCL as Heterogeneous, Parallel Programming Models for C++ Applications. In *Proceedings of the International Workshop on OpenCL*, IWOCL'19, New York, NY, USA. Association for Computing Machinery.

Hayashi, A., Paul, S. R., Grossman, M., Shirako, J., and Sarkar, V. (2017). Chapel-on-X: Exploring Tasking Runtimes for PGAS Languages. In *Proceedings of the Third International Workshop on Extreme Scale Programming Models and Middleware*, ESPM2'17, New York, NY, USA. Association for Computing Machinery.

Hewlett Packard Enterprise (2020). HPE SLINGSHOT: The Interconnect for the Exascale Era.

Huang, Z. and Nieplocha, J. (2008). Transforming Power Grid operations via High Performance Computing. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–8.

Intel Corp. Intel Graph Builder.

Kaiser, H., Heller, T., Adelstein-Lelbach, B., Serio, A., and Fey, D. (2014). HPX: A Task Based Programming Model in a Global Address Space. In *Proceedings of the 8th International Conference on Partitioned Global Address Space Programming Models*, PGAS '14, New York, NY, USA. Association for Computing Machinery.

Khronos OpenCL Working Group (2011). *The OpenCL Specification, Version 1.1*.

Lattner, C. and Adve, V. (2004). LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In *Proceedings of the International Symposium on Code Generation and Optimization: Feedback-Directed and Runtime Optimization*, CGO '04, page 75, USA. IEEE Computer Society.

Lee, J.-K., Hong, T., and Li, G. (2021). Traffic and overhead analysis of applied pre-filtering ACL firewall on HPC service network. *Journal of Communications and Networks*, pages 1–9.

Livermore Computing Center. Using LC's Sierra Systems.

Madduri, K., Ibrahim, K. Z., Williams, S., Im, E.-J., Ethier, S., Shalf, J., and Oliker, L. (2011). Gyrokinetic toroidal simulations on leading multi- and manycore HPC systems. In *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12.

Malin, A. and Van Heule, G. (2013). Continuous monitoring and cyber security for high performance computing. In *Proceedings of the First Workshop on Changing Landscapes in HPC Security*, CLHS '13, page 9–14, New York, NY, USA. Association for Computing Machinery.

Mclaughlin, M.-D., Cram, W. A., and Gogan, J. L. (2015). A High Performance Computing Cluster under Attack: The Titan incident. *Journal of Information Technology Teaching Cases*, 5.

Melesse Vergara, V., Joubert, W., Brim, M. J., Budiardja, R., Maxwell, D., Ezell, M., Zimmer, C., Boehm, S., Elwasif, W., Oral, H., Fuson, C., Pelfrey, D. S., Hernandez, O., Leverman, D. B., Hanley, J. A., Berrill, M., and Tharrington, A. (2019). Scaling the summit: Deploying the world's fastest supercomputer. In *International Workshop on OpenPOWER for HPC (IWOPH 2019)*.

Minutoli, M., Sambaturu, P., Halappanavar, M., Tumeo, A., Kalyanaraman, A., and Vullikanti, A. (2020). Preempt: Scalable epidemic interventions using submodular optimization on multi-gpu systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

National Vulnerability Database (2019). CVE-2019-15666 Detail.

NIST (2016). NSCI: High-Performance Computing Security Workshop.

NIST (2018). High-Performance Computing Security Workshop.

Nixon, L. (2006). The Stakkato Intrusions: What Happened and What Have We Learned? In *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*.

Oakridge Leadership Computing Facility. FRONTIER Spec Sheet.

OakRidge National Laboratory (2019). TITAN: Advancing the Era of Accelerated Computing.

Pandey, A. (2019). DDR5: Fifth-generation of DDR Memory Module.

Patwardhan, A. A. and Upadrasta, R. (2019). Polyhedral Model Guided Automatic GPU Cache Exploitation Framework. In *2019 International Conference on High Performance Computing Simulation (HPCS)*, pages 496–503.

Peisert, S. (2017). Security in High-Performance Computing Environments. *Communications of the ACM*, 60(9).

Pelley, S., Chen, P. M., and Wenisch, T. F. (2014). *Memory Persistency*, volume 42. Association for Computing Machinery, New York, NY, USA.

Peres, G. (2003). HPC in Astronomy: overview and perspectives. *Memorie della Societa Astronomica Italiana Supplementi*, 1:223.

Pourzandi, M., Gordon, D., Yurcik, W., and Koenig, G. (2005). Clusters and security: distributed security for distributed systems. In *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.*, volume 1, pages 96–104 Vol. 1.

Proctor, W. C., Storm, P., Hanlon, M. R., and Mendoza, N. (2017). Securing HPC: development of a low cost, open source multi-factor authentication infrastructure. In *SC '17: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*.

Prout, A., Arcand, W., Bestor, D., Bergeron, B., Byun, C., Gadepally, V., Houle, M., Hubbell, M., Jones, M., Klein, A., Michaleas, P., Milechin, L., Mullen, J., Rosa, A., Samsi, S., Yee, C., Reuther, A., and Kepner, J. (2019). Securing HPC using federated authentication. *CoRR*, abs/1908.07573.

Prout, A., Arcand, W., Bestor, D., Bergeron, B., Byun, C., Gadepally, V., Hubbell, M., Houle, M., Jones, M., Michaleas, P., Milechin, L., Mullen, J., Rosa, A., Samsi, S., Reuther, A., and Kepner, J. (2016). Enhancing HPC security with a user-based firewall. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–4.

Rossum, G. V. (1995). The Python Tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Scheerman, M., Zarrabi, N., Kruiten, M., Mogé, M., Voort, L., Langedijk, A., Schoonhoven, R., and Emery, T. (2021). Secure Platform for Processing Sensitive Data on Shared HPC Systems.

Sergeev, A. and Del Balso, M. (2018). Horovod: Fast and Easy Distributed Deep Learning in TensorFlow. *arXiv preprint arXiv:1802.05799*.

Shanley, T. (2002). *Infiniband*. Addison-Wesley Longman Publishing Co., Inc., USA.

Stallman, R. M. et al. (2009). *Using The Gnu Compiler Collection: A Gnu Manual For Gcc Version 4.3.3*. CreateSpace, Scotts Valley, CA.

Stocksdale, T., Chang, M.-T., Zheng, H., and Mueller, F. (2017). Architecting HBM as a High Bandwidth, High Capacity, Self-Managed Last-Level Cache. In *Proceedings of the 2nd Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems*, PDSW-DISCS '17, page 31–36, New York, NY, USA. Association for Computing Machinery.

Stoll, C. (1988). Stalking the Wily Hacker. *Communications of the ACM*, 31(5).

The Trilinos Project Team. The Trilinos Project Website.

Tsymbal, V. and Kurylev, A. (2021). Profiling Heterogeneous Computing Performance with VTune Profiler. In *International Workshop on OpenCL*, IWOCL'21, New York, NY, USA. Association for Computing Machinery.

U.S. Department of Justice (2009). Swedish national charged with hacking and theft of trade secrets related to alleged computer intrusions at nasa and cisco.

U.S. Department of Justice (2013). Pennsylvania Man Sentenced to 18 Months in Prison for Hacking into Multiple Computer Networks.

Valeria Barra, Jed Brown, Jeremy Thompson, and Yohann Dudouit (2020). High-performance operator evaluations with ease of use: libCEED's Python interface. In Meghann Agarwal, Chris Calloway, Dillon Niederhut, and David Shupe, editors, *Proceedings of the 19th Python in Science Conference*, pages 85 – 90.

Vasilache, N., Baskaran, M. M., Henretty, T., Meister, B., Langston, H., Tavarageri, S., and Lethin, R. (2014). A Tale of Three Runtimes. *CoRR*, abs/1409.1914.

Wan, S., Li, Y., Li, W., Mao, X., Wang, C., Dong, J., Nie, A., Xiang, J., Liu, Z., Zhu, W., and Zeng, H. (2019). *Non-volatile Ferroelectric Memory Effect in Ultrathin -In2Se3*, volume 29. Wiley Online Library Full Collection 2016.

Wilen, A. H., Schade, J. P., and Thornburg, R. (2003). *Introduction to PCI Express: A Hardware and Software Developer's Guide*. Intel Press.