Novell®
Sentinel™

www.novell.com

5.1.3

Volume III – SENTINEL WIZARD USER'S GUIDE

July 7, 2006

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to www.novell.com/info/exports/ for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.novell.com/company/legal/patents/ and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Third-Party Legal Notices

Sentinel 5 may contain the following third-party technologies:

- Apache Axis and Apache Tomcat, Copyright © 1999 to 2005, Apache Software Foundation. For more information, disclaimers and restrictions, see http://www.apache.org/licenses/
- ANTLR. For more information, disclaimers and restrictions, see http://www.antlr.org
- Boost, Copyright © 1999, Boost.org.
- Bouncy Castle, Copyright © 2000-2004, the Legion of Bouncy Castle. For more information, disclaimers and restrictions see http://www.bouncycastle.org.
- Checkpoint. Copyright © Check Point Software Technologies Ltd.
- Concurrent, utility package. Copyright © Doug Lea. Used without CopyOnWriteArrayList and ConcurrentReaderHashMap classes.
- Crypto++ Compilation. Copyright © 1995-2003, Wei Dai, incorporating the following copyrighted work: mars.cpp by Brian Gladman and Sean Woods. For more information, disclaimers and restrictions see http://www.eskimo.com/~weidai/License.txt.
- Crystal Reports Developer and Crystal Reports Server. Copyright © 2004 Business Objects Software Limited.
- DataDirect Technologies Corp. Copyright © 1991-2003.
- edpFTPj, licensed under the Lesser GNU Public License. For more information, disclaimers and restrictions see http://www.enterprisedt.com/products/edtftpj/purchase.html.
- Enhydra Shark, licensed under the Lesser General Public License available at: http://shark.objectweb.org/license.html.
- ICEsoft ICEbrowser. ICEsoft Technologies, Inc. Copyright © 2003-2004.
- ILOG, Inc. Copyright © 1999-2004.
- Installshield Universal. Copyright © 1996–2005, Macrovision Corporation and/or Macrovision Europe Ltd.
- Java 2 Platform, Standard Edition. Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see http://java.sun.com/j2se/1.4.2/j2re-1_4_2_10-license.txt.

  The Java 2 Platform may also contain the following third-party products:

  - CoolServlets © 1999
  - DES and 3xDES © 2000 by Jef Poskanzer
  - Crimson © 1999-2000 The Apache Software Foundation
  - Xalan J2 © 1999-2000 The Apache Software Foundation
  - NSIS 1.0j © 1999-2000 Nullsoft, Inc.
  - Eastman Kodak Company © 1992

- Lucinda, a registered trademark or trademark of Bigelow and Holmes
- Taligent, Inc.
- IBM, some portions available at: http://oss.software.ibm.com/icu4j/

For more information regarding these third-party technologies and their associated disclaimers and restrictions, see: http://java.sun.com/j2se/1.4.2/j2se-1_4_2-thirdpartylicensereadme.txt.

- JavaBeans Activation Framework (JAF). Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see http://www.java.sun.com/products/javabeans/glasgow/jaf.html and click download > license.
- JavaMail. Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see http://www.java.sun.com/products/javamail/downloads/index.html and click download > license.
- Java Ace, by Douglas C. Schmidt and his research group at Washington University and Tao (with ACE wrappers) by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine and Vanderbilt University. Copyright © 1993-2005. For more information, disclaimers and restrictions see http://www.cs.wustl.edu/~schmidt/ACE-copying.html and http://www.cs.wustl.edu/~pjain/java/ace/JACE-copying.html
- Java Authentication and Authorization Service Modules, licensed under the Lesser General Public License. For more information, disclaimers and restrictions see http://free.tagish.net/jaas/index.jsp.
- Java Network Launching Protocol (JNLP). Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions, please see http://www.java.sun.com/products/javawebstart/download-jnlp.html and click download > license.
- Java Service Wrapper. Portions copyrighted as follows: Copyright © 1999, 2004 Tanuki Software and Copyright © 2001 Silver Egg Technology. For more information, disclaimers and restrictions, see http://wrapper.tanukisoftware.org/doc/english/license.html.
- JIDE. Copyright © 2002 to 2005, JIDE Software, Inc.
- jTDS is licensed under the Lesser GNU Public License. For more information, disclaimers and restrictions see http://jtds.sourceforge.net/.
- MDateSelector. Copyright © 2005, Martin Newstead, licensed under the Lesser General Public License. For more information, disclaimers and restrictions see http://web.ukonline.co.uk/mseries.
- Monarch Charts. Copyright © 2005, Singleton Labs.
- Net-SNMP. Portions of the code are copyrighted by various entities, which reserve all rights. Copyright © 1989, 1991, 1992 by Carnegie Mellon University; Copyright © 1996, 1998 to 2000, the Regents of the University of California; Copyright © 2001 to 2003 Networks Associates Technology, Inc.; Copyright © 2001 to 2003, Cambridge Broadband, Ltd.; Copyright © 2003 Sun Microsystems, Inc. and Copyright © 2003 to 2004, Sparta, Inc. For more information, disclaimers and restrictions, see http://net-snmp.sourceforge.net.
- The OpenSSL Project. Copyright © 1998-2004. the Open SSL Project. For more information, disclaimers and restrictions, see http://www.openssl.org.
- Oracle Help for Java. Copyright © 1994-2006, Oracle Corporation.
- RoboHELP Office. Copyright © Adobe Systems Incorporated, formerly Macromedia.
- Skin Look and Feel (SkinLF). Copyright © 2000-2006 L2FProd.com. Licensed under the Apache Software License. For more information, disclaimers and restrictions see https://skinlf.dev.java.net/.
- Sonic Software Corporation. Copyright © 2003-2004. The SSC software contains security software licensed from RSA Security, Inc.
- Tinyxml. For more information, disclaimers and restrictions see http://grinninglizard.com/tinyxmldocs/index.html.
- SecurityNexus. Copyright © 2003 to 2006. SecurityNexus, LLC. All rights reserved.

NOTE: As of the publication of this documentation, the above links were active. In the event you find that any of the above links are broken or the linked webpages are inactive, please contact Novell, Inc., 404 Wyman Street, Suite 500, Waltham, MA 02451 U.S.A.

Preface

The Sentinel Technical documentation is general-purpose operation and reference guide. This documentation is intended for Information Security Professionals. The text in this documentation is designed to serve as a source of reference about Sentinel's Enterprise Security Management System. There is additional documentation available on the Sentinel web site.

Sentinel Technical documentation is broken down into five different volumes. They are:

- Volume I – Sentinel™ 5 Install Guide
- Volume II – Sentinel™ 5 User's Guide
- Volume III – Sentinel™ 5 Wizard User's Guide
- Volume IV – Sentinel™ 5 User's Reference Guide
- Volume V – Sentinel™ 3rd Party Integration

Volume I – Sentinel Install Guide

This guide explains how to install:

- Sentinel Server
- Sentinel Console
- Sentinel Correlation Engine
- Sentinel Crystal Reports
- Wizard Collector Builder
- Wizard Collector Manager
- Advisor

Volume II – Sentinel User's Guide

This guide discusses:

- Sentinel Console Operation
- Sentinel Features
- Sentinel Architecture
- Sentinel Communication
- Shutdown/Startup of Sentinel
- Vulnerability assessment
- Event monitoring
- Event filtering
- Event correlation
- Sentinel Data Manager
- Event Configuration for Business Relevance
- Mapping Service
- Historical reporting
- Wizard Host Management
- Incidents
- Cases
- User management
- Workflow

Volume III – Wizard User's Guide

This guide discusses:

- Wizard Collector Builder Operation
- Wizard Collector Manager
- Collectors
- Wizard Host Management
- Building and maintaining Collectors

Volume IV - Sentinel User's Reference Guide

This guide discusses:

- Wizard scripting language
- Wizard parsing commands
- Wizard administrator functions
- Wizard and Sentinel meta-tags
- Sentinel correlation engine
- User Permissions
- Correlation command line options
- Sentinel database schema

Volume V - Sentinel 3rd Party Integration Guide

- Remedy
- HP OpenView Operations
- HP Service Desk

Contents

1 Wizard Introduction

NOTE: The term Agent is interchangeable with Collector. Going forward, Agents will be referred to as Collectors.

The Wizard User's Guide is your introduction to the operation of Novell Wizard. This guide will explain each component and how all the components operate.

This guide assumes that you are familiar with Network Security, Database Administration, Windows and UNIX operating systems.

Contents

This guide contains the following chapters:

- Chapter 1 – Wizard Introduction
- Chapter 2 – Managing Wizard Hosts
- Chapter 3 – Building and Maintaining Collectors
- Appendix A – Syslog Connector
- Appendix B – Socket Server

Conventions Used

Notes and Cautions

NOTE: Notes provide additional information that may be useful.

CAUTION: Cautions provide additional information that may keep you from performing damage or loss of data to your system.

Commands

Commands appear in courier font. For example:

```
useradd –g dba –d /export/home/oracle –m –s /bin/csh
   oracle
```

Wizard

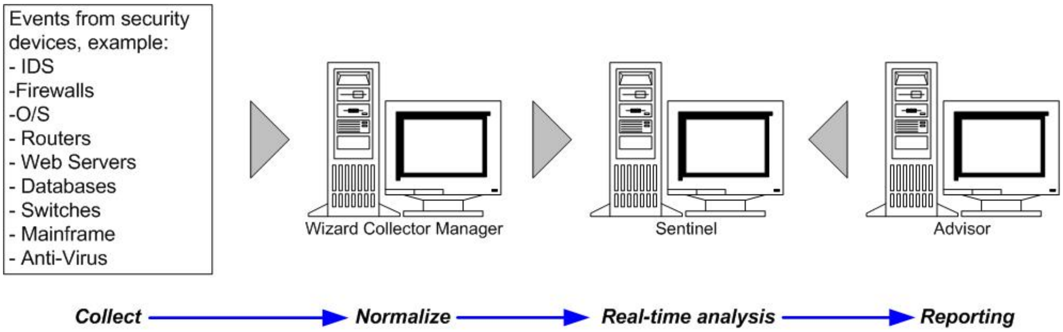
Wizard enables you to build, configure and control Collectors. Collectors are used to collect and normalize events from security devices and programs. These normalized events are then sent to Sentinel for use in real time analysis correlation, reporting and incident response.

NOTE: Although not a requirement, it is recommended that in a multiple Wizard Collector Builder configuration that one Collector Builder be designated as the Primary Collector Builder. This machine is then used to store, develop or modify Collectors and configure ports.

The following components comprise Wizard:

- Collector Builder is the Wizard user interface that enables you to build, configure, deploy and control Collectors. In addition to running Collectors locally, the Collector Builder can be used to upload, download and control Collectors on remote systems.
- Collector Manager is the Wizard back-end that manages Collectors, system status messages and performs global filtering of events.

A Collector is the receptor that collects and normalizes raw events from security devices and programs and outputs normalized events that can be correlated, reported and used for incident response. Your Sentinel software comes with tier 1 level Collectors.



Collectors

Collectors are used to filter and standardize critical event data into a normalized format and make it available to the Sentinel process. There are three levels of Collectors, they are:

- Supported Collectors (T1) – these are Collectors that are:
    - documented
    - have meta-data
    - available to all customers
    - Technical Support
- Documented Collectors (T2) – these are Collectors that are:
    - destined for the Collector library
    - documented
    - have metadata
    - based on the standard Sentinel templates
    - limited Technical Support
- Sample Collectors (T3) – these are Collectors that:
    - have proof of concept
    - developed for a specific customer
    - may not have meta-data or any supported documentation
    - limited Technical Support

Collectors allow you to gain access to event data from many sources, including:

- Intrusion Detection Systems (host)
- Intrusion Detection Systems (network)
- Firewalls
- Operating Systems
- Policy Monitoring
- Authentication
- Anti-Virus
- Web Servers
- Databases
- Mainframe
- Vulnerability Assessment
- Directory Services

- Routers & Switches
- VPN
- Network Management
- Proprietary Systems

Collectors are made of:

- Template files
- Parameter files
- Lookup files
- Mapping files
- Parameter Description File and Manifest Files

The template file and its associated parameter file are merged into different script files when the Collector Script is built.

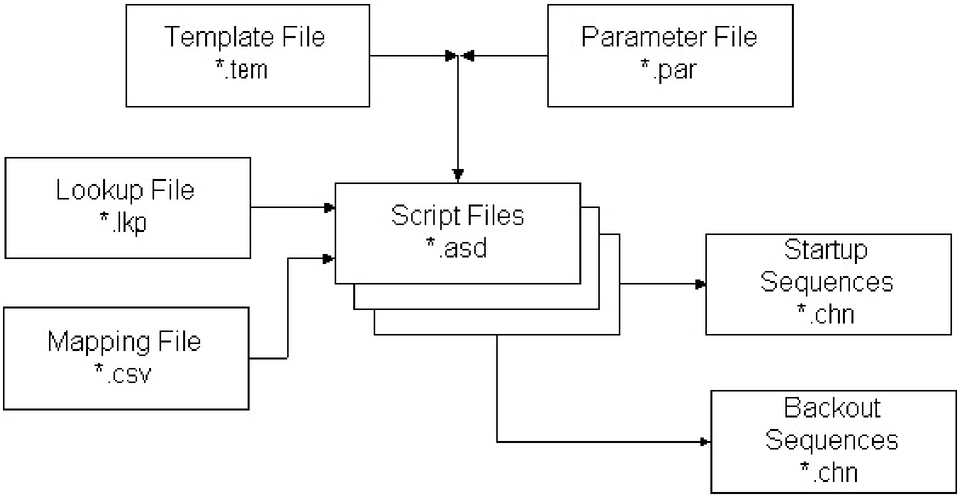
Each script file is named after the column name of the set of values in the parameter file. Script files are grouped in an ordered sequence into startup and backout sequences.

Startup and backout sequences are assigned to a port, which executes the series of scripts that it contains when it is started or stopped. A script must be included in a startup or backout sequence in order to be used by a port. Ports enable a Collector to locate Wizard hosts on the network by providing the IP address or file name of the host. They also provide Sentinel with information on the location of sensors and the Collector that is used to manage data for those sensors. The following options are configurable for ports:

- Connection type
    - Serial – data read from a RS-232C serial port
    - Socket - a TCP socket connection
    - File New - reads only security event data that is added to a file after the script has started (reads from the end of the file)
    - File All - reads all the security event data in a file
    - Persistent Process - launches a persistent process when the port is started, communicates between the assigned Collector to that port and an external application through receive and transmit states and continues to run for the active life of the port.
    - Transient Process - communicates between the assigned Collector to that port and an external application through receive and transmit states. Transient process may be started multiple times.
    - SNMP - receives SNMP v1, v2 and v3 traps
    - None
- Collector name – you can rename, copy and add Collectors

When a template uses the LOOKUP() parsing command, the appropriate lookup file is searched for a specific block of parsing commands to run.

When a template uses the TRANSLATE parsing command, the command loads a mapping file allowing for fast lookup of key entries.



Template Files

You can create, add states to, edit and delete templates. Templates determine how records will be processed. Most of the decisions about templates revolve around what types of records you are working with and their format. There is an equivalent template file with a .tem extension. They located at %WORKBENCH_HOME%\elements\<Collector Name>.

Template files are based on states. A state is a decision point within the logical flow or path of a template. Each point (state) contains information indicating the process to perform. States include reference to parameters, when the template is merged with a parameter file specific values replace the parameters. When the parameters are replaced by specific values, one or more script files are created.

As a state is inserted into a template, it is assigned a number that remains with it no matter where it is moved in the template. There are three groupings of states.

- The Transmit, Receive, Decide and Parse states are numbered in the order that they are inserted in the template.
  - Transmit state (Tx) – transmits a string to a defined port
  - Receive state (Rx) - defines whether or not Wizard receives information from a security application into a buffer. Information is taken from the port definition.
  - Decide state - uses a string of data or variable to determine what state to move to next
  - Parse state - uses the parsing commands to create templates to process the information collected in the receive buffer
- The Next and Go To states are identified by the number of the state to which they are pointing.
  - Next state - indicates which state to jump to in the next script
  - Go state - used to move back to another state within the current script
- The Stop state is always number zero. Indicates when to stop the processing on a port.

Transmit State

The Transmit State sends either a string or variable (depending on what type of data is selected) to the connection type configured for that Collector. If the connection is broken when entering the transmit state and a value is entered in Rx/Tx Value box on the template's

Port Information panel, the following event occurs, attempts are made to re-establish the connection until a successful reconnection is achieved.

There is an inter-character delay that specifies the number of milliseconds (ms) between sending each byte of data.

Receive State

The Receive State specifies the method Wizard uses to determine when data has been received from the Collector. In the Receive State, you specify:

- Receive Type
- Minimum Bytes
- Delimiter Decide String

If the connection is broken when entering the transmit state and a value is entered in Rx/Tx Value box on the template's Port Information panel, the following event occurs, attempts are made to re-establish the connection until a successful reconnection is achieved.

After the Receive State of the RxBuffer, two variables are automatically populated with the results of the Receive State:

- s_RXBufferString contains the text received by the RxBuffer
- i_RXBufferLength contains the length of the s_RXBufferString

This is equivalent to executing the following script code after a Receive State:

- COPY(s_RXBufferString:)
- LENGTH(i_RXBufferLength,s_RXBufferString)

These automatically populated variables allow for easy comparison in a Decide State of whether or not the Receive State timed out (i_RXBufferLength = 0). They also allow for the direct use of the RXbuffer through the s_RXBufferString variable.

Receive Types - There are four Receive Types available in the Template editor. They are:

- Timeout - Allows a script to continue processing even if data is not received in a specified amount of time. Selecting timeout allows Wizard to receive data until the timeout period is reached, as defined by the variable, RX_TIMEOUT_DELAY.
- Wait - Used primarily when receiving unsolicited event messages. Wizard will wait for the "timeout" duration until data is received.

NOTE: For timeout and wait receive types, processing in the script will not continue until the minimum number of bytes has been received or when the timeout is reached for the timeout option.

- delim timeout - Uses a pre-defined string of characters to indicate to Wizard that data has been received. The data in the Delimiter Decide String box is verified against the data in the receive buffer as each byte is received.
- delim wait - Used when waiting for unsolicited messages. A user-defined string of characters indicates to Wizard that data has been received. Wizard uses the data in the Delimiter Decide String box to verify the receive data as each byte is received. The parameter RX_TIMEOUT_DELAY has no effect when using the delim wait option.

NOTE: For the delim timeout and delim wait receive types, processing in the script will not continue until the delimiter decide string evaluates to true and the minimum number of bytes has been received or the timeout is reached for the delim timeout option.

Minimum Bytes - The minimum number of bytes is the number of bytes that must be received before Wizard either uses the default timeout period or continues processing. Processing in the script will not continue until the minimum number of bytes is received.

Delimiter Decide String - The Delimiter Decide String is completed when the Receive Type is delim timeout or delim wait. Collector processing will not continue to the next state until the delimiter decide string matches data read in and the minimum number of bytes has been received.

The delimiter decide string is a POSIX 1003.2 compliant regular expression.

Receive Type Scenarios - There are four types of Receive Type scenarios. They are:

- Timeout Scenario - After the Receive State is entered, processing stops until minimum bytes is read or RX_TIMEOUT_DELAY seconds passes. After Wizard receives more than the minimum number of bytes specified, or the timeout has been exceeded, the Collector port processing continues to the next state of the script.
- Wait Scenario - The Wait Receive State type waits until the Wizard Collector receives the minimum number of bytes specified in the Minimum Bytes box. After Wizard receives more that the minimum number of bytes specified in the Minimum Bytes box, the Collector port processing continues to the next state of the script. If the minimum number of bytes is not received, the Collector port processing never times out.
- Delim Timeout Scenario - If the delimiter decide string is encountered after the minimum byte position set in the Minimum Bytes box is received, the data up to and including the delimiter is stored in the Rx Buffer. If the delimiter decide string is not encountered, no data is transferred to the receive buffer and the Collector port processing times out in the default timeout period.
- Delim Wait Scenario - If the delimiter decide string is encountered after the minimum number of bytes set in the Minimum Bytes box is received, the Collector port processing continues and the data is processed. If the delimiter decide string is not encountered, no data is transferred to the receive buffer and the port does not timeout. If the delimiter decide string is never encountered, the Collector port processing never times out. In addition, if the delimiter decide string is encountered, but the minimum bytes have not been received, the Collector port processing never times out.

Decide State

The Decide State evaluates the contents of the receive buffer or variable to determine what action to take. If the information in the receive buffer contains the selected decide type, the Collector Manager processes the command as true and the Yes route is followed. If the receive buffer does not contain the selected decide type, the Collector Manager processes the decision as false and the No route is followed.

The receive buffer (size Rxbuffer) is an editable parameter located:

```
$WORKBENCH_HOME/config/wizard.properties/
   system.max_receive_buffer_size
```

This parameter allows you to configure Collector Manager’s receive buffer (Rx buffer). The default is 50,000 events. The minimum is 5,000 events. When the Rx buffer reaches maximum size, new events are dropped as they are received because they are throttled.

There are four Decide Types. They are:

- String - Compares a user-defined decide string to the content of the receive buffer. The contents of the decide string are verified with the contents of the receive buffer, or a variable, to determine which decision route to process. The decide string is a POSIX 1003.2 compliant regular expression. A variable supports strings, integers and floats.
- True - Forces an evaluation of true, Collector Manager follows the Yes route.
- False - Forces an evaluation of false, Collector Manager follows the No route.
- Data - Compares a user-defined decide string to another string or the value of a variable.

Parse State

The Parse State is used to develop the scripts to be executed on the ports. The parsing commands can include parameters that are merged with the template when the scripts are created. A Visual Editor and a Text Editor are available to define the parsing commands.

The Parse State is also used to insert parsing commands into a template. The parsing commands may include parameters, which are replaced with specific values when the template is merged with a parameter file in the script building process. Merging a template and a parameter file can output multiple scripts to execute on the ports.

Parameter Files

Parameters are the equivalent of variables. Parameter files (.par files) are tables used to define variable names on the associated run script files. They are used when referenced in the parsing code. Parameters are stored as strings. Any numeric value needs to be converted into a string for manipulation. When new values for parameters are entered, they take effect after you build your script. They are merged with the template file when creating a script.

Run script file names are displayed in the first row of the table and the parameter names or labels are displayed in the first column of the table. The second row of the table is used to define the icons that appear in the Collector’s tree. The remaining row defines the variables or parameter values to be used for parameter as it relates to the particular script.

Values within a parameter file are:

- Meta-tags, information and comments – there are over 200 available meta-tags, 100 are user configurable and the rest are reserved.
- Rule – set file names appear in the header row of the table, while parameters themselves appear in the first column in the table
- Bitmap – second row of the table, defines the bitmap used for that file. The bitmap will appear in the Collectors list.

Lookup Files

Lookup files are optional tables (.lkp files) against which received values are compared to determine what actions, if any, to take in response to security events. Lookup files contain match clauses, which are used to compare individual strings. Based on the match clauses in a specific lookup file and the data received from source devices, the LOOKUP Command will determine whether the search string is found or is not found.

Optionally, parsing commands may be associated with the match string. The parsing commands are executed if a match is found.

Mapping Files

Mapping files are optional files (.csv) that allow for fast lookup of key entries. The csv file is a relative path from a Collector’s script directory. The editing of these files is currently not available within Collector Builder, but the files can be edited using Excel.

Example of a possible mapping file is:

```
~Month~   ~Number~
Jan              1
Feb              2
Mar              3
Apr              4
May              5
Jun              6
Jul              7
Aug              8
Sep              9
Oct             10
Nov             11
Dec             12
```

The entries can be a variable number of script variables (string, variable or float) used to indicate which variables to store the data. This particular example is used to translate (map) Month to a Number (e.g., Jan to 1).

Manifest Files

Manifest files are what differentiates version 5.* Collectors from previous Collectors. Manifest files support the deployment of Collectors from the Sentinel Console and Collector versioning. Collector parsing is defined in the agent.lkp file. These lookup cases are:

- Setup - one time setup of variables and parameters
- Check_Setup - one time check of those variables and parameters
- Initialize_Vars - the beginning of every loop, where variables are initialized once per parse
- Parse - the place where the parsing is performed

This allows plugging in new Collector parsing into existing templates. In addition, it provides the ability to overlay new versions of the Collector's parsing to update the code. The following is list of the Manifest files and their contents for v5.0:

- agent.nfo
    - product,Snort
    - product.vendor,GNU
    - product.version,2.0
    - product.security.type,IDS
    - product.sensor.type,N
    - product.name,IDSx_GNUx_SNRT
    - file.version,1

Other Sentinel References

The following manuals are available with the Sentinel install CDs.

- Sentinel™ Sentinel Install Guide
- Sentinel™ User’s Guide
- Sentinel™ Wizard User's Guide
- Sentinel™ User's Reference Guide
- Sentinel™ 3rd Party Integration Guide
- Release Notes

Contacting Novell

- Website: http://www.novell.com
- Novell Technical Support: http://support.novell.com/filefinder/20653/index.html
- International Novell Technical Support: http://support.novell.com/phone.html?sourceidint=suplnav4_phonesup
- Self Support: http://support.novell.com/support_options.html?sourceidint=suplnav_supportprog
- For 24x7 support, 800-858-4000

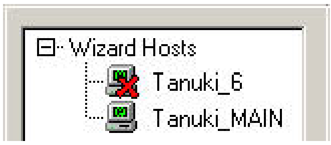
2 Managing Wizard Hosts

NOTE: The term Agent is interchangeable with Collector. Going forward, Agents will be referred to as Collectors.

Wizard hosts are machines that have Collector Manager installed. Hosts interact with Collector Builder machines and Sentinel over the network. Collectors receive and parse data. Based on this data, hosts will send alerts to Sentinel.

Wizard automatically detects hosts on the network and adds them to the list on the Wizard host tab. You cannot add hosts manually. You can rename existing hosts and delete hosts that are no longer physically present and communicating on the network.

Collector Builder collects health messages on hosts. If a host does not respond with a health message, the host will display a red X on the Wizard Hosts tree. You can remove a host with a red X, but if Collector Builder detects communications from that host, the host will re-appear in the Wizard Hosts tree. Similarly, if you remove a host that is already communicating, the health message returns it to the Wizard Hosts tree.



When a host is detected, it is assigned an identification number.

NOTE: For more information regarding configuration of the Demo Collectors, see the *Sentinel Install Guide, Testing the Installation*.

How a Wizard Host Gets Collector Data

To enable a Wizard Host (a machine with Collector Manager installed) to receive data from a Collector, you upload the Collector from a Collector Builder machine to the Wizard Host through a port configured on the Collector Builder. After a Collector is uploaded to a host, the host can receive data from that Collector.

Each Wizard Host may be connected to multiple ports and monitor data from multiple Collectors. A Wizard Host can have ports with Collectors that connect to many different types of data sources. Individual Collectors on a Wizard Port Host must be uploaded to run. In addition, ports provide Collector Manager with information on data source location.

Wizard Host Permissions

Wizard Host permissions is administered through the Sentinel Control Center Admin tab. Wizard Host user permissions are:

Permission Name	Description
View Collectors	▪ View the 'Collectors' tab in Sentinel Control Center ▪ View the 'Wizard Hosts' tab in Collector Builder
Control Collectors	▪ Includes all capabilities as the 'View Collectors' permission ▪ Allows for the Command and Control of Collectors from the Sentinel Control Center ▪ Allows for the Command and Control of Collectors from the Wizard Collector Builder
Collector Administration	▪ Includes all capabilities as the 'Command Collectors' permission ▪ In Collector Builder, Collector editing and deployment ▪ In Collector Builder, creating, editing, compiling and debugging Collectors. ▪ In Collector Builder, uploading and downloading Collectors. ▪ In Collector Builder, exporting Wizard Hosts ▪ In Collector Builder, adding, editing, deleting Ports ▪ In Collector Builder, setting Port options

Command and Control consists of:

- start/stop individual ports
- start/stop all ports
- restart hosts
- rename hosts

Wizard Host Management

The following are discussed in this chapter:

- Starting Collector Manager
- Stopping Collector Manager
- Collector Manager Administration
- Rename a host
- Delete a host
- Restart a host
- Export a host
- View host properties
- Edit a Template file
- Deleting a Template file
- Renaming a Lookup file
- Deleting a lookup file
- Deleting a Startup Sequence
- Starting and Stopping Wizard Ports
- Editing a Wizard Port
- Deleting a Wizard Port
- Uploading and downloading a Collector
- Debugging Wizard Ports

Starting and Stopping Collector Manager

NOTE: The first time the Wizard Collector Builder is run, you may see the following message: “Directory ‘Collectors’ does not exist.” It will be automatically created for you. Some information may have been lost.” Select OK; the directory will be created and the Wizard Collector Builder will launch. If this message displays beyond the first time the Collector Builder is run, the Collector directory might have been inadvertently deleted and you will need to verify if information was lost.

Starting or Stopping Collector Manager Service for Windows

Starting or Stopping Collector Manager Services for Windows

1. Click *Start* > *Settings* > *Control Panel.*
2. In the *Control Panel,* double-click *Administrative Tools* and click *Services.*
3. In the Services dialog box, right-click *Collector Manager* and click either *Start* or *Stop*.

Starting Collector Manager Services for Windows (command line)

1. Go to %WORKBENCH_HOME%
2. To start Collector Manager:
   - ./agent-manager start
   - ./agent-manager restart - starts the Collector Manager script in the background and automatically starts the Collector Manager process if it is stopped. If the agentmanager process is already running, it will stop and restart.
   - ./agent-manager.sh console – starts the Collector Manager process in foreground.

NOTE: While in console mode, ensure that you are only running one instance of Collector Manager on the machine.

Stopping Collector Manager Services for Windows (command line)

1. Go to %WORKBENCH_HOME%
2. To stop Collector Manager:

```
./agent-manager stop
```

Starting Collector Manager for UNIX (Normal and Console)

Starting Collector Manager for UNIX

1. As user esecadm, go to

```
$WORKBENCH_HOME
```

2. Enter the following command:

```
./agent-manager.sh start
```

   - ./agent-manager.sh restart - starts the Collector Manager script in the background and automatically starts the Collector Manager process if it is stopped. If the Collector Manager process is already running, it will stop and restart.
   - ./agent-manager.sh console – starts the Collector Manager process in foreground.

Stopping Collector Manager for UNIX

Stopping Collector Manager for UNIX

1. As user esecadm, go to

```
$WORKBENCH_HOME
```

2. Enter the following command:

```
./agent-manager.sh stop
```

Collector Manager Administration

There is a Collector Manager executable (Windows) and script (UNIX) that allow you to:

- Install the Windows Collector Manager Service (Windows only)
- Remove the Windows Collector Manager Service (Windows only)
- Set the Collector Manager Service
- Print extensive debug information
- Display the build version
- Display help

Installing the Windows Collector Manager Service (Windows only)

Installing the Windows Collector Manager Service (Windows only)

1. At the command prompt, go to %workbench_home%.
2. Enter the following command:

   ```
   agent-manager.bat install
   ```

3. To start the service, either:
   - At the command prompt enter:

     ```
     agent-manager.bat start
     ```

   - Click *Start* > *Settings* > *Control Panel*. Double-click on *Services*, select *Collector Manager*. *Start Collector Manager* service.

NOTE: If your Services window is already open, click *Action* > *Refresh* and start your *Collector Manager* service.

Removing the Windows Collector Manager Service (Windows)

Removing the Windows Collector Manager Service (Windows)

1. Stop the Collector Manager Service by either:
   - At the command prompt enter:

     ```
     agent-manager.bat stop
     ```

   - Click *Start* > *Settings* > *Control Panel*. Double-click on *Services*, select *Collector Manager*. Stop *Collector Manager* service. Close the Services window.
2. At the command prompt, go to %workbench_home%.
3. Enter the following command:

   ```
   agent-manager.bat remove
   ```

Changing Collector Manager Password for Windows

NOTE: In order to meet stringent security configurations required by Common Criteria Certification, It is strongly encouraged that a strong password be used with the following characteristics:

1. Choose passwords of at least 8 with characters in length that includes at least one UPPER CASE, one lower case, one special symbol (!@#$%^&*()_+), and one numeric (0-9).

2. Your password may not contain your e-mail name or any part of your full name.

3. Your password should not be a "common" word (for example, it should not be a word in the dictionary or slang in common use).

4. Your password should not contain words from any language, because numerous password-cracking programs exist that can run through millions of possible word combinations in seconds.

5. You should choose a password you can remember and yet is complex. For example, Msi5!YOld (My Son is 5 years old) OR IhliCf5#yN (I have lived in California for 5 years now).

Changing Collector Manager password for Windows

1. At the command prompt, go to %workbench_home%.
2. Enter the following command:

   CAUTION: You will not be prompted for password confirmation or for the old password.

   ```
   agent-manager.bat password <new password>
   ```

3. For the password to take affect, either:
   - At the command prompt enter:

     ```
     agent-manager.bat restart
     ```

   - At the Collector Builder, right-click on your host computer and select restart host.
   - Click *Start* > *Settings* > *Control Panel*. Double-click on *Services, select Agent Manager. Stop and start Agent Manager* service.

Changing Collector Manager Password for UNIX

NOTE: In order to meet stringent security configurations required by Common Criteria Certification, it strongly recommended that a strong password be used with the following characteristics:

1. Choose passwords of at least 8 with characters in length that includes at least one UPPER CASE, one lower case, one special symbol (!@#$%^&*()_+), and one numeric (0-9).

2. Your password may not contain your e-mail name or any part of your full name.

3. Your password should not be a "common" word (for example, it should not be a word in the dictionary or slang in common use).

4. Your password should not contain words from any language, because numerous password-cracking programs exist that can run through millions of possible word combinations in seconds.

5. You should choose a password you can remember and yet is complex. For example, Msi5!YOld (My Son is 5 years old) OR IhliCf5#yN (I have lived in California for 5 years now).

Changing Collector Manager password for UNIX

1. As user esecadm, go to $WORKBENCH_HOME.
2. Enter the following command:

CAUTION: You will not be prompted for password confirmation or for the old password.

```
./agent-manager.sh password <new password>
```

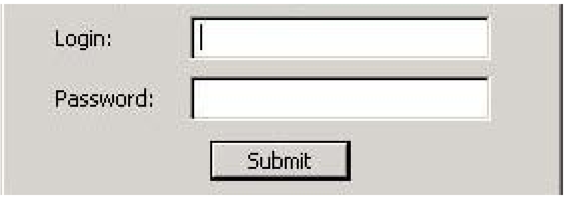
3. For the password to take affect, go to /usr/local/bin and enter the following command;

```
./agent-manager.sh restart
```

Starting Collector Builder

Starting Collector Builder

1. Click *Start* > *Programs* > *Sentinel* > *Collector Builder* or double-click on the *Collector Builder* icon on your desktop.
2. Depending upon your installation, login as esecadm or under your Windows authentication username.



Renaming a Wizard Host

Renaming a Wizard Host

1. Within Collector Builder (Wizard), click the Wizard Hosts tab to open the Wizard Hosts tree panel.
2. In the Wizard Hosts tree, right-click the host to be renamed and click *Rename Host*. You can only rename a host that is active.
3. Enter the new name of the host and press Enter.

NOTE: Renaming a host does not change the ID number that's assigned to a Wizard host when it is installed. This information is stored in %WORKBENCH_HOME%\wizard\agents\names.dat.

Deleting a Wizard Host

To delete a host, the host must first be removed from the network. Hosts that are communicating over the network cannot be removed. If a host is present on the network but not communicating, it will be displayed with a red X over the host icon in Wizard Hosts tree.

Deleting a Wizard Host

1. Click the *Wizard Hosts* tab to open the Wizard Hosts tree panel.
2. In the Wizard Hosts tree, right-click the host.
3. Click *Delete Host*.

Restarting a Wizard Host

Restarting a Wizard Host

1. Click the *Wizard Hosts* tab to open the Wizard Hosts tree panel and select a host.
2. Right-click a host and click *Start Ports*. You can only restart a Wizard Host that is active.

Exporting a Wizard Host

Exporting a Wizard Host

1. Click the Wizard Hosts tab to open the Wizard Hosts tree panel. Select a host.
2. Click *File* > *Export Host*. The following subdirectory is created:

   ```
   %WORKBENCH_HOME%\upload_<host name>
   ```

   This subdirectory can be moved to a remote machine using Secure Shell (SSH) or a disk. After the subdirectory is put on the remote machine, run the uploadhost command. This copies the necessary files to the proper directories.

   NOTE: If the SNMP settings are changed, the Collector Builder will not be able to communicate with the remote machine from the time the Export button is pressed until the exported Collector files are uploaded.

Viewing Wizard Host Properties

Viewing Wizard Host Properties

1. Click the Wizard Hosts tab to open the Wizard Hosts tree panel.
2. In the Wizard Hosts tree, right-click the host and click *Properties*. The Wizard Properties window displays the following information:
   - Name
   - ID
   - Hostname
   - IP Address
   - Version
   - Uptime
3. Click *OK* to close the Properties window.

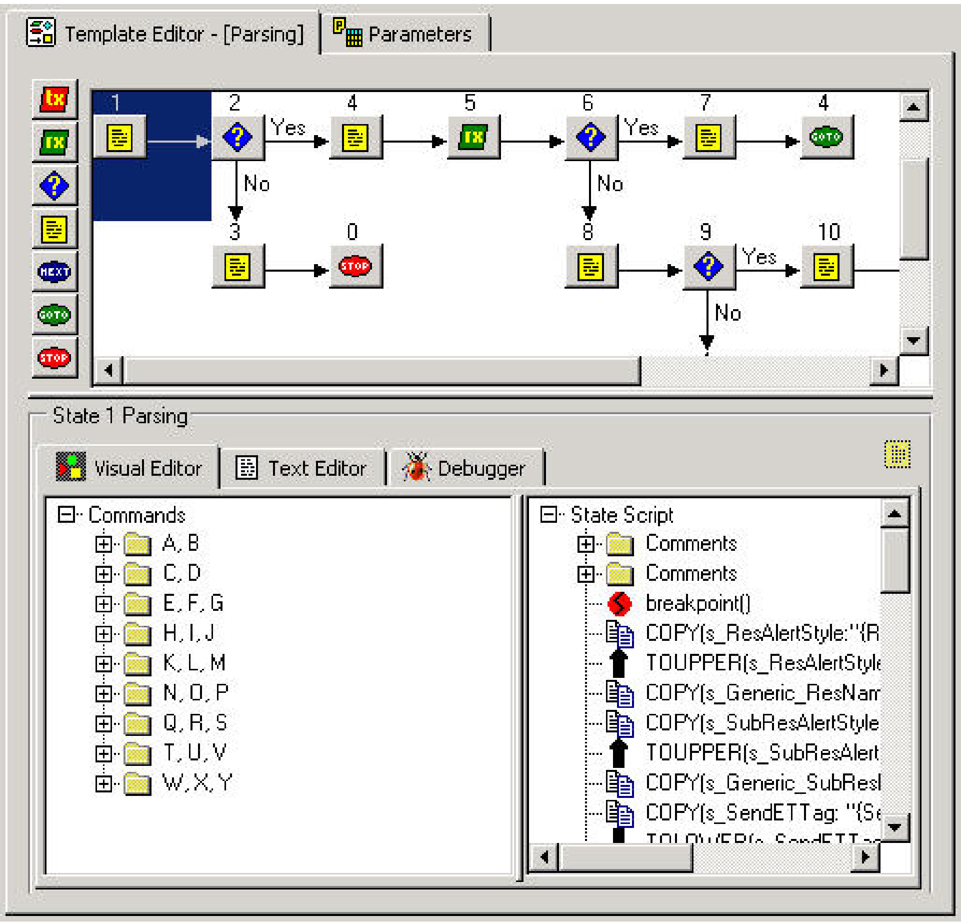
   NOTE: If the host is not running, a No Response window will display when you select Properties.

Editing a Template File

Editing a Template File

1. Click the Collectors tab to open the Collector tree panel.
2. In the Collector tree, click the template and click the Template Editor tab on the right.

3. In the Template Editor, click the state to edit and make your desired changes. You can edit a state using the Visual Editor or Text Editor. For information about Parsing Commands, see the Sentinel User's Reference Guide.



```
breakpoint()
COPY(s_ResAlertStyle:"{R
TOUPPER(s_ResAlertStyle
COPY(s_Generic_ResNam
COPY(s_SubResAlertStyle
TOUPPER(s_SubResAlert
COPY(s_Generic_SubResl
COPY(s_SendETTag: "{Se
```

Deleting a Template File

Deleting a Template File

1. Click the *Collectors* tab to open the Collectors tree panel.
2. In the Collectors tree, right-click a template and click *Delete Template*.

Renaming a Lookup File

Renaming a Lookup File

1. Click the *Collectors* tab to open the Collectors tree panel.
2. Right-click the lookup file and click *Rename Lookup* File.
3. Type in the new name and press *Enter*.

Deleting a Lookup File

Deleting a Lookup File

1. Click the *Collectors* tab to open the Collectors tree panel.
2. Right-click the lookup file and click *Delete Lookup File*.

Deleting a Script

Deleting a Script

1. There are two ways to delete a script.
   - In the Collector tree, right-click a script and click *Delete*
   - Right-click the script in the Startup Scripts or the Backout Scripts column and select Delete Script.

Deleting a Startup Sequence

Deleting a Startup Sequence

1. In the Startup Scripts panel, select the startup sequence from the dropdown menu so that sequence name displays in the Startup Scripts box.
2. Right-click the script in the Collectors tree and select Delete Current Startup Sequence. The startup sequence is removed from the Startup Scripts list.

   NOTE: If you delete the default startup sequence, any scripts assigned to the default startup sequence are removed from the Startup Scripts column, but default still displays in the Startup Sequences menu.

Wizard Ports

This section discusses how to stop, start, edit, delete and debug a Wizard Port.

Starting and Stopping a Wizard Port - GUI

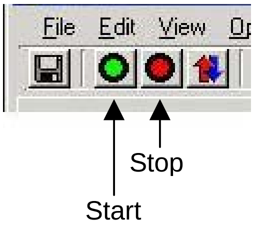
When a Collector is started or stopped, the Start or Stop button under the Start/Stop column will change once the Collector actually starts or stops. If you are working with a remote Collector, this change may be delayed while waiting for the Collector’s status to be received.

Starting or stopping a port executes the selected startup script and backout script.

When starting all ports, a port will only start if the Run Port at Startup box is checked under the Other Port Options of the Options menu.

Starting and Stopping all Wizard Ports

1. In the Wizard window, to:
   - To stop all ports, click the stop button in the tool bar.
   - To start all ports, click the start button in the tool bar.



Starting and Stopping an Individual Wizard Port

1. In the Wizard window, to:
   - To stop a port, click the stop button in the Start/Stop column corresponding to the port
   - To start a port, click the start button in the Start/Stop column corresponding to the port

Editing a Wizard Port

If you edit a port’s configuration while it is running, the port will stop. To avoid data loss, stop the port manually before editing the port configuration.

Editing a Wizard Port

1. For the proper host, stop the port.
2. Follow the steps for creating a Wizard Port in Chapter 3. The new configuration will overwrite the existing configuration when you save or upload the port.

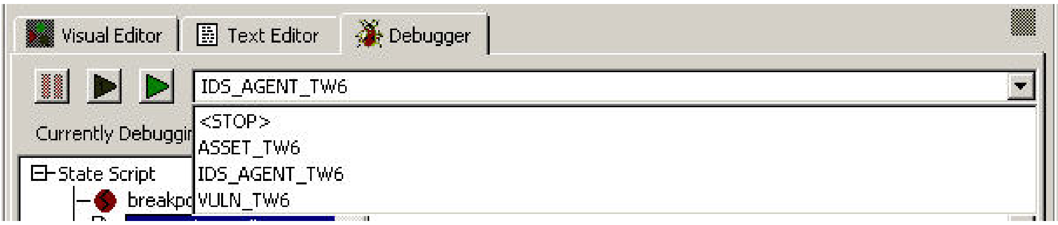
Deleting a Wizard Port

Deleting a Wizard Port

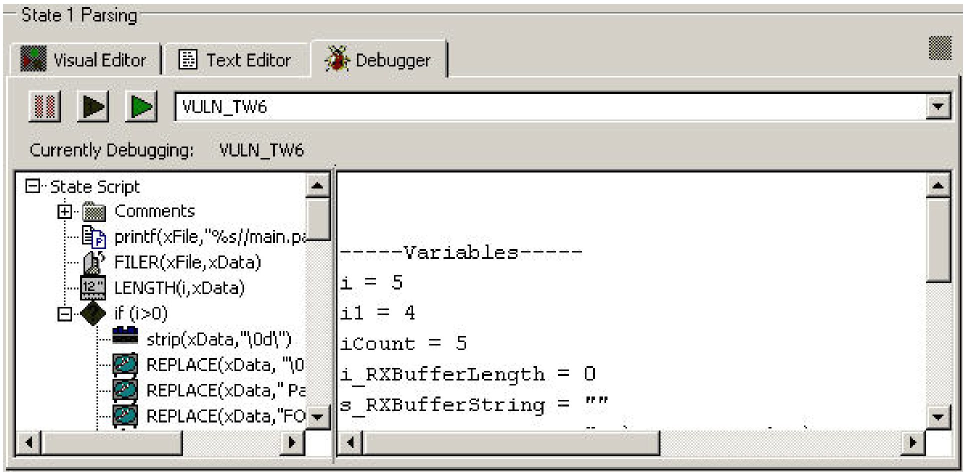
1. Stop the port.
2. In the Collector Builder’s Port Information panel, right-click the port name and click Delete Port. All ports below the deleted port will be stopped automatically.
3. If you are deleting from a:
   - Local host - Click *File* > *Save* and select Port Information
   - Remote host - Click *File* > *Upload/Download*

Debugging a Wizard Port

The Debugger allows you to troubleshoot the Collector code running on a port. The left side of the Debugger panel displays the State Script. The right side of the panel displays scripts and RX_Buffer variables, which can have names of up to 32 characters.



For the Debugger to be effective, you must have a parse state as the first state and have Breakpoint() Commands.



VULN_TW6

Currently Debugging: VULN_TW6

```
State Script
  Comments
  printf(xFile,"%s//main.pa
  FILER(xFile,xData)
  LENGTH(i,xData)
  if (i>0)
    strip(xData,"\0d\")
    REPLACE(xData, "\0
    REPLACE(xData," Pa
    REPLACE(xData,"FO
```

```
-----Variables-----
i = 5
i1 = 4
iCount = 5
i_RXBufferLength = 0
s_RXBufferString = ""
```

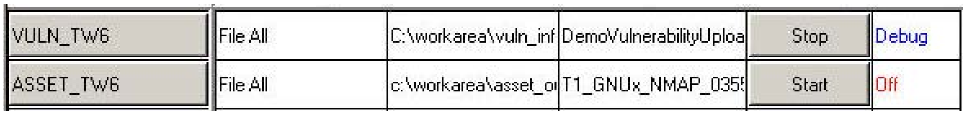
While debugging, wait for the Rx Buffer to update before performing another function.

> **NOTE:** If your Collector Manager host has lost connectivity (☒), you cannot debug a port for that Collector Manager host.

Debugging a Wizard Port

1. In the Template Editor, select the Debugger tab in the editing panel to access the Debugger. A blank panel displays allowing you to select the Wizard Port you want to debug from a dropdown list.

   If you click the Wizard Hosts tab, the port being debugged will indicate that it is in debug mode.



VULN_TW6	File All	C:\workarea\vuln_inf	DemoVulnerabilityUploa	Stop	Debug
ASSET_TW6	File All	c:\workarea\asset_o	T1_GNUx_NMAP_035	Start	Off

2. From the dropdown list, select a port to start the debug process. Debug your port by either:

   - Press F6 to step through the commands one at a time or click the Execute One Command button.

     ▷

     Click the button again or press F6 to resume script execution.

   - Press F7 run through commands or click the Resume Command Execution button.

     ▶

     Press F5 to pause or click the Pause Command Execution button.

     ❚❚

     It pauses until you press F7 or the Resume Command Execution button.

   The Debugger stops at all breakpoints, but continues to run. The port’s status is “on.”

   No events are sent out during pauses during debug mode.

When the parser exits, the buttons are grayed and the selection list displays “No Port is being Debugged.”

Debugger will not break out of a pause, so if you are debugging a parser that has hit a pause command, the Stop button or the Step button will wait until the pause is complete before taking effect.

Uploading and Downloading Collectors and Hosts

There are three tabs in Upload/Download window, they are:

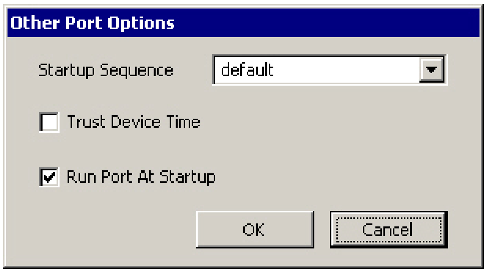
- Hosts – will upload each individual port configuration and Collector collection to each of the specified hosts. Each host will still have their own port configuration and Collector collection.
- Collectors – for uploading individual Collectors
- Populate Network – will upload the port configuration/agents of a single specified host to all of the selected hosts. All selected hosts will get the same port configuration and Collector collection as the source host.

When downloading, the port configuration on a remote Collector will appear on the host you choose to download and any Collector on the remote host with the same name as the local host will be overwritten.

Uploading a Collector to a Single Host

Uploading a Collector to a Single Host

1. If your Collector is already properly configured and you have built your script, you can skip steps 2 to 11.
2. Click the Wizard Hosts tab and select a host.
3. Under the Port Name column, double-click *new…* Enter a name of your choice.
4. Under the Collector column, select a Collector.
5. As per the Collector documentation (`%WORKBENCH_HOME%\Elements\<Collector name>\docs\<file name>.pdf`) configure the Collector.
6. Click the *Collectors* tab, expand the Collector and highlight the template file.
7. To the right, click the *parameters* tab.
8. As per the Collector documentation, set your parameter values.
9. (optional) If you want this Collector not to start upon startup or to trust device time, click the *Wizard Host* tab, right-click on the *Wizard Port* name, select *Other Port Options…* and unclick *Run Port At Startup* or click *Trust Device Time*. Click *OK*.


Other Port Options

Startup Sequence: default

☐ Trust Device Time

☑ Run Port At Startup

10. Click *save*.
11. Click the *Collectors* tab, right-click on the template file and select *Build Script*.

12. Click either:
    - *File* > *Upload/Download*
    - right-click on the *Collector* and click *Upload Collector*
    - click the *Upload/Download* button ⇅

    The *Upload/Download* window opens.
13. In the *Upload/Download* window, click the *Collectors* tab.
14. From the drop down list, select which *Collector* you would like to upload.
15. Click *Upload*. The first time you do this, you will be prompted for a *Collector Manager* password, even for a local Wizard host. The Transfer Progress window opens and shows the progress of the upload.

NOTE: You can use the Transfer Progress window to restart hosts after a transfer.

Uploading a Collector to Multiple Hosts

Uploading a Collector to Multiple Hosts

CAUTION: If you upload a host that has a Collector with the same name as one in your local host, the Collector in your remote host will be overwritten without notice.

1. If your Collector is already properly configured and you have built your script, you can skip steps 2 to 11.
2. Click the *Wizard Host*s tab and select a host.
3. Under the Port Name column, double-click *new…* Enter a name of your choice.
4. Under the *Collector* column, select a Collector.
5. As per the Collector documentation (`%WORKBENCH_HOME%\Elements\<Collector name>\docs\<file name>.pdf`) configure the Collector.
6. Click the *Collectors* tab, expand the Collector and highlight the template file.
7. To the right, click the *parameters* tab.
8. As per the Collector documentation, set your parameter values.
9. (optional) If you want this Collector not to start upon startup or to trust device time, click the *Wizard Host* tab, right-click on the Wizard Port name, select Other Port Options… and unclick ‘Run Port At Startup’ or click ‘Trust Device Time’. Click OK.
10. Click save.
11. Click the Collectors tab to open the Collectors tree panel
12. Click a Collector.
13. Click either:
    - *File* > *Upload/Download*
    - click on the Collector and select *Upload Collector*
    - click the Upload/Download button ⇅

    The Upload/Download window opens.
14. In the Upload/Download window, click the Hosts tab and select or clear the Upload Collectors When Uploading check box.

If this check box is selected, Collectors selected on the Collectors tab will be uploaded. This check box is selected by default. This option has no effect when downloading Collectors from a host.

15. From the list, select the Wizard hosts that you want to upload Collectors to.

    All Wizard hosts on the network will automatically be included in the list. The buttons indicate whether or not the host machine is online.

    Click Select All to select all Wizard hosts in the list. Click Select None to deselect all Wizard hosts in the list.

16. Click Upload to upload selected Collectors to the selected host(s). The first time you do this, you will be prompted for a Collector Manager password, even for a local Wizard host.

Downloading a Host

Downloading a Host

CAUTION: If you download a host that has a Collector with the same name as one in your local host, the Collector in your local host will be overwritten without notice.

1. Click the Wizard Hosts tab to open the Hosts tree panel.
2. In the Wizard Hosts tree, click the host you want to download.
3. Click either:
   - File > *Upload*/*Download*
   - click on the Collector and select *Upload Collector*
   - click the Upload/Download button

   The Upload/Download window opens. The Collector you have selected is checked by default.
4. Click *Download*. The first time you do this, you will be prompted for a Collector Manager password, even for a local Wizard host. The host is downloaded and added to the Wizard Hosts tree. The Transfer Progress window opens and shows the progress of the download.

NOTE: You can use the Transfer Progress window to restart hosts after a transfer.

NOTE: You can only download one host at a time. Checking more than one host will prevent downloads from occurring.

Downloading Collectors from a Single Host

Downloading Collectors from a Single Host

1. Click either:
   - *File* > *Upload*/*Download*
   - click the Upload/Download button

   The Upload/Download window opens.
2. From the list, select the Wizard host that you want to download Collectors from.

   All Wizard hosts on the network will automatically be included in the list. The buttons indicate whether or not the host machine is online.

Click Select All to select all Wizard hosts in the list. Click Select None to deselect all Wizard hosts in the list.

3. Click Download to download Collectors from the selected host.

Uploading Ports to Multiple Hosts

Uploading Ports to Multiple Hosts

1. Click either:
   - *File > Upload/Download*
   - click the Upload/Download button ⇅
2. The Upload/Download window opens.
3. In the Upload/Download window, click the Populate Network tab.
4. From the list labeled ‘Select which host’s port configuration and Collectors you would like to upload’, select the host you want to upload port configuration settings and Collectors.
5. From the list labeled ‘Select which hosts you would like to upload this configuration to’, select the host you want to upload the selected settings.

   All Wizard hosts on the network will automatically be included in the list. The buttons indicate whether or not the host machine is online.

   Click Select All to select all Wizard hosts in the list. Click Select None to deselect all Wizard hosts in the list.

Uploading Multiple Collectors to a Network

Uploading Multiple Collectors to a Network

1. From the Wizard main window, select a Collector on the Collectors tree.
2. Click either:
   - *File > Upload/Download*
   - click on the Collector and select *Upload Collector*
   - click the Upload/Download button ⇅
3. Select the Populate Network tab.
4. In the first selection box, from the dropdown menu, select which host’s port configuration and Collectors you would like to upload.
5. In the second selection box, from the dropdown menu, select which hosts you like to upload the configuration to.

   NOTE: You must have at least one selection in at least one of the selection boxes in order to upload its configuration.

   You may select a different Collector for each dropdown box. Each Collector checked in the main list will acquire the port configuration and Collectors of the host selected in the box labeled:

   “Select which host’s port configuration and Collectors you want to upload” unless None is selected.

6. When you have completed setting up the network configuration, select the Upload button to begin the upload process.

Upgrading Collectors

Upgrading Collectors

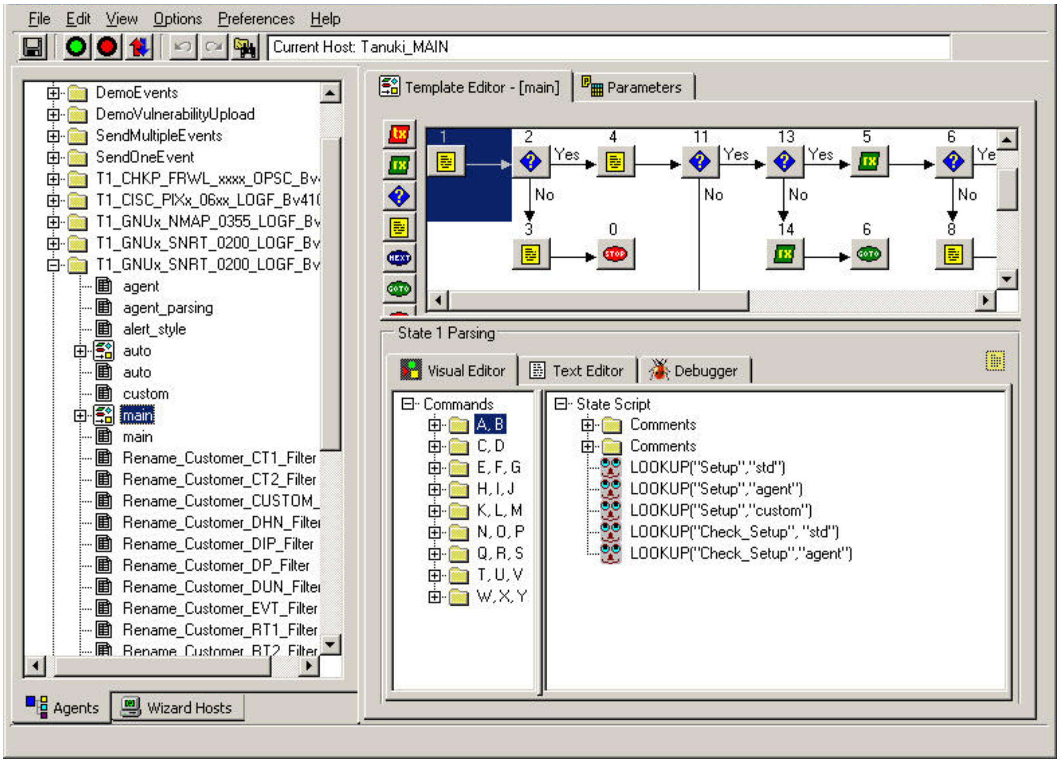
1. Read the documentation that comes with the new Collector that explains any changes.
2. Put the new version of the Collector in %workbench_home%/Elements directory on the PC that is the master for the Collector.
3. Open the parameter file of the Collector being replaced, cut and paste matching parameters into the new Collector.
4. If necessary as per the documentation for the new Collector, remove or add new parameter variables. If you are adding new parameter variables, populate the variable.
5. Save the parameter file in the new Collector.
6. Build the new Collector.
7. Edit the port configuration information to use the new Collector.
8. Save the port configuration information.
9. Upload the new Collector and port configuration.
10. Restart the port.

3 Building and Maintaining Collectors

NOTE: The term Agent is interchangeable with Collector. Going forward, Agents will be referred to as Collectors.

NOTE: For MS SQL 2000 users, the event size cannot exceed 8KB.

A Collector is responsible for parsing the data from a security event source and sending events to Sentinel. Collectors are built, activated and maintained using the Wizard Collector Builder. Click the *Collectors* tab to display the Collectors tree to see all of the Collectors and Collector components on your Sentinel system.



Collector Manager allows you to:

- Build Collectors
  - Creating and Configuring Template Files
  - Creating Parameter Files
  - Creating Lookup Files
  - Building Scripts
  - Creating a Wizard Port

Collector Building Basics

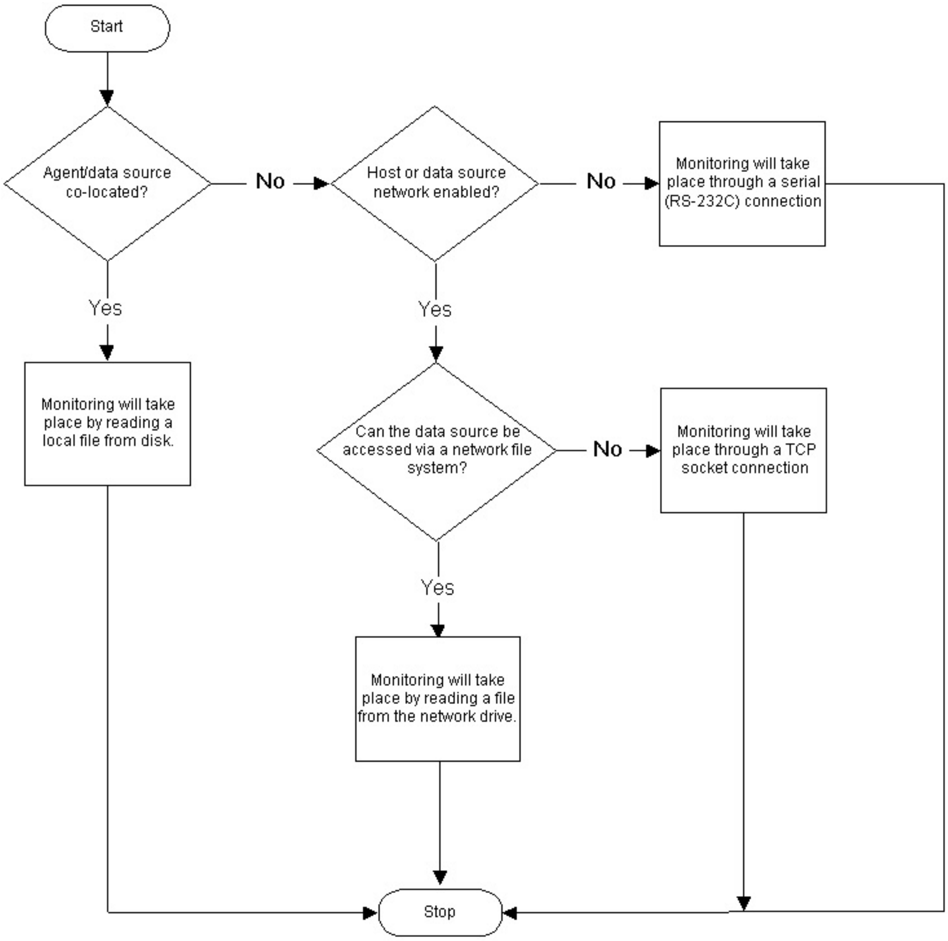
The basic Collector building steps are:

- Create and configure a template file, including decision points based on how you apply states
- Create and configure a parameter file
- Create and configure a lookup file (optional)
- Building a script
- Assign a startup sequence
- Create a port, assign the Collector to the port and start the port

Basic Collector Implementation Steps

The following are basic steps in implementing an Collector.

- Determine what you want to monitor
- Determine how to monitor the data
- Determine the product's operating system
  - If the host and product are co-located, the most logical way to obtain the data is to read it from the product's log file.
  - If the host and product are not located on the same machine, the needed data can be obtained through a network file system setup (such as NFS, Samba or SMB share), a TCP/IP socket connection or a serial connection.
- Build the Collectors and start the ports.
- If remote hosts are used, upload the Collector files to those remote hosts. Start the port to execute the startup scripts; the information collected will be reported through the Sentinel system.


Start
Agent/data source co-located?
No
Host or data source network enabled?
No
Monitoring will take place through a serial (RS-232C) connection
Yes
Yes
Monitoring will take place by reading a local file from disk.
Can the data source be accessed via a network file system?
No
Monitoring will take place through a TCP socket connection
Yes
Monitoring will take place by reading a file from the network drive.
Stop

Building a Collector

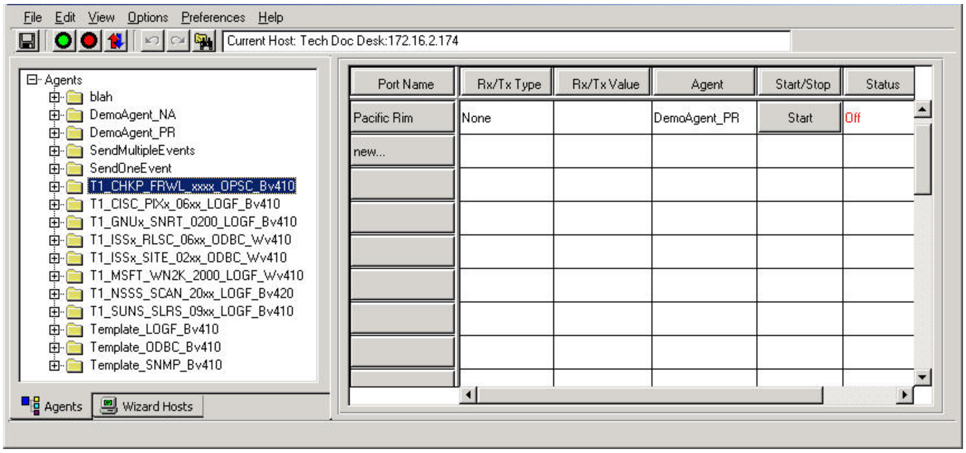
As previously discussed, building an Collector requires you to create:

- Template files
- Parameter files
- Lookup files (optional)
- Scripts
- Assign a Wizard Port Name to an Collector

Creating and Configuring Template Files

Creating and Configuring Template files

1. Start Collector Builder.
2. Click the *Collectors* tab to open the Collectors tree panel.
3. In the Collectors tree, right-click *Collectors* and click *New Collector*.
4. Enter the new Collector name in the space provided and press Enter.
5. Right-click on the new *Collector* and click *New Template.*



Current Host: Tech Doc Desk:172.16.2.174

- Agents
  - blah
  - DemoAgent_NA
  - DemoAgent_PR
  - SendMultipleEvents
  - SendOneEvent
  - T1_CHKP_FRWL_xxxx_OPSC_Bv410
  - T1_CISC_PIXx_06xx_LOGF_Bv410
  - T1_GNUx_SNRT_0200_LOGF_Bv410
  - T1_ISSx_RLSC_06xx_ODBC_Wv410
  - T1_ISSx_SITE_02xx_ODBC_Wv410
  - T1_MSFT_WN2K_2000_LOGF_Wv410
  - T1_NSSS_SCAN_20xx_LOGF_Bv420
  - T1_SUNS_SLRS_09xx_LOGF_Bv410
  - Template_LOGF_Bv410
  - Template_ODBC_Bv410
  - Template_SNMP_Bv410

Port Name	Rx/Tx Type	Rx/Tx Value	Agent	Start/Stop	Status
Pacific Rim	None		DemoAgent_PR	Start	Off
new...					

6. In the New Template box on the Collectors tree type a new template name and press Enter.
7. Select the new template and click the *Template Editor* tab.
8. In the *Template Editor* panel, drag and drop states to the editing area using the state buttons on the left of the panel. For information about adding states to a template, see Adding a States to a Template.
9. Click *Save*.

Adding a State to a Template File

All Collectors begin processing at state 1, regardless of where state 1 appears in the template. Assuming state 1 is a processing state, insert the new state following state 1.

Collector Builder automatically assigns the first state a state number of 1. It is recommended that this first state contain only a BREAKPOINT() parsing command. Putting only a breakpoint after State 1 will allow for easier debugging. When debugging, the parser will automatically stop on the next state.

When building a template, start with a ‘breakpoint only’ parse state. Then, add the working state (Receive state, Parse state, etc.) at State 2. If you need to add a state to the beginning of the template, insert it after the BREAKPOINT only.

Do not delete the BREAKPOINT only parse state unless it is necessary to add another state at the beginning of the template. Optionally, you can enter comments in this BREAKPOINT only about the functionality of the template.

How to Add a State to a Template

1. Click the *Collectors* tab to open the Collectors tree panel.
2. In the Collectors tree, select a template to display the Template Editor in the right panel.
3. Click *Options > Add State > Transmit*, *Receive*, *Decide*, *Parse*, *Next*, *Go To or Stop* states as needed or click the appropriate buttons.
   - tx Transmit
   - rx Receive

- Decide
- Parse
- Next
- Go To
- Stop

4. Using the editing panels at the bottom of the Template Editor panel, insert the new code into each state as you add it.

   Another method is to drag and drop a Parse State button from the left side of the Template Editor into the editing area.

   NOTE: Do not use double quotation marks as part of the decide string either in the receive state (to match the delimiter in a log file, for example) or in a decide state, you will get the following error message:

   ```
   ***ERROR: Reading Template File..."
   ```

   When one or more quotes is put into the decide or delimiter string, a quote mismatch occurs as follows:

   ```
   StateDecideString: "test"123"
   ```

   The workaround is to use \22\ instead of a quote (").

   NOTE: If you select another item in the Collectors tab (even in the same Collector) and then go back to the offending template, Collector Builder gives you this error message and will not display any part or states of the template. The error is occurring because the quote character (") is used to delimit field values in a .tem file. For example:

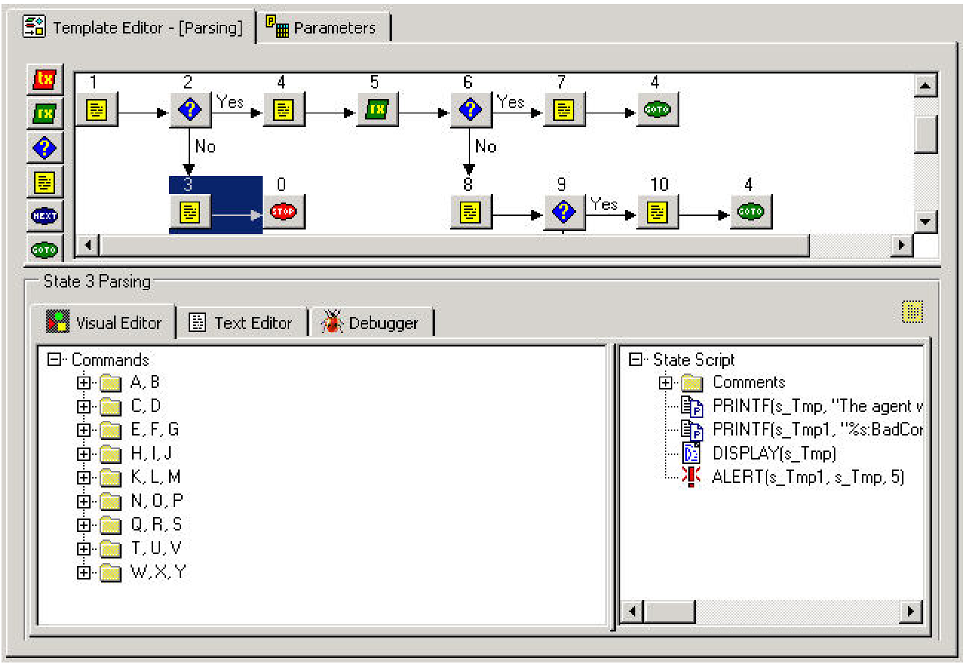
   ```
   StateDecideString: "test"
   StateDelimiterString: "123"
   ```

Entering a Parsing Command via the Visual Editor

There are two methods of entering a Parsing Command, using the Visual Editor or using the Text Editor. Limit your commands to no more than 4096.

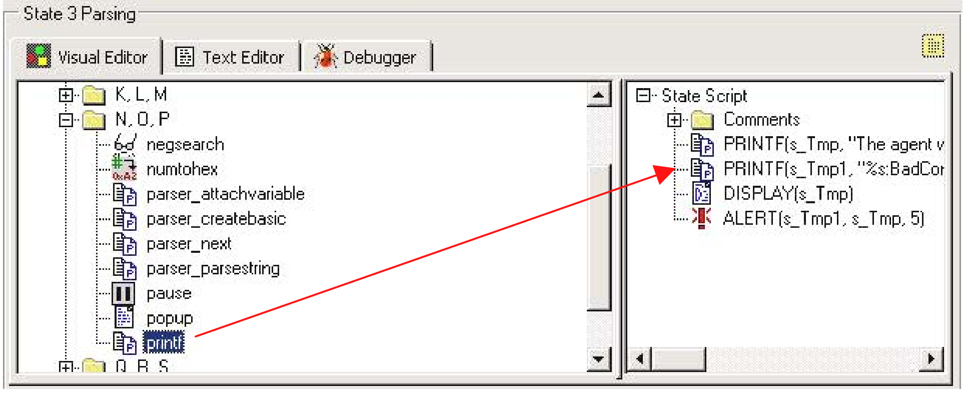
Entering a Parsing Command via the Visual Editor

1. In the Template Editor, select a parse state. The Visual Editor tab is open by default when you click a template to open.



```
PRINTF(s_Tmp, "The agent v
PRINTF(s_Tmp1, "%s:BadCor
DISPLAY(s_Tmp)
ALERT(s_Tmp1, s_Tmp, 5)
```

2. In the visual editor, drag the parsing commands to the right side of the panel.



```
PRINTF(s_Tmp, "The agent v
PRINTF(s_Tmp1, "%s:BadCor
DISPLAY(s_Tmp)
ALERT(s_Tmp1, s_Tmp, 5)
```

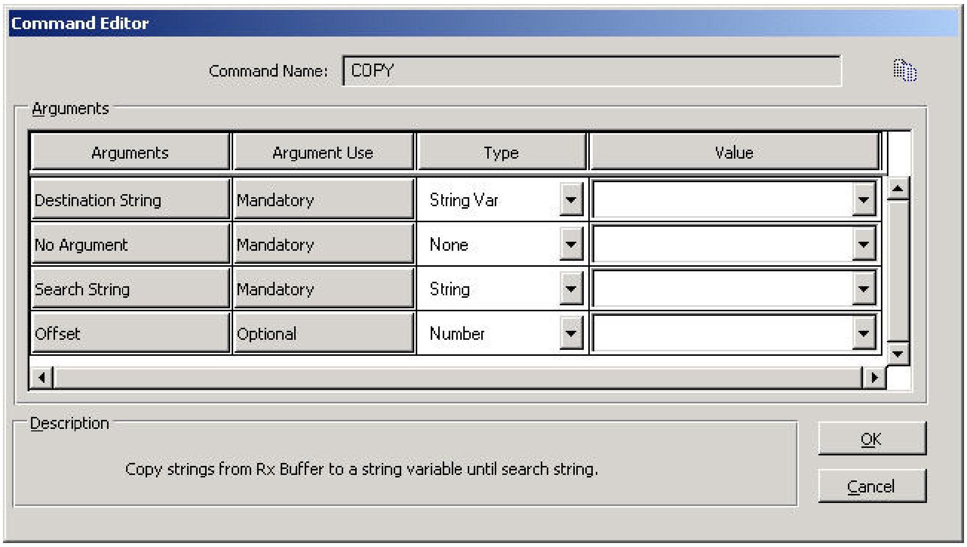
3. Enter the argument values in the Popup Command Editor window.
   - Select a Type – the types for each parsing command is described in the Sentinel User’s Reference Guide.
   - Specify a Value - Values are defined for a specific application. Examples of values for each parsing command are in the Sentinel User’s Reference Guide.

Entering a Parsing Command via the Text Editor

1. In the Template Editor, click the *Text Editor* tab.
2. Manually enter your parsing commands.

   Use the Tab key on the keyboard to line up text when using a fixed font. Copy, cut and paste options function like any standard text editor.

Editing a Parsing Command



Command Name: COPY

Arguments

Arguments	Argument Use	Type	Value
Destination String	Mandatory	String Var	
No Argument	Mandatory	None	
Search String	Mandatory	String	
Offset	Optional	Number	

Description

Copy strings from Rx Buffer to a string variable until search string.

- Arguments - Includes all of the possible arguments for the parsing command you selected in the Visual Editor
- Argument Use - Defines whether the argument is mandatory or optional
- Type - Determines the variable type; for example, strings, string variables, numbers, number variables, floats, float variables or predefined variables
- Value - Value you define for the variable that's named in the Type column

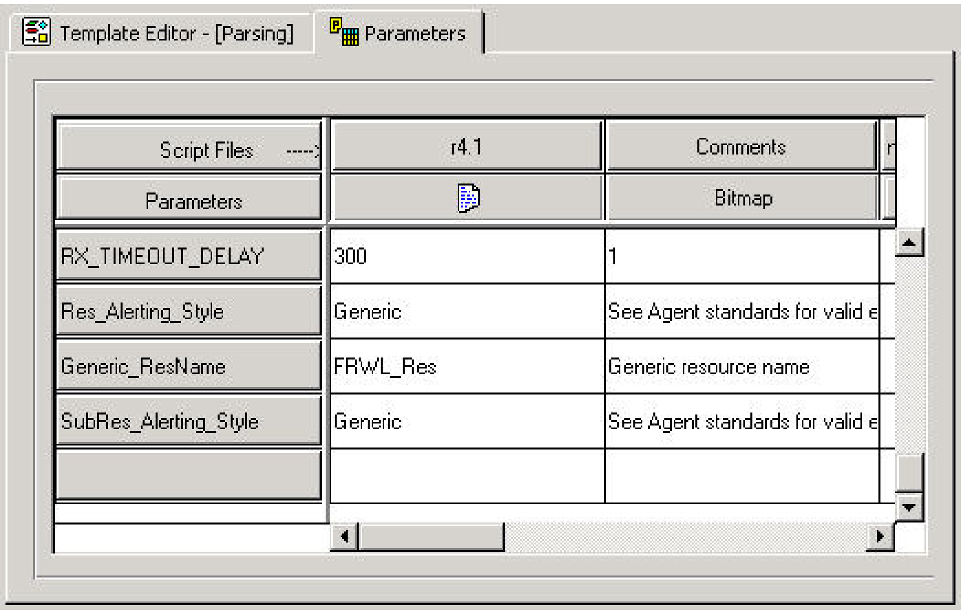
Editing a Parsing Command

1. In the visual editor, either:
   - Right click on a parsing command and choose *Add to State Parsing List*
   - Double-click on a parsing command, the Command Editor will open
2. Fill in the Type and Value boxes to complete the editing. See Sentinel User's Reference Guide for more information on Parsing Command descriptions.

Creating and Configuring Parameter Files

Creating and configuring Parameter Files

1. Click the *Collectors* tab.
2. Select a template and click the *Parameters* tab in the right panel.



Script Files ----->	r4.1	Comments	r
Parameters	🖹	Bitmap	
RX_TIMEOUT_DELAY	300	1	
Res_Alerting_Style	Generic	See Agent standards for valid e	
Generic_ResName	FRWL_Res	Generic resource name	
SubRes_Alerting_Style	Generic	See Agent standards for valid e	

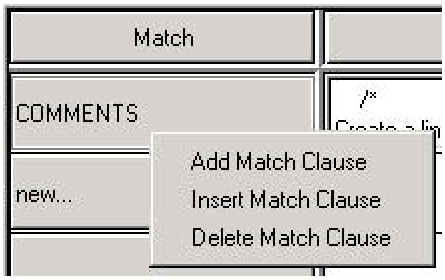
3. Double-click the *new...* button in the first column of the Parameters table.
4. Enter the new parameter name (this is the name of your script, such as r4.1) and press Enter.
5. (Optional) Right-click the *Bitmap button* (second column/second row) and click *Assign Bitmap*. In the Bitmap Assignment dialog box, select a *Bitmap* button.
6. Double-click each of the new parameter boxes and enter the appropriate values.
7. When all of the values are defined, the parameter and the template file need to be compiled to create a script. Go to section Building Scripts.

Creating and Configuring Lookup Files

This is an optional procedure.

Creating and configuring Lookup Files

1. Click the *Collectors* tab to open the Collectors tree panel.
2. Right-click a Collector and click *New Lookup File*.
3. In the New Lookup File box, type a new lookup file name and press Enter.
4. In the Match column, Double-click *new...* and enter the string to match and press Enter. You can add, insert and delete match clauses.
   - To add - In the Match column, right-click a match clause and click *Add Match Clause*.
   - To insert - In the Match column, right-click a match clause and click *Insert Match Clause*.
   - To delete - In the Match column, right-click a match clause and click *Delete Match Clause*.



5. (Optional) To enter parsing commands, right-click in the Parsing column to open the Visual Editor. For information about using the Visual Editor, see To Enter Parsing Commands Using the Visual Editor.
6. Select the parsing commands and complete them in the Command Editor window. The commands display in the Parsing column.
7. When all of the values are defined, it must be compiled to create a script. Go to section Building a Script.

Scripts

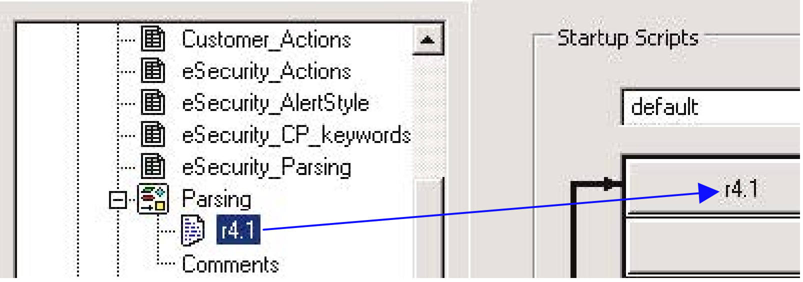
Scripts are generated from templates. You can generate multiple scripts from one template. Collector Manager allows you to:

- Build a script
- Debug a script
- Assigning a Startup Sequence to a Script

Building a Script

Building a script

1. Click the *Collectors* tab to open the Collectors tree panel.
2. In the left panel, select the template that you are building the scripts from.
3. Select *File* > *Build Scripts*.
4. In the Template Editor tab, drag a script from the template to the Startup Scripts or Backout Scripts column in the right panel.

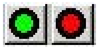


Scripts execute in the order they appear in the Startup Scripts and Backout Scripts columns. To rearrange the script order, drag the scripts up or down in the columns.

NOTE: The final script in a backout sequence must end with the Stop processing state.

5. (Optional) Debug using the debugger.
6. Click *File* > *Save*.

7. For the changes to take effect, stop and start the port using the Stop and Start buttons on the tool bar.

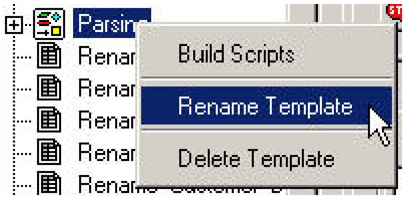


Enabling AutoBuild for Collectors Prior to Version 5.0

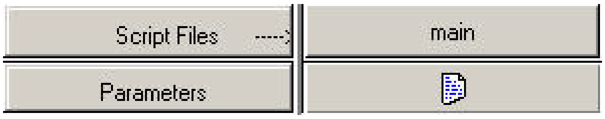
Enabling the AutoBuild feature allows you to skip the step of building your script when you configure and deploy Collectors.

To Enable AutoBuild for Collectors Prior to Version 5.0

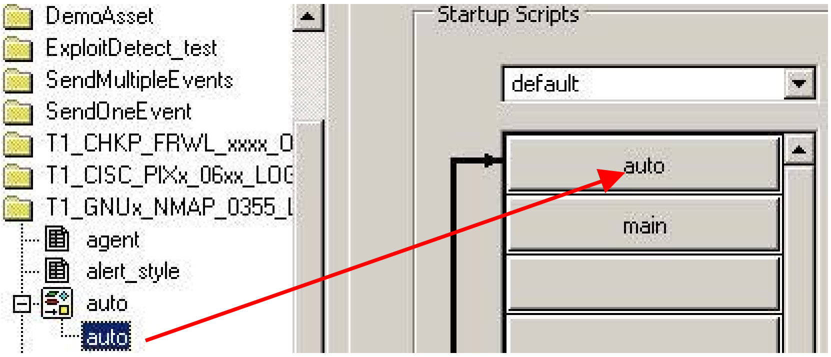
1. Copy the following files from an existing v5.* Collector and copy them into the Collector you wish to enable AutoBuilding.
   - auto.tem
   - auto.asd
   - auto.lkp
   - auto.par
2. Rename your template file to main.tem. You can do this in Collector Builder.



3. Highlight the renamed template file and click the *Parameters* tab. Change the column header name that has the name of your script file (for instance r4.1) to main and hit the enter key.



4. Click the *Save* button.
5. In the Startup chain, right click and drag your auto.asd before main.



Debugging a Script

When you begin the debugging process, the port's status is set to "Debug" on the Port Information panel. To debug a script, see Debugging a Wizard Port in Chapter 2.

Assigning a Startup Sequence to a Script

If you want a port to run at startup, you can assign a startup sequence to run a specified set of scripts at startup. A startup sequence is a file that contains the names of the scripts to run at startup.

Assigning a Startup Sequence to a Script

1. Right-click a script name in the Collectors tree and select New Startup Sequence. The New Startup Sequence dialog box displays.
2. In the New Startup Sequence dialog box, type the sequence name and click *OK*. The new startup sequence name is added to the menu at the top of the Startup Scripts panel. The following restrictions apply to sequence names:
   - Do not use startup or backout as sequence names
   - Do not use duplicate sequence names within the same Collector
3. Drag the script file names from the Collectors tree to the Startup Scripts column. The scripts execute in the order that they appear in the column, from top to bottom.
4. To rearrange the script order, drag the scripts from the column, or right-click the *Startup Scripts* panel and select *Reorder Startup Script.*

Creating a Wizard Port

You can create more than one port for an Collector. For some types of sensors, you may need to create more than one instance of the same Collector and assign each instance to a different port.

A port’s connection type determines how security data and what information will be read and when a connection will be established. The connection types are:

- Serial Connection Type
- Socket Connection Type
- File New Connection Type
- File All Connection Type
- Persistent Process Connection Type
- Transient Process Connection Type
- SNMP Trap Connection Type
- None Connection Type

Serial Connection Type

The Serial connection type is used if data will be read from an RS-232C serial port (through either a serial cable or a modem connection). You must specify the appropriate serial port (for example, COM1, COM2) in the Rx/Tx Value box. The host running the product to be monitored must also have a serial connection to the Collector’s host, either through a serial cable directly or through modems on each end of the connection.

When using this connection type, there may be other modifications and entries that need to be made.

Socket Connection Type

The Socket connection type is used if the data will be read from a TCP socket connection. You must specify the IP address and TCP port number of the remote host in the Rx/Tx Value box. The IP address and TCP port number must be separated by a colon. For example, to specify the SMTP port, enter the following in the Rx/Tx Value box:

```
<IP Address>:<port>
```

You may also need to place a TCP socket server process on the remote host and configure it to serve data to the TCP port.

For more information about configuring Collectors with this connection type, see the Collector documentation (such as Collectors Snort, Cisco PIX and Solaris Syslog) located in

```
%workbench_home%\elements\<Collector name>\docs
```

File New Connection Type

The File New connection type is used to read only security event data that is added to a file after the script has started. File New opens this file and reads from the end of the file. You must specify in the Rx/Tx Value box the path to the log file.

For more information about configuring Collector with this connection type, see the Collector documentation (such as Collector Solaris Syslog) located in

```
%workbench_home%\elements\<Collector name>\docs
```

File All Connection Type

The File All connection type is used to read all the security event data in a file.

If you select File New or File All, you can enter either inputfile or outputfile for the Rx/Tx Value. The format is as follows:

```
inputfile, outputfile
```

or

```
inputfile
```

or

```
outputfile
```

If you select File New or File All and the file shrinks, the file is read from the beginning.

For more information about configuring Collectors with this connection type, see the Collector documentation (such as Collectors Solaris Syslog and Windows 2000 Event Log) located in

```
%workbench_home%\elements\<Collector name>\docs
```

Persistent Process Connection Type

The Persistent Process connection type is used to launch a persistent process when the port is started. The process communicates between the Collector assigned to that port and an external application through receive and transmit states.

A Persistent Process starts at the first read/write state and continues to run for the active life of the port. The process is terminated by the port as part of the port’s shutdown process. When the port stops, a level 5 event is sent. When the port is started, a level 1 event is sent.

For more information, go to section Persistent and Transient Processes. For information about configuring the Rx/Tx Value for this connection type, go to section Configuring Rx/Tx Value for Persistent and Transient Connection (Rx/Tx Type). For more information about configuring Collectors with a persistent connection type, see the Collector documentation (such as the Collector Check Point Firewall & VPN) located in

```
%workbench_home%\elements\<Collector name>\docs
```

Transient Process Connection Type

The Transient Process connection type is used to launch a transient process when the port is started. The process communicates between the Collector assigned to that port and an external application through receive and transmit states.

A transient process may be started multiple times. The process is terminated by the port as part of the port's shutdown process.

NOTE: If you select Persistent Process or Transient Process, the Rx/Tx Value must include the path and the filename of the process to execute. You can use either the full path and filename or a relative (to %WORKBENCH_HOME%) path and filename. For example:

Full path:

```
C:\Program Files\Cisco\Csids_client – start
```

Relative Path:

```
.\elements\Cisco\Csids_client – start
```

For persistence process, the process will assume relative unless Rx/Tx Value is entered as a full path.

Transient Process Termination - If the Transient Process stops prior to parser termination, it is restarted on the next read or write state with no event being sent.

For more information, go to section Persistent and Transient Processes. For information about configuring the Rx/Tx Value for this connection type, go to section Configuring Rx/Tx Value for Persistent and Transient Connection (Rx/Tx Type).

SNMP Trap Connection Type

The SNMP Trap connection type is used to receive SNMP v1, v2 and v3 traps. These traps are sent by sensors to the Wizard server IP address. Based on the IP address and the object identifier (OID) of the sending device, the Collector Manager enables parsing through the appropriate Collector. The Rx (parsing) state relays inbound SNMP trap data to the Collector.

The information you use to collect and parse SNMP v1 and v3 traps are all configurable:

- SNMP v1 traps are identified using the IP address and the object identifier (OID), along with a trap code.
- SNMP v2/v3 traps are identified using the IP address, security name, engine ID, authentication and encryption keys (if enabled in the trap) and the trap's object identifier (OID).

The original format of the trap, in terms of trap values, is maintained as closely as possible. The format is usually defined in the MIB (management information base) for the sensor that originated the trap.

For more information, see SNMP Trap Setup.

None Connection Type

The None Connection Type is used without a communication port. It is more efficient because it does not try to connect. This type of connection should be used if an Collector does not use the Receive State and just processes commands.

For more detailed information about setting up Collectors with a None connection type, see the Collector documentation (such as Collectors ISS RealSecure and ISS SiteProtector) located in

```
%workbench_home%\elements\<Collector name>\docs
```

Creating, Assigning, Starting and Stopping a Wizard Port

Creating a Wizard Port

1. Refer to the Collector documentation located %workbench_home%\elements\<Collector name>\docs for Collector configuration information.
2. Click *Collectors* tab and select a Collector.
3. In Collector Builder, click the Wizard Hosts tab and select a host.
4. In the Port Information panel on the right, double-click *new*, type the name of the port and press Enter.
5. Select an *Rx/Tx Type*.
6. Specify setup options based on the connection type selected:
   - For Serial and Socket connections - In the Port Name box, right-click the port name and select *Edit Rx/Tx Value*. Specify one of the following sets of options:
     - For Serial connections - Select the Baud Rate, Word Size, Parity and Stop Bits. Click OK.
     - For Socket connections - Enter the IP Address and Port Number for the host machine, separated by a colon. If no receive state will be used, set the type to None and click *OK*.
   - For all other connection types - Double-click the *Rx/Tx Value* cell, enter the appropriate information and press Enter.
   - For SNMP Trap connection type, see SNMP Trap Setup.
7. Double-click the Collector cell and select a Collector name.
8. Right-click the *Port Name* and click *Other Port Options*. The Other Port Options dialog box displays.
9. In the *Other Port Options* dialog box, check or clear the *Run Port at Startup c*heck box, select a *Startup Sequence* and click *OK.*
10. If you are creating a port for the local host - Click *File* > *Save* and select *Port* Information.

    If you are creating a port for a remote host - Click *File* > *Upload/Download*.

    The port is added to the Port Information panel. You do not need to restart the system to implement the new port. Click *Start* to change the status of the new port from Off to On.

Persistent and Transient Processes

Using the Persistent Process or the Transient Process, Wizard is able to interface to another application using scripts that receive or transmit data and parse responses. Each of these scripts runs on a separate port and each port is connected to a specific application.

NOTE: Other Application is specified in the Rx/Tx Value box.

Process names can include the following items:

- Spaces
- Forward and backward slashes (to accommodate various operating systems)
- Command arguments
- Absolute and relative paths (the WORKBENCH_HOME environment variable is considered relative HOME)

When a receive/transmit (Rx/Tx) state occurs, the process specified in the Rx/Tx Value box is launched. When the parser terminates, the process terminates.

When a persistent process terminates, a level 5 event will be sent. When a persistent process starts, a level 1 event will be sent.

The standard output (stdout) of the Persistent/Transient Process is connected to the parser's receive "read" state. The standard input (stdin) of the Persistent/Transient Process is connected to the parser's transmit "write" state.

Configuring Rx/Tx Value for Persistent and Transient Connection (Rx/Tx Type)

There are three connector processes available when configuring Persistent and Transient Connections. They are:

- DBConnector (JDBC process connector)
- Lea Client
- Remote Data Exchange Protocol (RDEP)

Do not use quote marks in the Rx/Tx Value box for Persistent and Transient processes. If the process is an absolute path to a long executable name with spaces, then enter it without quotes. For example:

```
%WORKBENCH_HOME%\elements\checkpoint\lea_client
   checkpoint\lea_client.conf –new
```

Do not use spaces in arguments to the executable in the Rx/Tx Value box. These arguments are space delimited, so if they contain spaces the software will assume two arguments where there is only one. If the arguments are passing in the location of a configuration file, such as for Check Point, use a relative path from %WORKBENCH_HOME%. For example:

```
checkpoint/\lea_client checkpoint/\lea_client.conf -new
```

DBConnector

DBConnector (a JDBC process connector) runs a client that connects to a database server, executes an SQL query on that database and returns the result to standard output in name-value-pair format. The SQL query to execute is read from either standard input or a file. The name in the name-value-pair result is pulled from the column name of the result set. Due to this, the desired column name should be explicitly stated in the SQL. The exact syntax varies by database server.

This application is installed with Collector Manager in $WORKBENCH_HOME/dbconnector.

For more information about using DBConnector, see the README file that is found alongside the application, Sentinel Collector documentation for Entercept Host IDS 4.0 (via JDBC).

Lea Client

Sentinel's lea_client uses OPSEC's Log Export API to pull data from Check Point Firewall-1 and output it in name-value-pair format. The lea_client is usually used to feed data to Sentinel's Check Point Firewall-1 Collector, where the data is normalized and, depending on the action of the event (for example, drop, reject or accept), an alert is sent to the Sentinel server.

This application is installed with Collector Manager in $WORKBENCH_HOME/checkpoint.

For more information about Check Point lea_client, see the README file that is found alongside the application, Sentinel Collector documentation for Check Point Firewall & VPN Collector (via OPSEC).

Remote Data Exchange Protocol (RDEP)

The rdep_client, a Java application, pulls data from remote Cisco IDS v4.0 sensors running RDEP. The rdep_client connects to the remote IDS sensor using an HTTP or HTTPS connection. After the client connects, it opens a subscription or uses a previously opened subscription. The subscription describes the type of data the IDS sensor will send to the client. The type of data a new subscription will retrieve can be modified by editing the rdep_client configuration file. Using the subscription, the client initiates a request for event data from the IDS sensor. The event data is returned by the IDS sensor in XML format, converted into name-value-pairs by the Sentinel RDEP client, then parsed and normalized by the Collector. The Collector then sends the normalized event on to the Sentinel.

This application is installed with Collector Manager in $WORKBENCH_HOME/cisco/rdep_client.

For more information about RDEP, see the README file that is found alonsgide the application, Sentinel Collector documentation for Cisco IDS 4.0 Collector (via RDEP).

SNMP Trap Setup

Sentinel is able to receive SNMP traps representing security events that have occurred from a sensor on a network. These events are sent to Sentinel over a network using SNMP protocol. SNMP v1, v2 and v3 are supported. To enable Sentinel to receive SNMP traps a Wizard Collector must be created that uses the SNMP Trap connection (Rx/Tx) type.

You can configure the SNMP trap settings to specify the parameters that will allow Wizard SNMP Collectors to pass traps on to Sentinel as binary events.

The SNMP Trap Setting window is used to configure settings for Wizard SNMP Collectors, including the port used for SNMP traps, trap codes, authentication and encryption information.

How to access the SNMP Trap Window

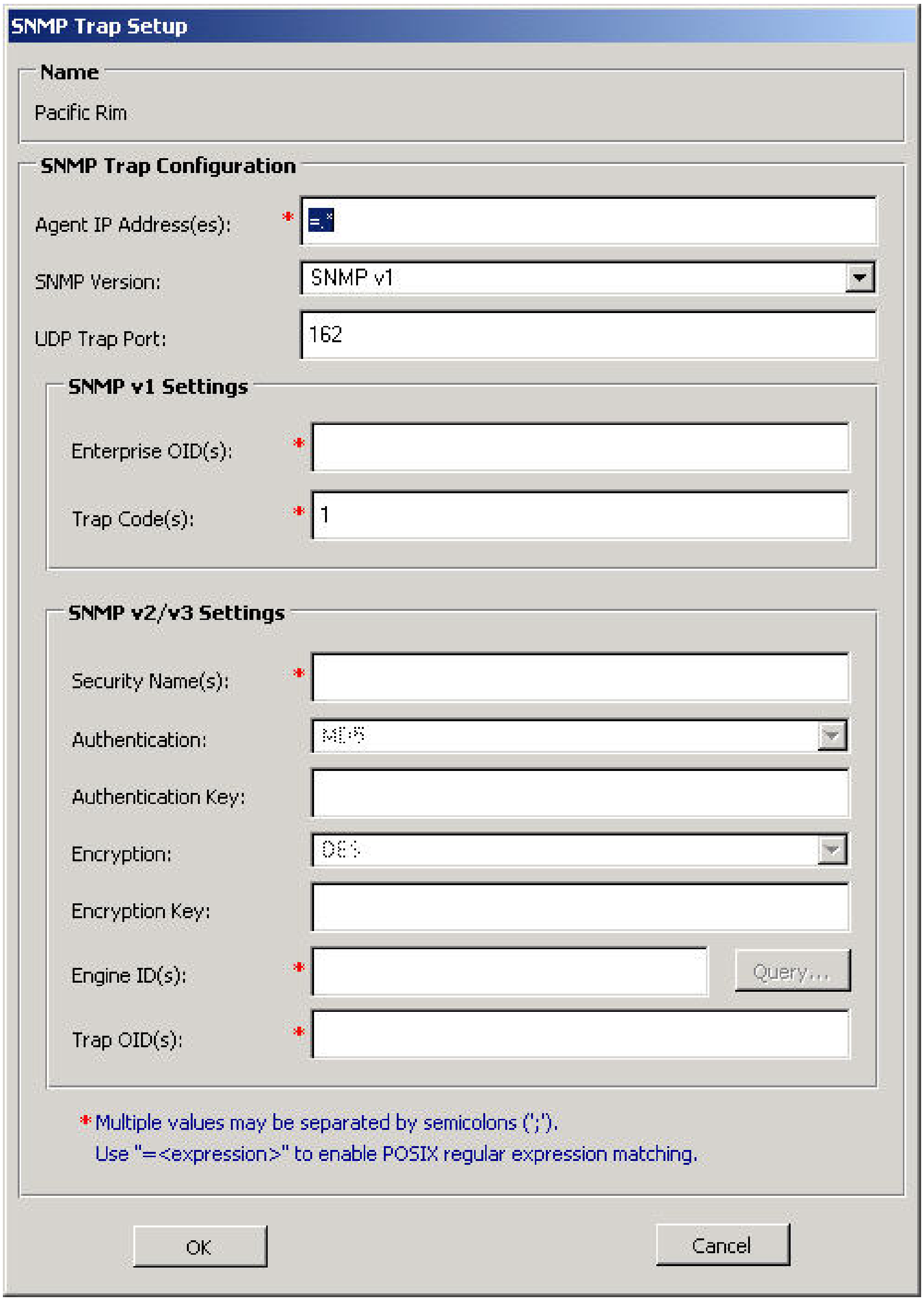
1. Within Collector Builder, assign a port name to your SNMP Collector.
2. For Rx/Tx Type select SNMP Trap.
3. Right-click on the port name, select *Edit Rx/Tx Value*.
4. Enter your SNMP information.

   NOTE: Default UDP trap port is 162. Ensure that that this port is available. If not you may choose another port number.

   NOTE: Unlike other Collector Ports, the Rx/Tx Value field will populate as per your settings this the SNMP Trap Setup window. For an SNMP Collector, you cannot manually edit the Rx/Tx Value field.

5. Saved and uploaded your SNMP Collector.
6. Activate this Collector by stopping and restarting your Collector Manager.

   NOTE: To activate this Collector, you must stop and restart your Collector Manager as stated in step 6.



SNMP Trap Setup

Name

Pacific Rim

SNMP Trap Configuration

Agent IP Address(es): * =.*

SNMP Version: SNMP v1

UDP Trap Port: 162

SNMP v1 Settings

Enterprise OID(s): *

Trap Code(s): * 1

SNMP v2/v3 Settings

Security Name(s): *

Authentication: MD5

Authentication Key:

Encryption: DES

Encryption Key:

Engine ID(s): *

Trap OID(s): *

* Multiple values may be separated by semicolons (';').
Use "=<expression>" to enable POSIX regular expression matching.

SNMP Setup consists of:

- Collector IP Address(es)
- SNMP Version
- UDP Trap Port
- SNMP v1 Settings
  - Enterprise OID(s)
  - Trap Code(s)

- SNMP v2/v3 Settings
    - Security Name(s)
    - Authentication
    - Authentication Key
    - Encryption
    - Encryption Key
    - Engine ID(s) with Query Button
    - Trap OID(s)

In the SNMP Trap Setup dialog box (opened by right-clicking a port name on Collector Builder’s Port Information panel and then clicking Edit Rx/Tx Value), you can configure Wizard to:

- Receive traps on ports other than UDP port 162 (the default).
- Create a single Wizard parsing script to process traps from multipleIP addresses, with such information as multiple trap codes and multiple trap object identifiers (OIDs).
- Allow for POSIX regular expression matching on IP address, enterprise object identifier (OID), trap code and trap OID fields.
- After decoding the trap, Wizard sets values for variables included in the script.

Collector IP Address(es)

Collector IP Address(es) are IP addresses you want to receive traps. Separate multiple values with semicolons (;). You can use the format =<expression> to match POSIX-compliant regular expressions. The asterisk (*) is a modifier of the preceding character or expression and the period (.) may be used as a wild card character and may appear anywhere in the string if using regular expressions.

The most common regular expressions that you are likely to use are:

`=`	matches any sequence of characters of any length
`= 192\.168.*`	matches any sequence of characters that contains 192.168 To get “begins with behavior,” use: ^192.168... With ^ being the beginning-of-line anchor. To get “ends with behavior,” use 0.47$... With $ being the end-of-line anchor
`= [abc]`	matches a or b or c
`= [a-zA-Z0-9]`	matches any single character in the alphabet (uppercase or lowercase) or any digit from 0 to 9

Basically, the rules in the above examples of common regular expressions are:

`.`	matches any single character
`*`	matches zero or more occurrences of the preceding pattern
`[]`	matches any single character from the pattern defined in the brackets

NOTE: These rules can be combined.

SNMP Version

Only one SNMP version can be configured. The options in the SNMP v1 Settings and SNMP v2/v3 Settings panes are enabled based on which version you select.

UDP Trap Port

The UDP Port destination port default is 162.

SNMP v1 Settings

These settings are enabled only if you select SNMP v1 from the SNMP Version list.

- Enterprise OID(s) - Object ID(s) used to identify the type of Collector that sent the trap. Separate multiple values with semicolons (;).
- Trap Code(s) - Trap codes for sensors sending the SNMP traps. These trap codes represent the types of trap sent by the given SNMP Collector. Separate multiple values with semicolons (;).

SNMP v2/v3 Settings

- Security Name(s) - User name used to access the Collector. Security names are case-sensitive. Separate multiple values with semicolons (;).
- Authentication - Authentication method. Values are:
  - None - No authentication is performed on SNMP v3 traps.
  - MD5 - Security Name is configured to use the MD5 algorithm to create a digital signature for authentication.
- Authentication Key - Password used to authenticate the user on the Collector. Enabled only if Authentication is MD5. Must be at least eight characters long. Authentication keys are case-sensitive. The same key must be configured on the sending SNMP Collector.
- Encryption - Encryption method. Values are:
  - None - No encryption is performed on SNMP v3 traps.
  - DES - Expects to receive traps encrypted with the DES (Data Encryption Standard) encryption method.
- Encryption Key - Key used to decrypt traps sent to Wizard Collectors. Must be at least eight characters long. The encryption key is case-sensitive. Enabled only is DES is selected in the Encryption list.
- Engine ID(s) - A unique identifier for an SNMP v3 Collector. There is an Engine ID Query Button that finds the IP address that you want to query. A successful query returns back the information and adds the engine ID. If you have an engine ID in the box, it appends a new one.
- Trap OID(s) - The trap’s object ID that identifies the specific type of trap received.

NOTE: If you specify multiple security names and engine IDs, the same authentication and encryption scheme is used for all.

NOTE: If different authentication and encryption keys are needed for different SNMP Collectors, then separate ports must be configured for each.

SNMP Trap Variables

Some trap variables are valid for all traps (SNMP v1 and v3) and some are valid only for one version. The following tables list all the SNMP trap variables, grouped by the SNMP version that they work with:

- SNMP Trap Variables for SNMP v1 and v3
- SNMP Trap Variables for SNMP v1
- SNMP Trap Variables for SNMP v3

SNMP Trap Variables for SNMP v1 and v3

Variable	Description
`s_Trap_IP`	IP address of the Collector/sensor that sent the trap.
`s_Trap_Time`	Uptime value reported by the Collector/sensor that sent the trap. Usually, this is the amount of time the Collector has been running. Format: D:HH:MM:SS.ss (days, hours, minutes, seconds, hundredths of a second).
`i_Trap_Version`	Value for a specific SNMP version: 1 = SNMP v1 3 = SNMP v3
`i_Trap_Vars`	Number of variable bindings in the trap.
`s_Trap_OID[]`	An array (of size “i_Trap_Vars”) of the names of the MIB variables bound in the trap message. Each element of the s_Trap_OID array is an OID, for example “.1.3.6.1.4.1.4286....”
`s_Trap_Value[]`	An array (of size “i_Trap_vars”) of the values of the MIB variables bound in the trap message. The indices of this array and the s_Trap_OID array match up with each other, such that s_Trap_OID[0] is the name and s_Trap_Value[0] is the value.

SNMP Trap Variables for SNMP v1

Variable	Description
`s_Trap_Ent`	Enterprise object identifier (OID) of the Collector/sensor that sent the trap.
`s_Trap_Code_Generic`	Generic trap code of the trap. Values are: 1-5 = standard, IETF-defined (Internet Engineering Task Force) trap types 6 = enterprise-specific trap (code is defined in s_Trap_Code_Specific)
`s_Trap_Code_Specific`	Specific trap code of the trap. Only relevant if s_Trap_Code_Generic = 6.

SNMP Trap Variables for SNMP v3

Variable	**Description**
`s_Trap_Engine_ID`	Engine ID of the SNMP v3 Collector that sent the trap.
`s_Trap_OID`	Object Identifier (OID) that identifies the type of SNMP v3 trap received. For trap identification purposes, the SNMP v3 trap OID takes the place of the SNMP v1 enterprise OID and generic/specific trap codes.
`s_Trap_Security_Name`	Security name by which the SNMP v3 Collector that sent the trap is known.

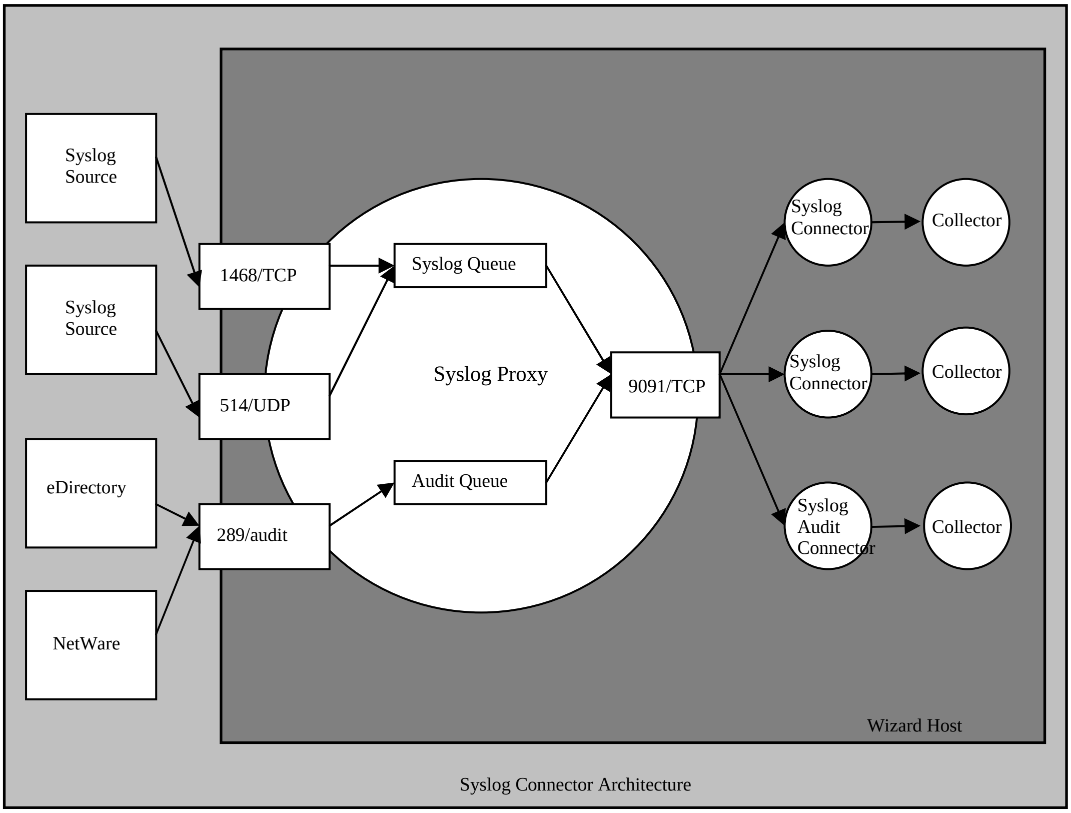
A Syslog Connector v5.1.3

NOTE: The term Agent is interchangeable with Collector. Going forward, Agents will be referred to as Collectors.

Novell has released this syslog connector in order to facilitate smooth integration between Sentinel Collectors and with those products able to generate syslog messages. This document is meant to explain the architecture, installation, usage and options of the syslog connector.

Architecture

The syslog connector is comprised of two parts. One part is the syslog proxy and the other is the syslog connector client. The syslog proxy listens to selected UDP and TCP ports. The UDP port by default is 514. The default TCP port is 1468, which is the port commonly used by Cisco PIX when sending syslog messages over the TCP protocol and 289, which is the port commonly used by the Novell Audit Platform Agent to send the messages.



The functions each syslog connector component performs are described below:

- Syslog proxy

    - Listens to either a TCP and/or UDP port for syslog messages.
    - Parses incoming message looking for syslog standard message components (Priority, Date, Hostname and Message)
    - In the event the message source sends a message missing Priority, Date, or Hostname, it follows the RFC 3164 "BSD Syslog Protocol" and inserts supplementary data.
    - Having determined the Facility and Level from the Priority as well as the Hostname, the proxy effectively publishes the message to those syslog connector sessions interested in the message.
    - In the event the syslog connector client session terminates, the syslog proxy will queue incoming messages for that client for 10 minutes. This behavior is meant to ensure the Collector does not miss messages while it is being restarted or temporarily stopped.
    - Syslog proxy listens on a TCP port, typically 9091, in order to service syslog connector client sessions.
- Syslog connector client
    - Connector is started as a Persistent Process with all the syslog connector runtime options entered into the RX/TX value.
    - One runtime parameter is the ID. The ID configured for a particular syslog connector must be unique among all syslog connectors connecting to the same syslog proxy.
    - A content filter can be specified at runtime in order to limit the scope of messages submitted to the Collector for parsing.
    - The syslog connector makes a connection to the proxy's connector client service.
    - The syslog connector register's its ID and content filter to the syslog proxy.
    - Messages the syslog proxy associates to the ID are read by the syslog connector and directed to its standard output.
    - Currently the structure and content of the message is passed to the Collector as is. In the future, the syslog connector will be able to format the message in order to comply with the Collector's parsing requirements.

The syslog protocol has traditionally and is currently defined as a UDP based protocol. In the absence of a diverse population of applications/devices capable of sending messages over TCP or a recognized standard for syslog over TCP, the Cisco PIX approach to syslog message termination (carriage return + line feed) was adopted. Message termination is necessary for syslog over TCP, since there is no standard defined or natural boundary between messages. Syslog over UDP has natural message termination, since a UDP packet transports a single message and UDP is connection-less.

Configuration of Audit Client

The client machine contains a file named logevent.cfg, which should be configured to point to the machine running the syslog proxy. The value for “LogHost” should be the ipaddress of the machine that has the syslog proxy running and the “LogEnginePort” should be the port its listening for the audit messages (the port by default is 289).

For example: If the syslog is running on a machine 172.16.1.1 and the audit port being 289 the logevent.cfg file should look like:

```
LogHost=172.16.1.1

LogEnginePort=289
```

On Windows machine the file is located in the folder %WINDIR%

On Netware machine the file is located in the folder SYS: /ETC/

On Linux machine the file is located in the folder /etc/

Additional Files required for audit

Installing the product will copy two files naudit.conf and nudit.keystore. The naudit.keystore contains the certificate of the proxy server which is used during the SSL handshake with the audit clients. The naudit.conf file contains configuration properties for the audit part in the syslog.

Event filtering for audit

The events coming from the platform agent can be filtered by creating event configuration xml files. The location of the xml files is specified by the “FilterDir” property in the naudit.conf file. The name of the xml files is the name of the application with an .xml extension.

For example: If the name of the application is DirXML the event configuration xml file name should be DirXML.xml location in the location pointed by the “FilterDir” property in the naudit.conf file. If the application name contains spaces they are replaced by the character underscore (_). If the name of the application is Modular Authentication Service the event configuration xml file name should be Modular_Authentication_Service.xml.

Sample event configuration xml file

```
<NAuditLogAppConfiguration id="0033">
<EventConfiguration enabled="true">
 <event>00330042</event>
 <event>00330002</event>
 <event>00330023</event>
 </EventConfiguration>
 </NAuditLogAppConfiguration>
```

If “EventConfiguration” enabled is set to true only the events specified in the xml file are sent to the syslog.

Events can also be configured using iManager using the following steps.

- After logging into the iManager, select the “Logging Server Options” in the Auditing and Logging section. Then select appropriate logging server object from the “Logging Services” tree.
- In the “Log Applications” section select (click on) the application to be configured.
- Then navigate to the Configuration > Events section and select all the events that should be sent to the syslog.

The above steps will create an xml files and get stored in eDirectory. To view the xml files containing the event configurations, go to the General > Valued attributes Section in the application window and select the “NAuditConfiguration”. This attribute contains the event configuration xml data.

NOTE: Some applications do not do event filtering so they send all the events to the syslog even when the event configuration is set properly.

Installing and Uninstalling

The syslog connector was designed to operate on any Wizard platform. Due to this portability requirement, both components are written in Java. The necessary software and hardware requirements are listed below.

System Requirements

Software

- Java 1.4.1 or higher
- Wizard 4.2 or higher
- Windows (2000/XP/2003), Solaris (8/9), RedHat Enterprise Linux (v3 ES/AS)

Hardware

- Additional 14 MB of RAM (45 MB virtual memory) for each instance of syslog connector and the proxy

Installation

Both the syslog proxy and connector client files are automatically installed when the Collector Service is installed. The syslog files are in the directory:

For UNIX:

```
$ESEC_HOME/wizard/syslog
```

For Windows:

```
%ESEC_HOME%\wizard\syslog
```

The Wizard will not start the syslog proxy automatically. If you wish to have the syslog proxy to start automatically, it must be installed as a service. Follow the following instructions to install the syslog proxy as a service.

Install as Windows Service (Windows)

NOTE: The syslog proxy can be installed as a Windows Service to run automatically. To install proxy to run automatically in windows set the property “wrapper.ntservice.starttype” to “AUTO_START” in the “syslog-Windows.conf” file in the %ESEC_HOME%\wizard\syslog\config directory. To install the syslog proxy as a service, execute the following commands at the command prompt:

1. cd /d “%ESEC_HOME%\wizard\syslog”
2. syslog-server.bat install

   This will create a Windows Service named “Sentinel Syslog Server”.

Install as service (UNIX)

NOTE: The syslog proxy can be installed as a service on UNIX so that it will run automatically when the machine starts. To install the syslog proxy as a service, execute the following commands:

1. Log in as root.
2. cd $ESEC_HOME/wizard/syslog
3. ./syslog-server.sh install

   This will cause the syslog proxy to be started automatically when the machine starts. By default, the syslog proxy will run as the root user. This is required because the syslog proxy will, by default, bind to port 514, which requires root privileges. To have the syslog proxy run as a user other than root, modify the script /etc/init.d/esyslogserver. You will need to make sure this user has privileges to bind to the port it will be listening for messages on. Some examples of how this can be done are:

   - Use the "sudo" command to start the syslog proxy, giving the user "sudo" privileges to bind to the required port.
   - Modify the syslog configuration (syslog.conf) and make the syslog proxy bind to a port that does not require root privileges (i.e. - >1024). In this case, you'll probably need to redirect messages sent to port 514 to the new port you've selected to use.

Uninstallation

To uninstall this Windows Service, execute the following commands at the command prompt.

Uninstall as Windows Service (Windows)

1. cd /d "%ESEC_HOME%\wizard\syslog"
2. syslog-server.bat remove

Uninstall as Service (UNIX)

To uninstall the syslog proxy as a service, execute the following commands:

1. Log in as root.
2. cd $ESEC_HOME/wizard/syslog
3. ./syslog-server.sh remove

Usage

Syslog Proxy Server

The Wizard will not startup the syslog proxy server automatically. If you wish to have the syslog proxy startup automatically, it must be installed as a service. Follow the instructions in the section Installation to install the syslog proxy as a service.

The syslog proxy configuration is stored in the file:

For UNIX:

```
$ESEC_HOME/wizard/syslog/config/syslog.conf
```

For Windows:

```
%ESEC_HOME%\wizard\syslog\config\syslog.conf
```

The syslog proxy is setup to use the following configuration by default:

- Listener on UDP port 514 for syslog messages

- Listener on TCP port 1468 for syslog messages
- Listener on TCP port 289 for audit message
- Listener on TCP port 9091 for connector connections

The syslog proxy can be configured to listen to other ports for either receiving syslog messages or accepting client connections. Those switches respectively are:

-udp <port>	port to listen for UDP messages from devices; default 514
-tcp <port>	port to listen for TCP connections from devices; default 1468
-audit <port>	port to listen for TCP connections from devices; default 289
-connector <port>	port to listen for TCP connections from connectors; default 9091

To edit these settings, modify the following section of the syslog.conf file:

```
wrapper.app.parameter.3=-tcp
wrapper.app.parameter.4=1468
wrapper.app.parameter.5=-udp
wrapper.app.parameter.6=514
wrapper.app.parameter.7=-connector
wrapper.app.parameter.8=9091
wrapper.app.parameter.11=-audit
wrapper.app.parameter.12=289
wrapper.app.parameter.13=-authentication
wrapper.app.parameter.14=open
wrapper.app.parameter.15=-auditQueueSize
wrapper.app.parameter.16=10000
```

For example, if you want to modify the port settings to the following:

- Listener on UDP port 4514 for syslog messages
- Listener on TCP port 4168 for syslog messages
- Listener on TCP port 4991 for connector connections

The section of syslog.conf posted above should be modified to:

```
wrapper.app.parameter.3=-tcp
wrapper.app.parameter.4=4168
wrapper.app.parameter.5=-udp
wrapper.app.parameter.6=4514
wrapper.app.parameter.7=-connector
wrapper.app.parameter.8=4991
```

By default, the syslog proxy configuration is setup to accept client connections from any host. To increase security, the syslog proxy can be setup to only accept client connections residing on the same host. This is a security precaution, since there is no privacy, access control or authentication between the client connectors and the proxy. The following switches address this:

`-private`	listen for connector connections on loopback

`-shared` listen for connector connections on the external IP, not local host

The –shared switch will instruct the proxy to bind the client connection listener to a socket accessible to remote hosts.

To edit these settings, modify the following section of the syslog.conf file:

```
wrapper.app.parameter.2=-shared
```

For example, to only allow client connections from the same host, you should modify the settings to the following:

```
wrapper.app.parameter.2=-private
```

The syslog proxy can be configured to record all received messages to a log file. The format of the messages will appear in the form the syslog proxy would use if it were to relay the messages on to another syslog server. As a result, the <PRI> or Priority used by the receiving syslog server to evaluate the messages' Facility and Level will be present at the beginning of each message. This kind of logging is enabled by the following switch.

`-log <filename>` Name of the log file to append to.

To enable this kind of logging, add the following two lines to the syslog.conf file after the last "wrapper.app.parameter":

```
wrapper.app.parameter.17=-log
wrapper.app.parameter.18=<filename>
```

For example, to enable this kind of logging to the file $ESEC_HOME/wizard/syslog/messages.log, you should modify the settings to the following:

```
wrapper.app.parameter.7=-connector
wrapper.app.parameter.8=9091
wrapper.app.parameter.9=-messageSize
wrapper.app.parameter.10=5000
wrapper.app.parameter.11=-audit
wrapper.app.parameter.12=289
wrapper.app.parameter.13=-authentication
wrapper.app.parameter.14=open
wrapper.app.parameter.15=-auditQueueSize
wrapper.app.parameter.16=10000
```

If an absolute path is not specified for the filename, the path will be relative to the directory $ESEC_HOME/wizard/syslog.

NOTE: The log file could become fairly large, so make sure the location where the file will be written has plenty of space (e.g. – a directory other than under $ESEC_HOME).

It is recommended that the syslog proxy be run with a minimum of 64MB and maximum of 256MB of JVM heap memory. With this configuration you can expect the following performance:

Proxy server limits:

▪ Max number of events:	500 eps (total for all client ports)
▪ Max connector Q size:	5000 messages (this is the default if none is specified)
▪ Max connectors:	5

To modify the memory settings, edit the following section of the syslog.conf file:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=64

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=256
```

Syslog connector client

The syslog connector client connects to the syslog proxy collecting the messages for which it subscribed. The messages collected by the client are then delivered to standard output. The client's session with the server does not end until the client process or the syslog proxy terminates. This operational and output behavior makes it suitable for use by the Collector engine as a Persistent Process connector.

In Collector Builder's Port Configuration window, configure a port with an Rx/Tx Type of Persistent Process and Rx/Tx Value similar to the generic syntax below.

For UNIX:

```
syslog/SyslogConnectorAgent.sh <arguments>
```

For Windows:

```
syslog\SyslogConnectorAgent.bat <arguments>
```

After having completed the Rx/Tx Value, select the appropriate Collector from the library and upload the port configuration and possibly the Collector as well to the remote Wizard.

The syslog connector client has been designed to use a number of default arguments in order to simplify general usage. The simplest command line for the syslog connector client would be:

For UNIX:

```
syslog/SyslogConnectorAgent.sh –id MyUniqueID
```

For Windows:

```
syslog\SyslogConnectorAgent.bat –id MyUniqueID
```

The interpretation of this command line is the following:

- Connect to the syslog proxy listening for this connection on 127.0.0.1:9091
- Subscribe to all messages sent with all possible syslog Facilities
- Subscribe to all messages sent with all possible syslog Levels
- Subscribe to all messages irrespective of the source IP address contained in the IP header.
- Subscribe to all messages irrespective of the host designation within the message.
- Assign these session subscription parameters the ID of 'MyUniqueID'

The syslog connector client session will be registered with the syslog proxy with the above subscription filter under the ID of 'MyUniqueID'. The ID is required. The ID chosen was arbitrary, but must be unique among all syslog connector client sessions with the same syslog proxy. If a different syslog connector client is configured with the same ID, one of the two connections with the same ID will be dropped. The last session to connect with the same ID will survive.

The generic filter in the previous filter may waste Collector-processing effort if the messages that meet the filter requirements, which are all received messages, are not relevant to this particular Collector’s operation. From the example above it should be apparent that the filter expression is very versatile. The following example, for UNIX, establishes a more restrictive albeit precise description of what messages are relevant to the Collector.

```
syslog/SyslogConnectorAgent.sh -facilities "user,kernel" -
   levels "warning,error" -sender
   "192.168.0.12,192.168.0.0/16" -host
   "172.16.8.0/24,127.0.1.13" -id "MyOtherUniqueID"
```

The interpretation of this command line is the following:

- Connect to the syslog proxy listening for this connection on 127.0.0.1:9091
- (-facilities) Subscribe to all messages sent with Facilities of user or kernel
- (-levels) Subscribe to all messages sent with Levels of warning or error
- (-sender) Subscribe to messages identified by the source IP address of the incoming messages to the syslog proxy. This argument has the syslog proxy look at the IP header information in order to evaluate these criteria. This allows the filter to accommodate syslog relay servers, relay servers do not identify themselves in the messages they relay. Although this argument was designed to accommodate relayed messages, this argument could be used to filter messages sent directly from the syslog source. In particular the relay servers or syslog sources of interest are 192.168.0.12 and 192.168.0.0/16. The second item actually represents a range of IP addresses; so long as the source IP address is within 192.168.0.0 and 192.168.255.255 those messages pass the filter criteria. Hostnames are not valid for no hostname resolution is performed to determine the hostnames of source IP addresses.
- (-host) Subscribe to messages that contain host designators 172.16.8.0/24 within the syslog message. The first item is an IP address range. If the message contains a host designator in IP address form and is within the range of 172.16.8.0 through 172.16.8.255, then the message passes this condition within the filter. Hostnames are supported by the –host argument. The hostname can either be designated literally or by regular expression. Keep in mind that no hostname resolution is performed for this argument either. One cannot expect that configuring either a hostname or IP address will result in the filter accommodating the opposite naming schema. For example, configuring –host 172.16.0.90 will not result in a filter match for a message containing the hostname of “testbox1” even though name resolution services would have mapped 172.16.0.90 to “testbox1”. So IP host designation will only match IP addresses and hostname host designation will only match hostnames.

The filter from the example above can be described in the following Boolean expression:

```
(Facility=”user” or Facility=”kernel”) and
   (Level=”warning” or Level=”error”) and
   (Sender=”192.168.0.12” or Sender=192.168.0.0/16”) and
   (Host=”172.16.8.0/24”)
```

The number of possible combinations of these arguments is the Cartesian product of argument types, where each argument type is a set. According to PRINCIPIA CYBERNETICA WEB (http://pespmc1.vub.ac.be/ASC/CARTES_PRODU.html) the Cartesian product is:

> “The collection of all ordered n-tuples that can be formed so that they contain one element of the first set, one element of the second,... and one element of the n-th set. This collection can be seen as constituting an n-dimensional space in which each n-tuple designates a cell. The simplest Cartesian product of two sets is a two-dimensional table or a cross-tabulation whose cells may be used to enter frequencies, to designate possibilities (see relation) or impossibilities (see constraint), or to chart the transitions comprising the behavior of a system. (Krippendorff)”

NOTE: As of the date of publication of this document, the above mentioned website was correct.

This implies that in theory quite a few distinct messages could pass this filter. Only practical operational conditions will truly dictate the number of distinct messages.

In addition to filter command line arguments, there are also the following optional command line arguments:

`-proxy <server_address>:<port#>`	The syslog server proxy host address and port number to connect to.
`-log <filename>`	Enables logging to the specified file.

The –proxy argument is used to configure the connector client to connect to either a non-default TCP port or host other than localhost. The syslog proxy expects a connector client connection is on 9091 by default. In the event that 9091 is not suitable for the host on which the syslog proxy is running, the port can be adjusted during syslog proxy startup and by using the –proxy argument the clients can be instructed to connect to this alternate port. Additionally, the target host of the connector client can be specified to be a host other than the local system. In the event a syslog proxy is accepting remote connector client sessions, a syslog connector client can be configured to establish a session with that remote syslog proxy. The IP address and connector client port of the syslog proxy would be configured with the –proxy argument.

The –log argument enables the logging feature of the connector client. The connector client will write out messages as it receives them from the syslog proxy. Unlike the syslog proxy log file, the message content will be filtered per the registered subscription details as well as each message logged will not contain the <PRI> or Priority field. The content will be consistent with what the Collector receives from the same syslog connector client.

The –retry argument enables the client to connect to the syslog proxy in case the proxy goes down. The client waits a default time of 1 minute to connect to the proxy; this time can be overridden with the –retry argument.

NOTE: The log file could become fairly large, so make sure the location where the file will be written has plenty of space (e.g. – a directory other than under $ESEC_HOME).

An example, for UNIX, of using the –proxy and –log arguments is:

```
syslog/SyslogConnectorAgent.sh –proxy localhost:9091 –log
   connector_messages.log –id 'MyUniqueID'
```

Configuring Logging for the Syslog Proxy Server

The Syslog Proxy server prints logging messages to the file:

```
$ESEC_HOME/wizard/syslog/syslog_trace*.*.log
```

The logging levels can be modified by editing the logging properties file:

```
$ESEC_HOME/wizard/syslog/syslog_log.prop
```

This is the logging properties file as specified by the following line in the syslog.conf file:

```
wrapper.java.additional.1=-
   Djava.util.logging.config.file=syslog_log.prop
```

Make modifications to the following section to adjust logging levels:

```
###### Configure the logging levels
# Logging level rules are read from the top down.  Start
   with the most general, then get more specific.
…
######
```

Sample Command Line Arguments

It is possible to run the Syslog Proxy server and client connector without using the scripts provided with the installation. To do this, you'll need to use the command line arguments found in this section.

Syslog proxy:

```
java –server –Xms64m –Xmx256m –
   Djava.util.logging.config.file=syslog-logger.prop -
   Dsentinel.audit.configuration=config/naudit.conf -
   Dsentinel.audit.keystore=config/naudit.keystore -
   Dsentinel.audit.password=star1111 -jar syslog.jar [-udp
   <port>] [-tcp <port>] [-audit <port>] [-connector
   <port>] [-private|-shared] [-log <file path>] [-
   messageSize <number>] [-authentication
   {open|loose|normal|strict}] [-auditQueueSize <number>]
```

Valid arguments:

`–server`	Always should be used. Used by the JVM
`–Xms64m`	This specifies the initial memory size of the syslog proxy. We recommend 64 megabytes.
`–Xmx256m`	This specifies the maximum memory size of the syslog proxy. Our recommended default is 256 megabytes.

	This allows the proxy server to handle spikes in data volumes, multiple client connectors and handle buffers if the connectors reconnect. This value can be changes to a higher number if there is available memory and data volumes and the number of clients connectors that connect. This should not exceed 1.2 Gigabytes per syslog proxy server, i.e. '–Xmx1200m'
`–Djava.util.logging.config.file`	This property specifies the name of the debug logging config path/file. So it needs to point to where thew file exists. If no path is specified it will look in the current directory from where the JVM was run. Example: %workbench_home%\syslog-logger.prop
`-Dsentinel.audit.configuration`	This property specifies the name and location of the audit proxy configuration file.
`-Dsentinel.audit.keystore`	This property points out to the location of the keystore containing the audit proxy server certificate
`-Dsentinel.audit.password`	This property provides the key for the certificate.
`-udp <port>`	port to listen for UDP messages from devices, default 514
`-tcp <port>`	port to listen for TCP connections from devices, default 1468
`-audit <port>`	port to listen for TCP connections from devices; default 289
`-connector <port>`	port to listen for TCP connections from connectors, default 9091
`-private`	listen for connector connections on loopback, is default
`-authentication`	Type of authentication to use when connecting to the audit clients. The acceptable values are open – Trust any client. loose - Trust any valid cert including self-signed certificates. normal - Any certificate that supplies valid chain of trust. strict - Ensure client has same name as cert DN.

	The default value is “open” for the server. This option is valid only for the audit connections.
`-shared`	Listen for connector connections on the externl IP. If this isn't set, a communication error will generate.
`-log`	Name of a log file to append to
`-help`	Provide this help message
`-version`	Output the version of the proxy (0.91-poc)
`-messageSize`	Number of messages buffered to send again for the temporarily lost connections. If the option value is not used or if the option value is less than 0, command will default to 5000. This option is used for non audit clients.
`-auditQueueSize`	Number of messages buffered by the syslog for audit connections if none of clients are currently active. By default the value is set to 10000 if not overridden by this parameter. This option is not used for non audit clients.

Syslog connector client:

```
java -jar syslogconnector.jar -id <UniqueId> [-proxy
   <host:port number>] [-facilities
   <facility1,facility2,…>] [-levels <level1, level2,…>]
   [-sender <Source IP1[/integer subnet mask], Source
   IP2[/integer subnet mask],…>] [-host < IP1[/integer
   subnet mask]|Hostname1 | Hostname Regex1, IP2[/integer
   subnet mask]|Hostname2 | Hostname Regex2, …>] [-log
   <file path to log file>] [-retry <number>] [-audit] [-
   body <text to search for in the event>]
```

Valid Arguments:

`-proxy <host:port number>`	The Syslog proxy to connect to host:port, default is 127.0.0.1:9091
`-facilities <facility1,facility2,…>`	Comma separated list of desired facilities, default is all facilities
`-levels <level1, level2,…>`	Comma separated list of desired severities, default is all levels

-sender <Source IP1[/integer subnet mask], Source IP2[/integer subnet mask],…>	Comma separated list of desired senders, default is all senders
-host < IP1[/integer subnet mask]|Hostname1 | Hostname Regex1, IP2[/integer subnet mask	Comma separated list of desired hosts, default is all host
-log <file path to log file>	Name of a log file to append to
-id <UniqueId>	Specify connector identity (REQUIRED)
-help	Provide this help message
-audit	Configures the client to accept the binary audit events and parses them to NVP pair. This option is valid only when listening for audit messages from proxy.
-body	Regular Expression string found in the desired message bodies. This option is not valid for audit messages.
-version	Output the version of the connector (0.91-poc)
-retry	Time in milliseconds the client waits before attempting to reconnect to the proxy. Default value is 1 minute.

Table of Supported Facilities

Facility names are case insensitive when specified within the syslog connector client command line.

KERNEL
USER
MAIL
DAEMON
AUTH
SYSLOG
LPR
NEWS

UUCP
CRON
SECURITY
FTP DAEMON
NTP
LOG AUDIT
LOG ALERT
CLOCK DAEMON

LOCAL0
LOCAL1
LOCAL2
LOCAL3
LOCAL4
LOCAL5
LOCAL6
LOCAL7

Table of Supported Levels

Level names are case insensitive when specified within the syslog connector client command line.

EMERGENCY
ALERT
CRITICAL
ERROR

WARNING
NOTICE
INFORMATIONAL
DEBUG

Deployment Notes

Relayed Messages to syslog Proxy

Most syslog servers are able to redirect syslog messages they receive to an alternate syslog server as well as process the incoming messages. In a deployment scenario it may be attractive to alter an existing log host to provide message relaying to the syslog proxy. There are unfortunate behaviors among some syslog servers that may make this a poor deployment choice.

It has been observed that Solaris 7, 9 and Linux 8 (which may be representative of other deployed versions) syslog server libraries do not place the hostname or IP address of the host in the messages they send off host. The receiving syslog server associates the source IP address or hostname (via name resolution) to the received messages in the log files it generates. In the case of the Solaris 9 acting as a relay to our proxy, it does not populate the messages it forwards to the proxy with either the IP address or hostname of the original message source. This is odd, since the log file on the Solaris 9 system shows an IP address or hostname. Without having the supplemental hostname in the message, the syslog proxy is forced to deduce the message originated from the relay server not the original host. The syslog proxy will supplement the message with IP address of the relay host in each message it receives from a Solaris 9 relay. The consequences of this are serious. The origin of a security event is not visible to the Collector and therefore the Sentinel solution.

It is strongly encouraged that the proxy not be a recipient of relayed messages if the messages do not contain the IP address or hostname of the true origin. This recommendation may have significant logistical consequences if the proxy were to be used in production.

Example:

An su event takes place on ultrabookIIi (172.16.0.70) running Solaris 7, which is forwarding syslog messages to talkabout (172.16.0.72) running Solaris 9, which in turn is relaying to the syslog proxy. The following are messages the Sentinel connector generated.

Proxy:

```
<37>Apr 02 06:54:11 [172.16.0.72.151.234] su: 'su root'
   succeeded for oespadm on /dev/pts/0
```

Connector Client:

```
Apr 02 06:54:11 [172.16.0.72.151.234] su: 'su root'
   succeeded for oespadm on /dev/pts/0
```

Below is the packet trace of the same message first arriving at talkabout and then relayed to syslog proxy on pes020.esecurity.net.

```
# snoop -x0 udp port 514
Using device /dev/dmfe0 (promiscuous mode)
ultrabookIIi -> talkabout    SYSLOG C port=42830 <37>Apr
   1 18:54:11

0: 0000 83cd 1395 0040 2082 202b 0800 4500     .......@ .
   +..E.
```

```
16: 0061 fa09 4000 ff11 28d3 ac10 0046 ac10
   .aú.@...(....F..
32: 0048 a74e 0202 004d 5d7e 3c33 373e 4170
   .H.N...M]~<37>Ap
48: 7220 2031 2031 383a 3534 3a31 3120 7375    r  1
   18:54:11 su
64: 3a20 2773 7520 726f 6f74 2720 7375 6363    : 'su root'
   succ
80: 6565 6465 6420 666f 7220 6f65 7370 6164    eeded for
   oespad
96: 6d20 6f6e 202f 6465 762f 7074 732f 30      m on
   /dev/pts/0

talkabout -> pes020.esecurity.net SYSLOG C port=38890
   <37>Apr  1 18:54:11

0: 000a 5e02 a335 0000 83cd 1395 0800 4500
   ..^..5........E.
16: 0061 304b 4000 ff11 f031 ac10 0048 ac10
   .a0K@....1...H..
32: 02a6 97ea 0202 004d 6a82 3c33 373e 4170
   .......Mj.<37>Ap
48: 7220 2031 2031 383a 3534 3a31 3120 7375    r  1
   18:54:11 su
64: 3a20 2773 7520 726f 6f74 2720 7375 6363    : 'su root'
   succ
80: 6565 6465 6420 666f 7220 6f65 7370 6164    eeded for
   oespad
96: 6d20 6f6e 202f 6465 762f 7074 732f 30      m on
   /dev/pts/0
```

The following was recorded on talkabout:

```
Apr  1 18:54:11 ultrabookIIi su: 'su root' succeeded for
   oespadm on /dev/pts/0
```

B Configuring a Socket Server on a UNIX Host

NOTE: The term Agent is interchangeable with Collector. Going forward, Agents will be referred to as Collectors.

A socket server provides an end-point for socket connections from the UNIX Wizard Collector Manager. For example, if you want to monitor a log file or a UNIX box from a remote Wizard and you need to get through a firewall to get to the port on the UNIX box.

The following instructions are for setting up a socket server on a UNIX host and that you will be monitoring an ASCII log file on the UNIX host.

To setup a socket server process on a UNIX host

1. Create the script that will serve the data to the TCP socket connection. To do this, create a new text file and copy the following lines into it, replacing <log file> with the full path name of the file you wish to monitor:

   ```
   #!/bin/sh

   /bin/tail -f <log file>
   ```

   Save the file (path and filename is arbitrary), but the file should be located where it won’t be deleted and named to reflect its function: For example:

   ```
   /usr/local/bin/logfileserver
   ```

2. Pick an unprivileged TCP port on the UNIX host to use for the server process. The unprivileged port number is an arbitrary number between 1025 and 65,535. To check if your port number is already in use, use the following command (replacing <port number> with your desired port):

   ```
   netstat -an | grep LISTEN | grep <port number>
   ```

   If a line is output (as in the following), the port is currently in use, so you’ll have to pick a different one.

   ```
   *.5555*.*0000 LISTEN
   ```

3. As user root, edit file /etc/services and add an entry for your new socket service at the end of the file. The following example adds a line for a service called “syslog_monitor,” configured to listen on TCP port 5555:

   ```
   syslog_monitor5555/tcp
   ```

4. Edit the file /etc/inetd.conf and add an entry for your new socket service at the end of the file. The following example adds a line for a service called “syslog_monitor,” configured to run the script /usr/local/bin/in.syslog_monitor.

   The following should be entered as fields separated by tabs on a single line in the file, regardless of the pagination shown.

```
syslog_monitor stream tcp nowait nobody
   /usr/local/bin/in.syslog_monitor in.syslog_monitor
```

5. Execute the following command to enable your socket server process:

```
kill -HUP `/bin/ps -ef | grep inetd | grep -v grep |
   awk '{print $2}'`
```

6. Try out the socket server. To do this, telnet to the port you choose and you should be greeted by the content of your log file:

```
% telnet localhost 5555
```

   To break out of the telnet session, do a ^] (control-]), then type quit at the telnet> prompt.