SAND2015-3546C

Weaving Security into Large-Scale HPC Systems

Brian Gaines and Kevin Pedretti
Center for Computing Research
Sandia National Laboratories
{bgaines, ktpedre}@sandia.gov

Focus Area: Trustworthy Supercomputing

Introduction

The high performance computing community has largely been able to ignore the need for pervasive computer security. The accepted belief is that security is handled by the information technology staff or handled at the head node or by the network administrator. The reality is that this is only a small start toward securing these systems. Securing these systems will require dedicated resources and a shift in the mindset of the community toward a more secure and resilient HPC environment.

A subtlety of integrity in scientific computing is that data originates from a diverse set of sources that often generate vast amounts of information. In order to verify the integrity of the data, the security of the originating systems must not be forgotten. Of primary concern is the supervisory control and data acquisition (SCADA) systems used to control, monitor, and collect information from sensitive scientific equipment (e.g., particle colliders).

There are three areas of research that would enable security advances: the introduction and use of virtualization and other isolation techniques to support secure computing in HPC; exploration of distributed security primitives through software-hardware co-design initiatives; and the thorough exploration of securing SCADA systems utilized at research facilities.

Virtualization and Isolation

Virtualization has been a hot topic for some time in the computing world but has not been prevalent in the HPC community. Virtualization provides a hardware backed isolation mechanism that could be leveraged by software to monitor and protect operating system kernels and run-time systems [4].

Virtualization provides one form of isolation, however modern processors from AMD, ARM and Intel offer other mechanisms for providing isolation to running code [1, 2]. These additional isolation mechanism are implemented differently but generally fall into two categories, static root of trust and dynamic root of trust. The additional features offer yet another layer of isolation and in the case of Intel Secure Guard Extensions (SGX) instructions, provide a mechanism to prevent a malicious kernel or hypervisor from monitoring the actions of a task utilizing SGX.

Monitoring from isolated environments has been a research endeavor for several years and has been primarily focused on the use of virtual machine introspection (VMI) [3] to determine actions within the virtual machine (guest). Methods like VMI have the inherent problem of a semantic gap between the guest actions and interpreting those actions from the virtual machine monitor (hypervisor). While there has been extensive research in this area no general solution exists. Primarily the research has been focused on automatically bridging this gap through the use of debugging symbols and replicating the functionality of existing guest monitoring tools within the hypervisor. The reason for these approaches is due to the closed source nature of the operating systems the researcher wishes to introspect. One possible avenue for the HPC community is the integration of hypervisor or other isolation modes with the guest and run-time systems. Redesigning the system with the OS completely aware and collaborating with the monitoring services could reduce or eliminate the semantic gap issue while preserving the expectations of speed required by the HPC community.

Isolation and monitoring are not a panacea, as it has been demonstrated that these same primitives can be used by those with malicious intent as well [5, 6].

Distributed Security Co-design

Commodity processors have only recently begun to focus on security primitives provided by hardware. While this may improve single-host security, this leaves open a wide area of research for distributed security mechanisms. This leaves room for improvement between the various hardware components working in concert within HPC systems.

Research into the use of distributed security primitives at the CPU or network level could be leveraged by both applications and operating systems to provided needed security guarantees. Utilizing co-design the HPC community could work with HPC system vendors to add the distributed security primitives needed to provide an end-to-end security solution.

SCADA Security Exploration

A large part of the prior workshop appears to have focused purely on HPC security, which is absolutely necessary to support trusted computation in the future. However, the single-point focus on HPC hides the need for enhanced security between HPC systems and scientific instruments producing vital input data. These instruments typically require extensive SCADA systems to control their operation, monitor their safety, and collect the vast amount of data produced. Malicious modifications to the output of these systems or even false readings would affect the integrity of the data collected, and possibly the safety of the overall system.

Conclusion

Securing the scientific community's resources is necessary for continued success and will likely be fought with significant resistance, due to perceived overhead. We see the need for improvements in the areas of system monitoring, which could be done from isolated environments supported by virtualization; the implementation of distributed security mechanisms to support increased security within HPC systems; and the often overlooked area of SCADA security required to assure the collected data is accurate and that the systems are functioning as intended.

These significant challenges will not be solved without dedicated research funding and time to make the solutions viable to meet high performance expectations of our researchers.

References

[1] Trustzone. http://www.arm.com/products/processors/technologies/trustzone/index.php.

[2] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata. Innovative technology for cpu based attestation and sealing. https://software.intel.com/en-us/articles/innovative-technology-for-cpu-based-attestation-and-sealing, 2013.

[3] T. Garfinkel and M. Rosenblum. A virtual machine introspection based architecture for intrusion detection. In *In Proc. Network and Distributed Systems Security Symposium*, pages 191–206, 2003.

[4] J. Lange, K. Pedretti, T. Hudson, P. Dinda, Z. Cui, L. Xia, P. Bridges, A. Gocke, S. Jaconette, M. Levenhagen, and R. Brightwell. Palacios and Kitten: New High Performance Operating Systems For Scalable Virtualized and Native Supercomputing. In *Proc. 24th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2010.

[5] J. Rutkowska. Subverting vista$^{\text{TM}}$kernel for fun and profit. SyScan, July 2006.

[6] R. Wojtczuk, J. Rutkowska, and A. Tereshkin. Another way to cirumvent intel trusted excution technology. Invisible Things Lab, December 2009.