# Rule Set Based Access Control (RSBAC)

Securing Linux from the Inside

Amon Ott <ao@rsbac.org>

## Contents:

1 Introduction
1.1 History
1.2 Motivation

2 Overview of RSBAC

3 Architecture and Implementation of the Framework
3.1 Subjects and Objects
3.2 List of Requests with Targets
3.3 Architectural Diagram
3.4 Module Registration (REG)
3.5 Network Templates

## Contents II:

4 Selection of Implemented Models
4.1 Authentication Enforcement (AUTH)
4.2 Role Compatibility (RC)
4.3 Access Control Lists (ACL)
4.4 File Flags (FF)
4.5 Linux Capabilities (CAP)
4.6 Process Jails (JAIL)
4.7 Resource Control (RES)
4.8 Pageexec Support (PAX)

## Contents III:

5 Installation
5.1 Linux Kernel
5.2 Administration tools
5.3 First Boot

6 Administration
6.1 Command Line Tools
6.2 Menues

7 Usage Areas
7.1 Workstations
7.2 Servers

# Contents IV:

8 Practical Experience
8.1 Running Systems
8.2 Stability
8.3 Performance

9 Online Ressources

10 Outlook

# 1 Introduction

1.1 History
1.2 Motivation
1.3 Design Goals

# 1.1 Introduction: History

- RSBAC Project started as Master Thesis in November 1996
- First public RSBAC version 0.9 for Linux kernel 2.0.30 on January, 9, 1998
- Current stable release 1.2.3 for kernels 2.4.26-27 and 2.6.6-8
- 1.2.4 with many changes (see Outlook)

# 1.2 Introduction: Motivation

- Classic Linux/Unix Access Control is insecure
  - Small Granularity
  - Discrete Control
    - Trusted user?
    - Malware: Invitation to Trojans and Viruses
  - Superuser root
    - Full Access
    - Too often needed
    - Too many exploits (root kits, kernel module attacks etc.)
- Better models for other administration goals
- Flexible Model selection and combination
- Good portability.

# 2 Overview of RSBAC

- Free 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.

# 2 Overview of RSBAC II

- Support for current 2.4 and 2.6 kernels
- Stable for production use since March 2000
- Several publications (see Homepage)
- Linux distributions with RSBAC: Adamantix and Gentoo Hardened
- Debian kernel patch package, Sniffix Live CD System, Simple Live-CD
- Outdated Linux distributions with RSBAC: ALTLinux Castle and Kaladix.

# 2 Overview of RSBAC III

- Access Control Framework for current Linux Kernels
- Based on Generalized Framework for Access Control (GFAC) by Abrams and LaPadula
- Flexible structure
  - Separation between enforcement (AEF), decision (ADF) and access control information (ACI)
  - Only AEF and part of ACI system dependent
  - Almost any type of model supportable
  - Model independent -> meta policy
  - Runtime Module Registration (REG)

# 2 Overview of RSBAC IV

- Powerful logging system
  - Request and decision based
  - User based
  - Program based
  - Object based.

# 3 Architecture and Implementation of the Framework

3.1 Subjects and Objects
3.2 List of Requests with Targets
3.3 Architectural Diagram
3.4 Module Registration (REG)
3.5 Network Templates

# 3.1 Architecture: Subjects and Objects

- Subjects:
  - Processes acting on behalf of users,
  - executing one program file with a set of dynamic libraries

- Object Types (Target Types):
  - FILE
  - DIR
  - FIFO
  - SYMLINK
  - DEV (devices by block/char and major:minor)
  - IPC (Inter Process Communication)
  - SCD (System Control Data)
  - USER
  - PROCESS
  - NETDEV
  - NETTEMP
  - NETOBJ

# 3.2 Architecture: List of Requests

- Requests:
  - Abstraction of what a subject wants to do with an object

- 46 Request Types:

**R_ADD_TO_KERNEL:** NONE
**R_ALTER:** IPC
**R_APPEND_OPEN:** FILE, FIFO, DEV, IPC
**R_CHANGE_GROUP:** FILE, DIR, FIFO, SYMLINK, IPC, PROCESS, NONE
**R_CHANGE_OWNER:** FILE, DIR, FIFO, SYMLINK, IPC, PROCESS, NONE
**R_CHANGE_DAC_EFF_OWNER:** PROCESS
**R_CHANGE_DAC_FS_OWNER:** PROCESS
**R_CHDIR:** DIR
**R_CLONE:** PROCESS
**R_CLOSE:** FILE, DIR, FIFO, DEV, IPC, NETOBJ

# 3.2 Architecture: List of Requests II

**R_CREATE:** DIR (where), IPC, NETTEMP, NETOBJ
**R_DELETE:** FILE, DIR, FIFO, SYMLINK, IPC, NETTEMP, NETOBJ
**R_EXECUTE:** FILE
**R_GET_PERMISSIONS_DATA:** FILE, DIR, FIFO, SYMLINK, IPC, SCD
**R_GET_STATUS_DATA:** FILE, DIR, FIFO, SYMLINK, IPC, SCD, PROCESS, NETDEV
**R_LINK_HARD:** FILE, FIFO, SYMLINK
**R_MODIFY_ACCESS_DATA:** FILE, DIR, FIFO, SYMLINK
**R_MODIFY_ATTRIBUTE:** All target types
**R_MODIFY_PERMISSIONS_DATA:** FILE, DIR, FIFO, SYMLINK, IPC, SCD, NONE
**R_MODIFY_SYSTEM_DATA:** SCD, PROCESS, NETDEV
**R_MOUNT:** FILE, DIR, DEV
**R_READ:** FILE, DIR, FIFO, DEV, IPC, NETTEMP, NETOBJ
**R_READ_ATTRIBUTE:** All target types
**R_READ_OPEN:** FILE, FIFO, DEV, IPC

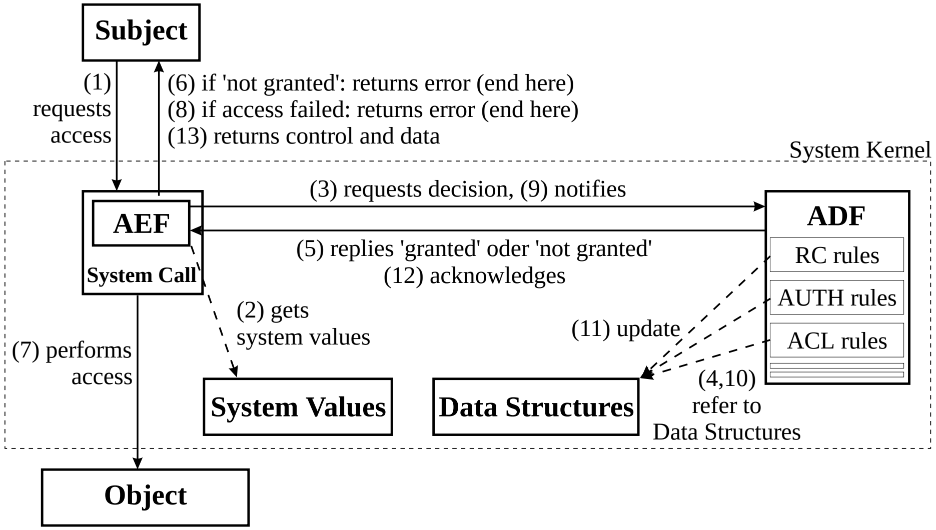
# 3.2 Architecture: List of Requests III

**R_READ_WRITE_OPEN:** FILE, FIFO, DEV, IPC
**R_REMOVE_FROM_KERNEL:** NONE
**R_RENAME:** FILE, DIR, FIFO, SYMLINK
**R_SEARCH:** DIR, SYMLINK
**R_SEND_SIGNAL:** PROCESS
**R_SHUTDOWN:** NONE
**R_SWITCH_LOG:** NONE
**R_SWITCH_MODULE:** NONE
**R_TERMINATE:** PROCESS (notify only)
**R_TRACE:** PROCESS
**R_TRUNCATE:** FILE
**R_UMOUNT:** FILE, DIR, DEV
**R_WRITE:** FILE, DIR, FIFO, DEV, SCD, NETTEMP, NETOBJ
**R_WRITE_OPEN:** FILE, FIFO, DEV, IPC
**R_MAP_EXEC:** FILE, NONE

# 3.2 Architecture: List of Requests IV

**R_BIND:** NETDEV, NETOBJ
**R_CONNECT:** NETOBJ
**R_LISTEN:** NETOBJ
**R_ACCEPT:** NETOBJ
**R_SEND:** NETOBJ
**R_RECEIVE:** NETOBJ

# 3.3 Architectural Diagram



Subject

(1) requests access

(6) if 'not granted': returns error (end here)
(8) if access failed: returns error (end here)
(13) returns control and data

System Kernel

AEF

System Call

(3) requests decision, (9) notifies

ADF

RC rules

AUTH rules

ACL rules

(5) replies 'granted' oder 'not granted'
(12) acknowledges

(2) gets system values

(11) update

(7) performs access

System Values

Data Structures

(4,10) refer to Data Structures

Object

# 3.4 Module Registration (REG)

- Runtime registration of decision functions (Rule Sets) and system calls
- Model implementation e.g. as kernel module
- Add or remove models, syscalls or generic (persistent) lists in a running system
- Easy control of module removal by the module itself
- Sample modules provided.

# 3.5 Network Templates

- Description of network endpoints
  - Ordering Number
  - Name (for human use only)
  - Address family (UNIX, INET, IPX, ...)
  - Address (E.g. 192.168.10.0 or "/dev/log")
  - Valid length (e.g. 24 Bits or 8 Byte)
  - Type (ANY, STREAM, DGRAM, ...)
  - Protocol (ICMP, TCP, UDP, ...)
  - Local network device (E.g. eth0)
  - Min and max port (E.g 1024-65535)

- Attribute values attached to templates
- Persistent default values for NETOBJ attributes

- Matched from lowest to highest template ordering number
- Used for local and remote endpoint, depending on request type.

# 3.5 Network Templates II: Examples

- Only apache may bind to port 80 at eth0

- Proxy may only connect to external addresses, not LAN
- Proxy may only accept connections from internal addresses

- Local users may only connect to mail and proxy server
- Local users (including root) may only use network families UNIX and INET.

# 4 Selection of Implemented Models

4.1 Authentication Enforcement (AUTH)
4.2 Role Compatibility (RC)
4.3 Access Control Lists (ACL)
4.4 File Flags (FF)
4.5 Linux Capabilities (CAP)
4.6 Process Jails (JAIL)
4.7 Resource Control (RES)
4.8 Pageexec Support (PAX)

# 4.1 Models: Authentication (AUTH)

- Restriction of CHANGE_OWNER with target PROCESS (setuid)

- CHANGE_OWNER capabilities (inherited from file to process): sets of reachable user IDs

- auth_may_setuid and auth_may_set_cap

- Daemon based authentication enforcable:
  - Process authenticates against daemon
  - Daemon sets capability for auth'd user at process
  - Process calls setuid.

# 4.1 Models: AUTH II

- Limited lifetime of all AUTH capabilities
- New in 1.2.2: Capabilities for effective and fs uids
- New in 1.2.3: AUTH learning mode.

# 4.2 Models: Role Compatibility (RC)

- Role and type based model:
  - User default role
  - Process current role
  - Object type
- Compatibility of roles
  - with object types (access rights in RSBAC framework granularity)
  - with other roles (change role actively)
- Forced and Initial Roles for program files.

# 4.2 Models: Role Compatibility (RC) II

- Separation of Administration Duties
  - Admin Roles
  - Assign Roles
  - Additional access rights: Admin, Assign, Access Control, Supervisor
- Lifetime limits for all compatibility settings.

# 4.3 Models: Access Control Lists (ACL)

- What subject may access which object with which requests
- Subjects:
  - RC roles (!)
  - Users
  - ACL Groups
- ACL Groups of users:
  - All users can have individual groups
  - Private and global groups
- Inheritance with masks (similar to Netware 3.xx)
- Default ACLs on top of hierarchy.

# 4.3 Models: Access Control Lists II

- Special Rights for administration:
  - Access Control
  - Forward
  - Supervisor
- Lifetime limits for all ACL entries and group memberships
- New in 1.2.3: ACL learning mode.

# 4.4 Models: File Flags (FF)

- Inheritable FILE, DIR, FIFO and SYMLINK attributes
- Valid for all users
- e.g. read-only, no-execute, secure-delete, append-only.

# 4.5 Models: Linux Capabilities (CAP)

- Minimum and maximum capability sets for users and programs
- Applied at CHANGE_OWNER on processes (setuid) and EXECUTE
- Precedence of Minimum over Maximum Sets
- Precedence of Program over User Sets
- Limit rights of root programs or extend rights of normal user programs
- E.g. limit mail server to never change network settings.

# 4.6 Models: Process Jails (JAIL)

- Preconfigured process encapsulation
- Sealed chroot jails
- No contact to processes outside the jail
- Many further restictions, some optional
- Specially limits administration and network accesses.

## 4.7 Models: Resource Control (RES)

- Minimum and maximum resource limits for users and programs
- Applied at CHANGE_OWNER on process (setuid) and EXECUTE
- Precedence of Minimum over Maximum Sets
- Precedence of Program over User Sets
- Only management of existing Linux process attributes
- Max. file size, number of processes, memory usage, etc.

## 4.8 Models: Pageexec (PAX)

- Management of process attributes for PaX kernel security extension
- PaX protects from common attack types against buggy programs
- Special protection against inserted program code
- More info: pax.grsecurity.net.

## 5 Installation under Linux

5.1 Linux Kernel
5.2 Administration tools
5.3 First Boot

## 5 Installation for Linux

- Linux Kernel (pre-patched)
  - Extract kernel source tar archive
  - Configure, touch Makefile, compile and install
  - RSBAC normal and maint kernels / Soft Mode
- Linux Kernel (patch yourself)
  - Extract RSBAC tar archive in kernel dir
  - Patch kernel (with patch-x.y.z-va.b.c.gz)
  - Apply bugfixes
  - Configure, touch Makefile, compile and install
  - RSBAC normal and maint kernels / Soft Mode
- Administration tools
  - Extract tar archive
  - ./configure && make && make install

# 5 Installation for Linux II

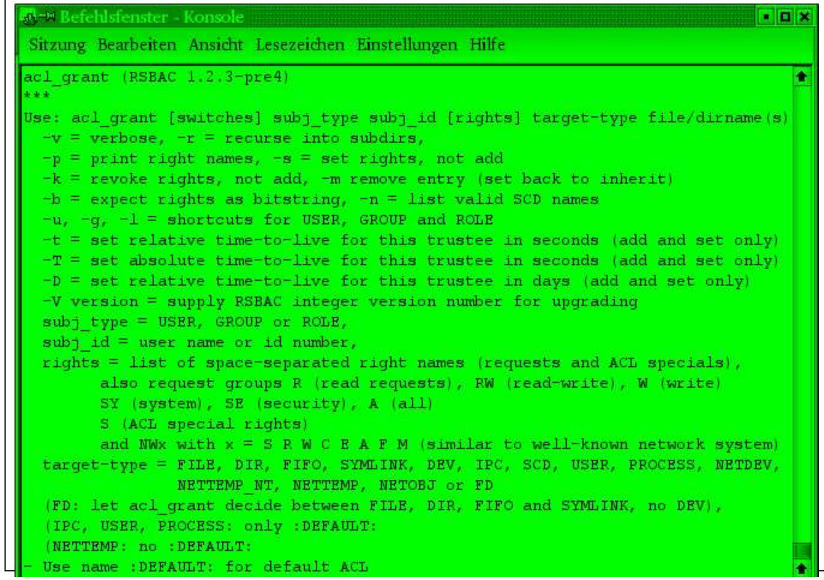
- First Boot
  - Kernel parameter rsbac_auth_enable_login
  - Add user 400 (Security Officer etc.)
  - Adjust AUTH capabilities for failed services or use AUTH learning mode.

# 6 Administration

6.1 Command Line Tools
6.2 Menues

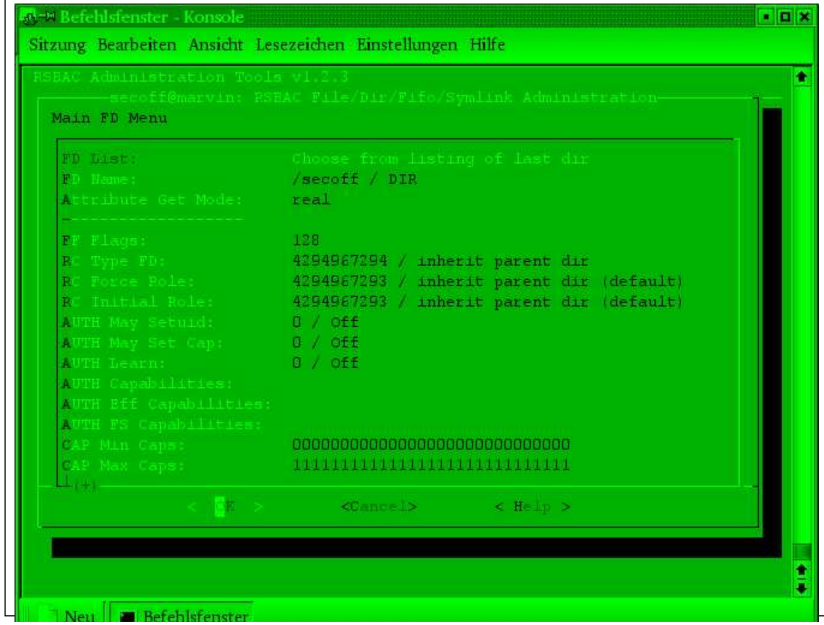
# 6.1 Administration: Command Line

- General and Model specific (RC, AUTH, ACL)



```
acl_grant (RSBAC 1.2.3-pre4)
***
Use: acl_grant [switches] subj_type subj_id [rights] target-type file/dirname(s)
  -v = verbose, -r = recurse into subdirs,
  -p = print right names, -s = set rights, not add
  -k = revoke rights, not add, -m remove entry (set back to inherit)
  -b = expect rights as bitstring, -n = list valid SCD names
  -u, -g, -l = shortcuts for USER, GROUP and ROLE
  -t = set relative time-to-live for this trustee in seconds (add and set only)
  -T = set absolute time-to-live for this trustee in seconds (add and set only)
  -D = set relative time-to-live for this trustee in days (add and set only)
  -V version = supply RSBAC integer version number for upgrading
  subj_type = USER, GROUP or ROLE,
  subj_id = user name or id number,
  rights = list of space-separated right names (requests and ACL specials),
        also request groups R (read requests), RW (read-write), W (write)
        SY (system), SE (security), A (all)
        S (ACL special rights)
        and NWx with x = S R W C E A F M (similar to well-known network system)
  target-type = FILE, DIR, FIFO, SYMLINK, DEV, IPC, SCD, USER, PROCESS, NETDEV,
                NETTEMP_NT, NETTEMP, NETOBJ or FD
  (FD: let acl_grant decide between FILE, DIR, FIFO and SYMLINK, no DEV),
  (IPC, USER, PROCESS: only :DEFAULT:
  (NETTEMP: no :DEFAULT:
  Use name :DEFAULT: for default ACL
```

# 6.2 Administration: Menues



```
RSBAC Administration Tools v1.2.3
        secoff@marvin: RSBAC File/Dir/Fifo/Symlink Administration
  Main FD Menu

   FD List:                 Choose from listing of last dir
   FD Name:                 /secoff / DIR
   Attribute Get Mode:      real
   -------------------
   FF Flags:                128
   RC Type FD:              4294967294 / inherit parent dir
   RC Force Role:           4294967293 / inherit parent dir (default)
   RC Initial Role:         4294967293 / inherit parent dir (default)
   AUTH May Setuid:         0 / Off
   AUTH May Set Cap:        0 / Off
   AUTH Learn:              0 / Off
   AUTH Capabilities:
   AUTH Eff Capabilities:
   AUTH FS Capabilities:
   CAP Min Caps:            00000000000000000000000000000000
   CAP Max Caps:            11111111111111111111111111111111
  (+)
```

# 7 Areas of use

7.1 Workstations
7.2 Server systems

# 7.1 Areas of use: Workstations

- Protection against unwanted configuration changes

- Malware protection

- Reduced administration work.

# 7.2 Areas of use: Server Systems

- Encapsulation of services
- Need-to-Know principle
- Malware protection

- Firewalls
  - DNS, Proxies, etc.
  - Advanced Protection of base system

- (Virtual) Webservers
  - Apache, Zope etc.
  - Separation of domains
  - Protection of critical data
  - Encapsulation of CGIs.

# 7.2 Areas of use: Server Systems II

- (Virtual) mail servers
  - sendmail, postfix, qmail, POP3, IMAP, Mailing Lists etc.
  - Separation of mail areas

- File servers
  - Samba, Coda, etc.
  - Separation of organizational areas

- Application servers
  - Separation between user accounts
  - Protection against user attacks

- Other servers.

# 8 Practical Experience

8.1 Running Systems
8.2 Stability
8.3 Performance

# 8.1 Experience: Running Systems

- Linux distributions Adamantix and Gentoo Hardened with RSBAC
- m-privacy TightGate-Pro
  - Extensive use of RSBAC
  - Application server system for secure Internet access
  - Strong encapsulation of all network services and users
  - Uses most of the models mentioned
- Many other stable production systems worldwide.

# 8.2 Practical Experience: Stability

- More than four years of very high stability
- SMP systems more than three years of high stability
- Few people reported problems with v1.2.3 on 2.6 kernels

# 8.3 Practical Experience: Performance

- Performance influences
  - Number and dynamic change of attribute objects
  - Number and type of decision modules
  - Logging
- Benchmarks
  - Celeron 333 system, 2.4.19 kernel, RSBAC 1.2.1
  - Three Linux kernel compile runs each
  - Runtime with framework only: +0.68% (Kernel +11.33%)
  - Runtime with RC, AUTH, network, logging enabled: +2.30% (kernel +43.02%)
  - Runtime with REG, FF, RC, AUTH, ACL, CAP, network (def. config): +4.21% (kernel +82.47%).

# 9 Online Ressources

- RSBAC Homepage: http://www.rsbac.org
- Mailing List
  - Requests: rsbac-request@rsbac.org
  - Mails: rsbac@rsbac.org
  - Archive available (see contact page)
- Adamantix
  - http://www.adamantix.org
- Gentoo Hardened Subproject RSBAC
  - http://hardened.gentoo.org/rsbac

# 10 Outlook for 1.2.4

- Kernel space user management
  - Full passwd/shadow compatible
  - Fine grained access control by all modules
  - Checking and account logic in kernel only
  - PAM and NSS modules for easy usage
  - Authentication enforcement: only setuid to authenticated uids
  - => Finally taking user control away from ordinary programs
- AUTH daemon for more secure network authentication
  - Alternative to kernel based user management
- Improved learning modes
- Many small changes (see online to-do list)
- ???

# Rule Set Based Access Control (RSBAC)

Securing Linux from the Inside

Amon Ott <ao@rsbac.org>

Thank you!