

[TOOL] Rule Set Based Access Control (RSBAC) for Linux

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-06/0013.html>

From: support@securiteam.com

Date: 06/05/02

From: support@securiteam.com

To: list@securiteam.com

Date: Wed, 5 Jun 2002 08:27:18 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Rule Set Based Access Control (RSBAC) for Linux

DETAILS

<<http://www.rsbac.org/>> RSBAC is a flexible, powerful and <<http://www.rsbac.org/benchmark.htm>> fast open source access control framework for current Linux kernels, which has been in stable production use since January 2000 (version 1.0.9a). All development is independent of governments and big companies, and no existing access control code has been reused.

The standard package includes a range of access control models like MAC, RC, ACL (see below). Furthermore, the <<http://www.rsbac.org/reg.htm>> runtime registration facility (REG) makes it easy to implement your own access control model as a kernel module and get it registered at runtime.

Key features:

- * Open Source (GPL) Linux kernel security extension
- * Independent of governments and big companies
- * Several well-known and new security models, e.g. MAC, ACL and RC
- * Control over individual user and program network accesses
- * Any combination of models possible
- * Easily extensible: write your own model for runtime registration
- * Support for current kernels

Securiteam: [TOOL] Rule Set Based Access Control (RSBAC) for Linux

* Stable for production use

The RSBAC framework is based on the Generalized Framework for Access Control (GFAC) by Abrams and LaPadula. All security relevant system calls are extended by security enforcement code. This code calls the central decision component, which in turn calls all active decision modules and generates a combined decision. This decision is then enforced by the system call extensions.

Decisions are based on the type of access (request type), the access target and on the values of attributes attached to the subject calling and to the target to be accessed. Additional independent attributes can be used by individual modules, e.g. the privacy module (PM). All attributes are stored in fully protected directories, one on each mounted device. Thus, changes to attributes require special system calls provided.

From version 1.2.0, all types of network accesses can be controlled individually for all users and programs. This gives you full control over their network behavior and makes unintended network accesses easier to prevent and detect.

As all types of access decisions are based on general decision requests, many different security policies can be implemented as a decision module. Apart from the built-in models shown below, the optional Module Registration (REG) allows for registration of additional, individual decision modules at runtime.

In the RSBAC version 1.2.0, the following modules are included. Please note that all modules are optional. They are described in detail in an extra text.

MAC

Bell-LaPadula Mandatory Access Control (compartments limited to a number of 64)

FC

Functional Control. A simple role based model, restricting access to security information to security officers, and access to system information to administrators.

SIM

Security Information Modification. Only security administrators are allowed to modify data labeled as security information

PM

Privacy Model. Simone Fischer-Hübner's Privacy Model in its first implementation. See our paper on <<http://www.rsbac.org/niss98.htm>> PM implementation (43K) for the National Information Systems Security Conference (NISSC 98)

MS

Malware Scan. Scan all files for malware on execution (optionally on all file read accesses or on all TCP/UDP read accesses), deny access if infected. Currently the Linux viruses Bliss.A and Bliss.B and a handful of others are detected. From v1.2.0, a generic interface allows to replace the scanning engine through a kernel module at runtime. Also, see our paper on <<http://www.rsbac.org/nordse98.htm>> Approaches to Integrated Malware Detection and Avoidance (34K) for The Third Nordic Workshop on Secure IT Systems (Nordsec'98)

FF

File Flags. Provide and use flags for dirs and files, currently `execute_only` (files), `read_only` (files and dirs), `search_only` (dirs), `secure_delete` (files), `no_execute` (files), `add_inherited` (files and dirs), `no_rename_or_delete` (files and dirs, no inheritance) and `append_only`(files and dirs). Only FF security officers may modify these flags.

RC

Role Compatibility. Defines roles and types for each target type (file, dir, dev, ipc, scd, process). For each role, compatibility to all types and to other roles can be set individually and with request granularity. For administration, there is a fine-grained separation-of-duty. Granted rights can have a time limit.

AUTH

Authorization enforcement. Controls all `CHANGE_OWNER` requests for process targets, only programs/processes with general `setuid` allowance, and those with a capability for the target user ID may become `setuid`. Capabilities can be controlled by other programs/processes, e.g. authentication daemons.

ACL

Access Control Lists. For every object there is an Access Control List, defining which subjects may access this object with which request types. Subjects can be of type user, RC role, and ACL group. Objects are grouped by their target type, but have individual ACLs. If there is no ACL entry for a subject at an object, rights are inherited from parent objects, restricted by an inheritance mask. Direct (user) and indirect (role, group) rights are accumulated. For each object, type there is a default ACL on top of the normal hierarchy. Group management has been added in version 1.0.9a. Granted rights and group memberships can have a time limit.

CAP

Linux Capabilities (new in 1.2.0). For all users and programs, you can define a minimum and a maximum Linux capability set ("set of root special rights"). This lets you e.g., run server programs as normal user, or restrict rights of root programs in the standard Linux way.

A general goal of RSBAC design has been to some day reach (obsolete) Orange Book (TCSEC) B1 level. Now it is mostly targeting to be useful as secure and multi-purposed networked system, with special interest in

Securiteam: [TOOL] Rule Set Based Access Control (RSBAC) for Linux

firewalls.

ADDITIONAL INFORMATION

The tool can be downloaded from:

<<http://www.rsbac.org/download.htm>> <http://www.rsbac.org/download.htm>

The information has been provided by <<mailto:ao@rsbac.org>> Amon Ott.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- *Previous message:* support@securiteam.com: "[NT] Internet Explorer DoS (window.open)"
 - *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)