Access Control for the SPIN Extensible Operating System

Robert Grimm Brian N. Bershad
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA 98195, U.S.A.

Extensible systems, such as SPIN or Java, raise new security concerns. In these systems, code can be added to a running system in almost arbitrary fashion, and it can interact through low latency (but type safe) interfaces with other code. Consequently, it is necessary to devise and apply security mechanisms that allow the expression of policies controlling an extension's access to other extensions and its ability to extend, or override, the behavior of already existing services. In the SPIN operating system [3, 4] built at the University of Washington, we are experimenting with a version of domain and type enforcement (DTE) [1, 2] that has been extended to address the security concerns of extensible systems. We are critically concerned with the performance of DTE, as extensible systems enable the fine-grained interaction between components with very low overhead; we intend to maintain that property while also applying rigid access controls.

The SPIN operating system [3] defines an extension infrastructure, together with a core set of extensible services, that allows for the fine-grained and safe composition of extensions within the operating system kernel. Extensions are written in Modula-3, a type-safe programming language, and execute within the same address space. They interact by calling other parts of the system and by extending existing interfaces to provide new services. A central event dispatcher [4] supports both mechanisms: to call on a service, an extension raises an event, and, to extend an existing interface, an extension registers a handler for that event. The invocation mechanism for events is simply a procedure call, and no context switches are required for the interaction between subsystems (since all extensions are co-located in the same address space).

DTE associates subjects with domains and objects with types and defines legal access modes for pairs of domains and types. In the original DTE model subjects are processes executing for a user. However, since all extensions execute within the same address space, with no clear separation between extensions, the notion of a process can not be maintained for extensible systems. We thus treat both extensions and threads as subjects. Access restrictions on subjects, i.e. on extensions and threads, are enforced at link time (when an extension wants to link against other extensions) and at call time (when a thread wants to call an extension). Access restrictions on objects are enforced by the extension that provides an object's abstraction (if an extension is not trusted to enforce access control on its objects but is expected to do so, DTE can be used to prevent other extensions from linking against the untrusted extension).

Due to the fine grained composition of extensions in SPIN, it is important to minimize the performance overhead of call time access control. We are therefore exploring optimizations that make it possible to avoid some dynamic access checks (for example, some core services can be called by all extensions and thus do not require dynamic access checks). These optimizations are based on a formal model for access control in extensible systems and are thus guaranteed to preserve the security of the system.

We believe that our access control mechanism can provide a solid foundation for future work on security in SPIN and other extensible systems, such as Java or VINO, since these systems, like SPIN, support fine-grained composition of extensions in a single address space.

References

[1] Lee Badger et al. Practical Domain and Type Enforcement for UNIX. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 66–77, Oakland, California, May 1995.

[2] Lee Badger et al. A Domain and Type Enforcement UNIX Prototype. In *Proceedings of the Fifth USENIX UNIX Security Symposium*, pages 127–140, Salt Lake City, Utah, June 1995.

[3] Brian N. Bershad et al. Extensibility, Safety and Performance in the SPIN Operating System. In *Proceedings of the 15th Symposium on Operating Systems Principles*, pages 267–284, Copper Mountain, Colorado, December 1995.

[4] Przemysław Pardyak and Brian N. Bershad. Dynamic Binding for an Extensible System. In *Proceedings of the Second Symposium of Operating Systems Design and Implementation*, pages 201-212, Seattle, Washington, October 1996.