Micro Focus Security ArcSight Connectors

SmartConnector for Oracle Solaris Basic Security Module Syslog

Configuration Guide

June, 2018

Configuration Guide

SmartConnector for Oracle Solaris Basic Security Module Syslog

June, 2018

Warranty

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

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated. Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. UNIX® is a registered trademark of The Open Group.

Revision History

Date	Description
10/17/2017	Added encryption parameters to Global Parameters.
11/30/2016	Updated installation procedure for setting preferred IP address mode.
05/15/2015	Added new parameters for Syslog File.
02/16/2015	Added parameter for Syslog Daemon connector configuration.
03/31/2014	Added support for sudo for Solaris SPARC and x86.
02/14/2014	Added GA support for Solaris SPARC 11.
11/15/2013	Added GA support for Solaris 11 x86. Modified Product Version and Device Severity mappings.

SmartConnector for Oracle Solaris Basic Security Module Syslog

This guide provides information for installing the SmartConnector for Oracle Solaris Basic Security Module Syslog and configuring the device for event collection. Event collection from Solaris SPARC versions 10 and 11 and Solaris 11 x86 version is supported.

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.

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 lets Solaris 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 location 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. See "Basic Configuration" later in this guide.

Configuration

BSM Auditing

For complete information about BSM auditing, see the *SunSHIELD 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).

For complete information about the commands mentioned in this section, see Sun Microsystems *man pages section 1M: System Administration Commands*.

The **audit_syslog** plugin module for Solaris Audit, `/sur/lib/security/audit_syslog.so`, provides realtime conversion of Solaris audit data to syslog-formatted (text) data and sends it to a syslog daemon as configured in the `syslog.conf` configuration file. See the **syslog.conf(4)** man page for more information. The plugin's path is specified in the audit configuration file, **audit_control(4)**.

Messages to syslog are written when selected using the plugin option in **audit_control**. Syslog messages are generated with the facility code of LOG_AUDIT (audit in **syslog.conf(4)**) and severity of LOG_NOTICE. Audit syslog messages contain data selected from the tokens described for the binary audit log. See the **audit.log(4)** man page for more information.

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
allocate(1M)	Allocate a device
audit(1M)	Control the audit daemon
audit_startup(1M)	Initialize the audit subsystem
audit_warn(1M)	Run the audit daemon warning script
auditconfig(1M)	Configure auditing
auditd(1M)	Control audit trail files
auditreduce(1M)	Merge and select audit records from audit trail files
auditstat(1M)	Display kernel audit statistics

Utility	Task
bsmconv(1M)	Enable a Solaris system to use the Basic Security Module
bsmunconv(1M)	Disable the Basic Security Module and return to Solaris
deallocate(1M)	Deallocate a device
auditon(2)	Manipulate auditing
auditsvc(2)	Write audit log to specified file descriptor
sudo(1M)	Generate audit log files (/var/audit)

Basic Configuration Steps

1 Enable BSM and ensure `auditd` is started at boot time.

   A Run `/etc/security/bsmconv` (as `root`) to enable auditing. Auditing is not enabled by default. See "Enabling BSM" 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 "Audit Class and Audit Event" 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 should 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`, be sure to add the location to the system PATH variable.

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

Including `lo` on the flags: line logs events regardless of whether it was a success or failure; to log only failures, put a hyphen (-) in front of the class name.

Enable BSM Auditing in Solaris 10

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

1 After becoming `root`, bring the system into the single-user mode:

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

2 In single-user mode, change directories to the `/etc/security` directory and execute the `bsmconv` script located there. The script sets up a standard Solaris machine to run BSM after a reboot.

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

3 After the script finishes, halt the system with the `telinit` command. Then reboot the system to bring it up as a multi-user BSM system.

```
# /etc/telinnit 6
```

Whenever you need to restart the BSM service, restart through a server reboot.

Enable BSM Auditing in Solaris 11

Auditing is enabled by default on Solaris 11, but only user login/logout events are monitored by default. For monitoring 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 following command to enable the auditing feature, if needed: `audit -s`

Set 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 may 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 for initializing 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 will audit. The remaining lines set other special auditing policy options:

`-setpolicy none`
: Blanks the audit policy for the system to start with a clean slate

`setpolicy +cnt`
: Tells 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`
: Means to track 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

The `audit_control` file appears simple:

```
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 system (this directory should only be accessible by the superuser). (Note that this is the directory name required during SmartConnector installation.) There is no built-in facility for writing audit logs to some other system, although some sites have attempted writing to an NFS-mounted directory from some central file server (note that 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; otherwise the system starts complaining. So, with `minfree` set at 20, once the audit partition goes above 80% full, the auditing subsystem starts sending the administrator warning messages.

`flags and naflags`
: Define to which audit events the system actually is going to pay attention (these are the lines at which the `auditconfig -conf` and `auditconfig -aconf` commands in `audit_startup` are looking). The two letter codes are groups (audit classes) of related events (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 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

- 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 vold. 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.

Configure the Syslog SmartConnectors

The three ArcSight Syslog SmartConnectors are:

Syslog Daemon
Syslog Pipe
Syslog File

The Syslog Daemon SmartConnector

The Syslog Daemon SmartConnector is a syslogd-compatible daemon designed to work in operating systems that have no syslog daemon in their default configuration, such as Microsoft Windows. The SmartConnector for Syslog Daemon implements a UDP receiver on port 514 (configurable) by default that can be used to receive syslog events. Use of the TCP protocol or a different port can be configured manually.

If you are using the SmartConnector for Syslog Daemon, simply start the connector, either as a service or as a process, to start receiving events; no further configuration is needed.

Messages longer than 1024 bytes may be split into multiple messages on syslog daemon; no such restriction exists on syslog file or pipe.

The Syslog Pipe and File SmartConnectors

When a syslog daemon is already in place and configured to receive syslog messages, an extra line in the syslog configuration file (`rsyslog.conf`) can be added to write the events to either a ***file*** or a system ***pipe*** and the ArcSight SmartConnector can be configured to read the events from it. **In this scenario, the ArcSight SmartConnector runs on the same machine as the syslog daemon.**

The **Syslog Pipe** SmartConnector is designed to work with an existing syslog daemon. This SmartConnector is especially useful when storage is a factor. In this case, syslogd is configured to write to a named pipe, and the Syslog Pipe SmartConnector reads from it to receive events.

The **Syslog File** SmartConnector is similar to the Pipe SmartConnector; however, this SmartConnector monitors events written to a syslog file (such as `messages.log`) rather than to a system pipe.

Configure the Syslog Pipe or File SmartConnector

This section provides information about how to set up your existing syslog infrastructure to send events to the ArcSight Syslog Pipe or File SmartConnector.

The standard UNIX implementation of a syslog daemon reads the configuration parameters from the **/etc/rsyslog.conf** file, which contains specific details about which events to write to files, write to pipes, or send to another host. First, create a pipe or a file; then modify the **/etc/rsyslog.conf** file to send events to it.

For syslog pipe:

1. Create a pipe by executing the following command:

   ```
   mkfifo /var/tmp/syspipe
   ```

2. Add the following line to your **/etc/rsyslog.conf** file:

   ```
   *.debug  /var/tmp/syspipe
   ```

   or

   ```
   *.debug  |/var/tmp/syspipe
   ```

   depending on your operating system.

3. After you have modified the file, restart the syslog daemon either by executing the scripts **/etc/init.d/syslogd stop** and **/etc/init.d/syslogd start**, or by sending a `configuration restart` signal.

   On RedHat Linux, you would execute:

   ```
   service syslog restart
   ```

   On Solaris, you would execute:

   ```
   kill -HUP `cat /var/run/syslog.pid´
   ```

This command forces the syslog daemon to reload the configuration and start writing to the pipe you just created.

For syslog file:

Create a file or use the default for the file into which log messages are to be written.

After editing the `/etc/rsyslog.conf` file, be sure to restart the syslog daemon as described above.

When you follow the SmartConnector Installation Wizard, you will be prompted for the absolute path to the syslog file or pipe you created.

Install the SmartConnector

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

Syslog Installation

Install this SmartConnector (on the syslog server or servers identified in the *Configuration* section) using the SmartConnector Installation Wizard appropriate for your operating system. The wizard will guide you through the installation process. When prompted, select one of the following **Syslog** connectors (see *Configure the Syslog SmartConnectors* in this guide for more information):

Syslog Daemon
Syslog Pipe
Syslog File

Because all syslog SmartConnectors are sub-connectors of the main syslog SmartConnector, the name of the specific syslog SmartConnector you are installing is not required during installation.

The syslog daemon connector by default listens on port 514 (configurable) for UDP syslog events; you can configure the port number or use of the TCP protocol manually. The syslog pipe and syslog file connectors read events from a system pipe or file, respectively. Select the one that best fits your syslog infrastructure setup.

Prepare to Install Connector

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

For complete product information, read the *Administrator's Guide* as well as the *Installation and Configuration* guide for your ArcSight product before installing a new SmartConnector. If you are adding a connector to the ArcSight Management Center, see the *ArcSight Management Center Administrator's Guide* for instructions, and start the installation procedure at "Set Global Parameters (optional)" or "Select Connector and Add Parameter Information."

Before installing the SmartConnector, be sure the following are available:

- Local access to the machine where the SmartConnector is to be installed
- Administrator passwords

Install Core Software

Unless specified otherwise at the beginning of this guide, this SmartConnector can be installed on all ArcSight supported platforms; for the complete list, see the *SmartConnector Product and Platform Support* document, available from the Micro Focus SSO and Protect 724 sites.

1 Download the SmartConnector executable for your operating system from the Micro Focus SSO site.

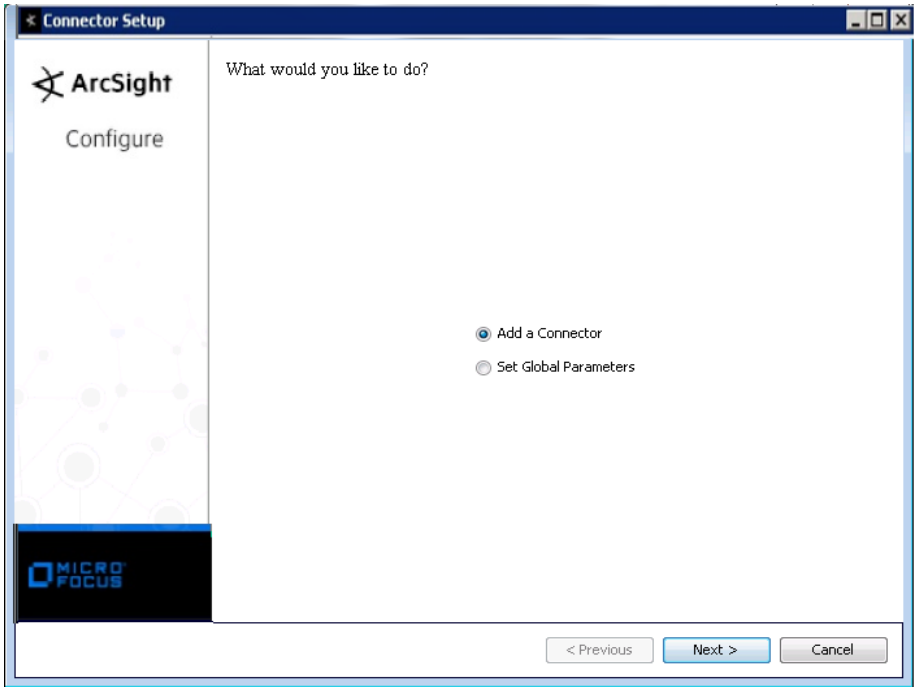
2 Start the SmartConnector installation and configuration wizard by running the executable.

> When installing a syslog daemon SmartConnector in a UNIX environment, run the executable as 'root' user.

Follow the wizard through the following folder selection tasks and installation of the core connector software:

Introduction
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...

3 When the installation of SmartConnector core component software is finished, the following window is displayed:



Set Global Parameters (optional)

If you choose to perform any of the operations shown in the following table, do so before adding your connector. You can set the following parameters:

Parameter	Setting
FIPS mode	Select 'Enabled' to enable FIPS compliant mode. To enable FIPS Suite B Mode, see the SmartConnector User Guide under "Modifying Connector Parameters" for instructions. Initially, this value is set to 'Disabled'.
Remote Management	Select 'Enabled' to enable remote management from ArcSight Management Center. When queried by the remote management device, the values you specify here for enabling remote management and the port number will be used. Initially, this value is set to 'Disabled'.
Remote Management Listener Port	The remote management device will listen to the port specified in this field. The default port number is 9001.
Preferred IP Version	When both IPv4 and IPv6 IP addresses are available for the local host (the machine on which the connector is installed), you can choose which version is preferred. Otherwise, you will see only one selection. The initial setting is IPv4.

The following parameters should be configured only if you are using Micro Focus SecureData solutions to provide encryption. See the *Micro Focus SecureData Architecture Guide* for more information.

Parameter	Setting
Format Preserving Encryption	Data leaving the connector machine to a specified destination can be encrypted by selecting ‘Enabled’ to encrypt the fields identified in ‘Event Fields to Encrypt' before forwarding events. If encryption is enabled, it cannot be disabled. Changing any of the encryption parameters again will require a fresh installation of the connector.
Format Preserving Policy URL	Enter the URL where the Micro Focus SecureData Server is installed.
Proxy Server (https)	Enter the proxy host for https connection if any proxy is enabled for this machine.
Proxy Port	Enter the proxy port for https connection if any proxy is enabled for this machine.
Format Preserving Identity	The Micro Focus SecureData client software allows client applications to protect and access data based on key names. This key name is referred to as the identity. Enter the user identity configured for Micro Focus SecureData.
Format Preserving Secret	Enter the secret configured for Micro Focus SecureData to use for encryption.
Event Fields to Encrypt	Recommended fields for encryption are listed; delete any fields you do not want encrypted and add any string or numeric fields you want encrypted. Encrypting more fields can affect performance, with 20 fields being the maximum recommended. Also, because encryption changes the value, rules or categorization could also be affected. Once encryption is enabled, the list of event fields cannot be edited.

After making your selections, click **Next**. A summary screen is displayed. Review the summary of your selections and click **Next**. Click **Continue** to return to proceed with "Add a Connector" window. Continue the installation procedure with "Select Connector and Add Parameter Information."

Select Connector and Add Parameter Information

1. Select **Add a Connector** and click **Next**. If applicable, you can enable FIPS mode and enable remote management later in the wizard after SmartConnector configuration.
2. Select **Syslog Daemon, Syslog File, or Syslog Pipe** and click **Next**.
3. Enter the required SmartConnector parameters to configure the SmartConnector, then click **Next**.

Syslog Daemon Parameters	*Network port*	The SmartConnector for Syslog Daemon listens for syslog events from this port.

	IP Address	The SmartConnector for Syslog Daemon listens for syslog events only from this IP address (accept the default (ALL) to bind to all available IP addresses).
	Protocol	The SmartConnector for Syslog Daemon uses the selected protocol (UDP or Raw TCP) to receive incoming messages.
	Forwarder	Change this parameter to 'true' only if the events being processed are coming from another SmartConnector sending to a CEF Syslog destination, and that destination also has CEF forwarder mode enabled. That allows attributes of the original connector to be retained in the original agent fields.
Syslog Pipe Parameter	*Pipe Absolute Path Name*	Absolute path to the pipe, or accept the default: /var/tmp/syspipe

Syslog File Parameters	*File Absolute Path Name*	Enter the full path name for the file from which this connector will read events or accept the default: \var\adm\messages (Solaris) or \var\log\messages (Linux).
		
		A wildcard pattern can be used in the file name; however, in realtime mode, rotation can occur only if the file is over-written or removed from the folder. Realtime processing mode assumes following external rotation.
		
		For date format log rotation, the device writes to 'filename.timestamp.log' on a daily basis. At a specified time, the device creates a new daily log and begins to write to it. The connector detects the new log and terminates the reader thread to the previous log after processing is complete. The connector then creates a new reader thread to the new 'filename.timestamp.log' and begins processing that file. To enable this log rotation, use a date format in the file name as shown in the following example:
		
		`filename'yyyy-MM-dd'.log;`
		
		For index log rotation, the device writes to indexed files - 'filename.log.001', 'filename.log.002', 'filename.log.003', and so on. At startup, the connector processes the log with highest index. When the device creates a log with a greater index, the connector terminates the reader thread to the previous log after processing completes, creates a thread to the new log, and begins processing that log. To enable this log rotation, use an index format, as shown in the following example:
		
		`filename'%d,1,99,true'.log;`
		
		Specifying 'true' indicates that it is allowed for the index to be skipped; for example, if 5 appears before 4, processing proceeds with 5 and will not read 4, even if 4 appears later. Use of 'true' is optional.

	Reading Events Real Time or Batch	Specify whether file is to be read in batch or realtime mode. For batch mode, all files are read from the beginning. The 'Action Upon Reaching EOF' and 'File Extension if Rename Action' parameters apply for batch mode only.
	Action Upon Reaching EOF	For batch mode, specify 'None', 'Rename', or 'Delete' as the action to be performed to the file when the connector has finished reading and reaches end of file (EOF). For realtime mode, leave the default value of 'None' for this parameter.
	File Extension If Rename Action	For batch mode, specify the extension to be added to the file name if the action upon EOF is 'Rename' or accept the default value of '.processed'.

Select a Destination

1 The next window asks for the destination type; select a destination and click **Next**. For information about the destinations listed, see the *ArcSight SmartConnector User Guide*.

2 Enter values for the destination. For the ArcSight Manager destination, the values you enter for **User** and **Password** should be the same ArcSight user name and password you created during the ArcSight Manager installation. Click **Next**.

3 Enter a name for the SmartConnector and provide other information identifying the connector's use in your environment. Click **Next**. The connector starts the registration process.

4 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 click **Next**. (If you select **Do not import the certificate to connector from destination**, the connector installation will end.) The certificate is imported and the **Add connector Summary** window is displayed.

Complete Installation and Configuration

1 Review the **Add Connector Summary** and click **Next**. If the summary is incorrect, click **Previous** to make changes.

2 The wizard now prompts you to choose whether you want to run the SmartConnector as a stand-alone process or as a service. If you choose to run the connector as a stand-alone process, select **Leave as a standalone application**, click **Next**, and continue with step 5.

3 If you chose to run the connector as a service, with **Install as a service** selected, click **Next**. The wizard prompts you to define service parameters. Enter values for **Service Internal Name** and **Service Display Name** and select **Yes** or **No** for **Start the service automatically**. The **Install Service Summary** window is displayed when you click **Next**.

4 Click **Next** on the summary window.

5 To complete the installation, choose **Exit** and Click **Next**.

For instructions about upgrading the connector or modifying parameters, see the *SmartConnector User Guide*.

Run the SmartConnector

SmartConnectors can be installed and run in stand-alone mode, on Windows platforms as a Windows service, or on UNIX platforms as a UNIX daemon, depending upon the platform supported. On Windows platforms, SmartConnectors also can be run using shortcuts and optional Start menu entries.

If the connector is installed in stand-alone mode, it must be started manually and is not automatically active when a host is restarted. If installed as a service or daemon, the connector runs automatically when the host is restarted. For information about connectors running as services or daemons, see the *ArcSight SmartConnector User Guide*.

To run all SmartConnectors installed in stand-alone mode on a particular host, open a command window, go to `$ARCSIGHT_HOME\current\bin` and run: `arcsight connectors`

To view the SmartConnector log, read the file `$ARCSIGHT_HOME\current\logs\agent.log`; to stop all SmartConnectors, enter `Ctrl+C` in the command window.

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 Mappings to ArcSight ESM Fields

ArcSight ESM Field	Device-Specific Field
Connector (Agent) Severity	High = alert, Medium = failed, Low = ok
Device Custom IPv6 Address 2	Source IPv6 Address
Device Custom String 1	access info
Device Custom String 2	session
Device Custom String 3	user group
Device Custom String 4	zone name
Device Event Class ID	msg
Device Process Name	auditd
Device Product	'solaris'
Device Severity	alert | failed | ok
Device Vendor	'Oracle'
Device Version	'10/11'
External ID	ID
File Name	filename
File Path	path
Message	msg
Name	Both("solaris BSM",Module)
Source Address	source ip
Source Host Name	source host name
Source User Name	source user name