Journal of Distributed Computing and Systems(JDCS)
Vol.6, Issue 12, 2024, Pages 12-24

Enhancing High-Performance Computing (HPC) Security: A Comprehensive Review of Detection and Protection Strategies

Sara Koleini[1], Bahareh Pahlevanzadeh[2]
[1]*Department of Hardware and infrastructure Islamic World Science & Technology Monitoring and Citation Institute (ISC)* Shiraz, IRAN , *Corresponding Author*
[2]*School of Information and Cybersecurity Technology University Dublin* Dublin, Ireland

Article History:
Received: 25 Feb 2024
Accepted: 10 March 2024
Available online: 20 March 2024

Abstract

The escalating demand for High-Performance Computing (HPC) systems and data analysis across diverse scientific domains has amplified network security issues that need to be addressed urgently. This study provides an in-depth exploration of security challenges, prevalent threats, and exploitable vulnerabilities in HPC systems. In the pursuit of fortifying High-Performance Computing (HPC) systems, the methodologies delineated in this study hold potential applicability. These strategies, which can be systematically classified into detection and protection categories, are designed to counteract a diverse range of vulnerabilities and threats inherent in HPC systems. This paper offers a comprehensive review and summarization of these strategies, encapsulated within a taxonomy diagram. Subsequent sections provide an in-depth exploration of some of the major subjects within this taxonomy. The paper presents two major approaches for software fault detection in HPC systems: static analysis and dynamic analysis. The paper also discusses the use of different monitoring systems by HPC system administrators to identify and stop malicious activities. The paper highlights that most software threats and insider attacks that aim to compromise the Confidentiality, Integrity, and Availability (CIA) of HPC systems can be detected by monitoring. It mentions DPEM and VARAN as examples of monitoring systems developed to combat runtime attacks and provide low performance overhead suitable for HPC systems, respectively. This paper discusses the protection strategies for HPC systems. It emphasizes the importance of Access Control, Randomization, Control Flow Integrity, Multi-Execution, and Fault Tolerance. In addition to the challenges of traditional HPC systems, the paper discusses some issues in cloud-based HPC systems, such as virtualization overhead and multi-tenancy. This paper also explores the most recent and relevant research for enhancing HPC security from software and hardware perspectives, and summarizes some important and outstanding case studies conducted in different countries /regions by focusing on HPC security in recent years. By offering a detailed taxonomy and a robust security management model, this study aims to empower researchers and system administrators with the knowledge and tools necessary to safeguard their HPC systems and the sensitive data they process.

Keywords: Attacks, Detection and Protection Mechanisms**,** High-Performance Computing, Security, Vulnerabilities, Threats, HPCaaS

I. INTRODUCTION

Due to the strategic importance of high-performance computing, it has been one of the most active research areas in both computer science and engineering over the past half-decade. As the adoption of HPC technology increases, there is also a rise in concerns about the security aspects of this technology[1]. Security problems in HPC systems are inherited from computers and network security issues. HPC systems have a valuable large-scale computing infrastructure [2] that needs to be carefully guarded and avoid being maliciously used. In an HPC system, a large number of concurrent thread executions occurred in an enormously parallel system. As illustrated in "Fig.1", the main components of each HPC system can be categorized as below:

- Advanced processors with as many CPU/GPU cores are recognized as compute nodes.
- High-speed communication network (Ethernet, Omni-Path, InfiniBand) for connecting all the hardware and software components.
- Task scheduler, either batch or interactive jobs with a different size run on a reliable and secure platform.
- Storage nodes include fast non-volatile memory and archival storage.

According to this architecture, the security for each level should be considered. To use HPC, registered users (HPC users) from various research centers can access this system via login nodes and submit their jobs to the service scheduler. The user's submitted job will be assigned to the number of compute nodes, and then the compute nodes will start running the job until the job is finished. All I/O data is stored in the storage nodes. It is essential to consider the high-speed computing platform which is a high-bandwidth network between nodes for efficient data transmission. Therefore, it is required to have security protection mechanisms to prevent suspicious activity or attempts to access and destroy the data at each level. In other words, valuable computing resources of the HPC system need to be secured and avoid being maliciously used. Also, the considered security mechanisms should be light enough to avoid reducing the computing speed of HPC systems in service nodes [3].

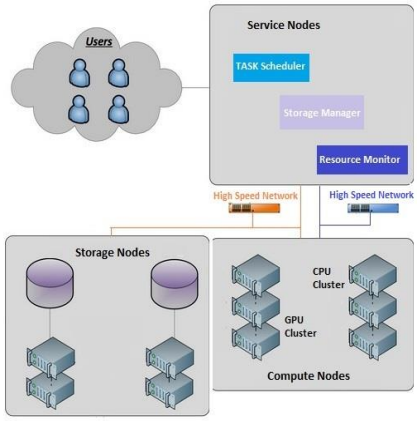

Fig. 1. HPC Systems' Architecture

Vulnerabilities are essential issues in HPC systems [4]. The main objectives of this paper are as follows:

- Discuss the HPC systems, applications, system infrastructure, and its characteristics
- Discuss the HPC challenges and propose a conceptual model to secure these systems.
- Suggest the key improvements after evaluating the case studies for HPC systems.

The rest of the paper is organized as follows: Section 2 presents the HPC applications, characteristics, and system infrastructure. Section 3 explores the existing challenges of these systems' threats, and vulnerabilities. Section 4 discusses a taxonomy of HPC vulnerabilities and the security strategies and solutions. Section 5 discusses the case studies from different regions and solutions. The paper concludes in the last section with possible future directions.

II. HPC Applications, Infrastructure And Characteristics

Different partners can incorporate the HPC for operational issues like ecosystems, such as academic computing, public research and information centers, Health information, industry, Research and Development (R&D), for example, climate forecast, air traffic signal, and transportation. "Fig. 2" shows the restrictions on operator usage by the regulatory burden of sensitive information. There are some limitations in HPC systems where the operators rely on the regulatory environment and valid obligations exist to provide confidential information with greater protection. Moreover, HPC systems may support basic and critical infrastructure with prerequisites for consistent availability. The operator restrictions might be more noteworthy. Generally, efficiency and convenience directly trade-off with security, with the thought being that security regularly profits from user restrictions. The regulatory burden of sensitive information imposes restrictions on operator usage. HPC capabilities used for operational purposes can also have extremely limited operating environments in which the information itself is not sensitive [5].

Traditionally, HPC in an academic environment has enormous computing power with domain scientists, programmers, and engineers composing code on bare metal. These systems are used to transfer the data and to perform fast parallel and repetitive mathematical calculations and numerical simulations for scientific problems. In other words, with the use of hardware accelerators and coprocessors for large quantities of parallel processing, HPC functionality is based on hybrid computing technology.

These systems have massive storage capacity, which allows users to store large volumes of data in the era of big data [6][7][8][9][10][11][12][13]. The complexities of HPC systems have increased with big data and HPC convergence, system heterogeneity, cloud computing, and many other developments. Data and information are critical and should be treated as an asset and need to be appropriately secured. Generally, due to the similarity of other distributed systems such as grid, cluster, and cloud computing with HPC systems in infrastructures and characteristics; many of the challenges posed by them are similar to those faced in HPC systems.

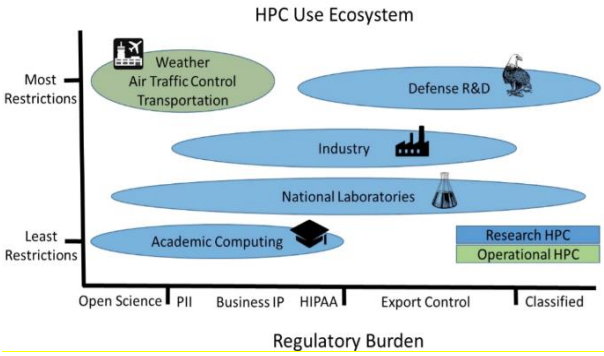

Fig. 2. Restrictions on Operator usage are Imposed by the Regulatory burden of Sensitive Information [4]

Moreover, with the spread of cloud computing, many HPC systems now serve through cloud technology. Using the cloud, powerful computing capabilities are placed in the hands of researchers, engineers, and organizations that do not have access to internal infrastructure. Cloud technology is also used when more HPC resources are needed and cannot be provided in the organization. Cloud flexibility and scalability create almost unlimited capacity, eliminating long job queues and waiting times and making it easy to access new services and apps needed by HPC management software. Cloud service models are categorized into three major service models specifically, Infrastructure-as-a-Service or IaaS (compute, storage, network, resources, etc.), Platform-as-a-Service or PaaS (OS, APIs, Development environments, etc.), and the Software-as-a-Service or SaaS (applications, software systems, etc.). IaaS and PaaS can be compared with HPC systems for resource provision and resources needed for scientific computations [14].

Since the implementation of HPC in the cloud, a new cloud service named HPC as a Service (HPCaaS) has been developed. HPCaaS provides high-level processing capacity for cloud users. With HPCaaS, several applications get permission to run simultaneously on the same system; therefore, system flexibility is increased and allows dynamic resource allocation when needed [15]. Thereby, it is necessary to consider security solutions such as policies, controls, procedures, and technologies that each platform needs to be more secure; therefore, those solutions can be transferred into the HPC systems [15][16].

III. HPC Systems Challenges

Generally, the specific vulnerabilities of traditional HPC systems can be traced back to the following three aspects; high bandwidth connections, extensive computational power, and massive storage capacity. Some unique aspects of HPC systems that make it secure. The HPC security challenges are as follows [5]:

- The scale of implication.
- On compute nodes, security can be eased for performance reasons.
- It is crucially significant to consider integrity.
- It is costly to build and test systems at the same time.
- It is challenging to maintain due to the fast and radically expanding environment and technology, and demands
- There is a large number of components and files to check.
- It is difficult to control the potential distribution nature of components' malfunction.

The HPC security is different and complex compared to traditional data center infrastructure. The design of HPC systems often tends to have very distinctive modes of operation. HPC is used for very distinctive purposes, particularly mathematical computations, runs extremely exotic hardware and software stacks, and is extremely accessible and open to users. Thus, its security issues are quite challenging in comparison with other technologies. This distinction, in other words, poses both opportunities and challenges; hence, there is no clearly articulated security strategy or policies defined globally expressing the threat that security measures are meant to avoid [5]. Moreover, it cannot use certain security solutions, such as network firewalls, as traditional infrastructures. In other words, concerning Confidentiality, Integrity, and Availability (CIA), the HPC system dedicated to basic research systems which has potentially faced low impact. In contrast, HPC systems focused on the industrial sector, engineering, or operational prediction, such as military vehicle design or weather prediction, have potentially high consequences. Thus, for research with high technology and innovation readiness levels, the necessities for validation and verification of models and code are much greater [5]. For instance, some science data (such as sensor data or distributed and or streaming data collection) is getting to us in new ways; thus, we have more data to protect.

In addition to the challenges of traditional HPC systems, there are some issues in cloud-based HPC systems. The two most important HPCaaS challenges can be explained as follows:

- ***Virtualization Overhead:*** Using virtualization in the cloud provides pools of resources, fast elasticity, and more flexibility. However, virtualization produces unwanted overhead when adding another software layer to the system and avoiding direct access to hardware by applications.
- ***Multi-Tenancy:*** The Multi-tenancy feature enables the cloud environment to share resources between multiple tenants to maximize benefits. However, HPC applications are more secure if they have direct access to devoted hardware [17].

Furthermore, the growing interest in using Artificial Intelligence (AI) techniques to solve "real world" problems necessitates the use of HPC tools to effectively compute and scale complex algorithms across thousands of nodes [3]. However, most data scientists are unfamiliar with the features of the HPC environment. Thus, deploying AI software on a secure HPC requires a huge amount of computing resources and a connection to external systems (for downloading varieties of open-source software with root privilege); that is considered a new challenging issue in HPC security.

IV. HPC Systems Attacks and Vulnerabilities

In comparison to a general computing facility, HPC architectures are highly specialized. As a result, they have unique security problems, vulnerabilities, and assets. In general, it can be divided the threats confronted by HPC into three main categories according to the CIA as the three main high-level cybersecurity objectives for security are as follows [6][1]:

- ***Confidentiality:*** Ensuring confidentiality means constraints on access to or access to sensitive information disclosure. The attacks that compromise confidentiality are data leakages and intrusion attacks due to open access to the resources in HPC systems.
- ***Integrity:*** Integrity lookouts against unintentional or malicious data or execution modification of flow. A minimal change in data or execution flow in HPC can lead to a serious impact. The attacks that compromise integrity are alterations of data or code.
- ***Availability:*** Availability means making sure that users can quickly access device resources reliably. The attack that compromises availabilities are disruption and denial-of-service attacks against HPC systems or networks that connect them.

As mentioned before, vulnerabilities are key problems in HPC systems because most of the jobs and resources run or stored are usually sensitive and high-profit data. The common issues affecting HPC facilities are probes, scans, brute-force login attempts, and buffer overflow vulnerabilities. According to a study [18], the most common threats discussed specifically in cloud-based HPC systems include data loss or leakage threats, data breaches, insecure application program interface, malicious insiders, and Denial of Service (DoS) or Distributed DoS (DDoS) for diminishing the resources and reputation of victims. In addition, account or service hijacking/identity theft insufficient due diligence, shared technology vulnerabilities, weak identity and access management, system and application vulnerabilities are also considered the vulnerabilities of systems. Advanced Persistent Threats (APT), abuse of cloud service, threats in the virtual machine, data theft, data security and privacy issues, trust issues, Service Legal Agreement (SLA)/Legal issues, threats with cloud service providers, availability and reliability problems, authentication and authorization, man-in-the-middle attack, spoofing, injection attack, loss of control, malware, phishing, backdoor, and social engineering are few more examples of system vulnerabilities. The researchers have precisely emphasized specific governess about APT in HPC systems.

Normally, insiders or outsiders of the HPC system can be the source of HPC attacks. In terms of HPC security, insiders (e.g., system administrators and user support staff) also have additional privileges because they can access the HPC infrastructure and resources. All security aspects of an HPC device can be compromised by an inside attack [5]. Therefore, on all the devices and networks they run, the account holders of these accounts need to take special care. Only after enabling a Virtual Private Network (VPN) should they access their privileged account from remote machines or an untrusted VPN. It is therefore much easier for insiders than others to launch attacks. The common method in which the HPC systems are compromised is by either user not securing their passwords or easy-to-use passwords like 'test123' or ‘zaqxsw’. It should be remembered that the behavior of hackers is usually entirely different from that of the account owner. Good indicators of potential compromise include a sudden burst of network operation, increased network latency, overloading the system with CPU consumption, unauthorized jobs, and bypassing the job scheduler[16][18].

The brute-force attack of the password scheme and man-in-the-middle attacks are two more sophisticated types of attacks most preferred by outsiders [19]. In other terms, hackers are trying to hijack the endpoints. Experienced hackers often try to hide their tracks and are normally capable of obfuscating their behaviors. They continue their exploit by installing rootkits, changing the RPM package repository, turning off or maneuvering the monitoring software and log files. Phishing is another type of attack that compromises users’ credentials. Users who use popular websites for social networking are typically susceptible to this type of attack; they invite hackers from social networking sites to HPC systems to use the same password for their HPC accounts. Intercepted e-mails are another source of compromise. Daemon process-based attack, interposition library attack, and prob-based login attack are some other attacks introduced by researchers [1][20]. Besides the examples of attacks mentioned above, there are several other threats to HPC systems [1] [18] [21]. One of the major threats to HPCs that should be considered is escalation attacks. In this attack, operating system vulnerabilities are exploited by accessing the administrator’s privilege to damage the entire system [22].

Recently, the integration of HPC and the Internet of Things (IoT) has become an exciting, innovative paradigm for the industry. Thus, a virtual threat by any Internet-connected device in this environment could impact a user’s physical security [23]. In IoT platforms, there is limited computational power and storage capacity, and this is considered another challenging security issue in new generation networks [24]. Protective security methods should be applied in the HPC systems to ensure the interconnection between the HPC and IoT.

To secure HPC systems the proposed methods can potentially be applied to HPC systems. These strategies and methods can be categorized into detection and protection against different vulnerabilities and threats. As depicted in “Fig. 3”, in this paper, we reviewed and summarized them in a taxonomy diagram. Further details on some major subjects are provided below:

A. Detection Solutions

1) Software Faults

In HPC systems, increasing the efficiency and high-speed platform sometimes causes the use of insecure software that is usually prepared for wide use by all researchers around the world. Many techniques are developed to find and resolve software errors. Two major approaches for software fault detection are created. Static analysis is a detecting solution that finds the software error by running the software code, this method does not run time overhead. Dynamic analysis resolves the software error during software execution. Authors in [25], presented an example of the debugger and high-speed software that is widely used in HPC systems. CAASCADE [26] is a system that is used for static analysis of HPC application software. Similarly, some other static analysis tools [27][28][29][30][31][32] as well as dynamic analysis tools [33][34][35][36] are proposed on top of Clang for different HPC applications as shown in the taxonomy diagram in “Fig. 3” [37].

2) Monitoring

The HPC system administrator uses different monitoring systems to identify and stop malicious activities. In the

performance monitoring implementation process, one of the main tasks that should be considered is the compromise between performance overhead, flexibility, and debugging. The HPC monitoring system needs to be scalable and provide fewer performance overheads on the system. If real-time data gathering is required, then the performance overhead could be high. Most software threats and insider attacks that aims to compromise CIA of HPC systems can be detected by monitoring. For instance, to combat run-time attacks DPEM [38], monitoring was developed. VARAN is an example of a performance monitoring system [39]. This solution runs over user space and prepares low performance overhead that is suitable for the HPC system.

advanced MAC mechanisms at the hardware level for access control approaches in HPC systems

2) Randomization

This method can protect the system against memory corruption attacks (e.g., buffer overflow attacks compromising confidentiality and integrity in HPC systems). Randomization techniques predict the next step execution of a program that is unpredictable. This technique has a low overhead.

Address Space Layout Randomization (ASLR) [44] and heterogeneous chip multiprocessors and heterogeneous-ISA (HIPStR) [45] are examples of randomization techniques that use software and hardware specifications for efficient randomization.

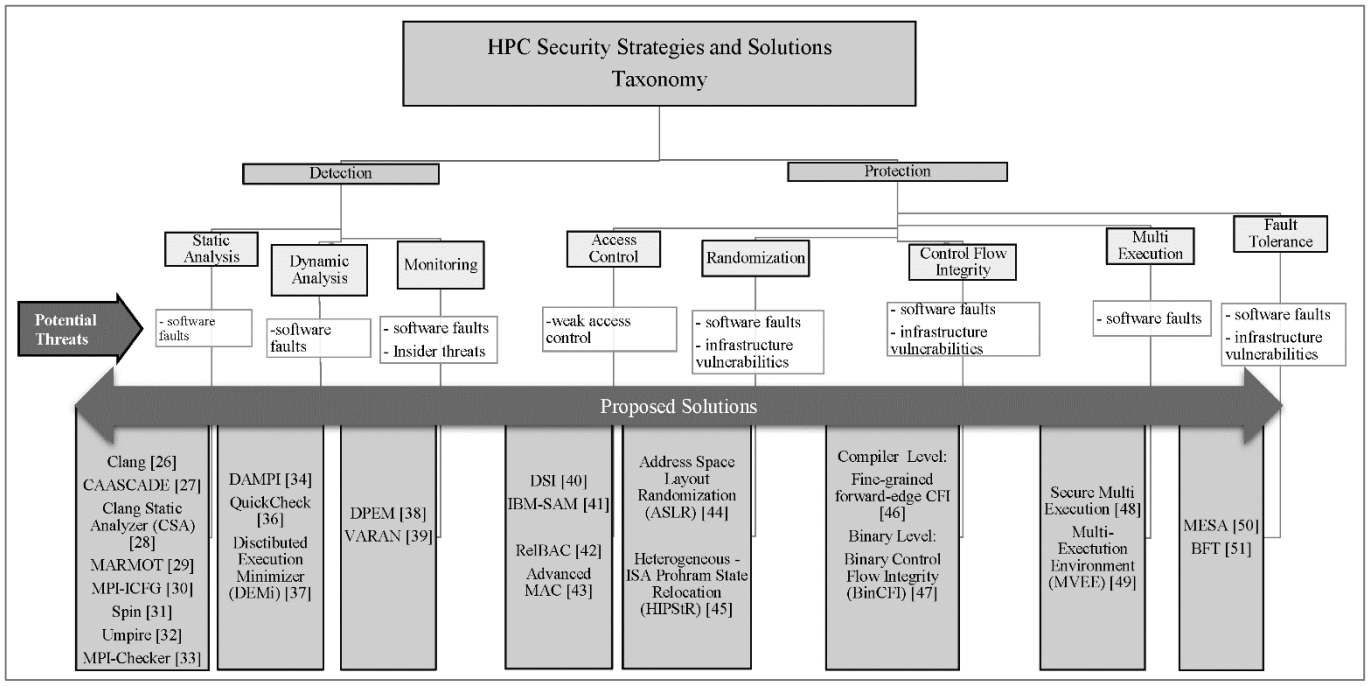

Fig. 3. Taxonomy of HPC Security Strategies and Solutions

B. Protection Solutions

1)Access Control

Different users from different organizations may have access to the HPC system. It is necessary to have a proper access control method to verify that valid users have access to sensitive resources and private data in the HPC system. HPC ACL is a protection method with low overhead. There are two fundamental access control policies: Mandatory Access Control (MAC) model that restricts access of a subject to an object by applying a set of rules. Discretionary Access Control (DAC) model that the owner of an object makes a limitation to access of different subjects to the object by defining an access control list. Many access control lists are currently used in HPC systems such as Distributed Security
Infrastructure (DSI) [40] that make available APIs for creating an access control mechanism at the kernel level. IBM
Security Access Manager (SAM) [41], RelBAC [42], and a proposed method [43] are some secure, efficient, and

3) Control Flow Integrity (CFI)

This technique can protect the system by limiting the program execution in an authorized state. Control flow integrity has two phases: the analysis phase that designates the valid states for the program execution flow [46]. The CFI built in the analysis phase should be considered in the enforcement phase with execution flow. Binary level and compile level are two levels
of implementing CFI. Due to the accessibility of source code in compile mode, it is more precise and efficient; whereas, the binary level can be working without the source code by analyzing the binaries. Some proposed methods for compiler level [46] and binary level CFI [47] are shown in “Fig. 3”.

4) Multi-Execution

In HPC systems, there is more sensitive data; therefore, information leaking can increase the concerns. In the secure

multi-execution method [47] or orchestra multi-execution environment [48], a program executes several times with different levels of sensitive data for different security levels. Therefore, it can be used to significantly reduce the leaking of sensitive information. The overhead of this method is very high [37].

5) Fault Tolerance

In HPC systems, availability is very important and should be considered closely. Typically, in an HPC system, different jobs are running by many users at the same time. Because the availability of HPC systems is a priority, these systems are designed to be capable of fault tolerance. Since the HPC systems are in large-scale systems, therefore the faults occur more in these systems. The facts such as power supply damage, hardware exposure, software threats, overheating, and industrial problems can produce faults in the HPC system [37]. Additionally, a fault happening on one node may sequentially affect the other nodes in HPC systems. MESA [49] is developed by Google to work in large-scale distributed systems to analyze this issue. Likewise, Microsoft introduced Byzantine Fault Tolerance (BFT) [50].

V. Related Case Studies

In addition to the taxonomy of HPC security strategies and solutions, this section explores the most recent and relevant researches for enhancing HPC security from software and hardware perspectives. Also, Table 1 summarizes some important and outstanding case studies conducted in different countries /regions by focusing on HPC security in recent years.

A. HPC Security: from Software Security Perspective

Managing a massive production HPC infrastructure with a varied customer base covering tens of different organizations presents several apparent administrative difficulties and challenges. Thus providing (intelligent) software-based monitoring techniques such as multi-factor authentication ecosystems [51], user-based firewall [52], or analyzing log files [20] more intelligently and accurately by deploying artificial intelligence techniques [3] are recommended by many researchers. Furthermore, most current security systems, such as intrusion detection or prevention systems, have only a limited number of logs to detect malicious attacks. Hence, more advanced machine learning techniques, especially natural language-based processing techniques, should be used to analyze various HPC systems logs to boost detection efficiency [20].

B. HPC Security: from Hardware Security Perspective

With the capabilities of security management, high performance, flexibility, and scalability, cloud computing is the dominant paradigm in today's IT world. Hence, using HPC fabric architecture for designing this environment is essential, and customers who trust these features will gain more. Cybersecurity, however, is still a significant concern, not just on the software side but also on the hardware side. While there are several studies on software-level threats, security at the hardware level has been mostly neglected. To provide users with high processing power with minimal failures, the design of the processors with cybersecurity in mind is highly desired. The security can relate to the executable code, the manipulated data, and the microcontroller's functioning. The execution environment is divided into two parts to ensure the safety of modern processors: secure resources and insecure resources, with the ability of the processor to switch through a control register between non-secure mode and secure mode [53][54].

Generally, many issues could be considered in hardware security in other infrastructures. One of the hardware security issue is the cryptographic process, which generates the secure key by special embedded hardware [24]. Another issue is to consider how to boot firmware and update them securely. Spectre and Meltdown flaws are recognized as hardware-level vulnerabilities that permit a miscreant process to access the memory, despite are not being allowed. Mitigating Spectre and Meltdown flaws are difficult, and many servers and cloud equipment and services were impacted by these hardware attacks in 2018 [55]. In 2019, one transient attack based on Meltdown, named Microarchitectural Data Sampling (MDS), was released. The MDS attack gathers data from shared CPU resources and causes data leaks between security network domains. CPU resources that are impacted by MDS include data cache, fill and store buffers [56]. These resources in the HPC system could hold the data regarding other processors and transfer them to the location where memory access can read the unauthorized data and make the vulnerability in the HPC system.

HPC systems sometimes use Hyper-Threading to allocate virtual CPUs per core on the node to increase application throughput [57]. Some MPI applications in the guest OS disable hyperthreading to prevent some vulnerabilities related to CPUs. Some studies [58][59][60] used a Moving Target Defense (MTD) solution as a self-defense system against attacks to build less vulnerable attack systems by constantly adjusting high-level configurations such as network configurations, parameters, memory address, instruction sets, and execution environment [61]. Some vendors, such as Intel that has effectively led the cloud computing and HPC system market with its hardware and software products (such as Xeon processors, Xeon Phi coprocessor, and so on), proposed automatic secure HPC fabric architecture using secure Omni-Path architecture. Omni-Path architecture is a dynamic solution with autonomic adaptive anomaly analysis, which uses diversity, obfuscation, redundancy, and autonomic management. These new features are the results of combining

software components over the Omni-Path fabric as the next generation interconnect architecture with the high-performance capability of its hardware components. Considering the high-performance capability of the hardware, such as better CPU, fabric integration with improved cost, power, density and increased bandwidth with reduced latency, can deliver additional security features [62]. Trusted Execution Environments (TEEs) are hardware-based security mechanisms that allow user-specified code and data to run securely in an enclave that cannot be breached by a compromised operating system or hypervisor. TEEs are particularly useful for running programs on untrusted platforms such as public clouds and third-party service providers. One popular TEE implementation is Intel SGX, which is available in most Intel server processors since Skylake CPUs in 2015. AMD EPYC CPUs since 2017 have also included the Secure Encrypted Virtualization (SEV) feature, which provides a secure enclave for each protected virtual machine. In the case of Intel SGX, it reserves a region of system memory called private reserved memory (PRM) to host the enclave. When a user wants to perform a secure computation, they create an enclave and execute the confidential code inside it. Before creating the enclave, an Intel service can verify the SGX support of the cloud provider through a remote attestation protocol. The user can then securely upload their code to the enclave, process encrypted data within the enclave, and return the encrypted results to the untrusted components. During the runtime of an enclave, access to enclave memory by other applications is denied by the CPU, ensuring the security of the enclave. An SGX application typically consists of both untrusted and enclave parts, and there are specific runtime interactions between these parts [67]. The advent of new GPU architectures like NVIDIA's Hopper and Blackwell is poised to significantly enhance high-performance computing (HPC) security. These GPUs incorporate advanced hardware-based security features, such as confidential computing capabilities and hardware firewalls, to protect data in use and prevent unauthorized access. The Hopper GPU, in particular, features a hardware-based trusted execution environment (TEE) anchored in an on-die hardware root of trust (RoT), which ensures the integrity and confidentiality of computations. This TEE is further secured through a chain of trust established during the boot process, involving secure boot protocols and data models (SPDM) sessions. Additionally, the Hopper GPU's confidential computing mode enables the creation of physically isolated TEEs, which can be used to run sensitive workloads securely. These advancements in GPU security have improved the overall resilience of HPC systems against various threats, including software attacks, physical attacks, and cryptographic attacks, thereby ensuring the trustworthiness of AI and HPC applications.[68]

VI. Suggested Improvements

After conducting a comprehensive survey, we further observed that security is not a product but a process; thus, admins need to learn and adapt their security knowledge from existing infrastructure and apply it to HPC systems. They can choose the right techniques to keep a good trade-off between HPC system security and performance. Moreover, security awareness and training are the key factors for the initial start of mitigating potential risks in HPC systems. Thus, organizations need a holistic approach, and they may share valuable information with peer organizations. Finally, it is worth promoting the creation of a national, regional or even international standard security working group on HPC systems. After a detailed discussion of HPC systems applications, characteristics, existing challenges, and solution, the following suggested improvements are as follows:

- Consider hardware as well as software security issues
- Understand users and define service level agreements (SLA) and usage policies
- Consider a trade-off between users' needs with security requirements
- Consider basic Operating System hardenings, such as password complexity enforcement, Expiration and Account Aging
- Consider network and host security
- Consider patching and auditing
- Establish policies for access control, restricted execution environments and file permissions
- Securing HPC using Federated and Multi-factor Authentication
- Verifying software
- Configuring a centralized logging server for systematical Log-collecting and analyzing
- File integrity monitoring and intrusion protection system (IPS) and instruction detection system (IDS)
- Applying an application layer firewall rather than a packet filtering firewall
- Share information with peer organizations

VII. Conclusion

Soaring adoption of supercomputers and HPC across diverse domains necessitates a heightened focus on their security aspects. This study meticulously dissects security challenges impacting HPC infrastructure, threats, and vulnerabilities from hardware, software, and hybrid viewpoints. We present a comprehensive taxonomy of HPC vulnerabilities mapped to corresponding mitigation strategies and solutions. Furthermore, we conduct a robust review of case studies encompassing diverse regions/perspectives (hardware, software, and hybrid) to offer insights into practical security implementations. This analysis empowers organizations to make informed decisions when selecting appropriate and effective solutions to combat security threats and vulnerabilities within their HPC systems. To secure HPC systems from possible security threats and vulnerabilities, we need to consider the main security aims of Confidentiality,

Integrity, and Availability (CIA). Attackers may exploit vulnerabilities or weaknesses in the architecture, such as infrastructure vulnerabilities, software bugs, weak access control, and insider threats. To mitigate these risks, admins need to learn and adapt their security knowledge from existing infrastructure and apply it to HPC systems. They can choose the right techniques to keep a good trade-off between HPC system security and performance. Security awareness and training are key factors for mitigating potential risks in HPC systems. Organizations need a holistic approach and may share valuable information with peer organizations. This study and review could provide a better understanding of implementing appropriate and practical solutions to prevent security threats and vulnerabilities in the HPC system . It is worth promoting the creation of a national, regional, or even international standard security working group on HPC systems .

REFERENCES

[1] R. Bulusu, P. Jain, P. Pawar, M. Afzal, and S. Wandhekar, "Addressing security aspects for HPC infrastructure," *2018 Int. Conf. Inf. Comput. Technol. ICICT 2018*, pp. 27–30, 2018, doi: 10.1109/INFOCT.2018.8356835.

[2] A. Netti, Z. Kiziltan, O. Babaoglu, A. Sîrbu, A. Bartolini, and A. Borghesi, "Online Fault Classification in HPC Systems Through Machine Learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11725 LNCS, pp. 3–16, doi: 10.1007/978-3-030-29400-7_1.

[3] D. Brayford and S. Vallecorsa, "Deploying Scientific Al Networks at Petaflop Scale on Secure Large Scale HPC Production Systems with Containers," Jun. 2020, doi: 10.1145/3394277.3401850.

[4] T. Y. Chou and Y. T. Hsu, "The strategic development and spatial information applications of smart cities in Taiwan," in *Proceedings of the 2017 Pacific Neighborhood Consortium Annual Conference and Joint Meetings: Data Informed Society, PNC 2017*, 2017, vol. 2017-Decem, doi: 10.23919/PNC.2017.8203515.

[5] W. Draft, "Working DRAFT: An Action Plan for High Performance Computing Security," pp. 1–14, 2016.

[6] T. Muhammed, R. Mehmood, A. Albeshri, and F. Alsolami, "HPC-Smart Infrastructures : A Review and Outlook on Performance Analysis Methods and Tools," 2020.

[7] S. Usman, R. Mehmood, and I. Katib, "Big data and hpc convergence for smart infrastructures: A review and proposed architecture," in *EAI/Springer Innovations in Communication and Computing*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 561–586.

[8] W.-M. Hwu, L.-W. Chang, H.-S. Kim, A. Dakkak, and I. El Hajj, *Transitioning HPC Software to Exascale Heterogeneous Computing*. .

[9] K. M. Mantripragada, A. P. D. Binotto, L. T. Tizzei, and M. A. S. Netto, "A feasibility study of using HPC cloud environment for seismic exploration," in *77th EAGE Conference and Exhibition 2015: Earth Science for Energy and Environment*, 2015, pp. 3822–3826, doi: 10.3997/2214-4609.201413291.

[10] D. A. Reed and J. Dongarra, "Exascale computing and big data," *Commun. ACM*, vol. 58, no. 7, pp. 56–68, Jul. 2015, doi: 10.1145/2699414.

[11] S. Usman, R. Mehmood, and I. Katib, "Big data and HPC convergence: The cutting edge and outlook," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018, vol. 224, pp. 11–26, doi: 10.1007/978-3-319-94180-6_4.

[12] Y. Arfat, S. Usman, R. Mehmood, and I. Katib, "Big data tools, technologies, and applications: A survey," in *EAI/Springer Innovations in Communication and Computing*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 453–490.

[13] M. Asch *et al.*, "Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry," *International Journal of High Performance Computing Applications*, vol. 32, no. 4. SAGE Publications Inc., pp. 435–479, Jul. 01, 2018, doi: 10.1177/1094342018778123.

[14] G. Mateescu, W. Gentzsch, C. R.-F. G. Computer, and undefined 2011, "Hybrid computing—where HPC meets grid and cloud computing," *Elsevier*, Accessed: May 25, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X1000213X.

[15] G. Shainer, T. Liu, … J. L.-… I. C., and undefined 2009, "Scheduling strategies for HPC as a service (HPCaaS)," *ieeexplore.ieee.org*, Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5289158/.

[16] F. S.-2011 I. 3rd I. C. on and undefined 2011, "Cloud computing security threats and responses," *ieeexplore.ieee.org*, Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6014715/.

[17] S. Jamalian, H. R.-2015 I. 8th I. Conference, and undefined 2015, "Data-intensive hpc tasks scheduling with sdn to enable hpc-as-a-service," *ieeexplore.ieee.org*, Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7214095/.

[18] A. K. Singh and S. D. Sharma, "High performance computing (HPC) data center for information as a service (IaaS) security checklist: Cloud data governance," *Webology*, vol. 16, no. 2, pp. 83–96, 2019, doi: 10.14704/web/v16i2/a192.

[19] P. Korambath, "Cyber Security In High-Performance Computing Environment," 2014.

[20] Z. Luo, Z. Qu, T. Nguyen, H. Zeng, and Z. Lu, "Security of HPC Systems: From a Log-analyzing Perspective," *ICST Trans. Secur. Saf.*, vol. 6, no. 21, p. 163134, 2019, doi: 10.4108/eai.19-8-2019.163134.

[21] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," 2017.

[22] T. Yamauchi, Y. Akao, … R. Y.-… I. C. on, and undefined 2018, "Additional kernel observer to prevent privilege escalation attacks by focusing on system call privilege changes," *ieeexplore.ieee.org*, Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8625137/.

[23] L. de S. Cimino, … J. de R.-2017 V. B., and undefined 2017, "IoT and HPC integration: revision and perspectives," *ieeexplore.ieee.org*, Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8116570/.

[24] B. Pahlevanzadeh, S. Koleini, and S. I. Fadilah, "Security in IoT: Threats and Vulnerabilities, Layered Architecture, Encryption Mechanisms, Challenges and Solutions," in *Communications in Computer and Information Science*, Dec. 2021, vol. 1347, pp. 267–283, doi: 10.1007/978-981-33-6835-4_18.

[25] "Control Flow Integrity — Clang 13 documentation." https://clang.llvm.org/docs/ControlFlowIntegrity.html#publications. (accessed May 26, 2021).

[26] M. G. Lopez, O. Hernandez, R. D. Budiardja, and J. C. Wells, "CAASCADE: A System for Static Analysis of HPC Software Application Portfolios," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11027 LNCS, pp. 90–104, doi: 10.1007/978-3-030-17872-7_6.

[27] T. Kremenek, "Finding software bugs with the Clang Static Analyzer." Accessed: May 30, 2021. [Online]. Available: https://llvm.org/devmtg/2008-08/Kremenek_StaticAnalyzer.pdf.

[28] B. Krammer, T. Hilbrich, V. Himmler, B. Czink, K. Dichev, and M. S. Müller, "MPI Correctness Checking with Marmot," in *Tools for High Performance Computing*, Springer Berlin Heidelberg, 2008, pp. 61–78.

[29] B. Kreaseck, M. M. Strout, and P. Hovland, "Depth Analysis of MPI Programs." Accessed: May 30, 2021. [Online]. Available: https://www.cs.rochester.edu/u/cding/amp/papers/full/Depth Analysis of MPI Programs.pdf.

[30] S. F. Siegel, "Verifying parallel programs with MPI-spin," in

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2007, vol. 4757 LNCS, pp. 13–14, doi: 10.1007/978-3-540-75416-9_8.

[31] "Dynamic Software Testing of MPI Applications with Umpire | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/1592764 (accessed May 30, 2021).

[32] A. Droste, M. Kuhn, and T. Ludwig, "MPI-checker - Static analysis for MPI," Nov. 2015, doi: 10.1145/2833157.2833159.

[33] A. Vo, S. Aananthakrishnan, G. Gopalakrishnan, B. R. De Supinski, M. Schulz, and G. Bronevetsky, "A scalable and distributed dynamic formal verifier for MPI programs," 2010, doi: 10.1109/SC.2010.7.

[34] K. Claessen and J. Hughes, "QuickCheck," 2000, pp. 268–279, doi: 10.1145/351240.351266.

[35] D. Clements-Croome, "Sustainable intelligent buildings for people: A review," *Intelligent Buildings International*, vol. 3, no. 2. pp. 67–86, 2011, doi: 10.1080/17508975.2011.582313.

[36] C. Scott, A. Panda, A. Krishnamurthy, V. Brajkovic, G. Necula, and S. Shenker, *Open access to the Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16) is sponsored by USENIX. Minimizing Faulty Executions of Distributed Systems Minimizing Faulty Executions of Distributed Systems*. 2016.

[37] T. Hou, T. Wang, D. Shen, Z. Lu, and Y. Liu, "Autonomous Security Mechanisms for High-Performance Computing Systems: Review and Analysis," in *Adaptive Autonomous Secure Cyber Systems*, Springer International Publishing, 2020, pp. 109–129.

[38] C. Ko, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach." Accessed: May 26, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/601332/.

[39] P. Hosek and C. Cadar, "VARAN the Unbelievable," *ACM SIGARCH Comput. Archit. News*, vol. 43, no. 1, pp. 339–353, May 2015, doi: 10.1145/2786763.2694390.

[40] M. Pouzandi, A. Apvrille, E. Gingras, A. Medenou, and D. Gordon, "Distributed Access Control for Carrier Class Clusters." Accessed: May 26, 2021. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.8289&rep=rep1&type=pdf.

[41] "IBM Security Verify Access - Overview | IBM." https://www.ibm.com/products/verify-access (accessed May 26, 2021).

[42] F. Giunehiglia, R. Zhang, and B. Crispo, "RelBAC: Relation based access control," in *Proceedings of the 4th International Conference on Semantics, Knowledge, and Grid, SKG 2008*, 2008, pp. 3–11, doi: 10.1109/SKG.2008.76.

[43] D. Gros, M. Blanc, J. Briffaut, and C. Toinard, "Advanced MAC in HPC systems: Performance improvement," in *Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012*, 2012, pp. 699–702, doi: 10.1109/CCGrid.2012.83.

[44] "CiNii Articles - PaX address space layout randomization (ASLR)." https://ci.nii.ac.jp/naid/10026762411/ (accessed May 29, 2021).

[45] A. Venkat, S. Shamasunder, H. Shacham, and D. M. Tullsen, "HIPStR- Heterogeneous-ISA program state relocation," *ACM SIGPLAN Not.*, vol. 51, no. 4, pp. 727–741, Apr. 2016, doi: 10.1145/2872362.2872408.

[46] M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti, "Control-flow integrity principles, implementations, and applications," in *ACM Transactions on Information and System Security*, Oct. 2009, vol. 13, no. 1, doi: 10.1145/1609956.1609960.

[47] D. Devriese and F. Piessens, "Noninterference Through Secure Multi-Execution." Accessed: May 25, 2021. [Online]. Available: http://example.com/img.jpg';

[48] B. Salamat, B. Salamat, T. Jackson, A. Gal, and M. Franz, "Orchestra: Intrusion detection using parallel execution and monitoring of program variants in userspace," *Proc. Eur. Conf. Comput. Syst.*, 2009, Accessed: May 31, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary;jsessionid=9874DF4291D699CBD7AA57BDD99F1099?doi=10.1.1.218.7044.

[49] A. Gupta *et al.*, "Mesa: Georeplicated, near realtime, scalable data warehousing," in *Proceedings of the VLDB Endowment*, 2014, vol. 7, no. 12, pp. 1259–1270, doi: 10.14778/2732977.2732999.

[50] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002, doi: 10.1145/571637.571640.

[51] A. Prout *et al.*, "Securing HPC using federated authentication," Sep. 2019, doi: 10.1109/HPEC.2019.8916255.

[52] A. Prout *et al.*, "Enhancing HPC Security with a User-Based Firewall." Accessed: May 25, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7761641/.

[53] "US7363491B2 - Resource management in security enhanced processors - Google Patents." https://patents.google.com/patent/US7363491B2/en (accessed May 29, 2021).

[54] "US6996725B2 - Encryption-based security protection for processors - Google Patents." https://patents.google.com/patent/US6996725B2/en (accessed May 31, 2021).

[55] A. Prout *et al.*, "Measuring the Impact of Spectre and Meltdown," Nov. 2018, doi: 10.1109/HPEC.2018.8547554.

[56] D. Moghimi, M. Lipp, B. Sunar, and M. Schwarz, "Medusa: Microarchitectural Data Leakage via Automated Attack Synthesis," 2020. Accessed: May 29, 2021. [Online]. Available: https://github.com/vernamlab/Medusa.

[57] "Best practices for running tightly coupled HPC applications on Compute Engine." https://cloud.google.com/architecture/best-practices-for-using-mpi-on-compute-engine (accessed May 30, 2021).

[58] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A Survey of Moving Target Defenses for Network Security," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1909–1941, Jul. 2020, doi: 10.1109/COMST.2020.2982955.

[59] J. Zheng and A. S. Namin, "A Survey on the Moving Target Defense Strategies: An Architectural Perspective," *J. Comput. Sci. Technol.*, vol. 34, no. 1, pp. 207–233, Jan. 2019, doi: 10.1007/s11390-019-1906-z.

[60] "US20160323313A1 - Moving-target defense with configuration-space randomization - Google Patents." https://patents.google.com/patent/US20160323313A1/en (accessed May 31, 2021).

[61] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront, "MT6D: A moving target IPv6 defense," in *Proceedings - IEEE Military Communications Conference MILCOM*, 2011, pp. 1321–1326, doi: 10.1109/MILCOM.2011.6127486.

[62] F. Fargo and S. Sury, "Autonomic Secure HPC Fabric Architecture," *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2018-Novem, pp. 1–4, 2019, doi: 10.1109/AICCSA.2018.8612872.

[63] X. Yu, F. Wei, X. Ou, M. Becchi, T. Bicer, and D. D. Yao, "GPU-Based Static Data-Flow Analysis for Fast and Scalable Android App Vetting," in *Proceedings - 2020 IEEE 34th International Parallel and Distributed Processing Symposium, IPDPS 2020*, May 2020, pp. 274–284, doi: 10.1109/IPDPS47924.2020.00037.

[64] P. Yellu, Z. Zhang, M. M. R. Monjur, R. Abeysinghe, and Q. Yu, "Emerging applications of 3D integration and approximate computing in high-performance computing systems: Unique security vulnerabilities," *2019 IEEE High Perform. Extrem. Comput. Conf. HPEC 2019*, 2019, doi: 10.1109/HPEC.2019.8916503.

[65] G. Zhu, Y. Zeng, and M. Guo, "A Security Analysis Method for Supercomputing Users' Behavior," *Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017*, pp. 287–293, 2017, doi: 10.1109/CSCloud.2017.19.

[66] M. Eldred, A. Good, and C. Adams, "A case study on data protection and security decisions in cloud HPC," *Proc. - IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2015*, pp. 564–568, 2016, doi: 10.1109/CloudCom.2015.114.

[67] K. Chen, "Confidential High-Performance Computing in the Public Cloud," in IEEE Internet Computing, vol. 27, no. 1, pp. 24-32, 1 Jan.-Feb. 2023, doi: 10.1109/MIC.2022.3226757.

[68] Confidential Computing on NVIDIA H100 GPUs for Secure and

Trustworthy AI , https://developer.nvidia.com/blog/confidential-computing-on-h100-gpus-for-secure-and-trustworthy-ai

[69] Nolte H, Spicher N, Russel A, Ehlers T, Krey S, Krefting D, Kunkel J.," Secure HPC: A workflow providing a secure partition on an HPC system." Future Generation Computer Systems. 2023 Apr 1;141:677-91.

[70] V. M. Weaver, "Improving HPC Security with Targeted Syscall Fuzzing," 2022 IEEE/ACM First International Workshop on Cyber Security in High Performance Computing (S-HPC), Dallas, TX, USA, 2022, pp. 1-8, doi: 10.1109/S-HPC56715.2022.00006.

[71] Nagarikar, A., Patel, A., Gupta, K. G., Sharma, S., Afzal, M., & Wandhekarl, S. (2024). User Login Behaviour Analysis in HPC clusters using Data Analysis and Probabilistic Technique. International Journal of Intelligent Systems and Applications in Engineering, 12(3s), 250-259.

[72] Huq, N., Lin, P., Reyes, R., & Perine, C. "A Survey of Cloud-Based GPU Threats and Their Impact on AI, HPC, and Cloud Computing", 2024, https://documents.trendmicro.com

TABLE I. RELATED CASE STUDIES (SOFTWARE AND HARDWARE SOLUTIONS FOR ENHANCING HPC SECURITY)

Title / Reference	Year/ Region	Proposed Method (Hardware, Software, Hybrid Solutions)	Methodology/Findings /Contribution
User Login Behaviour Analysis in HPC clusters using Data Analysis and Probabilistic Technique.[71]	2024 India	Software	Methodology: use of machine learning, data analysis and probabilistic techniques to identify patterns that can be used to identify HPC cluster anomalous login behaviour based on IP and timings. Findings: Developed a probabilistic model will monitor each user's login behaviour and stop any unauthorized activity in HPC clusters. also discusses the data preprocessing and feature engineering techniques to extract information from sshd logs. Contribution: to enhance HPC security by identifying unauthorized activities and improving the detection of anomalies in user login behavior.
A Survey of Cloud-Based GPU Threats and Their Impact on AI, HPC, and Cloud Computing.[72]	2024 USA	Hybrid solution -Hardware -Software	Methodology: The paper follows a survey-based approach related to GPU security threats. The various types of attacks, vulnerabilities, and security risks associated with cloud-based GPU environments are categorized and evaluated. Findings: Exploiting vulnerabilities in GPU drivers and firmware, GPU-assisted code obfuscation techniques, Memory snooping and cross-VM attacks in virtualized GPU environments, GPU side-channel attacks, GPU rootkits, API abuse and kernel manipulation, Denial-of-Service(DoS) attacks, GPU malware for cryptomining, Compromised AI models and Trojaning Contribution: proposes a set of security recommendations and best practices for mitigating the identified threats, including driver and firmware security, GPU usage monitoring, application-level security measures, hardware security modules (HSMs), access control policies, and education and awareness
Secure HPC: A workflow providing a secure partition on an HPC system[69]	2023 Germany	Software	Methodology: Design of a multi-node secure workflow with parallel I/O, a strict security model enforced by the system and network features, and lastly the demonstration of a medical use case. Findings: A workflow allowing users to transfer, store and analyze data with the highest privacy requirements Contribution: Proposing a practical and effective solution for creating secure partitions, demonstrating its applicability in real-world scenarios and Proposing a workflow allowing users to transfer, store and analyze data with the highest privacy requirements.
Improving HPC Security with Targeted Syscall Fuzzing[70]	2022 USA	Software	Methodology: Integrates domain-specific knowledge, tailored fuzzing strategies, and hybrid execution techniques for robust vulnerability detection in critical system calls essential for HPC applications. Findings: The paper highlights the efficacy of targeted syscall fuzzing as a key asset for enhancing HPC security, and uncovering vulnerabilities in the Linux performance infrastructure. Demonstrates superior effectiveness compared to conventional fuzzing methods. Contribution: Pioneers targeted syscall fuzzing for HPC security assessment, introducing a specialized framework for HPC system calls and workloads. Identifies and addresses critical security weaknesses in HPC libraries and system calls, showcasing the security-enhancing potential of fuzzing in HPC systems.
GPU-Based Static Data-Flow Analysis for Fast and Scalable Android App Vetting[63]	2020 China	- Hybrid solution -Hardware -Software	Methodology: Proposed GDroid is a highly optimized GPU implementation of static data-flow analysis tailored to Android applications, combining traditional algorithms with application specific optimizations and demonstrating significant performance improvements. Findings: The paper finds that while harnessing GPUs for Android app vetting presents challenges, specific optimizations can significantly improve performance and scalability compared to traditional CPU-based methods. However, further research is needed to address remaining issues like false positives and broader applicability. Contribution: GDroid is the first implementation of Android program analysis on HPC platforms and serves as the core of fast Android App security screening makes significant contributions to the field of Android app security research by demonstrating the potential of GPUs for faster and more scalable analysis, introducing novel application-specific optimizations, and highlighting important research directions for the future.

Emerging Applications of 3D Integration and approximate computing in High performance computing systems: Unique Security Vulnerabilities[64]	2020 USA	Hardware	Methodology: While the paper doesn't present concrete methodologies for exploiting vulnerabilities, it lays the groundwork for future research by identifying potential security weaknesses and outlining areas where further investigation is needed. It serves as a valuable starting point for developing techniques to ensure the security of these emerging HPC technologies. Findings: Introduced the security threats from the hardware perspective, particularly; considering supply chain attacks. It also reviews some Trojan detection techniques and their weaknesses in the new generation HPC hardware, aiming to inspire researchers to develop effective countermeasures to improve the resilience of HPC systems against security threats on the hardware components used in HPC. The paper's findings raise significant concerns about the security of emerging HPC technologies. By identifying and analyzing these vulnerabilities, the authors pave the way for further research and development of mitigation strategies to ensure the secure deployment and operation of these powerful computing systems. Contribution: the paper makes a significant contribution to overall research by raising awareness of novel security threats in HPC, promoting proactive mitigation strategies, and encouraging interdisciplinary collaboration between HPC security and hardware security experts. It paves the way for further research and development in securing these emerging technologies and ensuring the safe and secure future of High-Performance Computing.
Deploying Scientific AI Networks at Petaflop Scale on Secure Large Scale HPC Production Systems with Containers [3]	2020 European Countries	Software	Methodology: The paper employs a custom container image called Charliecloud to package the AI framework (TensorFlow) and its dependencies. This approach isolates the software environment from the HPC system, ensuring clean execution and security. shell scripts and a custom C program are used to unpack and activate the Charliecloud image within the HPC environment. Findings: The paper significantly advances scientific AI by showcasing the effectiveness and advantages of employing containers for secure and scalable AI deployments on HPC systems. It addresses challenges in deploying AI frameworks securely in HPC environments, covering open-source software challenges, HPC features, access controls, trade-offs between flexibility and resource usage, and highlights opportunities for future improvements in this domain. Contribution: Demonstrating the power of containerized deployments on HPC at petaflop scale, enhancing security, scalability, and performance. It fosters interdisciplinary collaboration and knowledge sharing, laying the groundwork for future advancements in this rapidly evolving field.
Autonomous Security Mechanisms for High-Performance Computing Systems: Review and Analysis [37]	2020 USA	No specific proposed solution	Methodology: The paper doesn't explore specific technical details or implementation methodologies. It studies the traditional network security strategies and newly proposed HPC security methods for discovering some applicable conventional methods for enhancing HPC security It serves as a valuable starting point for further research and development in this critical area of HPC security. Findings: The paper's findings raise awareness of the opportunities and challenges presented by autonomous security in HPC. It calls for further research and development of tailored solutions to overcome these challenges and unlock the full potential of this technology for securing future HPC systems. Contribution: the paper's contribution lies not in presenting a final solution but in providing a critical examination of a promising approach and outlining the necessary research directions for its successful implementation in HPC systems.
Security of HPC systems: from a log analyzing perspective [20]	2019 USA	Software	Methodology: The paper's methodology focuses on highlighting the value of log analysis for HPC security, outlining different types of log data and potential attack patterns, and discussing existing techniques and future directions for enhancing log-based intrusion detection in these complex environments. It provides a valuable framework for security professionals working with HPC systems to leverage the power of log analysis for improved security posture. Findings: The paper surveys security challenges in detecting malicious users or behaviors in HPC systems through log analysis, highlighting weaknesses in various intrusion detection systems (IDS) defending methods. It suggests a future software-based solution using AI and machine learning to enhance IDS systems and strengthen HPC security. Emphasizing the immense potential and key challenges of leveraging log analysis for HPC security, the paper calls for additional research and development in efficient and scalable analysis tools, integration with existing HPC infrastructure, and standardized logging practices to achieve truly effective log-based security solutions. Contribution: the paper's contribution lies not just in showcasing the current state of log analysis in HPC security but also in outlining the necessary steps for its advancement. It stimulates further research and development efforts, ultimately leading to more robust and reliable security solutions for the ever-growing and crucial world of High-Performance Computing.

Securing HPC using federated Authentication [51]	2019 USA	Software	Methodology: 1. Integrating with external identity providers: The paper details the process of integrating the MIT SuperCloud TX-E1 system with two large federation ecosystems: the U.S. Government PKI and InCommon. This allows users to authenticate using their existing credentials from affiliated institutions, simplifying access and reducing password management needs. 2. Impersonation and remote access through web portal: The paper describes how the web portal leverages the authenticated user's identity to impersonate them for system calls, enabling secure web-based access to files and interactive execution of HPC jobs. This eliminates the need for separate logins on individual nodes and promotes centralized control. Findings: The paper reports on a case study that was conducted in an HPC security at MIT Lincoln Laboratory Supercomputing Center (LLSC) using multi-factor and federated Authentication. Successful integration with federated providers, Enhanced security and convenience, Scalability and manageability, Potential limitations, and future exploration Contribution: The paper pioneers a user-friendly approach to HPC security, showcasing its benefits and paving the way for broader adoption. The proposed federated authentication method eliminates administrative overhead in user account management and enhances HPC system security through existing multi-factor authentication systems.
Autonomic secure HPC fabric architecture [62]	2018 USA	-Hardware (Focused aim) -Hybrid solution (SW and HW) also presented in the paper for optimized "controlling."	Methodology: the paper doesn't offer specific methodologies or implementation details but rather paints a broader picture of what an autonomic security architecture for HPC could look like. It raises valuable questions and identifies research areas crucial for enhancing secure and reliable HPC operations. It evaluates an autonomic HPC fabric architecture that leverages both resilient computing capabilities and adaptive anomaly analysis for further security. Findings: The proposed Omni-Path architecture will make Intel one step ahead of its competitors when security and adaptive anomaly behavior detection are the main concerns. The paper's findings underline the potential and challenges of applying autonomic security in HPC, encouraging further research and development in this crucial area. Contribution: the paper's contribution lies in sparking research and development in a crucial area for secure and reliable HPC operations. It establishes a foundational framework, identifies future research directions, and promotes collaboration, ultimately impacting the overall advancement of secure HPC technology.
Addressing security aspects of HPC infrastructure [1]	2018 India	No specific proposed solution	Methodology: The authors used a mixed-method approach that combined a literature review, Expert interviews and Case studies that hey analyzed two real-world HPC security incidents to identify common vulnerabilities and mitigation strategies. Findings: Common security threats in HPC include: Unauthorized access, Data breaches, Malware infections, Denial-of-service attacks. Factors contributing to HPC security vulnerabilities include the complexity of HPC systems, Lack of security awareness, and Inadequate security measures. Contribution: its proposed framework for improving security in HPC environments. The framework includes five key components: Security policy and awareness, Access control, Data security, System hardening, Incident response. The paper also provides specific recommendations for implementing each component of the framework.
Undefined IoT and HPC integration: Revision and perspectives [23]	2017 Brazil	No specific proposed solution	Methodology: The paper primarily uses a review and analysis methodology. The authors surveyed existing literature on both IoT and HPC technologies, focusing on their individual characteristics and potential synergies. They then analyzed the challenges and opportunities of integrating these two technologies, considering various aspects like resource management, security, and data processing. Lack of any proposed method or even an inclusive review Findings: The paper identifies several challenges in integrating IoT and HPC. HPC capabilities can enable advanced analytics on IoT data, leading to improved decision-making and insights. Integration can lead to more efficient utilization of HPC resources by dynamically adapting to varying IoT workloads. Combining IoT and HPC can spawn new application areas like smart cities, connected healthcare, and predictive maintenance. Contribution: The paper's main contribution lies in its comprehensive review and analysis of the challenges and opportunities in integrating IoT and HPC. It provides a valuable reference point for researchers and practitioners working in this field. Additionally, it proposes future research directions for overcoming the identified challenges and unlocking the full potential of this integration.
A Security Analysis Method for Supercomputing Users' Behavior [65]	2017 China	-Hardware -Software	Methodology: The authors used a data-driven approach that consisted of the following steps: Data collection, Data preprocessing, Feature extraction, Anomaly detection, Risk assessment Findings: The paper found that their method was able to successfully identify various types of anomalous user behavior, including Resource abuse, Unauthorized access, Malware activity, Data exfiltration Contribution: The paper's main contribution is its proposed data-driven method for analyzing user behavior in supercomputing environments. By implementing this method, supercomputing administrators can proactively identify and address security risks, protecting their systems and data from malicious activities.

Enhancing HPC security with a user-based Firewall [52]	2016 USA	Software	Methodology: The paper proposes a user-based firewall to enhance security in High Performance Computing (HPC) environments. This firewall operates at the Linux netfilter level, offering finer-grained control compared to traditional host-based firewalls. Findings: The proposed solution, overcomes the problem of traditional system or network-level firewalling techniques to address the software and application level in the multiuser environment with no changes to users' code when they need to work collaboratively with other users or have a high rate of short connections Contribution: The paper's main contribution is its pioneering proposal and implementation of a user-based firewall for HPC environments.
A case study on data protection and security decisions in cloud HPC [66]	2015 UK	No specific proposed solution	Methodology: Action research was used to examine the nuances throughout the project as the service was moved from on-premise to a public cloud HPC. Findings: Explored some emergent issues affecting initiation, technical security challenges and practicalities that occur within a cloud HPC project developed a method for making critical security decisions, and identified the evaluation of a significant change in an HPC provisioning model. Contribution: development of a method for making critical security decisions in a cloud HPC project.

How to cite: S.Koleini, B.Pahlevanzadeh, Enhancing High-Performance Computing (HPC) Security: A Comprehensive Review of Detection and Protection Strategies, Journal of Distributed Computing and Systems(JDCS), Vol 6, Issue 12, Page 12-24, 2024.