AIX System Security Audit

Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

About This Document

This document is intended to simplify the use of the auditing system provided in AIX and is applicable to all versions of AIX. It includes information on what auditing offers, what its requirements are, and what common problems are encountered. It's intention is not to answer every question about auditing, but to provide a starting point for understanding and setting up auditing.

Available Documentation

Documentation for System Auditing can be found in chapter 5 of the "System Management Guide" for AIX version 3.2, chapter 3 of the "System Management Guide: Operating System and Devices" for AIX version 4.1, and chapter 8 in IBM red book "Elements of Security: AIX 4.1" (GG24-4433-00). Fax Document # 2226, "Setting up An Auditing System to Monitor Cron Events" is also available.

Auditing Overview

The auditing subsystem provides the means to record security related information and to alert system administrators of potential and actual violations of the system security policy. The information collected by auditing includes, the name of the auditable event, the status (success/failure) of the event, and any additional event-specific information related to security auditing.

Auditing Events and Objects

A list of audit events built into AIX along with a list of pre-defined audit objects can be found in the file, /etc/security/audit/events.

In general, auditing events are defined at the system call level. Thus, a single operation at the command line would result in records of several events in the

audit trail. For example, when a user views a file using the "cat" or "more" command, one would see the following records logged into the audit trail;

FILE_Open (user opens the file)
FILE_Read (user reads the file)
FILE_Write (user writes the file to standard output)
PROC_Create (process creation for more OR cat)
PROC_Execute (command execution), and;
PROC_Delete (process completion).

Auditing all possible events for all users can produce a huge amount of data. Through audit controls (i.e., modifying the configuration files) one can select events to be recorded.

Audit events are grouped into classes. You can define which events are in a class. While the class names are arbitrary, they rather than individual event names are associated with userids when the audit subsystem is active.

Auditing objects, for the most part, are just individual files you want to monitor. Three operations can be audited: read, write, and execution. Objects are not associated with userids. Audit records are generated whenever an audited object is referenced by any user (including root).

Adding additional audit objects is accomplished by extending the /etc/security/audit/objects file.

Auditing Mode: BIN and STREAM

There are two modes of operation for audit: BIN and STREAM. BIN mode writes the audit trail to alternating temporary files (bins), then to a single trail file. STREAM mode writes to a circular buffer that is read synchronously through an audit pseudo-device (/dev/audit).

One can start audit in one (OR both) of these two modes.

Using audit configuration setup shipped with AIX, /etc/security/audit/config, the BIN mode alternates between /audit/bin1 and /audit/bin2. When one bin is full (binsize parameter determines the size of the bin), the audit switches to the other BIN file while adding the accumulated data in the first file to the audit trail (defined in /etc/security/audit/bincmds), /audit/trail. One should use "audit shut-down" to be certain that all audit records have been added to /audit/trail. The BIN mode audit record is in binary format. It can be read with audit commands such as "auditpr".

In STREAM mode, the default AIX configuration provides a program to read the STREAM buffer and processes each record with the commands found in /etc/security/audit/streamcmds. These commands format the output into human readable form and write it in /audit/stream.out. This file is NOT cumulative; it is

restarted every time the audit is restarted. The STREAM audit trail can be read in real time by reading /audit/stream.out, or by sending output directly to a terminal or printer.

Starting And Stopping Audit

There are five audit subcommands for invoking audit.

```
audit start    - to activate the audit subsystem. This is
                 the only correct way to start audit.
audit shutdown - to stop auditing subsystem, processing
                 final BIN records and removing the
                 /audit/auditb file that is used as an
                 "active" indicator by the audit modules.
audit off      - to suspend auditing temporarily.
audit on       - to resume audit after audit off.
audit query    - to display the status of auditing.
```

Note: Using audit commands in the wrong order can confuse the auditing subsystem. If auditing subsystem is confused, one can reset everything by deleting all files in /audit directory (except trail, stream.out and bin files).

Auditing can be run at the discretion of the system administrator. Depending on the environment, it is usually not necessary to have audit running at all times. If it is configured to monitor a large number of events or objects at all times, the amount of data generated would be so substantial its overhead would outweigh its benefit. It is worthwhile taking time to configure auditing to collect selected information.

If you want auditing to start at system startup, add the following line to the /etc/rc file, just prior to the line reading "dspmsg rc.cat 5 'Multi-user initialization completed\n'"

```
/usr/sbin/audit start
```

If auditing is running at all times, make sure to run or to add

```
"/usr/sbin/audit shutdown"
```

in the /usr/sbin/shutdown script to properly shutdown auditing when shutting down the system.

Auditing Configuration

All auditing related configuration files reside in /etc/security/audit. The /etc/security/audit/config file contains the key audit controls. It has these stanzas:

```
start            -  which specifies whether BIN or STREAM
                    (or both) should be used for auditing
bin and stream   -  Contain controls for each mode. The names
                    of the BIN files are specified here.
classes          -  Defines several groups (classes) of
                    auditing events.
```

The predefined classes are: general, objects, SRC, kernel, files, SVIPC, mail, cron, and TCPIP. You can define new classes using the auditing events in /etc/security/audit/events file. All audit classes except the objects class are associated with userids. For example, you may audit the events defined as "general" and "TCPIP" for user root.

```
user stanza      -  Lists specified users and the audit classes
                    assigned to them.
```

An example of this stanza is:

```
          users:
                       root = general
                       joe = general, files, TCPIP
```

When auditing starts, it ALWAYS audits the events specified for every userid defined in the config file and ALL the objects defined in /etc/security/audit/objects. If you do not want objects audit records, you should remove or comment out (using an *) the objects defined in the objects file.

The objects file contains all objects to be audited when auditing is active. A user defined object looks like:

```
                /home/joe/my.stuff
                        r = "JOE_READ"
                        w = "JOE_WRITE"
```

The names "JOE_READ" and "JOE_WRITE" are referenced in the /etc/security/events file to define the format of "auditpr" output:

```
                JOE_READ = printf "%s"
                JOE_WRITE = printf "%s"
```

Note: There is no need to add the newly added objects to the objects stanza in the /etc/security/audit/config file, as the objects line is not referenced. Only the objects file is referenced.

The streamcmds file has commands that are executed for STREAM audit records. The default file contains one command:

```
/usr/sbin/auditstream | auditpr > /audit/stream.out &
```

Adding -v flag for the "auditpr" command improves this command at the expense of having more information. Without -v, full path names for files are not shown in the audit output, only file descriptors are recorded.

The "bincmds" file contains commands that are executed whenever a BIN file fills or when audit is shut down. The file distributed reads like:

```
/usr/sbin/auditcat -p -o $trail $bin
```

The environment variables in the above command are defined while audit is active. One might add the "auditselect" command to select specific events, reducing the amount of audit records.

Auditing A User

You can add audit classes for a user thru smit.

```
smit
        Security and Users
             Users
                Change / Show Characteristics of a User
*User NAME              [joe]

AUDIT classes           [general, files]
```

You now should see a user stanza for joe in /etc/security/audit/config file.

At the command line, one can edit the /etc/security/config file. In the classes stanza, new class can be added:

```
procmon = PROC_Create, PROC_Delete, PROC_Execute
```

In the users stanza, one may have:

```
joe = procmon
```

The newly assigned audit classes will take effect at the next login for user joe.

The BIN mode audit trail can be read with

```
auditpr -v < /audit/trail | more
```

The STREAM mode audit file /audit/stream.out can be viewed directly. Remember that the /audit/stream.out file is rewritten each time you start the auditing subsystem. So one might want to save the old stream.out before starting audit.

If you do not want objects audit records when auditing a userid, you can comment out the objects defined in the /etc/security/audit/objects file or rename this file.

Auditing An Object

As an example, we will audit all processes writing to /etc/utmp file.

1. Edit the /etc/security/audit/objects file to add:

```
/etc/utmp:
w = "UTMP_WRITE"
```

2. Edit the /etc/security/audit/events file to include:

```
* /etc/utmp
UTMP_WRITE = printf " %s "
```

   The audit record looks like:

```
UTMP_WRITE      root    OK      Wed Jul 12 12:12:25 1995  init
```

In this case, the init process owned by root wrote to the file.

Note: The length of an audit event/object name cannot exceed 15 characters. This limit is defined in the header file /usr/include/sys/audit.h, ah_event [16]. The error message

```
"auditevents (): Invalid argument"
```

usually indicates an invalid event/object name.

Disk Space Consideration

Each record in the audit trail takes about 50 to 150 bytes depending on what mode is used and whether the verbose mode flag is specified. This means that 1 MB of data could contain about 6800 entries.

Understanding The Output

It is important to specify what information you wish to review while auditing. Though you may configure auditing to record events of interest, there may still be too much data to be useful, when viewed all at once. The "auditselect" command can be used with "auditpr" to sort through volumes of information and pull out only that which is needed for a specific report. It can be used to pull all data from a specific time period, for a specific user, or a specific event or any combination of the above.

The following command acts as an example:

```
/usr/sbin/auditselect -f /audit/pick \
/audit/trail | /usr/sbin/auditpr -v
```

where the /audit/pick file reads:

```
command == rlogin && \
time >= 08:00:00 && time <= 17:00:00 && \
data >= 04/01/96 && date <= 04/12/96
```

This command reports the use of the rlogin command within the specific time interval (8AM-5PM between April 1 and April 12).

The compressed trail data from the binmode auditing is not in the same format at AIX version 3.2 as 4.1. There is a utility to convert the data from a pre-AIX Version 4 format to the version 4 format. It is a command called "auditconv".

Common Problems With Auditing

Errors When Starting Audit

- There are certain errors that appear when executing "audit start".

  One error message reads like:

  ```
  ** failed setting kernel audit objects
  ```

  This occurs when there is a syntax error in the /etc/security/audit/objects file.

- Another error message is:

  ```
  auditbin: ** failed backend command
  /etc/auditcat -p -o /audit/trail -r /audit/bin1
  ```

  This error can be corrected by removing or renaming the bin files. It is sometimes helpful to run "audit shutdown" again then retry "audit start".

- Error in config File

  It is necessary to have the "user" stanza in the /etc/security/audit/ config file or you will get the following error when starting audit:

  ```
  Unable to find the user stanza in /etc/security/audit/config
  ```

If it is not obvious that the user stanza is missing, verify that each of the classes is defined on a single continuous line.

Data Overload

The way that cron and the TCP/IP code is written, they each set up their own set of audit events. These events will get written into the audit trail regardless of how your config files are set up. The workaround is to use audit select to exclude these events when generating the audit report. TCP/IP sessions, ftpd, rexecd, and telnetd all call auditproc() to set up process auditing using the class "tcpip" in /etc/security/audit/config. The same thing is done in the cron code (at, cron, and cronadm) for the "cron" class in /etc/seccurity/audit/config. These events will be written into the audit trail. The best thing to do is to filter them using "auditselect".

An example follows:

```
auditselect -e"event!=AT_JobAdd && event!=AT_JobRemove && ..."
```

This will exclude events AT_JobAdd and At_JobRemove etc.

Or you could select on the command name:

```
auditselect -e"command!=cron && command!=at && ..."
```

This excludes recording related to commands, at, cron, etc.

APARs for Auditing Subsystem

APARs for AIX 3.2.5

IX57470 - mkuser doesn't update auditclasses
IX55919 - security audit function fix
IX55000 - auditselect always returns an exit code of 4
IX53163 - auditpr not printing complete trail
IX51882 - auditcat -r -o trailf fails when trailf is empty
IX50831 - audit shutdown does not wait for auditiing to end
IX46801 - mkdir and rmdir generate audit events
IX44750 - chdir and chroot generate audit records

APARs AIX 4.1.x

IX58784 - activate auditing documentation
IX58783 - update security audit functionality
IX57815 - mkuser doesn't update auditclasses
IX57273 - cron generates cron audit events only
IX55919 - security audit function fix
IX54705 - audit trail corrupt
IX53962 - chmod command generates FILE_Mode record
IX53683 - auditselect always returns exit code of 4
IX53055 - auditcat -r -o trailf fails when trailf is empty
IX52720 - rshd does not audit creation of user session
IX51254 - audit shutdown does not wait for auditing to end

12/02/96, 4FAX# 4824

Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Note: If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

END OF DOCUMENT (audit.32-41.cmd, 4FAX #4824)