Systematic Threat Modelling of High-Performance Computing Systems: The V:HPCCRI Case Study

Raffaele Elia[a], Daniele Granata[b] and Massimiliano Rak[c]
Department of Engineering, University of Campania Luigi Vanvitelli, Via Roma 9, Aversa (CE), Italy

Keywords: HPC, Threat Modelling, Security Assessment, Supercomputer.

Abstract: High-Performance Computing (HPC) systems play a crucial role in different research and industry tasks, boasting high-intensity computing capacity, high-bandwidth network connections, and extensive storage at each HPC centre. The system's objectives, coupled with the presence of valuable resources and sensitive data, make it an attractive target for malicious users. Traditionally, HPC systems are considered ”trusted” with users having significant rights and limited protective measures in place. Additionally, its heterogeneous nature complicates security efforts. Applying traditional security measures to individual cluster nodes proves insufficient as it neglects the system's holistic perspective. To address these challenges, this paper presents a methodology for collecting threats affecting HPC environments from the literature analysis using a Systematic Search. Key contributions of this work include the application of the presented methodology to support the HPC domain through the definition of an HPC-specific threat catalogue and, starting from it, the generation of a threat model for a real-world case study: the V:HPCCRI supercomputer.

1 INTRODUCTION

High-Performance Computing (HPC) represents a computing paradigm characterized by exceptionally powerful computing capability. HPC systems are used for various research and industry tasks, with each HPC centre equipped with a wealth of highly desirable resources: high-intensity computing capacity, high-bandwidth network connections, and extensive storage (Mogilevsky et al., 2005). The objectives for which an HPC system is designed, along with the presence of attractive resources and sensitive, critical data within it, make the infrastructure an interesting target for malicious users. On the other hand, HPC systems historically originate in academic and research environments. They are often considered “trusted” systems (users have significant rights, and there are rarely sophisticated protective measures in place for those who have access to the system). Furthermore, the heterogeneity of the resources that may comprise the infrastructure further complicates the situation: the use of different technologies extends the attack vectors and makes it more challenging to ensure the security of the infrastructure.

Threat Modeling is accepted as a critical step for assessing system security. (Granata and Rak, 2023) illustrates a set of tools for fine-grained threat modeling. This approach involves meticulously identifying potential malicious behaviours that could impact a system, focusing on the various components involved. By undertaking threat modeling, the objective is to gain a comprehensive understanding of the potential threats that could target the system. This understanding allows for the development of suitable countermeasures to mitigate these threats effectively. Our approach relies on the concept of threat, which refers to malicious behavior that can be performed by a threat agent. However, it does not consider technical aspects such as security vulnerabilities or weaknesses. It's worth noting that threat modelling is a high-level practice compared to technological assessments. In other words, we prioritize understanding and mitigating potential threats over focusing on specific technical flaws or vulnerabilities in the security system. Following this research line, our work is based on a methodology (Granata et al., 2023) aimed at building a catalogue of all the *fine-grained* threats related to a specific domain (in this case, HPC). Once the catalogue is available, it can be used to produce threat models for specific scenarios. The main contributions

[a] https://orcid.org/0009-0002-1325-7094
[b] https://orcid.org/0000-0002-6776-9485
[c] https://orcid.org/0000-0001-6708-4032

Elia, R., Granata, D. and Rak, M.
Systematic Threat Modelling of High-Performance Computing Systems: The V:HPCCRI Case Study.
DOI: 10.5220/0012733000003711

In *Proceedings of the 14th International Conference on Cloud Computing and Services Science (CLOSER 2024)*, pages 327-337
ISBN: 978-989-758-701-6; ISSN: 2184-5042

of this work are: i) the application of our defined methodology aimed at extending a graph-based technique and building a threat catalogue on an HPC system; ii) the generation of a *fine-grained* threat model of a real case study: *V: HPCCRI*.
The structure of our work is outlined as follows: Section 2 presents an overview of the significant contributions made to threat modeling in the context of HPC, while also highlighting the specific gap our research aims to address. In Section 3, we elaborate on the methodology employed for gathering threats. Sections 4 and 5 delve into the detailed phases of this methodology within the HPC context. Furthermore, Section 6 offers insights into the practical application of our methodology through a real-world case study: the V:HPCCRI supercomputer. Finally, Section 7 summarizes the conclusions drawn from our research and outlines possible future work.

2 RELATED WORK

As anticipated above, our paper aims to build a catalogue containing fine-grained threats affecting HPC assets to support our threat modelling methodology. Accordingly, in this section, we provide a comprehensive analysis of the scientific papers that focused on HPC threat modelling, describing how the threats have been selected as well as an analysis of the technique used to assess the HPC infrastructure.

NIST Special Publication 800-223 (Guo et al., 2023) offers a detailed description of the HPC key components and the threats that can be affected by underlining the security of these assets. The HPC architecture, the main components and how they can be analyzed will be described in detail in Section 4. Anyway, it is important to note that the document divides an HPC architecture into **Zones** and evaluates the threats each zone may be affected. The selection phase plays a crucial role in the identification of threats. It simplifies and enhances the process of recognizing potential malicious behaviours. However, it is important to note that this approach operates at a high level when modelling threats. This means that it takes a broad perspective and may not align with our specific approach or methodology for addressing threats. In essence, while the selection phase facilitates the identification process, its high-level nature might diverge from our more detailed and nuanced approach to modelling and managing threats. A detailed analysis of the HPC architecture is reported in Sec. 4. Hou et al.'s recent work (Hou et al., 2020a) examines high-level security requirements specific to High-Performance Computing (HPC) systems. The study underscores distinctions from general-purpose computers and analyzes corresponding security threats. Notably, the authors emphasize the need for a robust access control policy to counter confidentiality-related threats in HPC. This insight reinforces the importance of stringent access control mechanisms and user access management for ensuring HPC system security, contributing significantly to the understanding of security challenges in this domain. Some relevant scientific authors emphasize the need for a systematic and comprehensive threat analysis approach tailored to the unique characteristics of HPC clusters. As an example, Mogilevsky et al. (Mogilevsky et al., 2005) advocate for the use of a structured Confidentiality, Integrity, and Availability (CIA) model as a basis for their proposed threat model. As a result, the techniques used to extract threats and security issues from an HPC system are limited in literature because most of the work does not describe the way the threats have been collected and extracted from the model. To fill this gap, we used an already-consolidated technique to systematically extend our threat catalogue in the context of HPC systems. The catalogue has been built to extract fine-grained threats from the model that affect parts of a supercomputer.

3 METHODOLOGY

This work aims to collect a detailed catalogue of HPC threats from literature to support our fine-grained threat model generation technique.

The technique (Granata et al., 2023) consists of four steps: (i) Domain Analysis; (ii) Systematic Threat Search; (iii) Data Analysis; (iv) Final Results. Domain analysis involves identifying the primary component types, including both hardware and software components, as well as protocols used in the systems within the target domain. Identifying assets is crucial because they are the elements valued by the owner and require protection. This process begins by referencing architectures in scientific papers, surveys and white papers. In our domain, the reference architecture we based our work on is the one proposed by NIST (Guo et al., 2023) and described in detail in section 4. The key outcome of domain analysis is the enhancement of our modeling technique, allowing the definition of new asset types to take into account when modelling an HPC scenario. The Systematic Threat Search phase aims at collecting threats affecting the HPC assets and protocols collected in the previous step from different sources. In this case, a common problem in literature is identifying a comprehensive set of threats for each HPC asset. Accordingly, our technique relies on a Systematic Literature Review (SLR) aimed at collecting all the threats in a structured way as well as an overview of the threat modelling techniques used to collect the threats. Resulting of the SLR, the data extracted will be analysed to derive threats, formulated structurally. A threat, in our context, is delineated as a triad of (threat agent, compromised asset, and malicious behaviour). In essence, it represents the proactive actions undertaken by a threat agent with the intention of compromising an asset. It is worth noticing that in this work, we did not take into account the threat agents since our aim is to collect threats and build a structured threat catalogue. For further details, a technique aimed at selecting threat agents in an automated way is shown in (Granata and Rak., 2021). Data Analysis phase describes the way threats have been selected from the papers as well as the data model used to describe a threat.

As a result, (i) an extension of our modelling technique for the considered domain is formulated; and (ii) the threat catalogue, which is a structured representation of all information related to the security of the system, highlighting the threats to which each asset type is exposed. The following sections will describe in detail each phase of the technique applied to the HPC context.

4 HPC DOMAIN ANALYSIS

This section presents a detailed analysis of the High-Performance Computing domain, taking into account the reference architecture proposed by NIST (Guo et al., 2023). Subsequently, starting from the reference architecture, the identified assets are described, highlighting the reasons why they need to be adequately protected. Lastly our modelling technique extension (Granata et al., 2022) is presented, focusing on new asset types.

4.1 HPC Reference Architecture

According to NIST (Guo et al., 2023), as in evidence in figure 1, an HPC system consists of four distinct function zones: (i) access zone; (ii) computing zone; (iii) data storage zone; and (iv) management zone.

The access zone consists of one or more nodes, connected to external networks, that provide services for authenticating and authorizing the access of users and administrators and, possibly data transfer services and web portals allowing for a range of web-based interfaces to access HPC system services. At least one node provides shells that can be used to launch interactive or batch jobs.

The computing zone involves a set of compute nodes connected by one or more high-speed networks through which it is possible to run parallel jobs at scale. Some nodes can be equipped with hardware accelerators (e.g., GPU) to speed up applications. High-performance communication networks (e.g., InfiniBand, Omni-Path) are characterized by high bandwidth and ultra-low latency; and they serve the purpose of connecting compute nodes with data storage zones. Instead, non-high-performance communication networks (e.g., Ethernet) are used as cluster internal networks to connect the high-performance computing zone with the management zone and access zone.

The data storage zone includes one or multiple high-speed parallel file systems to provide data storage services for user data. They are designed to handle vast amounts of data, offering efficient storage capabilities and rapid data access for both reading and writing purposes. Typical classes of storage systems encompass parallel file systems (PFS), node-local storage for low-latency workloads, and archival file systems that defend against data loss and support campaign storage.

The management zone encompasses a pool of nodes for HPC system operation and management. It provides necessary protocols and services required by the hosts within the other zones such as Domain Name Serivces (DNS), Network Time Protocol (NTP), as well as configuration definitions, authentication, and authorization services through an LDAP server. These services can run on dedicated hardware or virtual machines. Additionally, the management zone includes storage systems for configuration data and node images, as well as logging and analysis servers to alert administrators of events. Resource requests for specific workloads are coordinated by schedulers like SLURM and Portable Batch System (PBS) due to the distributed nature of HPC systems.

4.2 Asset Identification

As previously explained, assets denote what must be safeguarded. In this section, we identified 23 assets from the analysis of HPC reference architecture proposed by NIST. Each node in the described zone is treated as an asset, resulting in 10 initial asset types: login node, data transfer node, web portal node, compute node, storage node, storage array, storage disk, scheduler node, cluster services node, and provisioning node. Certain nodes, like login, data transfer, and web portal nodes, serve as access points and are con-

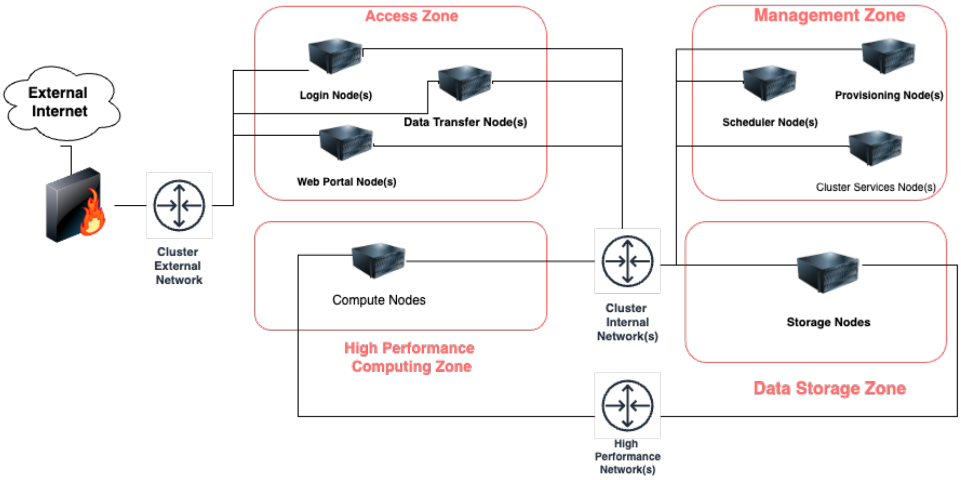

Figure 1: HPC Reference Architecture.

sidered assets. Compute and data storage nodes are vulnerable to compromise by malicious users seeking to run illicit jobs or access legitimate user data. In particular, to differentiate nodes with GPUs from those without, we added GPU nodes as asset types. Storage disks and arrays are critical assets, as compromising them could result in data loss or corruption, especially since HPC systems typically lack backup services. Each node of the management zone represent an asset: in particular, the scheduler node must be protected because its compromise might lead to scheduler tampering or an elevation of privileges; cluster services nodes are critical since they store log data and host crucial services such as LDAP. Provisioning nodes store node images and are similarly important. Communication networks constitute a valuable assets because a malicious user might attempt to compromise them to seek sensitive data or concretize various threats (e.g., topology disclosure); therefore, we have considered cluster external networks (typically the Internet network), cluster internal networks (typically Ethernet network), and high-performance network (typically InfiniBand network). Starting from the analysis of the services hosted within the infrastructure, we have identified other nine asset types such as DNS, because an attacker might exploit the DNS server with the aim to overload a target through DNS response traffic or generate a massive volume of requests for non-existent records, which can overload a recursive name server: as a result of this overload, the DNS server may respond with NXDOMAIN for these non-existent records, causing delays in DNS response times; DHCP, in fact a malicious user might assign all IP addresses available on the DHCP pool in order to prevent the assignment to legitimate devices; LDAP, because an attacker might attempt to list user accounts or organizational units within the LDAP directory or inject harmful code into LDAP queries in order to modify, or delete data; NTP, in fact an attacker might modify the time provided by the server causing synchronization problem; job scheduler (typically PBS or SLURM), because an attacker might get access to scheduler logs information to learn about currently running jobs and what jobs have users run in the past; container platform (e.g., Singularity), in fact an attacker might break out from a container to the underlying host in order to move to other containers from the host or perform actions on the host itself; distributed file system, because its manipulation or an improper configuration leads to any type of data compromise; “web service” since, representing any type of service provided via web, they are subjected to threats such as sensitive data exposure. This analysis allows us to identify all assets and expand our modelling technique (MACM). For readability purposes, the extension of the model is presented in Section 6 along with the case study.

5 SYSTEMATIC THREAT SEARCH

Once the assets have been identified and the model supports them and all possible interactions, our scope is to build a structured catalogue of threats affecting each HPC asset, considering both HPC components and the protocol involved. To ensure a comprehensive threat analysis, our approach relies on a Systematic Literature Review aimed at collecting security threats (i.e. Systematic Threat Search (Granata et al., 2023)).

5.1 SLR Protocol and Extraction

Consistently with Kitchenham et al. guidelines, (Kitchenham et al., 2009), we have conducted the SLR following three steps: Planning, Conducting, and Reporting. The Planning phase aims to create a protocol for querying various sources for articles with the goal of both including and excluding specific papers from the results to answer specific research questions. The Conduction phase involves applying the rules defined within the protocol to obtain a set of accepted papers that are suitable for addressing the research questions. The Reporting phase encompasses the documentation of the review's outcomes and the sharing of these results with potentially interested parties. In order to perform the first phase we have selected two different research questions:

- `RQ1`: What are the threats for an HPC system?
- `RQ2`: Which methodologies are used to produce a threat model of the HPC system?

To answer the mentioned questions it is necessary to individuate the appropriate papers from which the data must be extracted. The papers were collected through a keyword-based approach. It involves selecting specific keywords and then formulating one or more search queries. In particular, we have derived a specific query to answer RQ1 and RQ2.

```
((hpc OR "high performance computing")
AND ("system" OR "data center" OR
"architecture" OR "infrastructure"))
AND (threat AND (analysis OR model
OR modeling))
```

We opted for Scopus as our primary search engine due to its widespread usage and comprehensive coverage, which includes results from other common platforms such as Google Scholar, Springer, and IEEE Explore. To individuate the relevant papers that may answer the research questions, we have defined appropriate criteria: it is referred to as *Inclusion and Exclusion criteria* and they depend on the systematic literature review's purpose. The mentioned criteria are reported in Table 1.

The Conduction phase is by three steps: (i) study identification; (ii) selection; (iii) extraction. The initial step involves the identification of studies through the implementation of a search strategy. Therefore, we have used advanced search strings that rely on Boolean expressions; in particular, the previously established rules were applied within the mentioned digital library using its respective language. Additionally, sources of evidence (such as the paper related to HPC security technologies provided by PRACE, and other documents) were added to the comprehensive search results. As a result, we have obtained 106 papers. In the Selection step, the large number of studies is reduced through the criteria specified in the protocol. In particular, after reading and analyzing all abstracts, only the papers that meet the inclusion criteria are accepted. As an outcome of this step, out of the 106 starting papers, only 17 were selected as suitable for data extraction, while 89 papers were rejected according to the exclusion criteria. The objective of the final step is to extract data from the paper through meticulous reading. The reading of 17 extracted papers highlighted that only 9 met the inclusion criteria, while 8 met the exclusion criteria; so, 9 papers were selected to try to answer research questions. The outcomes of the systematic literature review have allowed us to answer the two research questions previously mentioned. Here is a more detailed description. The majority of the results describe the threats that affect a high-performance computing system highlighting which HPC zone is involved. Additionally, part of these also illustrates the possible attacks that a malicious actor can implement defining what CIA requirement is compromised. Each one of these papers also describes for which reason an HPC system can represent an interesting target for an attacker. What has just been said has allowed us to answer RQ1. It is important to stress that part of these results has provided us with an overview of high-performance computing systems describing architecture, the differences with a general purposes system, and other general concepts; furthermore, detailed insights into security recommendations, requirements, challenges, mechanisms, technologies, and enhancement methods have been acquired (Nowak, 2017) (Pleiter et al., 2021) (Hou et al., 2020b) (Yang et al., 2021). To answer RQ2, as already discussed in section 2, not-specific techniques have been adopted to derive threats of an HPC system. As an example, some authors listed all the threats affecting a supercomputer by considering the security requirements they compromise. Accordingly, some threats have been selected for confiden-

Table 1: Inclusion and Exclusion Criteria.

Inclusion Criteria	Exclusion Criteria
Proposes threat analysis/model for HPC system	It is not written in English
Describes threats for HPC system	Does not cover HPC security
-	Does not concern threat analysis for HPC system
-	Does not concern HPC security threats

Table 2: Threat Catalogue: Data Model.

Threat Catalogue Field	Description
Asset Type	The kind of asset
Threat	Threat that can affect an asset type
Description	Brief description of threat
STRIDE	STRIDE classification
Compromised	Considers indirect threats, which are threats that impact a particular component and are then transmitted to neighbouring components
PreCondition	How much confidentiality, integrity and availability have to be compromised in order to perform the threat
PostCondition	How much the threat compromises the confidentiality, integrity and availability

tiality, others for integrity and availability of services. The result is significant for us because our approach systematically derives threats from a well-structured model.

5.2 HPC Threat Catalogue

Our threat catalogue is a structured representation of the threats that may affect some assets. From the study of SLR selected papers we have built an Excel sheet including the threats related to all HPC assets. We have followed two steps to build the HPC Threat Catalogue: (i) collecting the threats for each selected asset; and (ii) enhancing each pair comprising an asset and a threat with the following data model fields: description, STRIDE, Compromised, PreCondition, PostCondition. Table 2 provides a description of the mentioned fields. The *Compromised* can either be *self* if it compromises the component itself, or it may follow a specific format: *role(relationship)*. The *role* field can be either *source* or *target* and determines whether the threat compromises the in-going or out-going connections originating from the component. Meanwhile, the *relationship* field acts as a filter for the type of relation through which the threat can propagate. The Precondition is articulated in the format of *[LossOfConfidentiality, LossOfIntegrity, LossOfAvailability]*, elucidating the extent to which the threat must exploit the CIA security requirements. Similarly, the PostCondition, presented in the same format, delineates the impact on each security requirement. Each compromising level is denoted by: n (no compromise), p (partial compromise), and f (full compromise).

As a result, a part of the threat catalogue is reported in the table 3. It is important to note that only a portion of the threat catalogue has been included in this paper, aligning with its length constraints. Interested readers may obtain the complete catalogue by reaching out to the authors. Also, the Precondition field is not taken into account since we considered it as *None* (no precondition required).

6 V:HPCCRI CASE STUDY

The case study is represented by the University of Campania Luigi Vanvitelli, with its own supercomputer: V:HPCCRI. It is composed of forty-two nodes and tree networks; in particular, there are two login nodes, two management nodes, two storage nodes, twenty-six compute nodes, and 10 GPU nodes. The nodes are connected through both an Ethernet network, chosen for its widespread adoption, cost-effectiveness, and compatibility, and an InfiniBand network, selected for its superior performance, low latency, and scalability, particularly in the HPC context. Also, a Broadcom (BCM) network is used to provide robust networking solutions with advanced features, scalability, and reliability, ensuring efficient data transmission and network management within the system architecture. The two login nodes are connected to an external network (i.e., a University public network), additionally, there is a firewall in front of them. The management nodes host virtual machines – connected to a VLAN – which in turn hosts some services (i.e., OpenLDAP, zChild, and xClarity). Furthermore, also other machines host services such as container platforms. The job scheduler system in place is PBS. GPFS is the distributed file system present within the infrastructure.

6.1 MACM Extension

Once assets typology has been identified, we extended our modelling technique, the Multi-Application Composition Model (MACM) (Casola, 2019) to support HPC components. Our model relies on a graph-based model characterized by nodes and edges: each node aims to describe a system's asset, and each edge represents the relationship that exists between two distinct assets. Each MACM node is defined by a primary label that identifies the component's class and an optional secondary label that provides additional details. The most important parameter in our model is asset type, defining the typology of the considered component. It is a mandatory label since it describes the functional behaviour of each component and, accordingly, can be associated with

Table 3: Part of HPC Threat Catalogue.

Asset Type	**Threat**	**Description**	**STRIDE**	**Compromised**	**PostCon**	**Source**
HW.PC. Login Node	Authentication Abuse	An attacker is able to access the node abusing the authentication system	S	self, target(hosts)	[p,p,n]	(Guo et al., 2023)
HW.PC. Cluster Services Node	Log Tampering	An attacker modifies and manipulates system logs or records	T	self, source(hosts)	[n,p,n]	(Mogilevsky et al., 2005)
Service. Job Scheduler	Scheduler tampering	An attacker gives their own job higher priority and/or modifies the legitimate users' job priorities	T, D	self, source(uses)	[n,p,p]	(Mogilevsky et al., 2005)

security issues. As a result of this phase, part of the new asset types related to HPC components is shown in Table 4.

6.2 System Modelling

We modelled the architecture described above using the MACM model, as shown in Figure 2. It's worth emphasizing that the figure doesn't include all modeled nodes, but rather focuses on the key ones essential for providing a clear overview of the model.

It is composed of 61 nodes. Each label influences the colour of the nodes, whereas attributes are not visible in the image. To provide a concise summary of our model, we included only the essence of the MACM relationships in Table 5. It is important to emphasize that we have decided to use the symbol * to refer to all nodes in the MACM that fall within a specific category. Therefore, for example, the expression *ComputeNode** highlights that the relationship defined in the table applies to all twenty-six computing nodes.

6.3 Threat Model Generation

The procedure for generating the threat model, as detailed in our prior work (Rak et al., 2022), involves selecting all threats impacting the system described by the MACM, forming a list of pairs denoted as *(CompromisedAsset, MaliciousBehaviour)*. The procedure relies on the data model outlined in section 5 and associates threats with each asset, considering parameters such as asset type, protocol, and role in communication, as well as the compromised field. Initially, threats related to the asset type are enumerated for each asset. For example, a *Service.Web* asset type has different threats compared to a *Service.DNS*. Protocols described by the in-going and out-going arcs are considered, using the direction of the edge in the MACM's directed graph to assign roles to assets involved in communication. For instance, if a client (CSC) communicates with a web application via the HTTP protocol, the MACM model adds HTTP attributes to the *uses* relationship, designating the CSC as the HTTP client and the application as the server. Once assets are classified as client or server, role filtering is applied based on the *Role* field, determining if a threat applies to the client, server, or both. The *Compromised* field considers indirect threats affecting a specific component and propagating to neighbours. It can be *self* if the component is compromised or has a template like *Role(relationship)*. The *Role* field, as explained, can be *source* or *target*, indicating whether the threat compromises in-going or out-going edges. The relationship applies a filter to the type of relation the threat can propagate. For example, *[self, source(uses)]* means the threat compromises the asset and all nodes using that asset, while *source(connects)* applies the threat to all networks connecting the asset.

It is worth noticing that our approach semi-automatically derives all threats from the MACM model.

6.4 Case Study Results

According to our case study, the assets are the ones already anticipated above and summarized in tables 4 and 5. Applying the threat modelling approach and, in particular, considering only the criteria reported below, we produced a lists of threats for each asset: *An asset Ai can be affected by a threat Ti if the Asset Type of Ai is the same of the Asset Type of Ti.* The resulting threat model is characterized by six fields: asset name, asset type, threat, CIA, STRIDE, and be-

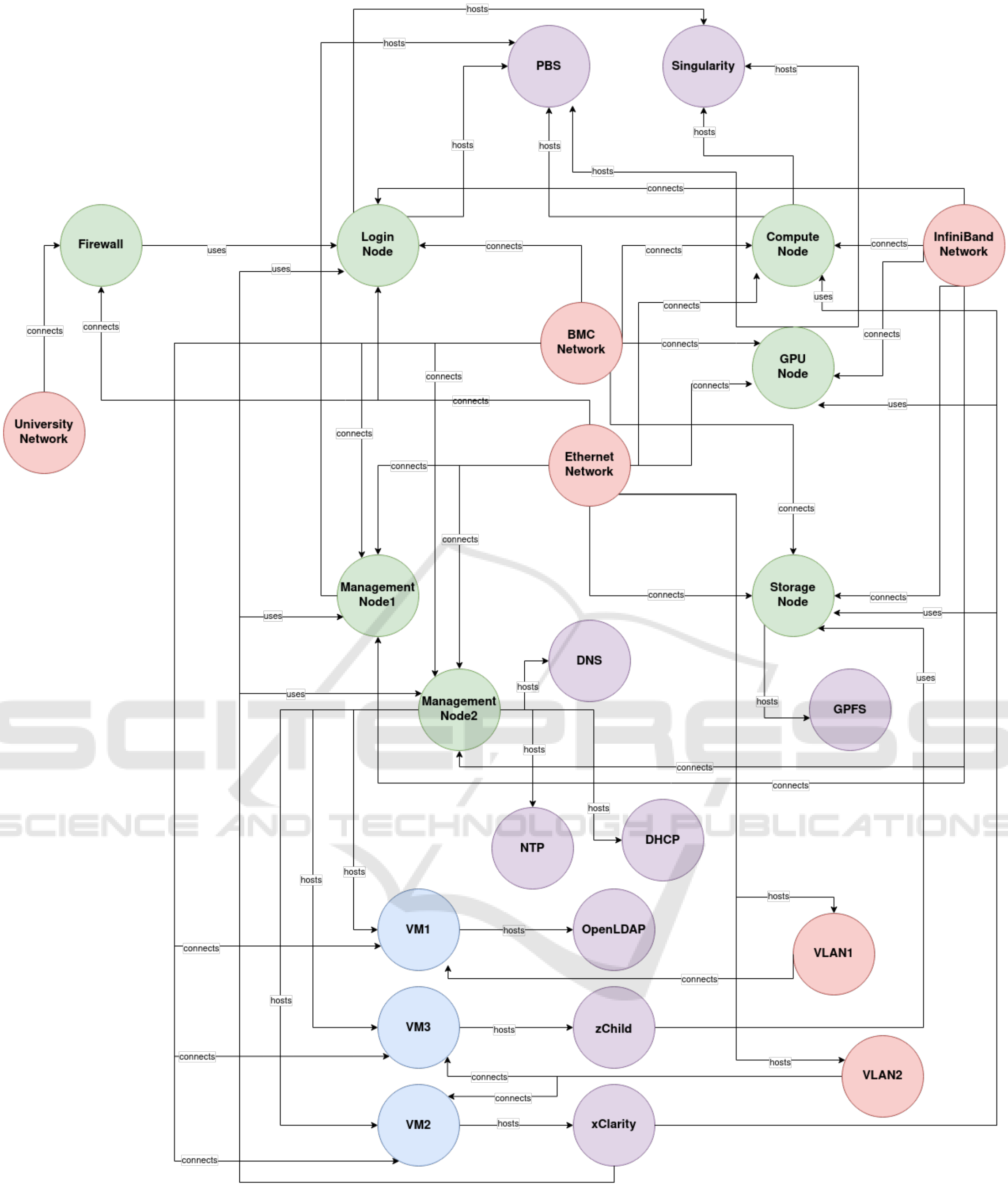

Figure 2: V:HPCCRI MACM.

haviour. “Asset name” represents the name associated with a specific V:HPCCRI asset; these assets are all reported in MACM model. “Asset type” represents the typology of assets that can be affected by a specific threat. “Threat” represents a label that defines the typology of threat. *CIA* reports the CIA requirements that are compromised by a threat. The *STRIDE* field indicates the STRIDE classification. “Behaviour” is a brief description of a specific threat. An abstract of the threat model is reported in table 6.

In our study, we approached protocol modelling by prioritizing the services they support rather than solely focusing on communication perspectives. Thus, we chose not to present examples of threats directly affecting these protocols (e.g. DNS, DHCP, NTP); instead, we concentrated on threats associ-

Table 4: Part of MACM Node Labels and Assets.

Primary Label	Secondary Label	Asset Type(s)	Description	Technology	HPCZone
HW	Server	HW.PC	A physical hosting hardware		
HW	Server	HW.PC.LoginNode	Node that provides login services		HPCAccessZone
HW	Server	HW.PC.DataStorageDisk	Disk storage		HPCStorageZone
HW	Server	HW.PC.SchedulerNode	Node that manages the HPC system		HPCManagementZone
HW	Server	HW.PC.ComputeNode	Compute node		HPCComputingZone
Network	LAN	Network.Wired.HPC	Local Access Network used in HPC system that guarantee high bandwidth and low latency.	InfiniBand, Omni-Path, Slingshot	
Network	LAN	Network.Wired.Ethernet	Local Access Network used in HPC system that aims to connect nodes		
service	SaaS	Service.DNS	Domain Name System Protocol		
service	SaaS	Service.LDAP	Lightweight Directory Access Protocol		
service	SaaS	Service.JobScheduler	Job Scheduler, the assets vary based on the technologies involved	PBS, SLURM, Torque	HPCAccessZone, HPC-ManagementZone, HPCComputingZone

Table 5: Part of relations between components in the case study.

Start Node	Relation	End Node
InfiniBandNetwork	connects	LoginNode*
EthernetNetwork	connects	LoginNode*
InfiniBandNetwork	connects	StorageNode*
InfiniBand Network	connects	GPUNode*
StorageNode	hosts	GPFS
ComputeNode	hosts	PBS
Management Node2	hosts	VM*
VM1	hosts	OpenLDAP
VM2	hosts	xClarity
VM3	hosts	zChild
EthernetNetwork	hosts	VLAN*
xClarity	uses	LoginNode*
zChild	uses	StorageNode*

ated with the services that utilize them. Some other threats have been selected considering the *Compromised* field, as already described. A part of the threat model referred to this parameter is shown in the table 7.

As an example, *User Session Hijacking* consist of the stealing of a session token to get unauthorized access to the system, compromising the PBS service. Since the LDAP Injection threat has *source(hosts)* in Compromised field, it compromises not only the service LDAP, but also the virtual machine hosting the service. Some threats can affect indirectly the network connecting the services. For example, the download of malicious content from the Login node can affect its communications. Different threats can target the network infrastructure, causing partitions that disrupt communication in certain segments, ultimately rendering them inaccessible. This type of threat also undermines the integrity of all nodes linked within the network. Other threats can affect the way containers are handled by Singularity. As an example, an intruder can breach the boundaries of a container, successfully accessing the underlying host to transition to other containers from the host or carrying out operations directly on the host. This can compromise each Node (i.e. Login, Compute and GPU) because of its virtualization mechanisms. Finally, our threat analysis revealed that the supercomputer is exposed to 164 distinct threats, with redundancy not factored in from the presence of multiple nodes and virtual machines. By considering each node and, therefore, each service hosted on the node, the number of threats increases to more than 1100. It is crucial to note that, an HPC system typically hosts a significant and variable set of services, depending on the demand. Accordingly, the potential for threats significantly escalates, providing ample opportunity for threat agents to launch attacks.

7 CONCLUSION

Since HPC systems originated in an academic and research environment, they may be considered as trusted and secure systems; actually, the reality is quite different. In fact, the heterogeneity of the resources that characterized them may extend attack vectors leading to a compromise of infrastructure in terms of confidentiality, integrity, or availability of services. To understand which are the security issues that affect HPC systems, this paper presented a methodology for collecting existing threats in the literature by conducting a Systematic Literature Review. In particular, our methodology allowed us to extend a graph-based modelling technique (MACM) and build an HPC-specific threat catalogue starting from a do-

Table 6: Part of Threat Model per Asset.

Asset name	Asset type	Threat	CIA	STRIDE	Behaviour
InfiniBand Network	Network.Wired. HPC	Key Tampering	I, A	T	An attacker tampers the key used by devices
InfiniBand Network	Network.Wired. HPC	Topology Disclosure	C	I	An attacker can exploit fofwarding updates between the various nodes to know network topology
Management Node 1	HW.PC. Scheduler Node	Elevation of privileges	C, I, A	E	An attacker is able to change its privileges in access to the system services and data
OpenLDAP	Service. LDAP	Directory Enumeration	C	I	An attacker attempts to list user accounts or organizational units within the LDAP directory
DNS	Service. DNS	NX Domain	C, I, A	D	Attacker floods server with requests, leading to DNS delays and ”NXDOMAIN” responses for non-existent records due to overload

Table 7: Some Threats due to Compromised Field.

Threat	Post Condition	STRIDE	Asset	Due to
User Session Hijacking	[p, p, n]	S	PBS	Login Node
LDAP Injection	[n, p, n]	T, D	VM1	OpenLDAP
Download Malicious Content	[p, p, n]	S	Each Network	Login Node
Network Partitioning	[n, p, p]	D	Each Node	InfiniBand network
Contained Escape	[p, n, n]	I, E	Each Login Node, Each Compute Node, Each GPU Node	Singularity

main analysis (i.e. the identification of the primary hardware and software component types, and protocols) based on the reference architecture proposed by NIST. Accordingly, using the threat catalogue, we produced a fine-grained threat model for V:HPCCRI real case study. As a result, our research highlighted that this supercomputer can be affected by at least 164 different threats. In this regard, it is worth noting that the work carried out is preliminary, as the current security level of the case study has not been determined, and no security measures currently implemented have been identified. Additionally, potential security controls necessary to enhance the security level of V:HPCCRI have not been considered. Therefore, as future work, we plan to conduct a security assessment of the case study using specific methodologies tailored for high-performance computing systems (assuming they exist) to individuate the security level of the considered supercomputer, identify implemented security measures, and assess the need for additional security controls. At this point, we plan to perform a risk analysis for all the threats to which each identified asset is exposed to understand if and which security countermeasures should be applied and in which order, since it’s crucial to implement security controls, but not every single one is necessarily mandatory.

ACKNOWLEDGEMENTS

This work was partially supported by the UPSIDE project (B63D23000820004) and the project DEFEDGE (E53D23016380001) under the PRIN program funded by the Italian MUR.

REFERENCES

Casola, V. (2019). Toward the automation of threat modeling and risk assessment in IoT systems. *Internet of Things*, page 13.

Granata, D. and Rak., M. (2021). Design and development of a technique for the automation of the risk analysis process in it security. In *Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER,*, pages 87–98. INSTICC, SciTePress.

Granata, D. and Rak, M. (2023). Systematic analysis of automated threat modelling techniques: Comparison of open-source tools. *Software Quality Journal*.

Granata, D., Rak, M., and Mallouli, W. (2023). Automated generation of 5g fine-grained threat models: A systematic approach. *IEEE Access*, 11:129788–129804.

Granata, D., Rak, M., and Salzillo, G. (2022). Risk analysis automation process in it security for cloud applications. In Ferguson, D., Helfert, M., and Pahl, C., editors, *Cloud Computing and Services Science*, pages 47–68, Cham. Springer International Publishing.

Guo, Y., Chandramouli, R., Wofford, L., Gregg, R., G. Key, A. C., Hinton, C., Prout, A., Reuther, A., Adamson, R., Warren, A., Bangalore, P., Deumens, E., and Farkas, C. (2023). NIST Special Publication 800-223. White Paper, NIST.

Hou, T., Wang, T., Shen, D., Lu, Z., and Liu, Y. (2020a). *Autonomous Security Mechanisms for High-Performance Computing Systems: Review and Analysis*, pages 109–129.

Hou, T., Wang, T., Shen, D., Lu, Z., and Liu, Y. (2020b). *Autonomous Security Mechanisms for High-Performance Computing Systems: Review and Analysis*, pages 109–129.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

Mogilevsky, D., Lee, A., and Yurcik, W. (2005). Defining a comprehensive threat model for high performance computational clusters.

Nowak, M. (2017). Security in hpc centres.

Pleiter, D., Varrette, S., Krishnasamy, E., Özdemir, E., and Pilc, M. (2021). Security in an evolving european hpc ecosystem.

Rak, M., Salzillo, G., and Granata, D. (2022). Esseca: An automated expert system for threat modelling and penetration testing for iot ecosystems. *Computers and Electrical Engineering*, 99:107721.

Yang, B., Yu, Y., Wang, Z., Li, S., Xiao, H., Gao, H., Liang, Z., and Zhou, X. (2021). Research on network security protection of application-oriented supercomputing center based on multi-level defense and moderate principle. *Journal of Physics: Conference Series*, 1828(1):012114.