M&Ms: Measure and Manage Exascale-System Behavior for Security and Integrity

Vitus Leung[‡], Manuel Egele[*], Ayse Coskun[*]
vjleung@sandia.gov, megele@bu.edu, acoskun@bu.edu
[‡]Sandia National Laboratories [*]Boston University, Electrical and Computer Engineering Department

Focus area: (2) Extreme Scale Data, Knowledge, and Analytics.

Problem Description and Goal:

Exascale systems, even more than contemporary cluster deployments, will be crucially important for our country's infrastructure and security. These systems can be used, for example, to process high-fidelity simulation workloads for diverse applications, such as power-grid load-leveling, or simulations of uncontrolled nuclear chain reactions. To confidently draw conclusions from such simulations, it is mandatory that the computation leading up to the results is not influenced by attacks.

Cluster deployments are uniquely positioned to enable attack detection techniques by leveraging similarities between cluster nodes and configurations. For example, for a typical high-performance computing workload, it is viable that a group of systems running this load show similar behavior over a time period. Imagine now that one node is successfully attacked and exploited. The changes on that node will result in differences of observable behavior and these differences can be leveraged to detect and manage malice. However, any practical exascale deployment will show varying levels of heterogeneity that complicate such similarity-based detection attempts. For example, it is unlikely that a supercomputer is stocked with exclusively identical hardware. Even now, the NCSA Blue Waters system has heterogeneous nodes. It is generally more practical to gradually expand the available computing resources within the limits of physical constraints (e.g., space, power, and cooling). Similarly, it is common practice to upgrade subsets of nodes over time rather than all nodes simultaneously. Therefore, to apply similarity-based detection techniques such as the one described above, we need to solve a variety of challenges in data acquisition and analysis. The scale and heterogeneity of future exascale deployments call for novel data collection and analysis methods that are capable of interpreting and correlating signals (measurements) obtained from systems in an efficient and scalable way.

The goal of this paper is to discuss possible approaches that can identify and manage divergent and potentially malicious behavior in a cluster deployment where individual nodes have largely similar but not identical configurations, including hardware, hypervisors, operating systems, and applications.

Possible Approach:

In many attack scenarios in cluster deployments, the attacker initially compromises only a few nodes, and targets other nodes as time progresses. One possible approach is to exploit this observation that initially only a few of the nodes exhibit anomalous behavior to detect the infected nodes (systems) and provide opportunities for quick response and recovery. This approach features four specific steps:

1) *Collect low (performance) overhead sensor data from systems and applications.* Sensory data can be derived from a variety of sources, such as hardware sensors, system call logs, or application-level data collected from library call information. Access times to read sensor data varies greatly between different sources. For example, accessing hardware-based performance counters can be achieved at millisecond rate. Collecting library call events that include argument values is often a much slower operation. From a security perspective, event monitors are best implemented at a higher privilege level than the monitored target. This architecture ensures that a compromised target cannot affect sensor readings. For example, if one is only interested in monitoring user-space applications, a kernel-level event monitor is sufficient. However, if the threat model includes kernel-level attacks (e.g., rootkits), event monitoring should rather be performed in a hypervisor. This higher level of security comes at the cost of the so-called *semantic gap*, which is a consequence of the different abstraction layers employed in modern computing systems Bridging the semantic gap [1] is frequently a resource-intensive task (e.g., memory and CPU cycles).

   In this type of work, we expect to encounter two sources of overhead: overhead (1) due to the collection of individual signals, and (2) due to the transmission of collected sensor data to decision points. Type-1 overhead occurs with each measurement of a signal. Thus, we should target signals that can be collected with small overheads already and that allow us to adjust the monitoring frequency as needed. Many processor and server vendors include

sensors that can be tracked at reasonable overhead. To address type-2 overhead, we should employ sub-sampling techniques. Using sub-sampling entails that decisions (i.e., detection of malice) are made based on partial information. Thus, sub-sampling inherently opens an avenue for evasion to the attacker. To minimize this window of opportunity for the attacker, we need to investigate whether dynamic modification of the sampled signals can be leveraged to deliver probabilistic assurances that an attack can be detected. That is, instead of deciding at deployment time what signals to sample, we can permute the sampled signals at runtime.

2) *Use statistical methods to identify groups of nodes that experience largely consistent behavior.* A straight-forward approach to assess behavioral similarity of a cluster deployment is to centrally collect information about the computing nodes and assess similarity based on this global view [2]. We do not anticipate this to be a feasible approach for exascale deployments. Instead, we should exploit locality in terms of (a) the physical proximity of similar-type hardware, and (b) the spatial proximity of tasks of an application as a result of the performance-aware job allocation algorithms in place. This locality-based approach divides a given deployment into a variety of hierarchies based on such proximity measures. Following a divide-and-conquer approach, similarity detection mechanisms can be applied at the different layers of the hierarchy. Lower layers provide access to finer-grained data, but upper layers have a wider view of the system. The construction of the hierarchy allows the tuning of data-granularity and communication overheads against detection capabilities and performance.

3) *Label potentially malicious or compromised nodes by identifying divergent behavior.* The heterogeneity in realistic exascale deployments complicates the assessment of behavioral similarity. Naturally, these effects will result in differences in measured behavior. However, a practical security solution should not raise alerts because of such benign deviations. Thus, to discriminate between benign behavioral variations and variations caused by an attack, we can employ supervised machine learning techniques. Based on a labeled training dataset, these techniques allow the ranking of individual signals with respect to their predictive ability to detect attacks. The findings from this step can be leveraged to refine the sub-sampling presented in step (1). This refinement should further narrow the window of opportunity for the attacker to evade detection. This refinement may also allow for the continued operation of some resources in step (4).

4) *Manage affected and unaffected resources*. Finally, there is a need for research into techniques to allow exascale system operators and users to effectively manage the tradeoffs between throughput and computing integrity. This has been done in related areas [3]. The hierarchies mentioned in step (2) may be one natural way to partition affected and unaffected resources. The rankings mentioned in step (3) will be crucial to the efficiency of these techniques. These techniques may be related to some of the management ideas mentioned in Focus Area (3), e.g. virtualization, containerization, secure tagging.

Impact to DOE:

The systems to be protected with an approach as described above generate information that is used as the basis to make decisions with wide ranging impacts on energy, economy, and national security. A scalable and reliable solution to detect and manage malicious activity increases the decision maker's confidence that the data supporting the decision has not been tampered with through malicious interference. Confidence in data accuracy is even more crucial in situations where collected information is used for autonomous decision making (e.g., load leveling in the power grid).

We expect that data and implementations resulting from this line of research will be made available to interested parties in pursuit of an open science agenda. Dissemination of individual datasets generated during this research that are subject to confidentiality restrictions (e.g., the precise configuration of DOE clusters) will have to be excluded from such dissemination. Nonetheless, the methods studied and developed during this research should be applicable to other similarly structured datasets found in the industry and other organizations.

References

[1] M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. X. Song. "Dynamic spyware analysis". In Proc. of the USENIX Annual Technical Conference, Santa Clara, CA, June 2007.

[2] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. "Compa: Detecting compromised accounts on social networks". In Proc. of the Network and Distributed System Security Symposium, San Diego, CA, February 2013.

[3] J. Meng, S. McCauley, F. Kaplan, V. Leung, A. Coskun. "Simulation and optimization of HPC job allocation for jointly reducing communication and cooling costs". Sustainable Computing: Informatics and Systems, In Press.