An Integrated Approach to Software Process Control

Computer Measurement Laboratory, Inc.
info@cmlab.biz

The Concept

Process control is considered to be an integral part of any modern industrial process. CML has leveraged the notion of control theory in the development of its dynamic software process monitoring system. In this guise, a software system is seen as an abstract machine whose operation is monitored and controlled by a software process control system. The CML Attack Recognition and Management Architecture, CARMA, is an instantiation of this concept and has as its foundation the notion of process control.

We have clearly established that it is possible to measure and monitor the execution of software system in real time. These measurements will reflect the current state of the executing software. The feasible state space of these systems is very large. This state space may be divided into normal and abnormal program activity. It turns out that the state space of normal activity is very small. It is possible to model this nominal activity with succinct mathematical models. Measurements taken against an executing program can be compared with a standard model of nominal execution to ascertain whether the program is executing within its nominal envelope or is performing some abnormal functionality.

Due to the fact that state space is too large, it has heretofore not been possible to insure the integrity of complex software system. Further, these systems rapidly evolve incorporating new functionalities and new vulnerabilities. It is not likely that vulnerabilities introduced during the design and development process will be detected. It is not necessary that extreme effort be devoted to this cause. The potential vulnerabilities in the software are not the problem. The problem is that an adversary may exploit them. The focus should on the detection of the abnormal behavior, the exploit, and not on the vulnerabilities. A system that is executing a security flaw is clearly operating outside of its nominal execution profile. This abnormal activity may be detected instantly in real time. Once this activity has been detected, it may be ameliorated also in real time. Attacks on software systems are now a fact of information warfare. These attacks succeed because the software is essentially operating without control or supervision; nobody is minding the store.

End Node Monitoring

Single computer systems can be viewed as leaf nodes in a network. Using the CARMA technology we are able to monitor the activity of all processes executing on these nodes. We can determine, in real time, whether these machines and the processes running on them are operating normally or have been compromised in some measure.

If a system has been compromised, there is a good likelihood that this abnormal activity had its origins from some external source on the network. CARMA will permit us to identify the nature

of the abnormality and the IP address that created the abnormality. That is the current state of the art.

From a systems perspective, however, we would like to identify the source of the problem and begin to manage potential problems at the end computers from a larger perspective -- that of the network.

Distributed Processing

In terms of network structure, the first level of network integration would be the distributed processing system. Here, the network is used to associate multiple machines in a virtual processing environment. In this environment exploits may be directed to a distributed software system. In this type of system organization, the monitor function must also be distributed. The logical solution to the monitoring of a distributed system is to introduce a meta-monitor system, Meta-CARMA that is able to measure and control the activity of multiple systems executing a distributed software system.

In the Meta-CARMA architecture, there would be a meta-analytical engine function that would model and monitor the activity of a software system executing on multiple machines.

Networks

Individual computers as leaf nodes are woven into a fabric through their connections to the Internet. With our CARMA Software Process Control system we are in a unique position to assert complete control over the individual machines that are the leaf nodes of the network structure on an individual basis. This means that we can determine with a high degree of certainty whether any one of them has been compromised by activity originating somewhere in the network. The problem is, however, that the network itself is operating out of control.

At the intersection points on the network are special purpose computers that control the packet routing function. The CARMA concept may be deployed in these routers in one of two distinct fashions. First, the network protocol may be enhanced to permit the CARMA monitored systems to communicate with the routers directly. This would permit the virtually instantaneous identification of IP addresses that are creating disturbances on the end nodes of the networks. With this knowledge in hand an IP address responsible for the disturbance may be removed from the network before further damage can occur. Routers would simply communicate the offending IP information back upstream through other CARMA enabled routers until the point of origin is identified and managed.

The second aspect of router control relates directly to their functionality in their computational role. These routers should also be running under software process control in that they, too, may be compromised in their routing function. It is also quite possible that a router will begin to fail in its routing function as a result of reliability issues. The distress caused by this malfunction could easily be sensed by peers in the network long before the router itself failed.

Finally, changes in the flow of packets through the network will induce changes in the operation of the routers. This change can be detected as a change in the activity of the router software. In a network control function, the Network CARMA architecture can monitor the flow of packets through the network and identify abnormalities in a timely fashion.