Managing Multiple Perspectives on Trust

Dr. Clifford Neuman
Information Sciences Institute
University of Southern California
(http://clifford.neuman.name)

Trusted computing provides methods for software components to establish confidence in the code with which they communicate. While commonly used for digital rights management, the same underlying mechanisms can be used to protect users from untrustworthy service providers and to provide strong isolation for critical functions running on common infrastructure

This abstract discusses ongoing work to develop trusted computing architectures and policy models supporting multiple perspectives on trust. The TrustView Security Architecture enables strong separation for critical functions. By moving some basic support for separation into the network infrastructure, the architecture enables limited performance isolation across function. The trusted computing reference monitor mediates requirements and obligations for each software component providing mutual protection to all involved.

The TrustView architecture leverages trusted computing technologies to protect multiple, possibly competing, interests within a system, including the interest of the end user against abuse by the companies with which they interact. The architecture supports strong isolation of functions at a coarse level of granularity. Such policies are easier to understand and can be readily implemented by virtualization technologies [3]. Such coarse grained policies are specified in a less dynamic way than traditional fine grained policies: the allowable flow of information between software components is specified through the creation of virtual systems [2]. Protection is provided by limiting the flow of information across virtual system boundaries.

Most systems today allow programs to run in two modes, kernel and application. A system's Trusted Computing Base (TCB) resides in the hardware and kernel, and user applications run untrusted. Early systems like Multics [1] provided more structure, with innermost rings composed of compact, heavily trusted code, and successively less trust required as one moves to outer rings. The problem with this model when it is applied to distributed systems is that it assumes a fictitious hierarchy of trust. Software is either trusted or not, and the software implements whatever policy was decided by its implementer.

In distributed, loosely managed systems like today's Internet, certain processes may be more trusted by some entities, while other processes are more trusted by others. This was illustrated in the past by the installation of root-kits on the PC's of users who played CD's produced by Sony. To Sony, the user's PC was not trusted, but their own software was (trusted does not mean worthy of trust). The users soon discovered that it was Sony who should not be trusted. For a security architecture to protect all users it must provide mechanisms to deal with such mutual suspicion.

Surprisingly, by weakening our trust requirements for modules that are certified for use in a trusted computing environment – i.e. if we accept and certify even partially trusted components – then we can derive significant security benefit for our systems and networks as a whole. We can then use the trusted computing infrastructure to develop a virtual system abstraction that defines policies for interconnecting and managing the flow of information between instances of software modules running in a distributed system. Considering partial trust strengthens security because modules that providers considered trusted (for example, those used by DRM systems can be reclassified as only partially trusted since they might not really be worthy of trust from the perspective of the end user.

The TrustView Security Architecture (TVSA) allows software components and protected resources to be placed in overlapping rings of protection. Collections of functions and applications are associated with “virtual systems” that define views of trust from a particular perspective. A process and its associated persistent data may reside in different rings within different virtual systems so that it is considered more trusted by some and less trusted by others. At run time, information does not flow across ring boundaries except through processes that are members of multiple rings. Virtualization techniques are used to provide strong separation between processes running in different virtual systems. Individual processes mediate the flow across the boundaries which they span. When a process joins a virtual system, obligations are negotiated which constrain the process’s ability to participate in other virtual systems based in part on attestation of the process’s ability to protect the flow of information across virtual system boundaries.

In our architecture, the security attributes of applications that communicate across a network (and within individual hosts) are negotiated and communicated by code in the operating system and network stack on the communicating processors. Applications run in ‘virtual systems’, distributed across network nodes, whose policies for membership are specified during the installation of an application, and managed external to the application. These virtual systems correspond to the rings in our system architecture.

The components of a virtual system are the hardware, OS, and applications on participating nodes. The level of trust placed in each of these components varies according to perspective: thus from the perspective of a node running part of a virtual system, those components running locally may be more trusted, while from the perspective of a server, those same elements of the system may be less trusted.

BIOGRAPHY

Clifford Neuman is director of the Center for Computer Systems Security at the Information Sciences Institute (ISI) of the University of Southern California (USC), and a faculty member in the Computer Science Department at USC. He earned a Bachelor's degree at the Massachusetts Institute of Technology and subsequently worked for Project Athena. He received M.S. and Ph.D. degrees from the University of Washington. Dr. Neuman conducts research in distributed systems, computer security, and electronic commerce and is the principal designer of Kerberos authentication system, the NetCheque and NetCash systems, and the Prospero Directory Service. Dr. Neuman’s current research is focused on trusted computing architectures that support multiple views of trust.

REFERENCES

[1] M. D. Schroeder and Jerome H. Saltzer. *A hardware architecture for implementing protection rings*. Communications of the ACM, 15(3):157-- 170, March 1972.

[2] B. Clifford Neuman, The Virtual System Model: A Scalable Approach to Organizing Large Systems, Ph.D. Thesis, University of Washington, Department of Computer Science and Engineering Technical Report 92-06-04, June 1992.

[3] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Xen and the Art of Virtualization (2003) .Proceedings of the ACM Symposium on Operating Systems Principles. 2003.

Related infomration and updates to this paper may be found at: http://clifford.neuman.name/publications/2007/200705-neuman-csiirw-managing-multiple-perspectives-on-trust/