Linux Basic Security Modules Installation and Admin Guide

2nd Edition – September 2002

Contents

Preface

How to Use This Manual

This manual describes how to install and configure the Linux Basic Security Modules (LBSM). To install this module, you don’t have to recompile the Linux kernel. You have to install some kernel loadable modules and some libraries, some utility functions. Please refer to this manual when you make installation or configuration of the LBSM.

This manual consists of the following sections.

Chapter 1
This chapter provides an overview of the LBSM.

Chapter 2
This chapter explains how to install the LBSM. The LBSM is based on RedHat 7.2 and is based on the 2.4.9 Linux kernel. The LBSM is Linux kernel loadable module. So you will be able to use this on another environment.

Chapter 3
This chapter describes how to run an auditing daemon and collect auditing logs.

Chapter 4
This chapter describes how to audit user and system activities using the LBSM. It includes description how to configure audit flags for the LBSM.

Chapter 5
This chapter explains how to use the commands available with the LBSM.

If you are installing the LBSM in your own system, please refer to Chapters 1 through 3. If you are setting up LBSM-based auditing functions, please refer to Chapter 4.

Conventions

The document uses the following conventions:

text The running text is written in characters like this.

`Code` All computer variables, codes, commands and function name are shown in this font.

variable Program variables and other values, i.e., anything that is dependent on the user environment, is written in italic.

Reference References to another part of the IDA documentations are made as shown here.

For More Information

For more information on LBSM specifications and API, please refer to the following documents:

About LBSM specifications

Linux Basic Security Modules Specification (don't available)

About LBSM library programming

Linux Basic Security Modules Programming Guide (don't available)

1. Introduction

This chapter describes an overview of the LBSM.

LBSM Overview

The LBSM (**L**inux **B**asic **S**ecurity **M**odules) is a set of extended security modules for the Linux. By installing this set of modules into the Linux kernel, you can get some useful security functions on the Linux. You can use additional security functions, such as auditing functions and mandatory access control which were not available with the Linux.

LBSM Features

The LBSM has two security functions:

1. An **auditing function** necessary for auditing users and process activities (log gathering function)
2. A **mandatory access control function** with enhanced access control

○ **Auditing function**

The auditing function module allows you to monitor execution of system calls in the Linux kernel and collect corresponding logs. By grouping events to be audited using the concept of an “audit class,” it become more easily to choose target events.

○ **Mandatory access control function**

The primary concern for system operations security is controlling access to important files and processes to prevent them from being used or altered without permission. With currently available OSs, once the root privilege has been taken away, control over the entire system may be lost, resulting in significant damage. This requires skillful partitioning of administrator rights and assignment of users who control access to the system’s important files and processes separate from the root user. Use of the secure access control function makes it possible to introduce a user with the resource management privilege into the system, in addition to one with the root privilege. This user can control access to files and devices or processes. The access rights set by this user are also applied to the root user, making it possible to set up files that cannot be accessed with the root privilege. Users with the resource management privilege are managed by a mechanism with rigorous security, making it more difficult to deprive this privilege than to deprive the root privilege.

2. How to Install the LBSM

This chapter explains how to install the LBSM.

platform support

The LBSM support the following platforms. We currently tested LBSM under following platforms. But you may be able to use this on another platforms.

RedHat 7.2 (Linux kernel 2.4.9)

If you test the LBSM on another platform, please report your experience.

Getting the RPM package

We assume that you are installing the LBSM in RedHat 7.2 environment. First, download the LBSM RPM package from the following site.

http://www.jri.co.jp/

Installation using RedHat Package Manager

The LBSM is available as an RPM package that enables to be installed and removed with relative ease on systems that can use RPM packages, including RedHat, SuSE, Debian and Mandrake. There are one rpm file associated with LBSM installation.

lbsm-0.81-1.i386.rpm
This provides the necessary binaries to install the auditmodule and audit kernel components of the LBSM. This binary built for the default Redhat 7.2 (kernel2.4.9). If you have a different version of kernel, the you will need to recompile LBSM from source RPMs.

Installation LBSM component.

(1) Download the lbsm RPM.

(2) Logon root user, ie; enter the command /bin/su – at the command prompt, and enter the root password when prompted. issue the commands, as root.

```
> rpm-Uvh lbsm-0.81-1.i386.rpm
```

3. Starting up the Audit Daemon

This chapter explains how to invoke the audit daemon and begin auditing. In the as-installed condition, auditing is not possible. You must first set up the auditing function.

Enabling LBSM for use

Log in as the root user and the system should be brought to run level 1 (single-user mode) using the following command.

```
# /sbin/telinit  1
```

In single-user mode, change the directory to /etc/security/audit directory and execute the lbsmconv script installed in that directory. After execution of this script, the machine is set up and the LBSM's audit daemon is started.

```
# cd /etc/security
# ./lbsmconv
```

After the script runs, stop it using the telinit command. Next, reboot the system and start it in multiuser mode.

```
# /sbin/telinit 6
```

At this point, auditing is enabled. A log file shuold be present in the /var/audit directory.

Disabling the LBSM

If the LBSM is no longer required, you can disable it by using the lbsmunconv script. Place the system in single-user mode using the telinit command again, change the directory to /etc/security directory, and execute the lbsmunconv script.

```
# /sbin/telinit 1
# cd /etc/security
# ./lbsmunconv
```

Reboot the system and start the machine in multiuser mode.

```
# /sbin/telinit 6
```

At this point, auditing is disabled.

4. Configuring Audit Function

Here, we'll explain how to configure the LBSM's auditing function. The LBSM's auditing function makes it possible to collect audit logs at the kernel's system call level. The administrator can set the system calls to be audited in any way that suits the administrative environment. Narrowing down the system calls to be audited is important both in terms of auditing costs and analytical convenience.

Basic terms

Before discussing configuration details, the following define the basic terms used in the auditing setup.

Audit class and event

The LBSM audits security-relevant actions. These actions is related to system calls. The system actions that are auditable are defined as ***audit events*** in the /etc/security/audit_event file. The following four items are defined for each **audit event**:

number:name:description:flags

number	The event number
name	The event name
description	The description of the event
flags	Flags specifying classes to which the event is mapped

In the LBSM, a total of 382 system calls are defined as *audit events*. (For the contents of definition of the system calls, see the /etc/security/audit_event file.) Given the difficulty of specifying events to be audited individually, the LBSM introduces a concept known as the "**audit class**" to allow collective handling of multiple *audit events*. The *audit classes* are defined in the /etc/security/audit_class file. The relationship between each *audit event* and the *audit class* to which it belongs is defined in the /etc/security/audit_event file.

The *audit classes* defined in the LBSM are as follows:

	Abbreviated name	Long name	Description of class
1	no	no_class	Turns preselection of event off
2	fr	file_read	Reads data and opens a file for read
3	fw	file_write	Writes data and opens a file for write
4	fa	file_attr_acc	Accesses object attribute
5	fm	file_attr_mod	Modifies object attribute
6	fc	file_creation	Creates an object
7	fd	file_delete	Deletes an object
8	cl	file_close	Closes a file
9	pc	process	Operates on process
10	nt	network	Network event
11	ip	ipc	IPC
12	na	non_attrib	Event not attributable to user
13	ad	administrative	Administrative event
14	lo	login_logout	Login, logout
15	ap	application	Event defined by application
16	io	ioctl	ioctl
17	ex	exec	Program execution
18	ot	other	Other
19	all	all	All classes
20	ds	ida	Used for IDA (Intrusion Detection Agent System)
21	et	linux_only	Linux-inherent system call

Audit record and token

The audited event logs are written out to /proc/audit in the form of an **audit record**. Each audit record consists of multiple **audit tokens**. For LBSM, the following **audit tokens** are defined.

	Audit token	Description of token
1	header	Indicates information common to audit records (e.g., creation time)
2	trailer	Indicates the end of an audit record
3	arbitrary	Indicates any data
4	arg	Indicates arguments for a system call
5	attr	Indicates a file attribute
6	exit	Indicates the exit state of a program
7	file	Written at the beginning of an audit file
8	groups	Indicates the current group entry
9	in_addr	Indicates Internet protocol address
10	ip	Indicates Internet protocol header
11	ipc	Indicates IPC message/semaphore/shared-memory handle
12	ipc_perm	Indicates IPC access information
13	iport	Indicates port number
14	opaque	Indicates any byte
15	path	Indicates an access path to an object
16	process	Indicates process information
17	return	Indicates return value from system call
18	seq	Indicates audit token generation number
19	socket	Indicates Internet socket
20	subject	Indicates process information
21	text	Indicates text string

22	comm	Indicates command name that invoked a system call
23	parent	Indicates information on parent process for command that invoked system call
24	net-session	Indicates information on network session
25	unix-domain	Indicates information on unix domian socket

The audit record for each event is created by a combination of the above audit tokens.

Audit flag

The LBSM requires specification of an **audit flag** to define the audit event to audit. The audit flag is written using an audit class and three prefixes. Abbreviated names (two Roman alphabet characters) are used for specifying an **audit class**es that comprise an audit flag. The following prefixes are available.

Prefix	Meaning of prefix
None	Audits when event succeeds and fails
+	Audits only when event succeeds
-	Audits only when event fails
^	Turns auditing off

For example, if you want to audit only when an event belonging to the ad class fails, write the audit flag as follows:

```
-ad
```

To specify multiple classes, separate each audit class with a comma as you write the audit flag, as shown below.

```
-ad,+lo
```

Method for setting kernel event auditing

There are two methods for setting which event to audit--one, by modifying the flag line of the audit_control file located under /etc/security, and one, by altering the audit flag using the auditconfig command. Here, we'll explain the method for modifying the audit_control file. The audit_control file is a file that is loaded into the system by the audit daemon. The audit daemon loads this file at startup and notifies the kernel of the event to be audited.

The audit_control file contains four discrete lines.

Line	Label	Description of line
Audit flag line	flags:	Defines what classes of events are audited for all users
Nonattributable flag line	naflags:	Defines what classes of events are audited when an action cannot be attributed to a specific user.
Audit threshold line	minfree	Defines with a percentage the minimum free space in the directory to which the audit daemon writes logs. The minfree percentage must be greater than or equal to 0.
Directory definition line	dir:	Defines which directory will use to store the audit trail files. If the free space in this directory falls short of the minfree value, the directory is automatically switched to another directory. Therefore multiple directories may be specified on this line.

The event to audit can be defined by specifying flags: or naflags: for the audit flag.

The following is a sample of audit_control. In this example, the audit classes "lo" and "ad" are defined as the audit target for all processes, while for nonattributable events, the audit classes "lo" and "nt" are defined as the audit target. The directory in which to store the audit trail file is specified as /var/audit, and its minimum free size is set to 20%.

```
flags:lo,nt
naflags:lo,ad
minfree:20
dir:/var/audit
```

Method for setting per-user auditing

To audit a user by a different method, edit the audit_user file located under /etc/security. This lets you set a user-specific auditing state. The audit_user file consists of entries in the following form:

```
username:always-audit:never-audit
```

For username, specify the user name to which you want to apply auditing. For always-audit, specify the audit flag that is always applied to the specified user. For never-audit, specify the audit flag that will never be applied for auditing.

6. Commands

This chapter explains the utility commands used to set up LBSM and for other purposes.

The LBSM offers various tools for use in operational management. It also offers a number of LBSM APIs necessary to create new LBSM applications. Here, only the commands are explained. For more information on APIs, please refer to the ***Linux Baisc Security Modules Programming Guide***.

The LBSM has the following commands.

	Command	Description
1	audit	Controls audit daemon
2	auditconfig	Changes audit settings from command line
3	auditd	Audit daemon
4	auditreduce	Outputs audit record from audit trail file after combining
5	auditstat	Outputs kernel audit statistic
6	praudit	Outputs contents of audit trail file

audit

[Format] audit –n | -s | -t

[Description] The audit command is used to control audit daemon. This command sends a signal to audit daemon to terminate audit daemon and changes the audit trail as the output destination or enables reloading of the setup file by audit daemon.

[Options]

-n Sends a signal to audit daemon to close the current audit trail file and open a new audit trail file in the current audit directory.

-s Sends a signal to audit daemon to read audit control file. The audit daemon restore information.

-t Sends a signal to audit daemon to terminate. The audit daemon close current audit trail file, and disable auditing and die.

[Errors] If the command fails, a value equal to or less than 0 is returned.

auditconfig

[Format] auditconfig [arg]

[Description] The auditconfig command is used to set and get the kernel's audit parameters.

[Options]

-cnkconf Checks the configuration of the kernel's audit event to class mappings. If the runtime kernel's audit class mask does not match the class mask currently set in the kernel, the mismatch is reported.

-conf Configures the kernel's audit events in the class map that has been set. The runtime class mappings are changed to the specified event status in the class database file.

-getcond Displays the kernel's auditing condition. The displayed condition is "auditing", "noaudit", or "disabled". Of these, "auditing" indicates that the kernel's auditing module is generating audit records; "noaudit" indicates that auditing, although started, is not being executed; while "disabled" indicates that the auditing module is not ready to run.

-setcond[auditing|noaudit]
Sets the kernel's auditing condition to the one specified. Specifying "auditing" causes auditing to run; specifying "noaudit" causes auditing to stop.

-getclass event Displays the preselection mask that matches the specified kernel audit event. For "event", specify the kernel event number or the event name.

-setclass event audit_flag[, audit_flag ...]
Assigns the class specified by "audit_flag" to the kernel event "event". For "event", specify the event number or event name. For audit_flag, specify a two-character string representing the audit class.

-lsevent Displays the currently configured kernel event and user level event information.

-getpinfo pid Displays the audit ID, preselsection mask, terminal ID, and session ID of the process specified by "pid".

-setpmask pid flgs

Sets a preselection mask for the specified process. To write "flags", use a format similar to the one specified in audit_control.

-setsmask asid flags

Sets a preselection mask for all processes that have the session ID specified by "asid".

-setumask auid flags

Set a preselection mask for all processes that have the audit ID specified by "auid".

-lspolicy Displays the kernel's audit policies with a description for each policy.

-getpolicy Displays the audit policy specified for the kernel.

-setpolicy [+|-] policy_flag[, policy_flag ...]

Sets the kernel's audit policy. For "policy_flag", specify the string specified as audit policy. The prefix + adds a specified policy to the current audit policy. The prefix - deletes a specified policy from the current audit policy. Shown below are the correct policy flag strings.

arge Sets the audit policy so that environment variables are included in the audit record of execve system call. By default, this information is not included.

argv Sets the audit policy to make parameter information included in the audit record of execve system call. By default, this information is not included.

cnt Sets the audit policy not to suspend the process when audit resources have been used up. Audit records are deleted, with only the deleted record counts retained. By default, the process remains suspended until audit resources become available.

group Sets the audit policy to include supplementary group tokens in the audit record. By default, group tokens are not included.

path Adds supplementary path tokens to the audit record. These tokens are the path names of dynamically linked shared libraries or shell script command interpreter. By default, these tokens are not included.

trail Sets the audit policy to include trailer tokens in all audit records. By default, trailer tokens are not included.

seq Sets the audit policy to include sequence tokens in all audit records. By default, these tokens are not included. The sequence token assigns all audit records serial numbers.

[Errors] The auditconfig command returns 0 when it succeeds or a value equal to or greater than 1 when it fails.

auditd

[Format] auditd

[Description] The audit daemon generates audit traces and controls audit trails. The auditd command is used to read audit setup information from the audit_control file and to initialize the kernel with it.

When auditd receives the signal SIGUSER1, the current audit trail file is closed and another file opens. If SIGHUP is received, the setup file is reread and the kernel is reinitialized with it. If SIGTERM is received, the audit trail file is closed and auditd is terminated.

The audit daemon invokes a file named audit_warn when one of the following conditions occurs.

audit_warn soft *pathname*

If the file system specified by *pathname* exceeds the free space limit specified in the minfree item of the audit_control file, audit_warn is invoked in the format shown above. If multiple audit directories are specified in the dir item of the audit_control file, the command moves to a new audit directory and generates a new audit trail file there.

audit_warn allsoft

If all of the available audit directories exceed the free space limit, audit_warn is invoked in the format shown above. A new audit trail file is generated in one of the audit directories.

audit_warn hard *pathname*

If the file system specified by *pathname* fills up, audit_warn is invoked in the format shown above. An audit trail file is generated in another audit directory with free space.

audit_warn allhard count

If all of the audit directories fill up, audit_warn is invoked in the format shown above.

[Options] -d debug mode

Outputs a message for debugging use. The command is not invoked as a daemon and operates in the foreground.

auditreduce

[Format] auditreduce [options] [audit_trail_file ...]

[Description] The auditreduce command merges one or more audit trails and generates a new audit trail after filtering.

[Options] **Audit file select options**

These options specify which audit file is selected as the filter target.

-A Selects all audit records of the input file as the filter target. This option nullifies options -a, -b, and -d.

-C Excludes the file currently being used by auditd from the filter target. Whether any file is being used by auditd is determined by checking whether the audit file name is finished with not_terminated.

-D *suffix* Deletes the target audit file after filtering is finished. For *suffix*, specify the file name in which to store the result of filtering. If an error occurs in the middle of filtering, the audit file is not deleted regardless of whether this option is specified.

-M *machine*
Selects the audit files whose file names include the name specified by *machine* as the filter target.

-O *suffix* Sends the output result to the file specified by *suffix*. If -O is not specified, the result is forwarded to the standard output device. (If -D is specified, the result is forwarded to the file specified by -D, even though -O is not specified.) Note that if options -D and -O are both specified, the result is forwarded to the file that is specified second.

-Q Disables display of error messages even when an error occurs.

-R *pathname*
Specifies the root audit directory. The default is /etc/security/audit/*/files. If *pathname* is specified, *pathname*/*/files is used as the audit directory.

-S *server*
Uses the value specified by *server* and the value of the root audit directory to create an audit directory. In this case, /audit_root_dir/*server*/files is the audit directory.

-V Displays the audit file name as the target and the number of records processed.

Audit record select options

These options specify which audit records to select.

-a *date-time*

Selects the audit records generated after the time specified by *date-time*. A range can be specified using options -a and -b.

-b *date-time*

Selects the audit records generated before the time specified by *date-time*.

-c *audit-class*

Selects the class specified by *audit-class*. The classes specified by *audit-class* are defined in the audit_class file.

-d *date-time*

Selects the audit records generated at the same time as the time specified by *date-time*. This option cannot be used simultaneously with options -a or -b.

-e *effective-user*

Selects the audit records having the effective user ID specified by *effective-user*.

-f *effective-group*

Selects the audit records having the effective group ID specified by *effective-group*.

-g *real-group*

Selects the audit records having the real group ID specified by *read-group*.

-j *subject-ID*

Selects the audit records having the process ID specified by *subject-ID*.

-m *event*

Selects the audit records having the event number specified by *event*.

-o *object-type*=*objectID-value*

Selects audit records depending on whether the audit token specified by *object-type* contains a value that matches *objectID-value*. The items that can be specified for each are given below:

file=*pathname*

Selects audit records if the path audit token contains the value specified by *pathname*. Multiple file names can be specified by separating each entry of *pathname* with a comma. Normal representation can be used for *pathname*.

msgid=*ID*

Selects those that have the IPC key ID specified by *ID* in the ipc audit token. However, this is limited to the case in which ipc type is msg.

pid=*ID*
Selects those that have the process ID specified by *ID* in the process audit token.

semid=*ID*
Selects those that have the IPC key ID specified by *ID* in the ipc audit token. However, this is limited to the case in which ipc type is sem.

shmid=*ID*
Selects those that have the IPC key ID specified by *ID* in the ipc audit token. However, this is limited to the case in which ipc type is shm.

socket=*machine*
Selects the audit records containing the machine (IP address) specified by *machine* in the socket audit token.

-r *real-user*
Selects the audit records having the real user ID specified by *real-user*.

-u *audit-user*
Selects the audit records having the audit ID specified by *audit-user*.

[Option parameters]

date-time
The *date-time* specified in options -a, -b, or -d takes the form yyyymmdd[hh[mm[ss]]]. Specify the year for yyyy, the month for mm (01-12), and the date for dd (01-31). Specify hours for hh (00-23). Specify minutes for mm (00-59). Specify seconds for ss (00-59). The default values for hh, mm, and ss are 00. In addition, you can use +nd|h|m|s. Use a numeric value for n. The letters d, h, m, s respectively denote the day, hour, minute, and second.

event
Specify the event name or event number specified in the audit_event file.

group
Specify a group name or group ID.

pathname
Specify a file name in normal representation.

user

Specify a user name or user ID.

[Errors]

auditstat

[Format]

auditstat [-c count] [-h numline] [-i interval] [-n] [-v]

[Description]

The auditstat command displays the kernel's audit data statistics. The following fields are displayed:

aud
: A total number of audit records processed by audit system calls.

drop
: A total number of audit records having been discarded. The kernel audit policy is followed when discarding records.

enq
: A total number of audit records having been put in the kernel's audit queue.

gen
: A total number of audit records generated.

kern
: A total number of audit records having been generated by user processes.

mem
: A total memory capacity in kilobytes currently being used by the kernel audit modules.

nona
: A total number of audit records without attribute values.

[Options]

-c count
: Displays statistical information for the total "count" time. If "count" = 0, the displayed information is indeterminate. Always be sure to specify *count*.

-h numlines
: Displays the header for each "numline" of the printed statistics. By default, the header is displayed every 20 lines. If the number of lines = 0, no headers are displayed.

-i interval
: Displays statistical information every "interval" seconds. The "interval" specifies an interval in seconds at which intervals information is collected.

-n
: Displays the number of the kernel audit events currently being set.

-v
: Displays the version number of the kernel audit module.

[Errors]

The auditstat command returns 0 when it succeeds or 1 when it fails.

praudit

[Format]

```
praudit    [-lrs]    [-ddel]   [  filename  ...]
```

[Description]

The praudit command reads data from the file specified by filename and converts audit records into a format readable by humans. By default, time, userID, and groupID are converted into ASCII format. The type of audit record and event ID also are converted into ASCII format. Up to 100 files can be specified on the command line.

[Options]

-l Displays one record on one line.

-r Displays audit records in row format. Time, UID, GID, record type, and event are represented numerically. If this option is specified, the -s option is ignored.

-s Displays audit records in short format. All numeric fields are converted into ASCII before being displayed. The short ASCII is used to show the record type and event type.

-ddel Uses “del” as the field delimiter. By default, a comma (,) is used.

-t Behaves the same way as tail -Of logfile | praudit. Logs are read out from the tail of logfile and reshaped by praudit before being output.

[Errors]

A. Bug & Error

B. List of auditable System Calls

Shown here is a list of auditable system calls to be audited. This list also shows the audit records that will be output.

Audit records are comprised of a combination of audit tokens, as described before. In this list, audit records are represented using the abbreviations shown below.

h.........header token
s ....subject token
a ....attribute token
r ....return token
p..........process token
ag ....argument token
t...........text token
ipc ....ipc token
perm ....ipc-perm token
so ....socket token

The system calls marked with @ are not supported by the LBSM (LINUX version). These system calls are the ones supported by Sun Basic Security Modules, which comprise system calls in Solaris. Both types of system calls are listed here to show the difference between the two in supported system calls. The LINUX-inherent system calls not supported by Sun Basic Security Modules are listed in a separate table.

System call	Event name	Event ID	Event class	Mask	Audit record format
access	AUE_ACCESS	14	fa	0x00000004	h,p,[a],s,r
acct	AUE_ACCT	18	ad	0x00000800	h,a,s,r h,p,[a],s,r
adjtime	AUE_ADJTIME	50	ad	0x00000800	h,s,r
audit	AUE_AUDIT	211	no	0x00000000	h,s,r
auditon - get car	AUE_AUDITON_GETCAR	224	ad	0x00000800	h,s,r
auditon - get event class	AUE_AUDITON_GETCLASS	213	ad	0x00000800	h,s,r
auditon - get audit state	AUE_AUDITON_GETCOND	229	ad	0x00000800	h,s,r
auditon - get cwd	AUE_AUDITON_GETCWD	223	ad	0x00000800	h,s,r
auditon - get kernel nasj	AUE_AUDITON_GETKMASK	221	ad	0x00000800	h,s,r
auditon - get audit statistics	AUE_AUDITON_GETSTAT	225	ad	0x00000800	h,s,r
auditon - GPOLICY command	AUE_AUDITON_GPOLICY	114	ad	0x00000800	h,s,r
auditon - GQCTRL command @	AUE_AUDITON_G	145	ad	0x00000800	h,s,r

	QCNTRL				
auditon - set event class	AUE_AUDITON_SETCLASS	232	ad	0x00000800	h,[a],[a],s,r
auditon - set audit state	AUE_AUDITON_SETCOND	230	ad	0x00000800	h,[a],s,r
auditon - set kernel mask	AUE_AUDITON_SERKMASK	222	ad	0x00000800	h,[a],[a],r
auditon - set mask per session ID	AUE_AUDITON_SETSMASK	228	ad	0x00000800	h,[a],[a],s,r
auditon - reset audit statistics @	AUE_AUDITON_SETSTAT	226	ad	0x00000800	h,s,r
auditon - set mask per uid	AUE_AUDITON_SETUMASK	227	ad	0x00000800	h,[a], [a],s,r
auditon - SPOLICY command	AUE_AUDITON_SPOLICY	147	ad	0x00000800	h,[a],s,r
auditon - SQCTRL command @	AUE_AUDITON_SQCTRL	146	ad	0x00000800	h,[a],[a],[a],[a],s,r
auditsvc @	AUE_AUDITSVC	136	ad	0x00000800	h,[p],[a],s,r h,a,s,r
chdir	AUE_CHDIR	8	pc	0x00000080	h,p,[a],s,r
chmod	AUE_CHMOD	10	fm	0x00000008	h,ag,p,[a],s,r
chown	AUE_CHOWN	11	fm	0x00000008	h,ag,ag,p,[a],s,r
chroot	AUE_CHROOT	24	pc	0x00000080	h,p,[a],s,r
close	AUE_CLOSE	112	cl	0x00000040	h,ag,[p],[a],s,r
creat	AUE_CREAT	4	fc	0x00000010	h,p,[a],s,r
enter prom @	AUE_ENTERPROM	153	na	0x00000400	h,t,s,r
exec	AUE_EXEC	7	pc,ex	0x40000080	h,p,[a],s,r
execve	AUE_EXECVE	23	pc,ex	0x40000080	h,p,[a],s,r
exit prom @	AUE_EXITPROM	154	na	0x00000400	h,t,s,r
exit	AUE_EXIT	1	pc	0x00000080	h,s,r
fchdir	AUE_FCHDIR	68	pc	0x00000080	h,[p],[a],s,r
fchmod	AUE_FCHMOD	39	fm	0x00000008	h,ag,[p],[a],s,r h,ag,ag,s,r
fchown	AUE_FCHOWN	38	fm	0x00000008	h,ag,[p],[a],s,r h,ag,ag,ag,s,r
fchroot	AUE_FCHROOT	69	pc	0x00000080	h,[p],[a],s,r
fcntl	AUE_FCNTL	20	fm	0x00000008	h,ag,a,s,r h,ag,ag,s,r
fork	AUE_FORK	2	pc	0x00000080	h,[ag],s,r
fork1 @	AUE_FORK1	241	pc	0x00000080	h,[ag],s,r
fstatfs	AUE_FSTATFS	55	fa	0x00000004	h,[p],[a],s,r h,ag,s,r
getaudit	AUE_GETAUDIT	132	ad	0x00000800	h,s,r
getauid	AUE_GETAUID	130	ad	0x00000800	h,s,r
getmsg	AUE_GETMSG	217	nt	0x00000100	h,ag,ag,s,r
getmsg - accept	AUE_SOCKACCEPT	247	nt	0x00000100	h,so,ag,ag,s,r
getmsg - receive	AUE_SOCKRECEI	250	nt	0x00000100	h,so,ag,ag,s

	VE				,r
getpmsg	AUE_GETPMSG	219	nt	0x00000100	h,ag,s,r
getportaudit @	AUE_GETPORTAUDIT	149	ad	0x00000800	h,s,r
iocntl	AUE_IOCNTL	158	io	0x20000000	h,p,[a],ag,ag,s,r h,ag,ag,s,r h,ag,ag,ag,s,r h,ag,ag,ag,s,r
kill	AUE_KILL	15	pc	0x00000080	h,ag,[p],s,r h,ag,ag,s,r
lchown	AUE_LCHOWN	237	fm	0x00000008	h,ag,ag,p,[a],s,r
link	AUE_LINK	5	fc	0x00000010	h,p,[a].,p,s,r
lstat	AUE_LSTAT	17	fa	0x00000004	h,p,[a],s,r
lxstat @	AUE_LXSTAT	236	fa	0x00000004	h,p,[at],s,r
memcntl @	AUE_MEMCNTL	238	ot	0x80000000	h,ag,ag,ag,ag,ag,ag,s,r
mkdir	AUE_MKDIR	47	fc	0x00000010	h,ag,p,[a],s,r
mknod	AUE_MKNOD	9	fc	0x00000010	h,ag,ag,p,[a],s,r
mmap	AUE_MMAP	210	no	0x00000000	h,ag,ag,[p],[a],s,r h,ag,ag,ag,s,r
modcntl - bind module @	AUE_MODADDMAJ	246	ad	0x00000800	h,[t],[t],t,t,ag,t,s,r
modcntl - configure module @	AUE_MODCONFIG	245	ad	0x00000800	h,t,t,s,r
modcntl - load module @	AUE_MEDLOAD	243	ad	0x00000800	h,[t],t,s,r
modcntl - unlocad module @	AUE_MODUNLOAD	244	ad	0x00000800	h,ag,s,r
mount	AUE_MOUNS	62	ad	0x00000800	h,ag,t,p,[a],s,r h,ag,t,t,ag
msgctl -IPC_RMID command	AUE_MSGCTL_RMID	85	ip	0x00000200	h,ag,[ipc],s,r
msgctl - IPC_SET command	AUE_MSGCTL_SET	86	ip	0x00000200	h,ag,[ipc],s,r
msgcntl - IPC_STAT command	AUE_MSGCTL_STAT	87	ip	0x00000200	h,ag,[ipc],s,r
msgget	AUE_MSGGET	88	ip	0x00000200	h,[ipc],s,r
msgrcv	AUE_MSGRCV	89	ip	0x00000200	h,ag,[ipc],s,r
msgsnd	AUE_MSGSND	90	ip	0x00000200	h,ag,[ipc],s,r
munmap	AUE_MUNMAP	214	cl	0x00000040	h,ag,,ag,s,r
old nice	AUE_NICE	203	pc	0x00000080	h,s,r
open - read	AUE_OPEN_R	72	fr	0x00000001	h,p,[a],s,r
open - read,creat	AUE_OPEN_RC	73	fc,fr	0x00000011	h,p,[a],s,r
open - read,creat,trunc	AUE_OPEN_RTC	75	fc,fd,fr	0x00000031	h,p,[a],s,r

open - read,trunc	AUE_OPEN_RT	74	fd,fr	0x00000021	h,p,[a],s,r
open - read, write	AUE_OPEN_RW	80	fr,fw	0x00000003	h,p,[a],s,r
open - read, write, creat	AUE_OPEN_RWC	81	fr,fw,fc	0x00000013	h,p,[a],s,r
open - wriite,creat,trunc	AUE_OPEN_WTC	79	fw,fc,fd	0x00000032	h,p,[a],s,r
open - write,trunc	AUE_OPEN_WT	78	fw,fd	0x00000022	h,p,[a],s,r
pathconf	AUE_PATHCONF	71	fa	0x00000004	h,p,[a],s,r
pipe	AUE_PIPE	185	no	0x00000000	h,s,r
priocntlsys @	AUE_PRIOCNTLSYS	212	pc	0x00000080	h,ag,ag,s,r
process dump core @	AUE_CORE	111	fc	0x00000010	h,p,[a],ag,s,r
putmsg	AUE_PUTMSG	216	nt	0x00000100	h,sg,sg,s,r
putmsg-connect	AUE_SOCKCONECT	248	nt	0x00000100	h,so,ag,ag,s,r
putmsg-send	AUE_SOCKSND	249	nt	0x00000100	h,so,ag,ag,s,r
putpmsg	AUE_PUTPMSG	218	nt	0x00000100	h,ag,s,r
readlink	AUE_READLINK	22	fr	0x00000001	h,p,[a],s,r
rename	AUE_RENAME	42	fc,fd	0x00000030	h,p,[a],[a],s,r
rmdir	AUE_RMDIR	48	fd	0x00000020	h,p,[a],s,r
semctl - getall	AUE_SEMCTL_GETALL	105	ip	0x00000020	h,ag,ipc,s,r
semctl - GETCNT command	AUE_SEMCTL_GETCNT	102	ip	0x00000200	h,ag,[ipc],s,r
semctl - GETPID command	AUE_SEMCTL_GETPID	103	ip	0x00000200	h,[ipc],s,r
semctl - GETVAL command	AUE_SEMCTL_GETVAL	104	ip	0x00000200	h,ag,[ipc],s,r
semctl - GETZCNT command	AUE_SEMCTL_GETZCNT	106	ip	0x00000200	h,ag,[ipc],s,r
semctl - IPC_RMID command	AUE_SEMCTL_RMID	99	ip	0x00000200	h,ag,[ipc],s,r
semctl - IPC_SET command	AUE_SEMCTL_SET	100	ip	0x00000200	h,ag,[ipc],s,r
semctl - SETALL command	AUE_SEMCTL_SETALL	108	ip	0x00000200	h,ag,[ipc],s,r
semctl - SETVAL command	AUE_SEMCTL_SETVAL	107	ip	0x00000200	h,ag,[ipc],s,r
semctl - IPC_STAT command	AUE_SEMCTL_STAT	101	ip	0x00000200	h,ag,[ipc],s,r
semget	AUE_SEMGET	109	ip	0x00000200	h,[ipc],s,r
semop	AUE_SEMOP	110	ip	0x00000200	h,ag,[ipc],s,r
setaudit @	AUE_SETAUDIT	133	ad	0x00000800	h,ag,ag,ag,ag,ag,ag,s,r h,s,r
setauid	AUE_SETAUID	131	ad	0x00000800	h,sg,s,r
setegid	AUE_SETEGID	214	pc	0x00000080	h,ag,s,r
seteuid	AUE_SETEUID	215	pc	0x00000080	h,ag,s,r
old setgid	AUE_SETGID	205	pc	0x00000080	h,ag,s,r
setgroups	AUE_SETGROUPS	26	pc	0x00000080	h,ag,s,r
setpgrp	AUE_SETPGRP	27	pc	0x00000080	h,s,r
setrlimit	AUE_SETRLIMIT	51	ad	0x00000800	h,s,r

old setuid	AUE_OSETUID	200	pc	0x00000080	h,ag,s,r
shmat	AUE_SHMAT	96	ip	0x00000200	h,ag,ag,[ipc],[perm],s,r
shmctl - IPC_RMID command	AUE_SHMCTL_RMID	92	ip	0x00000200	h,ag,[ipc],s,r
shmctl - IPC_SET command	AUE_SHMCTL_SET	93	ip	0x00000200	h,ag,[ipc],s,r
shmctl - IPC_STAT	AUE_SHMAT_STAT	94	ip	0x00000200	h,ag,[ipc],s,r
shmdt	AUE_SHMDT	95	ip	0x00000200	h,ag,[ipc],[perm],s,r
stat	AUE_STAT	16	fa	0x00000004	h,p,[a],s,r
statfs	AUE_STATFS	54	fa	0x00000004	h,p,[a],s,r
statyfs @	AUE_STATVFS	234	fa	0x00000004	h,p,[a],s,r
stime	AUE_STIME	201	ad	0x00000800	h,s,r
symlink	AUE_SYMLINK	21	fc	0x00000010	h,t,p,[a],s,r
sysinfo	AUE_SYSINFO	39	fc	0x00000800	h,ag,t,s,r
system booted	AUE_SYSTEMBOOT	113	na	0x00000400	h,t,r
umount - old version	AUE_UMOUNT	12	ad	0x00000800	h,p,[a],s,r
unlink	AUE_UNLINK	6	fd	0x00000008	h,p,[a],s,r
old utime	AUE_UTIME	202	fm	0x00000008	h,p,[a],s,r
utimes	AUE_UTIMES	49	fm	0x00000008	h,p,[a],s,r
utssys @	AUE_USTSYS	233	ad	0x00000800	h,p,[a],s,r
vfork @	AUE_VFORK	25	pc	0x00000080	h,ag,s,r
vtrace @	AUE_VTRACE	36	pc	0x00000080	h,s,r
xmknod	AUE_XMKNOD	240	fc	0x00000010	h,p,[a],s,r
xstat @	AUE_XSTAT	235	fa	0x00000004	h,p,[a],s,r

LINUX-inherent System Calls

System call	Event name	Event ID	Event class	Mask	Audit record format
alarm	AUE_ALARM	275	et	0x00010000	h,s,r
getsid	AUE_GETSID	276	et	0x00010000	h,s,r
setsid	AUE_SETSID	277	et	0x00010000	h,s,r
getpriority	AUE_GETPRIORITY	278	et	0x00010000	h,s,r
ioperm	AUE_IOPERM	279	et	0x00010000	h,s,r
iopl	AUE_IOPL	280	et	0x00010000	h,s,r
modify_ldt	AUE_MODIFY_LDT	281	et	0x00010000	h,s,r
create_module	AUE_CREATE_MODULE	282	et	0x00010000	h,s,r
init_module	AUE_INIT_MODULE	283	et	0x00010000	h,s,r
delete_module	AUE_DELETE_MODULE	284	et	0x00010000	h,s,r
get_kernel_syms	AUE_GET_KERNEL_SYMS	285	et	0x00010000	h,s,r
nanosleep	AUE_NANOSLEEP	286	et	0x00010000	h,s,r

pause	AUE_PAUSE	287	et	0x00010000	h,s,r
personality	AUE_PERSONALITY	288	et	0x00010000	h,s,r
sched_getparam	AUE_SCHED_GETPARAM	289	et	0x00010000	h,s,r
sched_setparam	AUE_SCHED_SETPARAM	290	et	0x00010000	h,s,r
sched_getscheduler	AUE_SCHED_GETSCHEDULER	291	et	0x00010000	h,s,r
sched_setscheduler	AUE_SCHED_SETSCHEDULER	292	et	0x00010000	h,s,r
sched_get_priority_min	AUE_SCHED_GET_PRIORITY_MIN	293	et	0x00010000	h,s,r
sched_get_priority_max	AUE_SCHED_GET_PRIORITY_MAX	294	et	0x00010000	h,s,r
sched_yield	AUE_SCHED_YIELD	295	et	0x00010000	h,s,r
sched_rr_get_interval	AUE_SCHED_RR_GET_INTERVAL	296	et	0x00010000	h,s,r
getgroups	AUE_GETGROUPS	297	et	0x00010000	h,s,r
getitimer	AUE_GETITIMER	298	et	0x00010000	h,s,r
getrlimit	AUE_GETRLIMIT	299	et	0x00010000	h,s,r
getrusage	AUE_GETRUSAGE	300	et	0x00010000	h,s,r
sysctl	AUE_SYSCTL	301	et	0x00010000	h,s,r
gettimeofday	AUE_GETTIMEOFDAY	302	et	0x00010000	h,s,r
times	AUE_TIMES	303	et	0x00010000	h,s,r
wait4	AUE_WAIT4	304	et	0x00010000	h,s,r
dbflush	AUE_BDFLUSH	305	et	0x00010000	h,s,r
llseek	AUE_LLSEEK	306	et	0x00010000	h,s,r
readdir	AUE_READDIR	307	et	0x00010000	h,s,r
sync	AUE_SYNC	308	et	0x00010000	h,s,r
fsync	AUE_FSYNC	309	et	0x00010000	h,s,r
fdatasync	AUE_FDATASYNC	310	et	0x00010000	h,s,r
sysfs	AUE_FSYSFS	311	et	0x00010000	h,s,r
uselib	AUE_USELIB	312	et	0x00010000	h,s,r
umask	AUE_UMASK	313	et	0x00010000	h,s,r
vhangup	AUE_VHANGUP	314	et	0x00010000	h,s,r
mprotect	AUE_MPROTECT	315	et	0x00010000	h,s,r
mremap	AUE_MREMAP	316	et	0x00010000	h,s,r
msync	AUE_MSYNC	317	et	0x00010000	h,s,r
getpid	AUE_GETPID	318	et	0x00010000	h,s,r
getppid	AUE_GETPPID	319	et	0x00010000	h,s,r
getuid	AUE_GETUID	320	et	0x00010000	h,s,r
getgid	AUE_GETGID	321	et	0x00010000	h,s,r
getegid	AUE_GETEGID	322	et	0x00010000	h,s,r
geteuid	AUE_GETEUID	323	et	0x00010000	h,s,r
setpgid	AUE_SETPGID	324	et	0x00010000	h,s,r
getpgid	AUE_GETPGID	325	et	0x00010000	h,s,r
getpgrp	AUE_GETPGRP	326	et	0x00010000	h,s,r
dup	AUE_DUP	327	et	0x00010000	h,s,r