Withdrawn Draft

Warning Notice

The attached draft document has been withdrawn and is provided solely for historical purposes. It has been followed by the document identified below.

Withdrawal Date	February 9, 2024
Original Release Date	February 6, 2023

The attached draft document is followed by:

Status	Final
Series/Number	NIST SP 800-223
Title	High-Performance Computing Security: Architecture, Threat Analysis, and Security Posture
Publication Date	February 2024
DOI	https://doi.org/10.6028/NIST.SP.800-223
CSRC URL	https://csrc.nist.gov/pubs/sp/800/223/final
Additional Information	

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication
NIST SP 800-223 ipd

High-Performance Computing (HPC) Security:

Architecture, Threat Analysis, and Security Posture

Initial Public Draft

Yang Guo
Ramaswamy Chandramouli
Lowell Wofford
Rickey Gregg
Gary Key
Antwan Clark
Catherine Hinton
Andrew Prout
Albert Reuther
Ryan Adamson
Aron Warren
Purushotham Bangalore
Erik Deumens
Csilla Farkas

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-223.ipd

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication
NIST SP 800-223 ipd

High-Performance Computing (HPC) Security:

Architecture, Threat Analysis, and Security Posture

Initial Public Draft

Yang Guo
Ramaswamy Chandramouli
Computer Security Division
Information Technology Laboratory

Lowell Wofford
Amazon.com, Inc.

Rickey Gregg
Gary Key
HPCMP

Antwan Clark
Laboratory for Physical Sciences

Catherine Hinton
Los Alamos National Laboratory

Andrew Prout
Albert Reuther
MIT Lincoln Laboratory

Ryan Adamson
Oak Ridge National Laboratory

Aron Warren
Sandia National Laboratories

Purushotham Bangalore
University of Alabama

Erik Deumens
University of Florida

Csilla Farkas
University of South Carolina

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-223.ipd

February 2023

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Authority

NIST Technical Series Policies

Publication History

Approved by the NIST Editorial Review Board on YYYY-MM-DD [will be add upon final publication]

How to Cite this NIST Technical Series Publication:

Guo Y, Chandramouli R, Wofford L, Gregg R, Key G, Clark A, Hinton C, Prout A, Reuther A, Adamson R, Warren A, Bangalore P, Deumens E, Farkas C (2023) High-Performance Computing (HPC) Security: Architecture, Threat Analysis, and Security Posture. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-223 ipd. https://doi.org/10.6028/NIST.SP.800-223.ipd

Author ORCID iDs

Yang Guo: 0000-0002-3245-3069
Ramaswamy Chandramouli: 0000-0002-7387-5858
Lowell Wofford: 0000-0003-1003-5090
Albert Reuther: 0000-0002-3168-3663
Aron Warren: 0000-0002-5090-2198
Purushotham Bangalore: 0000-0002-1098-9997

Erik Deumens: 0000-0002-7398-3090

Public Comment Period
February 6, 2023 – April 7, 2023

Submit Comments
sp800-223-comments@list.nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

Security is an essential component of high-performance computing (HPC). HPC systems often differ based on the evolution of their system designs, the applications they run, and the missions they support. An HPC system may also have its own unique security requirements, follow different security guidance, and require tailored security solutions. Their complexity and uniqueness impede the sharing of security solutions and knowledge. This NIST Special Publication aims to standardize and facilitate the information and knowledge-sharing of HPC security using an HPC system reference model and key components as the basics of an HPC system lexicon. This publication also analyzes HPC threats, considers current HPC security postures and challenges, and makes best-practice recommendations.

Keywords

high-performance computing (HPC); HPC security; HPC reference model; security guidance; HPC threat analysis; HPC security posture.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

    i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

    ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: sp800-223-comments@list.nist.gov

Table of Contents

List of Figures

1. Introduction

In 2015, Executive Order 13702 established the National Strategic Computing Initiative (NSCI) to maximize the benefits of high-performance computing (HPC) for economic competitiveness and scientific discovery. The ability to process large volumes of data and perform complex calculations at high speeds is a key part of the Nation's vision for maintaining its global competitive edge.

Security is an essential to achieving the anticipated benefits of HPC. HPC systems bear some resemblance to ordinary IT computing, which allows for the effective application of traditional IT security solutions. However, they also have significant differences. An HPC is designed to maximize performance so its architecture, hardware components, software stacks, and working environment are very different from ordinary IT. As such, security solutions must be tailored to the HPC system's requirements. Furthermore, HPC systems are often different from one another due to the evolution of their system designs, the applications they run, and the missions they support. An HPC system frequently has its own unique security requirements and follows different security guidance, which can impede the sharing of security solutions and knowledge.

This NIST Special Publication aims to standardize and facilitate the sharing of HPC security information and knowledge through the development of an HPC system reference model and key components, which are introduced as the basics of the HPC system lexicon. The reference model divides an HPC system into four function zones. A zone based on the HPC reference model captures the most common features across the majority of HPC systems and segues into HPC system threat analysis. Key HPC security characteristics and use requirements are laid out alongside the major threats faced by the system and individual function zones. HPC security postures, challenges, and recommendations are also included.

2. HPC System Reference Model and Main Components

The HPC system is complex and evolving so a common lexicon can help describe and identify an HPC system's architecture, critical elements, security threats, and potential risks. An HPC system is divided into four function zones:

1. The *high-performance computing zone* consists of a pool of compute nodes connected by one or more high-speed networks. The high-performance computing zone provides key services specifically designed to run parallel jobs at scale.
2. The *data storage zone* comprises one or multiple high-speed parallel file systems that provide data storage service for user data. The high-speed parallel file systems are designed to store very large data sets and provide fast access to data for reading and writing.
3. The *access zone* has one or more nodes that are connected to external networks, such as the broader organizational network or the internet. This zone provides the means for authenticating and authorizing the access and connections of users and administrators. The access zone provides various services, including interactive shells, web-based portals, data transfer, data visualization, and others.
4. The *management zone* comprises multiple management nodes and/or cloud service clusters through which HPC management services are provided. The management zone allows HPC system administrators to configure and manage the HPC system, including the configuration of compute nodes, storage, and networks, provisioning, identity management, auditing, system monitoring, and vulnerability assessment. It also offers, through a portal in the access zone, users an interface to acquire high-performance computing services and to configure access to data storage services. Various management software modules – from job scheduler, workflow management, and the Domain Name System (DNS) – run in the management zone.

The HPC system reference model is depicted in Figure 1.

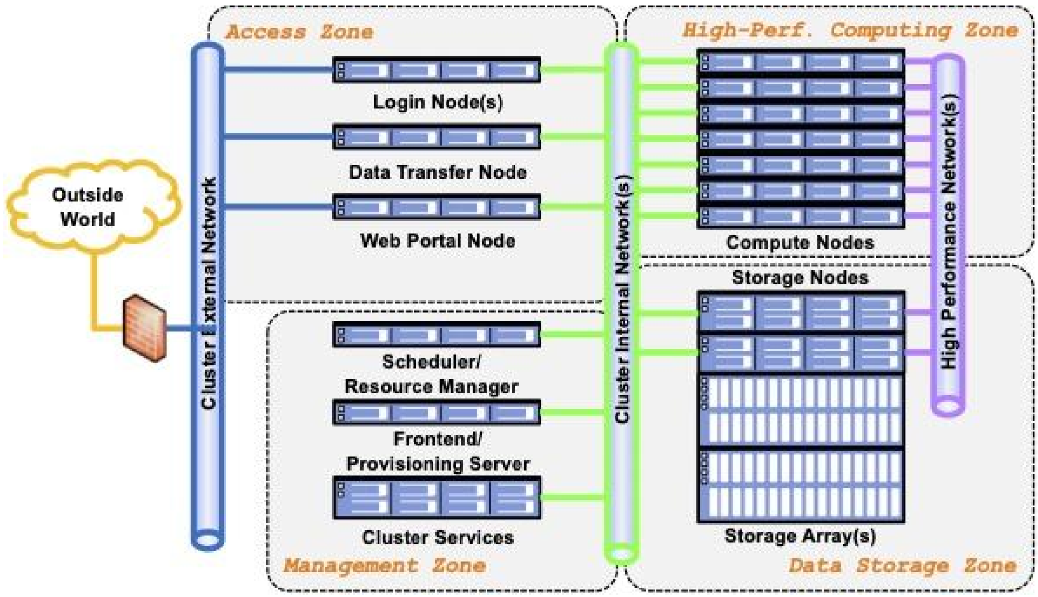

Fig. 1. HPC System Reference Model

2.1. Main Components

2.1.1. Components of the High-Performance Computing Zone

An HPC cluster consists of a collection of independent computing systems, called compute nodes, which are interconnected via high-speed networks. Compute nodes have the same components as a laptop or desktop, including central processing unit (CPU) cores, memory, disk space, and networking interface cards. However, they are equipped with a much larger number of CPU cores and more memory than a laptop or desktop. In some HPC architectures, a compute node may not have local disks and uses the data storage services of remote storage servers instead. In addition, there may be different types of nodes for different types of tasks, and some compute nodes are equipped with hardware accelerators to speed up specific applications. For instance, compute nodes often utilize graphics processing units (GPUs) [1] to accelerate modeling and simulation or AI and machine learning (ML) model training.

An HPC compute node installs its own software stack (e.g., operating system [OS], compilers, software libraries, etc.) to support applications. The installation and configuration of the software stacks are cluster-wide, centrally managed, and controlled by the management zone. The number of compute nodes in an HPC ranges from a few nodes to hundreds and even thousands of nodes.

A critical requirement of HPC networking that interconnects computer nodes is to have massive amounts of scalable bandwidth (throughput) while keeping the latency ultra-low so that the compute nodes and parallel file system (PFS) in the data storage zone can work as one supercomputer. HPC networking often employs specifically designed protocols, networking cards, processor nodes, and switches to optimize network performance. The popular HPC interconnect networking includes InfiniBand [2], Omni-Path [3], Slingshot [4], and others.

A high-performance computing zone typically utilizes non-high-performance communication networks, like ethernet, as cluster internal networks that connect the high-performance computing zone with the management zone and access zone for traffic associated with maintenance activities as opposed to HPC traffic.

2.1.2. Components of the Data Storage Zone

Several different classes of storage systems may be present inside of the data storage zone. In general, storage systems within this zone cannot be effectively separated from the HPC resources that they support from an administrative privilege perspective. Typical classes of storage found within this zone include parallel file systems (PFS), node-local storage for low-latency workloads, and archival file systems that support campaign storage and protect against data loss. HPC systems may have other file systems that store non-user data. For instance, the management zone often has its own file system that stores the OS images and configuration files. In that case, the file system is included in the corresponding function zone.

HPC applications' initial data, intermediate results, and final results are stored in the data storage zone and can be accessed during the application runtime and after the application's completion. External HPC users can also access user data through the login nodes and/or data transfer nodes in the access zone.

The storage capacity of these file systems is often measured in petabytes and can reach up to exabytes. File systems within the data storage zone will generally use a transport mechanism appropriate to the tier. For example, high-bandwidth file systems may be attached to the HPC resource's high-performance network, while lower bandwidth file systems may use 10 Gbps or 100 Gbps ethernet. Access control for most HPC file systems is enforced by the operating system software of nodes on which these file systems are mounted. As such, file systems should not be mounted outside of their security boundaries. A rogue system that can mount a file system will have complete control of all file system data, can spoof packets on the high-speed network, and can possibly gain privileges elsewhere within other zones of the HPC security enclave.

2.1.2.1. Parallel File System

Since HPC workloads can vary significantly, a parallel file system (PFS) is often required to support read-intensive and write-intensive applications with sequential and random-access patterns at speeds of up to terabytes per second. Commonly seen file systems include Lustre [5], GPFS [6], and IBM Spectrum Scale [7]. During procurement, a PFS will typically be designed to hit a particular aggregate bandwidth target rather than a capacity requirement. These PFS will typically consist of a cluster of systems to maintain metadata about files and locations as well as servers that act as storage targets. Clients that mount the file system will typically load the file system client software via a kernel module. Storage target servers will have backing storage arrays configured with dozens of disks in a redundant array of inexpensive disk (RAID) strategy.

Both GPFS and Lustre-based PFS are prone to performance degradation when a certain capacity threshold is reached. These file systems may be regularly pruned of unwanted files with a strategy decided by the file system administrators. Some deployments will sweep files older than a certain age, which forces HPC users to transfer job output to a longer-term file system, such as campaign storage. PFS tends to be somewhat unreliable depending on the types of activities being performed by running jobs and users. Because these are distributed file systems, file-system software must solve distributed locking of files to ensure deterministic file updates when multiple clients are writing to the same file at once. Additionally, PFS are susceptible to denial-of-service conditions even during legitimate user operations, such as listing a directory with millions of files or applications that perform poor file locking semantics.

2.1.2.2. Archival and Campaign Storage

Archival and campaign storage systems represent a class of storage that is more resilient to failure conditions than PFS and is often less expensive per GigaByte. These advantages come at the cost of bandwidth and an increased latency of data transfer. While PFS acts as a temporary short-term scratch file system, campaign storage supports the longer-term storage needs of a project over its life cycle. Finished data products that support scientific publications or other high-value datasets may also be stored in an archival file system. The retention time for data on a campaign storage file system is measured in years, while the retention of data within an archival storage system is measured in decades. Both campaign and archival storage systems might employ low-latency disks – such as solid state drives (SSD) or non-volatile memory express (NVMe) drives within a small tier of storage that acts as a cache – and are backed with cheaper, higher capacity media, such as spinning disk and/or tape media.

2.1.2.3. Burst Buffer

For applications that require extremely low latency or high-bandwidth memory-to-disk data transfer during runtime, intermediate storage layers that contain “burst buffers” have been incorporated as brokers to primarily mitigate the effects of input/output (I/O) contention and the bandwidth burden on parallel file system (PFS). These burst buffers can pre-fetch data from the parallel file system (PFS) before a computing job begins and stage data out to a parallel file system after a computing job has completed. This saves job runtime that would normally be spent performing bulk I/O to the PFS and allows it to be spent on computation instead. Typical HPC infrastructures contain the following intermediate storage architectures:

- **Node-local burst buffer architectures:** Each burst buffer is collocated with a corresponding HPC compute node [8]. This is advantageous for its scalability and also improves the checkpoint bandwidth because the aggregate bandwidth increases in proportion with the number of compute nodes.
- **Remote-shared burst buffer architectures:** Burst buffers are shared between multiple HPC compute nodes that are hosted on an I/O node [8]. This is advantageous for facilitating the development, deployment, and maintenance of these architectures.

There are also HPCs that can contain mixed burst buffer intermediate storage architectures, which are a mixture of the strengths of node-local and remote-shared burst buffer architectures.

2.1.3. Components of the Access Zone

A typical HPC system provides one or more nodes through which users and administrators access the system. At least one of these nodes is a login node where users have access to shells to launch interactive or batch jobs. Some of these login nodes may also have specialized visualization hardware and software with which users can conduct interactive and/or post-execution visualization of their datasets. There may also be one or more data transfer nodes that provide services to transfer data into and out of the HPC system and may even provide storage-mounting services like Network File System (NFS) [9], Common Internet File System (CIFS) [10], Server Message Block (SMB) [11], and Filesystem in Userspace (FUSE) [12] based SSH Filesystem (SSHFS) [13]. Many HPC systems now provide web portals via web portal nodes that enable a variety of web interfaces to HPC system services.

2.1.4. Components of the Management Zone

The complexity of HPC systems requires a significant infrastructure to operate and manage it, which is collectively referred to as the management zone. The management zone may consist of servers and network switches that enable various functions for operating the system with efficiency, effectiveness, and stability.

2.1.4.1. General Architecture and Characteristics

One important characteristic of the management zone is that it has a separate security posture because non-privileged users do not need to access the management servers or services in a direct way. Privileged users responsible for configuring, maintaining, and operating the HPC system access the management zone servers and switches through extra security controls. For example, from the public-facing login nodes, they may go through a bastion host that is typically located in the management zone, or they may establish a private virtual network (VPN) with separate authentication and authorization or other appropriate security controls to reach the management zone. All systems are configured on networks that are not routed beyond the perimeter of the HPC system so that only nodes like compute nodes and storage nodes can access the services.

The services provided by the management zone have clearly defined protocols and can be implemented as running on assigned hardware platforms or run as virtual machines on a dedicated set of hardware resources. The fact that the management zone has a clear and separate security posture helps with risk assessment and the selection of controls to secure the management zone and manage the risk.

2.1.4.2. Basic Services

The HPC resources inside of the computing and data storage zones need various services to operate. Examples include Domain Name Services (DNS) [14]; the Dynamic Host Configuration Protocol (DHCP) [15]; configuration definitions, authentication, and authorization services, such as those provided by an LDAP server [16]; and the Network Time Protocol (NTP) [17] for synchronization, log management, version-controlled repositories.

The management zone includes storage systems to store configuration data and node images, current versions, development and test versions, and historical versions. Storing logs from the entire HPC system is also part of the management zone as well as the servers to process the logs and alert administrators of events, problems, and incidents. Many of these services will be implemented with high availability and failover capabilities to avoid failure of the HPC resources. The network switches for the management network (ethernet) and the fast interconnects (e.g., InfiniBand, Omni-Path, Slingshot) are often managed as part of the management zone because non-privileged users do not need direct access to these resources.

2.1.4.3. Configuration Management

Automated configuration management is crucial to ensure the stable operation of complex systems like HPC. The systems that hold the configuration database and run the server to place configurations on compute nodes, storage servers, and network switches are part of the management zone. The nodes are subject to a regularly scheduled process to verify configuration and enforce consistency with what the configuration management nodes and databases specify.

Often, the configuration management systems in the management zone have an even more restricted security posture than the management zone as a whole, with a smaller number of privileged users having access.

2.1.4.4. HPC Scheduler and Workflow Management

Because of the distributed nature of HPC systems, requesting resources for given workloads is coordinated by a scheduler or workload manager, such as Slurm [18] and Kubernetes [19]. These services are run on servers in the management zone alongside the configurations and job logs. Non-privileged users access the scheduler through specific commands or an application programming interface (API). Access to the service is restricted to nodes within the HPC system perimeter. There may also be a web interface that provides a separate authenticated and authorized path for scheduling workloads, often within the strict constraints of certain application domains.

2.1.5. HPC Software

In addition to the management software that installs, boots, configures, and manages HPC-related systems, HPC application codes rely on several layers of scientific and performance-enhancing libraries. The layers of software that are available to users is referred to as the software stack. The lower layers of this stack are typically focused on performance and include compilers, communication libraries, and user-space interfaces to HPC hardware components. The middle layer includes performance tools, math libraries, and data or computation abstraction layers. The top of the stack consists of end-user science or production applications. Each software product within this stack may require certain versions or variants of other products and have many dependencies. For example, Hierarchical Data Format version 5 (HDF5) [20] is a scientific data formatting library with only seven dependencies, while Data Mining Classification and Regression Methods (rminer) [21] – an R-based data mining application – has 150 software dependencies. The full software stack can be split into three general categories that differ based on the maintainer: user software, facility software, and vendor software.

2.1.5.1. User Software

Often, the end users themselves best understand how to tune their software to the bespoke hardware of an HPC system to ensure sufficient performance for their workload. Users regularly modify and recompile their software to enhance performance, fix bugs, and adapt to changes in the underlying dependencies or kernel interfaces over time. The sharing of user-built software between teams may be common. User software that is widely used is often open source and, therefore, subject to open-source software supply chain concerns.

Continuous integration (CI) pipelines [22] and tests of scientific code on HPC platforms have recently become commonplace. Industry-standard identification of software weaknesses and the publication of Common Vulnerabilities and Exposures (CVEs) [23] is not routine, but the identification and remediation of performance regressions is generally a higher priority within the user community. There is a value-per-cycle trade-off for CI tests since cheaper cycles on commodity hardware may not expose bugs on much more expensive HPC resources. Complex test suites will eat into user allocations, and users and staff prefer that only a cardinal set of smoke tests run within user-developed testing pipelines on HPC systems.

2.1.5.2. Site-Provided Software and Vendor Software

Site staff and administrators generally build applications and libraries that are most likely to be used. Tools such as Conda [24], EasyBuild [25], and Spack [26] are often used to manage the complexity of software dependency resolution. Staff may also wrap compiler and job submission utilities with custom scripts to collect usage information about software libraries, I/O read and write patterns, or other system telemetry that is useful for decision making.

Vendor software includes low-level system tools to facilitate the running of other software. For instance, remote direct memory access, inter-node memory sharing, performance counters, temperature and power telemetry, and debugging are all vendor-provided software.

Users can choose specific versions of installed vendor and site-provided software libraries by manipulating environment variables. Tools such as wrapper scripts or module files are usually provided to help users find and choose which versions of installed software to use.

2.1.6. Container Usage in HPC

A container is a software package that contains an application's entire runtime environment. It consists of an application program and all of the dependencies, libraries, other binaries, and configuration files needed to run it. Containers provide self-contained, portable, and reproducible environments that abstract away the differences in OS distributions and underlying infrastructure. Containers make applications more portable and the deployment easier. For instance, containers allow users to use a package manager (e.g., apt [27] or yum [28]) to install software without changing anything on the host system or to run the latest software built for newer Linux OS versions.

Security is a major concern in deploying containers in HPC environments. Containers possess large attack surfaces due to the different underlying images, each of which can have vulnerabilities. In addition, securing the host is not enough to ensure protection. Container permissions and proper isolation are also necessary. Finally, monitoring containers can be difficult due to its dynamic nature.

2.2. HPC Architecture Variants

2.2.1. Diskless Booting HPC

A variant of HPC cluster design is diskless booting clusters in which the OS is not loaded from storage located on the node itself. Diskless nodes typically boot up, obtain an IP address via DHCP, and receive a kernel and partial OS over the network through a protocol, such as the Preboot eXecution Environment (PXE) [29] . To finish the boot process, the remaining OS is typically loaded over NFS or Internet Small Computer System Interface (ISCSI) [30] .

For scalability, some HPC clusters may implement a hierarchy design in which intermediate servers boot diskless but have local storage that is used to serve the OS to leaf nodes.

2.2.2. Virtual and Hybrid HPC Environments

Virtualization technologies can be used to recreate and maintain traditional HPC architectures that allow them to leverage the virtual infrastructure effectively while also providing additional scalability. For example, virtual machines (VMs) can be used to redistribute scientific applications across heterogeneous supercomputing architectures that can be bundled more easily without the need for certain special privileges. Virtual HPC (vHPC) architectures typically contain the following main components [31]:

- **Hypervisors:** These elements create and run VMs in vHPC architectures, which allows them to use multiple resources, such as memory and processing. The benefits to these elements include their speed, efficiency, flexibility, convenience, and portability.
- **vCenter Server Appliance (VCSA):** This element provides centralized management of all virtualized infrastructure with a single management interface that interacts, manages, and monitors the virtualization configuration, settings, and services.
- **Network virtualization and security platform:** This platform provides software-defined networking (SDN) management and protection capabilities to the vHPC infrastructure.

Some optional technologies include Kubernetes-based platforms and technologies that accelerate the deployment and management of applications and services, optimize systems and application environments, and securely connect and operate remotely. vHPC architectures also contain access, management, compute, and data storage zones, as highlighted in Figure 1, in which the management zones contain VMs that manage the vHPC environment. The compute zones contain virtual compute clusters that are responsible for HPC workloads, while the storage zones consist of virtualized PFS capabilities that store user data. Virtual compute clusters can contain low-latency burst buffer storage that is local to each node or shared across nodes.

Cloud technologies can also be employed to enhance traditional HPC infrastructures, which result in the creation of hybrid HPC environments that contain a combination of traditional and cloud HPC architectures [32]. These types of schemas typically use private cloud capabilities because they are more flexible, customizable, and manageable than using public cloud technologies. Recently, high-performance computing as a service (HPCaaS) has been adopted as a flexible solution that offers similar supercomputing capabilities on the cloud. Organizations can work with vendors or service providers to customize these architectures to their specific needs and maintain control by changing their capacities and architectures. HPCaaS is also cost-effective because these solutions can typically be purchased via pay-as-you-go subscriptions that are managed by their service providers.

3. HPC Threat Analysis

HPC poses unique security and privacy challenges, and collaboration and resource-sharing are integral. HPC workloads are often different from their traditional counterparts. For instance, scientific experiments frequently employ unique hardware, software, and configurations that may not be maintained or well-vetted or that present entirely new classes of vulnerabilities absent in more traditional environments. HPC can store large amounts of sensitive research data, personally identifiable information (PII), and intellectual property (IP) that need to be safeguarded. Finally, HPC data and computation are encumbered with a variety of different security and policy constraints. The solutions to protecting data, computation, and workflows must balance these trade-offs.

3.1. Key HPC Security Characteristics and Use Requirements

HPC systems possess some unique security characteristics and distinctive use requirements that differentiate themselves from average IT system:

- **Tussles between performance and security:** HPC users may consider security valuable only to the extent that it does not significantly slow down the HPC system and impede research. Ensuring the usability of security mechanisms with a tolerable performance penalty is therefore critical to adoption by the scientific HPC community.
- **Varying security requirements for different HPC applications:** Individual platforms, projects, and data may have significantly different security sensitivities and need to follow different security policies. An HPC may need to enforce multiple security policies simultaneously.
- **Limited resources for security tools:** Most HPC systems are designed to devote their resources to maximizing performance rather than acquiring and operating security tools.
- **Open-source software and self-developed research software:** Open-source software and self-developed research software are widely used in HPC. Open-source software is vulnerable to open-source software supply chain threats, while HPC software input data may also be vulnerable to data supply chain threats. Self-developed software is susceptible to low software quality.
- **Granular access control on databases:** Since different research groups may have a need to know for different portions of data, granular access control capabilities are necessary. Access control requirements may need to be dynamically adjusted as some scientific experiments may increase data needs based on the outcome of experiments.

3.2. Threats to HPC Function Zones

3.2.1. Access Zone Threats

The access zone provides an interface for external users to access the HPC system and oversees the authentication and authorization of users. Among the four function zones, the access zone is the only one that is directly connected to the external networks. Hence, the nodes and their software stacks in this zone are susceptible to external attacks, such as denial of service (DoS) attacks, perimeter network scanning and sniffing, authentication attacks (e.g., brute force login attempts and password guessing), user session hijacking, and machine-in-the-middle attacks. In addition, some nodes are subject to extra attacks due to their specific software stacks. For instance, a web server may be subject to website defacement, phishing, misconfiguration, and code injection attacks. The access zone also provides access to the file systems hosted in the data storage zone. It is important that permissions to directories and files are only given to authorized accesses.

Authenticated users sometimes use external networks to download data or code for use inside of the HPC system, which introduces the risk of unintentionally downloading malicious content. The nodes in the access zone are usually configured to support limited computation (e.g., modest debugging). Access zone nodes are susceptible to computational resource abuse.

The access zone is also shared by multiple users. One user's activities, such as commands issued and jobs submitted, can be viewed by other users. A port opened by one user can potentially be used by others. Fortunately, the nodes in the access zone work similarly to enterprise servers, and general IT security tools and measures are available to harden the zone.

3.2.2. Management Zone Threats

The management zone is responsible for managing the entire HPC system. It is connected to clustered internal networks through which other zones can be reached. It runs a plethora of system management, out-of-band hardware management, job scheduling, and workflow management software, all of which are susceptible to unique threats.

Processes running in the management zone, such as schedulers and data tiering/orchestration processes, act on behalf of users. These are privileged processes, and if they are spoofed, it can lead to privilege escalation, which is a distributed system-to-system trust problem. Due to the implied delegation of authority within distributed HPC and file systems, 'root' on a compute node may be, depending on configuration, equivalent to 'root' on all systems within the HPC zones. Only administrators with privileged access authorization are allowed to log into the management zone, where a privileged administrator logs into the access zone first and then logs into the management zone. A malicious user may attempt to log into the management zone.

The management zone may also be implemented as a service running on a cloud via virtualization technologies. In such cases, the risks associated with the cloud also apply to the management zone.

3.2.3. High-Performance Computing Zone Threats

The high-performance computing zone offers core computational functions in an HPC system. The computer nodes are shared by multiple users or tenants. The exploitation of multi-tenancy environments is a major threat (e.g., side-channel attacks, user data/program leakage, etc.). Other threat sources that often cause extreme resource consumption, performance degradation, or the outage of the HPC system entirely include accidental misconfiguration, software bugs introduced by user-developed software, and system abuse by running applications that are not aligned with the HPC mission. Container escape, side-channel attacks, and DoS can also be threats if virtualization technologies – such as containers – are used in the high-performance computing zone.

As a security mitigating technology, the applications in HPC are mostly run in the user space, except for system calls that must run in the kernel with elevated privilege. Accelerators, high-performance interconnects, special protocols, and direct memory access between nodes are commonly used in the high-performance computing zone. Some of these technologies may not be thoroughly tested, and their speed, novelty, and complexity can make monitoring and detecting suspicious activity difficult. Direct memory access and communication between nodes may bypass the kernel and the protections provided by the kernel (e.g., Security-Enhanced Linux [SELinux] [33]) is lost.

3.2.4. Data Storage Zone Threats

Protecting the confidentiality and integrity of user data is essential for the data storage zone. Data integrity can be compromised by malicious data deletion, corruption, pollution, or false data injection so gaining unauthorized privileged access is a major threat. Legitimate users may also mishandle sensitive data, leading to confidentiality breakdown. File metadata (e.g., file name, author, size, creation date) can also leak sensitive information about the files.

HPC file systems in the data storage zone provide superior data access speed and much larger storage size compared to average enterprise file systems. Hard disk failure is a threat due to the large number of disks deployed in the data storage zone. Incident response and contingency planning controls may not be easy to implement, and file backup, recovery, and forensic imaging may become infeasible. The security measures that can be implemented on the enterprise file systems may take an unacceptably long time and degrade HPC file system performance in an unacceptable way.

Providing data backup services is another challenge in HPC due to its large volume. By default, user data is often not backed up, and users are responsible for maintaining their own data copies. Inadvertent operations (e.g., accidently deleting a file subdirectory) can cause the permanent loss of data, though making data READ ONLY is one way to combat such a risk. Some organizations offer backup services using their own HPC systems, but these systems may be in the same geographic locations and subject to the same environmental threats.

3.3. Other Threats

In addition to the threats unique to individual function zones, the general HPC systems face the following threats:

- **Environmental and physical threats:** Physical or cyber attacks against facilities (e.g., power, cooling, water), unauthorized physical access, and natural disaster (e.g., fire, flood, earthquake, hurricane, etc.) are all potential threats to an HPC system.

- **Vulnerabilities introduced by prioritizing performance in HPC design and operation:** HPC is designed to process large volumes of data and perform complex computations at very high speeds. Achieving the highest performance possible is a priority in HPC design and operation. Such prioritization, however, has its security implications. For instance, designers often make conscious decisions to build a less redundant system to achieve high performance, making the system less robust and potentially more vulnerable to attacks, such as DoS attacks. As another example, using a backup system to improve system robustness and high availability is a proven technology. Building a backup HPC system, however, is often infeasible since it is too costly. HPC often lacks storage backup due to the vast size of the data stored. Similarly, most HPC missions do not have a service-level agreement to justify the need for a full backup system at a backup location. All resources are poured into building the single best HPC system possible.

- **Supply chain threats:** The HPC supply chain faces a variety of threats, from the theft of proprietary information to attacks on critical hardware components and software manipulation to gain unauthorized access. Some HPC software (e.g., OS, BIOS, applications), firmware, and hardware components have limited manufacturers, suppliers, and integrators, which make diversification difficult. Limited suppliers also lead to shortages in the qualified workforce who can perform required technical support.
- **Insider threats:** Insider threats come from people within the organization who have internal information and may have the privileges needed to access the HPC system. Insider threats can be classified into accidental/unintentional threats and malicious/intentional threats. Unintentional threats come from the unintended side effects of normal actions and activity. In contrast, a malicious insider may intentionally upload malicious code into the HPC system.

4. HPC Security Posture, Challenges, and Recommendations

4.1. HPC Access Control via Multiple Physical Networks

Access control is a security technique that regulates who can access and/or use resources in a computing environment. In HPC, multiple physical networks are constructed as an effective means of access control:

- **Management network:** The management network is a dedicated network that allows system administrators to remotely control, monitor, and configure computer nodes in an HPC system. Modern computers are often equipped with the Intelligent Platform Management Interface (IPMI) [34], which provides management and monitoring capabilities that are independent of the host system's CPU, firmware (e.g., BIOS [35]or Unified Extensible Firmware Interface [UEFI] [36]), and operating system. For example, IPMI allows system administrators to remotely power on/off unresponsive machines and install custom operating systems. IPMIs are connected to the management network, which can only be accessed by authorized system administrators.
- **High-performance networks:** High-performance networks offer high bandwidth and low latency to connect computer nodes inside of the high-performance zone and data storage zone. They also support features that are unique to HPC, such as remote memory access over the network and the message passing interface (MPI) [37]. High-performance networks often use special communications standards and architectures to achieve the high performance (e.g., InfiniBand, Slingshot, Omni-Path, etc.).
- **Auxiliary networks:** Additional auxiliary networks can be added to support usability and system manageability. For instance, a user network is constructed to allow users to manipulate or share data or remotely log into and access the compute nodes. Depending on the purpose of the networks, a subset of nodes from different zones are selected to be party to the networks. As an example, a user network contains the nodes in the access zone and the computer nodes in the high-performance computing zone.

There are many benefits to having multiple networks in an HPC system. First, all of these networks are private and use different IP address ranges. Network traffic will remain on one network, which facilitates monitoring and measurement. Second, individual networks often serve specific purposes so only the relevant nodes are connected to the network. The networks effectively segment the HPC system into smaller segments, which improves security. Finally, multiple physical networks also provide a degree of fault tolerance. When one network goes down, the system administrator can use the other network to diagnose and troubleshoot.

The compute nodes in the access zone are connected to the external network and assigned public IP addresses, which allow users to remotely access the HPC systems. The user data can be shared through the login nodes or the data transfer nodes. Some systems allow storage to be exported using CIFS [10] or SMB [11] (e.g. via a SAMBA [38] server). If necessary, a network address translation (NAT) [39] or a Squid proxy [40] can be installed to allow users on a private network to access the internet and download new versions of software or share software data. However, a NAT and Squid proxy can also be security weaknesses that demand extra caution and mitigation considerations.

Employing multiple physical networks is a common and effective means for access control and fault tolerance, and it is highly recommended.

4.2. Compute Node Sanitization

High-performance compute nodes are used by multiple tenants and projects. At the end of a task run, the previous project may leave behind a residual "footprint," such as the data in memory and GPUs. It is important to sanitize the compute node so that data from previous jobs are not accidently leaked and the new job can start with a clean slate.

The common practice of compute node sanitization includes:

- Conducting a node health check at the end of a job
- Removing a node or forcing a reboot if a node is deemed "unhealthy"
- Working with hardware and software vendors to provide management hooks to sanitize the GPU
- Resetting GPUs to remove residual data between jobs
- Validating and checking firmware
- Rebooting nodes after the completion of a job at the OS level to remove accumulated residuals and ensure a consistent node state
- Checking critical files to ensure that they have not been changed

Compute node sanitization is highly recommended as the compute nodes are equipped with sophisticated hardware accelerators.

4.3. Data Integrity Protection

Data encryption is an effective means of providing data integrity. HPC data storage systems typically support uniform encryption at the file level or block level. Such data encryption at the file system level protects data from unauthorized access. However, it does not provide granular access (i.e., segmenting one user from others). Granular access can be achieved using user-level or group-level encryption. Even a system administrator cannot access a user's data with granular access. Additionally, file systems do not authorize users. Rather, users access the file system via the HPC access zone, which is responsible for authenticating the users and their access rights.

Hashing is another technique for protecting data integrity. Data files can be hashed at the beginning to acquire hashing keys. A file is not modified if its hashing key remains the same. Parallel file systems maintain many types of metadata (e.g., user ID, group ID, modification time, sh54 of file, etc.). Hashing metadata is also another way to check whether a file has been modified.

Periodically scanning file systems for malware is a proven technique for ensuring data integrity. However, scanning an HPC system for malware is challenging. HPC data storage can easily contain a petabyte or more of data. Existing malware scanning tools are only designed for a single machine or laptop using one thread. They are not efficient or fast enough to scan large

HPC data storage systems. Furthermore, the scanning operation can adversely affect the performance of running jobs.

Protecting data integrity is vital to the HPC security. Granular data access provides the best protection and is highly recommended when possible.

4.4. Securing Containers

Containers bundle an application's code, related libraries, configuration files, and required dependencies to allow the applications to run seamlessly across environments. Containers provide the benefits of portability, reproducibility, and productivity, but they can hide software. In a well-managed HPC system, a lot of software has already been installed as a baseline system environment. The applications developed using native libraries often run faster than a container. Hence, training users to develop programs in the HPC programming environment is one way to reduce exposure to container vulnerabilities.

Using containers in an HPC is a balancing act. When containers are supported, restrictions are recommended to mitigate vulnerabilities. This may include prohibiting independent network stacks or user namespaces (i.e., container namespace is always mapped to the host user account). These measures can ensure that the risks posed by containers are no different from other workloads. The threats of a container only affect the applications run by the same user. Other security measures include selecting a container runtime that does not require root access to the system and prohibiting container development in the HPC.

4.5. Achieving Security While Maintaining HPC Performance

HPC security measures often come with an undesirable performance penalty. The following are several effective ways to balance performance and security:

- Conduct tests to measure the performance penalty of security tools, which can be benchmarked to determine whether they are acceptable. Testing and measurement would also encourage more performance-aware tool design.
- Incorporate security requirements in the initial HPC design rather than as an afterthought. For instance, independent add-on security tools tend to have more impact on performance than native security measures that come with the HPC software stack.
- Avoid "one size fits all" security. Differentiate the types of nodes in the HPC system and apply appropriate security rules and controls to different node types. For instance, classify the nodes in HPC into three categories: external nodes, internal nodes, and backend nodes. Apply individual security controls to each node category. Such a differentiation also mitigates performance impacts.

4.6. Challenges to HPC Security Tools

Many industrial security tools are designed with stand-alone devices in mind (e.g., laptops, desktops, or mobile devices). HPC is a large-scale, complex system with strict performance requirements. Security tools that are effective for individual devices may not work well in an HPC environment. For example, a forensic tool that aids the recovery and preservation of a hard drive and memory for a single server works well in practice. It is unreasonable, however, to install forensic tools on all compute and storage nodes in an HPC system. As another example, HPC nodes may use remotely mounted storage, which may disable some security tools.

Moreover, different HPC applications may require different tools. Security tool vendors are often not accustomed to HPC use cases and requirements, which forces HPC security teams to develop analogous tools that may introduce new security vulnerabilities and are sometimes not accepted by organizations. The HPC community needs to work closely with security tool vendors to address these challenges.

A Security Technical Implementation Guide (STIG) [41] is a configuration standard and offers a security baseline that reflects security guidance requirements. The security checking tool can measure how well the STIG is satisfied. However, available STIGs are typically written for servers or desktops rather than for HPC. In addition, security baseline checking tools developed for commodity operating systems and applications require customizations to run on HPC. The Lawrence Livermore National Laboratory [42], Sandia National Laboratories [43], and the Los Alamos National Lab [44] have collaborated with the Defense Information Systems Agency (DISA) [45] to develop TOSS 4 STIG [46], which is geared toward HPC systems. Still, a more general STIG library and corresponding security checking tools are desirable to handle diverse subsystems and components inside an HPC.

5. Conclusions

Securing HPC systems is challenging due to their size; performance requirements; diverse and complex hardware, software, and applications; varying security requirements; and the nature of shared resources. The security tools suitable for HPC are inadequate, and current standards and guidelines on HPC security best practices are lacking. The continuous evolution of HPC systems makes the task of securing them even more difficult.

This Special Publication aims to standardize and facilitate the information and knowledge-sharing of HPC security. A zone based HPC system reference model is introduced to serve as a foundation for a system lexicon and captures common features across the majority of HPC systems. HPC system threat analysis is discussed, security postures and challenges are considered, and recommendations are made.

References

[1] "What Is a GPU?," [Online]. Available: https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html.

[2] R. Sheldon, "InfiniBand," [Online]. Available: https://www.techtarget.com/searchstorage/definition/InfiniBand.

[3] "Omni-Path," [Online]. Available: https://en.wikipedia.org/wiki/Omni-Path#:~:text=Omni%2DPath%20Architecture%20(OPA),this%20architecture%20for%20exascale%20computing..

[4] "HPE Slingshot interconnect," [Online]. Available: https://www.hpe.com/us/en/compute/hpc/slingshot-interconnect.html.

[5] "Lustre file system," [Online]. Available: https://www.lustre.org/.

[6] "Introducing General Parallel File System," [Online]. Available: https://www.ibm.com/docs/en/gpfs/4.1.0.4?topic=guide-introducing-general-parallel-file-system.

[7] "IBM Spectrum Scale," [Online]. Available: https://www.ibm.com/products/spectrum-scale.

[8] B. W. S. a. J. B. Lei Cao, "To share or not to share: comparing burst buffer architectures," in *Proceedings of the 25th High Performance Computing Symposium* , Virginia Beach, VA, USA, April 23 - 26, 2017.

[9] "Network File System," [Online]. Available: https://en.wikipedia.org/wiki/Network_File_System.

[10] "CIFS," [Online]. Available: https://cifs.com/.

[11] "Server Message Block Overview," [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831795(v=ws.11).

[12] "FUSE," [Online]. Available: https://www.kernel.org/doc/html/latest/filesystems/fuse.html.

[13] "SSHFS Homepage," [Online]. Available: https://github.com/libfuse/sshfs.

[14] "Domain Name Services," [Online]. Available: https://www.iana.org/domains.

[15] "DHCP," [Online]. Available: https://www.techtarget.com/searchnetworking/definition/DHCP.

[16] "LDAP (Lightweight Directory Access Protocol)," [Online]. Available: https://www.techtarget.com/searchmobilecomputing/definition/LDAP.

[17] "What is NTP?," [Online]. Available: http://www.ntp.org/ntpfaq/NTP-s-def.htm.

[18] "SLURM Workload Manager," [Online]. Available: https://slurm.schedmd.com/documentation.html.

[19] "Kubernetes," [Online]. Available: https://kubernetes.io/.

[20] "HDF5," [Online]. Available: https://www.hdfgroup.org/solutions/hdf5/.

[21] "rminer: Data Mining Classification and Regression Methods," [Online]. Available: https://cran.r-project.org/web/packages/rminer/index.html.

[22] "What is a CI/CD pipeline?," [Online]. Available: https://www.redhat.com/en/topics/devops/what-cicd-pipeline#:~:text=A%20continuous%20integration%20and%20continuous,development%20life%20cycle%20via%20automation..

[23] "CVE," [Online]. Available: https://cve.mitre.org/.

[24] "Conda," [Online]. Available: https://docs.conda.io/en/latest/.

[25] "EasyBuild: building software with ease," [Online]. Available: https://easybuild.io/.

[26] "Spack," [Online]. Available: https://spack.io/.

[27] "Package Management," [Online]. Available: https://ubuntu.com/server/docs/package-management#:~:text=Apt,upgrading%20the%20entire%20Ubuntu%20system..

[28] "Yum," [Online]. Available: https://en.wikipedia.org/wiki/Yum_(software).

[29] "PXE," [Online]. Available: https://www.techtarget.com/searchnetworking/definition/Preboot-Execution-Environment.

[30] "iSCSI (Internet Small Computer System Interface)," [Online]. Available: https://www.techtarget.com/searchstorage/definition/iSCSI.

[31] "Virtualizing High Performance Computing (HPC) Environments: Dell EMC Reference Architecture," May 2020. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/VMware-HPC-Virtualized-DTC.pdf.

[32] HPCWire, "Hybrid HPC: The time to embrace the cloud is now," 2020. [Online]. Available: https://www.hpcwire.com/2020/08/31/hybrid-hpc-the-time-to-embrace-the-cloud-is-now/.

[33] "What is SELinux?," [Online]. Available: https://www.redhat.com/en/topics/linux/what-is-selinux.

[34] "Intelligent Platform Management Interface," [Online]. Available: https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface.

[35] "BIOS," [Online]. Available: https://en.wikipedia.org/wiki/BIOS.

[36] "UEFI," [Online]. Available: https://uefi.org/.

[37] "MPI Forum," [Online]. Available: https://www.mpi-forum.org/.

[38] "How to share files with Samba," [Online]. Available: https://www.redhat.com/sysadmin/samba-file-sharing.

[39] "Network address translation," [Online]. Available: https://en.wikipedia.org/wiki/Network_address_translation.

[40] "Squid-cache.org," [Online]. Available: http://www.squid-cache.org/.

[41] "Security Technical Implementation Guides (STIGs)," [Online]. Available: https://public.cyber.mil/stigs/.

[42] "Lawrence Livermore National Laboratory," [Online]. Available: https://www.llnl.gov/.

[43] "Sandia National Laboratories," [Online]. Available: https://www.sandia.gov/.

[44] "Los Alamos National Lab," [Online]. Available: https://www.lanl.gov/.

[45] "Defense Information Systems Agency," [Online]. Available: https://disa.mil/.

[46] "DISA releases the TOSS 4 Security Technical Implementation Guide," [Online]. Available: https://public.cyber.mil/announcement/disa-releases-the-toss-4-security-technical-implementation-guide/.