Raising the Bar in Operating System Security:

SELinux and OpenSolaris FMAC

Abstract

Over the past several years, the Security-Enhanced Linux (SELinux) reference implementation of the Flask security architecture has undergone a rapid evolution in its capabilities and maturity thanks to a large and growing developer and user community. SELinux has also influenced a wide range of related work in other operating systems, hypervisors, and applications. In 2008, a new project was started to bring the same Flask security architecture demonstrated in SELinux to the OpenSolaris™ operating system via the OpenSolaris Flexible Mandatory Access Control (FMAC) project. These efforts have fundamentally changed the terms of debate about operating system security and ushered security features previously limited to separate niche products into the mainstream. This article describes the major advances and changes in SELinux that have occurred during the last several years; summarizes other related work that has flowed out of the SELinux project; and introduces the goals, design, and status of the OpenSolaris FMAC project.

Introduction

Security-Enhanced Linux (SELinux) was developed by the National Information Assurance Research Laboratory (NIARL) of the National Security Agency (NSA) starting in 1999 and was first released to the general public via the nsa.gov web site in December 2000. SELinux was created by NSA as a reference implementation of the Flask security architecture for flexible mandatory access control (MAC) in order to show how such controls could be added to a mainstream operating system and to demonstrate the value of MAC [1]. SELinux was intended to serve both as a technology transfer vehicle for encouraging adoption of flexible MAC into mainstream operating systems and as a research platform for advanced security research and development. Prior to the release of SELinux, MAC was only available in separate "trusted" operating system products and was limited to fixed hierarchical security models that were unable to express many kinds of real security goals.

The public release of SELinux drew the interest of both advanced Linux users and the Linux kernel developers, which led to an invitation to present SELinux at the Linux kernel developer

summit in March 2001. The resulting discussion at that summit led to the creation of the Linux Security Modules (LSM) project, an open source project to create a common security framework in the Linux kernel that could support a variety of security models. During the next couple of years, the SELinux developers served as core contributors to the development of the LSM framework and re-architected SELinux to use the LSM framework. The LSM framework began to be merged into the mainline Linux kernel in 2002, and the remaining portions of the framework and the SELinux security module were merged into the mainline Linux 2.6 kernel series by the end of 2003.

Even prior to its integration into the mainline Linux kernel, advanced Linux users had begun packaging SELinux kernel, policy, and application support for multiple Linux distributions so that they could use SELinux for protecting their own systems. SELinux packages for the Debian GNU/Linux distribution were made available as early as 2001, and the Hardened Gentoo project (a security-focused subproject of the Gentoo Linux distribution) began including SELinux support in 2002. The growing developer and user community around SELinux and the efforts to bring SELinux support into the mainline Linux kernel drew the interest of Red Hat, Inc., which began work to fully integrate SELinux support into its Linux distributions in 2003, starting with their new community-based Fedora distribution. The SELinux code was first included in the Fedora Core 2 release in May 2004, eliminating the need for separate patches for the kernel and applications. The introduction of a security policy configuration focused on confining specific network-facing services such as the Apache web server and the BIND domain name server made it possible to enable SELinux by default in the Fedora Core 3 release in November 2004. This security policy configuration was called the "targeted" security policy because it applied SELinux to protecting specific services (i.e., the "targets") that were likely points of attack into the system.

The Fedora SELinux integration work and the resulting community testing and refinement of SELinux formed the basis for including SELinux in the commercially supported Red Hat® Enterprise Linux® product. Red Hat Enterprise Linux 4, released in February 2005, shipped with SELinux as a default-enabled security feature providing out-of-the-box confinement of over a dozen system services. This release represented the first inclusion and use of MAC in a mainstream commercial operating system. MAC was no longer limited to separate "trusted" operating systems and had become a general-purpose security feature. The inclusion of MAC in a mainstream commercial operating system set the stage for the rapid advances in SELinux that have occurred since 2005.

> **"Linux security experts are reporting a growing list of real-world security situations in which the US National Security Agency's SELinux security framework contains the damage resulting from a flaw in other software."**
>
> *Don Marti , LinuxWorld.com*

SELinux: 2005–present

Policy technology advances

Over the past several years, a new generation of policy technology has been developed and deployed for SELinux. The advances in policy technology have included the introduction of the reference policy, the development of loadable policy module support, the creation of policy management infrastructure, and the convergence of strict and targeted policies.

SELinux was originally released by NSA with a small example policy configuration to demonstrate the concepts and the value of flexible MAC. Early adopters of SELinux used that example policy as a base and began contributing changes and additions to it, leading to very rapid growth in its coverage of different applications but at a cost in terms of understandability and ease of customization. NSA sponsored work by Tresys Technology to undertake a redesign of the base policy for SELinux, with a focus on modularity, understandability, tool support, and customization. This work has yielded the SELinux reference policy, which has supplanted the original example policy as the standard base policy for all modern Linux distribution releases that support SELinux, starting with the Fedora Core 5 release in March 2006.

The reference policy was also designed to take advantage of a new feature in the SELinux policy toolchain that was also being developed by Tresys Technology in the same timeframe: support for loadable policy modules. The original SELinux policy configuration and compiler were "monolithic." That is, in order to make any substantive change to policy beyond a few specific forms of customization (e.g., booleans, local file contexts), one needed to obtain a complete policy source tree, make corresponding changes to the source files, and rebuild the entire policy into the binary form required by the kernel. Loadable policy module support was developed to enable individual policy modules to be built and packaged separately from one another. This mechanism has enabled users to easily create local policy modules as needed for site customization, and it has enabled software developers to easily package policy for their applications. Loadable policy module support was also first deployed in Fedora Core 5.

While the loadable policy module support was being merged into the upstream SELinux userland, a new

software library, *libsemanage*, was developed jointly by Red Hat and by Tresys Technology to provide a standard API and infrastructure for managing policy. This library provides a programmatic interface for making changes to policy, as opposed to having to manually edit text files, and provides support for a wide range of local customizations to policy. Front-end tools such as *semodule* and *semanage* were created to enable users and higher level tools to perform policy management tasks. This library and the initial front-end tools also first appeared in Fedora Core 5.

A practical compromise made early in the Fedora SELinux integration was to create a separate "targeted" policy configuration that focused on protecting network-facing services and left ordinary user sessions unrestricted and use that policy as the default so that SELinux could be enabled by default without disrupting users. The complete example policy with significantly more coverage of services and applications and support for user roles became known as the "strict" policy configuration, and this strict policy configuration was not well supported and required significant expertise to successfully install and use. However, this compromise made it possible to incrementally expand the coverage of the targeted policy in each new release through the community testing and feedback process since SELinux was enabled by default. With the introduction of the reference policy, both the strict and targeted policy variants were built from the reference policy sources based on a single tunable setting.

As a result, the targeted policy has grown from covering over a dozen services in the earliest release to covering over two hundred applications in modern releases. The last significant difference between the targeted and strict policies was eliminated starting with the Fedora 8 release in November 2007, when support for confining users was introduced in targeted policy. The Fedora 9 release in May 2008 used this support to define several user roles available by default and to support a kiosk mode of operation where the user session is highly restricted and is completely purged of state after each session. As a result, the targeted policy and the strict policy have converged and there is no longer a separate strict policy. Administrators can largely obtain the behavior of the strict policy by mapping users to confined roles, and they can optionally remove the unconfined policy module entirely, although this last step can be destructive to running processes and requires some care to do safely.

Improved usability

Based on user feedback, the advances in policy technology described above have greatly improved the user experience of SELinux by enabling users to solve many of the problems that they encounter. In particular, the loadable policy module support and the management tools have enabled users to perform local customizations of policy to fit their particular needs and have enabled developers to ship customizations for their applications.

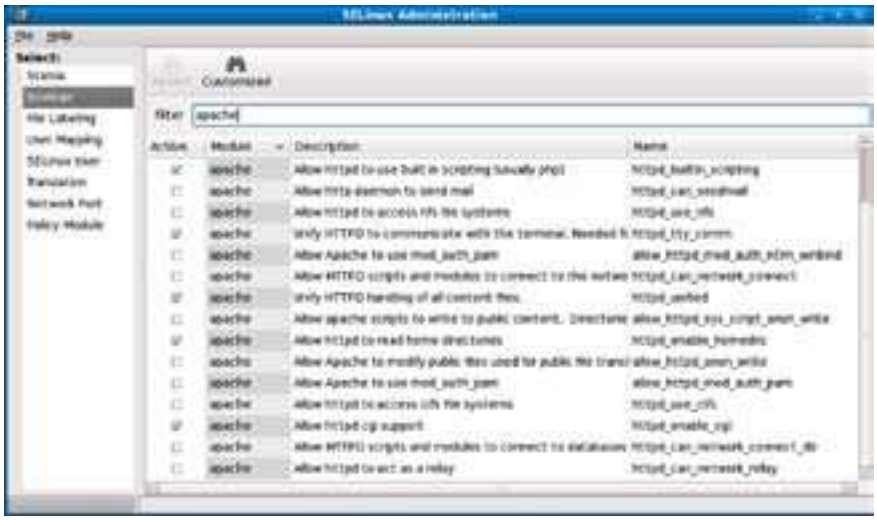

Figure 1: system-config-selinux screenshot

Figure 2: setroubleshoot screenshot

Several graphical tools have also been developed in recent years to assist users with different aspects of SELinux. These tools include *system-config-selinux*, *setroubleshoot*, and the SELinux Integrated Development Environment (SLIDE). The first two tools were developed by Red Hat and first included in Fedora Core 6 in October 2006. The SLIDE tool can be freely downloaded from the Tresys Technology open source server, *http://oss.tresys.com/*, and was included in the Fedora 9 release in May 2008.

A graphical front-end to the *semanage* functionality, *system-config-selinux* allows the administrator to easily see and modify the current enforcing status, policy booleans, label assignments for files and ports, and user authorizations. It also provides simple support for managing the set of loaded policy modules. Recent versions of the tool (see Figure 1) also include a policy wizard for creating new policy modules.

A service for notifying users of SELinux denials, *setroubleshoot* helps users to diagnose denials and resolve them. It has increased user awareness of SELinux and enabled users to identify and solve common kinds of configuration errors. The tool can be configured to display alert popups to the user on the desktop, or alerts can be handled via system logs or email notifications. See Figure 2 for a screenshot of *setroubleshoot*.

SLIDE is an Eclipse plug-in to provide a graphical user interface for policy developers with the conventional features of an integrated development environment, such as policy creation wizards, interface completion and searching, and policy syntax highlighting. Recent versions of SLIDE (see Figure 3) have incorporated support for remote policy debugging and integration with policy analysis tools.

Enhanced security functionality

The core security functionality of SELinux has undergone significant enhancements and improvements since 2005. These enhancements have included extended security audit functionality, enhanced Multi-Level Security (MLS) support, new network access controls, and labeled networking support along with numerous other smaller changes.

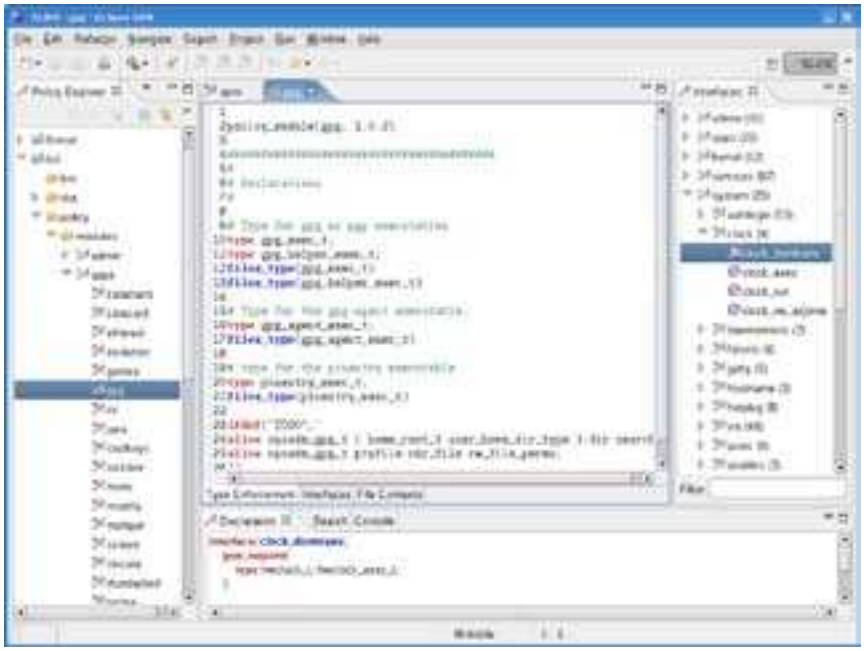

Figure 3: SLIDE screenshot

As part of the work to enable Linux to meet the Labeled Security Protection Profile requirements, SELinux was fully integrated with the Linux audit subsystem, enabling audit to include and filter based on security contexts. Likewise, the optional MLS model of the SELinux security server was enhanced and enabled by default, and user space support for MLS requirements was developed. This work was done with the help of a wide range of contributors from HP, IBM, Red Hat, and Trusted Computer Solutions (TCS).

Red Hat developed a new mechanism for flexible network access controls called SECMARK, which combined the power of the Linux packet filtering framework with SELinux policy. Developers from IBM, TCS, and HP created and integrated two independent implementations of labeled networking mechanisms, labeled IPSEC and NetLabel/CIPSO. These mechanisms enable SELinux protections to be applied across network communications.

Improved performance and scalability

The original SELinux implementation included an Access Vector Cache (AVC) to minimize the need to perform expensive security computations on each operation and sought to keep overheads to reasonably low percentages, but did not specifically engage in any detailed performance optimizations. The original SELinux implementation also only dealt with ensuring safety on systems using an SMP (symmetric multiprocessing) version of the Linux kernel through the use of coarse-grained locking. This approach did not scale well on large SMP systems. Since the original inclusion of SELinux in Linux distributions, a number of performance and scalability improvements have been developed and integrated.

An engineer from NEC Corporation undertook work to enable SELinux to scale well on large SMP systems. He replaced the coarse-grained lock of the AVC with a scheme known as Read-Copy-Update (RCU), enabling SELinux to achieve near-perfect scalability. This made the deployment of SELinux practical on large systems.

As the default targeted policy grew in its coverage of services, the amount of kernel memory used by the policy was increasingly becoming too high and beginning to cause problems for users. As a result, the SELinux core developers discussed approaches to improve memory

use, and a set of memory optimizations were implemented by NSA that radically reduced the kernel memory use by a factor of 20X.

A number of memory and performance optimizations have been developed in recent years by contributors from the Japanese SELinux community working on using SELinux on embedded systems, including contributions from NEC and Hitachi Software among others. Further optimizations to the policy data structures have yielded significant improvements in kernel memory use. The revalidation of read and write permission on individual read and write system calls was optimized to deal with significant overheads on the SuperH architecture, improving overhead by a factor of around 10X.

Meeting security criteria

In 2007, Red Hat Enterprise Linux 5 was validated against the Controlled Access Protection Profile (CAPP), the Labeled Security Protection Profile (LSPP), and the Role-Based Access Control Protection Profile (RBACPP) at Evaluation Assurance Level 4+ on HP and IBM hardware. This was the result of a collaborative effort among HP, IBM, Red Hat, and TCS that leveraged SELinux to provide the labeled security and role-based support. This work led to several improvements to SELinux, including improved audit, MLS, and labeled networking. SGI has also achieved validation of Red Hat Enterprise Linux 5 on its hardware in 2008. These validations represent the first time that a mainstream commercial operating system product has been validated against such criteria, which in the past have been limited to separate "trusted" operating system products. These evaluations were also distinctive in that the companies that sponsored the evaluations worked together to develop the necessary functionality and shared the resulting documentation and test suites despite being competitors.

The Systems and Network Analysis Center (SNAC) of NSA developed and released their first-ever security configuration guide for Linux in 2007, the *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*. Along with many other topics, the guide describes the benefits of SELinux for security and explains how to perform basic configuration and troubleshooting of SELinux. This guide joins the other configuration guides produced by the SNAC over the years for a wide variety of operating systems and is available from *http://www.nsa.gov/ia/guidance*.

The Certifiable Linux Integration Platform (CLIP) is a specific configuration of Linux and associated evidence designed to meet several established security requirements, including the Protection Level (PL) 4 requirements from the Director of Central Intelligence Directive (DCID) 6/3, "Protecting Sensitive Compartmented Information within Information Systems" and the High Impact requirements from the National Institute of Standards and Technology (NIST) Special Publication 800-53, "Recommended Security Controls for Federal Information Systems." CLIP defines a specific configuration of SELinux to provide the foundation for hosting security-relevant applications by ensuring that the underlying assumptions made by those applications are enforced by the operating system. In particular, CLIP leverages SELinux in order to enforce the strong separation of processes and data, support different user roles, and ensure that application security mechanisms are tamperproof and cannot be bypassed. The CLIP project is sponsored by NSA's Custom Solutions Group and is being developed by Tresys Technology. CLIP can be downloaded from *http://oss.tresys.com/projects/clip*.

Growing adoption, use, and community

For the past four years, SELinux has been a default-enabled security feature in the Fedora and the Red Hat Enterprise Linux distributions, providing out-of-the-box confinement of an increasing number of system services. The improved usability of SELinux has enabled users to expand their use of it and to directly solve problems. The anecdotal evidence of improved user experience from public mailing list discussions is further reinforced by statistics being collected by the Fedora project, which began to collect information about SELinux status starting with the Fedora 8 release. The majority of Fedora systems reporting into the Fedora project show that users keep SELinux enabled.

In addition to Fedora and Red Hat distributions, SELinux has continued to make advances in adoption in other Linux distributions. The Hardened Gentoo project has continued to support SELinux in the Gentoo Linux distribution and to integrate newer SELinux features. The Debian GNU/Linux distribution began including SELinux support in the Debian 4.0 release. The Ubuntu distribution began including minimal SELinux support in the Ubuntu 8.04 release, which was then further enhanced in the Ubuntu 9.04 release. Novell began including basic SELinux support as an optional feature in SUSE Linux 11.1.

The benefits of SELinux for mitigating vulnerabilities in software are increasingly being recognized. An article by Don Marti published on LinuxWorld.com in February 2008 stated, "Linux security experts are reporting a growing list of real-world security situations in which the US National Security Agency's SELinux security framework contains the damage resulting from a flaw in other software [2]." In discussing the migration of their mission-critical trading platform to Linux, Steve Rubinow, the Chief Information Officer (CIO) of the New York Stock Exchange (NYSE) Euronext was quoted as saying, "We are very security conscious because we have to be….We maintain the security of our systems by relying on the SELinux features within Red Hat Enterprise Linux [3]."

SELinux has also served as a secure foundation for a number of secure solutions developed for the government. These systems include the NetTop® system originally prototyped by NIARL

and later productized by HP along with several derivative systems. It also includes the TCS Secure Office® Trusted Thin Client system. A number of Cross Domain Solution (CDS) systems have been developed by NSA and by other organizations that leverage SELinux to enforce separation and to ensure the assured invocation of the CDS application. As mentioned earlier, SELinux is also being leveraged by the CLIP project.

Along with growth in its user community, SELinux has experienced significant growth in its developer community since 2005. Developers from HP, Hitachi Software, IBM, NEC, NSA, Red Hat, Tresys Technology, and TCS along with many individual developers have worked collaboratively to enable SELinux to advance rapidly in its feature set and maturity.

Platform for advanced R&D

In addition to serving as a technology transfer vehicle for encouraging adoption of flexible MAC by industry, SELinux has also served as a useful platform for advanced research and development. By providing a base system that supports flexible MAC and exports interfaces to support security-aware applications, SELinux enables research to proceed in understanding MAC in a complete system, from the low-level operating system up through infrastructure layers to the end-user applications.

Securing the desktop environment experienced by typical users is one area of active research. This area is particularly challenging to secure due to the tight coupling of applications typical in such environments and the lack of consideration to any security boundary between processes within a desktop session. Addressing these challenges is critical in order to be able to protect against exploitation of flaws in commonly used desktop applications such as browsers and mail clients, so that a flaw in a single program does not expose all of the user's data to risk. To date, work has been done by NSA to implement the D-BUS message service, the GConf configuration system, and the X Window System server with the necessary support for applying flexible MAC to their objects and operations [4, 5]. Work is ongoing by NSA to develop library support for these extensions, address other components of the desktop infrastructure, and assist in developing policy for the X server that supports simple security goals. Future work includes addressing performance challenges, refining the controls based on experience with real applications, securing the direct rendering interface, providing trusted input and display, and integrating with desktop applications.

Beyond the desktop, a wide range of application security research is leveraging SELinux as a base platform and as an architecture for providing flexible MAC services to higher layers. This includes the SE-PostgreSQL project, an effort by NEC Corporation to develop flexible MAC for database objects and transactions in the Postgres database management system. Research has also been performed by NSA into enforcing Risk Adaptive Access Control (RAdAC) by leveraging the SELinux operating system functionality to protect and isolate an application policy enforcer and by using the Flask architecture and user space security server to provide policy decisions and revocation support [6].

Enabling secure file sharing among networked or distributed systems is another area of active research being led by NSA with support from SPARTA, Inc. This effort requires addressing challenges posed by systems with potentially different security policies that need to share data securely as well as providing the basic mechanisms for conveying security attributes for processes and files across the network. Given the common use of such networked file systems in enterprise environments, enabling flexible MAC to be effectively applied to such file systems is likewise a crucial challenge. Experimental extensions to the NFSv4 protocol have been proposed and prototyped, and work is ongoing to standardize the protocol changes and to get the implementation into a form acceptable to the Linux developer community. Future work will include dealing with heterogeneous policies.

While flexible MAC provides new capabilities for improved security, it also introduces its own set of challenges, including policy scalability and usability. Hence, research by NSA with support from Tresys Technology is ongoing into how to create an abstract layer for policy and how to more closely link the enforcement, debugging, and development of policy to enable users to more effectively develop, debug, and understand policy. Research is also underway at NSA into more advanced policy language features to facilitate policy customization and extension.

Policy management continues to be an area of active investigation. Early work in this area by Tresys Technology and Red Hat has yielded the current policy management infrastructure and tools such as *semanage*. Research by Tresys Technology, which was sponsored by NSA, has also yielded experimental prototypes of a policy management server to support fine-grained access control over the policy itself and of policy management infrastructure to support management of collections of systems. Work has recently started at Penn State University to investigate how to manage policy for virtualized environments with different collections of policy enforcing components.

Influencing other systems

The Flask security architecture demonstrated in SELinux has strongly influenced the security of a number of other systems and software components. In the application arena, this has included the D-Bus message bus software, the X Window System, and PostgreSQL, as previously noted, each of which now has a set of flexible MAC controls implemented that can extend the reach of the policy enforcement to their higher level objects and operations. In the virtualization arena, the Flask architecture has been applied to the Xen hypervisor, yielding the Xen Security Modules (XSM) framework

and the Xen Flask security module developed by NSA, enabling enforcement of policy over virtual machines and their interactions.

The Security-Enhanced BSD (SEBSD) and Security-Enhanced Darwin (SEDarwin) projects demonstrated that the Flask architecture could also be applied to other operating systems. Although SEBSD and SEDarwin are not integrated into their respective mainstream operating system distributions, they helped to drive the requirements for the MAC framework that can be found today in mainstream FreeBSD® and in MacOS X operating systems. They also proved that the architecture was applicable to a variety of operating systems and provided an alternative reference implementation from which others can learn.

OpenSolaris FMAC: origin and goals

In late 2007, NSA and Sun Microsystems, Inc., began a dialogue about integrating support for the Flask architecture into the Solaris™ operating system. This dialogue led to the launching of the Flexible Mandatory Access Control (FMAC) project on OpenSolaris.org in March 2008. The project is a joint effort among NSA, Sun, and the OpenSolaris developer community to bring support for flexible MAC to the OpenSolaris operating system environment.

Unlike Linux, where there was no support for MAC at all prior to the integration of SELinux, the Solaris operating system has an existing MAC solution. Prior to Solaris 10, this functionality was provided by Sun via a separate product, the Trusted Solaris operating system. Like other trusted operating systems of its genre, Trusted Solaris was limited to a fixed MLS security model and tended to lag behind the latest release of Solaris due to the additional engineering and evaluation requirements. In Solaris 10, some of the security functionality of Trusted Solaris, such as support for roles and privileges, was integrated into the main Solaris product and released as part of OpenSolaris. The MLS support was redesigned around the Solaris "zones" mechanism and provided as an optional set of extensions to Solaris known as Trusted Extensions (TX), which have also been subsequently integrated into OpenSolaris.

Solaris zones are a mechanism for lightweight virtualization; they provide an illusion of multiple virtual operating system instances while sharing a single kernel by placing groups of processes and objects into separate "zones" and isolating them from one another. Since TX relies on the zones mechanism, it is limited to per-zone security labels (i.e., all processes and objects within a zone share the same security label). This represents a change from the prior Trusted Solaris product and a difference from SELinux, both of which support per-process and per-object security labeling. TX is also presently limited to a fixed MLS model like its predecessors.

FMAC aims to address these limitations by supporting per-process and per-object labeling and permission checks, and by introducing flexible MAC support to OpenSolaris that can support a wide range of security models. The zone-based mechanism will still be useful as a way of providing coarse-grained isolation and namespace separation, while FMAC will be used in a complementary fashion to provide intra-zone protection, hardening of the global zone, and control over cross-zone channels. In this manner, FMAC and TX should be able to complement one another and ultimately form an integrated solution.

FMAC also aims to complement the existing Solaris privilege and Role-Based Access Control (RBAC) mechanisms. Just as SELinux provides a way to control the use of Linux superuser capabilities based on policy, FMAC will provide a way to control the use of Solaris privileges based on policy. This control will include the ability to bind privileges to specific processes and programs, to limit the use of privileges by a given process to specific objects, and to protect privileged processes from untrustworthy inputs, just as in SELinux. Unlike Linux prior to SELinux, Solaris has an existing RBAC mechanism, but at the present time the Solaris RBAC mechanism is primarily enforced by trusted applications, with the kernel only aware of the privilege model. FMAC offers a means of binding the Solaris role construct to processes and directly enforcing RBAC restrictions in the kernel, strengthening the existing mechanism.

While complementing these existing Solaris security mechanisms, FMAC will preserve existing Solaris interfaces and provide full compatibility for applications, just as SELinux provided full compatibility for Linux applications. FMAC will also provide a set of new APIs that will support security-aware applications, and these APIs will provide the same semantics as the corresponding SELinux APIs so that security-aware applications can be written portably to run on either SELinux or OpenSolaris FMAC.

Ultimately, the OpenSolaris FMAC project will provide a wider set of platforms that support the Flask architecture for flexible MAC and will expand the developer and user community for Flask. It should also help encourage independent software vendors (ISVs) to improve application-level support for flexible MAC and to provide policies for their applications.

FMAC status

The initial FMAC code base was contributed by NSA to OpenSolaris based on a version of the Flask code that predated any involvement by the Linux community. This code was then integrated into the OpenSolaris code and adapted by John Weeks, a Sun engineer who is the co-lead of the FMAC project. This code was first released publicly as an Alpha 1 release on the FMAC project web site in May 2008. When built, it produced a policy compiler and a kernel capable of loading the resulting policy into the security server.

Since that first release of FMAC, joint development by NSA and Sun Microsystems has proceeded rapidly. Support for new system calls and utilities and for per-process security labeling was introduced during the summer of 2008, leading to an Alpha 2 release in early September. Shortly thereafter, prototype support for per-file security labeling in the ZFS file system was introduced, which paved the way for supporting security context transitions upon program execution and for performing a basic set of process and file mandatory access control checks. This work produced a basic working example of how flexible mandatory access controls could be applied to an OpenSolaris system. This functionality along with subsequent enhancements to support labeling in the TMPFS file system and improve the Access Vector Cache (AVC) interfaces and implementation in FMAC was included in the Alpha 3 release made in late October 2008.

The next two major areas of focus for FMAC integration are privileges and RBAC. Significant work also remains to label and control other objects and operations provided by the Solaris kernel, create a complete example policy configuration, and integrate support for FMAC fully into user space. More advanced development and collaboration with the SELinux project in areas such as securing the desktop, policy usability and management, and labeled NFS will likely follow as FMAC matures.

Conclusion

The SELinux project has brought flexible MAC into the mainstream, achieving success both as a technology transfer vehicle and as a platform for advanced research and development. It has influenced a wide range of systems and software components, as shown most recently in the OpenSolaris FMAC project. The developer and user community that has arisen around the core ideas embodied in SELinux and OpenSolaris FMAC gives confidence that this technology will continue to be preserved and built upon in future computing systems, providing a solid foundation for addressing the threats posed by flawed and malicious applications. The advances in usability, performance, and functionality over the past several years have made these benefits far more accessible to end users.

Resources

NSA SELinux web site, http://www.nsa.gov/research/selinux

SELinux project wiki, http://selinuxproject.org

Tresys Open Source Server, http://oss.tresys com

OpenSolaris FMAC web site, http://opensolaris.org/os/project/fmac

References

[1] Loscocco P, Smalley S. Integrating Flexible Support for Security Policies into the Linux Operating System. In: Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference; June 2001.

[2] Marti D. A seatbelt for server software: SELinux blocks real-world exploits. Available from: http://www.linuxworld.comnews/2008/022408-selinux.html

[3] Red Hat. NYSE Euronext Chooses Red Hat Solutions for Flexibility and Reliable, Fast-Paced Performance. Available from: http://customers.press.redhat.com/2008/05/12/nyse/

[4] Walsh E. Application of the Flask Architecture to the X Window System Server. In: Proceedings of the 2007 SELinux Symposium; March 2007.

[5] Carter J. Using GConf as an Example of How to Create an Userspace Object Manager. In: Proceedings of the 2007 SELinux Symposium; March 2007.

[6] Gregory M. Using the Flask Security Architecture to Facilitate Risk Adaptable Access Controls. In: Proceedings of the 2007 SELinux Symposium; March 2007.

Trademarks