Cluster Security as a Unique Problem with Emergent Properties: Issues and Techniques*

William Yurcik† Gregory A. Koenig†‡ Xin Meng† Joseph Greenseid†

†National Center for Supercomputing Applications (NCSA)
‡Department of Computer Science
University of Illinois at Urbana-Champaign

{byurcik,koenig,xinmeng,jgreen}@ncsa.uiuc.edu

Abstract. Large-scale commodity cluster systems are finding increasing deployment in academic, research, and commercial settings. Coupled with this increasing popularity are concerns regarding the security of these clusters. While an individual commodity machine may have prescribed best practices for security, a cluster of commodity machines has emergent security properties that are unique from the sum of its parts. This concept has not yet been addressed in either cluster administration techniques or the research literature. We highlight the emergent properties of cluster security that distinguish it as a unique problem space and then outline a unified framework for protection techniques. We conclude with a description of preliminary progress on a monitoring project focused specifically on cluster security that we have started at the National Center for Supercomputing Applications.

1. Introduction

Large-scale commodity cluster systems are finding increasing deployment in academic, research, and commercial settings. Over the last several years, the trend has been towards an increase in both the absolute number of cluster installations and in the average number of nodes per cluster. The increase in the average sizes of clusters introduces a new set of challenges for system administrators. Perhaps one of the most important tasks assigned to system administrators is ensuring the security of cluster installations. While a great deal of effort has been expended in creating tools to aid in the installation, administration, and monitoring of clusters, very little effort has been expended in creating tools that address the unique issues of cluster security, particularly for very large cluster installations.

When commodity clusters were still a new technology, most of the development focus was centered on simply getting them to work; the issue of cluster security was given relatively little consideration for at least two reasons. First, many people thought it was unlikely that hackers would disrupt scientific systems and jobs. Second, many people believed that the issues related to cluster security were the same as for general computer security. ("What works for one system should work for a collection of 100 systems.") However, as cluster systems have become more widespread and powerful, they have become increasingly desirable targets to attackers due to a few functional characteristics:

(1) *High bandwidth connections* – To facilitate its computational goals, a cluster must have high bandwidth connections to the outside world, allowing interactive use by many users, transfer of large datasets into and out of the cluster, and fast inter-node communication. These high bandwidth connections are attractive to attackers because the attacker can subsequently leverage them for purposes such as launching denial-of-service flood attacks against other sites.

(2) *Extensive computational power* – Legitimate cluster users marshal the aggregate processing power of multiple machines with the goal of solving grand challenge scientific problems. In contrast, this computational power could be used by an attacker for purposes such as carrying out brute-force attacks against authentication mechanisms on other network resources to which the attacker wishes to gain unauthorized access. For example, we have observed cases where attackers have used parallel versions of traditional password cracking tools [2, 8, 11] running on a compromised cluster in an attempt to decrypt stolen password files. Decrypting an encrypted password typically involves either a dictionary-type attack or a brute-force search through the entire space of possible passwords.

* This work is supported in part by a grant from the Office of Naval Research (ONR) under the auspices of the Technology, Research, Education, and Commercialization Center (TRECC) "On-Demand Secure Cluster Computing" Project 2003-04.

Because both of these are "embarrassingly parallel" problems, a cluster gives near linear speedup for the task, thus making the computational power of a cluster an attractive target to hackers.

(3) *Massive storage capacity* – Many high-performance cluster environments include storage capacity measured in terabytes, used for storing large scientific datasets and the results produced by computations involving these datasets. To a hacker, large-capacity disk storage is an attractive target for use in creating repositories of stolen copyrighted software and multimedia files.

The issues related to cluster security are not the same as those related to general computer security. Even though the behavior of individual nodes may be simple and could be approached with traditional computer security techniques, we believe that effective security management in the context of cluster systems requires tools that evaluate the state of the cluster as a whole. ("A 100-node cluster is different from 100 standalone systems.") Consider the example of a traditional security monitoring tool that examines the flow of communication into and out of individual cluster nodes. This tool is limited to evaluating security based only on streams of data that it considers independently of any cluster-specific context. On the other hand, a cluster-aware security monitoring tool could evaluate whether a given node should even be communicating at all, based on information from sources such as the cluster's job management system. That is, if no job is currently scheduled for execution on a given node, that node should most likely not be sending or receiving data on the network.

The idea that cluster security must be considered as a whole is further underscored by realizing that while the behavior of individual cluster components may be simple, the combined interactions of multiple components may result in complex, unintended, and non-intuitive behaviors that are difficult or impossible to predict. That is, even if certain hardware or software components that make up a cluster are certified as "assured," these components must co-exist in a cluster environment that most likely consists of "non-assured" components. Furthermore, even if a cluster were built entirely from certified components, it is unlikely that the entire cluster, considered as a single entity, would have been evaluated in any kind of certification process. Simple combinatorics make it infeasible to use formal methods to identify and protect against all known vulnerabilities from component interactions. For example, attack trees, initially proposed by Schneier, are a good technique for prioritizing risks from known attacks [15]. However, despite work that generates attack trees for limited, and some would say artificial, scenarios, attack trees have been shown not to scale to practical environments where new attacks cannot be modeled in advance and where the scale of components and their interactions are intractable for realistic computation [17].

We propose that cluster security is an *emergent property* because it arises from the independent security aspects of the individual cluster nodes and is at the same time irreducible with regard to the overall cluster system. Accordingly, the goal of our research is to identify the security characteristics that are unique to cluster environments with the aim of using these characteristics as a foundation upon which to develop a unified security tool suite specifically targeted to clusters. The National Center for Supercomputing Applications (NCSA) is in a unique position to study this topic due to a vast array of resources including high-performance cluster hardware and software, a highly experienced operations staff, and a large number of researchers from around the world who converge on NCSA to apply cluster computing to a wide range of topics in various scientific disciplines.

The remainder of this paper is organized as follows. Section 2 provides a review of related work in the field of cluster security, including an examination of both research topics and current state-of-the-art practical techniques. Given this background, Section 3 states specifically why cluster security is more than the sum of its constituent parts. Section 4 provides details of some empirical observations regarding the security of clusters installed at the leading-edge site at NCSA. Since our motivation is not just to identify the unsolved problems that are related to cluster security but also to provide solutions, in Section 5 we outline objectives for a tool suite that we are developing to specifically address the unique issues related to cluster security. We end with conclusions and directions for further study in Section 6.

2. Related Work

The current approach to cluster security is divide-and-conquer. Existing well-understood security techniques are deployed independently against various cluster components with the logic that, upon composition of the final cluster system, the overall security of the cluster should follow from the fact that all cluster components are secured. At the very least, this approach ignores a huge volume of cluster-specific context that can be applied to the problem of securing the cluster, for example by leveraging information available from the cluster job manager. More importantly, however, theories of composition cannot be applied for security properties; a uniform theory for composition of secure systems is

an open research problem [9]. Emergent properties may appear for the cluster system as a whole that cannot or do not exist within individual cluster components [5, 6, 9, 20]. One such property is aggregate information leaks in which the sum of information leaked from all nodes is much higher than the information leaks from a single machine [10].

In this section, we consider the current state of the art for cluster security, described both in terms of current cluster security research and in terms of existing best practices in securing cluster systems.

2.1. Cluster Security Research

To our knowledge, there are only two previous papers directly related to cluster security. This does not necessarily represent lack of operational activity on the topic; indeed, we summarize significant state-of-the-art operational activity in Section 2.2. However, we believe that the dearth of papers does signify that research has focused on cluster performance and has regarded cluster security as a non-research operational issue based on the incorrect assumption that cluster security is composable by combining individually secured components.

In [4], the tradeoffs between performance and security in a high-performance computing environment are discussed. Although not all points are relevant to a cluster environment, the authors do make the following pertinent observations:

- Cluster applications are increasingly encapsulated in more fine-grained distributed components (as opposed to fewer, larger centralized components) that are spread across a cluster and thus dependent upon high-speed non-blocking interconnection networks. Authentication, availability against denial-of-service attacks, and confidentiality against eavesdropping have not been addressed in many cluster environments. For example, in the TeraGrid community, many cluster implementations explicitly do not consider security but rather assume an inherently trusted environment (e.g., with password-less *ssh* keys stored in NFS for access by individual cluster nodes).

- Performance predominates over security. While login authentication and associated access rights are common, even minimal overhead from the use of encrypted communications is often considered unacceptable. For example, there have been extensive discussions in the TeraGrid community regarding the use of unencrypted protocols such as *rsh* and *ftp* versus more computationally costly protocols such as *ssh* and *scp*.

In [12] and [13], Pourzandi et al. present the Distributed Security Infrastructure (DSI), a security architecture designed to operate on Linux clusters in a carrier-class telecommunications environment of high availability, reliability, and scalability. In this context, clusters are used to provide the redundancies required for always-on telecom applications, and the use of strong yet efficient security mechanisms is critical for preventing fraud. DSI provides a distributed framework of mechanisms for authentication, communication integrity, access control, and auditing. In the DSI architecture, each node in a cluster runs a Security Manager that communicates with the Security Managers on other nodes and with a central security management console via encrypted channels. The console is used to monitor and manage the security context of the cluster. DSI presents a very compelling architecture from the standpoint of considering cluster security as an emergent property. The types of clusters on which DSI is designed to work tend to have very circumscribed environments in which a few critical processes run under the control of a few specific user identities. This contrasts the environment typical of facilities like NCSA in which computational clusters tend to run a wide variety of user-compiled applications from a large number of different user identities. Still, the design of DSI is an important point of comparison for our work.

Monitoring is an important part of cluster management, providing situational awareness and leading to the ability to quickly recover from detected anomalous events. Examples of cluster monitoring efforts include Ganglia [7], Supermon [18], Rvision [19], and Clumon [14]. Since most cluster monitoring systems require software agents to be installed on each node of a cluster, there has been a trend for custom cluster monitoring solutions that evolve over time to address specialized in-house requirements.[1] For example, the development of Clumon has been influenced greatly by the Linux cluster environment at NCSA. To our knowledge, all current cluster monitoring tools are designed only for performance monitoring and do not address issues related to security monitoring. We believe that this is an unfortunate deficit that needs to be addressed due to the fact that perhaps no other component in a cluster environment has such a profound vantage point from which to make decisions regarding cluster security. That is, a cluster monitoring tool is at the ideal

[1] The installation of any software on cluster nodes is discouraged and only accepted after being rigorously scrutinized due to any overhead that would degrade performance or the introduction of vulnerabilities.

point in the management loop to take into consideration the context of the cluster as an indivisible entity and thus uncover emergent properties related to the overall security of the entire cluster installation.

2.2. State-of-the-Art Best Practices

As with non-clustered systems, clusters should be maintained according to good systems management practices. Due to the lack of cluster-specific security tools, many tools designed for non-clustered machines are being used to address issues related to cluster security. The common approach is to group several existing independent security tools into one package. The downside of this approach is that it causes subsequent problems in interoperability between security mechanisms. Further difficulty comes when the various tools are upgraded on different schedules. Yet another downside is that many potential security solutions cannot be used in the context of high-performance computational clusters due to excessive resource requirements in implementing the solution. Best practice security strategies adapted from non-clustered systems are multi-layered with a security policy enforced by good network design and software maintenance, and the use of secure authentication mechanisms, intrusion detection systems, and a logging infrastructure.[2] *The key is to strike a balance between performance, usability, and security*.

One of the most effective cluster security protection solutions comes from good network design. The idea is to hide the majority of cluster nodes (the "compute nodes") on a private network and to harden a small set of visible nodes (the "head nodes") that cluster users log on to when accessing cluster resources. This is a so-called "enclosed" cluster design in which intra-cluster communication takes place entirely over the private network. In contrast, in an "exposed" cluster design, intra-cluster communication takes place over an open, Internet-accessible network. Securing and monitoring a few head nodes is much easier than protecting a potentially much larger number of compute nodes visible to the Internet. The downside of using an enclosed cluster design is that some types of cluster applications may require each compute node to be exposed. Examples of such applications include distributed grid-computing applications that may join compute nodes from multiple clusters into a single computational resource, and applications that may be "steered" at runtime by special client software running on user workstations. Furthermore, while hiding a cluster's compute nodes behind a private network may help to shield them from casual scans, the machines are still vulnerable to intrusion from the head nodes. Thus, an enclosed network design does not entirely eliminate the need for a good security structure on the compute nodes.

A second technique commonly used to promote cluster security is the careful selection and maintenance of software installed on the cluster. Only those software packages that are necessary for the computational success of the cluster should be installed. Additionally, a list of any software packages that run with special privileges (i.e., in Unix terminology, this would be any software installed to run "setuid root") should be maintained so a careful watch can be kept on these packages. When security updates are issued for any cluster software, appropriate patches must be installed throughout the cluster. There are at least two challenges, however, to patching software in a production cluster environment. First, if the amount of software installed on the cluster is great, there could be a potentially large number of updates to apply to a potentially large number of cluster nodes. Clearly, software tools for automating software distribution across the cluster nodes are necessary. Second, patching software may introduce new flaws or performance deficiencies into previously functioning software, thus disrupting the user-visible cluster environment.

Secure authentication and communication tools such as SSH and Kerberos are used widely to eliminate plaintext passwords, a potential source of attack against cluster head nodes or exposed compute nodes. Furthermore, Kerberos provides a convenient mechanism for authenticating users on all cluster resources from a central authentication database.

Finally, the use of an Intrusion Detection System (IDS) is a well-established technique for maintaining system security. In the context of a cluster, the IDS may be installed on just the cluster head nodes or may be installed on the compute nodes as well. If the IDS is installed on compute nodes, it is important that the IDS monitoring be low-overhead so it does not disrupt the computational goals of the cluster. Furthermore, it is possible that the number of alerts generated by multiple IDS installations, one per cluster node, may be of such a volume that they may overwhelm a human monitor. A mechanism that can automatically filter IDS alerts, along with other logs generated by traditional Unix-style syslog-type subsystems, is critical.

It is important to underscore the discussion in this section with a reminder that none of the techniques described here address the emergent aspects of cluster security by evaluating the state of a cluster in its entirety. Each technique is

---

[2] Record all rejected requests, record all authenticated actions [1], and record all raw alarms generated by attack signatures or anomaly detection or integrity checks to be processed by scripting languages like SWATCH. Of course the logging infrastructure itself also becomes a target for attacks, so it too must be secured.

directed towards protecting individual nodes in the cluster as separate entities. Furthermore, each technique operates independently of the other techniques. We believe that to be effective, cluster security tools must monitor the state of the entire cluster and base decisions of this context treated as a whole. This idea is also suggested in [1], which points out that protecting against resource violations requires application-level monitoring. Stated from a different perspective, protecting the resources in a cluster environment requires "closing the loop" and considering all facets of the cluster security problem.

3. The Unique Nature of Cluster Security

The result of not treating cluster security as different from non-cluster security is an increased vulnerability to attacks that simultaneously target multiple cluster components. In this section, we describe six ways in which cluster security is different from traditional enterprise-level security. We argue that, in order to be effective, cluster protection schemes must take these into account.

First, a cluster encompasses a collection of *distributed resources to be protected*. By definition, clusters are multiple, closely-coupled machines that are centrally administered. These machines share common resources such as network access, compute cycles, and storage. The challenge is to secure these internal distributed resources against unauthorized access while at the same time permitting easy access by legitimate users. In contrast, the resources found in a typical enterprise-type environment are often very loosely coupled and exhibit minimal coherence of these types of resources.

Second, a cluster must provide mechanisms for *resource management*. The challenge here is to manage a cluster such that legitimate users can consume resources efficiently in an authorized way using an agreed-upon job prioritization system.[3] This is distinguished from enterprise-type environments that usually do not need to manage resources between competing interests. When a user executes a job on a cluster, it is often difficult to differentiate legitimate versus illegitimate use unless there are obvious malicious process signatures. For example, legitimate cluster users are potentially able to tamper with shared data or to excessively consume compute cycles to the extent of disrupting the service available to other cluster users.

Third, clusters present a *heterogeneous management environment*. That is, a cluster may be composed of different hardware and software node configurations (heterogeneous clusters). Even in the case of clusters containing the same hardware and software node configurations, there is usually a separation of cluster nodes by specialized function into "head nodes," "compute nodes," "storage nodes," and "management nodes." The challenge is to coordinate security across different node platforms and different specialized function nodes. This is different from enterprise-type security in that cluster security management must be simultaneously platform independent and specialized for different-functioning node types.

Fourth, clusters have *large-scale management requirements*. As Schneier points out, security is a process, not a product [16]. As the sizes of clusters continue to increase, the task of maintaining and monitoring cluster security becomes an intractable problem. For example, one production cluster at NCSA consists of 1,500 nodes. At this scale, it is not practical to manage a cluster without leveraging the use of automation in conjunction with human interaction. Because of the heterogeneous management environment described above, tools to automate security management need to be aware of the similarities (and differences) present among cluster resources. In this way, cluster security is different from enterprise-level security because the tools that target enterprise-level security typically assume that every resource is subtly different.

Fifth, clusters, considered as a coherent unit, exhibit *characteristic behavior* different from non-clustered machines. This is exhibited in network traffic patterns, number of bytes transferred, applications executed, and compute loads. The challenge is first to identify, and later to understand, these behaviors via profiling in order to provide appropriate protections.

Finally, and perhaps most relevant to the idea that cluster security is an emergent property, cluster resources exhibit *dependent risk*. In enterprise-level security, a single compromise on a machine may result in unauthorized access, destruction of data, and a platform for future attacks. However, a compromised machine in an enterprise can be quarantined to prevent cascading damages. In contrast, the security of the resources in a cluster environment is dependent on the integrity of all nodes. A single compromised node in a cluster represents a dramatically-increased risk to the rest of the cluster nodes due to the fact that many nodes share identical configurations. In this way, clusters are much more

---

[3] Basney and Livny define a cluster resource access policy as (1) who may use a resource, (2) how they may use a resource, and (3) when they may use a resource [1].

vulnerable to "class break" types of attacks as described in [16]. Our experience also suggests that security failures in clusters are worse than enterprise-level failures due to the fact that cluster users tend to coordinate access across various geographically-distributed resources. This coordination necessitates crossing security domains, and when one of these security domains is compromised, the attacker has a much easier job of compromising the other security domains.

4. Empirical Observations

As is the case with many research projects, this project started with experiences indicating a need for investigation. NCSA is a leading-edge site for high performance computing and maintains a dedicated security operations staff to manage the integrity of its production services (e-mail, web servers, Kerberos, etc.) as well as its high-performance clusters. This staff also participates with other security teams both nationally and internationally to exchange knowledge.[4]

Data collected from daily monitoring suggest that clusters are attacked differently from other resources at NCSA. Table 1 shows data comparing network traffic and security events between two public servers and a cluster at NCSA. *Flows* are associated packet streams to and from each machine as an indicator of network activity. *Suspicious* are caution alarms from a general-purpose Intrusion Detection System (Bro) that may or may not indicate problems but do indicate unusual activity requiring further investigation. *Alarms* are security attack signatures that have been recognized by the IDS and indicate with high probability that an attack is either taking place or that a security breach has already occurred. The data in Table 1 suggest that NCSA clusters have high activity levels, and, more pertinent to this investigation, have significantly more security events than non-cluster resources. It is an open question whether our clusters are being specifically targeted or whether some other mechanism is in play, and whether this data is consistent with other supercomputing sites.

Table 1. Sample Traffic and Security Event Data

Machine	Flows	Suspicious	Alarms
Traffic over one day			
Public Server 1	147	13	0
Public Server 2	1999	0	0
Cluster A	6669	414	40
Traffic over two days			
Public Server 1	268	32	0
Public Server 2	2138	6	0
Cluster A	12171	657	62
Traffic over five days			
Public Server 1	450	59	0
Public Server 2	2306	6	0
Cluster A	27462	749	62

We are developing general traffic visualization tools that can be used to monitor distinctive patterns in cluster traffic. Figures 1-3 show screenshots of output produced by these tools. Figure 1 shows a high-level view of the entire NCSA Class B IP address space that can be configured to highlight various activities. For example, the figure shows details of *ssh* connections into the address space. Two dense blocks of addresses, denoted in the figure with arrows, are immediately visible and correspond to the two primary high-performance cluster installations at NCSA.

[4] An example of national cooperation is the Committee on Institutional Cooperation - IT Security Working Group (the academic consortium of Big Ten Universities plus the University of Chicago) http://www.cic.uiuc.edu/groups/ITSecurityWorkingGroup/; an example of international cooperation is the Forum of Incident Response Security Teams (FIRST) http://www.first.org/.

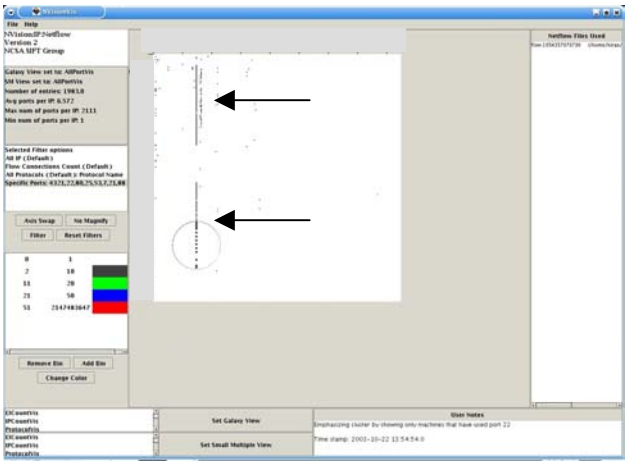

Figure 1. Distinctive Cluster Application Traffic (SSH)

Figures 2 and 3 show views of individual flows into and out of NCSA. Lines beginning on the left sides of the figures and running to the centers of the figures represent traffic originating from external IP addresses and terminating on machines at NCSA. Similarly, lines beginning at the centers of the figures and running to the right sides of the figures represent traffic originating from NCSA and terminating on machines outside of the NCSA address space. Figure 2 shows legitimate traffic from an external user "touching" many nodes on an NCSA cluster over a short sampling period. This traffic pattern is also similar to the malicious pattern generated when a hacker port-scans a range of machines, and demonstrates how information from other sources such as the cluster job manager can be marshaled to provide the context necessary to distinguish the two.

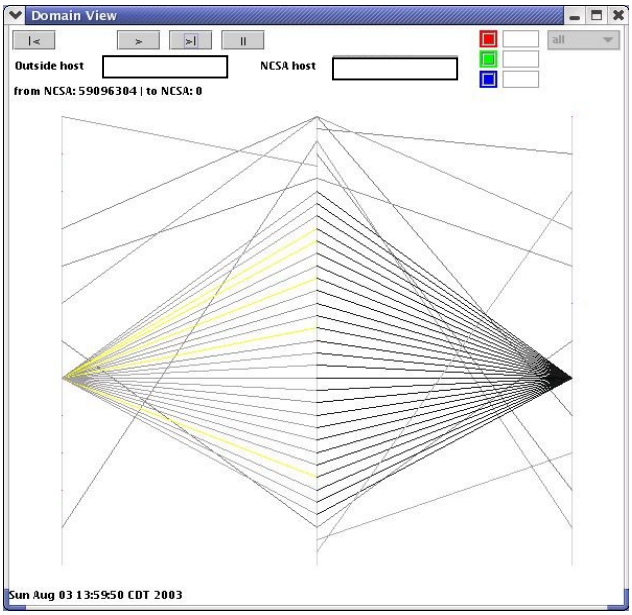

Figure 2. Distinctive Cluster Connections from a Single Host

Figure 3 depicts another distinctive traffic pattern that we have found within our clusters. This legitimate traffic represents direct communications between a remote cluster and a cluster within NCSA. While each of the individual connections is not noteworthy, the combination of traffic connections creates a distinct visual pattern that is obvious to a human monitor. This is an example of an emergent property of cluster traffic.

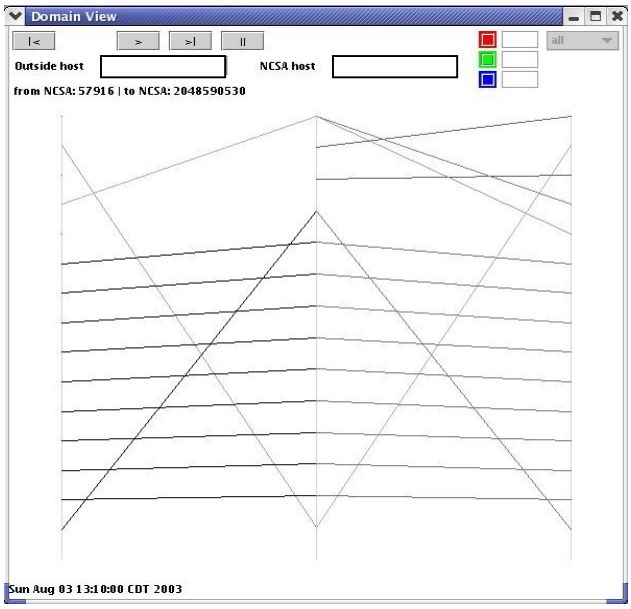

Figure 3. Distinctive Cluster Connections Between Multiple Hosts

Our preliminary conclusions based on the data examined in this section are twofold. First, we conclude that clusters experience security events in a pattern different from other systems. Second, we conclude that cluster traffic has emergent properties that make it distinctive, and thus useful in monitoring for security. Analyzing cluster traffic patterns in real time to distinguish legitimate traffic for uses such as distributed grid-based computing from malicious scans is a challenging problem.

5. Techniques for Managing Cluster Security

The aim of our investigation of cluster security issues is to produce techniques that are effective in managing security as an emergent property of cluster environments. Our expectation is that these techniques will most likely be realized in the form of tools to monitor and manage cluster security. To that end, this section outlines our approach for comprehensive cluster security. From previous security experience, we build on the basic assumptions that no individual cluster component can be protected from all attacks, accidents, and design errors, and no amount of hardening can guarantee that cluster components can be made invulnerable to attack.

A *threat model* describes the resources that an attacker can be expected to have along with the corresponding attacks an attacker can be expected to mount. Understanding the threat model for a given environment is important when developing an appropriate security solution. The threat model for a high performance cluster environment includes both authorized users of resources and unauthorized outsiders. Attackers may have access to large amounts of computational cycles, bandwidth, and storage, such as in the extreme case where another cluster has been compromised and is being used to launch an attack. The three primary threats that we foresee can be classified into the areas of (1) *confidentiality attacks*, where unauthorized access is given to cluster resources and the information stored within those resources, (2) *integrity attacks*, where unauthorized modification is made to the state of the cluster and to the information stored within the cluster, and (3) *availability attacks*, where access to the cluster and the information stored within is denied by unauthorized means. The consensus security approach to these CIA (confidentiality, integrity, availability) threats is to (1) *deter* attacks by avoiding or preventing the occurrence of preventable information security breaches, (2) *protect* against attacks by safeguarding assets from security breaches, (3) *detect* the occurrence of security breaches, (4) *respond* rapidly to security breaches, and (5) *recover* the confidentiality, integrity, and availability of cluster assets to their expected state. Our approach is to develop tools that specifically target the detection phase by focusing on the emergent property of cluster security. We feel that this is the key to opposing the threat model that we have identified for cluster security.

Figure 4 shows a graphical overview of the cluster security monitoring problem. The left side of the figure depicts the physical resource being monitored, a large and complex cluster, along with a puzzled cluster administrator. The center of the figure depicts the raw performance data collected across the running cluster and the processing of this data into useful information. Finally, the right side of the figure depicts our cluster monitoring tool as it evaluates the processed cluster data and presents a synthesized abstract view that is more easily comprehended by the cluster administrator. At this high-level view, if the monitoring tool raises an alert, the cluster administrator can then drill down through the layers of abstraction to reveal the source of the alert in the underlying raw data.

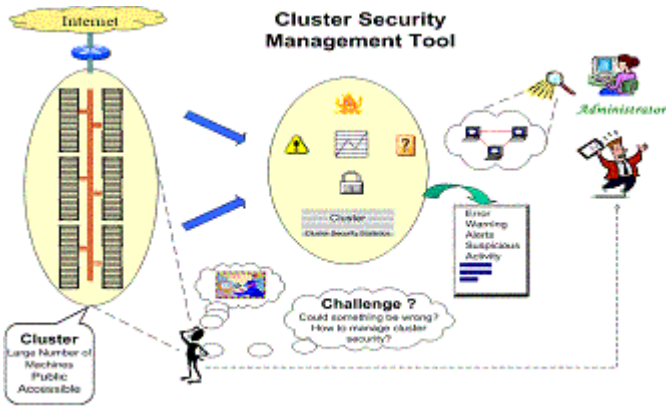

Figure 4. A Comprehensive Framework for Cluster Security Monitoring

Three primary criteria guide the design of our cluster security tool development. First, the cluster security tools must not require significant redesign of existing cluster architectures. Second, the cluster security tools will leverage existing general-purpose security tools whenever possible, re-targeting these tools for use in clusters, instead of developing entirely new tools. Finally, the tools must address several Human Computer Interaction issues such as allowing the administrator to set alert thresholds for limiting the amount of data generated by the tool or providing visualization mechanisms to allow administrators to easily see output from the tool.

We specifically propose to address the overall cluster security problem with three complementary techniques that we will integrate into a single coherent tool:

(1) Process monitoring – Examining the individual processes running on each cluster node is critical for overall cluster security. We are developing tools based on the Clumon monitoring framework [3] to collect information about every process on every node in a cluster, analyze the set of processes found, and visually alert the cluster administrator when anomalous conditions are discovered. Such anomalies might include system-related processes that should be running on a node but are, in fact, missing, processes that are running on a node when the node should be idle (particularly in the case of "root" processes), and an unusually-large number of processes running on an individual node or over the context of the entire cluster. Detecting these types situations within a cluster is possible because a cluster presents a relatively limited search space for anomaly detection versus an enterprise network with machines of different types (servers, workstations, laptops) running an unbounded number of different software processes.

(2) Network port scanning – Unexpected network ports that are opened on a cluster node can be a good indicator of suspicious activity. We are working on a port scanner that monitors ports usage tailored to a cluster environment and presents the results to cluster administrators using visualization. This effort leverages the existing open source scanning tool, Nmap, modified to optimize it for use in a cluster. The underlying idea is that network ports must be opened in order for an attacker to interact with a cluster, otherwise compromising a cluster is of limited value since there can be little or no interaction with compromised nodes.

(3) Traffic analysis – Applications running on cluster systems have unique patterns of communication, making the task of distinguishing legitimate traffic from abnormal traffic difficult. This difficulty is compounded to the growing use of grid computing software that exhibit communication patterns that cross cluster boundaries by joining multiple geographically-distributed clusters into a single computational resource. We plan to correlate information from the

cluster job scheduler with network traffic into and out of the cluster in order to distinguish typical cluster traffic patterns from suspicious or known malicious traffic patterns. For example, an automated traffic analysis tool can use contextual information from the job manager as well as a constrained set of legitimate IP addresses belonging to one or more well-known clusters to aid in recognizing patters of communication in parallel computations such as localized neighbor communication, many-to-many communication, or all-to-all communication. That is, if a set of nodes are communicating with each other within the context of a single job, the traffic is most likely legitimate. This is in contrast to a machine on an enterprise network that is not attached to any unifying context. The ultimate goal is to automatically detect the types of traffic patterns shown in Figures 1-3.

We expect to be able to leverage the fact that cluster nodes are usually separated by specialized function into broad categories such as "head nodes," "compute nodes," "storage nodes," and "management nodes" in order to create profiles of expected node states for which our cluster monitoring tool will scan. For example, storage nodes typically have a very predictable and unchanging set of running processes, open ports, and network traffic patterns since users typically do not directly access these nodes. Similarly, the state of compute nodes consists of a baseline set of well-known processes, ports, and traffic patterns that are modified as user processes are scheduled on compute nodes under the direction of the job scheduler. By defining such characteristics for each node type, new nodes can be added to the cluster very easily by simply assigning one or more profiles to each node, thus defining its expected behavior patterns to the monitoring tool. Through this practice, we believe that the problem of monitoring the state of an entire cluster can be reduced to the more tractable problem of monitoring for deviations from a set of expected behavior patterns and finding correlations between these deviations and contextual information provided from sources such as the job manager. For example, if an additional process is found that deviates from the baseline set of processes for a compute node, the process is very likely to be legitimate if it belongs to a user for which the job manager has scheduled a job on that compute node.

Finally, we are aware of performance constraints in high-performance computational clusters and are evaluating our tool to ensure that it does not adversely impact the performance of grand challenge computations running on the resources being monitored. Our initial results indicate that the performance impact of our monitoring framework is negligible.

6. Conclusion

In order to secure a cluster as a holistic system it must be treated as a single unit and not as a collection of independent networked machines. In this paper we have identified the unique characteristics of cluster security as an emergent property, and have argued that successful cluster security management strategies and tools must treat cluster security differently from traditional enterprise-level security.

Cluster security is an important area that deserves additional investigation. Future research in our group will focus on re-targeting the best non-cluster security tools to the unique requirements of cluster security. This effort will revolve around the three-pronged cluster security research plan focusing on process monitoring, network port scanning, and traffic monitoring that we have outlined in this paper.

7. Acknowledgments

We extend a special thanks to Joseph P. (Joshi) Fullop, IV of NCSA for partnering with us on this project. Joshi is focused on monitoring cluster performance, and we leverage his work on Clumon to create our own cluster security monitoring tools. We also acknowledge the unique insights of Jim Barlow, the Head of NCSA Security Operations and lead NCSA contact for the TeraGrid Security Working Group <http://security.teragrid.org>. Other members of our NCSA security research group have also made significant, although indirect, contributions to this paper: (in alphabetical order) Cristina Abad, Jim Basney, Randy Butler, Neil Gorsuch, Anand Krishnan, Kiran Lakkaraju, Yifan Li, Doru Marcusiu, Jeff Rosendale, Ken Sartain, Aashish Sharma, and Xiaoxin Yin. We also acknowledge guidance and instrumental details from the following NCSA high performance cluster system administrators: (in alphabetical order) Tim Bouvet (Copper), Karen Fernsler (Tungsten), Eric Kretzer (Platinum and Titan), Dan Lapine (Mercury), and Thomas E. Roney (TRECC) along with their Technical Program Manager Wayne Louis Hoyenga. Externally we acknowledge an open exchange of ideas on general cluster monitoring topics with Thu Nguyen at Rutgers University. Finally, we thank the anonymous reviewers for their invaluable comments that helped improve the clarity and quality of this paper.

8. References

[1] J. Basney and M. Livny, "Deploying a High Throughput Computing Cluster," Chapter Five within the book *High Performance Cluster Computing Volume 1: Architectures and Systems* (edited by R. Buyya), Prentice Hall 1999.

[2] Cisilia Project Webpage, http://www.cisiar.org/proyectos/cisilia/home_en.php.

[3] Clumon Project Webpage, http://clumon.ncsa.uiuc.edu/main.html.

[4] K. Connelly and A. A. Chien, "Breaking the Barriers: High Performance Security for High Performance Computing," *ACM New Security Paradigms Workshop*, 2002.

[5] D. A. Fisher and H. F. Lipson, "Emergent Algorithms – A New Method for Enhancing Survivability in Unbounded Systems," *32nd Hawaii Intl. Conf. On Systems Science (HICSS*), 1999.

[6] H. Hinton, "Under-Specification, Composition and Emergent Properties," *ACM New Security Paradigms Workshop*, 1997, pp. 83-93.

[7] F. Hoffman, "Cluster Monitoring with Ganglia," *Linux Magazine*, 2003.

[8] R. Lim, "Parallelization of John the Ripper (JtR) Using MPI," University of Nebraska at Lincoln, Department of Computer Science and Engineering Technical Report, http://cse.unl.edu/~rlim/jtr-mpi/report.pdf, 2003.

[9] H. Mantel, "On the Composition of Secure Systems," *IEEE Symposium On Security and Privacy*, 2002.

[10] M. B. Paz and V. M. Gulias, "Cluster Setup and its Administration," Chapter Two within the book *High Performance Cluster Computing Volume 1: Architectures and Systems* (edited by R. Buyya), Prentice Hall 1999.

[11] T. Perrine, D. Kowatch, "Teracrack: Password Cracking Using TeraFLOP and Petabyte Resources," San Diego Supercomputer Center Security Group Technical Report, http://security.sdsc.edu/publications/teracrack.pdf, 2003.

[12] M. Pourzandi, I. Haddad, C. Levert, M Zakrewski, and M. Dagenais, "A New Architecture for Secure Carrier-Class Clusters," *IEEE International Workshop on Cluster Computing*, 2002.

[13] M. Pourzandi, I. Haddad, C. Levert, M Zakrewski, and M. Dagenais, "A Distributed Security Infrastructure for Carrier Class Linux Clusters," *Ottawa Linux Symposium*, 2002.

[14] T. Roney, A. Bailey, and J. Fullop, "Cluster Monitoring at NCSA," *Proceedings of Linux Revolution Conference*, 2001.

[15] B. Schneier, "Attack Trees," *Dr. Dobb's Journal*, December, 1999.

[16] B. Schneier, Secrets and Lies: Digital Security in a Networked World, Wiley, 2000.

[17] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated Generation and Analysis of Attack Graphs," *IEEE Symposium on Security and Privacy*, 2002.

[18] M. Sottile and R. Minnich, "Supermon: A High-Speed Cluster Monitoring System," *IEEE International Workshop on Cluster Computing*, 2002.

[19] Tiago C. Ferreto, Cesar A. F. De Rose, and Luiz De Rose, "RVision: An Open and High Configurable Tool for Cluster Monitoring," *IEEE International Workshop on Cluster Computing*, 2002.

[20] A. Zakinthinos and E. S. Lee, "Composing Secure Systems that have Emergent Properties," *11th IEEE Computer Foundations Workshop*, 1998.