Cybersecurity Design Techniques Targeting and Leveraging HPC

Joshua Letchford, Jackson R. Mayo, and Robert C. Armstrong
Sandia National Laboratories, Livermore, CA
{jletchf,jmayo,rob}@sandia.gov

Focus Area: Trustworthy Supercomputing

With high-stakes decisions being made based on HPC application outputs, and with HPC platforms themselves becoming more complex and heterogeneous, cybersecurity concerns for HPC increasingly resemble those for general digital devices and networks. This represents both a challenge and an opportunity to develop effective cybersecurity approaches compatible with the performance and scale that define HPC. Many of the issues inherent in this problem have strong parallels to techniques developed for other domains, and research should leverage these similarities when possible.

One salient problem area involves detecting and preventing corruption in scientific computations. Maintaining resilience to the increased frequency of accidental hardware faults at extreme scale is already a priority for HPC, and this investment can likely be leveraged to foster resilience to cyber attacks as well. Containment through sandboxing is one promising approach for a flexible and performant HPC design that is securable against internal and external compromise [2]. Sandboxes allow safe and authorized access to the HPC system or between its partitions through gateways, without restriction on the architecture or performance internally. Formal verification can help establish fundamental reliability and security properties of sandboxes and ensure that programs and data are accessed safely and remain uncorrupted. While it is unlikely that HPC applications will be formally verified as a whole, critical library functions like internet access protocols and I/O can be verified for safety requirements without loss of performance.

Beyond the elements that can be formally verified, in-depth resilience can guard against both glitches and tampering via cross-checks among different, overlapping computations, which can be randomized in a "moving target" system to increase the uncertainty and difficulty for attackers [1]. This approach, analogous to algorithms being developed for mitigating accidental "silent errors" at extreme scale [4], can damp perturbations regardless of their accidental or malicious origin, making it more difficult for an attack to have a lasting effect. Algorithmic error detection of this sort can also alert users or system administrators to tampering.

To determine the best choice of techniques for mitigating corruption in any particular application, we should develop methods for balancing tradeoffs among overhead costs, security requirements, and potential benefits, taking into account the knowledge and goals of potential attackers for a given system design. Adversarial modeling with techniques such as game theory can help determine the effect of HPC security and resilience decisions on attacker behavior and optimize these tradeoffs between security and performance; imposing and quantifying attacker uncertainty can be an important element in this optimization. Similar decisions may need to be made even when users have no explicit ill intent but their actions out of self-interest threaten security and reliability due to misaligned incentives.

Furthermore, both the security techniques themselves and the meta-evaluation of the system rely on data for training, validation, and verification. One issue that arises is how to deal with the vast quantities of data that these systems can generate. Here, methods from data sciences and machine learning (such as automated feature selection [5]) will be important in identifying what part of this heavy stream of data is useful. However, while the analysis of such data is both necessary and valuable, it also creates another vector for adversarial manipulation. As an example, consider a spam filter and an attacker wishing to slip spam by unmarked. A successful attack might target the integrity of the filter itself, poisoning the filter by influencing the data it uses to retrain itself until the filter becomes ineffective. Among research efforts that relax the assumption of correct data, common approaches include assuming the data has quantifiable uncertainty or that it requires some cleaning to remove corrupted or erroneous data points. There are also domains such as robust optimization [3] where this data uncertainty is treated adversarially. Analogously, while not all

output of a potentially compromised HPC system can be taken at face value, we propose that with suitable design principles the needed computational results can be robustly inferred.

[1] R. C. Armstrong and J. R. Mayo. Leveraging complexity in software for cybersecurity. In *Proc. 5th Cyber Security and Information Intelligence Research Workshop*. Oak Ridge, TN, Apr. 2009.

[2] R. C. Armstrong, J. R. Mayo, et al. Complexity science challenges in cybersecurity. SAND2009-2007, Mar. 2009. `wiki.cac.washington.edu/download/attachments/7478403/Complexity+Science+Challenges+in+Cybersecurity.pdf`.

[3] H.-G. Beyer and B. Sendhoff. Robust optimization: A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(3334):3190 – 3218, 2007. ISSN 0045-7825.

[4] G. Bosilca, R. Delmas, et al. Algorithm-based fault tolerance applied to high performance computing. *Journal of Parallel and Distributed Computing*, 69:410–416, 2009.

[5] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502, April 2005. ISSN 1041-4347.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy under contract DE-AC04-94AL85000.