



ArcSight SmartConnectors

Software Version: 8.4.3

Configuration Guide for Oracle Solaris Basic Security Module SmartConnector

Document Release Date: October 2023

Software Release Date: October 2023

Legal Notices

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Copyright Notice

Copyright 2023 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Trademark Notices

“OpenText” and other Open Text trademarks and service marks are the property of Open Text or its affiliates. All other trademarks or service marks are the property of their respective owners.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<https://www.microfocus.com/support-and-services/documentation>

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://www.microfocus.com/documentation/arcsight/

Contents

Configuration Guide for Oracle Solaris Basic Security Module SmartConnector	5
Product Overview	7
Configuration	8
BSM Auditing	8
Audit Events	8
Audit Records	9
Audit Flags	9
The Audit Trail	9
The auditreduce Command	10
Overview of Audit Setup	10
Basic Configuration Steps	12
Basic Configuration	12
Audit Startup	14
Audit Control	15
Audit Class and Audit Event	15
General Events - lo (login) and ad (administrative)	16
Process Events - ex (execution) and pc (process control)	16
File Attribute Modification - fm Class	16
Other File Actions - fc (create), fr (read), fw (write), fd (delete)	17
Custom Audit Classes	17
Audit Log File Rotation	17
BSM Caveats	18
Installing the SmartConnector	19
Preparing to Install the SmartConnector	19
Installing and Configuring the SmartConnector	19
Device Event Mapping to ArcSight Fields	21
Oracle Solaris 10 and 11 BSM Common Mappings to ArcSight ESM Fields	21
Event Type AUE_su	22

Event Type AUE_rexecd	22
Event Type AUE_passwd	22
Event Type AUE_rexd	23
Event Type AUE_ftp_access	23
Event Type AUE_login-ssh	23
Event Type AUE_role_login	23
Event Type AUE_newgrp_login	23
Event Type AUE_zlogin	24
Event Type AUE_sudo	24
 Send Documentation Feedback	 25

Configuration Guide for Oracle Solaris Basic Security Module SmartConnector



Solaris versions 8 and 9 are no longer supported for SmartConnector installation and have been removed from connector configuration selections. To continue running these versions with the SmartConnector, do not upgrade the connector. To upgrade, you must be using Solaris version 10 or later.

This guide provides information for installing the SmartConnector for Oracle Solaris Basic Security Module on a Solaris platform and configuring the device for audit log event collection.

Intended Audience

This guide provides information for IT administrators who are responsible for managing the ArcSight software and its environment.

Additional Documentation

The ArcSight SmartConnector documentation library includes the following resources:

- [Technical Requirements Guide for SmartConnector](#), which provides information about operating system, appliance, browser, and other support details for SmartConnector.
- [Installation and User Guide for SmartConnectors](#), which provides detailed information about installing SmartConnectors.
- [Configuration Guides for ArcSight SmartConnectors](#), which provides information about configuring SmartConnectors to collect events from different sources.
- [Configuration Guide for SmartConnector Load Balancer](#), which provides detailed information about installing Load Balancer.

For the most recent version of this guide and other ArcSight SmartConnector documentation resources, visit the [documentation site for ArcSight SmartConnectors 8.4](#).

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the comment on this topic link at the bottom of each page of the online documentation, or send an email to MFI-Documentation-Feedback@opentext.com.

For specific product issues, [contact Open Text Support for Micro Focus products](#).

Product Overview

The Oracle Solaris Basic Security Module (BSM) provides a security auditing subsystem. The auditing mechanism lets administrators detect potential security breaches. It performs kernel auditing and provides a device allocation mechanism for the Solaris operating system, which enable Solaris to meet C2 level criteria.



C2 is a security rating originally defined in the Trusted Computer System Evaluation Criteria (TCSEC), published by the United States National Computer Security Center (NCSC), commonly referred to as the Orange Book.

The BSM audit trail is written to binary files on the local system (or NFS mount). Audit records are initiated from two distinct places in Solaris-privileged user land programs (such as login) and the Solaris kernel. All security-sensitive kernel system calls generate an audit record when BSM auditing is enabled.



Reading or executing privileged audit files requires administrator access.

BSM is not enabled by default under Solaris. The administrator is required to run the `bsmconv` script to set up the initial auditing environment for the system.

Configuration

This section has the following topics:

BSM Auditing

For complete information about BSM auditing, see the *Oracle Solaris Basic Security Module Guide*. Additional helpful information includes "Solaris BSM Auditing" by Hal Pomeranz of Deer Run Associates (<http://www.deer-run.com/~hal/sysadmin/SolarisBSMAuditing.html>).

Successful auditing depends upon two other security features: identification and authentication. At login, after the user supplies a user name and password, a unique audit ID is associated with the user's process. The audit ID is inherited by every process started during the login session. Even if a user changes identity, all actions performed are tracked with the same audit ID.

After audit data is collected, audit reduction and interpretation tools allow the examination of interesting parts of the audit trail. For example, you can look at audit records for individual users or groups, look at all records for a certain type of event on a specific day, or select records that were generated at a certain time of day.

Audit Events

System actions that are auditable are defined as audit events in the `/etc/security/audit_event` file. Each auditable event is defined in the file by a symbolic name, an event number, a set of pre-selection classes, and a short description. Most events are attributable to an individual user. However, some events are non-attributable because they occur at the kernel-interrupt level or before a user is identified and authenticated. Non-attributable events are auditable as well.

Each audit event is also defined as belonging to an audit class or classes. By assigning events into classes, an administrator can more easily deal with large numbers of events. When naming a class, you simultaneously address all of the events in that class. The mapping of audit events to classes is configurable and the classes themselves are configurable. These configuration changes can be made in the `audit_event` file.

Whether an auditable event is recorded in the audit trail depends upon whether the administrator pre-selects a class for auditing that includes the specific event. Out of 32

possible audit classes, 18 are defined. The 18 classes include the two global classes `all` and `no`.

Audit Records

Each audit record describes the occurrence of a single audited event and includes such information as who performed the action, which files were affected, what action was attempted, and where and when the action occurred.

Audit records are collected in a trail and can be converted to human-readable format by using `praudit` (see the `praudit(1M)` man page).

Audit Flags

Audit flags indicate classes of events to audit. Machine-wide defaults for auditing are specified for all users on each machine by flags in the `audit_control` file. The system administrator can modify what is audited for individual users by putting audit flags in a user's entry in the `audit_user` file. The audit flags also are used as arguments to `auditconfig` (see the `auditconfig(1M)` man page).

The Audit Trail

The audit trail is created by the audit daemon (see the `auditd(1M)` man page). The audit daemon starts on each machine when the machine is brought up. After `auditd` starts at boot time, it collects the audit trail data and writes the audit records into audit files.

The audit daemon runs as `root`. All files it creates are owned by `root`. Even when `auditd` has no classes to audit, it continuously operates, looking for a place to put audit records. The `auditd` operations continue even if the rest of the machine's activities are suspended due to the kernel's audit buffers becoming full. The audit operations can continue because `auditd` is not audited.

Only one audit daemon can run at a time. An attempt to start a second one results in an error message and the new one exits. If there is a problem with the audit daemon, try using `audit -t` to terminate `auditd` gracefully, then restart it manually.

The `audit_warn` script is run by `auditd` whenever the daemon switches audit directories or encounters difficulty (such as a lack of storage). As distributed, the `audit_warn` script sends mail to an `audit_warn` alias and sends a message to the console. Your site should customize `audit_warn` to suit your needs.

When `auditd` starts on each machine, it creates the file `/etc/security/audit_data`.

To keep audit files at a manageable size, a cron job can be set up that periodically switches audit files (see the `cron(1M)` man page). Intervals range from once per hour to twice per day, depending upon the amount of audit data being collected. The data then can be filtered to remove unnecessary information and then compressed.

The `auditreduce` Command

Use `auditreduce` to merge together audit records from one or more input audit files or to perform a post selection of audit records. See the `auditreduce(1M)` man page. To merge the entire audit trail, the system administrator enters the command on the machine on which all the audit file systems for the distributed system are mounted.

When multiple machines running BSM are administered as part of a distributed system, each machine performs auditable events, and each machine writes audit records to its own machine-specific audit file. Using `auditreduce`, you can read the logical combination of all audit files in the system as a single audit trail without regard to how the records were generated or where they are stored.

`auditreduce` selects records from one or more audit files and merges them into a single, chronologically ordered output file. The merging and selecting functions of `auditreduce` are logically independent. `auditreduce` selects messages from the input files as the records are read, before the files are merged and written to disk.

Without options, `auditreduce` merges the entire audit trail (which consists of all of the audit files in all of the sub-directories in the audit root directory `/etc/security`) and sends all the audit records to standard output. The location of this directory will be required during SmartConnector installation.

The SmartConnector accesses the directory you specify and invokes the `praudit` command to generate the ASCII output the connector uses to map event data.

Overview of Audit Setup

The following steps are included here to provide an overview of what is required to set up audit directories and specify which audit classes will be audited.

- 1 Format and partition the disks to create the dedicated audit partition or partitions. A rule of thumb is to assign 100 MB of space for each machine that will be on the distributed system; however, the disk space requirements at your site will be based upon how much auditing you perform and may be far greater than this figure per machine.

2 Assign the audit file systems to the dedicated partitions. Each disk full machine should have a backup audit directory on the local machine in case its NFS-mounted audit file system or file systems are not available.

3 While each machine is in single-user mode, run `tunefs -m 0` on each dedicated audit partition to reduce reserved file system space to 0%.

A reserved space percentage (called the `minfree` limit) is specified for audit partitions in the `audit_control` file. The default is 20%, and this percentage is tunable. Because this value is set by each site in the `audit_control` file, you should remove the automatically reserved file system space that is set aside by default for all file systems.

4 Set the required permissions on each of the audit directories on the audit server and make a subdirectory in each audit directory called **files**. Use `chown` and `chmod` to assign the required permissions to each audit directory and to each `files` subdirectory.

5 If using audit servers, export the audit directories using the `dfstab(4)` file.

6 Create the `audit_control` file entries for all the audit directories in the `audit_control` file on each machine, specifying the `files` subdirectory.

7 On each audit client, create the entries for the audit file systems in the `vfstab(4)` files.

8 On each audit client, create the mount point directories and use `chmod` and `chown` to set the correct permissions.

The following table summarizes the commands to use to configure auditing.

Utility	Task
<code>allocate(1M)</code>	Allocate a device
<code>audit(1M)</code>	Control the audit daemon
<code>audit_startup(1M)</code>	Initialize the audit subsystem
<code>audit_warn(1M)</code>	Run the audit daemon warning script
<code>auditconfig(1M)</code>	Configure auditing
<code>auditd(1M)</code>	Control audit trail files
<code>auditreduce(1M)</code>	Merge and select audit records from audit trail files
<code>auditstat(1M)</code>	Display kernel audit statistics
<code>bsmconv(1M)</code>	Enable a Solaris system to use the Basic Security Module
<code>bsmunconv(1M)</code>	Disable the Basic Security Module and return to Solaris
<code>deallocate(1M)</code>	Deallocate a device

Utility	Task
auditon(2)	Manipulate auditing
auditsvc(2)	Write audit log to specified file descriptor
sudo(1M)	Generate audit log files

Basic Configuration Steps

Basic Configuration

1. Enable BSM and ensure that auditd is started at boot time.
 - a. Run `/etc/security/bsmconv` as root to enable auditing. See [Enabling BSM Auditing](#) for more detailed information.
 - b. Set up the `/etc/security/audit_control` file to indicate the type of auditing to be performed. See [Audit Control](#) for more information.
 - c. Reboot the system so the `c2audit` module is properly loaded and the internal audit settings are configured.
2. Set up the classes of events for which you want to generate audit records and where those records are to go. These are defined in `/etc/security/audit_control`. See [Setting up Class and Events](#) for more information

For example, to record the login events for all users, add the class `lo` to the `flags:` line of `/etc/security/audit_control`. The `dir:` line specifies the directory into which audit records are to be written. This is the directory name you must enter for the **praudit Output File** parameter during SmartConnector installation.



The default path for `praudit` is `/usr/sbin`. If you use another path for `praudit`, make sure that you add the location to the system `PATH` variable.

```
dir: /var/audit
flags: lo
minfree: 20
naflags: lo
```

`flags: lo` logs events regardless of whether the event was a success or a failure.

To log only failures, add a hyphen (-) in front of the class name.

Enabling BSM Auditing

Select one of the following procedures depending on your Operating System:

Enabling BSM Auditing in Solaris 10



Note: Enabling BSM on a server automatically enables the BSM feature on all of that server's clients.

1. Log in as root.
2. Bring the system into the single-user mode:

```
# /etc/telinit 1
```

3. Change to the /etc/security directory and execute the bsmconv script

```
# cd /etc/security  
# ./bsmconv
```

The script sets up a standard Solaris machine to run BSM after a reboot.

4. After the script finishes, halt the system with the telinit command.

```
# /etc/telinit 6
```

5. Reboot the system to bring it up as a multi-user BSM system.



Note: Reboot the server to restart the BSM service.

Enabling BSM Auditing in Solaris 11

Auditing is enabled by default on Solaris 11, but only user login/logout events are monitored by default. To monitor both the OS File change events and OS USER logins/logout events, you can execute the following command with root privilege:

```
# /usr/sbin/auditconfig -setflags fw,fd,fc,fm,fr,lo
```



The bsmconv command has been removed on Solaris 11. Use the audit -s command to enable the auditing feature, when required.

The bsmconv script creates a number of files in the /etc/security directory, including:

Setting Up Classes and Events

bsmconv creates a number of files in the /etc/security directory, including:

- The `audit_startup` script is invoked at boot time and sets a number of different audit policies for the system.
- The `audit_control` file is the primary configuration file for BSM.
- The `audit_class` and `audit_event` files can be used when more fine-grained control of the audit configuration is required.

The following sections describe the `audit_startup` and `audit_control` files, audit classes and events, and custom audit classes you might access when setting up auditing.

Audit Startup

The existence of a file with the path name `/etc/security/audit_startup` causes the audit daemon to be run automatically when the system enters multi-user mode. A default `audit_startup` script that automatically configures the event to class mappings and sets the audit policies is set up during the BSM package installation.

The `audit_startup` script is a series of `auditconfig` commands to initialize the system auditing policy:

```
#!/bin/sh
/usr/sbin/auditconfig -conf
/usr/sbin/auditconfig -aconf
/usr/sbin/auditconfig -setpolicy none
/usr/sbin/auditconfig -setpolicy +cnt
/usr/sbin/auditconfig -setpolicy +argv,arge
```

The first two lines pull configuration information out of the `audit_control` file and set up the basic events the system audits. The remaining lines set other special auditing policy options:

`-setpolicy none`

Clears audit policy so that the system to starts fresh.

`setpolicy +cnt`

Indicates the system to continue running even if the auditing partition on the machine fills up. High security sites are required to have the machine shut down if auditing becomes impossible.

`-setpolicy -cnt` and `-setpolicy +argv,arge`

Tracks the full command line and all environment settings for any command executed on the system. Note that the `-setpoloicy +argv,arge` line is not part of the default BSM configuration set up by the `bsmconv` script.

Audit Control

An `audit_control` file looks similar to the following:

```
dir:/var/audit
minfree:20
flags:lo,ad,pc,fm,fw,-fc,-fd,-fr
naflags:lo,ad,ex
```

`dir`

is the directory into which audit logs will be written on the . This directory must be accessible only by the superuser. You must specify this directory name required during SmartConnector installation. There is no built-in facility to write audit logs to some other system, although some sites have attempted writing to an NFS-mounted directory from some central file server. This configuration requires the client system to have root write privileges into the NFS volume, which has some significant security implications.

`minfree`

Specifies the amount of free space, as a percentage, that must exist in the auditing partition. For example, if `minfree` is set at 20, and the audit partition goes above 80% full, the auditing subsystem starts sending warning messages to the administrator

`flags` and `naflags`

Define to which audit events the system actually is going to pay attention. The `auditconfig -conf` and `auditconfig -aconf` commands in `audit_startup` looks for these flags. The two letter codes are groups or audit classes of related events or system calls defined through the `audit_class` and `audit_events` files.

The `flags` line defines the audit vector for normal user sessions on the machine. The `naflags` line catches all events that are not associated with a particular user's session. Usually, these events are the result of system processes and do not occur often.

Audit Class and Audit Event

In tuning BSM auditing, you should strike a balance between getting the events you need to reconstruct what has been happening on the system while filtering out uninteresting events that add "noise" to the audit trail and consume huge amounts of disk space.

A recommended minimum set of classes is `lo`, `ad`, and `na`. These include login/out events (`lo`), admin events (`ad`), such as file system mounts and creation of users, and non-attributable (`na`) events.

General Events - lo (login) and ad (administrative)

The lo (login) class covers all forms of system logins as well as use of the su command.

The ad (administrative) class covers a wide variety of administrative actions, including rebooting the system, adding and deleting users, changing auditing and logging parameters, mounting and dismounting both local and remote file systems, changing quotas, loading kernel modules, and even setting the system clock.

Process Events - ex (execution) and pc (process control)

The ex and pc classes deal with process execution on the system.

The two events in the ex class (exec () and execve () system calls) are used to execute programs on the system. These events also are contained in the pc class, so if your audit vector includes pc, you need not worry about ex.

The pc class also tracks everything that the process might do during its lifetime, such as changing directories, calling setuid() and setgid() to change its privilege level, making chroot() calls, creating child processes with fork() and vfork(), and so on. The pc class also tracks administrative interaction with processes on the system, such as kill and nice.

pc tracks various system calls that usually are not interesting. For example, you probably do not need to know every time your mail server forks a new child process to deal with an incoming connection. What you really want to know is when new processes get started on the system, typically with a fork() followed by an exec(). So you really want to track just the exec()s. To track the important events from the pc class but ignore the uninteresting ones requires creating a new custom class that includes just the events you want (see "Custom Audit Classes").

File Attribute Modification - fm Class

The fm class tracks changes to file attributes such as ownership (chown) and permissions (chmod), and even extended file ACL settings. However, fm also tracks file locking and updating timestamps on files. These latter events are too frequent on normal UNIX systems to be useful. To track the important events from the fm class but ignore the uninteresting ones requires creating a new custom class that includes just the events you want (see "Custom Audit Classes").

Other File Actions - fc (create), fr (read), fw (write), fd (delete)

Oracle recommends avoiding these audit classes in order to reduce the size of the audit trail. However, the DoD guidelines require tracking at least failure for these classes (the specific recommendation is fw, -fc, -fd, -fr). These classes really can generate an enormous number of audit events and consume huge amounts of disk space. The default recommendation is to not turn on any auditing of these classes.

Custom Audit Classes

Audit class names are defined in the `audit_class` file. The following are the audit class entries for the classes discussed thus far:

```
0x00000001:fr:file read
0x00000002:fw:file write
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete
0x00000080:pc:process
0x00000800:ad:administrative
0x00010000:lo:login or logout
0x40000000:ex:exec
0xffffffff:all:all classes
```

The first field of each line is a unique bit mask used to represent the audit class in the internals of the auditing subsystem. The second field is the class code used in the `flags` and `naflags` lines in `audit_control`, and the third field is a brief descriptive name for the use of the system administrator.

When creating a custom class, pick a bit mask and a two-letter code that are not currently in use by any other class. In the default `audit_class` file installed by `bsmconv`, bit masks from `0x00010000` through `0x08000000` are not used. The following example uses `cc` (custom class) with a bit mask of `0x08000000`:

```
0x08000000:cc:CIS custom class
```

Audit Log File Rotation

Audit logs are written to binary files in your audit directory. The file naming convention used is `<start>.<end>.<hostname>`, where `<start>` and `<end>` are time/date stamps in the format `YYYYMMDDhhmmss` and `<hostname>` is the fully-qualified hostname of the local machine. The current audit log that is actively being written is named `<start>.not_`

terminated.<hostname> to distinguish it from the other audit logs in the directory.

The command `audit-n` signals the system audit daemon to close its current audit log file and start a new one. Unless told otherwise, the audit daemon will simply continue writing to the current audit log and it will grow without bound until it reaches the file size limit for the machine or fills the partition. To force audit logs to be restarted at the top of every hour:

```
0 8 8 8 8 /usr/sbin/audit -n
```

Once the new audit log has been started, the old log can be compressed or moved off of the local system for archival.

BSM Caveats

- If the connector's event log file tailing process executes more slowly than the BSM's event log generation process, and if a newly rotated log file is detected while the current log file is still being tailed, the current tail process is terminated without having a chance to complete reading the current log file or the subsequent log files. The connector starts tailing the newly rotated log file, leading to an event loss.
- Enabling BSM automatically disables the <Stop>-A keyboard sequence on the machine. This occurs to be able to monitor shutdown and reboot events and associate them with a particular user. Disabling <Stop>-A means somebody has to log in, become root, and halt the machine, all of which are auditable events.
- Enabling BSM disables auto-mounting of CD-ROMs and floppies using `vol`. Again, there is an audit trail issue if a system process spontaneously mounts and dismounts file systems.
- There are known interoperability problems between OpenSSH (particularly with PrivSep enabled) and BSM. The most noticeable issue is that OpenSSH sessions will not appear in the audit logs at all. A patch[4] is available to fix this and some other issues.

Installing the SmartConnector

The following sections provide instructions for installing and configuring your selected SmartConnector.

Preparing to Install the SmartConnector

Before you install any SmartConnectors, make sure that the OpenText ArcSight products with which the connectors will communicate have already been installed correctly (such as ArcSight ESM or ArcSight Logger).

For complete product information, refer to the *Administrator's Guide to ArcSight Platform*, available on [ArcSight Documentation](#).

If you are adding a connector to the ArcSight Management Center, see the *ArcSight Management Center Administrator's Guide* available on [ArcSight Documentation](#) for instructions.

Installing and Configuring the SmartConnector

1. Start the installation wizard.
2. Follow the instructions in the wizard to install the core software.
3. Specify the relevant [Global Parameters](#), when prompted.
4. Select **Oracle Solaris Basic Security Module** from the **Type** drop-down, then click **Next**.
5. Specify the following parameters to configure the SmartConnector, then click **Next**.

Parameter	Description
Solaris Version	Select your Solaris version: 10.x or 11.x
Log Directory	Enter the absolute path to the directory containing the log files. The default value is /var/audit.

6. Select a [destination and configure parameters](#).
7. Specify a name for the connector.
8. (Conditional) If you have selected **ArcSight Manager** as the destination, the certificate import window for the ArcSight Manager is displayed. Select **Import the certificate to the connector from destination**, and then click **Next**. The certificate is

imported and the **Add connector Summary** window is displayed.



Note: If you select Do not import the certificate to connector from destination, the connector installation will end.

9. Select whether you want to install the connector as a service or in the standalone mode.
10. Complete the installation.
11. [Run the SmartConnector.](#)

For instructions about upgrading the connector or modifying parameters, see [Installation and User Guide for SmartConnector.](#)

Device Event Mapping to ArcSight Fields

The following section lists the mappings of ArcSight data fields to the device's specific event definitions. See the *ArcSight Console User's Guide* for more information about the ArcSight data fields.

Oracle Solaris 10 and 11 BSM Common Mappings to ArcSight ESM Fields

ArcSight ESM Field	Device-Specific Field
Destination Host Name	Host
Destination User Name	subject-audit-uid
Device Action	return-errval
Device Custom Number 1	subject-sid
Device Custom Number 1 Label	'Session ID'
Device Custom String 2	exec_args
Device Custom String 2 Label	'exec_args'
Device Custom String 3	subject-tid-host
Device Custom String 3 Label	'Terminal Host'
Device Custom String 4	One of (return-retval, return-errval-reason)
Device Custom String 4 Label	'Reason or Error Code'
Device Custom String 5	subject-rgid or subject-gid
Device Custom String 5 Label	'Source User Group'
Device Custom String 6	subject-rgid
Device Custom String 6 Label	'Destination User Group'
Device Event Class ID	Event
Device Host Name	Host
Device Process Name	'auditid'
Device Product	'BSM'

ArcSight ESM Field	Device-Specific Field
Device Receipt Time	DateTime
Device Vendor	'Oracle'
Device Version	_DEVICE_VERSION
External ID	subject-pid
File Name	Path
Message	text
Name	Event

Event Type AUE_su

ArcSight ESM Field	Device-Specific Field
Destination User Name	Text
Device Custom String 4	Text
Device Custom String 5	subject-rgid
Device Custom String 6	NA
Source Host Name	subject-tid-host
Source User Name	subject-ruid

Event Type AUE_rexecd

ArcSight ESM Field	Device-Specific Field
Source Host Name	Text

Event Type AUE_passwd

ArcSight ESM Field	Device-Specific Field
Destination User Name	One of (subject-audit-uid,text)
Device Custom String 6	NA
Source Host Name	host
Source User Name	subject-audit-uid

Event Type AUE_rexd

ArcSight ESM Field	Device-Specific Field
Source Host Name	text

Event Type AUE_ftp_access

ArcSight ESM Field	Device-Specific Field
Device Custom String 4	text
Source Host Name	subject-tid-host

Event Type AUE_login-ssh

ArcSight ESM Field	Device-Specific Field
Source Host Name	subject-tid-host

Event Type AUE_role_login

ArcSight ESM Field	Device-Specific Field
Destination User Name	subject-ruid
Device Custom String 5	subject-gid
Device Custom String 6	subject-rgid
Source Host Name	subject-tid-host
Source User Name	subject-audit-uid

Event Type AUE_newgrp_login

ArcSight ESM Field	Device-Specific Field
Destination User Name	subject-ruid
Device Custom String 5	subject-rgid

ArcSight ESM Field	Device-Specific Field
Device Custom String 6	text
Source Host Name	host
Source User Name	subject-ruid

Event Type AUE_zlogin

ArcSight ESM Field	Device-Specific Field
Device Custom String 1	Zone

Event Type AUE_sudo

ArcSight ESM Field	Device-Specific Field
Device Custom String 5	subject-gid
Source User Name	subject-audit-uid

Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Configuration Guide for Oracle Solaris Basic Security Module SmartConnector (SmartConnectors 8.4.3)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to MFI-Documentation-Feedback@opentext.com.

We appreciate your feedback!