A SECURITY ARCHITECTURE FOR HIGH PERFORMANCE COMPUTING FACILITIES

Peter McMahon and Andrew Hutchison

{pmcmahon,hutch}@cs.uct.ac.za

Department of Computer Science, University of Cape Town, Rondebosch 7701

ABSTRACT

High Performance Computing facilities that use cluster computing to provide computational services to scientists and engineers have become widespread, with such facilities available at most major research universities worldwide, as well as in government and industrial research settings. Until recently HPC facilities have largely neglected security, or at the very least treated security as an afterthought in a world where performance is the number one priority.

In this paper we present a security architecture, and associated security best practices, for high performance computing facilities. Our architecture aims to address concerns about HPC security raised in the literature [1-4], and specifically to mitigate the risks identified in the leading threat model for cluster computing [5], developed at the National Center for Supercomputing Applications.

Our architecture and best practices attempt to provide the best tradeoff between adequate security measures and high performance. We also consider privacy and intellectual property issues, and how shared HPC facilities may put measures in place to convince their users that their data is protected from both external threats and internal, authorized cluster users not affiliated with their projects.

KEY WORDS

High Performance Computing, Cluster Computing, Network Architecture, Intellectual Property, Access Control

A SECURITY ARCHITECTURE FOR HIGH PERFORMANCE COMPUTING FACILITIES

1 INTRODUCTION

High performance computing (HPC) facilities have seen increasing deployment in recent years, and in particular the rise of HPC facilities using clusters has been remarkable. Since 1998, when cluster computers first broke into the top 500 supercomputer list [6], their growth has been enormous: clusters now account for 72% of the top 500 supercomputers worldwide. Since cluster computers typically use standard PC hardware, and modified versions of standard desktop and server operating systems, they offer much better value for money than their proprietary, custom-designed supercomputer contemporaries. However, their use of standard hardware and software also makes them vulnerable to standard security attacks. Yurcik et. al. in [1] also argue that cluster security has "emergent properties", and should be studied in its own right.

In the high performance computing community the primary goal for the design of a high performance computing facility is to obtain the maximum performance within the allowed budget. All other considerations are secondary. This mentality is exacerbated by the fact that many users, and managers, of HPC facilities are not computer scientists, but rather engineers and scientists from other disciplines who rightly view such facilities as tools, rather than as subjects for research in their own right. Much research in computer science has been done on the performance aspects of computational clusters, but until recently [1, 5] the field has largely neglected security.

In [5], Mogilevsky et. al. provide a comprehensive threat model for HPC clusters. We present a proposed architecture that aims to mitigate many of the risks outlined in their model, whilst minimizing the effects on the performance of the cluster.

This paper is organized as follows: we first review previous work relating to threats to HPC facilities. We then introduce a proposed architecture, and explain our design choices. We offer best practices relating to the setup and maintenance of various sections in the architecture, where appropriate. Finally, we briefly review the existing tools that are available to aid in implementing the proposed architecture.

2 THREATS TO HIGH PERFORMANCE COMPUTING CLUSTERS

HPC clusters are attractive targets to attackers for several reasons. Since clusters by their very nature have impressive processing capabilities, an attacker may wish to illegally take advantage of this power. Mogilevsky et. al. [5] give the example of brute force password cracking. However, other uses of the computational power of a cluster that might appeal to attackers can certainly be imagined; for example, digital video encoding is computationally expensive and an attacker could use a cluster to encode counterfeit videos. HPC clusters also typically have very large storage facilities, as well as high bandwidth Internet connections. Together these features make HPC clusters attractive targets for attackers to set up illegitimate FTP servers on. An effective Denial-of-Service attack could also be launched from a compromised cluster [5]. In 2004 attackers struck the TeraGrid network [7], which served as a reminder that research facilities are valued targets, and are vulnerable.

Mogilevsky et. al. [5] proceed to define a threat model for HPC clusters in three steps. Broadly, they identify assets in HPC clusters, identify potential entry points for attack and finally describe what attacks can be launched to gain illegitimate access to each asset. They use the Confidentiality-Integrity-Availability (CIA) threat classification. The following are identified as assets: user login data, user job data, system logs, scheduler, storage systems, intranode network fabric, computing cycles and network packets. The entry points are identified as: vulnerabilities in

remote login systems (e.g. SSH), remote cluster management software, open ports, stolen login information and rootkits.

Confidentiality is important to most users, but is a crucial concern in shared HPC facilities that make their resources available to industry. The WestGrid facility in Canada has noted that some companies are reluctant to used shared facilities such as theirs, citing competitive concerns – companies working on the same problem understandably don't want to share computing resources for fear of corporate espionage [8] unless they have guarantees on their data safety. For HPC centres wanting to serve industrial clients, ensuring confidentiality of user data is just as important as securing hardware resources.

Mogilevsky et. al. [5] identify five areas where confidentiality may be compromised: snooping on the internal and external networks to illegitimately obtain data and control packets; scheduler compromise, resulting in access to scheduler logs; direct access to computational nodes, allowing the attacker to launch jobs without using the scheduler, and access to the cluster's storage resources.

Integrity is similarly important in HPC centres: just a small amount of corrupt data can render thousands of hours worth of computations worthless, or worse still, cause erroneous results that the user assumes are correct. Mogilevsky et. al. identify several means by which an attacker can affect the integrity of the cluster: internal network packet injection, whereby packets with incorrect data are sent to nodes in the network; log tampering, effectively allowing the attacker to modify users' quotas, and data tampering, whereby an attacker with access to the storage resources can modify user data.

Ideally one would like an HPC facility to have 100% availability, but even though this might not be possible due to scheduled and unscheduled maintenance, it is certainly undesirable to have a cluster unavailable as a result of an attack. Besides "conventional" Denial-of-Service attacks on a cluster's outward-facing scheduling mechanism, a sufficiently successful attacker can also cause reduced availability by the following methods [5]: exhausting log space; exhausting space allocated per node for temporary data; exhausting storage space, and scheduling junk jobs to run on the cluster, thus wasting cycles while legitimate users wait for their jobs to be scheduled.

Despite the considerable threats against HPC clusters, and the importance of their resources, there is very little guidance in the literature on how to architect clusters with an eye towards security, nor what specific security measures can be put in place to mitigate the risks posed by the threats.

3 A PROPOSED SECURITY ARCHITECTURE AND RECOMMENDED SECURITY PRACTICES

When designing a computer network, security should ordinarily be taken into account, so calling a network architecture a "security architecture" is a tautology. However, as we have already noted, research in cluster and high performance computing has been focused on performance, so we denote our proposed architecture for a cluster as a "security architecture" to emphasise our focus on creating a network architecture for HPC clusters that addresses some of the security issues relevant to such networks.

Figure 1 shows our proposed architecture for a heterogeneous computational cluster. We have relied heavily on the principle of security-by-isolation, and encourage the use of access control mechanisms wherever possible.

Users are separated from the cluster by a firewall, and only have direct access to a server hosting a web-based front-end for submitting jobs, obtaining results and viewing billing details. The firewall will block the user from accessing any machines other than the web server, and on any ports other than those required for web access, and possibly temporary file transfer, as the web

server acts as an intermediary between the user and the storage server when the user uploads input data or downloads output results from the cluster.

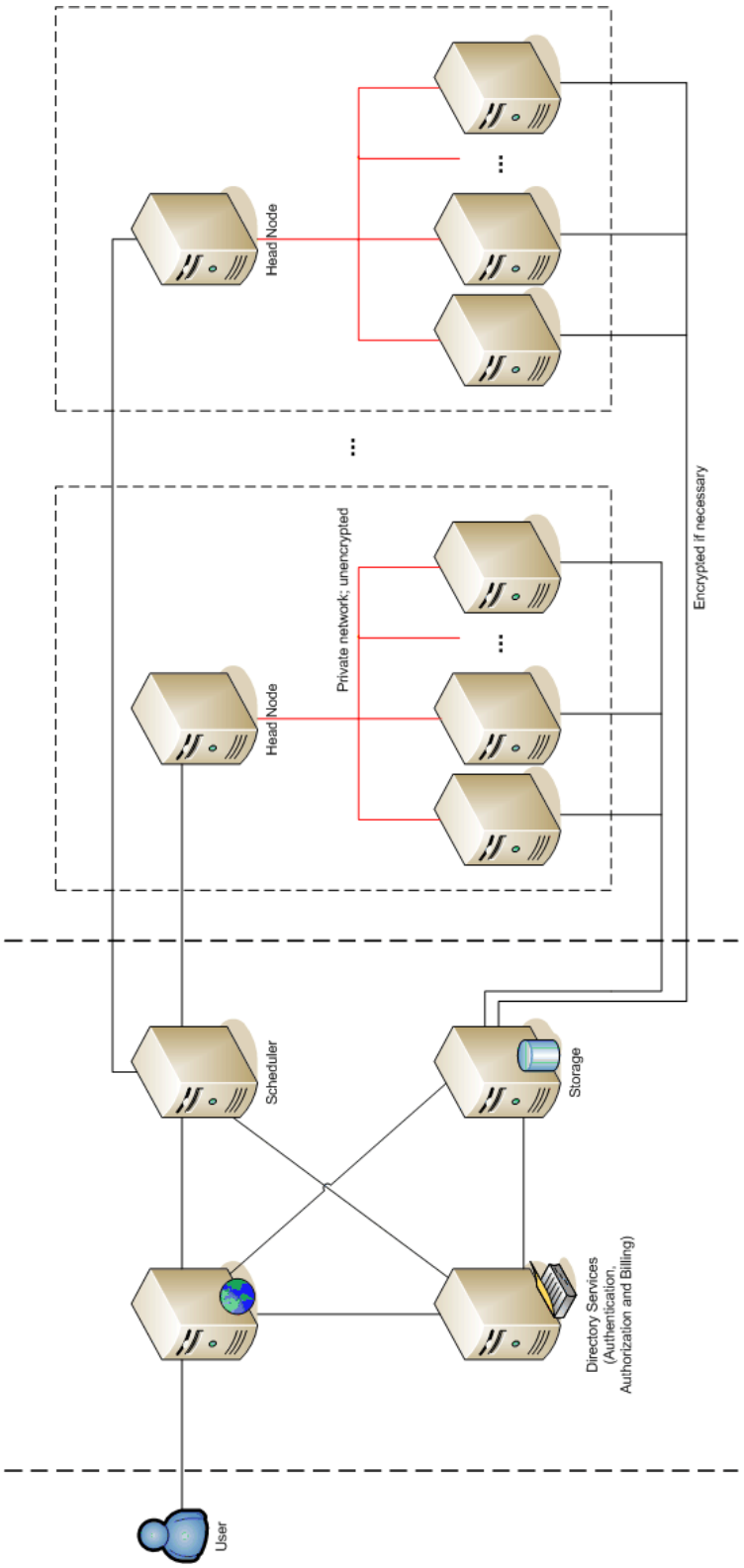

Figure 1. A security architecture for a computational cluster

Users will have SSL-protected sessions with the web server, thus mitigating the risk of external attackers snooping on the public networks to view packets sent between users and the cluster facility. All users using the web server will be required to authenticate first. Any user request mediated by the web front-end would require that the user has sufficient access rights and is thus authorized to perform the requested operation. One server is responsible for authentication, authorization and billing. Its connections with the other servers on the internal network will be encrypted. An example of a similar system in deployment at Fermilab in Chicago is the use of Microsoft Windows Active Directory to store user data, and the use of Kerberos to authenticate users [9]. All Fermilab users are issued with certificates, and no passwords are sent over the network. Certificates are given short lifetimes based on the premises that if they are kept for to long, they have an increasing risk of being tampered with or otherwise cracked, and that generating certificates can be done fast enough to serve a large number of users using this scheme.

Often large HPC facilities have several clusters, each using a different machine architecture. For example, there may be an x64 cluster, and a cluster of Sun UltraSparc RISC machines. In general clusters consisting of different machine architectures aren't used together. We advocate isolating the various clusters within an HPC facility: two clusters are shown in figure 1, and both have completely separate connections to the scheduler and storage. This may help to reduce the risk that if one cluster is compromised due to a security problem for that particular set of machines and their software, the rest of the facility is vulnerable.

Network traffic between infrastructure equipment in a facility (i.e. front-end, scheduler, directory services, storage and head nodes) can generally be encrypted without incurring an unreasonable performance penalty. Even an attacker who has gained some limited access to the internal networks will then not be able to breach confidentiality by snooping. However, encrypting traffic between the compute nodes in the clusters will typically result in unsatisfactory performance – parallel programs often make heavy use of message passing communication between compute nodes, and so the overhead of encrypting each message will dramatically reduce efficiency. Of course, "embarrassingly parallel" programs (those that are easily divided into parallel tasks, with little or no dependence amongst those tasks) may not require much inter-node communication and so encrypting traffic may be viable, and it may be useful to allow the concerned user the option of encrypting inter-node traffic. However, given that encryption may not be viable, the next best option is to isolate the network fabric between compute nodes and the head node. Set up the interconnect system as a private network: this way a snooping attacker will have to have access to one of the compute nodes, or the head node, to view traffic sent between these machines. If this is the case, the fact that the attacker can sniff traffic will be a lesser worry.

During parallel computations, compute nodes typically provide a "scratch space" where local processes can store data. Depending on how often the processes need to access or write to this storage space, it may be viable to encrypt data in the scratch space (perhaps by making use of built-in encrypted file system features in the node's operating system). However, the effect of this local encryption on performance will be highly dependent on the behaviour of the executing program. Users can choose to use an encrypted scratch space if they are sufficiently concerned about the privacy of their data. As noted in figure 1, it may also be viable to encrypt the traffic between compute nodes and the central storage server(s).

The storage server is key and needs to be properly protected. At a minimum, each user should have an isolated space on the server and only be allowed access to his files. However, this will likely not satisfy users with competitive concerns: giving such users the option to encrypt their data is essential.

To make the storage space in a cluster facility less attractive to attackers we suggest limiting the number of connections allowed to access files in any single directory. If an arbitrary limit of, say, 10 connections per directory are allowed, this should be enough that legitimate users aren't hindered, but attackers can no longer viably use the cluster storage resources as a means for

distributing material to many other parties. The storage server should be configured to only allow access to the web server and compute nodes, and not external machines. The requirement that storage requests be mediated by the web front-end may make attacking and subverting the storage server more difficult.

Another crucial component that needs adequate protection is the scheduler, and the scheduling capability. External attackers may wish to steal CPU cycles by illegitimately accessing the scheduler, and internal users may wish to subvert the billing mechanism to steal cycles. The scheduler is hidden from the outside world by a firewall, as is the rest of the facilities computing equipment. It should require that the user impersonated by the web front-end server, through which users submit jobs, be properly authenticated by the appropriate server. Job submissions should only be accepted by the scheduler from the web server. Likewise the head nodes should only accept instructions from the scheduler.

4 EXISTING TOOLS AND TECHNOLOGIES FOR SECURING HPC CLUSTERS

There are many existing tools designed for securing "ordinary" networks that are appropriate for use with HPC clusters. In addition, tools designed to enable secure "grid computing" can often be readily used in a cluster environment.

Standard firewall, directory services, encrypted file system, web server and network management software is readily available for computers used in clusters. Almost by definition clusters in recent years have used standard hardware, and so no special modifications to much of the software made for standard PC's is necessary. In addition, the Globus Toolkit [10], developed as a basic piece of infrastructure for grid computing networks, is well-suited to providing several mechanisms necessary in an HPC cluster facility: communication, authentication, network information and data access. Furthermore, Globus is designed to make extensive use of web services, exposing services in this way, which makes it a suitable choice for partially implementing our web server front-end suggestion. While Globus may not cover all areas, its use can certainly jumpstart the development of systems when setting up a cluster facility.

5 CONCLUSION

A cluster can never be made completely secure without crippling its functionality. However, we believe our proposals represent a reasonable compromise between security, performance and user convenience. Many of the standard techniques and tools for security in standard computing environments can be easily and appropriately adapted to secure aspects of cluster facilities. However, instead of simply focusing on securing individual machines and treating the cluster merely as a network of computers, if one takes into consideration the broader view of a cluster and its functionality, more appropriate choices for security mechanisms to be used can be made. Our proposed architecture is designed based on this premise, yet also allows for the use of standard security tools, and those being developed more recently for Grid computing, in its implementation.

6 ACKNOWLEDGMENTS

The authors wish to thank Radha Nandkumar at the NCSA, Jana Makar and Paul Wellings at WestGrid, and Bruno Schulze at the National Lab for Scientific Computing (LNCC) for helpful discussions and suggestions. They also wish to thank two anonymous reviewers for their comments.

7 REFERENCES

[1] Yurcik, W., Koenig, A., Meng, X. and Greenseid, J. "Cluster Security as a Unique Problem with Emergent Properties: Issues and Techniques". *5th LCI International Conference on Linux Clusters*, Presentation, May 2004.

[2] Yurcik, W., Lee, A., Koenig, A., Kiyanclar, N., Mogilevsky, D. and Treaster, M. "Cluster Security Research Challenges". *Cluster Computing Infrastructure Experience Workshop*, July 27, 2005.

[3] Naqvi, S. and Riguidel, M. “Threat Model for Grid Security Systems”. *European Grid Computing Conference 2005*.

[4] Pourzandi, M., Gordon, D., Yurcik, W. and Koenig, G. “Clusters and Security: Toward Distributed Security for Distributed Systems”. *IEEE Cluster Computing and Grid (CCGrid)*, 2005.

[5] Mogilevsky, D., Lee, A. and Yurcik, W. “Defining a Comprehensive Threat Model for High Peformance Computational Clusters”. Preprint: arXiv:cs.CR/0510046. 16 October, 2005.

[6] Top500.org. “Charts for November 2005”. Available at http://www.top500.org.

[7] Associated Press. “Hackers breach powerful research networks”. Available at http://www.teragrid.org/news/apps/0404/chicagonews.html. 15 April, 2004.

[8] Thibodeau, P. “Supercomputing pushes toward the corporate IT mainstream”. Available at http://www.techworld.com/opsys/features/index.cfm?featureid=2008&inkc=0. 28 November, 2005.

[9] Skow, D. “Use of Kerberos-Issued Certificates at Fermilab”. *GridWorld/Global Grid Forum 15, 2005*.

[10] Foster, I. “Globus Toolkit Version 4: Software for Service-Oriented Systems”. *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS 3779, pp 2-13, 2005.