

IBM RealSecure

# Server Sensor Policy Guide

Version 7.0

Windows  
Solaris  
HP-UX  
AIX

© Copyright IBM Corporation 1998, 2008.  
IBM Global Services  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America.

All Rights Reserved.

IBM and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. ADDME, Ahead of the threat, BlackICE, Internet Scanner, Proventia, RealSecure, SecurePartner, SecurityFusion, SiteProtector, System Scanner, Virtual Patch, X-Force and X-Press Update are trademarks or registered trademarks of Internet Security Systems, Inc. in the United States, other countries, or both. Internet Security Systems, Inc. is a wholly-owned subsidiary of International Business Machines Corporation.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

**Disclaimer:** The information contained in this document may change without notice, and may have been altered or changed if you have received it from a source other than IBM Internet Security Systems (IBM ISS). Use of this information constitutes acceptance for use in an "AS IS" condition, without warranties of any kind, and any use of this information is at the user's own risk. IBM Internet Security Systems disclaims all warranties, either expressed or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall IBM ISS be liable for any damages whatsoever, including direct, indirect, incidental, consequential or special damages, arising from the use or dissemination hereof, even if IBM Internet Security Systems has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages, so the foregoing limitation may not apply.

Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by IBM Internet Security Systems. The views and opinions of authors expressed herein do not necessarily state or reflect those of IBM Internet Security Systems, and shall not be used for advertising or product endorsement purposes.

Links and addresses to Internet resources are inspected thoroughly prior to release, but the ever-changing nature of the Internet prevents IBM Internet Security Systems, Inc. from guaranteeing the content or existence of the resource. When possible, the reference contains alternate sites or keywords that could be used to acquire the information by other methods. If you find a broken or inappropriate link, please send an email with the topic name, link, and its behavior to [support@iss.net](mailto:support@iss.net).

November 26, 2008

# Contents

## Preface

Overview . . . . .	7
How to Use RealSecure Server Sensor Documentation . . . . .	9
Getting Technical Support . . . . .	10

## Part I: Introduction

### Chapter 1: Introduction to RealSecure Server Sensor

Overview . . . . .	13
About RealSecure Server Sensor . . . . .	14
How RealSecure Server Sensor Works . . . . .	16

### Chapter 2: RealSecure Server Sensor Policies

Overview . . . . .	17
About Policies . . . . .	18
Predefined Policies for Version 7.0 Sensors . . . . .	19
Importing Policies to Version 7.0 Sensors . . . . .	21
About Policy Files . . . . .	23

### Chapter 3: RealSecure Server Sensor Signatures

Overview . . . . .	25
About Signatures . . . . .	26
Signature Organization in the Policy Editor . . . . .	27

## Part II: Configuring Signatures

### Chapter 4: Configuring Firecell Signatures

Overview . . . . .	31
About Firecell Signatures . . . . .	32
Precedence of Firecell Signatures . . . . .	34
Creating and Configuring Firecell Signatures . . . . .	35
Allowing Reply Traffic . . . . .	37
Example of How A Firecell Signature Works . . . . .	38
Disabling Firecell Signatures . . . . .	39
Examples of How to Use Firecell Signatures . . . . .	40

### Chapter 5: Configuring Suspect Connection Signatures

Overview . . . . .	43
Monitoring for Suspect Connections . . . . .	44
Monitoring for Custom Suspect Connections . . . . .	46

### Chapter 6: Configuring Network Protection Signatures

Overview . . . . .	47
Customizing Predefined Network Signatures . . . . .	48
User-Defined Network Signatures . . . . .	49
Adding a User-defined Network Signature . . . . .	51

## Chapter 7: Auditing System Integrity and Policy Compliance

Overview	55
<b>Section A: Prerequisites to Configuring Audit Signatures</b>	57
Overview	57
Monitoring for Audit Related Events	58
Monitoring Local Syslog Events	60
Monitoring the Mail Subsystem on HP-UX Systems	61
Prerequisite to C2 Audit Logging on an HP-UX System	62
<b>Section B: Configuring Predefined Audit Signatures</b>	65
Overview	65
Customizing Predefined Audit Signatures	66
<b>Section C: Configuring User-Defined Audit Signatures</b>	67
Overview	67
Using Regular Expressions in User-Defined Signatures	68
Auditing the Windows Event Log	70
Task 1: Creating a User-Defined Signature to Audit the Windows Event Log	71
Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures	73
Task 3: Identifying Relevant Windows Event Log Information	74
Task 4: Refining Your Windows Event Log User-Defined Signature	76
Auditing Files and Registry Entries with a Windows Platform Sensor	77
Monitoring a Specific Log File	81
Monitoring Custom Events in UNIX Syslogs	85
Monitoring the Wtmpx Binary Log File	87
Selecting Logs	89
Specifying Exceptions	90
<b>Section D: Configuring User-Defined C2 Audit Signatures</b>	93
Overview	93
About C2 Audit Signatures	95
Task 1: Creating a User-Defined C2 Audit Signature	96
Task 2: Generating the Event	99
Task 3: Using the Audit Log Display Command to Examine the Audit Log File	100
Task 4: Configuring the Information That Will Generate a Response	103
Task 5: Configuring the Information Fields Responses Should Return	105
Task 6: Choosing Responses	106
Configuring Name Resolution for BSM Auditing	107
Configuring C2 Audit Log Management	109
Disabling Sensor Management of the BSM Log	110
Examples of User-Defined C2 Audit Information	111

## Part III: Advanced Configuration

### Chapter 8: Configuring Fusion Scripting

Overview	115
<b>Section A: Introduction to Fusion Scripting</b>	117
Overview	117
Introduction to Fusion Scripting	118
Data Available to Fusion Scripting	120
Tcl Script Categories Used in Fusion Scripting	122

<b>Section B: Working with Fusion Scripting</b> . . . . .	125
Overview . . . . .	125
Predefined Tcl Extensions . . . . .	126
Adding or Modifying a Fusion Script . . . . .	128
Configuring a Fusion Scripting Response . . . . .	129
Configuring a Fusion Scripting SNMPv3 Response . . . . .	130
Returning a True or False Result in a Validation Script. . . . .	134
Using Fusion Scripts . . . . .	135
Disabling Fusion Scripting . . . . .	140
<b>Chapter 9: Capturing Packet Information</b>	
Overview . . . . .	143
About Packet Logging. . . . .	144
Enabling Packet Logging . . . . .	145
About Evidence Logging . . . . .	147
Enabling Evidence Logging. . . . .	148
<b>Chapter 10: Monitoring for Buffer Overflow Exploits</b>	
Overview . . . . .	151
About Buffer Overflow Exploit Protection . . . . .	152
Enabling Default Buffer Overflow Exploit Protection . . . . .	153
Monitoring Additional Directories . . . . .	154
Changing the Action for All Monitored Directories. . . . .	156
Changing the Action for a Specific Program File . . . . .	159
<b>Chapter 11: Fine-Tuning RealSecure Server Sensor</b>	
Overview . . . . .	163
Excluding an Interface from Monitoring . . . . .	164
Customizing the Buffer Size for Interfaces . . . . .	167
Excluding Packets from Analysis. . . . .	170
Defining the Sensor Pass-through Mode . . . . .	173
 <b>Part IV: Troubleshooting</b>	
<b>Chapter 12: Troubleshooting</b>	
Overview . . . . .	177
Isolating Policy Problems . . . . .	178
Tcl Script Problems . . . . .	185
No Communication Between the Sensor and the Console . . . . .	186
No Communication Between the Sensor on an ISA Server and the Console . . . . .	187
Failure to Open a Control Channel Error . . . . .	188
Not Seeing Any BOEP Events. . . . .	189
BOEP Action for a Specific Program File Not Working . . . . .	190
Api Read Queue Messages in the Syslog . . . . .	191
 <b>Appendixes</b>	
<b>Appendix A: Configuring the Web Server Monitoring Component</b>	
Overview . . . . .	195
Configuring an Apache Web Server Monitoring Component Manually. . . . .	196
Configuring an IIS Web Server Monitoring Component Manually . . . . .	197
<b>Index</b> . . . . .	199



# Preface

## Overview

<b>Introduction</b>	This guide provides the information you need to configure IBM RealSecure Server Sensor version 7.0. After you read this guide, you should be able to configure the sensor to protect your servers against attacks and misuse.
<b>Purpose</b>	This policy guide describes how to use RealSecure Server Sensor to protect your system from attacks and misuse.
<b>Scope</b>	This policy guide describes features that are specific to the RealSecure Server Sensor. General information about sensors, such as managing policies and configuring responses, is described in the SiteProtector Help.
<b>Audience</b>	<p>This guide is intended for security managers who manage sensors from SiteProtector.</p> <p><b>Note:</b> If you plan to use the Fusion Scripting feature, you must have a thorough understanding of the Tool Control Language (Tcl) scripting language.</p>
<b>What's new in this guide for Service Release 4.2 for AIX platforms</b>	<p>This guide was updated for RealSecure Server Sensor for AIX, Service Release 4.2 and includes new or revised information about the following topics:</p> <ul style="list-style-type: none"><li>● you can now define a pass-through mode for the sensor. See "Defining the Sensor Pass-through Mode" on page 173.</li></ul>
<b>What's new in this guide for Service Release 4.4 for Windows platforms</b>	<p>This guide was updated for RealSecure Server Sensor for Windows, Service Release 4.4 and includes new or revised information about the following topics:</p> <ul style="list-style-type: none"><li>● the sensor can now exclude traffic from a specific IP address from analysis. See "Excluding Packets from Analysis" on page 170.</li><li>● customizing buffer overflow exploit protection, see "Customizing protection" on page 152</li><li>● customizing the global buffer overflow exploit protection action, see "Changing the Action for All Monitored Directories" on page 156</li></ul>
<b>What's new in this guide for Service Release 4.3 for Solaris platforms</b>	<p>This guide was updated for RealSecure Server Sensor for Solaris, Service Release 4.3 and includes new or revised information about the following topics:</p> <ul style="list-style-type: none"><li>● the sensor can now exclude a specific Network Interface Card (NIC) from protection. See "Excluding an Interface from Monitoring" on page 164.</li></ul>

- the sensor can now allocate specific buffer sizes to individual NICs. See “Customizing the Buffer Size for Interfaces” on page 167.
- the sensor can now exclude traffic from a specific IP address from analysis. See “Excluding Packets from Analysis” on page 170.
- the sensor can now leave management of the Basic Security Module (BSM) audit logs to an independent process. See “Disabling Sensor Management of the BSM Log” on page 110.
- the sensor can now resolve numerical user and group information from BSM logs to their corresponding user and group names. See “Configuring Name Resolution for BSM Auditing” on page 107.

**What's new in this guide for Service Release 4.1 for HP-UX platforms**

This guide was updated for RealSecure Server Sensor for HP-UX, Service Release 4.1 and includes new or revised information about the following topics:

- the sensor now uses the `AllowAllAcknowledgementPackets` advanced tuning parameter to allow reply traffic through the firewall. See “Allowing Reply Traffic” on page 37.
- the sensor can exclude a specific Network Interface Card (NIC) from protection. See “Excluding an Interface from Monitoring” on page 164.
- the sensor can allocate specific buffer sizes to individual NICs. See “Customizing the Buffer Size for Interfaces” on page 167.
- the sensor can exclude traffic from a specific IP address from analysis. See “Excluding Packets from Analysis” on page 170.
- the sensor blocking behavior and configuration is now consistent with RealSecure Server Sensor for other platforms. See the Help for more information.



# How to Use RealSecure Server Sensor Documentation

**Using this guide** Read the entire guide before you begin to configure the sensor so you can become familiar with the protection the sensor provides. After you are more familiar with features of the sensor, refer to the guide as necessary to configure protection.

**Related publications** For additional information about RealSecure Server Sensor, see the following publications:

- *RealSecure Server Sensor Installation Guide*
- *RealSecure Server Sensor Advanced Tuning Parameters Reference Document*
- *SiteProtector Policies and Responses Guide*
- *SiteProtector Configuration Guide*

**License agreement** For licensing information on IBM Internet Security Systems products, download the IBM Licensing Agreement from:

[http://www-935.ibm.com/services/us/iss/html/contracts\\_landing.html](http://www-935.ibm.com/services/us/iss/html/contracts_landing.html)

# Getting Technical Support

**Introduction** IBM Internet Security Systems provides technical support through its Web site and by email or telephone.

**The IBM ISS Web site** The Customer Support Web page (<http://www-935.ibm.com/services/us/index.wss/offerfamily/iss/a1029129>) provides direct access to online user documentation, current versions listings, detailed product literature, white papers, and the Technical Support Knowledgebase.

**Hours of support** The following table provides hours for Technical Support at the Americas and other locations:

Location	Hours
Americas	24 hours a day
All other locations	Monday through Friday, 9:00 A.M. to 6:00 P.M. during their local time, excluding IBM ISS published holidays <b>Note:</b> If your local support office is located outside the Americas, you may call or send an email to the Americas office for help during off-hours.

**Table 1:** *Hours for technical support*

**Contact information** For contact information, go to the Contact Technical Support Web page at <http://www-935.ibm.com/services/us/index.wss/offering/iss/a1029178>.

## Introduction



## Chapter 1

# Introduction to RealSecure Server Sensor

## Overview

### Introduction

This chapter introduces RealSecure Server Sensor, an automated, real-time intrusion detection and response system that unobtrusively analyzes activity across your computer systems and networks, and describes how it protects your servers.

### In this chapter

This chapter contains the following topics:

Topic	Page
About RealSecure Server Sensor	14
How RealSecure Server Sensor Works	16

## About RealSecure Server Sensor

<b>Introduction</b>	RealSecure Server Sensor monitors traffic to and from a single server. In addition to detecting intrusions, the sensor can also prevent intrusions by blocking specified network packets. The sensor can also identify attacks destined for active services on the protected host.
<b>Management</b>	Manage the sensor with SiteProtector Version 2.0, Service Pack 5.2 or later.
<b>Sensor attributes</b>	RealSecure Server Sensor has the following attributes: <ul style="list-style-type: none"><li>● detects both network and system events</li><li>● detects events at the application layer</li><li>● detects events before they reach the IP stack</li><li>● monitors traffic to and from the host it is installed on</li><li>● prevents intrusions</li><li>● extends validation and response options with Fusion Scripting</li></ul>
<b>Network and system event detection</b>	RealSecure Server Sensor monitors both network and operating system (OS) activity.
<b>Event detection at the application layer</b>	<p>RealSecure Server Sensor monitors traffic before it reaches a running application. The sensor detects sophisticated, high-level protocol-specific attacks at this level. Exploits found at this level are typically multi-packet attacks, such as the HTTP PHF attacks.</p> <p>Monitoring at the application layer provides the following benefits:</p> <ul style="list-style-type: none"><li>● The sensor can analyze streams of data in addition to analyzing individual packets.</li><li>● The sensor is not susceptible to packet fragmentation because TCP packets are reassembled above the stack.</li><li>● The sensor can monitor traffic that is encrypted with software, such as SSL, IPSEC, or SKIP, because the traffic is not encrypted above the stack.</li><li>● When blocking in response to an event that was detected at the application layer, the sensor drops the traffic associated with the event and prevents the event from affecting the application.</li></ul>
<b>Event detection before reaching the IP stack</b>	<p>RealSecure Server Sensor monitors network traffic as it moves through the server's kernel. The sensor watches for protocol violations, such as header violations, and other single packet events. Watching packets at this level allows the sensor to detect and block simple events, such as winnuke, before the packet can even enter the TCP/IP stack.</p> <p>When blocking in response to an event that was detected before it reached the IP stack, blocking drops the packet so that it never reaches the IP stack.</p>
<b>Monitoring traffic to and from one host</b>	RealSecure Server Sensor monitors traffic to and from one computer to determine if an intruder has gained access to that computer. Advantages to monitoring traffic to and from one computer include the following:

- The coverage is not adversely affected by switched networks because traffic is monitored at the host instead of on a network segment.
- High-traffic networks do not adversely affect the performance of the sensor because the traffic workload is distributed at the host level.

**Intrusion prevention** RealSecure Server Sensor can prevent intrusions by blocking unacceptable network packets all the time or in response to a specific event. You can block events using one of the following methods:

- the block response

**Reference:** For more information about the block response, see the SiteProtector Help.

- user-defined firecell signatures

**Reference:** For more information about firecell signatures, see “About Firecell Signatures” on page 32.

**Note:** The sensor only identifies attacks destined for active services. Attacks destined for services that do not exist will not be caught because the system does not pass the attack traffic up the stack. This specific monitoring reduces the number of false positives on the system.

**Extended validation and response with Fusion Scripting** You can use Fusion scripts to validate and respond to the information that the sensor collects.

**Reference:** For more information see Chapter 8, "Configuring Fusion Scripting" on page 115.

## How RealSecure Server Sensor Works

<b>Introduction</b>	This topic provides a high-level overview of how RealSecure Server Sensor protects your system against attacks and misuse.
<b>Events</b>	Events are attacks or misuse detected by the sensor that may result in an alert being sent to the Console.
<b>Policies</b>	<p>Policies control sensor behavior. Policies contain items, called signatures, which determine what types of events the sensor monitors for. Policies control the following sensor behaviors:</p> <ul style="list-style-type: none"><li>● the type of security events a sensor detects</li><li>● the priority assigned to each event</li><li>● the sensor's response to the event</li></ul> <p>You view, create, and edit policies from the Policy Editor accessed from the SiteProtector Console. You also use SiteProtector to apply new or updated policies to a sensor.</p>
<b>Signatures</b>	A signature is the internal code that the system uses to detect an attack or a misuse that might signal an attack on your system. Signatures can provide security-related information. By enabling or disabling the signatures in a policy, you determine what types of events the sensor monitors for.
<b>Policy files</b>	Policy files control how policies are applied to a sensor. Each sensor has its own set of policy files.
<b>Number of sensors a Console can manage</b>	<p>There is no limit to the number of sensors a single Console can manage; however, the practical number of sensors that can report effectively to a single Console depends on the following considerations:</p> <ul style="list-style-type: none"><li>● system configuration of the system running SiteProtector</li><li>● amount of traffic flowing between the sensor and SiteProtector</li><li>● number of signatures enabled in the policy</li><li>● the geographic and organizational limitations of the controlling organization</li></ul>



## Chapter 2

# RealSecure Server Sensor Policies

## Overview

### Introduction

This chapter describes RealSecure Server Sensor policies, lists the predefined policies that come with the sensor, and describes how policy files ensure that the sensor runs the correct policy.

### In this chapter

This chapter contains the following topics:

Topic	Page
About Policies	18
Predefined Policies for Version 7.0 Sensors	19
Importing Policies to Version 7.0 Sensors	21
About Policy Files	23

## About Policies

- Introduction** Policies define the types of events the sensor monitors for and how the sensor responds to those events. This topic describes predefined policies and user-defined policies.
- Predefined policies** RealSecure Server Sensor comes with several predefined policies. The Policies tab on the Server Sensor Policies window lists the predefined policies. You cannot edit predefined policies, but you can customize a predefined policy by saving the policy under a different name before you make any changes.
- User-defined policies** You can create user-defined, or customized, policies to monitor your system when none of the predefined policies meet your needs. Create a user-defined policy by deriving a new policy from the predefined policy that most closely resembles the policy configuration you need. After you derive a new policy, edit the copy to meet your specific needs.
- Reference:** For more information about how to make an editable copy of a predefined policy, see the Help.
- How to work with policies** You can view, create, and edit policies from the Policy Editor, which you access from the SiteProtector Console. You also use the Console to apply new or updated policies to a sensor. For information on working with policies, see the Help.

## Predefined Policies for Version 7.0 Sensors

### Introduction

RealSecure Server Sensor has several predefined policies. Predefined policies come with certain settings already enabled; this allows you to quickly implement protection while you determine the best configuration for your sensor. This topic describes the predefined policies that come with version 7.0 sensors and the types of events they monitor for.

### Events monitored for by predefined policies

Predefined policies monitor for the following types of intrusions and events:

- some network events
- some OS events

### Events not monitored for by predefined policies

Predefined policies do not monitor for the following types of events:

- UNIX syslog events
- suspicious connection events
- firecell events

### Description of predefined policies

The following table describes the predefined policies for version 7.0 sensors:

Policy	Description
Blank_<platform>	<p>This policy has no signatures enabled by default so that you can enable only those signatures that you need.</p> <p><b>Important:</b> This is the default policy. To enable protection, you must derive a copy of this policy (or another policy with a configuration similar to the one you want), customize the copy, and then apply the policy to the sensor.</p>
Network_Attacks_<platform>	<p>This policy, by default, enables all network attack signatures that are applicable to the server platform.</p> <p>This policy does not enable the following:</p> <ul style="list-style-type: none"> <li>• operating system signatures</li> <li>• network signatures not applicable to the server platform</li> <li>• any network audit signatures</li> <li>• BLOCK response for any network signatures</li> <li>• dynamic block for any network signatures</li> </ul> <p><b>Important:</b> If you want to enable blocking and auditing for this policy, you must derive a copy of this policy and customize the copy.</p>

**Table 2:** *Predefined policies for 7.0 server sensors*

Policy	Description
Attacks_And_Audits_<platform>	<p>This policy enables the following by default:</p> <ul style="list-style-type: none"> <li>• operating system audit event log signatures</li> <li>• network attack and audit signatures applicable to the platform</li> </ul> <p>This policy does not enable the following:</p> <ul style="list-style-type: none"> <li>• network signatures not applicable to the platform</li> <li>• BLOCK response for any network signatures</li> <li>• dynamic block for any network signatures</li> </ul> <p><b>Important:</b></p> <ul style="list-style-type: none"> <li>• To receive events that rely on an audit flag setting, you must also enable the Enforce Audit Policy setting on each sensor you apply this policy to. For more information, see “Monitoring for Audit Related Events” on page 58.</li> <li>• If you want to enable blocking for this policy, you must derive a copy of this policy and customize the copy.</li> </ul>

**Table 2:** *Predefined policies for 7.0 server sensors (Continued)*

# Importing Policies to Version 7.0 Sensors

## Introduction

To save time configuring policies, you can import customized policies from version 6.5 sensors to use with version 7.0 sensors. This topic describes the implications of importing version 6.5 policies for use with version 7.0 sensors.

**Important:** You cannot import earlier version policies to version 7.0 sensors for the HP-UX or AIX platforms because there are no earlier version policies for these platforms.

## Importing earlier versions of policies

The following table outlines how a version 6.5 policy will import to a version 7.0 sensor, and describes, where applicable, the actions you must take to configure the imported policy:

Function	6.5 Policy	Action/Notes
Dynamic block setting	will import	In version 6.5 sensors, you configured dynamic blocking at the signature level. In version 7.0 sensors, you can configure dynamic blocking at the sensor level or at the signature level. If you import a version 6.5 policy to version 7.0, the signature level dynamic block configuration imports as signature level dynamic block configuration. <b>Note:</b> If dynamic blocking is enabled at the sensor level in the version 7.0 sensor, the sensor level setting overrides the signature level setting. <b>Reference:</b> For more information about dynamic blocking and the Block response, see the Help.
Default Block response settings	will not be different	
User-defined events	will import	
Signature settings	will import	All signature settings except the dynamic block setting (as noted above) will import.
Signature responses	will import	
SSL events	will import	
HTTP, email, and FTP event detection	will function appropriately	
New 7.0 signatures groups		Apply the policy to a 7.0 sensor to add the new signature groups.

**Table 3:** *Importing version 6.5 policies to a version 7.0 server sensor*

**Importing a version 6.5 policy to a version 7.0 sensor**

To import a policy:

1. In the Sensor Policies window, select the **Server** tab.
2. Click **Import Policy**.

The Choose Source Policy File window appears.

3. Locate and select the policy you want to import, and then click **Open**.

The Choose Target Policy File window opens.

**Note:** You can change the location of the policy, but new imported policies are saved in the default directory. The default directory for sensor policies is `ISS\SiteProtector\Console\Server Policies`.

4. Type a name for the imported policy.
5. Select **Current** as the **Save As type**.
6. Click **Save**.

A confirmation message appears.

7. Click **Yes**.

A confirmation message informs you that the policy was imported successfully.

8. Click **OK**.

The newly imported policy appears on the Sensor Policies window.

9. Apply the policy to a version 7.0 sensor to add the new sensor signatures.



**Caution:** After you upgrade a version 6.5 policy to a 7.0 policy, you must not push the upgraded policy to a 6.5 sensor. Do not use version 7.0 policies with earlier versions of sensors.

## About Policy Files

### Introduction

Policy files are files that control how policies are applied to the sensor.

### Understanding policy files

The following table describes the policy files used by the sensor:

Policy file	Description
audit.policy	The audit.policy file contains a list of global audit flags and a list of registry keys with audit flags set by the sensor when you select the Enforce Audit Policy check box on the Server Sensor tab.
common.policy	The common.policy file contains information used by the daemon and sensor, as follows: <ul style="list-style-type: none"> <li>• available cryptographic providers</li> <li>• destination and community for any SNMP traps</li> <li>• flags for automatic SNMP trap generation (upon sensor start, upon sensor policy change, etc.)</li> <li>• a list of variable response types</li> </ul>
current.policy	The current.policy file contains the currently active policy, if the sensor is running. When the sensor starts, it loads current.policy first. When the sensor has finished initializing, it rewrites its configuration to the current.policy file.
Default.policy	The Default.policy file contains the default policy of the sensor. This file is only loaded if current.policy file is missing. To revert to a default configuration, stop the sensor, delete current.policy and push.policy, and then restart the sensor. <b>Note:</b> For version 7.0 sensors on the Windows and HP-UX platforms, Default.policy has the same settings as the Network_Attacks_<platform> policy. For version 7.0 sensors on the Solaris and AIX platforms, Default.policy has the same settings as the Blank_<platform> policy.
eventlog.policy	The eventlog.policy file maintains a list of events that were transferred to the Console during the last EventLog query for a daemon sensor. The eventlog.policy is located in the issDaemon directory.
issCSF.policy	The issCSF.policy file is the main policy file for the Common Sensor Framework (CSF). CSF reads this policy file to learn which sensor and response plug-ins it needs to load to become a fully functional sensor.
issDaemon.policy	The issDaemon.policy file contains information used by the daemon, for example: <ul style="list-style-type: none"> <li>• sensor utilization timeout: how long the daemon tries to connect to a sensor before it gives up</li> <li>• daemon port: the port number used by the daemon for sensor communication (normally 2998)</li> <li>• master console: the Console that has master status of the sensor controlled by the daemon.</li> </ul>

**Table 4:** Policy file descriptions

Policy file	Description
push.policy	The push.policy file contains a copy of the last policy that was applied to the sensor. When you make changes to a policy from the Console, the changes are transmitted using TCP on port 2998 (the ISS daemon port), and then saved as push.policy. After push.policy is merged with current.policy, the sensor deletes push.policy.
ruledef.policy	The ruledef.policy file contains definitions for predefined signatures.
update.policy	The update.policy file contains the configurations for new signatures. Because new signatures are included with new versions of the sensor, the configurations for these signatures are distributed in update.policy. When you apply an imported policy to the sensor, the sensor merges the contents of update.policy with the imported policy so that all the new signatures are available. The resulting configuration is written to current.policy so that the current.policy file represents the latest policy configuration for the sensor.

**Table 4:** Policy file descriptions (Continued)



## Chapter 3

# RealSecure Server Sensor Signatures

## Overview

### Introduction

RealSecure Server Sensor uses signatures to detect security events; therefore, it is important to understand the different types of signatures and how the sensor uses them to protect your system.

### In this chapter

This chapter contains the following topics:

Topics	Page
About Signatures	26
Signature Organization in the Policy Editor	27

## About Signatures

<b>Introduction</b>	Policies, discussed in Chapter 2, consist of signatures. This topic describes sensor signatures.
<b>Types of signatures</b>	RealSecure Server Sensor includes the following types of signatures: <ul style="list-style-type: none"><li>● firecell signatures, which protect the server from attacks on specific ports or attacks contained in certain IP traffic</li><li>● network signatures (block response), which protect the server from attacks directed across the network</li><li>● operating system signatures, which monitor system integrity and policy compliance through system log files</li></ul>
<b>Predefined signatures</b>	RealSecure Server Sensor comes with several predefined signatures contained in the predefined policies.
<b>Predefined signature attributes you can customize</b>	You can customize a predefined signature to meet your specific security needs by editing the following attributes: <ul style="list-style-type: none"><li>● priority settings</li><li>● responses</li><li>● protocol ports (for Network Event signatures)</li><li>● list of important files and list of registry keys (for certain audit signatures)</li></ul>
<b>User-defined signatures</b>	You can create custom signatures if the predefined signatures do not meet your security needs. See the applicable chapters for more information about configuring user-defined signatures.

## Signature Organization in the Policy Editor

### Policy Editor organization

The policy editor organizes sensor signatures in the tabs described in the following table:

Tab	Description
Protect	<p>Displays firecell signatures and connection event signatures. These signatures focus on intrusion prevention rather than intrusion detection. Use firecell signatures to block packets that meet a certain criteria or to monitor and respond to specific traffic without blocking packets.</p> <p><b>Reference:</b> For more information about firecell signatures, see Chapter 4, "Configuring Firecell Signatures" on page 31.</p> <p>Use connection event signatures to monitor for suspicious connections to ports and services.</p> <p><b>Reference:</b> For more information about connection event signatures, see "Monitoring for Suspect Connections" on page 44.</p>
Network Events	Displays network-based signatures. Network-based signatures monitor network traffic for content that can indicate an attack or other suspicious activity.
OS Events	Displays log-based signatures. The OS Events signatures audit activity at the operating system level by monitoring system log files.
X-Press Updates	If you install an X-Press Update that contains new signatures, an X-Press Update tab appears in the policy editor. This tab lists the signatures contained in every X-Press Update you have installed since the last release of the sensor.

**Table 5:** *Server sensor signature categories*



## Configuring Signatures



## Chapter 4

# Configuring Firecell Signatures

## Overview

### Introduction


Firecell signatures provide a host-based firewall that focuses on intrusion prevention rather than intrusion detection. Firecell signatures can block packets that meet a certain criteria or they can monitor and respond to specific traffic without blocking packets. This chapter describes firecell signatures and explains how to create and use them.

### In this chapter

This chapter contains the following topics:

Topic	Page
About Firecell Signatures	32
Precedence of Firecell Signatures	34
Creating and Configuring Firecell Signatures	35
Allowing Reply Traffic	37
Example of How A Firecell Signature Works	38
Disabling Firecell Signatures	39
Examples of How to Use Firecell Signatures	40

## About Firecell Signatures

<b>Introduction</b>	Firecell signatures are based on protocol and packet detection. They work like a firewall to ensure that only authorized clients can access the server. This topic explains how firecell signatures work so that you can effectively design the protection they provide.
<b>Firecell signatures</b>	Firecell signatures do the following: <ul style="list-style-type: none"><li>● allow only subnet traffic</li><li>● monitor IP traffic that is not from your local subnet</li><li>● allow only typical Internet traffic</li><li>● drop unauthorized traffic</li></ul>
<b>Parameters used to define firecell signatures</b>	Use the following parameters to define firecell signatures: <ul style="list-style-type: none"><li>● protocol type (IP, TCP, UDP, or ICMP)</li><li>● specific IP address or class of addresses</li><li>● port number (for TCP and UDP firecell signatures) <b>Note:</b> When you define a port number, the following rules apply:<ul style="list-style-type: none"><li>■ you can monitor all ports by entering a 0 (zero) in the Port box</li><li>■ you cannot specify a range of ports other than all ports</li><li>■ you cannot enter more than one number in the Port box (you must enter either 0 for all ports, or the specific port number)</li></ul></li></ul>
<b>Responses available to firecell signatures</b>	You can configure the sensor to generate any predefined or user-defined response when it detects a packet that matches the criteria of the firecell signature. You can also configure the sensor to allow the packet to pass through the network stack if you want to record and respond to the packet but you do not want to stop it.   <b>Caution:</b> Be careful when you assign responses. The sensor generates responses for each packet it detects, so, if the sensor detects many packets that match a firecell signature, you may overload the processor or use all your disk space.
<b>When to use firecell signatures</b>	Use firecell signatures to block certain packets all the time, even if the packets match a default or other user-defined signature.
<b>Firecell signatures versus dynamic blocking</b>	Firecell signatures that conflict with dynamic block settings are ignored in favor of the dynamic block setting. When the dynamic block setting that conflicts with a firecell signature expires, the firecell signature will resume functioning.  <b>Note:</b> This applies to RealSecure Server Sensor Version 7.0 for Solaris, RealSecure Server Sensor Version 7.0, Service Release 4.2 and later for Windows, RealSecure Server Sensor Version 7.0, Service Release 4.1 for HP-UX, and RealSecure Server Sensor Version 7.0, Service Release 4.1 for AIX.  <b>Example:</b> You configure a firecell signature to specifically allow all traffic from an IP address. The sensor is also configured to dynamically block specific activity from the IP



address. The sensor blocks the traffic because the dynamic block setting takes precedence over the firecell rule that allows all traffic.

**Specifying ports**

If you specify a port number when you configure a firecell signature, that signature will only be applied to the port number specified. Packets that may have otherwise triggered the signature will not trigger if they are using another port as the signature is specific to the port.

## Precedence of Firecell Signatures

### Introduction

The order in which you list firecell signatures is very important. When you look at signatures listed under a particular category, such as TCP Inbound, the first signature listed in the category takes precedence over the next signature in the category. You can rearrange the order of firecell signatures in the policy editor.

### Example

You create two TCP Inbound signatures; the first signature allows incoming traffic on port 80 from a particular subnet, but the second signature does not allow TCP traffic on port 80 from any IP address (0.0.0.0/0), the first signature has precedence and allows incoming traffic on port 80 from the local subnet.

### Rearranging firecell signatures

To rearrange firecell signatures:

1. Open the policy that contains the signatures you want to rearrange.
2. Select the **Protect** tab, and then open the **Protect** folder.
3. Open the **Firecell** folder.  
A list of firecell signature types, based on protocol, appears.
4. Select the firecell category that contains the signatures you want to rearrange.

**Example:** To rearrange signatures that control inbound TCP traffic, select **TCP Inbound**.

The signatures in that category appear in the right pane.

5. Use a drag-and-drop operation to move the signature to the new position.

	Enabled	Event	Priority	Response	Port
1	<input type="checkbox"/>	Block_HTTP_Incomi	High		80
2	<input checked="" type="checkbox"/>	Disallow all TCP inb	High		0
3	<input checked="" type="checkbox"/>	Allow port 80	High		80
4	<input checked="" type="checkbox"/>	Allow port 25	High		25




Figure 1: Rearranging firecell signatures

# Creating and Configuring Firecell Signatures

## Introduction

This topic describes how you create and configure firecell signatures. When you configure a firecell signature, you can have the sensor block or allow packets based on port number (for TCP and UDP protocols only) and IP address.

## Important considerations

Consider the following before you create and configure firecell signatures:

- the sensor needs certain traffic to run properly, create firecell signatures that specifically allow this traffic before you create firecell signatures that block traffic
- to allow traffic from specific systems (for example, from SiteProtector components) that reside on a different subnet than the sensor, you must include the IP address of the sensor gateway in the signature



**Caution:** If you configure firecell signatures incorrectly, you can disable all traffic to and from a server, including communication between the Console and the sensor.

**Reference:** See “Precedence of Firecell Signatures” on page 34.

## Creating a firecell signature

To create a firecell signature:

1. Open the policy you want to add the firecell signature to.
2. Select the **Protect** tab, and then open the **Protect** folder.
3. Open the **Firecell** folder.

A list of firecell signature types, based on protocol, appears.

4. Select the type of firecell signature you want to add.
5. Click **Add**.

The Enter a name window appears.

6. Type a name for the firecell signature you want to create, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the type of firecell signature you selected in Step 4.

## Configuring a firecell signature

To configure a firecell signature:

**Note:** Before you can configure a firecell signature, you must create the signature. See “Creating a firecell signature” above.

1. Open the policy that contains the signature you want to configure.
2. Select the **Protect** tab, and then open the **Protect** folder.
3. Open the **Firecell** folder.

A list of firecell signature types, based on protocol, appears.

4. In the left pane, select the signature you want to configure.

The properties of the signature appear in the right pane.

5. Set the priority of the event in the **Priority** box.
6. Is this signature monitoring for TCP or UDP traffic?

- If *yes*, go to Step 7.
  - If *no*, go to Step 8.
7. In the **Port** box, type the number of the port this signature applies to.  
**Note:** Use zero (0) to apply this signature to all ports.
8. Click **Add**, and then specify the IP address(es) of the packets that you want to allow or to not allow as described in the following table:

If you want to specify...	Then...
one IP address	<ol style="list-style-type: none"> <li>1. Select <b>IP Address</b>.</li> <li>2. In the <b>Address</b> box, type the IP address.</li> <li>3. Click <b>OK</b>.</li> </ol>
a range of IP addresses in a network class	<ol style="list-style-type: none"> <li>1. Select <b>IP Address</b>.</li> <li>2. In the <b>Address</b> box, type an address that is in the range you want to specify.</li> <li>3. In the <b>Mask</b> box, type a number to represent the level of network class.  <b>Reference:</b> For more information about using masks and other items in this window, click <b>Help</b>.</li> <li>4. Click <b>OK</b>.</li> </ol>
all IP addresses	<p><b>Important:</b> Do not select this option as it will block communication with SiteProtector.</p>
a network asset	<p><b>Prerequisite:</b> The asset must exist before you can specify an asset here.</p> <ul style="list-style-type: none"> <li>• Select <b>Network Asset</b>, choose an asset from the list on the right, and then click <b>OK</b>.</li> </ul>

9. In the Actions section, specify how you want the sensor to treat the packets that match the specified addresses.  
**Reference:** See “Example of How A Firecell Signature Works” on page 38.
10. Select the responses the sensor should take when a match to this signature occurs.  
**Reference:** For more information about each response, see the Help.
11. Click **Save**.
12. Apply the policy to the sensor(s) that you want to use the newly configured signature.  
**Reference:** For more information about applying policies, see the Help.

---

## Allowing Reply Traffic

### Introduction

RealSecure Server Sensor can differentiate inbound TCP connection requests from inbound TCP reply traffic. This means that the sensor can allow reply traffic even if there are firecell signatures configured to block traffic from a specific IP address. This feature is available in the following versions of the sensor:

- versions 6.5 and later for the Windows platform
- version 7.0 and later for the AIX and Solaris platforms
- version 7.0, Service release 4.1 and later for the HP-UX platform

### Behavior

When you enable this feature, the sensor does the following:

- blocks all incoming TCP connection attempts  
**Note:** In some situations it may appear that a connection was initiated from the computer on which the sensor is installed when it was actually initiated from a remote computer. In these situations the sensor blocks the connection request because the connection was not initiated from the computer on which the sensor is installed. For example, because active mode FTP connections appear to originate from the computer where the sensor is installed when they are in fact initiated from a remote computer, the sensor blocks active mode FTP connections.
- prevents firecell events from triggering for any incoming reply packets on port 1024 or above

### Procedure

To allow reply traffic:

**Note:** Use this procedure for all sensors except server sensor for Windows version 7.0, Service Release 4.1 or earlier. See Knowledge Base article 2144 for information about how to configure this feature for that sensor.

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `AllowAllAcknowledgementPackets` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.  
**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
3. In the **Name** box, type the parameter name, `AllowAllAcknowledgementPackets`.  
**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **Boolean**.
5. In the **Value** box, select **True**.
6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.  
The tuning parameter with the new setting is listed in the parameters table.
8. Click **OK**.

## Example of How A Firecell Signature Works

**Example**

The following table gives examples of how each option works if the following conditions are true:

- 196.131.16.2 is the only IP address listed in the IP Addresses list
- the port number is set to zero to monitor traffic on all ports

Selected option	If the sensor detects a packet with a source IP of...	the sensor...	and...	If the sensor detects a packet with a source IP of...	the sensor...	and...
In the range of listed IP addresses, drop the packet and generate selected responses	196.131.16.2	drops the packet	generates responses.	208.21.28.3	accepts the packet	does not generate responses.
In the range of listed IP addresses, accept the packet, but continue to generate responses	196.131.16.2	accepts the packet	generates responses.	204.20.3.41	accepts the packet	does not generate responses.
Not in the range of listed IP addresses, drop the packet and generate selected responses	208.21.16.3	drops the packet	generates responses.	196.131.16.2	accepts the packet	does not generate responses.
Not in the range of listed IP addresses, accept the packet, but continue to generate selected responses	104.20.3.41	accepts the packet	generates responses.	196.131.16.2	accepts the packet	does not generate responses.

**Table 6:** Example of action options for firecell signatures

# Disabling Firecell Signatures

**Introduction** If you want to stop a firecell signature from blocking traffic or from generating responses to traffic, but you do not want to remove the signature, you can disable it.

**Procedure** To disable a firecell signature:

1. From the **Protect** tab of the policy, clear the check box of the signature.
2. Click the **Save** icon to save your changes.
3. Apply the policy to the sensor(s) that you want to stop using the signature.

**Reference:** For a procedure on applying a policy, see the Help.

## Examples of How to Use Firecell Signatures

### Introduction

This topic describes how you can use firecell signatures to do the following:

- allow only local subnet traffic
- monitor IP traffic that is not from your local subnet
- allow only typical Internet traffic

### Allow only local subnet traffic

You can use a firecell signature to allow local subnet traffic but to block all other network traffic.

**Example:** To block all IP traffic that does not originate from the subnet 172.25.50.0/24, create an IP Inbound signature with the following parameters:

Parameter	Value
IP Addresses	172.25.50.0/24
Actions	#3 Not in the range of listed IP addresses, drop the packet and generate the selected responses

**Table 7:** Firecell example—allowing access to only a local subnet

### Monitor IP traffic that is from outside your local subnet

You can use a firecell signature to monitor, but not block, all traffic that does not originate from your local subnet. This option records or monitors the traffic using the responses you choose.

**Example:** To monitor all IP traffic that does not originate from the subnet 172.25.50.0/24, create an IP Inbound signature with the following parameters:

Parameter	Value
IP Addresses	172.25.50.0/24
Actions	#4 Not in the range of listed IP addresses, accept the packet but generate the selected responses

**Table 8:** Firecell example—monitoring access from remote subnets

### Example: Allow only typical Internet traffic

You can block all but the most common Internet TCP traffic by only allowing TCP traffic destined for ports 80 and 25. To accomplish this, you must create the three TCP Inbound signatures shown in Table 9 on page 41.



**Caution:** This example limits all TCP traffic to ports 25 and 80, which prevents the Console from communicating with a sensor that is using this policy. If you intend to use this example, make sure you create firecell rules that explicitly allow communication between the Console and the sensor before you create these signatures. If you do not, you can disable all traffic to and from a server, including the communication between the Console and the sensor.



**Reference:** For more information about using multiple signatures, refer to “Precedence of Firecell Signatures” on page 34.

Signature	Parameter	Value
1	IP Addresses	Any address (0.0.0.0/0)
	Port	80
	Actions	In the range of listed IP addresses, accept the packet, but continue to generate the selected responses
2	IP Addresses	Any address (0.0.0.0/0)
	Port	25
	Actions	In the range of listed IP addresses, accept the packet, but continue to generate the selected responses
3	IP Addresses	Any address (0.0.0.0/0)
	Port	0 (any port)
	Actions	In the range of listed IP addresses, drop the packet and generate the selected responses

**Table 9:** Firecell example—blocking all TCP traffic except for typical Internet traffic



# Configuring Suspect Connection Signatures

## Overview

### Introduction

A connection event occurs if a computer attempts to open a connection to a port on the computer that hosts RealSecure Server Sensor. Unlike other events, the sensor notifies the Console of a connection event whenever it detects that another device is attempting to open a connection, regardless of the type of activity, the type of network packets, or the content of the network packets that are exchanged.

**Example:** If you enable an FTP connection event signature, the sensor alerts the Console when it detects any FTP connection attempt, regardless of whether the content of the connection indicates an attack or other malicious behavior.

### In this chapter

This chapter contains the following topics:

Topic	Page
Monitoring for Suspect Connections	44
Monitoring for Custom Suspect Connections	46

## Monitoring for Suspect Connections

**Introduction** To monitor for suspect connections, configure the appropriate connection event signature.

**Predefined connection event signatures** The sensor policy contains predefined connection event signatures for different types of connections, such as HTTP, FTP, or Telnet.

**User-defined connection event signatures** If there is no predefined connection event signature to monitor a connection you specifically need to monitor, you can create a custom event connection signature.

**Reference:** For more information about user-defined connection event signatures, see “Monitoring for Custom Suspect Connections” on page 46

**Connection events and your policy** By default, no connection event signatures are enabled in a policy. If you want the sensor to monitor connection events, you must enable and configure (if necessary) one or more connection event signatures.

**Procedure** To configure a connection-event signature:

1. Open the policy you want to add a connection event signature to.
2. Click the **Protect** tab.
3. Double-click **Protect**→**Connections**→**Suspect Connections**.
4. Select the type of connection event signature you want to monitor.

The properties of the signature appear in the right pane.

5. Set the priority of this signature in the **Priority** box.
6. Select the responses that you want the sensor to take when it detects this type of connection.

**Reference:** For more information about each response, see the SiteProtector Help.

7. Click the **Save** icon.

The system saves your changes.

8. Apply the changed policy to the appropriate sensor(s).

**Reference:** For more information about applying policies, see the SiteProtector Help.

**Field descriptions** The following table describes the connection-event signature fields you can configure from the Security Events pane:

Field	Description
Enabled	Enables or disables the signature
Priority	Defines the severity of the event that triggers this signature.

**Table 10:** Customizable connection event signature attributes

Field	Description
Response	Configure the sensor to respond when it detects an event that matches the signature. Each signature can have any combination of responses or no response at all. <b>Reference:</b> For more information about responses, see the SiteProtector Help.

**Table 10:** Customizable connection event signature attributes (Continued)

## Monitoring for Custom Suspect Connections

### Introduction

To monitor for a suspect connection that the predefined connection event signatures do not monitor for, create a user-defined connection event signature.

### Procedure

To create a user-defined connection event signature:

1. Open the policy you want to add this signature to.
2. Click the **Protect** tab.
3. Double-click **Protect**→**Connections**→**User Defined Suspect Connections**.

All signatures in the User Defined Suspect Connections group appear.

4. Click **Add**.

The Enter a name window appears.

5. Type a name for the user-defined signature, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the User Defined Suspect Connections group.

6. In the left pane, select the signature that you just created.

The properties of the signature appear in the right pane.

7. Select the priority for this signature in the **Priority** box.
8. Type the port on the local computer that you want to monitor.
9. In the **Response** box, select the responses you want the sensor to take.

**Reference:** For more information about responses, see the Help.

10. Click **Save**.

The system saves your changes to the policy.

11. Apply the policy to the sensor(s) that you want to use the signature.

**Reference:** For the procedure on applying a policy, see the Help.

## Chapter 6

# Configuring Network Protection Signatures

## Overview

### Introduction

Network protection-based signatures monitor network traffic for content that can indicate an attack or other suspicious activity.

### In this chapter

This chapter contains the following topics:

Topics	Page
Customizing Predefined Network Signatures	48
User-Defined Network Signatures	49
Adding a User-defined Network Signature	51

## Customizing Predefined Network Signatures

### Introduction

RealSecure Server Sensor comes with predefined signatures that analyze network traffic. If the sensor detects malicious content, it can block the traffic to protect your system. You can customize certain attributes of these predefined signatures to better meet your security needs.

### Procedure

To customize a network event signature:

1. Open the policy that contains the signature you want to customize.
2. Click the **Network Events** tab.
3. Double-click the top level folder to open the list.
4. Select the signature that you want to customize.

The properties of the signature appear in the right pane.

5. In the **Priority** box, set the priority of this signature.
6. Select the responses you want the sensor to take if it detects this type of event.

**Reference:** For more information about each response, see the SiteProtector Help.

7. Click the **Save** icon.

8. Apply the new policy to the sensor(s) that you want to use this policy.

**Reference:** For more information about applying policies, see the SiteProtector Help.

### Field descriptions

The following table describes the predefined signature attributes you can customize from the Security Events pane:

Use...	To...
Enabled	enable or disable the signature
Event	determine the name of the signature
Priority	define the priority level assigned to the signature
Response	specify the response the sensor should take when the selected event occurs. Each signature can have any combination of responses or no response at all. <b>Reference:</b> For more information about responses, see the SiteProtector Help.
Advanced	open the Advanced Properties window. <b>Note:</b> This option appears only after you select a signature in the Security Events tab.
Ports	specify port numbers that you want the sensor to associate with a particular protocol <b>Note:</b> This option is only supported in version 6.x policies.
Tuning	open the Sensor Tuning window <b>Reference:</b> For more information about policy file tuning, see the <i>Server Sensor Advanced Tuning Parameters Reference Document</i> .

**Table 11:** Predefined signature field descriptions



# User-Defined Network Signatures

## Introduction

User-defined network signatures allow you to define certain network events you want the sensor to monitor for. RealSecure Server Sensor version 7.0 for Windows platforms allows you to add user-defined network signatures to your policy. This section describes how to edit sensor files to add a user-defined network signature.

## Supported user-defined network signature categories

The following table lists the user-defined network signature categories supported by RealSecure Server sensor version 7.0 for Windows platforms:

Category	Issue Number	Description
User-specified filename	2010000-2010999	Reserved for user-specified file names; triggers an intrusion detection based on a file name.
User-specified URL	2011000-2011999	Reserved for user-specified URLs; triggers an intrusion detection based on a URL name. The URL can contain a wildcard as the path, the filename, or the type; you can also specify a partial path name.
User-specified email recipient	2012000-2012999	Reserved for user-specified email recipients; triggers an intrusion detection based on the recipient's name.
User-specified email pattern	2013000-2013999	Reserved for user-specified email patterns; triggers an intrusion detection based on a pattern match with a regular expression.
User-specified MIME-attached filename	2014000-2014999	Reserved for user-specified MIME-attached filenames; triggers an intrusion detection based on a MIME-attached filename. Both incoming and outgoing email is checked for the specified filenames.
User-specified TCP probe port	2015000-2015999	Reserved for user-specified TCP probe port numbers; triggers an intrusion detection based on a failed TCP connection attempt to a particular port number.
User-specified UDP probe port	2016000-2016999	Reserved for user-specified UDP probe port numbers. By setting a configuration parameter, you can trigger an intrusion detection based on a failed UDP connection attempt to a particular port number.
User-specified registry key	2017000-2017999	Reserved for user-specified registry keys; triggers an intrusion detection based on remote access to a registry key.
User-specified TCP trojan response	2018000-2018999	Reserved for user-specified TCP trojan horse strings; triggers an intrusion detection based on a specific string seen in the first data frame of a connection.
User-specified IRC channel name	2019000-2019999	Reserved for user-specified IRC channel names; triggers an intrusion detection based on a specific IRC channel name seen.

**Table 12:** *User-defined signature categories, issue numbers, and descriptions*

Category	Issue Number	Description
User-specified Java pattern	2020000-2020999	Reserved for user-specified Java patterns; triggers an intrusion detection based on a pattern match with a regular expression.

**Table 12:** *User-defined signature categories, issue numbers, and descriptions (Continued)*

**Reference**

For information about configuration information, see the PAM Help.

**Viewing the PAM Help file**

To view the PAM Help:

1. In the SiteProtector Console, select **Help**→**Attack Signatures**→**Protocol Analysis Module**.
2. Click **Save**.
3. Browse to an appropriate location and save the file.
4. Unzip the file.
5. Open the PAM.chm file.
6. Click the advanced tuning parameters link, and then scroll down to locate the tuning parameter of interest.

# Adding a User-defined Network Signature

## Introduction

Before RealSecure Server Sensor version 7.0 for Windows platforms can monitor for a user-defined network event, you must add the user-defined network signature to the policy. You add a user-defined network signature to a policy by editing the policy.



**Caution:** Do NOT add user-defined network signature information to the `issuelist.csv` file, only edit the policy.

## User-defined network signature template

To add a user-defined network signature to a policy, you must define the signature in the policy. Use the following template as the basis for your user-defined network signature:

**Note:** You cannot edit a predefined policy. You must derive a new policy or edit a customized policy to add a user-defined network signature.

```
[\Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\];
Priority          =L          1;
CheckDescription =S          user-defined network event template;
Enabled =B          0;
Type =L          1;
TID =S          -1;
TIDDescription =S          BlackICE ID;
TValue =S          add userdefined signature configuration parameter
                    information here;
TValueDescription =S          BlackICE userdefined signature entry;
DynamicBlock =B          0;
DynamicBlockDescription =S          DynamicBlock enabled;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\];
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\BANNER\];
Enabled =B          0;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\BLOCK\];
Enabled =B          1;
Choice =S          ;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\DISABLE\];
Enabled =B          0;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\DISPLAY\];
Enabled =B          1;
Choice =S          Default;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\LOGDB\];
Enabled =B          1;
Choice =S          LogWithoutRaw;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\RSKILL\];
Enabled =B          0;
[\Advanced\userdefinedsignatures\MicroAgent
 Rules\UDnetwork_template\Response\SUSPEND\];
Enabled =B          0;
```

### Editing the policy

To edit the policy to add a user-defined network signature:

1. Locate the policy you want to add the signature to in the following directory:  
ISS/SiteProtector/Console/Server Policies
2. Right-click the policy you want to add the signature to.  
**Note:** You cannot add a user-defined network signature to a predefined policy. You must make an editable copy of a predefined policy and then customize the derived policy.  
**Reference:** For more information about making an editable copy of a predefined policy, see the SiteProtector Help.
3. Click **Open With** → **Notepad**.
4. Add the template information from “User-defined network signature template” on page 51 to the end of the policy, or copy it from the existing UDnetwork\_template in the policy file.
5. Change every instance of UDnetwork\_template to the appropriate rule name.  
**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.  
**Important:** Do not change any other information in the location path.
6. Change the CheckDescription to a descriptive definition of the signature.
7. Change the TID to the appropriate Issue ID.  
**Note:** In the template, the TID is -1. Make sure you replace the -1 with the appropriate Issue ID.
8. Change the TValue to the appropriate configuration parameter information.  
**Note:** See the PAM Help for configuration information needed for this Step.
9. Change the TValueDescription to a descriptive definition of the TValue parameters used in this signature.
10. Click **File** → **Save**.  
The User Defined Events group on the Network Events tab of the Policy Editor contains the new signature.
11. Configure the signature for use.  
**Reference:** For more information about configuring the user-defined network event signature, see “Configuring a user-defined network event signature” on page 53.

### Example

The following example shows how you might edit the user-defined network signature template to detect remote access to a registry key that includes the path /SOFTWARE/Microsoft/Windows/CurrentVersion/:

```
[\Advanced\userdefinedsignatures\MicroAgent Rules\userdefined_reg1\];  
Priority =L 1;  
CheckDescription =S userdefined network event for registry;  
Enabled =B 0;  
Type =L 1;  
TID =S 2017001;  
TIDDescription =S BlackICE ID;  
TValue =S registrykey.2017001.1=*/SOFTWARE/Microsoft/Windows/  
CurrentVersion/*;  
TValueDescription =S User-specified entry;
```

```

DynamicBlock =B      0;
DynamicBlockDescription =S      DynamicBlock enabled;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\];
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\BANNER\];
Enabled =B      0;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\BLOCK\];
Enabled =B      1;
Choice =S      Default;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\DISABLE\];
Enabled =B      0;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\DISPLAY\];
Enabled =B      1;
Choice =S      Default;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\LOGDB\];
Enabled =B      1;
Choice =S      LogWithoutRaw;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\RSKILL\];
Enabled =B      0;
[\Advanced\userdefinedsignatures\MicroAgent
  Rules\userdefined_reg1\Response\SUSPEND\];
Enabled =B      0;

```

### Configuring a user-defined network event signature

To configure a user-defined network event signature:

1. In the Sensor Policies window, select the policy that contains the user-defined network signature, and then click **Customize**.

The Policy Editor window appears with the policy open for editing.

2. Click the **Network Events** tab, and then open the **Network Events** folder.
3. Expand the **User Defined Events** group.
4. Select the signature that you want to configure.

The signature parameters appear in the right pane.

5. Click **Advanced** to change the Name/Value pairs or the event propagation settings.
6. Select the responses for this signature.
7. Select the check box for the signature to enable it.
8. Click **File** → **Save**.
9. Apply the policy to the sensor or sensors that you want to use this signature.

**Reference:** For a procedure on applying a policy, see the SiteProtector Help.



## Chapter 7

# Auditing System Integrity and Policy Compliance

## Overview

### Introduction

RealSecure Server Sensor can help you ensure system integrity and security policy compliance by auditing system log files for suspicious activity. The sensor can audit the following types of log files:

- Windows Event Logs
- any ASCII log file
- UNIX syslog messages from a local UNIX system
- C2 audit security system log files

### In this chapter

This chapter contains the following sections:

Section	Page
Section A, "Prerequisites to Configuring Audit Signatures"	57
Section B, "Configuring Predefined Audit Signatures"	65
Section C, "Configuring User-Defined Audit Signatures"	67
Section D, "Configuring User-Defined C2 Audit Signatures"	93





# SECTION A: Prerequisites to Configuring Audit Signatures

## Overview

**Introduction** Before you can audit your system, there are some prerequisites you should be aware of. This section describes the prerequisites to using log-monitoring signatures.

**In this section** This section contains the following topics:

Topic	Page
Monitoring for Audit Related Events	58
Monitoring Local Syslog Events	60
Monitoring the Mail Subsystem on HP-UX Systems	61
Prerequisite to C2 Audit Logging on an HP-UX System	62

## Monitoring for Audit Related Events

### Introduction

Almost every operating system comes with audit capabilities. The sensor audit alert system integrates with the operating system audit subsystem to ensure the system has the auditing configuration necessary to trigger the signatures you set in the sensor policy. Before the sensor can monitor for audit-related events, you must ensure that *one* of the following is true:

- the enforce audit policy setting is enabled and the policy the sensor is using has audit-related signatures enabled

**Note:** IBM ISS recommends using this method to monitor for audit-related events.

- the enforce audit policy setting is disabled, the operating system's audit feature is manually configured, and the policy the sensor is using has audit-related signatures enabled

### Default setting

By default, the enforce audit policy setting is disabled; this setting may have been enabled at any of the following times:

- during the sensor installation process
- following the sensor installation process

### Monitoring with enforce audit policy enabled

Even if the enforce audit policy setting is enabled, the sensor only detects audit-related events if audit event signatures are also enabled. You can enable audit event signatures in the following ways:

- by applying the Attacks\_And\_Audits policy, which contains enabled audit event signatures
- by applying any customized policy that has audit event signatures enabled

### Monitoring with enforce audit policy disabled

If the enforce audit policy setting is disabled, the sensor only monitors for audit-related events if you have done the following:

- correctly, *manually* configured your system's audit feature
- enabled audit-related event monitoring signatures

### When to monitor for audit-related events with enforce audit policy disabled

You would only monitor for audit-related events with the enforce audit policy setting disabled if you also wanted to manually control your audit feature settings.

### Monitoring for audit-related events

To monitor for audit-related events:

1. In the Server Sensor Properties window, select the **Server Sensor** tab.
2. Select the **Enforce audit policy** check box.
3. Apply a policy that has audit-related signatures enabled.

The sensor begins monitoring for events that rely on an audit feature setting.

---

**Stop monitoring for audit-related events on Windows and AIX platforms**

To stop monitoring for audit-related events:

1. In the Server Sensor Properties window, select the **Server Sensor** tab.
2. Clear the **Enforce audit policy** check box.
3. Do one of the following:
  - apply a policy that has audit-related signatures disabled
  - manually unset your system's audit feature

The sensor stops monitoring for events that rely on an audit feature setting.

**Note:** Disabling enforce audit policy does not unset any previously set audit feature; if you do not also disable audit-related signatures, you will still see audit-related events.

**Stop monitoring for audit-related events on HP-UX and Solaris platforms**

To stop monitoring for audit-related events:

**Note:** This procedure applies to RealSecure Server Sensor for HP-UX version 7.0, Service Release 4.1 and later and RealSecure Server Sensor for Solaris, Service Release 4.3 and later. For earlier versions of these sensors, see "Stop monitoring for audit-related events on Windows and AIX platforms."

1. In the Server Sensor Properties window, select the **Server Sensor** tab.
2. Clear the **Enforce audit policy** check box.
3. Apply a policy that has audit-related signatures disabled.

**Note:** Disabling the enforce audit policy setting restores your previously set audit feature; if you do not also disable audit-related signatures, you may still see audit-related events.

## Monitoring Local Syslog Events

### Introduction

Before the sensor can monitor for local syslog events on Unix systems, you must configure the `syslog.conf` file to indicate the location of the syslog file you want the sensor to monitor. You can configure the `syslog.conf` file in either of the following ways:

- use the `sensor.syslogfile` tuning parameter to add the path
- manually edit the `syslog.conf` file

### Configuring the `syslog.conf` file with a tuning parameter

To configure the `syslog.conf` file with the `sensor.syslogfile` tuning parameter:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `sensor.syslogfile` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
3. In the **Name** box, type the parameter name, `sensor.syslogfile`.
 

**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **String**.
5. In the **Value** box, type the path to the syslog file.
6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.
8. Click **OK**.

### Manually editing the `syslog.conf` file

To manually edit the `syslog.conf` file:

1. Open the syslog configuration file, `/etc/syslog.conf`, using a text editor.
 

**Example:** `vi /etc/syslog.conf`
2. Add the following line:
 

```
*.info    /path/messages_file
```

This identifies the path to your syslog file. The default path is `/var/adm/messages`.

**Example:** `*.info /var/adm/messages`
3. To use the new `syslog.conf` file, do one of the following:

To...	Type...
restart the syslog daemon on AIX systems	<code>refresh -s syslogd</code>
restart the syslog daemon on Solaris systems	<code>/etc/init.d/syslog stop</code> <code>/etc/init.d/syslog start</code>
restart the syslog daemon on HP-UX systems	<code>/sbin/init.d/syslogd stop</code> <code>/sbin/init.d/syslogd start</code>
reread the <code>syslog.conf</code> file	<code>kill -HUP syslogd_process_id</code>

---

# Monitoring the Mail Subsystem on HP-UX Systems

## Introduction

HP-UX systems log messages generated by the mail subsystem to `/var/adm/syslog/mail.log`. Before server sensor can monitor events generated by the mail subsystem, you must configure the mail messages to be logged to the syslog.

## Procedure

To log mail subsystem messages to the syslog:

1. Open the syslog configuration file, `/etc/syslog.conf`, using a text editor.

**Example:** `vi /etc/syslog.conf`

2. Change the following line:

```
*.info;mail.none /var/adm/syslog/syslog.log
```

to:

```
*.info /var/adm/syslog/syslog.log
```

3. To use the new `syslog.conf` file, type the following commands to restart the syslog daemon, `syslogd`:
  - `/sbin/init.d/syslogd stop`
  - `/sbin/init.d/syslogd start`

## Prerequisite to C2 Audit Logging on an HP-UX System

**Introduction** RealSecure Server Sensor relies on the HP-UX C2 auditing subsystem to trigger some operating system related events. Before the sensor can monitor C2 security audit system logs, you must configure your system.

**Process overview** The following table outlines the process for configuring your system to monitor C2 security audit system logs:

Task	Description
1	Convert the system to a trusted system.
2	Enable and configure the C2 auditing feature
3	Enable monitoring for audit related events. See "Monitoring for Audit Related Events" on page 58.

**Task 1: Converting to a trusted system** To convert the target HP-UX system to a trusted system:

1. Log on as a root user.
2. Execute SAM, and then choose the following path:

**Auditing and Security** → **Audited Events**

The system prompts you to start the conversion to a trusted system.

3. Click **Yes**.

**Reference:** For more information about how to set up a trusted system, see *HP-UX System Administration Tasks* at the following location on the Hewlett-Packard Web site:

<http://docs.hp.com/hpux/pdf/B2355-90672.pdf>

**Trusted system security policies** When you convert the target HP-UX system to a trusted system, the system enforces the following security policies:

- Every time you try to change a password (including the root password), you must enter the current password of the user.
- The default login failure count is three. After three failed login attempts, the system disables the account (including the root account). If the system disables the root account, you can only log on to the system through the system console. If you are unable to log on through the console, you must turn off the system, and then restart the system in single-user mode. Use the following command to re-convert the system from trusted mode to single-user mode:

```
# /usr/sbin/tsconvert -r
```



**Caution:** You may encounter issues if you run the tsconvert command on a trusted system.

**Task 2: Enabling and configuring the C2 audit feature**

To enable and configure the C2 auditing feature:

1. Log on using a superuser account, such as `root`.
2. Type the following command:  
`vi /etc/rc.config.d/auditing`
3. Assign the following values to the listed variables:

Variable	Value	Example
AUDITING	1	
PRI_AUDFILE	<primary audit log file name>	/.secure/etc/audfile1
PRI_SWITCH	<max log file size in KB>	1000
SEC_AUDFILE	<secondary audit log file name>	/.secure/etc/audfile2
SEC_SWITCH	<max log file size in KB>	1000

**Note:** For a complete explanation on how to use these variables, see the comments in the header of the configuration file.

4. Enter the following command to restart the system:

```
/usr/sbin/reboot
```

**Note:** If it is inconvenient to turn off and restart the system, see “Applying variable changes without turning off the system” later in this topic.

**Example**

The following example, uses 1000 as the default value for the `PRI_SWITCH` variable; the primary log file named `/.secure/etc/audfile1` grows to 1000 KB.

```
AUDITING=1
PRI_AUDFILE=/.secure/etc/audfile1
PRI_SWITCH=1000
SEC_AUDFILE=/.secure/etc/audfile
SEC_SWITCH=1000
```

**C2 audit log file rollover**

When the primary log file reaches its maximum size, audit logging switches to the secondary log file. Normally, the secondary log file grows beyond the maximum size specified in the `/etc/rc.config.d/auditing` file to the maximum size allowed by the file system’s amount of free space. However, the sensor enforces the maximum size for both log files and instructs C2 auditing to recycle and alternate the logging between the primary and secondary log files. When logging switches to the next log file, the sensor processes and purges the contents of the previous log file.

**Syslog warning messages**

When C2 audit log file switching occurs, the syslog logs the following types of warning messages:

```
Sep 28 23:53:53 SecureHpHost vmunix: Notify the security officer to specify a backup.
```

```
Sep 29 06:40:43 SecureHpHost vmunix: The current audit file is switched from /.secure/etc/audfile2 to /.secure/etc/audfile1.
```

```
Sep 29 06:40:43 SecureHpHost vmunix: Notify the security officer to specify a backup.
```

Sep 29 13:27:32 SecureHpHost vmunix: The current audit file is switched from /.secure/etc/auditfile1 to /.secure/etc/auditfile2.

**Note:** These warning messages are normal and do not require user intervention.

**Applying variable changes without turning off the system**

To apply variable changes without turning off the system:

1. Log on using a superuser account, such as **root**.
2. Use the following command to shut down the C2 audit system:  

```
# /sbin/init.d/auditing stop
```
3. Use the following command to clean up the log file control file:  

```
# rm /.secure/etc/audnames /.secure/etc/auditfile*
```
4. Use the following command to start the C2 audit system:  

```
# /sbin/init.d/auditing start
```



## SECTION B: **Configuring Predefined Audit Signatures**

### Overview

**Introduction** RealSecure Server Sensor comes with a number of predefined audit signatures with certain attributes that you can configure to meet your security needs.

**In this section** This section contains the following topic:

Topic	Page
Customizing Predefined Audit Signatures	66

## Customizing Predefined Audit Signatures

### Introduction

RealSecure Server Sensor comes with several predefined signatures that can audit system log files for suspicious activity that may indicate a threat to system integrity or a violation of your security policy.

### Procedure

To customize an audit signature:

1. Open the policy that contains the signature you want to customize.
2. Click the **OS Events** tab.
3. Double-click the top level folder to open the list.
4. Select the signature that you want to customize.

The properties of the signature appear in the right pane.

5. In the **Priority** box, set the priority of this signature.
6. Select the responses you want the sensor to take if it detects this type of event.

**Reference:** For more information about each response, see the SiteProtector Help.

7. Click the **Save** icon.

8. Apply the new policy to the sensor(s) that you want to use this policy.

**Reference:** For more information about applying policies, see the SiteProtector Help.

### Field descriptions

The following table describes the predefined signature attributes you can customize from the Security Events pane:

Use...	To...
Enabled	enable or disable the signature.
Event	determine the name of the signature.
Priority	define the priority level assigned to the signature.
Response	specify the response the sensor should take when the selected event occurs. Each signature can have any combination of responses, or no response at all. <b>Reference:</b> For more information about responses, see the SiteProtector Help.
Tuning	open the Sensor Tuning window. <b>Reference:</b> For more information about policy file tuning, see the <i>RealSecure Server Sensor Advanced Tuning Parameters Reference Document</i> .
Important Files list	view the files that the sensor monitors. <b>Note:</b> This list is only available on signatures that monitor important files.
Registry keys list	view the registry keys that the sensor monitors. <b>Note:</b> This list is only available on signatures that monitor registry keys.

**Table 13:** Predefined signature field descriptions

# SECTION C: Configuring User-Defined Audit Signatures

## Overview

**Introduction** User-defined signatures can audit any primary log source for specific data. This section describes the log sources that user-defined signatures can monitor, and also describes how to create user-defined signatures for log auditing.

**Data sources** You can create user-defined signatures to audit the following primary log sources:

- Windows Event Logs
- any ASCII log file
- UNIX syslog messages from a local UNIX system
- C2 audit security system log files

**Note:** For more information on monitoring the C2 audit security system, see Section D, "Configuring User-Defined C2 Audit Signatures" starting on page 93.

**In this section** This section contains the following topics:

Topic	Page
Using Regular Expressions in User-Defined Signatures	68
Auditing the Windows Event Log	70
Task 1: Creating a User-Defined Signature to Audit the Windows Event Log	71
Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures	73
Task 3: Identifying Relevant Windows Event Log Information	74
Task 4: Refining Your Windows Event Log User-Defined Signature	76
Auditing Files and Registry Entries with a Windows Platform Sensor	77
Monitoring a Specific Log File	81
Monitoring Custom Events in UNIX Syslogs	85
Monitoring Local Syslog Events	60
Monitoring the Wtmpx Binary Log File	87
Selecting Logs	89
Specifying Exceptions	90

## Using Regular Expressions in User-Defined Signatures

### Introduction

When you create a user-defined signature, you frequently use regular expressions to define the signature. You can use regular expressions to do the following:

- specify the information you want the agent to monitor for
- configure the information the agent retrieves about the event

Because user-defined signatures frequently use regular expressions, you should be familiar with how to use regular expressions.

### Regular expression libraries used by the sensor

The sensor uses the Henry Spencer Regular Expression Library. This is the library on which the Perl scripting language bases its syntax. If you are familiar with regular expressions in Perl, then you can apply that knowledge to the regular expressions in the sensor. The tools egrep, awk, lex and flex also use this syntax.

### Where to use regular expressions

You can use regular expressions in the following places:

- wherever you specify the event information the agent should include in responses
- wherever you define the log content to monitor for

### Specifying Name/Value pairs in the Info window

In the Info window you specify information the sensor should include in responses when the sensor detects an event that matches a user-defined signature. The info box has two parts:

- **Name**—A name, provided by you, that describes the information
- **Value**—A static value that contains information that appears in a response when the event occurs, or a regular expression that extracts a value from a string or box in the event.

### Supported data identifiers

The following table describes the data identifiers you can use in regular expressions to extract a value from an event:

Data Identifiers	Description	Example
@StringN	Used to extract a string from the string section of the event where N is the relative number of the string in the entry. When counting the number of the string, start from zero. <b>Note:</b> This identifier applies only to Windows platforms.	If the application name is in the eighth string, use the following settings for the Name/Value pair: <ul style="list-style-type: none"> <li>• Application</li> <li>• @String7</li> </ul>

Table 14: Supported data identifiers

Data Identifiers	Description	Example
@FieldN	Used to extract one information field in a log entry where N is the relative number of the field in the entry. When counting the number of the field, start from zero.	<p>A typical syslog message might have the following format:</p> <pre>Mar 15 10:25:30 everest sendmail[28244]: authdes_refresh: keyserv(1m)...</pre> <p>To extract the host name from this message, you would use the following values:</p> <ul style="list-style-type: none"> <li>• Host</li> <li>• @Field3</li> </ul>
{!}	Used to extract a substring from a string. {!} is a wildcard that pulls a substring located between two defined text entries in the entry.	<p>To capture the user name for a user with a failed login, use the following settings for the Name/Value pair:</p> <ul style="list-style-type: none"> <li>• User</li> <li>• User {!} failed to</li> </ul>

Table 14: Supported data identifiers

# Auditing the Windows Event Log

## Introduction

If the predefined Windows signatures do not audit the Windows Event Log for the specific data you need, you can create a user-defined signature to monitor any data that is recorded in the Windows Event Log.

## Process overview

The following table outlines the process, which is described in detail on the following pages, for creating a user-defined signature to audit the Windows Event Log:

Task	Description
1	Task 1: Creating a User-Defined Signature to Audit the Windows Event Log
2	Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures
3	Task 3: Identifying Relevant Windows Event Log Information
4	Task 4: Refining Your Windows Event Log User-Defined Signature

## Task 1: Creating a User-Defined Signature to Audit the Windows Event Log

### Introduction

This task describes how to create a shell signature you can use to monitor the Windows Event Log. In “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 73, you will learn how to identify the specific information in the Event Log that you want to monitor for.

### Procedure

To create a signature to monitor the Windows Event Log:

1. Open the policy you want to add this signature to.
2. Click the **OS Events** tab.
3. Double-click **OS Events**→**Windows**→**User Defined Events**→**EventLog Rules**.

All signatures that currently exist in the EventLog Rules group appear.

4. Click **Add**.

The Enter a name window appears.

5. Type a name for the user-defined signature, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the EventLog Rules group.

6. In the left pane, select the signature that you just created.

The properties of the signature appear in the right pane.

7. Complete the properties using the following table as a guide:

Option	Description
Enabled	Enables or disables the signature.
Priority	Assigns the priority for this event.
Response	Specifies responses the sensor will take when it detects an event that matches the signature. Each signature can have any combination of responses or no response at all. <b>Reference:</b> For more information about responses, see the SiteProtector Help.
Origin	Specifies the log file the event originates from. The values for this field are as follows: <ul style="list-style-type: none"> <li>• Application (Windows NT, Windows 2000, and Windows 2003)</li> <li>• Directory Service (Windows 2000 and Windows 2003)</li> <li>• DNS Server (Windows 2000 and Windows 2003)</li> <li>• File Replication Service (Windows 2000 and Windows 2003)</li> <li>• Security (Windows NT, Windows 2000, and Windows 2003)</li> <li>• System (Windows NT, Windows 2000, and Windows 2003)</li> </ul> These values match the Event Viewer values for Event Logs.

Option	Description
Source	<p>Specifies the application that generates the event. This field matches the Source column in the Event Viewer.</p> <p><b>Note:</b> If you are specifying a security/system log event, you do not have to specify the event source. However, you should always specify the Source in the application log when defining an application log event.</p>
Type	<p>Matches the Windows Event Log. There are six values for this field:</p> <p>0 = all types            1 = type of Error            2 = type of Warning            4 = type of Information            8 = Success Audit            16 = Failure Audit</p>
Category	<p>Matches the Windows Event Log category (type 0 to match all categories).</p>
ID	<p>Matches the specific Windows event ID. This can be determined by inspecting the event in the Windows Event Log Viewer.</p>
Info	<p>Specifies the information about the event that the sensor returns when an event matches the signature.</p> <p><b>Reference:</b> See “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 73.</p>
Regular Expression	<p>Matches the regular expression or simple character string against Windows Event Log messages.</p> <p><b>Note:</b> The default regular expression is a series of numbers that identify the common rule numbers used in the Windows Event Log. These numbers are a good place to start when creating a user-defined signature to monitor the Event Log.</p> <p><b>Example:</b> The regular expression Virus (Found Detected) monitors the operating system log file for the following entries:</p> <ul style="list-style-type: none"> <li>• Virus Found</li> <li>• Virus Detected</li> </ul>

8. Click **Save**.



## Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures

### Introduction

When you create a signature to monitor Windows Event Log events, you must know the kind of information the Event Log saves and on what line of the event the information appears. You can display this information on the Console or in responses. The information fields log information to the management database. This topic describes how to determine where the Event Log records the information.

### Procedure

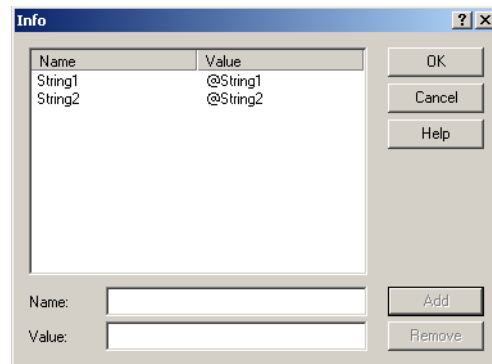
To determine which lines in an event contain the information you want to save to the database:

1. Select the signature you created in “Task 1: Creating a User-Defined Signature to Audit the Windows Event Log” on page 71.
2. Click **Info**, and then add an information field to record each line of the event in the Windows Event Log.

**Note:** Each information field you enter is just a place holder until you determine the correct name of the string displayed in the Event Log.

**Example:** To add an information field for the first line in the event (line 0), type `String 0` in the **Name** box and `@String0` in the **Value** box.

The Info window might have a list of values similar to the following example:



3. Click **OK**.
4. Click **Save**.
5. Apply the policy to a sensor you can use to test the information fields.

## Task 3: Identifying Relevant Windows Event Log Information

### Introduction

In this task you will generate the event you want to audit the Windows Event Log for so that you can match the Windows Event Log information with the information fields in the signature. This allows you to identify the relevant information from the log and define meaningful names for information fields in the Console and included in responses.

### Procedure

To identify relevant log information:

1. Generate the event you want to monitor for.  
 This triggers the signature created in “Task 1: Creating a User-Defined Signature to Audit the Windows Event Log” and generates an event in the Console.
2. In the Console, use the **Attribute Name** and **Attribute Value** columns of the Event Details to determine the relevant information fields.
3. Compare this event to the event recorded in the Windows Event Log.

### Example

The following figures show how the information fields in the SiteProtector Console that you set up in “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” match the lines of text in the Windows Event Viewer Description box. As you compare the event in the two viewers, you can see how the @StringN information fields relate to the lines in the Windows Event Log.

Figure 2 shows event details in the SiteProtector Console:

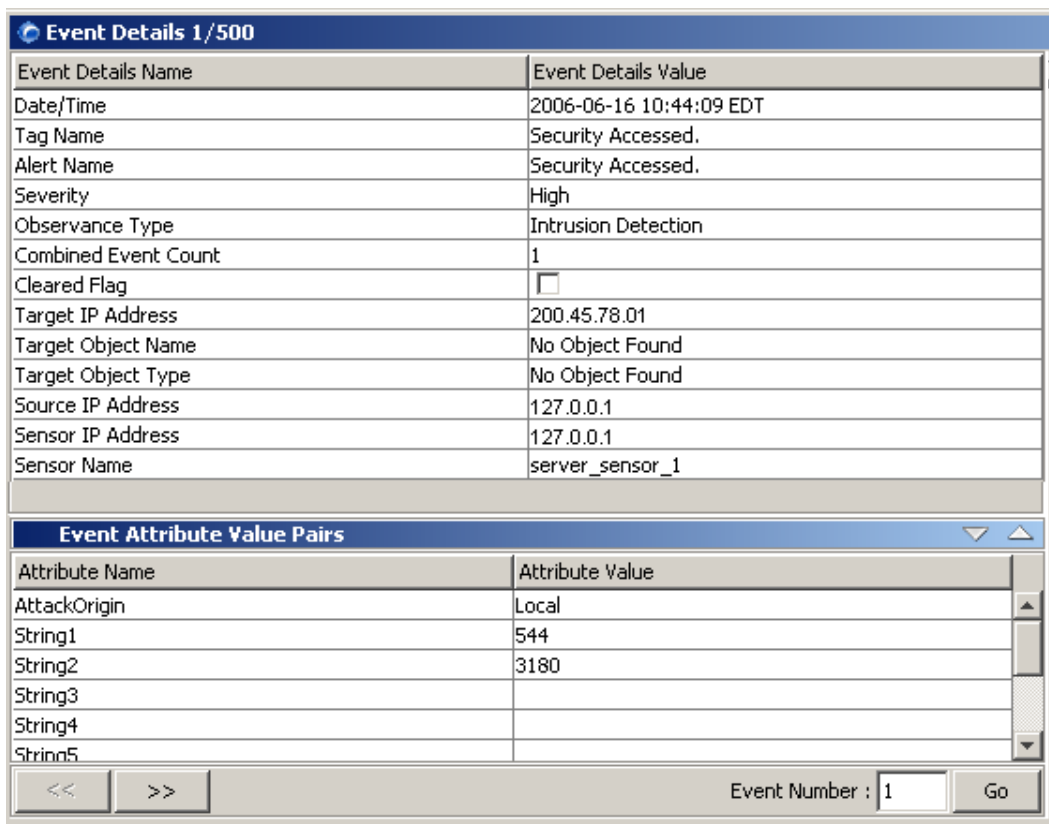


Figure 2: Sample event details

Figure 3 shows event details for the same event in the Windows Event Viewer:

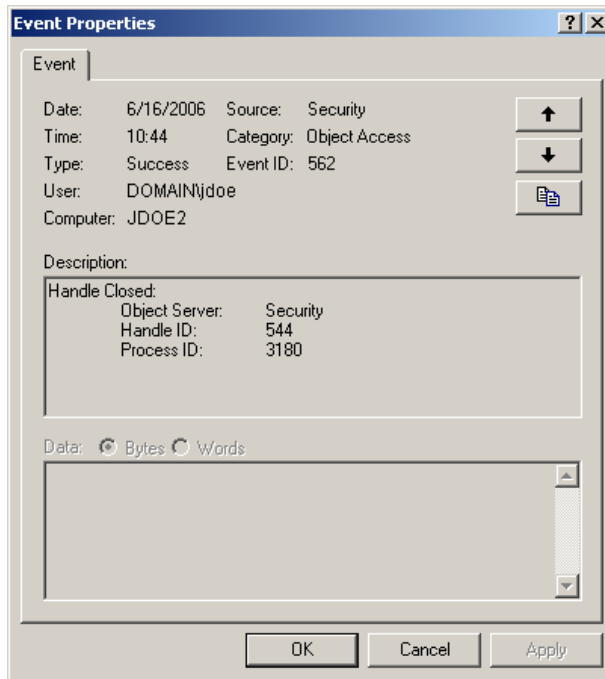


Figure 3: Sample Windows Event Viewer details

## Task 4: Refining Your Windows Event Log User-Defined Signature

### Introduction

In this final task in the process, you will refine the user-defined signature created in “Task 1: Creating a User-Defined Signature to Audit the Windows Event Log” to include the pertinent information fields from “Task 3: Identifying Relevant Windows Event Log Information.”

### Procedure

To refine the signature:

1. Select the signature you created in “Task 1: Creating a User-Defined Signature to Audit the Windows Event Log.”
2. Click **Info**, and then do the following:
  - change the **Name** in the information fields to match the real name of the string as displayed in the Windows Event Log
  - delete the information fields that do not contain information you want to see
3. Click **OK**.
4. Click **Save**.
5. Apply the policy to any sensor that should monitor for this event.

# Auditing Files and Registry Entries with a Windows Platform Sensor

## Introduction

When you audit files, you can monitor changes, such as when a user changes permission settings on a file, when a user reads a file, or when a user writes to a file. You can create a signature to monitor a file that does not yet exist because the sensor periodically searches for new files that match a file name or pattern you specify. This topic describes how to create a signature that audits files or registry entries.

## Prerequisites

Before you audit a file or registry entry, you must do the following:

- configure the sensor to monitor for audit-related events  
**Reference:** For more information, see “Monitoring for Audit Related Events” on page 58.
- look at the information in the event log when changes to the file or registry entry are made and remember the sequence number of the lines that you want to record. Count the lines starting from zero.  
**Reference:** For more information, see “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 73.

## Procedure

To create a signature to audit a file or registry entry:

1. Open the policy you want to add this signature to.
2. Click the **OS Events** tab.
3. Double-click **OS Events**→**Windows**→**User Defined Events**→**EventLog Rules**.

All signatures that currently exist in the EventLog Rules group appear.

4. Click **Add**.

The Enter a name window appears.

5. Type a name for the user-defined signature, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the EventLog Rules group.

6. In the left pane, select the signature that you just created.

The properties of the signature appear in the right pane.

7. Type or select the following information:

Box	Description
Priority	The priority that you want to assign to this signature.
Origin	The security level you want to assign to this signature.
Response	The responses the sensor should make when it detects an event that matches the signature. Each signature can have any combination of responses or no responses at all. <b>Reference:</b> For more information about responses, see the SiteProtector Help.

8. Do you want to audit a file?
  - If *yes*, go to Step 9.
  - If *no*, go to Step 11.
9. Click **Audit**.  
The Audit window appears.
10. Click the **File** tab, and then refer to the following table:

To audit...	In the File List box, type...
a directory but not the files in the directory	1. The full path to the directory. <b>Example:</b> To monitor changes to the <code>c:\temp</code> directory, type: <code>c:\temp\</code> 2. Click <b>Add</b> .
a directory recursively	1. The full path to the directory, and then ... to recursively monitor that directory. <b>Example:</b> To monitor the <code>c:\temp</code> directory recursively, type: <code>c:\temp\...</code> 2. Click <b>Add</b> . <b>Important:</b> You cannot configure the sensor to monitor the disk drive.
all files in a directory but not the directory	1. The full path to the directory, and then * to monitor all files in that directory. Files in subdirectories are not monitored. <b>Example:</b> To monitor all files in the <code>c:\temp</code> directory, type: <code>c:\temp\*</code> 2. Click <b>Add</b> .
a single file	1. The full path to the directory and the name of the file. <b>Example:</b> To monitor changes to the <code>log.txt</code> file in the <code>\temp</code> directory, type: <code>c:\temp\log.txt</code> 2. Click <b>Add</b> .
files with names that match a specific pattern	1. The pattern to match. You can use the * or the ? wildcards to define the pattern. <b>Example:</b> To monitor all files with a .log extension, type: <code>*.log</code> 2. Click <b>Add</b> .

11. Do you want to audit a registry entry?
  - If *yes*, go to Step 12.
  - If *no*, go to Step 14.
12. Click the **Registry** tab, and then select the types of changes you want to record.
13. In the **Key List** box, type the name of the registry key you want to audit.
14. Click **OK**.  
The system saves your changes and returns you to the Policy Editor.

15. Click **Info**.

The Info window appears.

## 16. Set up the information fields you want displayed on the Console and logged to the database.

**Reference:** For a detailed procedure on setting up information fields, see “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 73.

17. Click **OK**.

## 18. Do you want to use a regular expression?

- If *yes*, go to Step 19.
- If *no*, go to Step 20.

19. Type the regular expression in the **Regular Expression** box.20. Click the **Save** icon.

## 21. Apply the policy to the sensor(s) that you want to use the signature.

**Reference:** For a procedure on applying a policy, see the SiteProtector Help.

### Check boxes in the Audit window

The following table describes the check boxes in the Audit window:

Audit Type	Audit Name	Monitors attempts to...
Global	Logon/Logoff	log on or off the system.
	File/Object Access	access certain files or objects.
	User Rights	use a granted user right.
	User/Group Management	change the properties for a user or a group in a user manager.
	Security Policy	change the audit policy and a user's rights.
	Restart, Shutdown, and System	restart or shutdown the system, or otherwise affect system security or the security log.
	Process Execution	activate programs, duplicate handles, indirectly access objects, and exit processes.
File	Read	read the contents of a watched file.
	Write	edit the content of a watched file.
	Execute	run a watched file.
	Delete	remove a watched file from the system.
	Take Ownership	claim ownership of a watched file.
	Change Permissions	modify the access permissions of a watched file.

**Table 15:** *Audit window check boxes*

<b>Audit Type</b>	<b>Audit Name</b>	<b>Monitors attempts to...</b>
Registry	Query	read a key's value.
	Set Value	create or modify a key's value.
	Create Subkey	create a subkey.
	Enum Subkeys	collect key values in a list when the specific names are not known (enumerated).
	Notify	trigger a condition that results in a notification message sent to a monitored response. Notification messages of this type are sent when: <ul style="list-style-type: none"> <li>• a subkey is added or deleted</li> <li>• a value is added, changed, or deleted</li> <li>• the attributes of the key are changed</li> <li>• the key's security descriptor is changed</li> </ul>
	Create Link	create or open a registry key with the permission to create a symbolic link to the key.
	Delete	delete a registry key.
	Write DAC	determine which users and groups have access to the key (DAC - discretionary access control).
	Read Control	determine who is the key owner.

**Table 15:** *Audit window check boxes (Continued)*



# Monitoring a Specific Log File

**Introduction** If the sensor does not monitor the log files of an application or process that you want to monitor, you can create a log file signature to monitor them. These log file signatures can monitor any ASCII file.

**Prerequisites** Before you can monitor log files, you must know the following:

- how to use regular expressions and how to use exceptions in regular expressions if the information that you want the sensor to detect varies

**Reference:** For more information, see “Regular expression libraries used by the sensor” on page 68 and “Specifying Exceptions” on page 90.

- where the log files that you want to monitor reside
- the exact information in the logs you want to monitor

**Specifying newest or all files** When you monitor log files, you can specify whether all the files that match the file name pattern should be monitored, or if only the most recently changed file should be monitored. The sensor determines the newest file by comparing the last modification time (not the time the file was created).

**File rotation** When a log file grows to a certain size, the underlying application may rotate the log to a new log file. If you create a log group with a file name pattern that includes all the possible names of the log file and then set the Newest Only flag, the sensor can monitor the log file regardless of the file name.

For example, if an application writes to `/tmp/mylog1.log`, then to `/tmp/mylog2.log`, and so on, then you can specify a pattern, such as `/tmp/mylog*.log` as the filename and set the Newest Only flag to monitor the most recent log file.

**File switching** Sometimes, to retain log file information, an application renames a log file that becomes large. Subsequent log messages go to a file with the original file name, even though this is really a new file. For this situation, the agent keeps a mark for each log file it monitors. Each time a mark is changed, the agent treats it as a new file, reopens it, and monitors it from the beginning of the file. On a Windows platform, a file mark is the time the file was created. On a UNIX platform, a file mark is the inode number of the file.

**Using wildcards to select logs** When you monitor log files, you can use wildcards to specify generic log files.

Platform	Wildcards
Windows NT, 2000, or 2003	* = 0 or more characters ? = any one character
UNIX	* = 0 or more characters ? = any one character [...] = range of characters

**Table 16:** Wildcards for specifying generic log files

**Monitoring a log file** To create a signature to monitor a log file:

1. Open the policy you want to add the signature to.
2. Click the **OS Events** tab.
3. Double-click **OS Events**→**Syslog and Text Log Events**→**User Defined Events**.

All signatures in the User Defined Events group appear.

4. Click **Add**.  
The Enter a name window appears.
5. Type a name for the user-defined signature, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the User Defined Events group.

6. Select the signature that you just created.  
The properties of the signature appear in the right pane.
7. Click the ellipses [...] to select the group of logs you want to monitor with this signature.

The Logs window appears.

8. Have you created a group for the logs you want to monitor?
  - If *yes*, go to Step 11.
  - If *no*, go to Step 9.

9. Click **Add**, and then provide the following information:

Box	Description
Log Name	Type the name for this group of log files.
Log Paths	Type the full path and file name to each log in this group. Use the pipe symbol ( ) between logs. <b>Example:</b> c:\temp\logerror*.log c:\temp\log*.txt <b>Reference:</b> For information about using wildcards, see “Using wildcards to select logs” on page 81.
Monitor only most recently changed log	When checked, monitors only the newest log file. Clear the check box if you want to monitor all the log files.

10. Click **OK**.
11. In the Logs window, click the group you want to use, and then click **OK**.
12. In the Policy Editor, type or select the following information:

Box or list	Description
Priority	Select the priority that you want to assign to this event.

Box or list	Description
Regular Expression	<p>The string of text or regular expression that you want to trigger this signature.</p> <p><b>Example:</b> The regular expression [Ss]uccess   [Ff]ailure monitors the file for any of the following words:</p> <ul style="list-style-type: none"> <li>• Success</li> <li>• success</li> <li>• Failure</li> <li>• failure</li> </ul> <p><b>Reference:</b> See “Regular expression libraries used by the sensor” on page 68 and “Specifying Exceptions” on page 90 for more information.</p>
Response	<p>Configure the sensor to respond when it detects an event that matches the signature. Each signature can have any combination of responses or no responses at all.</p> <p><b>Reference:</b> For more information about responses, see the SiteProtector Help.</p>

13. Click **Info**, and then add information fields in the Info window.

**Reference:** For information about using regular expressions, see “Specifying Name/ Value pairs in the Info window” on page 68.

14. Click **Save**.

15. Apply the policy to the sensor(s) that you want to use the signature.

**Reference:** For a procedure on applying a policy, see the SiteProtector Help.

**Note:** If the log file is not present when the sensor attempts to monitor it, the sensor checks for the file every five seconds until the file exists.

### Fields in generic log file signatures

The following table describes the fields you can configure when creating a generic log file signature:

Field	Description
Enabled	Enables or disables the signature.
Event	Displays the name of the user-defined signature. This is a display-only field.
Priority	Assigns the priority for this event.
Response	<p>Configures the sensor to respond when it detects an event that matches the signature. Each signature can have any combination of responses or no responses at all.</p> <p><b>Reference:</b> For more information about responses, see the SiteProtector Help.</p>
Regular Expression	<p>Configures the signature to match specific information in the log. You can use a simple character string or a regular expression with exceptions.</p> <p><b>Reference:</b> See “Regular expression libraries used by the sensor” on page 68 and “Specifying Exceptions” on page 90 for more information.</p>
Logs	Selects the logs the sensor should monitor for this signature.

**Table 17:** Field descriptions for custom syslog and generic log signatures

**Buttons in generic log file signatures**

The following table describes the buttons in generic log file signatures:

<b>Button</b>	<b>Description</b>
Info	Specifies the information about the event that the sensor returns when an event matches the signature. <b>Reference:</b> For more information about using the Info window, see "Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures" on page 73.
Fusion Scripting	Displays Fusion scripts used by this signature. <b>Reference:</b> For more information about using Fusion Scripting, see Chapter 8, "Configuring Fusion Scripting" starting on page 115.

**Table 18:** *Button descriptions for custom syslog and generic log signatures*

# Monitoring Custom Events in UNIX Syslogs

- Introduction** You can create a custom signature if you need to monitor for an event that the predefined UNIX syslog signatures do not detect. UNIX syslog signatures monitor the local syslog and any syslogs that are forwarded to that system. This topic describes how to create a syslog signature and how to define the logs that you want to monitor.
- Prerequisite** Before you can monitor the local syslog or any syslogs forwarded to the system, you must enable logging by configuring the `syslog.conf` file. For more information, see “Monitoring Local Syslog Events” on page 60.
- Procedure** To create a signature to monitor syslog events:
1. Open the policy you want to add this signature to.
  2. Click the **OS Events** tab.
  3. Double-click **OS Events**→**Syslog and Text Log Events**→**User Defined Events**.  
All signatures in the User Defined Events group appear.
  4. Click **Add**.  
The Enter a name window appears.
  5. Type a name for the user-defined signature, and then click **OK**.  
**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.
  6. In the left pane, select the signature that you just created.  
The properties of the signature appear in the right pane.
  7. In the **Priority** box, select the priority for this signature.
  8. In the **Response** box, select the responses you want the sensor to take.  
**Reference:** For more information about responses, see the SiteProtector Help.
  9. Select the system logs that you want this signature to monitor.  
**Reference:** For a detailed procedure on selecting logs to monitor, see “Selecting Logs” on page 89.
  10. Do you want to generate a response only when certain text appears in a log?  
**Note:** The sensor will generate an event each time something is written to the log unless you specify that certain text must be present first.
    - If *yes*, go to Step 11.
    - If *no*, go to Step 12.
  11. Type the text in the **Regular Expression** box.  
**Reference:** For information about the regular expressions you can use in this column, see “Regular expression libraries used by the sensor” on page 68 and “Specifying Exceptions” on page 90.
  12. Do you want the sensor to report certain attributes?
    - If *yes*, go to Step 13.
    - If *no*, go to Step 15.

- From the left pane, select the signature you created (it appears under Log Rules), and then click **Info**.

The Info window appears.

- Set the following values to retrieve certain information:

To see this information if the event occurs...	Create an information field that uses this value...
month	@Field0
date	@Field1
time	@Field2
hostname	@Field3

- Click the **Save** icon.
- Apply the policy to the sensor(s) that you want to use the signature.  
**Reference:** For a procedure on applying a policy, see the SiteProtector Help.

# Monitoring the Wtmpx Binary Log File

- Introduction** RealSecure Server Sensor for Solaris platforms can monitor the wtmpx binary log file. To monitor this log file, create a user-defined signature.
- About the wtmpx log file** RealSecure Server Sensor provides a Binary Log Engine. The Binary Log Engine detects signatures that are based on `/var/adm/wtmpx` files on Solaris systems. The wtmpx log file resides as a binary file in the file system. The wtmpx file is updated by many processes on the operating system and records user login activities and some system process activities. The sensor spawns an “Engine thread” which is dedicated to monitoring the `/var/adm/wtmpx` file for new records.
- Activities logged in the wtmpx log** Activities logged in the wtmpx log include the following:
- user/root login using FTP
  - user/root login using telnet
  - user/root login using rlogin
  - user/root login using an X-window console
  - user/root logout
  - processes spawned by “init”
  - system boot ups
- Creating user-defined binary log signatures** To create a user-defined binary log signature:
1. Open the policy you want to add this signature to.
  2. Click the **OS Events** tab.
  3. Double-click **OS Events**→**Solaris**→**User-Defined Events**.
  4. Check the **Wtmp Events** check box.
  5. Select **Wtmp Events**.
  6. Click **Add**.  
The Enter a name window appears.
  7. Type a name for the new signature, and then click **OK**.  
**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.  
The new signature appears under **Wtmp Events**.
  8. Select the signature you just created.  
The signature parameters appear in the right pane.
  9. Set the parameters using the following fields:
    - User Name
    - Entry Type
    - Init ID (Regular Expression)**Note:** For more information about these fields, see the Help.

10. To use Info pairs, click **Info**, type the Name/Value pairs, and then click **OK**.  
**Note:** For more information about the Info window, see the Help.
11. To use Fusion scripts, click **Fusion Scripting**, type the scripts into the **Procedures**, **Initial Script**, and **Validation Script** fields, and then click **OK**.  
**Note:** For more information about the Fusion Scripting window, see the Help.
12. From the File menu, click **Save**.
13. Apply the updated policy to the sensor.

### Enabling or disabling wtmpx log monitoring

To enable or disable monitoring of the wtmpx log:

1. Open the policy that contains the wtmpx signature.
2. Click the **OS Events** tab.
3. Double-click **OS Events**→**Solaris**.
4. Double-click the **Binary Log Events** folder.
5. Double-click **Wtmp Events**.
6. Do *one* of the following:
  - To enable wtmpx monitoring, select one or more signatures.
  - To disable wtmpx monitoring, clear all the signatures.**Note:** The wtmpx monitor engine is active if any signature is selected, so to disable the engine, you must clear all the signatures.
7. Save the policy, and then apply the changed policy to the sensor.



---

## Selecting Logs

### Introduction

When you create a user-defined log signature, you can specify one or more logs for the signature to monitor.

### Procedure

To select the logs to monitor:

1. In the policy, select the signature you want to configure.  
The properties of the signature appear in the right pane.
2. Click the ellipses [...] next to the **Use logs** box.  
The Logs window appears.
3. Click **Add**.
4. In the **Log Name** box, type a name to represent the log(s) you want to use.  
**Note:** Do not use commas in the name.
5. In the **Log Paths** box, type the path to the logs.  
**Note:** Use the pipe symbol (|) to separate each log. Do not use spaces between the pipe symbol and the log path.  
**Example (UNIX log file):** `/var/adm/messages|/space/adm/messages`  
**Example (Windows):**  
`c:\log.txt|c:\Program Files\App\log.txt|c:\temp\programlog.txt`
6. Click **OK**.
7. Select the log name from the list, and then click **OK** to set up the signature to use the set of logs.

### Field and button descriptions for syslog and generic log signatures

The fields and buttons for syslog and generic log signatures are the same.

**Reference:** See Table 17, "Field descriptions for custom syslog and generic log signatures" on page 83 and Table 18, "Button descriptions for custom syslog and generic log signatures" on page 84.

## Specifying Exceptions

**Introduction** An exception is a method of excluding certain data from being processed by the system. This is like creating an exception to a rule, where the rule is a signature that you have already configured. This topic describes the following:

- types of signatures that support exceptions
- how to use exceptions with Windows Event Log signatures

**Supported exceptions** The following types of signatures support exceptions:

- Generic text log (ASCII)
- Windows Event Log

**Generic text log signatures** You create exceptions for generic text log signatures in the Regular Expression box when you configure the signature.

**Reference:** For more information about creating a generic text log signature, see “Monitoring a Specific Log File” on page 81.

**Windows Event Log signatures** For Windows Event Log signatures, you must manually edit the policy file. You cannot use the Regular Expression box to specify exceptions.



**Caution:** Manually editing policy files is not supported by IBM ISS Customer Support. If you have a problem after you have manually edited a policy file, you will have to resolve the problem on your own or start over with a working policy file.

**Prerequisites** Before you create an exception for a Windows event, you must do the following:

- know the line in the event record that contains the information that will cause the sensor to exclude that record

Windows events are parsed as values from String0 to String14. To get the proper string value, view an event in the Event Log, and count from the first line. Remember to start counting from zero.

**Reference:** For more information, see “Task 2: Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 73.

- understand the keys in the policy that cause exceptions

**Understanding keys that cause exceptions** Before you create an exception in a user-defined Windows Event Log signature or a log file signature, you must understand the syntax of the keys in the policy that control exceptions.

**Exception keys in the policy file** The sensor uses the following keys to specify exceptions. The # is replaced by a number from 0 to 14:

```
IgnoreCase# =B [1|0]; Note that there is no space in IgnoreCase.  
RegExp# =S some string to match;  
Except# =B [1|0];
```

---

**Correlating new keys in the policy with string values**

The numbers 0 to 14 correspond to the index of string in event data. In other words, if you are looking for user name guest in event id 592, the user name is in string 3, so the matching expression would be as follows:

```
IgnoreCase3 =B 1;  
RegExp3 =S guest;  
Except3 =B 0;
```

**Creating an exception to a Windows Event Log signature**

To create an exception to a Windows Event Log signature, open the policy that contains the signature, and then add the three keys that control exceptions to the signature entry.

**Example:** To find successful logins for all users except an Administrator, you would create a matching expression for event ID 528. The user name is in string 0. You would type the following into the policy file:

```
IgnoreCase0 =B 1;  
RegExp0 =S administrator;  
Except0 =B 1;
```



# SECTION D: Configuring User-Defined C2 Audit Signatures

## Overview

**Introduction** This section describes how to create and configure user-defined sensor signatures to monitor C2 security audit system logs.

**Background** RealSecure Server Sensor version 6.5 and earlier only supports user-defined audit signatures that monitor the Solaris Basic Security Module (BSM); RealSecure Server Sensor version 7.0 can monitor audit logs for other UNIX platforms. Because RealSecure Server Sensor version 7.0 can monitor audit logs for other platforms, the user-defined audit signature groups in the policy editor have been renamed `<Platform> C2Audit Rules`. This renaming applies to any sensor managed by SiteProtector version 2.0 or later. Because, on the Solaris platform, the C2 security audit system is also known as the Basic Security Module (BSM), this section refers to both C2 audit and BSM.

**Important:** You must use SiteProtector version 2.0 or later to manage version 7.0 sensors.

**Prerequisite** Before you use this section, you should have a basic understanding of the C2 audit security system. You can find information as follows:

- For detailed information about the Solaris BSM, read the *SunSHIELD Basic Security Module Guide* for your version of Solaris. You can find these guides at the following location on the Sun Web site:  
<http://docs.sun.com/app/docs?q=basic+security+module+guide>
- For detailed information about HP-UX C2 audit, see the guide *Administering Your HP-UX Trusted System* at the following location on the Hewlett-Packard Web site:  
<http://docs.hp.com/hpux/pdf/B2355-90121.pdf>
- For detailed information about AIX auditing, read Redbook #SG246020, *Auditing and Accounting on AIX*. You can find this guide at the following location on the IBM Web site:  
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246020.pdf>

**Recommendation:** If you plan to reference these documents frequently, consider downloading the portable document format (PDF) version.

**In this chapter** This chapter contains the following topics:

Topic	Page
About C2 Audit Signatures	95
Task 1: Creating a User-Defined C2 Audit Signature	96
Task 2: Generating the Event	99
Task 3: Using the Audit Log Display Command to Examine the Audit Log File	100
Task 4: Configuring the Information That Will Generate a Response	103
Task 5: Configuring the Information Fields Responses Should Return	103

<b>Topic</b>	<b>Page</b>
Task 6: Choosing Responses	106
Configuring Name Resolution for BSM Auditing	107
Configuring C2 Audit Log Management	109
Disabling Sensor Management of the BSM Log	110
Examples of User-Defined C2 Audit Information	111

# About C2 Audit Signatures

## Introduction

C2 audit user-defined signatures allow you to take advantage of C2 audit security. This topic outlines the tasks you must perform to configure a user-defined C2 audit signature.

**Important:** Review Section A, "Prerequisites to Configuring Audit Signatures" starting on page 57.

## Process overview

The following table outlines the process for creating a C2 audit user-defined signature. The remainder of this section describes this process in more detail:

Task	Description
1	Create a user-defined signature in the sensor's policy to audit the log file.
2	Generate the event to create a record in the audit file. <b>Note:</b> This task is not required on AIX systems.
3	Use the audit display command to examine the record for the following: <ul style="list-style-type: none"> <li>the information that, when present in the record, will trigger the signature</li> <li>the information fields in the audit record you want returned to you when the signature is triggered</li> </ul> <b>Note:</b> This task is not required on AIX systems.
4	Configure the sensor to send an alert when it detects the information you identified.
5	Configure the sensor to return the specified information fields when an alert is triggered.
6	Choose how you want the sensor to respond to the event.

## What happens when I create a signature?

When you create a user-defined C2 audit signature, the sensor configures the C2 auditing system to audit the events you specified in the signature. Sensors that monitor the Solaris BSM use their own audit flag to keep track of the events audited by the BSM.

## What are Solaris audit flags?

Audit flags are classes of events that the BSM monitors. For example, the `lo` flag audits events that fall under the `login_logout` class. RealSecure Server Sensor creates an `rs` flag when you install it on a Solaris computer. The BSM uses the `audit_event` file to keep track of the events that each audit flag monitors. When you add events to a signature and then update the sensor's policy, the sensor associates these events with the `rs` flag in the `audit_event` file.

**Example:** The following text shows part of an `audit_event` file. The sensor monitors the events that have an `rs` flag beside them.

```
7:AUE_EXEC:exec(2):pc, ex, rs
8:AUE_CHDIR:chdir(2):pc
9:AUE_MKNOD:mknod(2):ad, rs
10:AUE_CHMOD:chmod(2):ad, rs
11:AUE_CHOWN:chown(2):fm, rs
12:AUE_UMOUNT:umount(2) - old version:ad, rs
13:AUE_JUNK:junk:no
```

## Task 1: Creating a User-Defined C2 Audit Signature

### Introduction

To create a user-defined C2 audit signature that monitors the C2 audit system for suspicious events, you must first add a signature to the policy of the sensor. You must know the name of the events as recognized by the C2 audit system.

### In this task

This task includes the following:

- what to do if you do not know the name of the event
- how to distinguish between kernel-level and user-level events
- examples of events that you can monitor for
- how to set up the sensor policy to monitor for certain events by creating or modifying a C2 audit user-defined signature

### Determining the name of an event on Solaris and HP-UX systems

If you do not know an event name, you have the following options:

If you...	Then...	And then...
can make a guess at the name	choose one or more events listed in the Event Log ID list that appear to be the event you want to monitor for	1. Generate the event (see "Task 2: Generating the Event" on page 99). 2. Look for the event in the log files (see "Task 3: Using the Audit Log Display Command to Examine the Audit Log File" on page 100).
have no idea what the name is	choose all events listed in the Event Log ID list except for read and write calls (because they generate enormous audit logs)	3. Remove the other audit events.

**Table 19:** Options for choosing the correct event name

### Determining the name of an event on AIX systems

Use Appendix A of Redbook #SG246020, *Auditing and Accounting on AIX*, to locate the event name associated with the system activity you want to monitor for. You can find this guide at the following location on the IBM Web site:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg246020.pdf>

After you identify the event name, use the list of supported AIX events in Appendix B, "Supported C2 Audit Events" starting on page 201, to identify the information fields associated with the event. After you identify the information fields associated with the event, you can then create a signature to monitor for the event.

### Kernel-level and user-level events and records for Solaris

The names of events and records always begin with the letters AUE. The second part of the name indicates the type of event. Kernel-level events, created by system calls in the kernel, are uppercase. User-level events, created by applications outside the kernel, are lowercase.

**Examples:** The following table lists some events and describes the type of activities they audit:

Auditing this event...	Monitors...
AUE_CREATE	the creation of new files.

**Table 20:** Examples of audit flags in the Solaris BSM



Auditing this event...	Monitors...
AUE_CHMOD	mode changes in files (for example, changes in read/write permissions).
AUE_at_create	at jobs.
AUE_cron_invoke	cron jobs.

**Table 20:** Examples of audit flags in the Solaris BSM

**Reference:** For more information about events, see the *SunSHIELD Basic Security Module Guide* available at the following Web site:

<http://docs.sun.com/?q=Basic+Security+Module+Guide>

### Kernel-level and user-level events and records for HP-UX

Kernel-level events are created by system calls. The names of syscall events always begin with the letters `SYS_`. The actual system call name follows this prefix to give the full name of the system call event. TCB commands, such as `passwd`, `audsys`, and Login/Logout actions, generate user-level events.

**Examples:** The following table lists some events and describes the type of activities they audit:

Auditing this event...	Monitors...
SYS_open	the opening and/or creation of files.
SYS_chmod	mode changes in files (for example, changes in read/write permissions).
Login	login attempts.
Logout	log out attempts.

**Table 21:** Examples of audit events in the HP-UX C2 audit system

### Kernel-level and user-level events and records for AIX

On AIX systems, you cannot use the event name alone to differentiate kernel-level events from user-level events.

**Examples:** The following table lists some events and describes the type of activities they audit:

Auditing this event...	Monitors...
AUD_Bin_Def	the modification of auditbin.
FS_Rmdir	the removal of a directory object.
FS_Mkdir	the creation of a directory.

**Table 22:** Examples of audit events in the AIX C2 audit system

### Procedure

To create a user-defined C2 audit signature in the policy of the sensor:

1. Open the policy you want to add this signature to.
2. Click the **OS Events** tab.

3. Double-click **OS Events** → <platform> → **User Defined Events** → <platform> **C2Audit Rules**.

All existing signatures in the group appear.

4. Click **Add**.

The Enter a name window appears.

5. Type a name for the user-defined signature, and then click **OK**.

**Note:** Do not use a name that is already assigned to a signature; using the same name with a different case does not make the name different.

The new signature appears under the group.

6. In the left pane, select the signature that you just created.

7. Click the ellipses [...], and then select the check box for each event you want to audit for.

**Reference:** If you do not know the exact name of the event, see “Determining the name of an event on AIX systems” or “Determining the name of an event on Solaris and HP-UX systems” on page 96.

8. Click **OK**, and then click the **Save** icon.

A message tells you that you have modified the policy.

**Note:** If you are configuring an audit signature for RealSecure Server Sensor for AIX platforms, go to Step 8 of the procedure in “Task 4: Configuring the Information That Will Generate a Response” on page 103.

9. Click **OK**.

10. Click **Apply to Sensor**.

11. Click **File** → **Exit**.

The Policy Editor closes.

12. Go to “Task 2: Generating the Event” on page 99.

---

## Task 2: Generating the Event

- Introduction** To test the user-defined C2 audit signature created in “Task 1: Creating a User-Defined C2 Audit Signature” on page 96, you must generate an event so that the system creates a record in the audit file.
- Note:** This task is not required on AIX systems, go to Task 4 on page 103.
- Procedure** To generate an event:
1. Manipulate your system to generate the event.
  2. Go to “Task 3: Using the Audit Log Display Command to Examine the Audit Log File” on page 100.
- Example** You want to monitor for changes to a particular file, so you generate the event by making a copy of the file and then saving the copy over the original file. The file is the same but the change in the time and date stamp triggers an event that indicates that the file has changed.

## Task 3: Using the Audit Log Display Command to Examine the Audit Log File

### Introduction

In HP-UX systems, you use the `audisp` audit log display command to examine audit files. In Solaris systems, you use the `praudit` audit log display command to examine audit files. These commands turn the binary audit log file into readable audit records. Before you use one of these audit log display commands, you should understand the components of audit records. This topic describes audit records and also describes how to use an audit log display command to examine your audit log files.

**Note:** This task is not required on AIX systems, go to Task 4 on page 103.

### Components of HP-UX audit records

The output of an `audisp` command displays a C2 audit event record for each logged event in the C2 log file. Each event record describes attributes of the logged event. The first line of an event record contains information such as the event time, the Process ID (PID) of the process that caused the event, the error status (success/failure), the Event ID, the user id, and the group id. Subsequent lines describe the parameters for the syscall or an event specific text message.

**Reference:** For more information about the `audisp` command and to learn more about the detailed audit log display facility, see the *HP-UX Reference* for the command, available at the following location on the Hewlett-Packard Web site:

<http://docs.hp.com>

### Components of Solaris audit records

Each line of an audit record is called an audit token. Each audit token describes an attribute of the event that was recorded. Each line begins with the name of the token. The rest of the line contains other fields, separated by commas (by default), that describe that attribute of the event. All audit records contain a header token. Records can contain other tokens depending on the type of event that generated the record.

**Reference:** For an example of an audit record, see “Example: Solaris audit record” on page 102.

**Reference:** For more information about the kind of information tokens contain, see the “Audit Token Structure” section of the *SunSHIELD Basic Security Module Guide*, available at the following location on the Sun Web site:

<http://docs.sun.com/?q=Basic+Security+Module+Guide>

### Procedure

To examine the audit file:

1. Log on using a superuser account, such as **root**.
2. Stop the sensor:
  - For the HP-UX platform, type `# /sbin/init.d/realsure stop`
  - For the Solaris platform, type `/etc/init.d/realsure stop`
3. Locate the audit log file paths:
  - For the HP-UX platform, type the following:

```
# cat /.secure/etc/audnames
# ls -l `cat /.secure/etc/audnames|cut -f1 -d,`
```

- For the Solaris platform, change to the `/var/audit` directory, and then type the following:
 

```
# cd /var/audit
# ls
```
- 4. Identify the most recent audit file, and then view the last part of its contents as follows:
  - on HP-UX systems, type `# audisp [audit file name] | more`
  - on Solaris systems, type `# tail -f [audit file name] | praudit -s`
- 5. Make a note of the following information from the audit record of the event you just created:
  - for the HP-UX platform, the name of the audit event
  - for the Solaris platform:
    - the name of the audit token
    - the names of records that contain information you want to monitor for

**Note:** If you do not know the name of the records, find the audit tokens in the *SunSHIELD Basic Security Module Guide* at the following location on the Sun Web site:  
<http://docs.sun.com/app/docs?q=basic+security+module+guide>
  - for both HP-UX and Solaris platforms, the exact data within the event parameters that should trigger the signature and the name of the field that this data was in
 

**Note:** For a list of currently supported HP-UX audit record field names, see “Supported HP-UX Field Names” on page 216.

You will use this information in “Task 4: Configuring the Information That Will Generate a Response” on page 103 and in “Task 5: Configuring the Information Fields Responses Should Return” on page 105.
- 6. Restart the sensor as follows:
  - for the HP-UX platform, type `# /sbin/init.d/realsecure start`
  - for the Solaris platform, type `/etc/init.d/realsecure start`
- 7. Go to “Task 4: Configuring the Information That Will Generate a Response” on page 103.

### Example: HP-UX audit record

The following audit record is the result of a successful attempt to rename the path to `/tmp/file1`. In this example, the parameters are shown as `PARAM #1` and `PARAM #2`. You can associate the parameter name for supported syscalls with one of the field names recognized by the sensor:

```
TIME    PID    E  EVENTPPIDAIDRUIDRGIDEUIDEGIDTTY
~~~~~
021108 12:07:241657S1281575003  03 pts/ta
[ Event=rename; User=someuser; Real Grp=staff; Eff.Grp=staff; ]

RETURN_VALUE 1 = 0;
PARAM #1 (file path) = 0 (cnode);
                                0x40000003 (dev);
                                1572 (inode);
                                (path) = /tmp/file1
PARAM #2 (file path) = 0 (cnode);
                                0x40000003 (dev);
```

```
                1272 (inode);
                (path) = /tmp/file2
                ~~~~~
```

**Reference:** For a list of field names currently supported by the server sensor, see “Supported HP-UX Field Names” on page 216.

The rename syscall has the following parameters:

- source path or existing file name
- destination path or new file name

This example record indicates the following:

- Event ID (128) is SYS\_rename.
- Path field (/tmp/file1) represents the source path.
- New Path (/tmp/file2) represents the destination path.

**Example: Solaris audit record**

The following audit record is the result of someone at attacker.com modifying the default html page on a Web server:

```
header,144,2,open,,Thu Aug 26 15:21:32 1999, + 536778858 msec
path,/opt/webserver/index.html
attribute,100644,root,other,136,47043,0
subject,root,root,other,root,other,1186,1052,24 3 ip-100-195. ip-100-
195.cld.attacker.com
return,success,8
```

The kind of information you might want the sensor to detect is in the Event ID field of the header token (open, which records modifications to files) and in the path token (opt/webserver, which records any changes to files in the Webserver directory). You might want the sensor to return information in the subject.audit ID field (root) and in the subject.machine ID field (ip-100-195.cld.attacker.com).

**Reference:** For more information on how to select which information to return, see “Task 5: Configuring the Information Fields Responses Should Return” on page 105.

**Tip for identifying users:** As you have probably noticed, several fields in the subject token indicate the login name of the user. The subject.audit ID field (the second field of the subject token) is the least likely to be changed by an attacker, so add it to the list of the information to return.

## Task 4: Configuring the Information That Will Generate a Response

### Introduction

This topic describes how to configure the information that, when present in an audit record, causes the sensor to generate a response.

### Procedure

To configure information that causes the sensor to generate a response:

1. Open the policy you want to configure a response for.
2. Click the **OS Events** tab.
3. Double-click **OS Events** → <platform> → **User Defined Events** → <platform> **C2Audit Rules**.  
All existing signatures in the group appear.
4. Click the signature that contains the events you set up in “Task 1: Creating a User-Defined C2 Audit Signature” on page 96.
5. Are you configuring an audit signature for a sensor on an AIX platform?
  - If *no*, go to Step 6.
  - If *yes*, go to Step 8.
6. When you set up the signature, did you know the exact event name?
  - If *no*, go to Step 7.
  - If *yes*, go to Step 8.
7. Click the ellipses [...], and then clear the check boxes of the events you no longer want the sensor to monitor for.
8. Click **Match**.
9. Continue as described in the following table:

**Note:** For signatures used on HP-UX and Solaris platforms, add the information that you identified in Step 4 of “Task 3: Using the Audit Log Display Command to Examine the Audit Log File” on page 100.

To...	Follow these steps...
Add information that the record should contain	<ol style="list-style-type: none"> <li>1. In the <b>Name</b> box, select the field that contains the information you want the record to contain. <b>Solaris only Note:</b> If the token and field are not in the list, you can type them in using a <code>Token.field</code> format. Use the SunSHIELD documentation as a reference. <b>Example:</b> <code>path</code></li> <li>2. In the <b>Value</b> box, type the string of text that the sensor should look for in that record. <b>Example:</b> <code>opt/webserver</code> <b>Caution:</b> Wildcards and other metacharacters used in regular expressions do not work in the <b>Value</b> box. Create separate signatures if you want to use “or” logic.</li> <li>3. Click <b>Add</b>.</li> </ol>

To...	Follow these steps...
	<p>4. Repeat Steps 1 through 3 for each piece of information the record should contain.</p> <p><b>Note:</b> If you define more than one field of information, the record must match the information in each field before a match is identified. For example, if you set up a match for three fields to contain x, y, and z, the record must contain x and y and z before the match is identified. If only x or y or z are found, no match is identified.</p>
Remove information that you previously added	<ul style="list-style-type: none"><li>• Select the information, and then click <b>Remove</b>.</li></ul>

10. Click **OK**.
11. Go to “Task 5: Configuring the Information Fields Responses Should Return” on page 105 to complete the configuration of the audit signature.



---

## Task 5: Configuring the Information Fields Responses Should Return

### Introduction

After you specify the information that should generate a response, you must select the following:

- the information in the record that you want logged to the database
- the information in the record that you want returned to you in a response

### Important consideration

On Solaris systems, every audit record contains the identity of the source computer (the IP address or DNS name). You can find this information in the `Machine ID` field in the Subject token (`Subject.Machine ID`), so be sure you include this return value in the list of items to return.

### Procedure

To configure the information fields responses should return:

1. In the Policy Editor, click **Info**.  
The Info window appears.
2. From the **Value** box, select the field you want server sensor to return.  
**HP-UX example:** `Audit ID`  
**Solaris example:** `Subject.Audit User`  
**AIX example:** `Parent Process ID`
3. In the **Name** box, give the field a name.  
**Example:** `Audit User ID`
4. Repeat Step 2 and Step 3 for each field that contains useful information, and then click **OK**.
5. Go to "Task 6: Choosing Responses" on page 106.

## Task 6: Choosing Responses

### Introduction

After you configure the information fields the sensor should include in a response, you must choose the responses the sensor should take if it detects an event.

### Procedure

To choose responses for this event:

1. In the Policy Editor, select the responses you want to use for this event from the **Response** list.

**Reference:** For more information on responses, see the SiteProtector Help.

2. Click **Save**.
3. Close the Policy Editor, and then apply the policy you just modified to the sensors that monitor for the event.

# Configuring Name Resolution for BSM Auditing

## Introduction

The Sun Solaris Basic Security Module (BSM) writes user and group information to the audit log in a numerical format. By default, RealSecure Server Sensor for Solaris, Service Release 4.3 and later reports this numerical user and group information when reporting an event. You can, however, configure the sensor to resolve these numerical values to their corresponding user and group names.



**Caution:** Do not configure the sensor to resolve names on systems that use the Network Information Service (NIS). On a NIS system, configuring the sensor to resolve the numerical values to user and group names can result in the network becoming unresponsive.

## Configuring name resolution at the sensor level

To configure name resolution:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `Lookup Names` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
3. In the **Name** box, type the parameter name, `Lookup Names`.
 

**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **Boolean**.
5. In the **Value** box, do one of the following:
  - select **true** to enable name resolution
  - select **false** to disable name resolution
6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.
 

The tuning parameter with the new setting is listed in the parameters table.
8. Click **OK**.

## Configuring name resolution at the policy level

To configure name resolution:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the `Lookup Names` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
5. In the **Name** box, type the parameter name, `Lookup Names`.
 

**Note:** Any typographical errors will render the parameter unusable.

6. In the **Type** box, select **Boolean**.
7. In the **Value** box, do one of the following:
  - select **true** to enable name resolution
  - select **false** to disable name resolution
8. In the **Description** box, type a descriptive comment for this parameter.
9. Click **OK**.

The tuning parameter with the new setting is listed in the parameters table.

10. Click **OK**.
11. Save the changed policy, and then apply the policy to the sensor.

### Example

You configure the Lookup Names parameter as follows:

- Name = Lookup Names
- Value = true

The sensor will resolve user and group IDs from the BSM audit log to the corresponding user and group names.

# Configuring C2 Audit Log Management

## Introduction

You can use the sensor to help manage the size of the C2 audit log file by setting a maximum C2 audit log file size. For Solaris systems, you can also select a C2 audit log reduction action for the sensor to take if the file exceeds this file size.

## C2 audit log reduction on HP-UX systems

HP-UX platforms use primary and secondary C2 audit log files to collect C2 audit event records. When event records fill one log file, the system collects any new event records in the alternate file. After the system switches to collecting event records in the alternate log file, the sensor processes the event records in the original log file and then purges the records.

## C2 audit log reduction on Solaris systems

When the C2 audit file grows larger than the maximum C2 audit file size setting, the sensor invokes the C2 audit reduce policy. You can set the audit reduce policy as follows:

- **REMOVE** — logs new audit events to a new file and deletes the older audit file after the sensor has finished processing the file entries.
- **REDUCE** — logs new audit events to a new file and, after the sensor has finished processing the file entries, reduces the file based on the audit reduce program.  
**Note:** This action results in numerous 24-byte audit log files in the audit directory. Remove these files regularly.
- **LEAVE** — logs new audit events to a new file and leaves the older audit file on the system.

**Note:** If you select **LEAVE**, you must manage the audit files manually.

**Reference:** See “Disabling Sensor Management of the BSM Log” on page 110.

## C2 audit log reduction on AIX systems

The AIX operating system kernel records audit events in a fixed-size, circular buffer that never grows in size; therefore, it is not necessary to configure audit log management on AIX systems.

## Procedure

To configure C2 audit log management:

1. In the Sensor Properties window, select the **Server Sensor** tab.
2. In the **Maximum C2 Audit Size** box, type the maximum file size for the audit log.  
**Note:** Valid sizes are from 1 MB to 99 MB.
3. Are you configuring a Solaris server sensor?
  - If *yes*, go to Step 4.
  - If *no*, go to Step 5.
4. In the **C2 Audit Reduction** box, select the action the sensor should take when the audit file reaches the specified size.
5. Click **OK**.

## Disabling Sensor Management of the BSM Log

### Introduction

By default, RealSecure Server Sensor manages your BSM audit logs based on the audit reduce policy you configure. If you do not want the sensor to manage your logs because you already have a process for managing them, you can disable the feature.

### Disabling BSM log management at the sensor level

To disable sensor management of the BSM log:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `sensor.noc2logsizeLimit` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.  
**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
3. In the **Name** box, type the parameter name, `sensor.noc2logsizeLimit`.  
**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **Boolean**.
5. In the **Value** box, select **true** to disable sensor management of the log.
6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.  
The tuning parameter with the new setting is listed in the parameters table.
8. Click **OK**.

### Disabling BSM log management at the policy level

To disable sensor management of the BSM log:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the `sensor.noc2logsizeLimit` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.  
**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
5. In the **Name** box, type the parameter name, `sensor.noc2logsizeLimit`.  
**Note:** Any typographical errors will render the parameter unusable.
6. In the **Type** box, select **Boolean**.
7. In the **Value** box, select **true** to disable sensor management of the log.
8. In the **Description** box, type a descriptive comment for this parameter.
9. Click **OK**.  
The tuning parameter with the new setting is listed in the parameters table.
10. Click **OK**.
11. Save the changed policy, and then apply the policy to the sensor.

## Examples of User-Defined C2 Audit Information

- Introduction** This topic discusses some information that user-defined C2 audit signatures monitor, including:
- user activity
  - password changes
- Monitoring users on HP-UX systems** You can monitor the activity of a suspicious user on an HP-UX system by watching for the user's name. The user's name appears in several fields in an audit record, such as the `Real User` field, the `Effective User` field, and the `Audit User` field. If you think the user might switch to another user account, watch for the user's original login name, which is in the `Audit User` field of the record. Even if the `Real User` or `Effective User` changes, the `Audit User` will stay the same as the login user name.
- Monitoring users on Solaris systems** You can monitor the activity of a suspicious user on a Solaris system by watching for the user's name. The user's name appears in several fields of the `Subject` token, such as the `User` field, the `RealName` field, and the `Audit User` field. If you think the user might switch to another user account, watch for the user's original login name, which is in the `Audit User` field of the `Subject` token (`Subject.Audit User`). Even if the `Subject.User` changes, the `Subject.Audit User` will, in most cases, stay the same.
- Monitoring users on AIX systems** You can monitor the activity of a suspicious user on an AIX system by watching for the user's name. The user's name appears in several fields in an audit record, such as the `Real User ID` field and the `Effective User ID` field. If you think the user might switch to another user account, watch for the user's original login name, which is in the `Real User ID` field of the record. Even if the `Effective User ID` changes, the `Real User ID` will stay the same as the login user name.
- Monitoring password changes** Using the `passwd` command, a user can change an account password. An attacker can launch a brute force attack and, using the `passwd` command, gain access to the system. By monitoring the use of the `passwd` command, you can watch for attempted intrusions of this type.
- Example: Monitoring passwords on HP-UX systems** You create a user-defined signature to detect all failed password change attempts. The `PASSWORD_Change` event (Event ID# 10282) logs the use of the `passwd` command. The information you want the sensor to detect is in the `EventLog ID` section of the C2 record. The `Error Status` and the `Text` tokens indicate whether the login succeeded or failed.

**Audit record sample**—If you create a user-defined signature to detect when password change fails, a failed attempt generates an audit record like the following:

```

010628 14:49:48 1580 F 10282 1565 10 417 20 0 20 pts/tb
[ Event=admin; User=someuser; Real Grp=users; Eff.Grp=users; ]

SELF-AUDITING TEXT: User= someuser Attempt to change passwd failed

```

An audit record like this causes the sensor to generate an alert.

**Example:  
Monitoring  
passwords on  
Solaris systems**

You create a user-defined signature to detect all failed password change attempts. The `AUE_passwd` event logs the use of the `passwd` command. The information you want the sensor to detect is in the `EventLog ID` field of the `header` token. The `Return.Error` and the `Text` tokens indicate whether the login succeeded or failed.

**Audit record sample**—If you create a user-defined signature to detect when a password change fails, a failed attempt generates an audit record like the following:

```
header,87,2,AUE_passwd,Thu Aug 26 16:55:33 1999, + 16246356 msec
subject,bill,root,nogroup,bill,nogroup,1551,1538,24 3 ip-100-
  195.cld.hacker.com
text,passwd change for bill
return,success,0
```

An audit record like this causes the sensor to generate an alert.

**Example:  
Monitoring  
passwords on AIX  
systems**

You create a user-defined signature to detect all failed password change attempts. The `PASSWORD_Change` event logs the use of the `passwd` command. The information you want the sensor to detect is in the `Event ID` section of the `C2` record. The `Error Status` token indicates whether the `passwd` command succeeded or failed.

**Audit record sample**—If you create a user-defined signature to detect when a password change fails, a failed attempt generates an audit record like the following:

```
event login status time command
-----
PASSWORD_ChangerootOKFri Aug 15 19:39:25 2003 passwd root
PASSWORD_ChangerootFAILFri Aug 15 19:41:32 2003 passwd root
```

An audit record like this causes the sensor to generate an alert.



## Advanced Configuration



## Chapter 8

# Configuring Fusion Scripting

## Overview

### Introduction

Fusion scripts reduce false positives and provide comprehensive forensics data that you can use to analyze intrusions and attacks. This chapter introduces Fusion Scripting and describes how to use it.

### In this chapter

This chapter contains the following topics:

Section	Page
Section A, "Introduction to Fusion Scripting"	117
Section B, "Working with Fusion Scripting"	125



# SECTION A: Introduction to Fusion Scripting

## Overview

**Introduction** This section introduces Fusion Scripting and describes how you might use Fusion Scripting to enhance the sensor's ability to protect your system.

**In this section:** This section contains the following topics:

Topic	Page
Introduction to Fusion Scripting	118
Data Available to Fusion Scripting	120
Tcl Script Categories Used in Fusion Scripting	122

## Introduction to Fusion Scripting

**Introduction** This topic introduces Fusion Scripting and explains why you might decide to use it.

**Background** In the past, sensors have not been able to analyze events. When a packet or an entry in a log file matches a certain pattern, the sensor takes action depending on the responses set for that signature. Sometimes a simple pattern match does not provide enough information for the sensor to accurately evaluate the importance of the event, which can lead to false positives.

**When to use** Fusion Scripting enables you to use Tool Command Language (Tcl), including any scripts or programs such as PERL or C that a Tcl command can invoke, to evaluate the information the sensor collects. This ability adds an extra layer of validation and reduces false positives and prevents unwanted events from flooding the Console.

**How sensors process events without Fusion Scripting** When signatures do not use Fusion Scripting, the sensor processes the events that match the criteria of the signature. The sensor does this by detecting an event that matches a signature and then generating responses to the event, which sometimes includes sending an alert to the Console.

For sophisticated events, a simple pattern match is not sufficient to ensure that the event is significant; to verify significance, you must perform other actions. Fusion Scripting allows you to incorporate logic and validation actions into the run-time recognition of events to ensure the events are truly important.

**Process overview** The following process describes how a sensor that uses signatures with Fusion Scripting handles an event:

Stage	Description
1	When the sensor starts, Fusion Scripting initialization scripts run. These scripts typically set up variables in memory or determine some information about the server's environment.
2	When an event that the sensor monitors for matches a signature, the sensor saves the data from information fields to transient (temporary) memory. Fusion Scripting can then use the saved data associated with the signature.
3	If the sensor uses a validation script, then the sensor runs it. The validation script can use the information fields, the information produced by the initialization scripts, and any other information gathered by Fusion scripts.
4	<p>Fusion scripts then examine the data from the information fields and other system-specific data to validate the importance of the event.</p> <ul style="list-style-type: none"> <li>If the scripts find that the event is worth reporting, then they report a True value, which causes the sensor to generate any responses associated with the signature. If this signature has Fusion Scripting responses associated with it, the responses can use any of the information produced by the initialization and validation scripts.</li> <li>If the scripts find that the event is not worth reporting, then they report a False value, and the sensor drops the event from memory and does not respond to the event.</li> </ul>

## How to add Fusion Scripting to signatures

You add Fusion Scripting to user-defined signatures in the following ways:

- If you are applying Fusion Scripting to user-defined signatures on a signature-by-signature basis, use the policy editor.
 

**Note:** Some predefined signatures contain Fusion Scripting, however, while you can modify Fusion Scripting response scripts associated with predefined signatures, you cannot add new scripts to predefined signatures.

**Reference:** For specific information about adding and modifying Fusion scripts, see “Adding or Modifying a Fusion Script” on page 128.
- If you are applying a Fusion Scripting response, use the Global Responses window or the Sensor Responses window.
 

**Reference:** For specific information about adding a Fusion Scripting response, see “Configuring a Fusion Scripting Response” on page 129.

## Prerequisites

Before you add Fusion scripts, it is important to understand that IBM ISS Customer Support cannot assist you with either creating or debugging scripts. IBM ISS recommends that you make sure you are familiar with the following concepts before you work with Fusion Scripting:

Concept	References
Tcl scripting language	General Tcl reference: <a href="http://dev.scriptics.com/">http://dev.scriptics.com/</a> Practical Programming in Tcl and Tk, by Brent B. Welch ISBN: 0130220280 Tcl tutorial: <a href="http://www.msen.com/~clif/TclTutor.html">http://www.msen.com/~clif/TclTutor.html</a>
Variables, info fields, and other data used by Fusion Scripting	See “Data Available to Fusion Scripting” on page 120.
Tcl script categories used in Fusion Scripting	See “Tcl Script Categories Used in Fusion Scripting” on page 122.

**Table 23:** Prerequisites to working with Fusion Scripting



**Caution:** Fusion Scripting is an advanced feature of RealSecure Server Sensor. The Tcl scripting language can be powerful, and, if used improperly, can severely impact a system or network.

## Data Available to Fusion Scripting

**Introduction** Fusion scripts use data to execute; this data can come from various sources. This topic describes the types of variables Fusion Scripting uses and the data Fusion Scripting can use.

**Variables** Fusion scripts can create the following types of variables:

- global
- transient

**Global variables** Global variables are stored in persistent memory. Once a global variable is stored, any other script can access or change the value of the global variable. These other scripts can be in other signatures or in responses. Global variables remain in persistent memory until one of the following occurs:

- The sensor loses all variables from memory because it is shut down or restarted.
- A script deletes the global variable from memory.

Global variables must have unique names so that other scripts do not unintentionally overwrite the global variable data that the scripts contain.

**Transient variables** Transient variables are stored in transient (temporary) memory. The sensor creates transient variables from information fields when an event matches a signature. You can also create your own transient variables. The sensor can pass transient variables to responses. The time that transient variables are removed from memory depends on the result of the validation script. The validation script verifies whether the signature requires more investigation (a true result) or is not important and can be dropped (a false result). The following tables describes when transient variables are removed from memory:

Validation Script Result	Transient Variable Removed from Memory
True	after the responses associated with the signature complete
False	immediately

**Table 24:** *When transient variables are deleted from memory*

Transient variables do not have to have unique names. You can create transient variables with the same name and use them in multiple scripts without overwriting any data.

**Information fields** Fusion Scripting can retrieve, process, and reset the value of information fields sent by a sensor. When an event matches a signature, the sensor automatically saves the information fields in that event as transient variables.

**Example:** When an event matches the `Changes_to_important_files` signature, it creates the following transient variables and sets the variable value to match the data in the corresponding information field:

- File Name
- User
- User's Domain



- ClientUser
- ClientDomain
- Access Flags

You can create global variables from these information fields if you want to use any values from this event in a script associated with another signature.

**Reference:** For more information about creating global variables, see “Saving an information field to a global variable” on page 135.

### Priority level

When a sensor detects an event, it creates a variable named `__iss_priority` (two underscores in the beginning and one in the middle) and assigns a priority to that event. Fusion Scripting allows you to process or change this value.

**Reference:** For more information about changing priority levels, see “Changing the priority of an event” on page 136.

### Read-only event data values

Fusion Scripting can process certain read-only event data values sent by a sensor. You can access the following read-only data using the GetData extension:

Name	Description
<code>__iss_rulename</code>	Name of the signature being triggered.
<code>__iss_srcip</code>	The source IP address of the event in the format xxx.xxx.xxx.xxx.
<code>__iss_srcport</code>	The source port of the event.
<code>__iss_dstip</code>	The destination IP address of the event in the format xxx.xxx.xxx.xxx.
<code>__iss_dstport</code>	The destination port of the event.
<code>__iss_attacktime</code>	The time of the event in the format yyyy/mm/dd hh:mm:ss.

**Table 25:** Read-only event data values available to Fusion scripts

**Reference:** For more information about the GetData extension, see “Predefined Tcl Extensions” on page 126.

### Read/write event data values

Fusion Scripting can process certain read/write event data values sent by a sensor. You can access the following read/write data using the GetData extension:

Name	Description
<code>__iss_priority</code>	The priority of the event.

**Table 26:** Read/write event data values available to Fusion scripts

**Reference:** For more information about the GetData extension, see “Predefined Tcl Extensions” on page 126.

## Tcl Script Categories Used in Fusion Scripting

### Introduction

The sensor categorizes Fusion scripts. This topic describes the four categories of scripts that scripts associated with signatures must belong to. They are as follows:

- initialization scripts
- validation scripts
- procedure scripts
- response scripts

### Fusion Scripting process

The following table describes when the sensor would run each type of script:

When...	The sensor runs...
the sensor starts or receives a new policy (which causes the sensor to restart)	initialization scripts to create baseline global variables.
the sensor detects an event that matches a signature	validation scripts assigned to that signature to confirm that the event is a meaningful security event.
a validation script produces a true result	any responses, including response scripts assigned to that signature.
a validation script produces a false result	no responses at all and the sensor drops the event from memory.
another script calls a procedure	the procedure that was called.

**Table 27:** *Fusion Scripting process*

### Initialization scripts

Initialization scripts set global variables into persistent memory and perform other functions, such as saving a backup copy of a file, when the sensor starts or receives a new policy. The validation and response scripts can compare this data to other information sent by the sensor.

**When initialization scripts run**—When the sensor starts or loads a new policy (which causes the sensor to restart), it runs initialization scripts and loads the values of the script's global variables into persistent memory.

**Sharing**—When you add initialization scripts, you associate them with a specific signature. However, because all initialization scripts run at startup, any signature can benefit from initialization script actions regardless of whether the script sets a global variable, saves a backup copy of a file, or performs some other function.

**Example:** You can use the initialization section to collect information about an important file you want to monitor. You can create an initialization script that first calculates the checksum of the file when the sensor starts. If someone later opens the file with write access, then the sensor detects the event and can use a validation script to recalculate the checksum and compare it to the value the sensor stored before the file was opened.

In this example you could get the sensor to respond in any of the following ways:

If...	And...	Then...
the checksum values are the same	you do not want to know that the file was opened but not changed	the sensor drops the event.
the checksum values are the same	you do want to know that the file was opened but not changed	the sensor changes the event priority to low and responds.
the checksum values are different		the sensor takes the appropriate responses for this signature.

**Table 28:** *Sensor responses to an example initialization script*

**Validation scripts**

Validation scripts help you verify the importance of an event. In the validation section you set up your scripts to return a true (1) or false (0) result. If the result of a validation script is true, then the event is significant and the sensor generates the responses assigned to the signature. If the result is false, then the event is not valid or significant and the sensor drops the event from memory.



**Caution:** When the sensor returns a false result, the event is neither recorded in the sensor database nor reported in any other way. If you want the sensor to report and record events even if they do not meet the validation criteria, you can use the validation script to set the priority of the event to low and then return a true result.

**When validation scripts run**—The sensor runs validation scripts after it detects an event that matches a particular signature.

**Sharing**—Validation scripts are specific to a signature. Other scripts in other signatures and responses can use the global variables that a validation script sets or changes however. Validation scripts are well-suited to sharing information; they form the basis of content-based correlation between signatures. Information sharing can be as simple or complex as you need.

**Procedure scripts**

Procedure scripts are Tcl procedures that all initialization, validation, and response scripts in a sensor’s policy can use. You should design procedure scripts to be self-contained scripts that perform specific computational logic. IBM ISS recommends that you do not create or call variables or perform any other function in the procedure unless the variable or function is only used within the procedure itself.

**When procedure scripts run**—When the sensor starts, it loads procedures into memory but does not run them. The sensor runs a procedure script when a Fusion script calls the procedure.


**Sharing**—Any responses or Fusion scripts can use a procedure script.

**Response scripts**

The responses section contains user-defined Fusion Scripting responses.

**When response scripts run**—The sensor runs response scripts, and any other selected responses, after a validation script returns a true result.

**Sharing**—You can assign a Fusion Scripting response to any signature. The global variables that response scripts set or change are available to other scripts in other signatures and responses.

-  **Caution:** Review the function of the response in comparison to the other scripts or core functionality of the signature before you assign Fusion Scripting responses to signatures. Running too many Fusion Scripting responses at once can overload the Tcl interpreter and slow the processing of validation scripts on incoming packets.

## SECTION B: Working with Fusion Scripting

### Overview

**Introduction** This section describes some of the ways you can work with Fusion Scripting.

**In this section:** This section contains the following topics:

Topic	Page
Predefined Tcl Extensions	126
Adding or Modifying a Fusion Script	128
Configuring a Fusion Scripting Response	129
Configuring a Fusion Scripting SNMPv3 Response	130
Returning a True or False Result in a Validation Script	134
Using Fusion Scripts	135
Disabling Fusion Scripting	140

## Predefined Tcl Extensions

### Introduction

This topic describes the predefined Tcl extensions you can use with your Fusion scripts.

### Predefined Tcl extensions

RealSecure Server Sensor provides the following set of Tcl extensions:

Extension	Description
GetData	Retrieves the value of a transient variable, such as an information field in an event or the priority of an event. <b>Example:</b> To obtain the value of an information field named User Name, use the following command: <code>GetData "User Name"</code>
SetData	Creates or resets the value of a transient variable. <b>Example:</b> To create a new transient variable called Login ID, and give it the value of the current User Name variable, use the following command: <code>SetData "Login ID" [ GetData "User Name" ]</code>
UnsetData	Removes a transient variable and its value. <b>Example:</b> To remove a transient variable called Login ID, use the following command: <code>UnsetData "Login ID"</code>
Store	Creates or resets the value of a global variable. <b>Example:</b> To create a new global variable called Login ID, and give it the value of the current User Name variable, use the following command: <code>Store "Login ID" [ GetData "User Name" ]</code>
Retrieve	Recalls the value of a global variable. <b>Example:</b> To recall the value of a global variable called Login ID and to use its value to set the value of a transient variable called Login, use the following command: <code>Set "Login" [ Retrieve "Login ID" ]</code>
Remove	Removes a global variable and its value. <b>Example:</b> To remove a global variable called Login ID, use the following command: <code>Remove "Login ID"</code>

**Table 29:** *Predefined Tcl extensions*

Extension	Description
SaveArray	<p>Stores a Tcl array of values in persistent memory.</p> <p><b>Example:</b> To create an array containing checksums of a set of files (using a <code>chksum</code> procedure that calculates the sum), use a script similar to the following:</p> <pre># Set the value of namelist to equal a list of # files that are being monitored for changes in # checksum values. set namelist [list c:\\TEMP\\importantfile.txt]; # Caluculate the checksum for each file and store # the value in an array called namelist foreach f \$namelist { # Extract the next file in the namelist array set f [eval file joinfile split \$f]]; # Calculate the checksum of the file using the # chksum procedure and store the value in the # checksum array set checksum ([string tolower \$f]) [chksum \$f] } # Now that the checksum array contains the values # of all the files, store the value in an array in # global memory SaveArray checksum;</pre>
RestoreArray	<p>Retrieves a Tcl array saved previously with the <code>SaveArray</code> command.</p> <p><b>Example:</b> To retrieve the value of an array called <code>checksum</code> (previously saved using <code>SaveArray</code>) and compare its value to another variable, use the following command:</p> <pre>RestoreArray checksum if { \$checksum ( [string tolower \$fname] ) == \$sum } {return 0;} else {return 1;}</pre>
RemoveArray	<p>Deletes an entire array from persistent memory.</p> <p><b>Example:</b> To delete an array called <code>checksum</code>, use the following command:</p> <pre>Removearray checksum</pre>
GetTid	<p>Returns a unique integer. You can use <code>GetTid</code> to distinguish multiple instances of the same event.</p> <p><b>Example:</b> To create a unique name for a variable, use the following command:</p> <pre>set myUniqueName "myVar"; append myUniqueName [ GetTid ]</pre>

Table 29: Predefined Tcl extensions (Continued)

## Adding or Modifying a Fusion Script

**Introduction** You can add or modify a Fusion script used in any user-defined signature.

**Prerequisite** Create your signature before adding or modifying Fusion Scripting.

**Procedure** To add or modify a Fusion script:

1. Open the policy that contains the user-defined signature you want to modify.
2. Select the signature from the appropriate folder.  
The properties of the signature appear in the right pane.
3. Click **Fusion Scripting**.  
The Fusion Scripting window appears.
4. Type or copy and paste the scripts you want to use into the **Procedures**, **Initial Script**, and **Validation Script** boxes.  
**Reference:** For information about using these sections, see “Tcl Script Categories Used in Fusion Scripting” on page 122.
5. Click **OK**.  
The system saves the scripts.
6. Apply the modified policy to the sensor.  
**Reference:** For more information about applying policies, see the SiteProtector Help.



# Configuring a Fusion Scripting Response

## Introduction

Fusion Scripting responses allow you to use Tcl scripts to respond to events. In Fusion Scripting responses, you can use and process data, such as the data contained in information fields. This data is created when the sensor detects an event that matches a signature.

## Procedure

To configure a Fusion Scripting response:

1. From the **View** menu, select **Global Responses**.

The Global Responses window appears.

2. Double-click **Fusion Scripting**, and then select the name of the Fusion Scripting response you want to configure.
3. Modify the scripts in the script section.
4. Click **OK**.

# Configuring a Fusion Scripting SNMPv3 Response

## Introduction

The Fusion Scripting Simple Network Management Protocol response (SNMPv3) sends an SNMP version 3 trap to the specified manager. This topic describes how to configure and enable the Fusion Scripting SNMPv3 response.

## Process overview

The following process describes the steps you must take to configure and enable the Fusion Scripting SNMPv3 response:

Task	Description
1	Configure the Fusion Scripting SNMPv3 response in the Console.
2	Install and configure a trap receiver that supports SNMPv3.
3	Enable the SNMPv3 response.

## Task1: Configuring the SNMPv3 Fusion Scripting response

To configure the SNMPv3 Fusion Scripting response:

1. In the Managed Assets window, select the sensor you want to configure the SNMPv3 response for.
2. Right-click on the sensor, and then select **Responses**.  
The Sensor Responses window appears.
3. Select the **Default** response policy, and then click **Derive New**.  
The Choose Policy Name window appears.
4. Type a name for the new response policy, and then click **OK**.  
The new policy appears on the **Responses** tab.
5. Select the new response policy, and then click **Customize**.
6. Expand the **Fusion Scripting** response, and then select **SNMPv3**.

## 7. Modify the lines in the script according to the following table:

Script line	Description
<pre># set trap receiver IP address or host name set host "mytrapconsole.mydomain.com"</pre>	<p>Sets the IP address or host name of the trap receiver.</p> <p><b>Example:</b> If the trap receiver is running on a host with an IP address of 172.40.50.30, change "mytrapconsole.mydomain.com" to "172.40.50.30".</p>
<pre># set the port number trap receiver is listening on set port "162"</pre>	<p>Sets the port number the trap receiver is listening on. The standard SNMP port is 162.</p> <p><b>Example:</b> If your trap receiver is listening on port 888, change "162" to "888".</p>
<pre># set the protocol to use when sending traps set protocol "UDP"</pre>	<p>Sets the protocol to use when sending traps. The standard SNMP protocol is UDP.</p> <p><b>Example:</b> If your trap receiver is using TCP, change "UDP" to "TCP".</p>
<pre># set SNMPv3 user name set userName "mySNMPuser"</pre>	<p>Sets the user name for the SNMPv3 user.</p> <p><b>Note:</b> SNMPv3 users must be created at the trap receiver and the trap receiver must be configured to use DES for privacy and MD5 for authentication.</p> <p><b>Example:</b> If the user name you created at the trap receiver is "testuser", change "mySNMPuser" to "testuser".</p>
<pre># set SNMPv3 authentication password - must be at least 8 characters long set authPass "mySNMPpass"</pre>	<p>Sets the SNMPv3 authentication password. This password must be at least eight characters long.</p> <p><b>Example:</b> If the authentication password you have set for your user is "secure87Pass", change "mySNMPpass" to "secure87Pass".</p> <p><b>Note:</b> You can use different authentication and privacy passwords, but this is not required.</p>
<pre># set SNMPv3 privacy password - must be at least 8 characters long set privPass "mySNMPpass"</pre>	<p>Sets the SNMPv3 privacy password. This password must be at least eight characters long.</p> <p><b>Example:</b> If the privacy password you have set for your user is "secure87Pass", change "mySNMPpass" to "secure87Pass".</p> <p><b>Note:</b> You can use different authentication and privacy passwords, but this is not required.</p>
<pre># set timeout in seconds set timeout "10"</pre>	<p>Sets the number of seconds the Console should wait for a "message received" confirmation from the trap receiver.</p> <p>Only change the default if the trap receiver you are using is slow to respond.</p> <p><b>Example:</b> If you want to set the timeout period to 15 seconds, change "10" to "15"</p>

Script line	Description
<pre># set retry count set retries "1"</pre>	<p>Sets the number of times to resend the trap if a confirmation from the trap receiver does not arrive within the timeout. Set this to 0 if you do not want the trap to be re-sent when the timeout expires.</p> <p><b>Example:</b> To resend the trap 5 times, change "1" to "5"</p>



**Caution:** When using the SNMPv3 response, SNMPv3 passwords will be visible in text files on both the Console and the server sensor hosts.

8. Click **OK**.  
The Sensor Responses window closes.
9. Click **Apply to Sensor**.  
The system applies the response policy to the sensor.
10. Click **OK**.  
The Server Sensor Responses window closes.

**Task 2: Installing and configuring the trap receiver**

To configure the trap receiver:

1. Install a trap receiver that supports SNMPv3 using the documentation that came with the trap receiver.
2. Create a user and user password using the same information you specified in the following lines of the response script:
 

```
set userName "mySNMPuser"
set authPass "mySNMPPass"
set privPass "mySNMPPass"
```

**Note:** The trap receiver must be configured to use DES for privacy and MD5 for authentication.
3. Copy the `iss.mib` file to your trap receiver's MIB directory.  
Locate the `iss.mib` file in the following directories in your sensor installation directory:
 

**Windows:** `\snmp\mibs\`

**Example:** `C:\Program Files\ISS\issSensors\server_sensor_1\snmp\mibs\iss.mib`

**UNIX:** `/snmp/mibs/`

**Example:** `/opt/ISS/issSensors/server_sensor_1/snmp/mibs/iss.mib`
4. Start the trap receiver on the computer specified in the following line of the response script:
 

```
set host "mytrapconsole.mydomain.com"
```
5. Confirm the host computer is listening for traps on the port specified in the following line of the response script:
 

```
set port "162"
```

---

**Task 3: Enabling the SNMPv3 response**

To enable the SNMPv3 Fusion Scripting response for a signature:

1. Open the policy that contains the signature you want to customize.
2. Select the signature.  
The properties of the signature appear in the right pane.
3. Select the **Fusion Scripting** response.
4. Click the arrow in the Response Name column, and then select **SNMPv3**.
5. Click the **Save** icon.  
The system saves the changes you made to the policy.
6. Apply the new policy to the sensors that you want to use this policy.

**Reference:** For more information about applying policies, see the SiteProtector Help.

## Returning a True or False Result in a Validation Script

### Introduction

Use a validation script to verify whether an event is an important security event. If the event is a security event, then the validation script must return a true result before the sensor can generate a response. This topic describes how to return a true or false value so the sensor responds appropriately to an event.

### Validation script values

The following table lists the values a validation script returns depending on the result:

Result	Value to return
True	1
False	0

**Table 30:** *Returning true and false results*

**Example:** The following example shows how code can be written to return a true (1) for false (0) value. The example compares two user names recorded at different times and can check whether a user logged on as another user and might, therefore, be trying to mask their actions by using the other user name.

```
if { [string equal "Bob" $user] } {  
    return 0  
} else {  
    return 1  
}
```

---

## Using Fusion Scripts

### Introduction

There are numerous ways to use Fusion scripts to enhance the effectiveness of the protection provided by the sensor. This topic describes several ways you can use Fusion scripts and Fusion Scripting response scripts.

### Accessing an information field in validation scripts

To access an information field from an event while a validation script is running, use the `GetData` extension. Use the following syntax in the validation section to call the information field:

```
GetData "Name of information field"
```

**Example:** To set the value of a variable called `myUserName` to the value of an information field called `User Name`, type the following:

```
Set myUserName [GetData "User Name" ]
```

### Creating a new information field in a validation script

Creating a transient variable creates a new information field for an event. This information field appears as a normal information field in alerts (DISPLAY response) on the Console and in other responses. Use the following syntax in the validation section to create an information field:

```
SetData "Name of information field" "value for the variable"
```

**Example:** To create a new information field called `myIPAddress`, type the following:

```
SetData "myIPAddress" "127.31.1.4"
```

### Saving an information field to a global variable

To correlate or compare an information field to information in another Fusion script, save the information field to a global variable. Use the following syntax in any Fusion Scripting section to save an information field to a global variable:

```
Store "Name of global variable" [GetData "Name of information field" ]
```

**Example:** To create a new global variable called `myUserName` and set its value equal to the `User Name` information field, type the following:

```
Store "myUserName" [GetData "User Name" ]
```

**Example:** To set the value of the Source IP information field to `127.12.23.0`, type the following:

```
Store "Source IP" "127.12.23.0"
```

### Setting baseline information with initialization scripts

To store baseline data in persistent memory when the sensor starts, add initialization scripts in the initialization section. This data can be accessed by other scripts in signatures or in responses.

**Procedure**

To add scripts to the initialization section:

1. Open the policy that contains the user-defined signature you want to modify.
2. Select the signature from the appropriate folder.  
The properties of the signature appear in the right pane.
3. Click **Fusion Scripting**.
4. Click **OK**.  
The Fusion Scripting window appears.
5. Type or paste the scripts you want to use to calculate or create the baseline data in the **Initial Script** box.
6. Click **OK**.  
The system saves the scripts.
7. Apply the modified policy to the sensor.

**Changing the priority of an event**

You must change the priority of an event during validation, not during the response. The new priority value is then passed to the responses, which record and display the priority of the event. When the sensor detects an event, it records the event priority in a transient variable called `__iss_priority` (two underscores at the beginning and one in the middle). If the sensor changes the priority of an event, it sends an additional information field to indicate that the priority was changed. This field helps to identify signatures that have been processed with Fusion Scripting.

**Available values**—You can set the priority to either high, medium, or low.

To change the priority of an event, type the following:

```
SetData "__iss_priority" "priority"
```

**Example:** To change the priority to high, type the following:

```
SetData "__iss_priority" "High"
```

**Passing information to Fusion Scripting responses**

By default, Fusion Scripting responses can use any data held in transient or persistent memory, which means that they can use any existing transient or global variables.

**Example:** For an example of passing a variable created in a validation script to a response script, see “Monitoring trusted users” on page 138.

**Correlating information between signatures**

To correlate information collected by one signature with information in another signature, use global variables.

**Example:** The following script demonstrates how you can use information from one event to set the priority of a second, related event. Normally, the `Startup_of_important_programs` signature has a low priority. This example monitors for two correlated events—a user attempting to make changes to a file, and then attempting to run a file. The following script resets the priority of the `Startup_of_important_programs` to High:

```
#Validation script for "Changes_to_important_files"  
# First retrieve the current user name  
set current_user [ GetData "User" ]
```



```

# Store it for later use by another signature
Store "Suspect_User" $current_user
return 1

#Validation script for "Startup_of_important_programs"
#First get the current user name
set current_user [ GetData "User" ]
# Now retrieve the stored user name from the other event
set stored_user [ Retrieve "Suspect_User" ]
# Do the comparison
if { [ string equal $current_user $stored_user ] } {
    # If the two users match, then change the priority to
    # high.
    SetData __iss_priority "High"
    return 1
} else {
    # If the users are different, keep the priority as low.
    SetData __iss_priority "Low"
    return 1
}

```

### Saving variables to a log file

If you want to save the value of a variable to a log file, use the `puts` command. This command saves the variable to a log file called `tclproc1.log`, which resides in the sensor installation directory. This file contains a log that details how the Tcl subprocess interacts with the sensor.

**Example:** To print the value of a variable called `myvar` into a log file, use the following code:

```
puts myvar = "$myvar\n"
```

### Monitoring an index.htm file on a Web server

You can use Fusion scripts to monitor and restore files. For example, you could use Fusion scripts to monitor whether an attacker changed the main page of your company Web site and then restore the file if the page was changed. The following table outlines how the sensor would monitor for this type of attack:

Stage	Description
1	When the sensor starts, it runs an initialization script that creates a backup copy of the main Web page, <code>index.html</code> . The script names the backup file <code>backup.html</code> and saves it to a hidden directory.
2	Someone (possibly an attacker) makes an unauthorized change to the main page of the Web site by replacing the <code>index.html</code> file with a file that contains unauthorized content.
3	The sensor signature that monitors changes to <code>index.html</code> detects the event and starts the validation scripts associated with that signature to verify whether the file changed.

**Table 31:** *Monitoring for an attack on a Web server*

Stage	Description
4	The validation script compares the size of the modified <code>index.html</code> file against the hidden <code>backup.html</code> file. If the files are not the same, the script returns a true (1) result, which indicates that the file was changed and this is a valid security event.  <b>Note:</b> Although this example uses a simple file size comparison to check the files, more secure methods of file comparisons are possible, such as calculating checksums (on UNIX) or calculating digital signatures for each file. These examples are much more complex than a simple size comparison and are beyond the scope of this documentation.
5	Because the result is true, the sensor generates all responses assigned to that signature. One response sends an email to the Web master to report that someone has made an unauthorized change to <code>index.html</code> . Another response starts a Fusion script that automatically copies <code>backup.html</code> to <code>index.html</code> to restore the main page of the Web site.

**Table 31:** *Monitoring for an attack on a Web server (Continued)*

**Example Initialization script:**

```
file copy -force "c:/wwwdir/index.htm" "c:/hidndir/backup.htm"
```

**Example Validation script:**

```
set fSrc [file
set fDest [file size "c:/hidndir/backup.htm"]
if { $fSrc == $fDest } {
# The file has not changed. Do not generate responses.
return 0
} else {
# The file has changed. Generate responses.
return 1
}
```

**Example Response script:**

```
# replace modified file with original file
file copy -force "c:/hidndir/backup.htm" "c:/wwwdir/index.htm"
```

**Monitoring trusted users**

You can use Fusion scripts in the `Logon_withadmin_privileges` signature to alert you when users that are not trusted attempt to log on to a system. Use Fusion Scripting to perform the following functions:

- create a list of users you trust
- be sure the users who attempt to log on are on this list
- generate a response if a user that is not on the list attempts to log on

**Example:** The following scripts show how you could implement this:

```
# Logon_withadmin_privileges

# In the initialization section of the Logon_withadmin_privileges
# signature, create a variable that contains a list of trusted users.

# Initialize = S
Store __iss_trust_usr_list [list "Bob"]

# In this case, Bob is the only person with administrative privileges
```

```
# on the system that the server sensor monitors. When Bob logs in with
# admin privileges, no action is required, because this activity is
# normal. If anyone besides Bob logs in with admin privileges, then
# there is cause for concern and the sensor will generate responses
# for that event.

# In the validation section, the following script compares each user
# that attempts to log in with the trusted user list.

# Validation = S

if { [catch { Retrieve "__iss_trust_user_list" } trust_user_list] } {
    set trust_usr_list ""
}
set user [GetData "User"]

if { $tcl_platform(platform) == "unix" } {

foreach i2User $trust_usr_list {
    if { [string equal $user $i2User] } {
#       If the user is on the trusted list, the script drops the
#       event by returning a false (0) result.
        return 0
    }
}

#       If the user is not on the trusted user list, then this is
#       an important security event. The sensor returns a true
#       (1) result and generates all associated responses.
return 1
```

## Disabling Fusion Scripting

### Introduction

You can disable Fusion Scripting on a signature-by-signature basis or you can prevent all Fusion scripts from running on a particular sensor. This topic describes how to implement both methods.

**Important:** Disabling all Fusion scripting disables the Buffer Overflow Exploit Protection component of the sensor.



**Caution:** Signatures can be based purely on Fusion Scripting. If you disable Fusion Scripting for one of these signatures, then the signature stops working.

### Disabling Fusion Scripting for one signature

You can disable all Fusion Scripting functionality associated with a particular signature. Disabling Fusion Scripting for a particular signature does the following:

- disables all Fusion Scripting initialization scripts, validation scripts, and procedures
- prevents the signature from running any Fusion Scripting response associated with the signature
- disables the signature, if it is based purely on Fusion Scripting



**Caution:** If you disable Fusion Scripting for a particular signature, scripts in the procedure section of the signature are unavailable to other scripts. If you use the scripts in the procedure section of this signature in other scripts, copy the procedures to the procedure section of a signature that has Fusion Scripting enabled.

### Disabling Fusion Scripting

To disable Fusion Scripting for one signature:

1. Open the policy that contains the signature you want to modify.
2. Select the signature from the appropriate folder.  
The properties of the signature appear in the right pane.
3. Click **Fusion Scripting**.  
The Fusion Scripting window appears.
4. Clear the **Enable Fusion Scripting** check box.
5. Click **OK**.
6. Apply the modified policy to a sensor.

### Disabling all Fusion scripts

For troubleshooting purposes, you can turn off all Fusion Scripting functionality on a sensor. Disabling Fusion scripts disables Fusion Scripting functionality as follows:

- Fusion scripts in standard and user-defined signatures
- Fusion Scripting responses
- any signature that is based totally on Fusion scripts

**Disabling all Fusion Scripting**

To disable all Fusion Scripting on a sensor:

1. In the Sensor Properties window, select the **Server Sensor** tab.
2. Clear the **Execute Fusion Scripts** check box, and then click **OK**.

The sensor no longer runs any Fusion scripts, responses, or signatures that are totally based on Fusion Scripting.



## Chapter 9

# Capturing Packet Information

## Overview

### Introduction

RealSecure Server Sensor can capture information about packets received by the system. The sensor uses the following types of logging to collect this information:

- packet logging
- evidence logging

**Note:** Evidence logging is currently supported only on Windows platforms.

### In this chapter

This chapter contains the following topics:

Topics	Page
About Packet Logging	144
Enabling Packet Logging	145
About Evidence Logging	147
Enabling Evidence Logging	148

## About Packet Logging

### Introduction

Packet logging creates a copy of every packet that arrives at the system. Packet logs can become very large and use a lot of hard disk space; however, they gather valuable information about activity on the system.

### Packet logging tuning parameters on Windows platforms

The following table lists the packet logging tuning parameters for Windows platforms:

Parameter	Value	Description
packetLog.logging	True	Enables packet logging
packetLog.logging	False	Disables packet logging

**Table 32:** Packet logging Name/Value pairs

**Note:** The default packet logging value is `False`. When you import policies from an earlier version sensor, the sensor does one of the following:

- retains any defined packet logging value
- sets the packet logging value to `False`, if the policy did not define a value

### Packet logging tuning parameters on Unix platforms

The following table lists the packet logging tuning parameters for Unix platforms:

Parameter	Value	Description
packetlog.enabled	True	Enables packet logging
packetlog.enabled	False	Disables packet logging

**Table 33:** Packet logging Name/Value pairs

**Note:** The default packet logging value is `False`.

### Packet logging log file naming convention

The sensor uses the following naming conventions for packet log files.

#### **RealSecure Server Sensor for Windows version 7.0, Service Release 4.2 and later—**

Packet logging creates a maximum of ten, 1400KB files for each NIC on the computer. The log files are named `pkt_macaddressxxx.enc`, where `xxx` is 000 through 009. The sensor creates these files in the `server_sensor_1/Logs` directory.

#### **RealSecure Server Sensor for Windows version 7.0, Service Release 4.1 and earlier—**

Packet logging creates a maximum of ten, 10MB files named `logxxx.enc`, where `xxx` is 000 through 009. The sensor creates these files in the `server_sensor_1/BlackICE` directory.

**RealSecure Server Sensor for Unix platforms—**Packet logging creates a maximum of ten, 1400KB files named `Logxxxx.enc`, where `xxxx` is 0000 through 0009. The sensor creates these files in the `/opt/ISS/issSensors/server_sensor_1` directory.



## Enabling Packet Logging

### Introduction

You can configure packet logging at either the policy level or at the sensor level. If you specify one value for packet logging at the policy level and another value for the same parameter at the sensor level, the value specified at the sensor level overrides the value specified at the policy level. This behavior is beneficial if you want a small number of sensors to behave differently from the majority of your sensors.

### Enabling packet logging at the sensor level

To enable packet logging at the sensor level:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you want to edit a tuning parameter that is already in the list?
  - If *yes*, select the tuning parameter to edit, click **Edit**, and then go to Step 5.
  - If *no*, click **Add**.

The Advanced Tuning Value window appears.

3. Continue according to the follow table:

Field	Description
Name	Type the name of the tuning parameter as follows: <ul style="list-style-type: none"> <li>• <code>packetLog.logging</code> for packet logging on Windows platforms</li> <li>• <code>packetlog.enabled</code> for packet logging on Unix platforms</li> </ul>
Type	Select <b>boolean</b> .
Value	Select <b>True</b> .
Description	Type a description that indicates the purpose of this tuning parameter.

4. Go to Step 6.
5. Select **True**.
6. Click **OK**.
7. Click **OK**.

### Enabling packet logging at the policy level

To enable packet logging at the policy level:

1. Open the policy you want to customize.
2. On the Network events tab, select any group of signatures.
3. Click **Tuning**.

The Sensor Tuning window appears.

4. Do you want to edit a tuning parameter that is already in the list?
  - If *yes*, select the tuning parameter to edit, click **Edit**, and then go to Step 7.
  - If *no*, click **Add**.

The Advanced Tuning Value window appears.

5. Continue according to the follow table:

<b>Field</b>	<b>Description</b>
Name	Type the name of the tuning parameter as follows: <ul style="list-style-type: none"><li>• <code>packetLog.logging</code> for packet logging on Windows platforms</li><li>• <code>packetlog.enabled</code> for packet logging on Unix platforms</li></ul>
Type	Select <b>boolean</b> .
Value	Select <b>True</b> .
Description	Type a description that indicates the purpose of this tuning parameter.

6. Go to Step 8.
7. Select **True**.
8. Click **OK**.
9. Click **OK**.
10. In the Policy Editor window, click **Save**, and then apply the changed policy to the sensor.

## About Evidence Logging

### Introduction

Evidence logging creates a copy of a packet that triggers an event. Evidence logs show exactly what the intruder did or attempted to do.

**Note:** Evidence logging is currently supported only on Windows platforms.

### Evidence logging tuning parameters

The following table lists the evidence logging tuning parameters:

Parameter	Value	Description
evidence.logging	True	Enables evidence logging
evidence.logging	False	Disables evidence logging

**Table 34:** Evidence logging Name/Value pairs

**Note:** The default evidence logging value is `True`. When you import policies from an earlier version sensor, the sensor does one of the following:

- retains any defined evidence logging value
- sets the evidence logging value to `True`, if the policy did not define a value

### Evidence logging log file naming convention

RealSecure Server Sensor uses the following naming conventions for evidence log files.

**RealSecure Server Sensor for Windows version 7.0, Service Release 4.1 and earlier—**  
Evidence logging creates up to 32 evidence files named `evdxxx.enc`, where `xxx` is 000 through 031. The sensor creates these files in the `server_sensor_1/BlackICE` directory.

**RealSecure Server Sensor for Windows version 7.0, Service Release 4.2 and later—**  
Evidence logging creates up to 32 evidence files named `evd_macaddressxxx.enc`, where `xxx` is 000 through 031. The sensor creates these files in the `server_sensor_1/Logs` directory.

## Enabling Evidence Logging

### Introduction

You can configure evidence logging at either the policy level or at the sensor level. If you specify one value for evidence logging at the policy level and another value for the same parameter at the sensor level, the value specified at the sensor level overrides the value specified at the policy level. This behavior is beneficial if you want a small number of sensors to behave differently from the majority of your sensors.

**Note:** Evidence logging is currently supported only on Windows platforms.

### Enabling evidence logging at the sensor level

To enable evidence logging at the sensor level:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you want to edit a tuning parameter that is already in the list?
  - If *yes*, select the tuning parameter to edit, click **Edit**, and then go to Step 5.
  - If *no*, click **Add**.

The Advanced Tuning Value window appears.

3. Continue according to the follow table:

Field	Description
Name	Type <code>evidence.logging</code> .
Type	Select <b>boolean</b> .
Value	Select <b>True</b> .
Description	Type a description that indicates the purpose of this tuning parameter.

4. Go to Step 6.
5. Select **True**.
6. Click **OK**.
7. Click **OK**.

### Enabling evidence logging at the policy level

To enable evidence logging at the policy level:

1. Open the policy you want to customize.
2. On the Network events tab, select any group of signatures.
3. Click **Tuning**.
 

The Sensor Tuning window appears.
4. Do you want to edit a tuning parameter that is already in the list?
  - If *yes*, select the tuning parameter to edit, click **Edit**, and then go to Step 7.
  - If *no*, click **Add**.

The Advanced Tuning Value window appears.

5. Continue according to the follow table:

Field	Description
Name	Type <code>evidence.logging</code> .
Type	Select <b>boolean</b> .
Value	Select <b>True</b> .
Description	Type a description that indicates the purpose of this tuning parameter.

6. Go to Step 8.
7. Select **True**.
8. Click **OK**.
9. Click **OK**.
10. In the Policy Editor window, click **Save**, and then apply the changed policy to the sensor.



## Chapter 10

# Monitoring for Buffer Overflow Exploits

## Overview

### Introduction

Buffer Overflow Exploit Protection (BOEP) can identify an intruder's attempts to exploit buffer overflow vulnerabilities and gain access to your computer. When the sensor detects a buffer overflow exploit, it identifies the program associated with the exploit, and then takes the specified action.

**Important:** BOEP is only available with RealSecure Server Sensor for Windows version 7.0, Service Release 4.2 or later on Windows 2000 and 2003 (including Service Release 1) platforms.

### In this chapter

This chapter contains the following topics:

Topic	Page
About Buffer Overflow Exploit Protection	152
Enabling Default Buffer Overflow Exploit Protection	153
Monitoring Additional Directories	154
Changing the Action for All Monitored Directories	156
Changing the Action for a Specific Program File	159

## About Buffer Overflow Exploit Protection

<b>Introduction</b>	<p>Certain applications are vulnerable to buffer overflow exploits. When RealSecure Server Sensor detects a buffer overflow exploit, it identifies the application associated with the exploit and then takes the specified action.</p> <p><b>Note:</b> RealSecure Server Sensor prevents exploits based on buffer overflows; it does not prevent buffer overflows themselves.</p>
<b>Actions available</b>	<p>Actions available to the sensor are as follows:</p> <ul style="list-style-type: none"><li>● ignore the buffer overflow</li><li>● terminate the entire process that caused the buffer overflow</li><li>● abort (fail) the operation within the process where the buffer overflow was detected</li><li>● report the overflow to the Console</li></ul> <p>By default, the sensor aborts (fails) the operation within the process where the buffer overflow was detected and then sends an event to the Console.</p>
<b>Monitored files</b>	<p>By default, the sensor monitors all program files in the %systemroot% directory and its subdirectories. The sensor also looks in the registry for the location of program files and monitors those files.</p> <p><b>Note:</b> The following file contains a list of the directories and executables the sensor is monitoring:</p> <pre><i>sensor_install_dir\vpatch\boep.ini</i></pre>
<b>Customizing protection</b>	<p>You can monitor any directory on your computer or any mapped or UNC network paths for buffer overflow exploits. You can also exclude any program file within a monitored directory from buffer overflow protection.</p>
<b>Buffer overflow protection at the sensor and policy levels</b>	<p>Using advanced tuning parameters, you can configure BOEP at either the policy level or at the sensor level. If you specify one value for buffer overflow protection at the policy level and another value for the same parameter at the sensor level (on the Sensor Properties page), the value specified at the sensor level overrides the value specified at the policy level. This level of configurability allows you to set the behavior for a large number of sensors at the policy level but configure a small number of sensors to behave differently by setting sensor-level parameters.</p>
<b>Buffer overflow protection and the Security Fusion Module</b>	<p>The Security Fusion Module does not include BOEP events in its analysis of intrusion attempts against your system.</p>



# Enabling Default Buffer Overflow Exploit Protection

**Introduction** When you enable BOEP, the sensor provides default protection against buffer overflow exploits.

**Default protection** Default Buffer Overflow Exploit Protection does the following:

- monitors directories and program files listed in `sensor_install_dir\vpatch\boep.ini`
- aborts (fails) the operation within the process where the buffer overflow was detected and then sends an event to the Console

If the default configuration does not meet your security needs, use the information in this section to reconfigure the settings.

**Process overview** The following table outlines the process for enabling Buffer Overflow Exploit Protection:

Task	Description
1	Enable Fusion Scripting
2	Enable Buffer Overflow Exploit Protection in the policy

## Task 1: Enabling Fusion Scripting

To enable Fusion Scripting:

1. In the Sensor Properties window, select the **Server Sensor** tab.
2. Select the **Execute Fusion Scripting** check box.
3. Click **OK**.
4. Restart the sensor.

## Task 2: Enabling BOEP in the policy

To enable Buffer Overflow Exploit Protection in the policy:

1. Open the policy you want to customize.
2. On the **OS Events** tab, expand **OS Events**.
3. Double-click **Windows** → **Service Events** → **Buffer Overflow Protection**.
4. Select the **BOEP** check box.
5. Select the responses you want the sensor to take if it detects this type of event.

**Reference:** For more information about each response, see the SiteProtector Help.

6. Click **Save**, and then apply the changed policy to the sensor.

## Monitoring Additional Directories

### Introduction

You can configure the sensor to monitor additional directories by adding directories to an include list. Customize the protection provided by each sensor by defining additional directories as follows:

- At the policy level, where all sensors that use the policy monitor the directory.
- At the sensor level, where only the sensor you configure monitors the directory.

### Prerequisite

You must enable Buffer Overflow Exploit Protection before the sensor can monitor any additional directories.

### Adding a directory to monitor at the policy level

To add a directory to monitor at the policy level:

1. Open the policy you want to customize.
2. On the **OS Events** tab, expand **OS Events**.
3. Double-click **Windows** → **Service Events**.
4. Select **Buffer Overflow Protection**.
5. Click **Tuning**.

The Sensor Tuning window appears.

6. Click **Add**.

The Advanced Tuning Value window appears.

7. Continue according to the following table:

Field	Description
Name	Type <b>vpatch.Include.name_of_directory</b> . Where <i>name_of_directory</i> is any unique word that identifies the directory to monitor. <b>Example:</b> vpatch.Include.MyDirectory
Type	Select <b>String</b> .
Value	Type <i>directory_to_protect</i> <b>Note:</b> You can include a local drive, a mapped network drive, or a UNC drive in the value. If you do not specify a drive, the sensor monitors all drives for the specified directory.
Description	Type a description that indicates the purpose of this parameter.

8. Click **OK**.
9. Click **OK**.
10. In the Policy Editor window, click **Save**, and then apply the changed policy to the sensor.

### Adding a directory to monitor at the sensor level

To configure buffer overflow protection at the sensor level:

1. In the Sensor Properties window, select the **Advanced** tab.
2. Click **Add**.

The Advanced Value window appears.

3. Continue according to the following table:

Field	Description
Name	Type <code>vpatch.Include.name_of_directory</code> . Where <code>name_of_directory</code> is any unique word that identifies the directory to monitor. <b>Example:</b> vpatch.Include.MyDirectory
Type	Select <b>String</b> .
Value	Type <code>\directory_to_protect</code> <b>Note:</b> You can include a local drive, a mapped network drive, or a UNC drive in the value. If you do not specify a drive, the sensor monitors all drives for the specified directory.
Description	Type a description that indicates the purpose of this parameter.

4. Click **OK**.
5. Click **OK**.

## Changing the Action for All Monitored Directories

### Introduction

Buffer Overflow Exploit Protection (BOEP) monitors file creation and process creation operations. By default, the sensor aborts (fails) the operation within the process where the buffer overflow was detected and then sends an event to the Console. You can customize the action the sensor takes against a buffer overflow exploit as follows:

- At the policy level, where all sensors that use the policy take the defined action.
- At the sensor level, where only the sensor you configure takes the defined action.

**Important:** The action you specify applies to all monitored directories. If you want the sensor to take a different action for a program file, you must specify an exception for that program file. See “Changing the Action for a Specific Program File” on page 159.

### Prerequisite

You must enable BOEP before the sensor can take any action against a buffer overflow exploit.

### About actions

The following table identifies the components of a BOEP action:

Component	Description
Protection action	Identifies what the sensor does with the actual buffer overflow exploit.
Logging action	Identifies whether the sensor notifies the Console of the buffer overflow exploit.

**Table 35:** Action components

### Available actions

The following table lists the actions the sensor can take when it detects a buffer overflow exploit:

Action	Option	Description
Protection actions	F	Aborts (fails) the file creation or process creation operation within the process where the buffer overflow was detected. <b>Caution:</b> Aborting a process may cause a variety of results that include a system restart or a system shutdown.
	K	Terminates (kills) the entire process where the buffer overflow was detected. <b>Caution:</b> Killing a process may cause a variety of results that include a system restart or a system shutdown.
	I	Ignores this buffer overflow exploit. <b>Note:</b> When you set Ignore as the protection action for all monitored directories, you cannot specify a different protection action for a specific program file.
Logging actions	L	Creates a buffer overflow exploit event.
	N	Does not create a buffer overflow exploit event.

**Table 36:** Actions available for buffer overflow exploits

**Changing the action at the policy level**

To change the action at the policy level:

1. Open the policy you want to customize.
2. On the **OS Events** tab, expand **OS Events**.
3. Double-click **Windows** → **Service Events**.
4. Select **Buffer Overflow Protection**.
5. Click **Tuning**.  
The Sensor Tuning window appears.
6. Is the `vpatch.BaseConfig` tuning parameter already in the list of parameters?
  - If *yes*, select the parameter, and then click **Edit**.
  - If *no*, click **Add**.  
The Advanced Tuning Value window appears.
7. Continue according to the follow table:

Field	Description
Name	Type <code>vpatch.BaseConfig.operation</code> . Where <i>operation</i> is any unique descriptor for the operation. <b>Example:</b> <code>vpatch.BaseConfig.FileOperation</code> <b>Example:</b> <code>vpatch.BaseConfig.ProcOperation</code>
Type	Select <b>String</b> .
Value	To change the process creation action, do one of the following: <ul style="list-style-type: none"> <li>• For Windows 2000 systems, type <code>NtCreateProcess:x:y</code></li> <li>• For Windows 2003 systems, type <code>NtCreateProcess:x:y</code> and <code>NtCreateProcessEx:x:y</code></li> </ul> To change the file creation action, type <code>NtCreateFile:x:y</code> Where x is the protection action and y is the logging action. <b>Important:</b> If you set the global protection action to Ignore, you cannot specify a different action for a specific file. <b>Example:</b> <code>NtCreateProcess:K:L</code> terminates the process that caused a buffer overflow in any of the monitored directories and sends a buffer overflow exploit event to the Console.
Description	Type a description that indicates the purpose of this parameter.

8. Click **OK**.
9. Repeat Steps 6 through 8 to add an action for each operation.
10. Click **OK**.
11. In the Policy Editor window, click **Save**, and then apply the changed policy to the sensor.

**Changing the action at the sensor level**

To change the action at the sensor level:

1. In the Sensor Properties window, select the **Advanced** tab.
2. Is the `vpatch.BaseConfig` tuning parameter already in the list of parameters?
  - If *yes*, select the parameter, and then click **Edit**.

- If *no*, click **Add**.

The Advanced Tuning Value window appears.

3. Continue according to the follow table:

Field	Description
Name	Type <code>vpatch.BaseConfig.operation</code> . Where <i>operation</i> is any unique descriptor for the operation. <b>Example:</b> <code>vpatch.BaseConfig.FileOperation</code> <b>Example:</b> <code>vpatch.BaseConfig.ProcOperation</code>
Type	Select <b>String</b> .
Value	To change the process creation action, do one of the following: <ul style="list-style-type: none"> <li>• For Windows 2000 systems, type <code>NtCreateProcess:x:y</code></li> <li>• For Windows 2003 systems, type <code>NtCreateProcess:x:y</code> and <code>NtCreateProcessEx:x:y</code></li> </ul> To change the file creation action, type <code>NtCreateFile:x:y</code> Where x is the protection action and y is the logging action. <b>Important:</b> If you set the global protection action to Ignore, you cannot specify a different action for a specific file. <b>Example:</b> <code>NtCreateProcess:K:L</code> terminates the process that caused a buffer overflow in any of the monitored directories and sends a buffer overflow exploit event to the Console.
Description	Type a description that indicates the purpose of this parameter.

4. Click **OK**.
5. Repeat Steps 2 through 4 to add an action for each operation you want to change the action for.
6. Click **OK**.
7. Click **OK**.

# Changing the Action for a Specific Program File

**Introduction** By default, an action applies to all monitored directories. If you want the sensor to take a different action for one program file, you can specify an exception for that program file. You can customize the action for a specific program file as follows:

- At the policy level, where all sensors that use the policy take the defined action.
- At the sensor level, where only the sensor you configure takes the defined action.

**Prerequisite** You must enable Buffer Overflow Exploit Protection before the sensor can take a custom action against a buffer overflow exploit.

**Program files you can create an exception for** You can create an exception for any program file that the sensor is monitoring for buffer overflow exploits.

**Creating an exception for a program file that is not monitored** If you want to create an exception for a program file that is not currently monitored by the sensor, you must add the file to the monitor list before you configure the exception. Add the file to the monitor list by monitoring the directory that contains the file.

**Reference:** See “Monitoring Additional Directories” on page 154.

**Changing the action at the policy level** To change the action for a specific program file at the policy level:

1. Open the policy you want to customize.
2. On the **OS Events** tab, expand **OS Events**.
3. Double-click **Windows** → **Service Events**.
4. Select **Buffer Overflow Protection**.
5. Click **Tuning**.  
The Sensor Tuning window appears.
6. Is the `vpatch.Exclude` tuning parameter already in the list of parameters?
  - If *yes*, select the parameter, and then click **Edit**.
  - If *no*, click **Add**.  
The Advanced Tuning Value window appears.
7. Continue according to the follow table:

Field	Description
Name	Type <code>vpatch.Exclude.file</code> . Where <i>file</i> is a unique word for the program file you want to change the action for. <b>Example 1:</b> <code>vpatch.Exclude.MyApplication</code> <b>Example 2:</b> <code>vpatch.Exclude.SuspiciousApplication</code>
Type	Select <b>String</b> .

Field	Description
Value	<p>Type the path to the program file and the actions you want the sensor to take for process creation and file creation operations.</p> <p><b>Important:</b> If you set the global protection action to Ignore, you cannot specify a different protection action for a specific file here.</p> <p><b>Example 1:</b> C:\MyDirectory\MyApplication.exe:NtCreateProcess, I,N:NtCreateFile,I,N</p> <p>Here, the sensor takes no protective action against the process or file operation (I) and does not send an event (N) to the Console for the buffer overflow exploit of MyApplication.exe.</p> <p><b>Example 2:</b> C:\MyDirectory\SuspiciousApplication.exe:NtCreateProcess, K,L:NtCreateFile,K,L</p> <p>Here, the sensor sends an event (L) to the Console for the buffer overflow exploit of SuspiciousApplication.exe and it terminates the entire process that caused the buffer overflow (K).</p> <p><b>Note:</b> This configuration will only work if the global protection action is not set to Ignore.</p> <p><b>Reference:</b> See Table 36 on page 156 for available actions.</p>
Description	Type a description that indicates the purpose of this parameter.

8. Click **OK**.
9. Click **OK**.
10. In the Policy Editor window, click **Save**, and then apply the changed policy to the sensor.

**Changing the action at the sensor level**

To change the action for a specific program file at the sensor level:

1. In the Sensor Properties window, select the **Advanced** tab.
2. Is the `vpatch.Exclude` tuning parameter already in the list of parameters?
  - If *yes*, select the parameter, and then click **Edit**.
  - If *no*, click **Add**.

The Advanced Tuning Value window appears.
3. Continue according to the follow table:

Field	Description
Name	<p>Type <code>vpatch.Exclude.file</code>. Where <i>file</i> is a unique word for the program file you want to change the action for.</p> <p><b>Example 1:</b> vpatch.Exclude.MyApplication</p> <p><b>Example 2:</b> vpatch.Exclude.SuspiciousApplication</p>
Type	Select <b>String</b> .



Field	Description
Value	<p>Type the path to the program file and the actions you want the sensor to take for process creation and file creation operations.</p> <p><b>Important:</b> If you set the global protection action to Ignore, you cannot specify a different protection action for a specific file here.</p> <p><b>Example 1:</b> C:\MyDirectory\MyApplication.exe:NtCreateProcess, I,N:NtCreateFile,I,N</p> <p>Here, the sensor takes no protective action against the process or file operation (I) and does not send an event (N) to the Console for the buffer overflow exploit of MyApplication.exe.</p> <p><b>Example 2:</b> C:\MyDirectory\SuspiciousApplication.exe:NtCreateProcess, K,L:NtCreateFile,K,L</p> <p>Here, the sensor sends an event (L) to the Console for the buffer overflow exploit of SuspiciousApplication.exe and it terminates the entire process that caused the buffer overflow (K).</p> <p><b>Note:</b> This configuration will only work if the global protection action is not set to Ignore.</p> <p><b>Reference:</b> See Table 36 on page 156 for available actions.</p>
Description	Type a description that indicates the purpose of this parameter.

4. Click **OK**.
5. Click **OK**.



## Chapter 11

# Fine-Tuning RealSecure Server Sensor

## Overview

### Introduction

This chapter provides information on tuning parameters that can help you improve the performance of your sensor.

**Note:** Not all parameters are available for all platforms of the sensor. Please note whether the parameter is applicable to your situation before you try to configure it.

### In this chapter

This chapter contains the following topics:

Topics	Page
Excluding an Interface from Monitoring	164
Customizing the Buffer Size for Interfaces	167
Excluding Packets from Analysis	170
Defining the Sensor Pass-through Mode	173

## Excluding an Interface from Monitoring

### Introduction

RealSecure Server Sensor monitors each individual Network Interface Card (NIC) that it detects on your system when the sensor starts. You can disable the monitoring of a NIC if, for example, you have a trusted interface that does not have to be monitored. Exclude an interface from being monitored by defining exclusions as follows:

- At the policy level, where all sensors that use the policy exclude the interface.
- At the sensor level, where only the sensor you configure excludes the interface.

### Sensors that have this tuning parameter

The following sensors support this tuning parameter:

- RealSecure Server Sensor for AIX, Service Release 4.1 or later
- RealSecure Server Sensor for HP-UX, Service Release 4.1
- RealSecure Server Sensor for Solaris, Service Release 4.3

### Interface name used

When you configure interface exclusion, you must use the low level driver name and the card offset to specify the interface you want to exclude from monitoring.

### Determining the name of the interface on an AIX or Solaris system

To determine the name of the interface:

1. Type the following command:  

```
ifconfig -a
```

The system displays information about any interface on the system.
2. Find the interface in the list.

### Determining the name of the interface on an HP-UX system

To determine the name of the interface:

1. Type the following command:  

```
lanscan -v
```

The system displays information about any LAN interface on the system.
2. Find the following lines:

Line	Information
Driver Specific Information	Low-level driver name
Crd In#	Card offset

### Excluding an interface at the sensor level

To exclude a NIC from monitoring:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `pcd.exclusion` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.

3. In the **Name** box, type the parameter name, `pcd.exclusion`.
 

**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **String**.
5. In the **Value** box, do one of the following:
  - on an HP-UX system, type `interface#`, where `interface` is the name of the interface to exclude and `#` is the card offset of the interface
  - on an AIX or Solaris system, type `interface`, where `interface` is the name of the interface to exclude

**Note:** You can exclude several interfaces by separating each entry with a comma.
6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.
 

The tuning parameter with the new setting is listed in the parameters table.
8. Click **OK**.
9. Restart the sensor.

### Excluding an interface at the policy level

To exclude a NIC from monitoring:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the `pcd.exclusion` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
5. In the **Name** box, type the parameter name, `pcd.exclusion`.
 

**Note:** Any typographical errors will render the parameter unusable.
6. In the **Type** box, select **String**.
7. In the **Value** box, do one of the following:
  - on an HP-UX system, type `interface#`, where `interface` is the name of the interface to exclude and `#` is the card offset of the interface
  - on an AIX or Solaris system, type `interface`, where `interface` is the name of the interface to exclude

**Note:** You can exclude several interfaces by separating each entry with a comma.
8. In the **Description** box, type a descriptive comment for this parameter.
9. Click **OK**.
 

The tuning parameter with the new setting is listed in the parameters table.
10. Click **OK**.
11. Save the changed policy, and then apply the policy to the sensor.
12. Restart the sensor.

**Example**

You configure the `pcd.exclusion` parameter as follows:

- Name = `pcd.exclusion`
- Value = `btlan0,igelan1`

The sensor will not monitor the `btlan0` interface or the `igelan1` interface.

# Customizing the Buffer Size for Interfaces

## Introduction

You can customize the amount of buffer space that is allocated to each interface monitored by the sensor. By allocating larger buffers to Gigabit interfaces and smaller buffers to slower interfaces, you use resources more efficiently. Allocate buffer sizes as follows:

- At the policy level, where all sensors that use the policy allocate the buffer space.
- At the sensor level, where only the sensor you configure allocates the buffer space.

## Sensors that have this tuning parameter

The following sensors support this tuning parameter:

- RealSecure Server Sensor for HP-UX, Service Release 4.1
- RealSecure Server Sensor for Solaris, Service Release 4.3

## Interface name used

When you customize the buffer size for an interface, you must use the low level driver name and the card offset of the interface to specify which interface you are customizing.

## Determining the name of the interface on a Solaris system

To determine the name of the interface:

1. Type the following command:

```
ifconfig -a
```

The system displays information about any interface on the system.

2. Find the interface in the list.

## Determining the name of the interface on an HP-UX system

To determine the name of the interface:

1. Type the following command:

```
lanscan -v
```

The system displays information about the LAN interfaces on the system.

2. Find the following lines:

Line	Information
Driver Specific Information	Low-level driver name
Crd In#	Card offset

## Customizing allocated buffer space at the sensor level

To customize the buffer space allocated to interfaces:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `pcd.buffertuning` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.

3. In the **Name** box, type the parameter name, `pcd.buffertuning`.

**Note:** Any typographical errors will render the parameter unusable.

4. In the **Type** box, select **String**.
5. In the **Value** box, type *interface# : XM*, as described in the following table

Element	Description	Example
interface	The name of the interface you are customizing the buffer size for.	btlan
#	The card offset of the interface. <b>Note:</b> You do not need to specify this element on AIX or Solaris systems.	0
X	The size (in megabytes) to allocate to the buffer.	4

**Note:** You can define the buffer size for several interfaces by separating each entry with a comma.

6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.  
The tuning parameter with the new setting is listed in the parameters table.
8. Click **OK**.
9. Restart the sensor.

**Customizing allocated buffer space at the policy level**

To customize the buffer space allocated to interfaces:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the *pcd.buffertuning* parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.  
**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
5. In the **Name** box, type the parameter name, *pcd.buffertuning*.  
**Note:** Any typographical errors will render the parameter unusable.
6. In the **Type** box, select **String**.
7. In the **Value** box, type *interface# : XM*, as described in the following table

Element	Description	Example
interface	The name of the interface you are customizing the buffer size for.	btlan
#	The card offset of the interface. <b>Note:</b> You do not need to specify this element on AIX or Solaris systems.	0
X	The size (in megabytes) to allocate to the buffer.	4

**Note:** You can define the buffer size for several interfaces by separating each entry with a comma.

8. In the **Description** box, type a descriptive comment for this parameter.



9. Click **OK**.

The tuning parameter with the new setting is listed in the parameters table.

10. Click **OK**.

11. Save the changed policy, and then apply the policy to the sensor.

12. Restart the sensor.

### **Example**

You configure the `pcd.buffertuning` parameter as follows:

- Name = `pcd.buffertuning`
- Value = `btlan0:4M,igelan0:8M`

The sensor will allocate 4MB of buffer to the `btlan` interface and 8MB to the `igelan` interface respectively.

## Excluding Packets from Analysis

### Introduction

You can configure the sensor to not analyze packets that come from a specified IP address. This may be useful when you are doing system backups, where traffic does not need to be analyzed. Exclude packets from analysis as follows:

- At the policy level, where all sensors that use the policy filter the traffic.
- At the sensor level, where only the sensor you configure filters the traffic.

### Sensors that have this tuning parameter

The following sensors support this tuning parameter:

- RealSecure Server Sensor for AIX, Service Release 4.1 or later
- RealSecure Server Sensor for HP-UX, Service Release 4.1
- RealSecure Server Sensor for Solaris, Service Release 4.3
- RealSecure Server Sensor for Windows, Service Release 4.4

### Excluding packet analysis at the sensor level

To exclude packets from analysis:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `pcd.packetfilters` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
3. In the **Name** box, type the parameter name, `pcd.packetfilters`.
 

**Note:** Any typographical errors will render the parameter unusable.
4. In the **Type** box, select **String**.
5. In the **Value** box, type `ip_address/CIDR_prefix_length/protocol/direction`, as described in the following table:

Element	Description	Example
<code>ip_address</code>	The IP address of the system to filter. <b>Note:</b> The IP address	172.16.1.2
<code>CIDR_prefix_length</code>	The number of bits used to specify the network prefix of the system to filter.	32
<code>protocol</code>	An integer or string for well-known protocols. <b>Note:</b> Valid values are TCP, UDP, ICMP, or ALL (for all IP protocols).	String = ICMP Integer = 1
<code>direction</code>	The direction of traffic to filter. Valid values are IN for inbound traffic, OUT for outbound traffic. <b>Note:</b> If you do not specify a direction, the sensor filters both inbound and outbound traffic.	IN

**Note:** You can define several filters by separating each entry with a comma.

6. In the **Description** box, type a descriptive comment for this parameter.

7. Click **OK**.

The tuning parameter with the new setting is listed in the parameters table.

8. Click **OK**.

### Excluding packet analysis at the policy level

To exclude packets from analysis:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the `pcd.packetfilters` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.
 

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.
5. In the **Name** box, type the parameter name, `pcd.packetfilters`.
 

**Note:** Any typographical errors will render the parameter unusable.
6. In the **Type** box, select **String**.
7. In the **Value** box, type `ip_address/CIDR_prefix_length/protocol/direction`, as described in the following table:

Element	Description	Example
ip_address	The IP address of the system to filter. <b>Note:</b> The IP address	172.16.1.2
CIDR_prefix_length	The number of bits used to specify the network prefix of the system to filter.	32
protocol	An integer or string for well-known protocols. <b>Note:</b> Valid values are TCP, UDP, ICMP, or ALL (for all IP protocols).	String = ICMP Integer = 1
direction	The direction of traffic to filter. Valid values are IN for inbound traffic, OUT for outbound traffic. <b>Note:</b> If you do not specify a direction, the sensor filters both inbound and outbound traffic.	IN

**Note:** You can define several filters by separating each entry with a comma.

8. In the **Description** box, type a descriptive comment for this parameter.

9. Click **OK**.

The tuning parameter with the new setting is listed in the parameters table.

10. Click **OK**.

11. Save the changed policy, and then apply the policy to the sensor.

**Examples**

The following table illustrates how you might configure the `pcd.packetfilters` parameter:

To filter...	Use the value...	To...
a single IP address	172.16.1.2/32/1/OUT	prevent analysis of outbound ICMP traffic to 172.16.1.2.
two IP addresses	172.16.1.2/32/6,172.16.1.0/24/6	prevent analysis of outbound and inbound TCP traffic to and from 172.16.1.2 and the 172.16.1.0 subnet.
a range of IP addresses	172.16.1.0/24/17/IN	prevent analysis of inbound UDP traffic from all ip addresses in the range 172.16.1.0-172.16.1.255.

**Table 37:** *Examples of how to use the `pcd.packetfilters` parameter*

## Defining the Sensor Pass-through Mode

### Introduction

RealSecure Server Sensor monitors all traffic to and from the server. During overload conditions, traffic may be delayed as the sensor tries to process the high load. You can configure the sensor to allow traffic to pass-through the sensor to prevent any possible delays. Define the sensor pass-through mode as follows:

- At the policy level, where all sensors that use the policy filter the traffic.
- At the sensor level, where only the sensor you configure filters the traffic.

**Important:** Allowing traffic to bypass the protection offered by the sensor may impact the integrity of your server.

### Sensors that have this tuning parameter

This tuning parameter is currently supported in RealSecure Server Sensor for AIX, Service Release 4.2.

### Defining the pass-through mode at the sensor level

To define the pass-through mode:

1. In the Sensor Properties window, select the **Advanced Parameters** tab.
2. Do you see the `pcd.failmode` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 4.
  - If *no*, click **Add**.

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.

3. In the **Name** box, type the parameter name, `pcd.failmode`.

**Note:** Any typographical errors will render the parameter unusable.

4. In the **Type** box, select **String**.
5. In the **Value** box, specify one of the following:

Value	Description
failopen	Instructs the sensor to allow traffic to pass-through; the sensor provides no protection against malicious packets in traffic that passes through. <b>Note:</b> This is the default setting.
failclosed	Instructs the sensor to process all traffic; the sensor provides protection against malicious packets, but some packets may be dropped if the packet queue fills up. <b>Note:</b> The sensor stops dropping packets when the packet queue has room to queue incoming packets.

6. In the **Description** box, type a descriptive comment for this parameter.
7. Click **OK**.

The tuning parameter with the new setting is listed in the parameters table.

8. Click **OK**.

**Defining the pass-through mode at the policy level**

To define the pass-through mode:

1. Open the policy you want to customize.
2. On the **Network Events** tab, select any group of signatures.
3. Click **Tuning**.
4. Do you see the `pcd.failmode` parameter in the table?
  - If *yes*, select the parameter, click **Edit**, and then go to Step 6.
  - If *no*, click **Add**.

**Important:** If you enter conflicting or duplicate tuning parameters, the parameter entered last overrides the parameter entered first.

5. In the **Name** box, type the parameter name, `pcd.failmode`.
 

**Note:** Any typographical errors will render the parameter unusable.
6. In the **Type** box, select **String**.
7. In the **Value** box, specify one of the following:

Value	Description
failopen	Instructs the sensor to allow traffic to pass-through; the sensor provides no protection against malicious packets in traffic that passes through. <b>Note:</b> This is the default setting.
failclosed	Instructs the sensor to process all traffic; the sensor provides protection against malicious packets, but some packets may be dropped if the packet queue fills up. <b>Note:</b> The sensor stops dropping packets when the packet queue has room to queue incoming packets.

8. In the **Description** box, type a descriptive comment for this parameter.
9. Click **OK**.
 

The tuning parameter with the new setting is listed in the parameters table.
10. Click **OK**.
11. Save the changed policy, and then apply the policy to the sensor.

## Troubleshooting





## Chapter 12

# Troubleshooting

## Overview

**Introduction** This chapter describes issues you may encounter while using RealSecure Server Sensor; it also describes how to troubleshoot them.

**Contacting Technical Support** If you encounter a problem that is not described in this chapter, see “Getting Technical Support” on page 10 for information about contacting Technical Support.

**In this chapter** This chapter contains the following topics:

Topic	Page
Isolating Policy Problems	178
Tcl Script Problems	185
No Communication Between the Sensor and the Console	186
No Communication Between the Sensor on an ISA Server and the Console	187
Failure to Open a Control Channel Error	188
Not Seeing Any BOEP Events	189
BOEP Action for a Specific Program File Not Working	190
Api Read Queue Messages in the Syslog	191

## Isolating Policy Problems

<b>Introduction</b>	This topic provides information about isolating and correcting possible policy problems.
<b>Error messages that indicate a policy problem</b>	<p>Error messages that might indicate policy file problems include the following:</p> <ul style="list-style-type: none"><li>● “No such file or directory” displays in the Control Status column when connecting to a sensor</li><li>● “Failure to transfer current policy when the control channel opened” displays in the Control Status column</li><li>● “Failure to read/transfer common.policy” displays in the Control Status column</li><li>● “Unknown” displays in the Policy column</li></ul>
<b>Troubleshooting techniques</b>	<p>When an error message indicates a possible policy file problem, you can try the following to isolate and correct the problem:</p> <ul style="list-style-type: none"><li>● Inspect the <code>issDaemon</code> directory and sensor component directory. Look for things such as file sizes of zero bytes, or no policy files listed.</li><li>● Check the event log (Windows) or the syslog (UNIX) of the sensor for any unusual events.</li><li>● Reapply a policy to the sensor.</li><li>● Revert the sensor to a default policy configuration by deleting the <code>current.policy</code> file, and then apply a new policy to the sensor. <b>Reference:</b> For more information, see “Deleting <code>current.policy</code>” on page 182.</li><li>● Determine whether the problem is the daemon. Start the daemon without automatically restarting the sensor to determine if the daemon loads correctly. <b>Reference:</b> For more information, see “Determining whether the problem may be the daemon” on page 182.</li><li>● Start the sensor manually to detect sensor component problems. <b>Reference:</b> For more information, see “Starting the sensor manually to detect sensor component problems” on page 183.</li><li>● If the problem persists, contact IBM ISS Technical Support. <b>Reference:</b> For technical support contact information, see “Getting Technical Support” on page 10.</li></ul> <p><b>Important:</b> To maintain the sensor’s intended configuration, change any newly applied policy to match the previous configuration.</p>

## Stopping the issDaemon

Some troubleshooting techniques for policy problems require you to stop the issDaemon. You can do this from the desktop (Windows only) or from the command line.

To stop the issDaemon, use the following decision table to determine your action:

For this operating system...	from the...	do this...
Windows NT	Desktop	To stop the issDaemon from the desktop: <ol style="list-style-type: none"> <li>1. Select <b>Start</b>→<b>Settings</b>→<b>Control Panel</b>.</li> <li>2. Double-click <b>Services</b>.</li> <li>3. Double-click <b>issDaemon</b>. The issDaemon Properties window appears.</li> <li>4. In the Service Status section, click <b>Stop</b>.</li> <li>5. Click <b>OK</b>. The issDaemon stops.</li> </ol>
Windows NT	Command line	To stop the issDaemon from the command line: <ul style="list-style-type: none"> <li>• Type <code>C:\&gt;net stop issdaemon</code> The issDaemon stops.</li> </ul>
Windows	Desktop	To stop the issDaemon from the desktop: <ol style="list-style-type: none"> <li>1. Select <b>Start</b>→<b>Settings</b>→<b>Control Panel</b>.</li> <li>2. Do one of the following:               <ul style="list-style-type: none"> <li>■ Double-click <b>Services</b>.</li> <li>■ Double-click <b>Administrative Tools</b>, and then double-click <b>Services</b>.</li> </ul> </li> <li>3. Double-click <b>issDaemon</b>. The issDaemon Properties window appears.</li> <li>4. In the Service Status section, click <b>Stop</b>.</li> <li>5. Click <b>OK</b>. The issDaemon stops.</li> </ol>
Windows	Command line	To stop the issDaemon from the command line: <ul style="list-style-type: none"> <li>• Type <code>C:\&gt;net stop issdaemon</code> The issDaemon stops.</li> </ul> <p><b>Note:</b> This procedure assumes that the product is installed in the default directory and is the first or only server sensor installed on the computer.</p>
Solaris	Command line	<ol style="list-style-type: none"> <li>1. Log in as <code>root</code>.</li> <li>2. At the command line, type: <code>/etc/init.d/realsecure stop</code> The issDaemon stops.</li> </ol>
HP-UX	Command line	<ol style="list-style-type: none"> <li>1. Log in as <code>root</code>.</li> <li>2. At the command line, type the following command: <code>/sbin/init.d/realsecure stop</code> The issDaemon stops.</li> </ol>

<b>For this operating system...</b>	<b>from the...</b>	<b>do this...</b>
AIX	Command line	<ol style="list-style-type: none"><li>1. Log in as <code>root</code>.</li><li>2. At the command line, type the following command: <code>/user/bin/realsecure stop</code> The <code>issDaemon</code> stops.</li></ol>

## Starting the issDaemon

Some troubleshooting techniques for policy problems require you to start the issDaemon. You can do this from the desktop (Windows only) or from the command line. To start the issDaemon, use the following decision table to determine your action:

For this operating system...	from the...	Do this...
Windows NT	Desktop	To start the issDaemon from the desktop: <ol style="list-style-type: none"> <li>1. Select <b>Start</b>→<b>Settings</b>→<b>Control Panel</b>.</li> <li>2. Double-click <b>Services</b>.</li> <li>3. Double-click <b>issDaemon</b>. The issDaemon Properties window appears.</li> <li>4. In the Service Status section, click <b>Start</b>.</li> <li>5. Click <b>OK</b>. The issDaemon starts.</li> </ol>
Windows NT	Command line	To start the issDaemon from the command line: <ul style="list-style-type: none"> <li>• Type <code>C:\&gt;net start issdaemon</code> The issDaemon starts.</li> </ul>
Windows	Desktop	To start the issDaemon from the desktop: <ol style="list-style-type: none"> <li>1. Select <b>Start</b>→<b>Settings</b>→<b>Control Panel</b>.</li> <li>2. Do one of the following:               <ul style="list-style-type: none"> <li>■ Double-click <b>Services</b>.</li> <li>■ Double-click <b>Administrative Tools</b>, and then double-click <b>Services</b>.</li> </ul> </li> <li>3. Double-click <b>issDaemon</b>. The issDaemon Properties window appears.</li> <li>4. In the Service Status section, click <b>Start</b>.</li> <li>5. Click <b>OK</b>. The issDaemon starts.</li> </ol>
Windows	Command line	To start the issDaemon from the command line: <ul style="list-style-type: none"> <li>• Type <code>C:\&gt;net start issdaemon</code> The issDaemon starts.</li> </ul>
Solaris	Command line	<ol style="list-style-type: none"> <li>1. Log in as <code>root</code>.</li> <li>2. At the command line, type: <code>/etc/init.d/realsecure start</code> The issDaemon starts.</li> </ol>
HP-UX	Command line	<ol style="list-style-type: none"> <li>1. Log in as <code>root</code>.</li> <li>2. At the command line, type the following command: <code>/sbin/init.d/realsecure start</code> The issDaemon starts.</li> </ol>
AIX	Command line	<ol style="list-style-type: none"> <li>1. Log in as <code>root</code>.</li> <li>2. At the command line, type the following command: <code>/user/bin/realsecure start</code> The issDaemon starts.</li> </ol>

### Deleting `current.policy`

You can revert the sensor to a default policy configuration by deleting the `current.policy` file and then applying a new policy to the sensor. To delete `current.policy`, you must first stop the `issDaemon`, then delete `current.policy`, and then restart the `issDaemon`. The sensor should restart automatically when you restart the daemon. When the sensor restarts it uses `default.policy` until it receives a new policy file from the Console.

**Important:** Do not delete `current.policy` while the sensor is running.

To delete `current.policy`:

1. Stop the `issDaemon`.

**Reference:** See “Stopping the `issDaemon`” on page 179.

2. In the directory where the sensor was installed, locate and then delete `current.policy`.

3. Restart the `issDaemon`.

**Reference:** See “Starting the `issDaemon`” on page 181.

4. If the sensor did not restart see “Determining whether the problem may be the daemon” on page 182.

5. Using Windows Explorer, locate the transfer directory at the following location:

```
C:\Program Files\ISS\SiteProtector\Application Server\temp\Sensor  
Controller
```

6. Delete the files that contain the sensor’s IP address in the filename.

7. Apply a default, customized, or imported policy to the sensor.

### Determining whether the problem may be the daemon

By default, the sensor starts automatically when you start the daemon. You can prevent the sensor from starting automatically by stopping the daemon, editing the `issDaemon.policy` file, and then restarting the daemon. If the daemon fails to load, then the problem may be caused by the daemon or one of its policy files. IBM ISS recommends that you troubleshoot daemon problems with the help of Technical Support.

To determine if the problem was caused by the daemon:

1. Stop the `issDaemon`.

**Reference:** See “Stopping the `issDaemon`” on page 179.

2. Locate the `issDaemon.policy` file.

- The Windows default location for this file is as follows:

```
C:\Program Files\ISS\IssDaemon
```

- The UNIX default location for this file is as follows:

```
/opt/ISS/issDaemon
```

3. Open the `issDaemon.policy` file in a text editor such as Notepad or vi.

4. Change the `auto_recovery` value for the selected sensor from 1 to 0.

**Note:** Setting the `auto_recovery` value to 0 causes the daemon to start without automatically starting the sensor.

5. Save and close the `issDaemon.policy` file, and then close the text editor.

6. Restart the issDaemon.

**Reference:** See “Starting the issDaemon” on page 181.

7. Check the running processes to determine whether the daemon was loaded correctly.

Use the following decision table to check the running processes:

For this operating system...	Do this...
Windows	<ol style="list-style-type: none"> <li>1. Press CTRL+ALT+DELETE.</li> <li>2. Click <b>Task Manager</b>.</li> <li>3. Select the Processes tab, and ensure that the issDaemon.exe process is running.</li> </ol>
Solaris, HP-UX, or AIX	At the command line, type: <pre>han[admin]# ps -ef  grep iss</pre>

8. Did the issDaemon load correctly?

- If *yes*, then you must start the sensor manually and determine if there are any sensor component problems. See “Starting the sensor manually to detect sensor component problems” on page 183.
- If *no*, then the policy problem may have been caused by the daemon. Contact IBM ISS Technical Support for further assistance.

**Reference:** See “Getting Technical Support” on page 10.

9. Stop the issDaemon.

**Reference:** See “Stopping the issDaemon” on page 179.

10. Locate the issDaemon.policy file as follows:

- on Windows systems, the default location for this file is C:\Program Files\ISS\IssDaemon
- on UNIX systems, the default location for this file is opt/ISS/issDaemon

11. Open the issDaemon.policy file in a text editor such as Notepad or vi.

12. Change the auto\_recovery value for the selected sensor to 1.

13. Save and close the issDaemon.policy file, and then close the text editor.

14. Restart the issDaemon.

**Reference:** See “Starting the issDaemon” on page 181.

15. Did the sensor start automatically after you started the daemon?

- If *yes*, determine whether the policy problem has been corrected. If the problem persists, see “If you need further help” on page 184.
- If *no*, then you must start the sensor manually to determine if the policy problem has been corrected. See “Starting the sensor manually to detect sensor component problems” on page 183.

### Starting the sensor manually to detect sensor component problems

The sensor may not start automatically when the daemon is started. This could happen if the sensor was in a stopped state before the daemon was stopped. You can start the sensor manually from the desktop (Windows only) or from the command line.

**Important:** Be sure that the issDaemon is stopped before you start the sensor manually. If you try to start the sensor manually without first shutting down the daemon, the following error message appears:

### Error creating FirstInstance Mutex

To start the sensor manually and determine whether sensor component problems may exist:

1. Start the sensor with CSF Trace set at level 6.
2. Check whether the sensor started correctly by looking for an active sensor status in SiteProtector.
  - If the sensor is active, then set the auto-recovery flag back to its original value. Go to Step 3.
  - If the sensor is not active, then contact IBM ISS Technical Support for further assistance. See “Getting Technical Support” on page 10 for contact information.

3. Stop the issDaemon.

**Reference:** See “Stopping the issDaemon” on page 179.

4. Locate the issDaemon.policy file as follows:

- on Windows systems, the default location for this file is C:\Program Files\ISS\IssDaemon
- on UNIX systems, the default location for this file is /opt/ISS/issDaemon

5. Open the issDaemon.policy file in a text editor such as Notepad or vi.

6. Set the auto\_recovery value for the selected sensor to 1.

7. Save and close the issDaemon.policy file, and then close the text editor.

8. Restart the issDaemon.

The sensor starts automatically.

**Reference:** See “Starting the issDaemon” on page 181.

### If you need further help

If you still cannot isolate the problem, send copies of the sensor’s current.policy file, common.policy file, and syslog to IBM ISS Technical Support for further assistance. See “Getting Technical Support” on page 10 for contact information. Include the output from the following commands:

```
netstat -a  
ps -ef
```



---

## Tcl Script Problems

**Introduction** This topic can help you troubleshoot problems with standard Tcl procedures.

**Background** When Tcl errors occur, the Tcl process creates a `TCL_ERROR` exception. This exception is reported to the Console and appears as a medium priority event. Under normal circumstances, the Tcl procedure that generated the error does not continue to run. However, you can use a standard Tcl procedure called `catch` to keep the process from failing.

**The catch procedure** The `catch` procedure looks for error codes that happen within another procedure. If an error occurs, you can have the script perform a different set of steps to complete the procedure.

**Example:** If you create a procedure called `counter` that increments the value of a certain global variable by one each time the procedure is called, you can use the `catch` procedure to make sure this variable exists. Under normal circumstances (without `catch`), the procedure would fail if the variable did not exist before the procedure started.

The following code demonstrates using `catch` with a procedure called `counter`:

```
proc counter {} {
    if {[catch {Retrieve mycount} n]} {
        set n 1
        Store mycount $n
    } else {
        incr n
        Store mycount $n
    }
    return $n
}
```

**Other troubleshooting options**—For troubleshooting purposes, you can turn off Fusion Scripting for one signature or for all signatures.

**Reference:** See “Disabling Fusion Scripting” on page 140 for more information.

## No Communication Between the Sensor and the Console

<b>Problem</b>	You have applied a policy to the sensor and now the sensor and the Console cannot communicate.
<b>Background</b>	<p>Firecell signatures work like a firewall to ensure that only authorized clients can access the server.</p> <p>Because of this, you can easily disable all traffic to and from a system, including communication between the Console and the sensor.</p>
<b>Solution</b>	<p>To reestablish communication to the sensor:</p> <ol style="list-style-type: none"><li>1. Manually stop the issDaemon service. <b>Reference:</b> See “Stopping the issDaemon” on page 179.</li><li>2. Manually delete <code>current.policy</code> on the affected sensor. <b>Reference:</b> See “Deleting <code>current.policy</code>” on page 182.</li><li>3. Manually restart the issDaemon. <b>Reference:</b> See “Starting the issDaemon” on page 181.</li></ol>

## No Communication Between the Sensor on an ISA Server and the Console

<b>Problem</b>	You have configured communication between RealSecure Server Sensor for Windows and SiteProtector, but there is no communication between these components.
<b>Background</b>	By default, the ISA Server blocks communication between the sensor and the SiteProtector components.
<b>Solution</b>	Configure the ISA server to allow incoming connections to port 2998 and port 902.

## Failure to Open a Control Channel Error

**Problem** You see the following error message when you try to apply SiteProtector command-jobs to the sensor:

```
Result Failed to open control channel to the sensor, cannot perform any actions on it.
```

**Background** While this message can be displayed for a variety of reasons, including improper exchange of authentication keys, it is possible that the network interface card (NIC) settings on the sensor system may also cause communications problems.

**Solution** Check the duplex settings for the NIC that communicates with the SiteProtector system and make sure they are statically set (for example, set to 100 Full), instead of set to auto. Restart the sensor and try to apply commands again.

## Not Seeing Any BOEP Events

<b>Problem</b>	You have enabled Buffer Overflow Exploit Protection but you are not seeing any BOEP events.
<b>Background</b>	By default, Data Execution Prevention (DEP) is enabled on Windows 2003 Server. DEP may block certain buffer overflow exploits before the BOEP module of the sensor can analyze them and send an alert.
<b>Solution</b>	Disable DEP if you want the sensor to monitor for BOEP events.

## BOEP Action for a Specific Program File Not Working

**Problem** You enabled Buffer Overflow Exploit Protection (BOEP) and set the global protection and logging actions. You then configured the sensor to take different actions if it detected an attempt to exploit a buffer overflow in a specific program file. There was an attempt to exploit a buffer overflow in the specific program file, but the sensor did not take the actions you specified.

**Background** By default, a BOEP action applies to all monitored directories. If you want the sensor to take a different action for a specific program file, you must specify an exception for that program file. If, however, you specify the global protection action as Ignore, you cannot specify a different protection action for the specific program file. You can specify a different logging action regardless of the global logging action.

**Solution** Change the global protection action from Ignore to either Fail (abort) or Kill (terminate).

**Reference:** “Changing the Action for All Monitored Directories” on page 156.

---

## Api Read Queue Messages in the Syslog

<b>Problem</b>	<p>When you run RealSecure Server Sensor 7.0, Service Release 4.1 for HP-UX with the Network Monitoring Component enabled, you may periodically see the following messages in the syslog or 'dmesg' output:</p> <pre>queue_packet: api read queue XXX is full: XX:XXXX can't put packet onto api read queue XXX</pre>
<b>Background</b>	<p>These messages can occur in high network traffic situations and they indicate that the sensor's packet capture driver (pcd) has determined that the message queue for sending packets to the sensor is full.</p> <p>When this occurs, the sensor will implement flow control and begin to send packets directly upstream, bypassing the sensor until enough free space is available on the api read queue for sending packets to the sensor.</p>
<b>Solution</b>	<p>This is a rare event; however, you can increase the queue size on interfaces that handle high amounts of traffic. The default message queue size is 2MB of kernel memory for each interface. To increase the queue size, you can use the pcd.buffertuning advanced tuning parameter.</p> <p><b>Reference:</b> See "Customizing the Buffer Size for Interfaces" on page 167.</p>





# Appendixes



## Appendix A

# Configuring the Web Server Monitoring Component

## Overview

### Introduction

A full upgrade to a sensor configures the Web server monitoring component if the Web server is installed in the default location. If the Web server is not installed in the default location, you must manually configure it after the upgrade is installed. This appendix describes how to manually configure your Web server monitoring component.

### In this appendix

This appendix contains the following topics:

Topics	Page
Configuring an Apache Web Server Monitoring Component Manually	196
Configuring an IIS Web Server Monitoring Component Manually	197

# Configuring an Apache Web Server Monitoring Component Manually

## Introduction

If you apply a full upgrade to a sensor installed on a system where the Web server monitoring component is not installed in the default location, you must manually configure the component after the upgrade. This topic describes how to manually configure your Apache Web server monitoring component.

## Configuring the Apache Web server

To manually configure the Apache Web server monitoring component:

1. Confirm the Apache Web server supports Dynamic Shared Object (DSO) as follows:
  - If you are using Apache, type `httpd -l`.
  - If you are using Apache+modssl, type `httpd -l`.
  - If you are using Apache+OpenSSL, type `httpsd -l`.

If the Web server supports DSO, the result contains `mod_so`.

**Note:** If the Apache Web server does not support DSO, go to <http://www.apache.org> to obtain the Apache source, and then compile the source with `mod_so` enabled.

2. Add the following line to the `httpd.conf` file:

```
LoadModule rs_module /opt/ISS/lib/mod_rs.so
```

3. Restart the Apache Web server.

**Note:** If the Web server is Apache+modssl, you may get a message each time you restart the Web server notifying you that the module is not compiled with the EAPI flag and that the Web server may crash. This message is the result of a RedHat bug and the Web server *will not* crash. You can use an EAPI flag compiled Apache module provided by IBM Internet Security Systems to prevent this message from appearing. See “Installing an EAPI flag compiled Apache module” on page 196 for more information.

## Installing an EAPI flag compiled Apache module

To install the EAPI flag-compiled Apache module:

1. Locate the following file on your system after the full upgrade:

```
/opt/ISS/lib/apache/mod_rs.so.ssl
```

2. Copy `mod_rs.so.ssl` to the following file:

```
/opt/ISS/lib/mod_rs.so
```

3. Restart the Apache Web server.

# Configuring an IIS Web Server Monitoring Component Manually

## Introduction

If you apply a full upgrade to a sensor installed on a system where the Web server monitoring component is not installed in the default location, you must manually configure the component after the upgrade. This topic describes how to manually configure your IIS Web server monitoring component.

## Procedure

To configure the IIS Web server monitoring component:

1. Click **Start** → **Programs** → **Administrative Tools** → **Internet Services Manager**.  
The Internet Information Services window appears.
2. Right-click on the IIS Web server, and then select **Properties**.  
The Web Server Properties window appears.
3. In the Master Properties section, select **WWW Service**, and then click **Edit**.  
The WWW Service Master Properties for Web Server window appears.
4. Select the **ISAPI Filter** tab, and then click **Add**.
5. Continue based on the following table:

For this server sensor...	Do this...
earlier than Service Release 4.4	<ol style="list-style-type: none"> <li>1. In the <b>Filter Name</b> box, type <b>rsiisfilter</b>.</li> <li>2. In the <b>Executable</b> box, type the full path to the <b>rsiisfilter.dll</b>. <b>Note:</b> If you installed server sensor in the default location, the full path should be: C:\Program Files\ISS\issSensors\server_sensor_1\ISAPI\rsiisfilter.dll</li> </ol>
Service Release 4.4	<ol style="list-style-type: none"> <li>1. In the <b>Filter Name</b> box, type <b>RSISapiPlugin</b>.</li> <li>2. In the <b>Executable</b> box, type the full path to the <b>RSSIISapiPlugin.dll</b>. <b>Note:</b> If you installed server sensor in the default location, the full path should be: C:\Program Files\ISS\issSensors\server_sensor_1\ISAPI\RSISapiPlugin.dll</li> </ol>

6. Click **OK**.  
The system applies your changes.
7. Restart IIS.



# Index

## symbols

.enc files 144, 147  
@FieldN 69  
@StringN 68  
\_\_iss\_attacktime 121  
\_\_iss\_dstip 121  
\_\_iss\_dstport 121  
\_\_iss\_priority 121  
\_\_iss\_rulename 121  
\_\_iss\_srcip 121  
\_\_iss\_srcport 121  
{!} 69

## a

about RealSecure Server Sensor 14  
actions  
    Buffer Overflow Exploit Protection 152, 156  
advanced tuning parameter  
    defining location of syslog file 60  
    for buffer overflow protection 152  
    for disabling sensor BSM management 110  
    for excluding NIC 164  
    for excluding packets from analysis 170  
    for setting buffer space 167  
    for setting pass-through mode 173–174  
    for user and group name resolution 107  
AIX start-up procedure 183  
all files, monitoring 81  
AllowAllAcknowledgementPackets  
    tuning parameter 37  
allowing  
    only Internet traffic 40  
    only local traffic 40  
api read queue 191

## b

binary log monitoring 87  
binary log signatures  
    user-defined 87  
blocking  
    at the application layer 14

    before reaching the IP stack 14  
    external network traffic 40  
BSM log management  
    disabling 110  
BSM See C2 audit  
Buffer Overflow Exploit Protection  
    advanced tuning parameter 152  
    and Security Fusion Module 152  
    available actions 152, 156  
    default action 152  
    default protection 153  
    enabling 153  
    enabling default 153  
    global protection action 156  
    limits on custom actions 156, 160–161, 190  
    monitored files 152  
    monitoring additional directories 154  
    vpatch.BaseConfig parameter 157–158  
    vpatch.Exclude customize action 159–160  
    vpatch.Exclude parameter 159–160  
    vpatch.Include parameter 154–155  
buffer space  
    allocating to NIC 167  
buffer space allocation  
    determining interface name AIX 167  
    determining interface name HP-UX 167  
    determining interface name Solaris 167  
    policy-level 168  
    sensor-level 167

## C

C2 audit  
    creating user-defined signatures 96  
    file size 109  
    log management 109  
    log reduction for AIX 109  
    log reduction for HP-UX 109  
    log reduction for Solaris 109  
changing event priority 136  
configuring  
    Apache Web server 196  
    fusion scripting responses 129  
    fusion scripting SNMPv3 response 130

IIS Web server 197  
 user-defined network signature 53  
 correlating information between signatures 136  
 customizing buffer size  
 example 169

## d

default action  
 Buffer Overflow Exploit Protection 152  
 default protection  
 Buffer Overflow Exploit Protection 153  
 Default.policy 23  
 definition  
 events 16  
 policies 16  
 policy files 16  
 signatures 16  
 disabling  
 enforce audit policy 59  
 firecell signatures 39  
 fusion scripting 140  
 wttmpx log monitoring 88

## e

enabling  
 Buffer Overflow Exploit Protection 153  
 default Buffer Overflow Exploit Protection 153  
 enforce audit policy 58  
 fusion scripting SNMPv3 response 133  
 user-defined network signature 53  
 wttmpx log monitoring 88  
 enforce audit policy  
 disabling 59  
 enabling 58  
 error message  
 failure to open a control channel 188  
 event data values 121  
 event detection  
 at the application layer 14  
 before reaching the IP stack 14  
 event priority  
 changing 136  
 events 16  
 monitoring mail.log (HP-UX) 61  
 monitoring syslog 60, 85  
 priority 136  
 priority of 121  
 processing with fusion scripting 118

processing without fusion scripting 118  
 uniquely identifying 127  
 evidence logging  
 enabling (Windows only) 148  
 policy-level 148  
 sensor-level 148  
 evidence logging (Windows only) 147  
 examples  
 customizing buffer size 169  
 excluding an interface 166  
 excluding packets 172  
 firecell signatures 40  
 resolving user and group names 108  
 excluding an interface  
 examples 166  
 excluding packets  
 examples 172  
 extensions  
 Tcl 126

## f

failclosed 173–174  
 failopen 173–174  
 failure to open a control channel 188  
 file rotation 81  
 file switching 81  
 firecell signatures 31, 43  
 about 32  
 allowing local subnet traffic 40  
 allowing only Internet traffic 40  
 allowing typical internet traffic, example 40  
 blocking traffic with 40  
 configuring 35  
 considerations to using 35  
 creating 35  
 disabling 39  
 example 38  
 examples 40  
 monitoring  
 external network traffic 40  
 ip traffic 40  
 order of 39  
 rearranging order of 34  
 relevance of order of 34  
 responses available to 32  
 specifying ports 33  
 using 40  
 when to use 32  
 fusion scripting  
 adding to signatures 128



- configuring responses 129
- correlating information between signatures 136
- disabling 140
- event data values and 121
- information fields and 120
- initialization scripts 122
- prerequisites 119
- priority level and 121
- procedure scripts 123
- process 122
- response scripts 123
- responses 129
- true and false values 134
- validation script values 134
- validation scripts 123
- when to use 118
- fusion scripting response
  - configuring 129
  - configuring SNMPv3 130
  - enabling SNMPv3 133
- fusion scripts
  - accessing information fields 135
  - adding 128
  - changing event priority 136
  - creating new information fields 135
  - modifying 128
  - monitoring trusted users 138
  - passing information to responses 136
  - saving information fields 135
  - saving variables to log files 137
  - using initialization scripts 135

## g

- GetData 126
- GetTid 127
- global variables 120

## h

- HP-UX start-up procedure 183

## i

- IBM Internet Security Systems
  - technical support 10
  - Web site 10
- IBM ISS Technical Support 10
- importing policies 21
- information fields 120

- initialization scripts 122
- interface exclusion
  - determining interface name AIX 164
  - determining interface name HP-UX 164
  - determining interface name Solaris 164
  - policy-level 165
  - sensor-level 164
- intrusion prevention
  - block response 15
  - firecell signatures 15
- IP address
  - blocking options 38

## I

- log file switching
  - syslog warning messages 63
- log files
  - monitoring 67
- logging
  - evidence (Windows only) 147
  - packet 144
- Lookup Names
  - tuning parameter 107

## m

- mail messages
  - monitoring (HP-UX) 61
- mail.log events
  - monitoring (HP-UX) 61
- monitored files
  - Buffer Overflow Exploit Protection 152
- monitoring
  - binary logs 87
  - external network traffic 40
  - for a text string 83
  - ip traffic using firecell signatures 40
  - log file 67, 82
  - mail messages (HP-UX) 61
  - mail.log events (HP-UX) 61
  - newest or all files 81
  - syslog events 60, 85
  - trusted users 138
  - wtmpx binary log file 87
- monitoring additional directories
  - Buffer Overflow Exploit Protection 154

**n**

Network Events tab 27  
 Network Information Service (NIS) 107  
 network signatures  
   user-defined 49  
 newest files, monitoring 81  
 Newest Only flag 81  
 NIC  
   allocating buffer space to 167  
   disable monitoring of 164

**o**

order of firecell signatures 39  
 ordering of firecell signatures 34  
 OS Events tab 27

**p**

packet analysis exclusion  
   policy-level 171  
   sensor-level 170  
 packet logging 144  
   policy-level 145  
   sensor-level 145  
 packets  
   exclude from analysis 170–171  
 PAM Help file 50  
 pass-through mode  
   policy-level 174  
   sensor-level 173  
 pcd.buffertuning  
   tuning parameter 167–168  
 pcd.exclusion  
   tuning parameter 164–165  
 pcd.failmode  
   default setting 173–174  
   tuning parameter 173–174  
 pcd.packetfilters  
   examples 172  
   tuning parameter 170–171  
 persistent memory 120  
 policies 16  
   importing to 6.5 21  
   importing to 7.0 21  
   isolating problems 178  
   predefined 18  
   troubleshooting 178  
   user-defined 18

  policies 7.0  
     Attacks\_and\_Audits\_ 20  
     Blank\_ 19  
     Network\_Attacks\_ 19  
 policy  
   editing to add user-defined network signature 52  
   files 16  
 port number  
   blocking options 38  
 ports  
   25 and 80 40  
 Practical Programming in Tcl and Tk 119  
 predefined policies 18  
 predefined policies 7.0  
   events monitored by 19  
   events not monitored by 19  
 predefined signatures 26  
 prerequisite  
   fusion scripting 119  
   monitoring log files 81  
   monitoring UNIX syslogs 85  
 procedure scripts 123  
 Protect tab 27

**r**

RealSecure Server Sensor  
   about 14  
 rearranging the order of firecell signatures 39  
 regular expression  
   monitoring for a string 83  
 regular expressions 68  
 Remove 126  
 RemoveArray 127  
 resolving user and group names  
   examples 108  
 response scripts 123  
 RestoreArray 127  
 Retrieve 126  
 returning  
   true and false results in fusion scripts 134

**s**

SaveArray 127  
 Security Fusion Module  
   and Buffer Overflow Exploit Protection 152  
 sensor.noc2logsizeimit  
   tuning parameter 110  
 sensor.syslogfile  
   tuning parameter 60

SetData 126

signatures 16

- about 26
- firecell 31, 43
- predefined 26
- user-defined 26
- user-defined BSM 93
- user-defined C2 audit 93

SNMPv3 fusion scripting response 130

Solaris start-up procedure 183

specifying generic logs with wildcards 81

Store 126

syslog events

- monitoring 60, 85

## t

tabs in the policy editor 27

Tcl

- extensions 126
- references for 119
- tclproc1.log 137
- troubleshooting 185
- tutorial 119

technical support, IBM Internet Security Systems 10

transient

- memory 120
- variables 120

troubleshooting

- api read queue 191
- BOEP action not working 190
- failure to open a control channel 188
- no sensor and Console communication 186
- no sensor and Console communication on ISA server 187
- not seeing any BOEP events 189
- policy problems 178
- Tcl problems 185

trusted users

- monitoring 138

tuning 48, 66

- reference document 9

tuning parameter

- AllowAllAcknowledgementPackets 37
- evidence.logging 147
- Lookup Names 107
- packetlog.enabled 144
- packetLog.logging 144
- pcd.buffertuning 167–168
- pcd.exclusion 164–165
- pcd.failmode 173–174

- pcd.packetfilters 170–171
- sensor.noc2logszelimit 110
- sensor.syslogfile 60
- vpatch.BaseConfig 157
- vpatch.Exclude 159
- vpatch.Include 154

## U

UnsetData 126

user-defined

- binary log signatures 87
- network signature example 52
- network signature template 51
- network signatures 49
  - configuring 53
  - enabling 53
  - supported 49
- network signatures, PAM Help file 50
- policies 18
- signatures 26

user-defined binary log signatures 87

users

- monitoring 138

using

- firecell signatures 40
- wildcards to specify generic logs 81

## V

validation script values 134

validation scripts 123

variables

- global 120
- transient 120
- types of 120

vpatch.BaseConfig parameter

- Buffer Overflow Exploit Protection 157–158

vpatch.Exclude parameter

- Buffer Overflow Exploit Protection 159–160
- customize action 159–160

vpatch.Include 154

vpatch.Include parameter

- Buffer Overflow Exploit Protection 154–155

## W

Web server

- monitoring index.htm 137

Web site, IBM Internet Security Systems 10

Welch, Brent B. 119

---

**Index**

---

wildcards 81  
Windows NT 4.0 start-up procedure 183  
wtmpt binary log file 87  
wtmpt log monitoring  
    disabling 88  
    enabling 88

**X**

X-Press Update tab 27