Cost of Security (Auditing Focus)

Computer System, Cluster and Networking Summer Institute

Authors: Matthew Chambers (Michigan Tech University)
Kevin Lopez (CSU San Bernardino)
Casey Mortensen (New Mexico Tech)

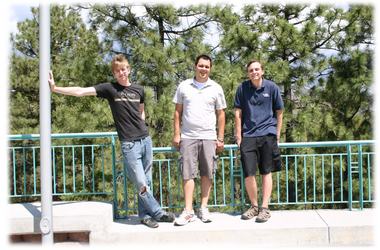
Mentor: David Kennel (DCS-1)
Instructor: Andree Jacobson (NMC)

Abstract:

Many high performance computing clusters are used to process highly sensitive or classified data. In these situations where security is tremendously important, system administrators will often run auditing programs to monitor access of important files or system calls. The actual cost of auditing large scale computing systems is relatively unknown. Our project focuses on performing several benchmarks that stress file I/O, memory bandwidth and the effects of auditing various common system calls to assess the viability of implementing these enhanced security measures in a high performance computing environment.

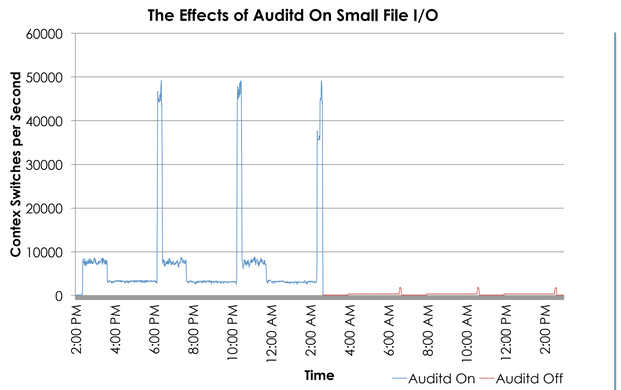
Background:

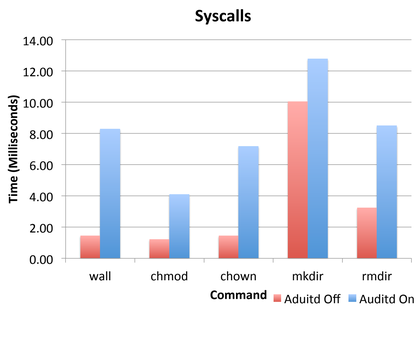
Auditing in the Linux kernel is managed by the auditd utility. Auditd can watch specific areas of the file system or monitor system calls. Auditd also is in charge of recording and logging the data to disk. Auditd requests a CPU interrupt and context switch when it needs to log an action, which can be noted in the graph below.



I/O Test

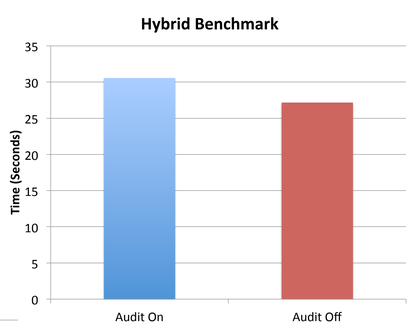
The File I/O tests were conducted using a Python script to create a massive amount of files in a directory, open and append data to the files, followed by removing all of the files. The graph to the right showcases the differences in context switches when auditd is on/off. Mass creation/deletion of small files in a single directory is the worst case scenario for auditd's file watching and logging capabilities. We found that watching files and directories causes a performance hit of about ~2.5%.





Syscall

The Syscall tests were performed using several scripts that modified permissions, ownership of files, and other commands. With auditd running there was a performance hit of up to 400%: however, it is not significant when compared to the additional security the service provides as the commands execute in a few milliseconds.

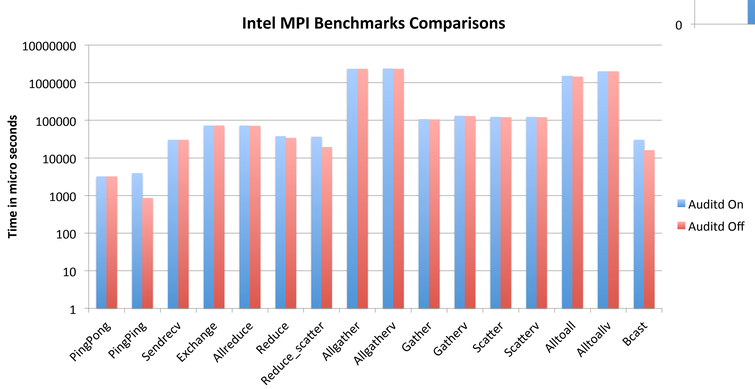


Hybrid Bench

A customized bench-mark combining various system administrative tasks, computational tasks, and file I/O tasks. The purpose of this test was to illustrate the impact of CAPP rules on a pseudo-realistic workload. Overall our test exhibited a 10% performance hit. This could be attributed the system calls being the main target of CAPP rules.

Intel MPI Benchmark

In order to compare performance hits to real life situations, we used the Intel MPI Benchmarking Suite to simulate normal computing situations at LANL. There are three types of benchmarks in the suite: single transfer, parallel transfer, and collective benchmarks. With auditd running, the three types of benchmarks had a minimal performance hit. These results suggest that system administrators should run auditd on their computing systems for the enhanced security without worry of additional overhead.



Conclusion:

There was a measurable but minimal impact on cluster performance. In our testing we could not cause auditd to create a significant amount of overhead. CPU interrupts maxed out at 10,000 per second and did not significantly hamper performance. The cost of enabling auditd is well worth the security benefits that it provides.

LA-UR-11-11193