01 Jun 2007

Survey of Supercomputer Cluster Security Issues

George Markowsky
Missouri University of Science and Technology, markov@mst.edu

Linda Markowsky
Missouri University of Science and Technology, markowskyl@mst.edu

Recommended Citation

G. Markowsky and L. Markowsky, "Survey of Supercomputer Cluster Security Issues," *Proceedings of the 2007 International Conference on Security and Management, SAM'07 (2007, Las Vegas, NV)*, pp. 474-480, CSREA Press, Jun 2007.

Survey of Supercomputer Cluster Security Issues

G. Markowsky
Department of Computer Science
University of Maine
Orono, ME 04469-5752, USA

L. Markowsky
Department of Computer Science
University of Maine
Orono, ME 04469-5752, USA

Abstract **-** The authors believe that providing security for supercomputer clusters is different from providing security for stand-alone PCs. The types of programs that supercomputer clusters run and the sort of data available on supercomputer clusters are fundamentally different from the programs and data found on stand-alone PCs. This situation might attract a different type of attacker with different goals and different tactics. This paper discusses the results of a questionnaire sent out to many supercomputer clusters in the United States and relates them to a literature search that was also undertaken. These results suggest approaches that can be taken to further secure supercomputer clusters.

Keywords: Supercomputer, cluster, security, questionnaire.

1 Motivation

As a result of working in cybersecurity and working on some classified supercomputer computations, we became curious about whether the security problems for supercomputer clusters were different from those faced by desktops. Some of the reasons for suspecting that there might be differences are:

- different types of data, including classified data;
- interest on the part of governments;
- access to great computing power;
- the greater sophistication of users;
- the greater sophistication of attackers.

Additionally, it is clear that supercomputer cluster operators are aware of each other, so the supercomputer cluster network might be a tempting target. Our goals were to get some insight into the state of the art in cluster security and to help cluster operators secure their clusters.

1.1 The Stakkato Intrusions

In the course of doing the research for this paper, we came across a discussion of the "The Stakkato Intrusions." A very detailed discussion of this protracted attack against supercomputer clusters can be found in Nixon [1]. Quoting from the abstract we have: *During 15 months, from late 2003 until early 2005, hundreds of supercomputing sites, universities and companies worldwide were hit in a series of intrusions, with the perpetrator leapfrogging from site to site using harvested ssh passwords. The damage has been estimated to exceed $100 million in the United States alone.* The intrusions were eventually traced to a Swedish teenager who was visited by police. After the visit the intrusions stopped. All together, approximately 1,000 sites were compromised to some degree.

2 Some Questions

Some of the questions that we were hoping to get some insight into were:

- What is the level of computer security expertise shown among cluster operators?
- To what extent are clusters targeted by organizations rather than random hackers?
- How common are physical or social engineering attacks?
- How sophisticated are the attackers?

3 The Survey

In designing a survey we worried about several factors. Of particular concern to us were the following:

- The survey must not be intrusive.
- We must get people to trust us enough to complete the survey.
- The survey must not reveal weak spots to potential adversaries.
- The survey must be short and easy to complete.
- The survey should preserve anonymity.

While we do not believe in security through obscurity, we also did not want to call attention to any particular institution because of our work.

We were very concerned that we would not be able to get enough responses to provide some interesting results, but we were able to get 61 valid responses along with nine e-mails discussing various aspects of the survey. Everyone who responded either sent an e-mail or filled out the questionnaire that we made available on the password protected website: http://www.cs.umaine.edu/~markov/clustersurvey/survey.html. We made no effort to track the source of the responses since we promised to keep all responses anonymous.

3.1 Pool of Potential Participants

We decided on a two-pronged distribution scheme. First, we created a list of e-mail addresses of supercomputer administrators and users. This took some time since people are taking more care to keep their e-mail addresses private. All told, we sent out well over 250 e-mails to various people. The addresses were all distinct, but in some cases we sent e-mail to more than one person at a location because we were not sure who would respond. Figure 1 shows the distribution of the e-mail addresses that we carefully tracked.

edu	159	nl	1	net	1
de	8	ru	2	ua	1
tr	1	ch	1	mil	4
gov	21	org	4	gr	1
au	13	hk	1	ie	1
ca	17	dk	3	us	4
be	1	uk	2	com	1

Figure 1: Distribution of the 247 e-mail Addresses

Second, we posted our questionnaire on various USENET user group sites and used the beowulf.org mailing list as well as some securityfocus.com mailing lists.

3.2 Responses

We were pleased to receive 61 completed and valid questionnaires before writing this paper. Completed questionnaires consisting entirely of "No Answer" responses were omitted from our results, which are presented below as a series of pie charts.

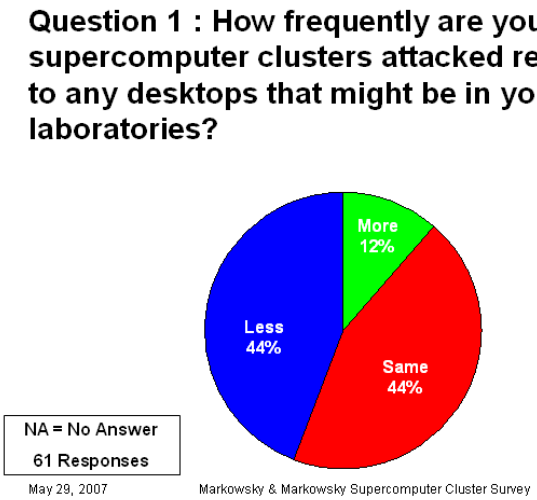

Figure 2: Question 1 Responses

Questions 1 through 5 are fairly straightforward and the responses are also easy to interpret, so we will not discuss them further.

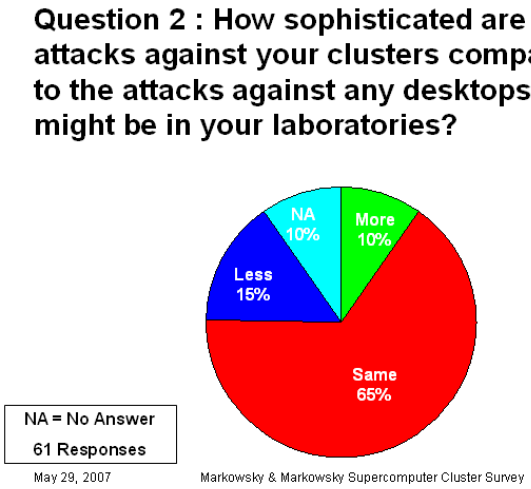

Figure 3: Question 2 Responses

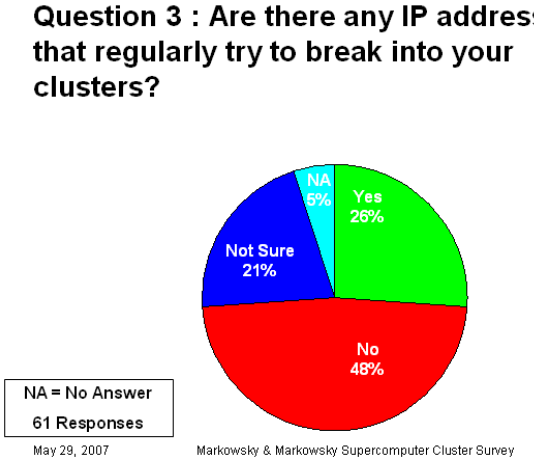

Figure 4: Question 3 Responses

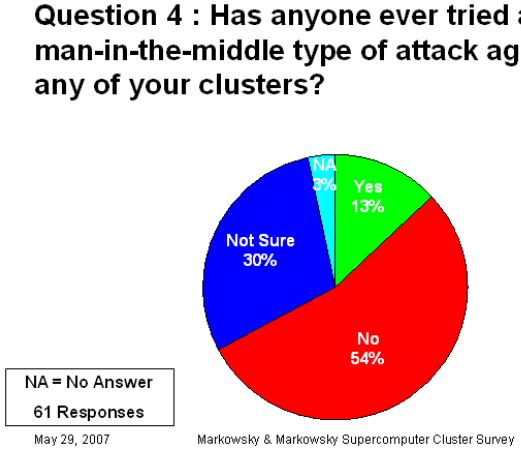

Figure 5: Question 4 Responses

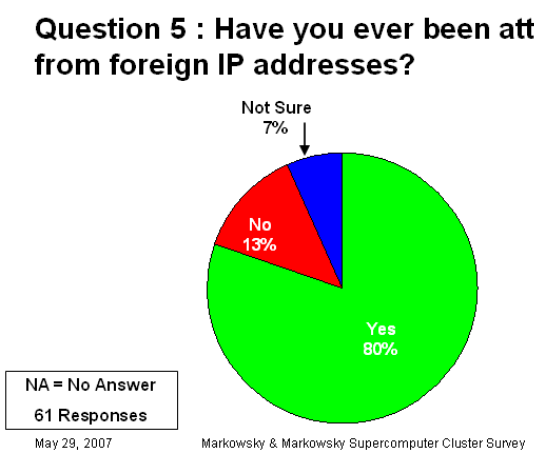

Figure 6: Question 5 Responses

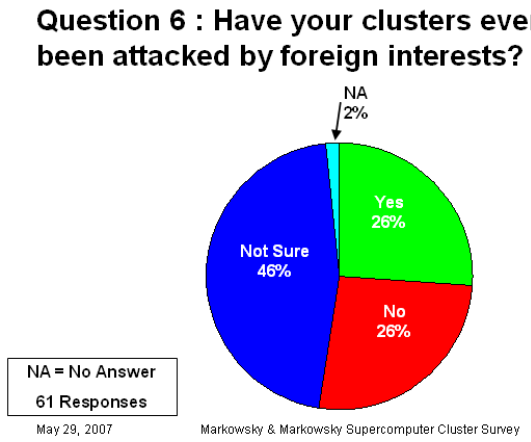

Figure 7: Question 6 Responses

We thought that the overwhelming majority would answer "Not Sure" to Question 6, since we were not sure how people would determine if they had been attacked by foreign interests. It is clear that the majority of respondents feel confident enough to give a "Yes" or "No" answer. This suggests that it is worthwhile to investigate this question further.

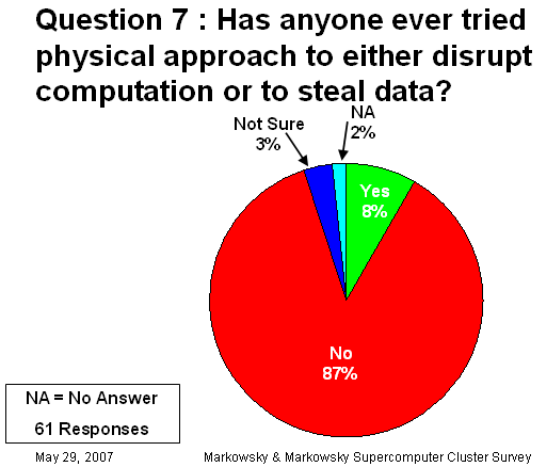

Figure 8: Question 7 Responses

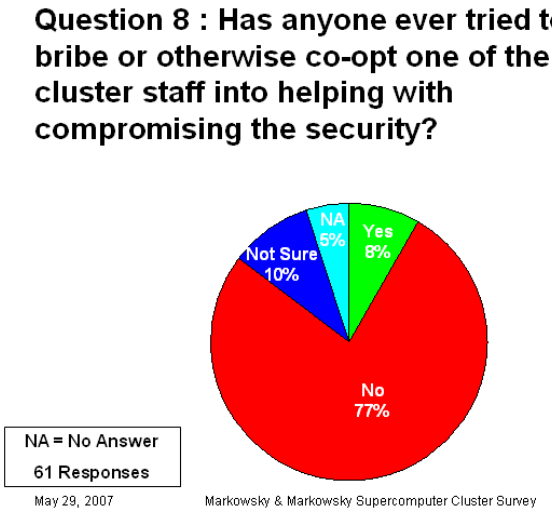

Figure 9: Question 8 Responses

We noticed that the same number of respondents (5) said "Yes" to Question 7 as said "Yes" to Question 8. We looked at the responses to see whether they were the same people. It turns out that only 2 people said "Yes" to both questions. Thus, a total of 8 respondents out of 61 either encountered a physical attempt to breach security or an attempt to co-opt one of the cluster operators.

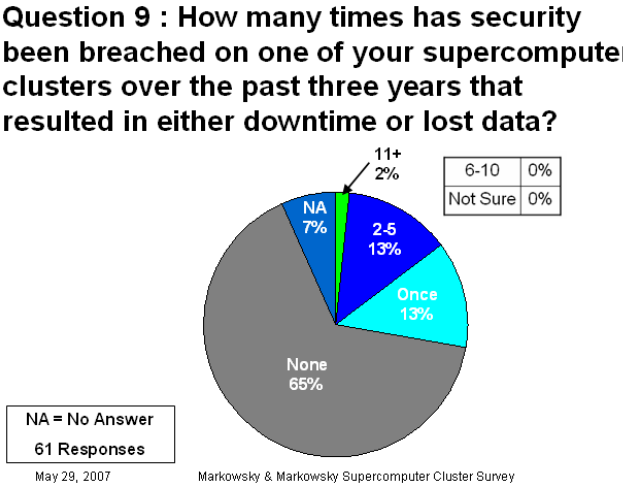

Figure 10: Question 9 Responses

It was interesting to note that no respondents reported between 6 and 10 security breaches, but that one respondent reported 11 or more incidents that resulted in downtime or lost data.

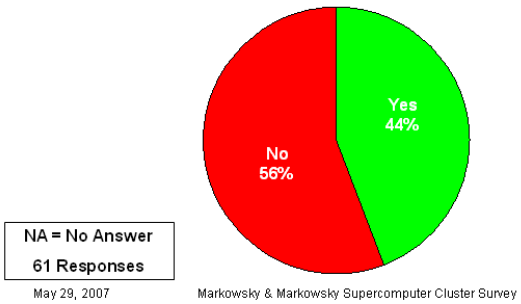

Figure 11: Question 10 Responses

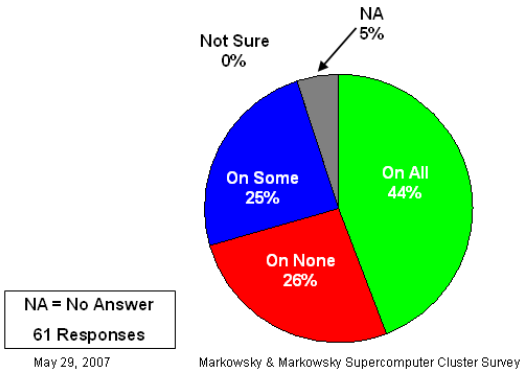

Figure 12: Question 11 Responses

We were surprised to learn that at least a quarter of the respondents are not running an intrusion detection system on their clusters. It seems that another 25% are only running intrusion detection systems on some of their clusters. We recommend that all clusters run an intrusion detection system of some kind.

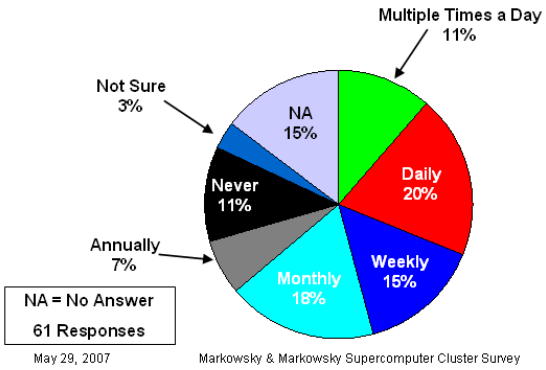

Figure 13: Question 12 Responses

We were surprised that only a third of the respondents stated that they checked for rootkits at least on a daily basis. The Stakkato Intrusions were characterized by the installation of rootkits on many servers.

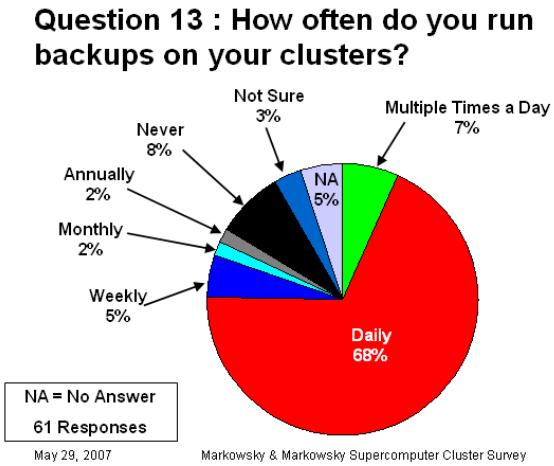

Figure 14: Question 13 Responses

The situation for backing up seems to be much better than checking for rootkits since 75% of respondents stated that they backed data up at least daily.

3.3 Some Letters

As we noted earlier, we also received nine e-mail letters in response to our questionnaire. We classified the letters into four categories as illustrated in Figure 15. Two letters wanted to verify that we were indeed the source of the questionnaire. One letter notified us of a duplicate request from us. Two letters stated an interest in receiving copies of the results. Finally, four letters offered some comments on the questionnaire and the issues of cluster security. We will discuss these four letters in greater detail.

Type of Letter	Number
Verification	2
Note of Duplication & Support	1
Interest in Results	2
Comments & Suggestions	4

Figure 15: Classification of Letters Received

Letter 1:

I believe that the most important question is missing: "Have you classified or proprietary data on your cluster?" Otherwise the cluster is "not worth attacking". A normal university or hobby cluster will be defended differently than a sensitive one. Of course the question itself is sensitive.

We were reluctant to ask questions of this type because we did not want people to think that we were footprinting their sites. Furthermore, we feel that all clusters are worth attacking because they can provide a trusted base for further attacks as was done in the Stakkato Intrusions. Finally, all clusters are worth attacking since they can provide a lot of computational power for such activities as password cracking. For this reason we believe that all clusters deserve to be defended vigorously.

Letter 2:

I would like to point out that the use model for clusters is what drives most of the important security risks and is the primary threat. Laptop/desktop and standard commercial activities do not normally involve giving a large number of users shell access. Very few HPC systems run through portals where the OS is isolated from the user.

So the starting point for a cluster is where the 85% of the standard security stops (don't let the bad guy get a shell). Compromised user desktops will immediately lead to a user level compromise of the cluster. If you don't manage the user desktops (they might not be yours) you must assume that there is an account compromise.

The important question is what do you do then?

The problem of protecting internal resources from compromised user desktops/laptops is very serious. In some organizations it has led to the creation of internal firewalls to protect internal resources from users. We also feel that a variety of steps can be taken to protect internal resources even if user computers are compromised.

Even if a keylogger has been installed on the desktop, there are ways of protecting the cluster. For example, users could be required to use key-based authentication, with the key kept on removable media, not on the hard drive. Alternatively, systems can require keys together with one-time passwords. In such cases, user-level compromise is still possible, but it would hardly be "immediate". Another possibility is the Trellis system [6] mentioned later.

Letter 3:

I think that the best way to keep clusters secure is to hide the compute nodes on a private network and keep them totally inaccessible from the outside world. Any access to compute nodes is through the master node. Compute nodes, if they need to be installed or updated, do so from a repository on the master node. The master node, then, is firewalled on the public interface, with only a few ports open-- ssh, http, https, and so on.

The internal interface is wide open, since you never know what researchers are going to want to run, but even if they start up something that tries to listen on the outside interface it remains unreachable by naughty people.

Once you lock it down like that, the number of potential problems decreases pretty drastically. For some things (apache server-status and so on), I have access restricted to my workstation in the apache config. For compute clusters, I can't see the need for lots of off-campus access, so even locking things down to on-campus users doesn't seem overly restrictive.

Overall, I've found our compute resources to be much less likely to attract security problems than our public systems. We aren't doing any DoD research or anything, though.

Monitoring the compute nodes as well as the master node enhances security. Furthermore, we feel that many users are security naive and we think that it is good to have some controls over what they run.

Letter 4 was quite long and covered too many topics for us to discuss it here. There is one passage from this letter that we would like to quote:

Over the decades I have accumulated a number of interesting anecdotes on security -- major cracks of systems in our medical center, a Bulgarian grad student who engaged in rampant data theft on a chemistry cluster shipping research data on rational drug design back to Bulgaria, and more.

The worst incidents that resulted in actual damage were of this sort -- "inside jobs" where lax security boundaries INSIDE an organization permitted unauthorized access, at least until they were detected and bopped.

4 Other Resources

We did an extensive literature search and uncovered many relevant papers and resources. We will put the materials and links at the following URL:

http://www.cs.umaine.edu/~markov/clustersecurity

Below we will list the abstracts of a representative set of papers that we feel would be of interest to many cluster operators.

Searching For Open Windows and Unlocked Doors: Port Scanning In Large-scale Commodity Clusters by Lee, Koenig, Meng, and Yurcik [2].

Current methods for monitoring the security of large-scale commodity clusters tend to treat these clusters as nothing more than collections of independent nodes. As such, the techniques used to secure these clusters have, for the most part, been adaptations of techniques developed for securing and monitoring enterprise computing

environments. We have previously proposed the idea of monitoring the security-state of large-scale commodity clusters by examining their emergent properties, that is, properties that are only visible when one ceases to look at a cluster as a collection of disparate nodes and begins to look at the properties of the cluster as a whole. We show that by correlating the open network ports observed on cluster nodes with other emergent properties - such as active processes and the contents of important system files- security analysts can make insightful observations that can greatly restrict the actions that an attacker can carry out undetected.

Intrusion-Tolerant Server Architecture for Survivable Services by Min [3].

Survivable systems are increasingly needed in a wide range of applications. As a step toward realizing survivable systems, this paper presents architecture of intrusion-tolerant servers. It is to deliver intended services transparently to the clients even when a computing node fails due to failures, intrusions, and other threats. In order to deliver only secure results to the client, we need an algorithm to decide agreement on results from replicated servers. For this purpose, a secure and practical decentralized voting algorithm for the architecture is proposed in the paper. Through the experiments on a test-bed, especially, for web services, the approach turned out very effective in terms of extra cost and considered to be able to cope with both confidentiality and integrity attacks.

NvisionCC: A Visualization Framework For High Performance Cluster Security by Yurcik, Meng, and Kiyanclar [4].

Large high performance clusters are gaining popularity as a means of harnessing vast computing resources at low cost using commodity components. Cluster system administrators face difficulty from two related problems. First, while several cluster monitoring solutions collect data on the node state and overall performance of a cluster, few monitors place emphasis on the meaningful visual presentation of this information which is increasingly important as cluster size in nodes grows beyond the human capability to manage using command line tools. Second, while cluster state and performance data have been visually monitored in several systems, there are currently no cluster monitors that visualize cluster security events. We have developed a framework for effective visualization of a high performance cluster security which we describe in this paper. We present GUI screenshots from a security visualization tool based on this framework and discuss our experience using this tool on high performance clusters at NCSA.

Detecting Anomalies In High-Performance Parallel Programs by Florez, Liu, Bridges, Vaughn, and Skjellum [5].

Message Passing Interface (MPI) is an effective programming technique for implementing parallel programs for distributed computation. As these applications run, a number of different types of irregularities can occur including those that result from intrusions, user misbehavior, corrupted data, deadlocks or failure of cluster components. In this paper, we perform a comparison of different artificial intelligence (AI) techniques that can be used to implement a lightweight monitoring and detection system for parallel applications on a cluster of Linux workstations. We study the accuracy and performance of deterministic and stochastic algorithms when we observe the flow of function library and OS system calls of parallel programs written with MPI. We demonstrate that monitoring of MPI programs can be achieved with high accuracy and in some cases with a 0% false positive rate in real-time, and we show that the added computational load on each node is small. Finally we demonstrate that simple deterministic methods perform poorly when the program flow grows in size and variety, and that more complex methods are required.

The Trellis Security Infrastructure For Overlay Metacomputers And Bridged Distributed File Systems by Lu, Closson, Macdonell, Nalos, Ngo, Kan, and Lee [6].

Researchers often have non-privileged access to a variety of high-performance computer (HPC) systems in different administrative domains, possibly across a wide-area network. Consequently, the security infrastructure becomes an important component of an overlay metacomputer: a user-level aggregation of HPC systems. The Trellis security infrastructure (TSI) is layered on top of the widely-deployed secure shell (SSH) and systems administrators only need to provide unprivileged accounts to the users. The contribution of TSI is in demonstrating that a single sign-on (SSO) system, for a variety of use-case scenarios, can be implemented without requiring a completely new security infrastructure. We describe the use of TSI for a Canada-wide overlay metacomputer, for computational workloads (i.e., CISS-3) that spanned 22 administrative domains, at its peak had over 4000 concurrent jobs, and included a new distributed file system (i.e., Trellis NFS).

NIDS Architecture for Clusters by Gadaud [7].

Intrusion detection is a security concept implemented on networks in various academic and commercial solutions. Most of them rely on sensors dedicated to local area networks or Internet. However clusters rely heavily on networks. Because of their uniformity, they are sensible to attacks: one compromised node can lead to the control

of whole cluster. In order to solve such security issues, we purpose a NIDS architecture which addresses the same constraints as a cluster: efficiency, scalability and reliability. It is based on the cluster paradigm. We stress on the facts that network packets must be dispatched according to streams and analysis must be load-balanced at the process level. Moreover two types of practical parallel analysis are presented, depending on the type of flows. Finally, we discuss implementations and dimensioning issues.

Instant Attack Stopper in Infiniband Architecture by Lee, Kim, Yum, and Yousif [8].

With the growing popularity of cluster architectures in datacenters and the sophistication of computer attacks, the design of highly secure clusters has recently emerged as a critical design issue. However, the majority of cluster security research has focused on how to detect and prevent attacks rather than on how to minimize the effect of attacks once detected. The action against detected attacks in the cluster is as important as the actual detection process since no detection mechanism is full-proof in its ability to protect cluster systems without the effective cluster-wide reaction. In this paper, we propose a scheme, referred to as the Instant Attack Stopper (IAS) that can instantly confront security attacks in a cluster. Specifically we provide detailed implementation methods of IAS in InfiniBand Architecture (IBA) - a new promising communication standard for future System Area Networks (SANs) and clusters. IAS focuses on removing malicious communication on the IBA fabric among processes involved in an attack, which is accomplished through the proposed Security Management Agent (SeMA). We will show IAS deployment in different security levels to meet various security requirements.

5 Recommendations

A number of respondents seem to feel that security is well taken care of. The Stakkato Intrusions suggest that no one should get too relaxed about security. It also seems that quite a number of clusters are being managed somewhat loosely, and that they would have difficulty withstanding a determined attack from a sophisticated foe.

We feel that it is beneficial for cluster operators to exchange information about practices and we hope that our survey makes a contribution in this area.

6 Future Work

Our experience with this questionnaire has encouraged us to consider repeating the process in a year or so with a redesigned questionnaire based on our experience. We are also planning to post these results and a more complete literature search at the URL mentioned earlier:

http://www.cs.umaine.edu/~markov/clustersecurity

Finally, we will try to organize more sessions devoted to the topic of security of supercomputer clusters, with the intention of helping this area emerge as a distinct specialty.

7 References

[1] Leif Nixon. "The Stakkato Intrusions: What Happened and What Have We Learned?" *6th IEEE International Symposium on Cluster Computing and the Grid*, 2006. CCGRID 06, May 16-19 2006, Singapore. Also available at http://www.nsc.liu.se/~nixon/stakkato.pdf

[2] Adam J. Lee, Gregory A. Koenig, Xin Meng, and William Yurcik. "Searching for Open Windows and Unlocked Doors: Port Scanning in Large-Scale Commodity Clusters." *2005 IEEE International Symposium on Cluster Computing and the Grid*, CCGrid 2005, pp. 146-151.

[3] Byoung Joon Min. "Intrusion-tolerant Server Architecture For Survivable Services"; *Journal of Supercomputing*, v 33, n 1, July, 2005, pp. 93-102.

[4] William Yurcik, Xin Meng, and Nadir Kiyanclar. "NvisionCC: A Visualization Framework For High Performance Cluster Security." *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, 2004, pp. 133-137.

[5] German Florez, Zhen Liu, Susan Bridges, Rayford Vaughn, and Anthony Skjellum. "Detecting Anomalies in High-performance Parallel Programs." *International Conference on Information Technology: Coding Computing*, ITCC 2004, pp. 30-34.

[6] Paul Lu, Michael Closson, Cam Macdonell, Paul Nalos, Danny Ngo, Morgan Kan, and Mark Lee. "The Trellis Security Infrastructure For Overlay Metacomputers And Bridged Distributed File Systems." *Journal of Parallel and Distributed Computing*, v 66, n 9, September, 2006, pp. 1181-1188.

[7] Fabrice Gadaud. "NIDS Architecture for Clusters." *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*, pp. 78-83.

[8] Manhee Lee, Eun J. Kim, Ki H. Yum, and Mazin Yousif. "Instant Attack Stopper in Infiniband Architecture." *2005 IEEE International Symposium on Cluster Computing and the Grid*, CCGrid 2005, pp. 105-110.