Server Sensor Policy Guide

supporting
Version 6.5 for Linux
Version 6.5 for Windows
Version 6.5 for Solaris
## Contents

**Preface**  
Overview  
How to use Product Name Variable Documentation  
Conventions Used in this Guide  
Getting Technical Support

### Part I: Introduction

**Chapter 1: Introduction to the Server Sensor**  
Overview  
About RealSecure  
Deploying RealSecure Sensors  
About Server Sensor  
How Server Sensor Works

**Chapter 2: Server Sensor Policies**  
Overview  
About Server Sensor Policies  
Server Sensor Pre-Defined Policies  
Importing Policies  
About Server Sensor Policy Files

### Part II: Signatures

**Chapter 3: Introduction to Signatures**  
Overview  
About Server Sensor Signatures  
Customizing Pre-Defined Network and Log-based Signatures  
Customizing Protocol Ports  
Capturing Packet Information  
Monitoring Suspect Connections

**Chapter 4: User-Defined Signatures**  
Overview  
Section A: About User-Defined Signatures  
Overview  
Section B: User-Defined Log Monitoring Signatures  
Overview  
Monitoring Events Logs in Windows NT and Windows 2000  
Setting Up Information Fields for Windows Event Log User-Defined Signatures  
Auditing Files and Registry Entries with Windows Server Sensor  
Monitoring Log Files  
Monitoring Custom Events in Unix Syslogs  
Monitoring Local Syslog Events  
Monitoring the Wtmpx Binary Log File
## Contents

**Chapter 6: Fusion Scripting** ............................................. 95
**Part III: Fusion Scripting** ........................................... 81

### Section C: User–Defined Connection Signatures .......................... 59
- Overview ....................................................................... 59

### Section D: User–Defined Network Signatures ............................... 61
- Overview ....................................................................... 61
- Supported User–Defined Network Signatures ..................... 62
- Adding a User–Defined Network Signature ....................... 63

### Section E: User–Defined BSM Signatures ................................... 67
- Overview ....................................................................... 67
- About Solaris Basic Security Module Signatures ............... 68
- Task 1: Creating a User–Defined BSM Signature ............... 69
- Task 2: Generating the Event .......................................... 71
- Task 3: Using Praudit to Examine the Audit File ............... 72
- Task 4: Configuring the Server Sensor to Generate a Response ......................................................................................... 74
- Task 5: Configuring the Information Fields Responses should Return ......................................................................................... 76
- Task 6: Choosing Responses ............................................. 77
- Configuring BSM Audit Management ................................. 78
- Examples of User–Defined BSM Information ................. 79

### Chapter 5: Firecell Signatures ............................................... 81
- Overview ....................................................................... 81
- About Firecell Signatures ................................................. 82
- Understanding Firecell Signatures .................................... 83
- Creating and Configuring Firecell Signatures ..................... 85
- Disabling Firecell Signatures ............................................ 89
- Using Firecell Signatures .................................................. 90

### Part III: Fusion Scripting ..................................................... 81

#### Chapter 6: Fusion Scripting ................................................ 95
- Overview ....................................................................... 95

#### Section A: Introduction to Fusion Scripting ............................. 97
- Overview ....................................................................... 97
- Data Available to Fusion Scripting .................................... 100
- Tcl Script Categories Used in Fusion Scripting ............... 102

#### Section B: Working with Fusion Scripting ............................... 105
- Overview ....................................................................... 105
- Predefined Tcl Extensions ............................................... 106
- Adding or Modifying a Fusion Script ................................. 108
- Configuring a Fusion Scripting Response ......................... 109
- Configuring a Fusion Scripting SNMPv3 Response .......... 110
- Returning a True or False Result in a Validation Script ...... 113
- Using Fusion Scripts ....................................................... 114
- Disabling Fusion Scripting ............................................... 119
Part IV: Troubleshooting

Chapter 7: Troubleshooting ................................................................. 123
  Overview ......................................................................................... 123
  Isolating Policy Problems ............................................................... 124
  Tcl Script Problems ........................................................................ 131
  No Communication Between the Sensor and the Console ............. 132

Appendixes

Appendix A: Default Block Response Settings in Server Sensor 6.5 for Windows 135
Appendix B: Signatures that Block by Default in Server Sensor 6.5 for Windows 139
Appendix C: User–Defined Network Signature Category Descriptions and Configuration Information ........................................... 141
Appendix D: RealSecure to BlackICE Signature Mapping .................. 147
Appendix E: Configuring the RealSecure Web Server Monitoring Component ...................................................... 153
Index ................................................................................................ 155
Preface

Overview

Introduction
This guide describes RealSecure® Server Sensor policies, policy files, and signatures. This
guide also tells how the server sensor uses policies and signatures to protect your system
from attacks and misuse.

Purpose
This policy guide describes how to use server sensor to protect your system from attacks
and misuse.

Scope
This policy guide describes features that are specific to the server sensor. General
information about RealSecure sensors, such as managing policies, configuring responses,
and configuring sensors, is described in the RealSecure Workgroup Manager User Guide and
in the SiteProtector™ Installation and Configuration Guide.

Audience
This guide is intended for current and new users of RealSecure.

Note: If you plan to use the Fusion Scripting feature, you must have a thorough
understanding of the Tool Control Language (Tcl) scripting language.

What’s new in this release
New features in server sensor 6.5.1 include the following:

Kernel independence—Unlike previous versions for Linux, server sensor 6.5.1 is kernel
independent and is no longer negatively impacted by the upgrade of kernels. When you
upgrade your kernel simply restart the server sensor to activate RealSecure Intrusion
Protection.

Web Application Protection—server sensor 6.5.1 provides SSL intrusion monitoring,
analysis, and response capability for Apache 1.3 and 2.0.

Enterprise management support—server sensor 6.5.1 adds SNMP v3 as a Server Sensor
Tcl response to allow users to integrate server sensor into their existing infrastructure with
enterprise management applications.

Enhanced protection—server sensor 6.5.1 includes new signatures for server sensor and
full technical support for user-defined network signatures. User-defined network
signatures allow sophisticated users to write or customize their own signatures to identify
new attacks specific to their environment.
This guide was updated for the server sensor version 6.5 release for Linux and includes new or revised information about the following topics:

- SecureLogic is now Fusion Scripting. For more information, see “Fusion Scripting” on page 95.
How to use Product Name Variable Documentation

Using this guide  Read the entire guide before using your server sensor and then refer to the guide as necessary when using your server sensor.

Related publications  For additional information about server sensor, see the following publications:

- RealSecure Server Sensor Installation Guide
- RealSecure Workgroup Manager Installation Guide
- RealSecure Workgroup Manager User Guide
- SiteProtector Installation and Configuration Guide
- SiteProtector Strategy Guide
Conventions Used in this Guide

Introduction

This topic explains the typographic conventions used in this guide to make information in procedures and commands easier to recognize.

In procedures

The typographic conventions used in procedures are shown in the following table:

<table>
<thead>
<tr>
<th>Convention</th>
<th>What it Indicates</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>An element on the graphical user interface.</td>
<td>Type the computer’s address in the IP Address box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select the Print check box. Click OK.</td>
</tr>
<tr>
<td><strong>SMALL CAPS</strong></td>
<td>A key on the keyboard.</td>
<td>Press ENTER.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Press the PLUS SIGN (+).</td>
</tr>
<tr>
<td><strong>Constant width</strong></td>
<td>A file name, folder name, path name, or other information that you must type exactly as shown.</td>
<td>Save the User.txt file in the Addresses folder.</td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>A file name, folder name, path name, or other information that you must supply.</td>
<td>Type Version number in the Identification information box.</td>
</tr>
<tr>
<td><strong>Æ</strong></td>
<td>A sequence of commands from the taskbar or menu bar.</td>
<td>From the taskbar, select StartÆRun. On the File menu, select UtilitiesÆCompare Documents.</td>
</tr>
</tbody>
</table>

Table 1: Typographic conventions for procedures

Command conventions

The typographic conventions used for command lines are shown in the following table:

<table>
<thead>
<tr>
<th>Convention</th>
<th>What it Indicates</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constant width bold</strong></td>
<td>Information to type in exactly as shown.</td>
<td>md ISS</td>
</tr>
<tr>
<td><strong>Italic</strong></td>
<td>Information that varies according to your circumstances.</td>
<td>md your_folder_name</td>
</tr>
<tr>
<td><strong>[]</strong></td>
<td>Optional information.</td>
<td>dir [drive:] [path] [filename] [/P] [/W] [/D]</td>
</tr>
<tr>
<td>**</td>
<td>**</td>
<td>Two mutually exclusive choices.</td>
</tr>
<tr>
<td><strong>[]</strong></td>
<td>A set of choices from which you must choose one.</td>
<td>% chmod {u g o a}=[r</td>
</tr>
</tbody>
</table>

Table 2: Typographic conventions for commands
Getting Technical Support

Introduction
ISS provides technical support through its Web site and by email or telephone.

The ISS Web site
The Internet Security Systems (ISS) Resource Center Web site (http://www.iss.net/support/) provides direct access to frequently asked questions (FAQs), white papers, online user documentation, current versions listings, detailed product literature, and the Technical Support Knowledgebase (http://www.iss.net/support/knowledgebase/).

Support levels
ISS offers three levels of support:

- Standard
- Select
- Premium

Each level provides you with 24-7 telephone and electronic support. Select and Premium services provide more features and benefits than the Standard service. Contact Client Services at clientservices@iss.net if you do not know the level of support your organization has selected.

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

<table>
<thead>
<tr>
<th>Location</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas</td>
<td>24 hours a day</td>
</tr>
<tr>
<td>All other locations</td>
<td>Monday through Friday, 9:00 A.M. to 6:00 P.M. during their local time, excluding ISS published holidays</td>
</tr>
</tbody>
</table>

Note: 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 3: Hours for technical support

Contact information
The following table provides electronic support information and telephone numbers for technical support requests:

<table>
<thead>
<tr>
<th>Regional Office</th>
<th>Electronic support</th>
<th>Telephone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Connect to the MYISS section of our Web site: <a href="http://www.iss.net">www.iss.net</a></td>
<td>Standard: (1) (888) 447-4861 (toll free) (1) (404) 236-2700</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select and Premium: Refer to your Welcome Kit or call your Primary Designated Contact for this information.</td>
</tr>
<tr>
<td>Latin America</td>
<td><a href="mailto:support@iss.net">support@iss.net</a></td>
<td>(1) (888) 447-4861 (toll free) (1) (404) 236-2700</td>
</tr>
</tbody>
</table>

Table 4: Contact information for technical support
<table>
<thead>
<tr>
<th>Regional Office</th>
<th>Electronic support</th>
<th>Telephone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe, Middle East, and Africa</td>
<td><a href="mailto:support@iss.net">support@iss.net</a></td>
<td>(44) (118) 959-3900</td>
</tr>
<tr>
<td>Asia-Pacific and Philippines</td>
<td><a href="mailto:asia-support@iss.net">asia-support@iss.net</a></td>
<td>(63) (2) 886-6014</td>
</tr>
</tbody>
</table>
| Japan                           | support@isskk.co.jp | Domestic: (81) (3) 5740-4065  
Overseas (APAC): (81) (3) 5740-4066 |

Table 4: Contact information for technical support (Continued)
Introduction
Chapter 1

Introduction to the Server Sensor

Overview

Introduction

This chapter describes RealSecure and the server sensor. It also explains how to deploy the server sensor.

In this chapter

This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About RealSecure</td>
<td>4</td>
</tr>
<tr>
<td>Deploying RealSecure Sensors</td>
<td>5</td>
</tr>
<tr>
<td>About Server Sensor</td>
<td>6</td>
</tr>
<tr>
<td>How Server Sensor Works</td>
<td>8</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction to the Server Sensor

About RealSecure

Introduction
RealSecure is an automated, real-time intrusion detection and response system that unobtrusively analyzes activity across your computer systems and networks. A RealSecure system consists of the following:

- management component
- sensors

Management components
Choose one of the following management components for your RealSecure deployment:

- Workgroup Manager
- SiteProtector

The management component serves the following purposes:

- collects data from sensors using one or more event collectors
- stores data from sensors in a database
- displays data from sensors in a console

This guide describes how to apply policies using the Workgroup Manager.

Console
The console is the graphic user interface (GUI) for the system; use the console to control any sensors the console is managing.

Command Line Interface
Use the Command Line Interface (CLI) to manage your sensors in the following situations:

- if you do not have access to the console
- when you must integrate console capabilities with an existing process

Use the CLI to control sensors just as you would use the console.

Reference: For more information about the Command Line Interface, see the RealSecure Workgroup Manager User Guide.

RealSecure sensors
Sensors monitor network and system traffic for attacks and other security-related events and then generate responses to these events. RealSecure supports the following sensors:

- server sensor
- network sensor
- gigabit network sensor
Deploying RealSecure Sensors

Introduction
The server sensor monitors traffic to and from a single server; therefore, you should deploy a server sensor on any server you want to monitor traffic for. This topic shows a sample deployment.

Sample deployment
The following diagram shows how you could deploy a set of sensors and a Workgroup Manager on a network:

![Sample RealSecure deployment](image)

Figure 1: Sample RealSecure deployment
About Server Sensor

Introduction
The server sensor monitors traffic to and from a single server. In addition to detecting intrusions, the server sensor can also prevent intrusions by blocking specified network packets. The server sensor can also identify attacks destined for active services on the protected host.

Server sensor overview
The server sensor has the following attributes:
- detects both network and system events
- detects events at the application layer
- detects events before they reach the IP stack
- monitors traffic to and from the host it is installed on
- prevents intrusions
- provides a command line interface
- extends validation and response options with Fusion Scripting

Network and system event detection
The server sensor monitors both network and operating system (OS) activity.

Blocking
Network signatures on the server sensor can block offending packets. Blocking drops the packet that triggers an event and, for connection-based signatures such as HTTP and FTP, blocks subsequent packets associated with the same connection. Blocking prevents the event or packet of data from being processed by the operating system, protocol stack, or application.

**Blocking on Windows server sensors**—In server sensor 6.5 for Windows, blocking is not user-configurable and certain signatures are configured to block network packets by default. For a list of the server sensor 6.5 for Windows signatures that have blocking enabled by default, see Appendix B, “Signatures that Block by Default in Server Sensor 6.5 for Windows”.

**Blocking on Unix platform server sensors**—In server sensor 6.5 for Solaris and server sensor 6.5 for Linux, use the Block response to configure blocking.

Dynamic blocking
Dynamic blocking enables you to define the period of time that the server sensor blocks traffic. Use the Block response to configure dynamic blocking.

Event detection at the application layer
The 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.

Monitoring at the application layer provides the following benefits:
- The server sensor can analyze streams of data in addition to analyzing individual packets.
- The server sensor is not susceptible to packet fragmentation because TCP packets are reassembled above the stack.
The server sensor can monitor traffic that is encrypted with software, such as SSL, IPSEC, or SKIP, because the traffic is not encrypted above the stack.

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.

**Event detection before reaching the IP stack**

The 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.

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.

**Monitoring traffic to and from one host**

The 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 server sensor because the traffic workload is distributed at the host level.

**Intrusion prevention**

The 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 RealSecure Workgroup Manager User Guide or the SiteProtector Help.

- user-defined firecell signatures

  **Reference:** For more information about firecell signatures, see “About Firecell Signatures” on page 82.

**Note:** The server sensor only identifies attacks destined for active services. Attacks destined for services that don’t 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 server sensor collects.

**Reference:** For more information see Chapter 6, “Fusion Scripting”.

---

Server Sensor Policy Guide 7
How Server Sensor Works

Introduction  
This topic describes events, policies, policy files, and signatures. This topic also explains some of the limitations to a server sensor installation.

Events  
Events are attacks or misuse detected by the sensor that may result in an alert being sent to the console.

Policies  
The server sensor uses policies to control sensor behavior. Policies contain items, called signatures, which determine what types of events the sensor monitors for. Policies control the following sensor behaviors:

- the type of security events a sensor detects
- the priority of each event
- the sensor’s response to the event

You view, create, and edit policies from the management component. You also use the management component to apply new or updated policies to a sensor.

Signatures  
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.

Policy files  
Policy files control how policies are applied to a sensor. Each sensor has its own set of policy files.

Number of sensors a console can manage  
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:

- system configuration of the host running the management console software
- amount of traffic flowing between the sensor and the console
- number of signatures enabled in the policy
- the geographic and organizational limitations of the controlling organization
Chapter 2

Server Sensor Policies

Overview

Introduction

This chapter describes server sensor policies, lists the pre-defined policies that come with server sensor, describes how policy files ensure that the sensor runs the correct policy, and identifies the implications of importing earlier version server sensor policies.

In this chapter

This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Server Sensor Policies</td>
<td>10</td>
</tr>
<tr>
<td>Server Sensor Pre-Defined Policies</td>
<td>11</td>
</tr>
<tr>
<td>Importing Policies</td>
<td>13</td>
</tr>
<tr>
<td>About Server Sensor Policy Files</td>
<td>17</td>
</tr>
</tbody>
</table>
Chapter 2: Server Sensor Policies

About Server Sensor Policies

Introduction
Policies define types of events a server sensor monitors for and how the sensor responds to those events. This topic describes pre–defined policies and user–defined policies.

Pre-defined policies
The server sensor comes with several pre–defined policies configured for various levels of protection.

You cannot customize pre–defined policies.

User-defined policies
You can create user–defined, or customized, policies to monitor your system when none of the pre–defined policies meet your needs. Create a user–defined policy by deriving a new policy from the pre–defined 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 pre–defined policy, see “Deriving New Policies from Existing Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

How to work with policies
You can view, create, and edit policies with the management console. You also use the console to apply new or updated policies to a sensor.

Workgroup Manager Reference: For general information about working with policies through the console, see the “Working with Policies” chapter in the RealSecure Workgroup Manager User Guide.

SiteProtector Reference: For information about working with policies in SiteProtector, see the SiteProtector Help.
Server Sensor Pre-Defined Policies

Introduction

The server sensor has several pre-defined policies for various levels of protection. This topic describes the pre-defined policies that come with the server sensor and the types of events they monitor for.

Intrusion prevention

Pre-defined server sensor policies can filter and/or block network and system events. When the sensor blocks events, it prevents the event from being processed by the operating system, the protocol stack, or the application, thereby preventing damage to your system.

Events monitored by pre-defined policies

The pre-defined policies monitor the following types of intrusions and events:

- network events
- some OS events

Events not monitored by pre-defined policies

The pre-defined policies do not monitor the following types of events:

- Unix syslog events
- suspicious connection events
- firecell events

Description of pre-defined policies

The following table describes the pre-defined server sensor policies:

Note: Policies for server sensor version 6.5 are platform specific and they will only work reliably on the platform they were designed for. In addition, some signature groups may appear empty because there are no applicable signatures in that group for the current platform.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum_Windows</td>
<td>Use this policy on Windows operating systems if either of the following are true:</td>
</tr>
<tr>
<td></td>
<td>• you are using a non-IIS Web server, or</td>
</tr>
<tr>
<td></td>
<td>• you are not using a Web server</td>
</tr>
<tr>
<td></td>
<td>This policy logs and displays unusual and suspicious activity on the console. It also logs normal activity, but it does not display normal activity on the console.</td>
</tr>
<tr>
<td></td>
<td>Use caution when applying this policy because it may detect an unmanageable number of events.</td>
</tr>
</tbody>
</table>

Table 5: Pre-defined policies
### Chapter 2: Server Sensor Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
</table>
| Maximum_Solaris         | Use this policy on Solaris operating systems if either of the following are true:  
                          - you are using a non-Apache Web server, or  
                          - you are not using a Web server  
                          This policy logs and displays unusual and suspicious activity on the console. It also logs normal activity, but it does not display normal activity on the console.  
                          Use caution when applying this policy because it may detect an unmanageable number of events. |
| Maximum_Linux           | Use this policy on Linux operating systems if either of the following are true:  
                          - you are using a non-Apache Web server, or  
                          - you are not using a Web server  
                          This policy logs and displays unusual and suspicious activity on the console. It also logs normal activity, but it does not display normal activity on the console.  
                          Use caution when applying this policy because it may detect an unmanageable number of events. |
| Maximum_Windows_IIS     | Use this policy if you are using an IIS Web server on a Windows 2000 operating system.                                                      |
| Maximum_Solaris_Apache  | Use this policy if you are using an Apache Web server on the Solaris operating system.                                                       |
| Maximum_Linux_Apache    | Use this policy if you are using an Apache Web server on the Linux operating system.                                                        |
| Original_Windows        | This is the default policy on Windows operating systems. This policy offers a medium level of security. It monitors unusual activity and security events but does not log normal activity to the database. |
| Original_Solaris        | This is the default policy on Solaris operating systems. This policy offers a medium level of security. It monitors unusual activity and security events but does not log normal activity to the database. |
| Original_Linux          | This is the default policy on Linux operating systems. This policy offers a medium level of security. It monitors unusual activity and security events but does not log normal activity to the database. |

Table 5: Pre-defined policies (Continued)
Importing Policies

Introduction

There are several differences between server sensor versions 6.5 and earlier versions of server sensor. Because of these differences, you must run the to65policy.exe executable after importing and applying an earlier version policy before you can use the policy on a version 6.5 server sensor. You may also need to edit the imported policy. This topic describes how to use earlier version server sensor policies with RealSecure 6.5.

Prerequisite

Server sensor 6.5 has platform–specific policies. If you were using one policy on Windows and Unix platforms, you must import this policy for each platform you are using it on and save the policy with platform–specific names.

Example: If you import a policy called My_Maximum.policy, include Windows, Solaris, and Linux in the new policy name, such as My_Maximum_Windows.policy, My_Maximum_Solaris.policy, and My_Maximum_Linux.policy.

Importing earlier version policies to server sensor 6.5

The following table outlines how an earlier version policy imports to server sensor 6.5 and it also describes, where applicable, the actions you must take to configure the imported policy:

<table>
<thead>
<tr>
<th>Function</th>
<th>5.5 Policy</th>
<th>6.0 Policy</th>
<th>6.0.1 Policy</th>
<th>Action/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic block setting</td>
<td>not applicable</td>
<td>DynamicBlock</td>
<td>DynamicBlock</td>
<td>For Windows server sensors, reset the Dynamic Block Period in the Server Sensor tab of the Server Sensor Properties window. For Unix platform server sensors, use the DynamicBlock Advanced parameter to configure the dynamic block period. Reference: For more information on the Block response, see the RealSecure Workgroup Manager User Guide.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advanced parameter ignored on Windows server sensors ignored</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Block response settings</td>
<td>will be different for some signatures</td>
<td>will be different for some signatures</td>
<td>will not be different</td>
<td></td>
</tr>
<tr>
<td>Outbound firecell signatures</td>
<td>will not import</td>
<td>will not import</td>
<td>will not import</td>
<td>RealSecure 6.5 does not support outbound firecell signatures.</td>
</tr>
<tr>
<td>User-defined events</td>
<td>will import</td>
<td>will import</td>
<td>will import</td>
<td></td>
</tr>
<tr>
<td>Signature settings</td>
<td>will import</td>
<td>will import</td>
<td>will import</td>
<td>All signature settings except the dynamic block setting (as noted above) will import.</td>
</tr>
<tr>
<td>Signature responses</td>
<td>will import</td>
<td>will import</td>
<td>will import</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Importing earlier version policies to server sensor 6.5
To import a policy:

1. From the **View** menu, choose **Sensor Policies**.
   The Sensor Policies window appears.
2. Select the **Server** tab.
3. Click **Import Policy**.
   The Choose Source Policy File window appears.
4. Locate and select the policy you want to import, and then click **Open**.
   The Choose Target Policy File window appears.

   **Note:** You can change the location of the policy, but new imported policies are saved in the default directory. The default directory for server sensor policies is `C:\Program Files\ISS\RealSecure 6.5 Console\Server Policies`.

5. Type a platform-specific name for the imported policy.
   **Example:** If you import a policy called My_Maximum.policy, include Windows, Solaris, or Linux in the new policy name, such as My_Maximum_Windows.policy, My_Maximum_Solaris.policy, and My_Maximum_Linux.policy.

6. Select **Current** as the **Save As type**.
7. Click **Save**.
   A confirmation message appears.

8. Click **Yes**.
   A confirmation message informs you that the policy was imported successfully.

9. Click **OK**.
   The new policy appears on the Sensor Policies window and is available to apply to any server sensor you manage.

   **Note:** The new server sensor 6.5 signatures do not appear in the policy until you apply the policy to a version 6.5 server sensor. After applying an imported policy to a 6.5 sensor, check the policy to ensure the settings are as you intended.

10. Apply the policy to add the new server sensor 6.5 signatures, and then run the `to65policy.exe` executable to convert the imported policy file to the server sensor 6.5 format. For more information about the `to65policy.exe` executable, see “The `to65policy.exe` executable” on page 15.

   **Caution:** After upgrading an earlier version policy to a 6.5 policy, do not push the policy to a 5.5, 6.0, or 6.0.1 version server sensor. Version 6.5 policies cannot be used with earlier version server sensors.

### The `to65policy.exe` executable

When importing earlier version server sensor policies to use with server sensor 6.5, you must run the `to65policy.exe` executable to complete the import process.

**Exception:** You can import Solaris and Linux policy files and use them without running the `to65policy.exe` executable if you do not plan to monitor SSL traffic.

### Using the `to65policy.exe` executable

To use the `to65policy.exe` executable:

1. Import an earlier version server sensor policy.
   **Reference:** For more information on importing an earlier version policy, see “Importing an earlier version server sensor policy to a version 6.5 server sensor” on page 14.

2. Apply the imported policy to a version 6.5 server sensor.
   **Reference:** For more information about applying policies, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide*.

3. From the command line, type the following to run the `to65policy.exe` executable:
   ```
   to65Policy.exe policy_file_name platform
   ```
   Where `platform` is defined as follows:
   - `unix_apache` - for Unix platforms with Apache monitoring
   - `unix` - Unix platforms without Apache monitoring
   - `win2K_IIS` - Windows 2000 platforms with IIS
   - `windows` - all other Windows platforms: Windows 2000 platforms without IIS and Windows NT platforms (with or without IIS)
   The executable runs.

4. Reapply the policy to the sensor.
   The system completes the process of importing the policy.
Chapter 2: Server Sensor Policies

Policy implications when managing earlier version server sensors from RealSecure 6.5

You can manage an earlier version server sensor from the RealSecure 6.5 console. The following table outlines features that are affected by this configuration:

<table>
<thead>
<tr>
<th>Managing</th>
<th>5.5 sensor</th>
<th>6.0 sensor</th>
<th>6.0.1 sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound firewall signatures</td>
<td>not manageable from a 6.5 console</td>
<td>not manageable from a 6.5 console</td>
<td>not applicable</td>
</tr>
<tr>
<td>Network Events folder</td>
<td>new groups added in 6.5 appear empty</td>
<td>new groups added in 6.5 appear empty</td>
<td>new groups added in 6.5 appear empty</td>
</tr>
<tr>
<td>OS Events folder</td>
<td>new groups added in 6.5 appear empty</td>
<td>new groups added in 6.5 appear empty</td>
<td>new groups added in 6.5 appear empty</td>
</tr>
<tr>
<td>DynamicBlock setting</td>
<td>not applicable</td>
<td>edit the DynamicBlock Advanced parameter</td>
<td>not manageable from a 6.5 console</td>
</tr>
<tr>
<td>Do not push these version policies to this sensor</td>
<td>6.0, 6.0.1, or 6.5</td>
<td>5.5, 6.0.1, or 6.5</td>
<td>5.5, 6.0, or 6.5</td>
</tr>
</tbody>
</table>

Table 7: Managing earlier version server sensors from a RealSecure 6.5 console

Importing an earlier version server sensor policy to manage an earlier version server sensor

To import a policy:

1. From the View menu, choose Sensor Policies.
   The Sensor Policies window appears.
2. Select the Server tab.
3. Click Import Policy.
   The Choose Source Policy File window appears.
4. Locate and select the policy you want to import, and then click Open.
   The Choose Target Policy File window appears.
   **Note:** You can change the location of the policy, but new imported policies are saved in the default directory. The default directory for server sensor policies is `C:\Program Files\ISS\RealSecure 6.5 Console\Server Policies`.
5. Type a name for the imported policy.
   Include the sensor version number this policy should be used with in the policy name.
   **Example:** If you import a 6.0 policy called My_Maximum.policy, name the policy My_6.0_Maximum.policy.
6. Select No Changes (As Is) as the Save As type.
7. Click Save.
   A confirmation message appears.
8. Click Yes to confirm.
   A confirmation message informs you that the policy was imported successfully.
9. Click OK.
   The new policy appears on the Sensor Policies window and is available to apply to any server sensor you manage.

**Caution:** Do not apply policies you want to use with earlier version server sensors to a version 6.5 server sensor. After applying a policy to a 6.5 server sensor, the sensor updates the policy and you will no longer be able to use the policy with the earlier version sensor.
About Server Sensor Policy Files

Introduction

Policy files are executable files. You can use any text editor to edit them. Policy files control how policies are applied to the sensor. This topic describes the policy files used by the server sensor.

Understanding policy files

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

<table>
<thead>
<tr>
<th>Policy file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit.policy</td>
<td>The audit.policy file contains a list of global audit flags and a list of registry keys with audit flags set by the server sensor when you select the Enforce Audit Policy box on the Server Sensor tab.</td>
</tr>
<tr>
<td>common.policy</td>
<td>The common.policy file contains information used by the daemon and sensor, as follows:</td>
</tr>
<tr>
<td></td>
<td>• available cryptographic providers</td>
</tr>
<tr>
<td></td>
<td>• destination and community for any SNMP traps</td>
</tr>
<tr>
<td></td>
<td>• flags for automatic SNMP trap generation (upon sensor start, upon sensor policy change, etc.)</td>
</tr>
<tr>
<td></td>
<td>• a list of variable response types</td>
</tr>
<tr>
<td>current.policy</td>
<td>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.</td>
</tr>
<tr>
<td>Default.policy</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Note: Default.policy does not provide SSL protection. To enable SSL attack detection, apply or derive a new policy from one of the following policies:</td>
</tr>
<tr>
<td></td>
<td>• Maximum_Solaris_Apache.policy</td>
</tr>
<tr>
<td></td>
<td>• Maximum_Linux_Apache.policy</td>
</tr>
<tr>
<td></td>
<td>• Maximum_Windows_IIS.policy</td>
</tr>
<tr>
<td>eventlog.policy</td>
<td>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.</td>
</tr>
<tr>
<td>issCSF.policy</td>
<td>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.</td>
</tr>
<tr>
<td>issDaemon.policy</td>
<td>The issDaemon.policy file contains information used by the daemon, for example:</td>
</tr>
<tr>
<td></td>
<td>• sensor utilization timeout: how long the daemon tries to connect to a sensor before it gives up</td>
</tr>
<tr>
<td></td>
<td>• daemon port: the port number used by the daemon for sensor communication (normally 2998)</td>
</tr>
<tr>
<td></td>
<td>• master console: the console that has master status of the sensor controlled by the daemon.</td>
</tr>
</tbody>
</table>

Table 8: Server sensor policy file descriptions
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 pre-defined 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>
<thead>
<tr>
<th>Policy file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>push.policy</td>
<td>The <code>push.policy</code> 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 <code>push.policy</code>. After <code>push.policy</code> is merged with <code>current.policy</code>, the sensor deletes <code>push.policy</code>.</td>
</tr>
<tr>
<td>ruledef.policy</td>
<td>The <code>ruledef.policy</code> file contains definitions for pre-defined signatures.</td>
</tr>
<tr>
<td>update.policy</td>
<td>The <code>update.policy</code> 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 <code>update.policy</code>. When you apply an imported policy to the sensor, the sensor merges the contents of <code>update.policy</code> with the imported policy so that all the new signatures are available. The resulting configuration is written to <code>current.policy</code> so that the <code>current.policy</code> file represents the latest policy configuration for the sensor.</td>
</tr>
</tbody>
</table>

Table 8: Server sensor policy file descriptions (Continued)
Signatures
Chapter 3

Introduction to Signatures

Overview

Introduction

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

In this chapter

This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topics</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Server Sensor Signatures</td>
<td>22</td>
</tr>
<tr>
<td>Customizing Pre–Defined Network and Log–based Signatures</td>
<td>24</td>
</tr>
<tr>
<td>Customizing Protocol Ports</td>
<td>25</td>
</tr>
<tr>
<td>Capturing Packet Information</td>
<td>26</td>
</tr>
<tr>
<td>Monitoring Suspect Connections</td>
<td>28</td>
</tr>
</tbody>
</table>
About Server Sensor Signatures

Introduction
Policies, discussed in Chapter 2, consist of signatures. This topic describes server sensor signatures.

The server sensor includes the following types of signatures:

- firecell signatures, which protect the server
- network signatures (block response), which protect the network
- operating system signatures, which monitor activity on the operating system through log files

Signature organization in the policy editor
The policy editor organizes server sensor signatures under the tabs described in the following table:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protect</td>
<td>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. <strong>Reference:</strong> For more information about firecell signatures, see Chapter 5, “Firecell Signatures” on page 81. Use connection event signatures to monitor for suspicious connection to ports and services. <strong>Reference:</strong> For more information about connection event signatures, see “Monitoring Suspect Connections” on page 28.</td>
</tr>
<tr>
<td>Network Events</td>
<td>Displays network–based signatures. Network–based signatures monitor network traffic for content that can indicate an attack or other suspicious activity.</td>
</tr>
<tr>
<td>OS Events</td>
<td>Displays log–based signatures. The OS Events signatures watch activity that happens at the operating system level by monitoring system log files.</td>
</tr>
<tr>
<td>X-Press Updates</td>
<td>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. <strong>Reference:</strong> For information about installing X–Press Updates, see the RealSecure Workgroup Manager User Guide or the SiteProtector Help.</td>
</tr>
</tbody>
</table>

Table 9: Server sensor signature categories

Pre–defined signatures
The server sensor comes with several pre–defined signatures contained in the pre–defined policies.
<table>
<thead>
<tr>
<th>Pre-defined signature attributes you can customize</th>
<th>You can customize a pre-defined signature to meet your specific needs by editing the following attributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- priority settings</td>
</tr>
<tr>
<td></td>
<td>- responses</td>
</tr>
<tr>
<td></td>
<td>- protocol ports (for Network Event signatures)</td>
</tr>
<tr>
<td></td>
<td>- list of important files and list of registry keys (for certain OS Event signatures)</td>
</tr>
<tr>
<td></td>
<td>Table 10, “Pre-defined signature customizable attributes” on page 24 explains these attributes in more detail.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-defined signature attributes you cannot customize</th>
<th>When you work with pre-defined network signatures in server sensor version 6.5 for Windows, you must not change the TID setting in the Optional tab in the Advanced window.</th>
</tr>
</thead>
</table>

| User-defined signatures | You can create custom signatures if the pre–defined signatures do not meet your exact needs. See Chapter 4, “User-Defined Signatures” starting on page 31 for more information about user-defined signatures. |

| Limitations | The Event Propagation options in the Advanced window do not function for server sensor for Windows 6.5. |
Chapter 3: Introduction to Signatures

Customizing Pre–Defined Network and Log–based Signatures

Introduction

This topic describes how customize a pre–defined network event or log–based signature.

Procedure

To customize a network event or log–based signature:

1. Open the policy that contains the signature you want to customize.
2. Click the **Network Events** tab or 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 “Working with Responses” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
7. Click the **Save** icon.
   The system saves the changes you made to the policy.
8. Apply the new policy to the sensor(s) that you want to use this policy.
   **Reference:** For more information about applying policies, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.

Field descriptions

The following table describes the pre–defined signature attributes you can customize from the Security Events pane:

<table>
<thead>
<tr>
<th>Use...</th>
<th>To...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>enable or disable the signature</td>
</tr>
<tr>
<td>Priority</td>
<td>define the priority level assigned to the signature</td>
</tr>
<tr>
<td>Response</td>
<td>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. <strong>Reference:</strong> For more information about responses, see “Working with Responses” in the <em>RealSecure Workgroup Manager User Guide</em> or the SiteProtector Help.</td>
</tr>
<tr>
<td>Ports</td>
<td>specify port numbers that you want the sensor to associate with a particular protocol <strong>Reference:</strong> For more information about using protocol ports, see “Customizing Protocol Ports” on page 25.</td>
</tr>
<tr>
<td>Important Files list</td>
<td>view the files that the sensor monitors <strong>Note:</strong> This list is only available on signatures that monitor important files.</td>
</tr>
<tr>
<td>Registry keys list</td>
<td>view the registry keys that the sensor monitors <strong>Note:</strong> This list is only available on signatures that monitor registry keys.</td>
</tr>
</tbody>
</table>

**Table 10:** Pre-defined signature customizable attributes
Customizing Protocol Ports

Introduction
The server sensor uses protocol ports to determine which ports signatures use. This topic discusses the use of protocol ports.

Using protocol ports
Protocol ports classify port numbers by protocol, by signature, or by any other type of classification. You can change the port numbers associated with these classifications in the Protocol Ports window in Workgroup Manager.

Sharing protocol ports among policies
Protocol ports are specific to each policy and are not shared among policies.

Associating protocol ports with signatures
Currently, you cannot associate protocol ports with a user-defined signature or change the protocol port that a particular pre-defined signature uses.

Changing port number classifications
To change the ports associated with a class of ports, make the changes on a policy-by-policy basis.

Caution: Changing the value of the ports affects any signature that is using that port classification.

Procedure
To change the value of a port:

1. Open the policy you want to change the protocol port for.
2. Select the Network Events tab, and then click the name of any group under the Network Events folder.
   A list of signatures in that group appears in the right pane.
3. Click Ports.
   The Protocol Ports window opens.
4. Select the type of protocol port that you want to change, and then click Edit.
   The Edit Protocol Ports window opens.
5. Type the numbers of the ports you want associated with this protocol in the Ports box.
   Note: Use spaces to separate ports; use a dash to indicate a range of ports.
6. Click OK.
7. Click OK.
8. Apply the changed policy to the appropriate sensor(s).

Reference: For more information about applying policies, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.
Capturing Packet Information

Introduction

In server sensor for Windows version 6.5, you can capture information about packets that are received by the local system. Server sensor uses the following types of logging to collect this information:

- packet logging
- evidence logging

Packet logging

Packet logging creates a copy of every packet that arrives at the local 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 creates a maximum of twenty, 10MB files named \logxxx.enc, where xxx is 000 through 019. These files are created in the server_sensor_1/BlackICE directory.

Packet logging Name/Value pairs

The following table lists the packet logging Name/Value pairs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PacketLogLogging</td>
<td>TRUE</td>
<td>Enables packet logging</td>
</tr>
<tr>
<td>PacketLogLogging</td>
<td>FALSE</td>
<td>Disables packet logging</td>
</tr>
</tbody>
</table>

Table 11: Packet logging Name/Value pairs

Note: The default PacketLogLogging value is FALSE. When you import policies from an earlier version server sensor, the PacketLogLogging value defaults to FALSE.

Evidence logging

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

Evidence logging creates up to 32 evidence files named evcxxx.enc, where xxx is 000 through 031. These files are created in the server_sensor_1/BlackICE directory.

Evidence logging Name/Value pairs

The following table lists the evidence logging Name/Value pairs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvidenceLogging</td>
<td>TRUE</td>
<td>Enables evidence logging</td>
</tr>
<tr>
<td>EvidenceLogging</td>
<td>FALSE</td>
<td>Disables evidence logging</td>
</tr>
</tbody>
</table>

Table 12: Evidence logging Name/Value pairs

Note: The default evidence logging value is TRUE. When you import policies from an earlier version server sensor, evidence logging defaults to TRUE.

Enabling or disabling packet capture

To enable or disable packet capturing:

1. Open the policy you want to enable or disable packet capturing for.
2. Click the Network Events tab.
3. Double-click Network Events, and then click Advanced_Settings.
4. Click the **Advanced** button.
   The **Advanced** window opens.
5. Double-click on **Evidence Logging** or **PacketLogLogging** to toggle the **Value** to **TRUE** or **FALSE**.
6. Click **OK**.
7. Save the policy, and then apply the changed policy to the sensor.
## Monitoring Suspect Connections

### Introduction
To monitor suspect connections, configure the appropriate connection–event signature. This topic describes how to configure connection–event signatures to monitor for suspect connections.

### Background:
**Connection events**
A connection event occurs if a computer attempts to open a connection to a port on the computer that hosts the server sensor. Unlike other events, the server 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.

### Pre-defined connection event signatures
The server sensor policy contains pre–defined connection event signatures for different types of connections, such as WWW, FTP, and Telnet.

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

**Reference**: For more information about user–defined connection event signatures, see “User–Defined Connection Signatures” on page 59

### 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 to which you want to add a connection event signature.
2. Click the **Protect** tab.
3. Double-click **ProtectÆConnectionsÆSuspect 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 “Working with Responses” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
7. Click **Save**.
   - The system saves your changes.
8. Apply the changed policy to the appropriate sensor(s).
   - **Reference**: For more information about applying policies, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
Field descriptions

The following table describes the connection–event signature fields you can configure from the Security Events pane:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Enables or disables the signature</td>
</tr>
<tr>
<td>Priority</td>
<td>Defines the importance of each signature. This is used to sort events into</td>
</tr>
<tr>
<td></td>
<td>the High, Medium, and Low Priority windows and to generate reports.</td>
</tr>
<tr>
<td>Response</td>
<td>Configure the sensor to respond when it detects an event that matches the</td>
</tr>
<tr>
<td></td>
<td>signature. Each signature can have any combination of responses or no</td>
</tr>
<tr>
<td></td>
<td>response at all. For more information about responses, see “Working with</td>
</tr>
<tr>
<td></td>
<td>Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.</td>
</tr>
</tbody>
</table>

Table 13: Customizable connection event signature attributes
Chapter 4

User-Defined Signatures

Overview

Introduction

You may need to create custom signatures if the pre-defined signatures do not meet your needs. This chapter describes how you create and configure user-defined signatures.

In this chapter

This chapter contains the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section A, &quot;About User-Defined Signatures&quot;</td>
<td>33</td>
</tr>
<tr>
<td>Section B, &quot;User-Defined Log Monitoring Signatures&quot;</td>
<td>35</td>
</tr>
<tr>
<td>Section C, &quot;User-Defined Connection Signatures&quot;</td>
<td>59</td>
</tr>
<tr>
<td>Section D, &quot;User-Defined Network Signatures&quot;</td>
<td>61</td>
</tr>
<tr>
<td>Section E, &quot;User-Defined BSM Signatures&quot;</td>
<td>67</td>
</tr>
</tbody>
</table>
SECTION A: About User–Defined Signatures

Overview

Introduction

You can create custom signatures to monitor your system when pre–defined signatures do not meet all of your needs. This section describes the types of events user–defined signatures can monitor for and how you set up a user-defined signature.

Monitoring with user–defined signatures

User–defined signatures can monitor the following:

- any primary log source for specific data. For more information on log monitoring, see Section B, “User–Defined Log Monitoring Signatures” starting on page 35.

  **Note:** For more information on monitoring the Solaris BSM, see Section E, “User–Defined BSM Signatures” starting on page 67.

- suspect connection events for detecting the suspicious use of unused ports. For more information on monitoring suspect connection events with user–defined signatures, see Section C, “User–Defined Connection Signatures” starting on page 59.

- network traffic. For more information on monitoring network traffic, see Section D, “User–Defined Network Signatures” starting on page 61.

- events in the BSM

  **Note:** For more information on monitoring the Solaris BSM, see Section E, “User–Defined BSM Signatures” starting on page 67.

Prerequisite

When you create a user–defined signature, you frequently have to use a regular expression to configure the sensor to detect the information you want to monitor for. You may also want to use a regular expression to configure the information the sensor retrieves about the event. Because user–defined signatures frequently use regular expressions, you should be familiar with how to use them.

Regular expression libraries used by the server sensor

The server 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 server sensor. The tools egrep, awk, lex and flex also use this syntax.

Where to use regular expressions

You can use the Regular Expression box to define an event. You can also use the Info window to define the information about the event that the sensor includes in responses.

Specifying Name/Value pairs in the Info window

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

- **Name**—a name, provided by you, that accurately 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.
# Info window data identifiers

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

<table>
<thead>
<tr>
<th>Data Identifiers</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| @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. | If the application name is in the eighth string, use the following settings for the Name/Value pair:  
  - Application  
  - @String7 |
| @Field N         | 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. | A typical syslog message might have the following format:  
  Mar 15 10:25:30 everest  
  sendmail[28244]: authdes_refresh: keyserv(1m)...  
  To extract the host name from this message, you would use the following values:  
  - Host  
  - @Field3 |
| {}               | Used to extract a substring from a string. {} is a wildcard that pulls a substring located between two defined text entries in the entry. | To capture the user name for a user with a failed login, use the following settings for the Name/Value pair:  
  - User  
  - User {} failed to |

Table 14: Info window data identifiers
SECTION B: User–Defined Log Monitoring Signatures

Overview

Introduction

User–defined signatures can monitor any primary log source for specific data. This section describes which log sources user–defined signatures can monitor and also describes how to create user–defined signatures for log monitoring.

Data sources

Use user–defined signatures to monitor the following primary log sources:

- Windows NT and Windows 2000 Event Logs
- any ASCII log file
- Unix syslog messages from a local Unix system
- Solaris Basic Security Module (BSM) log files

Note: For more information on monitoring the Solaris BSM, see Section E, "User–Defined BSM Signatures" starting on page 67.

In this section

This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring Events Logs in Windows NT and Windows 2000</td>
<td>36</td>
</tr>
<tr>
<td>Setting Up Information Fields for Windows Event Log User–Defined Signatures</td>
<td>39</td>
</tr>
<tr>
<td>Auditing Files and Registry Entries with Windows Server Sensor</td>
<td>42</td>
</tr>
<tr>
<td>Monitoring Log Files</td>
<td>46</td>
</tr>
<tr>
<td>Monitoring Custom Events in Unix Syslogs</td>
<td>50</td>
</tr>
<tr>
<td>Monitoring Local Syslog Events</td>
<td>52</td>
</tr>
<tr>
<td>Monitoring the Wtmpx Binary Log File</td>
<td>53</td>
</tr>
<tr>
<td>Selecting Logs</td>
<td>55</td>
</tr>
<tr>
<td>Specifying Exceptions when Monitoring Log Files</td>
<td>56</td>
</tr>
</tbody>
</table>
Chapter 4: User-Defined Signatures

Monitoring Events Logs in Windows NT and Windows 2000

Introduction
If the pre-defined Windows signatures do not capture the security information you need, you can create a user-defined signature that enables the server sensor to monitor and respond to anything that is recorded in the Windows Event Log. This topic describes how to create an Event Log signature for Windows NT or Windows 2000.

Prerequisites
Before you create a signature to monitor the Windows Event Log, you must know the following:

- the name of the log where the activity is recorded. In the Workgroup Manager, this field is called “Origin”
  
  **Reference:** To find out more about these logs, refer to the Help for the Event Viewer. To access the Event Viewer, see “Opening the Microsoft Event Viewer in Windows NT” in the RealSecure Help.
- the values of the source, type, category, and ID fields in the policy editor
- the kind of information the Event Log saves and on what line of the event the information occurs
- how to use regular expressions and how to use exceptions in regular expressions if the information that you want the sensor to detect varies

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.
4. Click **Add**. The Enter a name window appears.
5. Type a name for the user-defined signature, and then click **OK**. 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. Set the priority of this signature in the **Priority** box.
8. Configure the signature to detect specific events using the **Origin**, **Source**, **Type**, **Category**, and **ID** boxes.
   **Reference:** For information about how to use these boxes, see Table 15 on page 37.
9. Click **Info**, and then add the information fields that you want the sensor to return whenever it detects an event that matches this signature.
   **Reference:** “Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 39 describes how to find the lines of the Event Log entry you want and the format to use to extract each line.
10. Click **OK**.
11. In the Regular Expression box, type the text or regular expression for the sensor to use to determine if an event is a match for this signature.
**Note:** 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 thing to start with when creating a user–defined signature to monitor the Event Log.

**Example:** The regular expression Virus (Found | Detected) monitors the operating system log file for the following entries:

- Virus Found
- Virus Detected

**Reference:** For information about the regular expressions you can use in this box, see “Regular expression libraries used by the server sensor” on page 33 and “Specifying Exceptions when Monitoring Log Files” on page 56.

12. Select the responses that you want the sensor to take whenever it detects an event that matches this signature.

**Reference:** For information about responses, see “Working with Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

13. Click **Save**.

14. Apply the policy to the sensor or sensors that are going to use this signature.

**Reference:** For a procedure on applying a policy, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

### Fields in Windows Event Log signatures

The following table describes the fields you can configure when creating Windows Event Log signatures:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Enables or disables the signature.</td>
</tr>
<tr>
<td>Event</td>
<td>Displays the name of the user–defined signature.</td>
</tr>
<tr>
<td>Priority</td>
<td>Assigns the priority for this event.</td>
</tr>
<tr>
<td>Response</td>
<td>Configures 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.</td>
</tr>
<tr>
<td>Origin</td>
<td>Specifies the log file the event originates from. The values for this field are as follows:</td>
</tr>
<tr>
<td></td>
<td>- Application (Windows NT and Windows 2000)</td>
</tr>
<tr>
<td></td>
<td>- Directory Service (Windows 2000 only)</td>
</tr>
<tr>
<td></td>
<td>- DNS Server (Windows 2000 only)</td>
</tr>
<tr>
<td></td>
<td>- File Replication Service (Windows 2000 only)</td>
</tr>
<tr>
<td></td>
<td>- Security (Windows NT and Windows 2000)</td>
</tr>
<tr>
<td></td>
<td>- System (Windows NT and Windows 2000)</td>
</tr>
<tr>
<td></td>
<td>These values match the Event Viewer values for Event Logs.</td>
</tr>
</tbody>
</table>

**Table 15:** Field descriptions: Eventlog signatures
Chapter 4: User-Defined Signatures

Buttons in Windows Event Log signatures

The following table describes the buttons in Windows Event Log signatures:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Specifies the information about the event that the sensor returns when an</td>
</tr>
<tr>
<td></td>
<td>event matches the signature.</td>
</tr>
<tr>
<td></td>
<td>Reference: For more information about using the Info window, see</td>
</tr>
<tr>
<td></td>
<td>“Setting Up Information Fields for Windows Event Log User–Defined Signatures”</td>
</tr>
<tr>
<td>Audit</td>
<td>Specifies files or registry keys to audit.</td>
</tr>
<tr>
<td></td>
<td>Reference: For more information about using the Audit window, see</td>
</tr>
<tr>
<td></td>
<td>“Auditing Files and Registry Entries with Windows Server Sensor” on page 42.</td>
</tr>
<tr>
<td>Fusion Scripting</td>
<td>Displays Fusion scripts used by this signature.</td>
</tr>
<tr>
<td></td>
<td>Reference: For more information about using Fusion Scripting, see</td>
</tr>
<tr>
<td></td>
<td>Chapter 6, “SecureLogic” starting on page 95.</td>
</tr>
</tbody>
</table>

Table 15: Field descriptions: Eventlog signatures (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Specifies the application that generates the event. This field matches the</td>
</tr>
<tr>
<td>Source column</td>
<td>in the Event Viewer.</td>
</tr>
<tr>
<td>Note</td>
<td>If you are specifying a security/system log event, you do not have to</td>
</tr>
<tr>
<td></td>
<td>specify the event source. However, you should always specify the Source</td>
</tr>
<tr>
<td></td>
<td>in the application log when defining an application log event.</td>
</tr>
<tr>
<td>Type</td>
<td>Matches the Windows NT Event Log. There are six values for this field:</td>
</tr>
<tr>
<td></td>
<td>0 = all types</td>
</tr>
<tr>
<td></td>
<td>1 = type of Error</td>
</tr>
<tr>
<td></td>
<td>2 = type of Warning</td>
</tr>
<tr>
<td></td>
<td>4 = type of Information</td>
</tr>
<tr>
<td></td>
<td>8 = Success Audit</td>
</tr>
<tr>
<td></td>
<td>16 = Failure Audit</td>
</tr>
<tr>
<td>Category</td>
<td>Matches the Windows Event Log category (type 0 to match all categories).</td>
</tr>
<tr>
<td>ID</td>
<td>Matches the specific Windows event ID. This can be determined by</td>
</tr>
<tr>
<td></td>
<td>inspecting the event in the Windows Event Log Viewer.</td>
</tr>
<tr>
<td>Regular Expression</td>
<td>Matches the regular expression or simple character string with Windows</td>
</tr>
<tr>
<td></td>
<td>Event Log messages.</td>
</tr>
<tr>
<td>Reference</td>
<td>See “Regular expression libraries used by the server sensor” on page 33 and</td>
</tr>
<tr>
<td></td>
<td>“Specifying Exceptions when Monitoring Log Files” on page 56 for more</td>
</tr>
<tr>
<td></td>
<td>information.</td>
</tr>
</tbody>
</table>

Table 16: Button descriptions: Eventlog signatures
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 other responses and set up information fields. The information fields log information in the RealSecure 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. Create an NT Event Log user-defined signature to monitor the event you want to detect.
   
   Reference: For a detailed procedure, see “Monitoring Events Logs in Windows NT and Windows 2000” on page 36.

2. In the left pane of the Policy Editor, select the signature you just created.

3. Click the Info button, and then add an information field to record each line of the event in the Windows Event Log.
   
   Reference: For more information about the Info window, see “Specifying Name/Value pairs in the Info window” on page 33.

   Note: Each information field you enter is just a place holder until you determine the real 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:

   ![Info Window Example]

   4. Click OK, and then click Save.

   The system saves your changes to the information fields, and then saves your changes to the policy.

5. Apply the policy to a sensor you can use to test the information fields.
   
   Reference: For a procedure on applying a policy, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

6. Create the event you want to monitor for, and then use the management console to examine the information fields that show up in the event.
Example: The following figure shows how an event looks in the RealSecure event inspector. Notice the information fields in the **Info Type** and **Info Value** columns.

![Event Inspector Screenshot]

7. Compare this event to the event recorded in the Windows Event Log.

Example: The following figure shows an example of an event in the Event Viewer for Windows NT. The information fields (@StringN) that you set up in RealSecure match the lines of text in the Description box.

![Event Viewer Screenshot]

As you compare the events in the two viewers, you can see how the @StringN information fields match the lines in the Windows Event Log.

8. Go to the policy with the user-defined signature, 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
9. Click **OK**, and then click **Save**.

   The system saves your changes to the information fields, and then saves your changes to the policy.

10. Apply the policy to a sensor you can use to test the information fields.

    **Reference:** For a procedure on applying a policy, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
Auditing Files and Registry Entries with Windows Server 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. This topic describes how to create a signature that audits files or registry entries.

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

- Make sure that the file or directory you want to audit exists. If you apply a policy that contains a signature that monitors for something that does not exist, the sensor does not monitor that file or directory until it exists and you restart the sensor.
- 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 “Setting Up Information Fields for Windows Event Log User–Defined Signatures” on page 39.

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.
   All signatures that currently exist in the NT 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.
   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:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>The priority that you want to assign to this signature.</td>
</tr>
<tr>
<td>Origin</td>
<td>The security level you want to assign to this signature.</td>
</tr>
<tr>
<td>Response</td>
<td>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.</td>
</tr>
</tbody>
</table>

Reference: For more information about responses, see the “Working with Responses” chapter in the RealSecure Workgroup Manager User Guide or 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 the **Audit** button.

The Audit window appears.

10. Click the **File** tab, and then refer to the following table:

<table>
<thead>
<tr>
<th>To audit...</th>
<th>In the File List box, type...</th>
</tr>
</thead>
</table>
| all the files in a particular 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.  
  **Example:** To monitor all files in the \c:\temp directory, type: \c:\temp\*  
  2. Click Add. |
| a directory but no the files in the directory     | 1. The full path to the directory.  
  **Example:** To monitor changes to the \c:\temp directory, type: \c:\temp  
  2. Click Add. |
| files with names that match a specific pattern    | 1. The pattern to match.  
  You can use the * or the ? wildcards to define the pattern.  
  **Example:** To monitor all files with a .log extension, type: *.log  
  2. Click Add. |
| a single file                                     | 1. The full path to the directory and the name of the file.  
  **Example:** To monitor changes to the log.txt file in the \temp directory, type: \c:\temp\log.txt  
  2. Click Add. |

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 log to the database.

   **Reference:** For a detailed procedure on setting up information fields, see “Setting Up Information Fields for Windows Event Log User–Defined Signatures” on page 39.

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, and then go to Step 21.

20. Leave the Regular Expression box empty.

21. Click Save.

   The system saves your changes.

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

Reference: For a procedure on applying a policy, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

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

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

Table 17: Audit window check boxes
### Table 17: Audit window check boxes (Continued)

<table>
<thead>
<tr>
<th>Audit Type</th>
<th>Audit Name</th>
<th>Monitors attempts to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry</td>
<td>Query</td>
<td>read a key's value</td>
</tr>
<tr>
<td></td>
<td>Set Value</td>
<td>create or modify a key's value</td>
</tr>
<tr>
<td></td>
<td>Create Subkey</td>
<td>create a subkey</td>
</tr>
<tr>
<td></td>
<td>Enum Subkeys</td>
<td>collect key values in a list when the specific names are not known (enumerated)</td>
</tr>
</tbody>
</table>
| Notify     |            | triggers a condition that results in a notification message sent to a monitored response. Notification messages of this type are sent when:  
|            |            | • a subkey is added or deleted  
|            |            | • a value is added, changed, or deleted  
|            |            | • the attributes of the key are changed  
|            |            | • the key's security descriptor is changed |
| 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 |
Chapter 4: User-Defined Signatures

Monitoring Log Files

Introduction

If the server 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. This topic describes how you monitor generic log files.

Prerequisites

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

- 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 server sensor” on page 33 and “Specifying Exceptions when Monitoring Log Files” on page 56.

- where the log files that you want to monitor reside

- the exact information in the logs you want to monitor

Using wildcards to select logs

When you monitor log files, you can use wildcards to specify generic log files.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Wildcards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT or 2000</td>
<td>* = 0 or more characters</td>
</tr>
<tr>
<td></td>
<td>? = any one character</td>
</tr>
<tr>
<td>Unix</td>
<td>* = 0 or more characters</td>
</tr>
<tr>
<td></td>
<td>? = any one character</td>
</tr>
<tr>
<td></td>
<td>[...] = range of characters</td>
</tr>
</tbody>
</table>

Table 18: 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.
4. Click Add.
   
   The Enter a name window appears.
5. Type a name for the user-defined signature, and then click OK.
   
   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 ellipse [...] 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?
   a. If yes, go to Step 11.
   b. If no, go to Step 9.

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

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Name</td>
<td>Type the name for this group of log files.</td>
</tr>
</tbody>
</table>
| Log Paths                    | Type the full path and file name to each log in this group. Use the pipe symbol (|) between logs.  
   **Example:**
   c:\temp\logerror*.log|c:\temp\log*.txt
   **Reference:** For information about using wildcards, see “Using wildcards to select logs” on page 46. |
| 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:

<table>
<thead>
<tr>
<th>Box or list</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Select the priority that you want to assign to this event.</td>
</tr>
</tbody>
</table>
| Regular Expression   | The string of text or regular expression that you want to trigger this signature. 
   **Example:** The regular expression [Ss]ucc| [Ff]ailue monitors the file for any of the following words: 
   • Success 
   • success 
   • Failure 
   • failure 
   **Reference:** See “Regular expression libraries used by the server sensor” on page 33 and “Specifying Exceptions when Monitoring Log Files” on page 56 for more information. |
| 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 responses at all. 
   **Reference:** For more information about responses, see “Working with Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help. |

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 33.

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 RealSecure Workgroup Manager User Guide or the SiteProtector Help.
Chapter 4: User-Defined Signatures

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

### 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 server sensor determines the newest file by comparing the last modification time (not the creation time of the file.)

### Specifying newest or all files

To create a signature to specify monitoring for newest or all files:

1. Open the policy that contains the event you want to customize.
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. Select the log event that you want to modify.
5. Click the ellipse [...] to the right of the Use Logs section.
6. Click **Add**.
    
   The Add New Log window appears.
7. Type a log name and log path.
8. Do one of the following:
   
   - To specify the newest-only files, select the check box for **Monitor ONLY most recently changed log**.
   - To specify all files, clear the check box for **Monitor ONLY most recently changed log**.
9. Click **OK**.
10. Select the new entry from the list of log names, and then click **OK**.
11. Enable the signature.
12. Save the policy, and then reapply the policy to the server sensor.

### File rotation

When a log file grows to a certain size, the underlying application may rotate the log to a new log file. You can create a log group with a file name pattern that includes all the possible names of the log file and set the Newest Only flag.

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 the application renames a large log file so that the next log message goes to a file with the same name, but it is actually a new file. For this situation, the server sensor keeps a mark for each log file it monitors. Each time a mark is changed, the server sensor 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.
Fields in generic log file signatures

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Enables or disables the signature.</td>
</tr>
<tr>
<td>Event</td>
<td>Displays the name of the user–defined signature. This is a display-only field.</td>
</tr>
<tr>
<td>Priority</td>
<td>Assigns the priority for this event.</td>
</tr>
<tr>
<td>Response</td>
<td>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. Reference: For more information about responses, see “Working with Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.</td>
</tr>
<tr>
<td>Regular Expression</td>
<td>Configures the signature to match specific information in the log. You can use a simple character string or a regular expression with exceptions. Reference: See “Regular expression libraries used by the server sensor” on page 33 and “Specifying Exceptions when Monitoring Log Files” on page 56 for more information.</td>
</tr>
<tr>
<td>Logs</td>
<td>Selects the logs the sensor should monitor for this signature.</td>
</tr>
</tbody>
</table>

Table 19: 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:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Specifies the information about the event that the sensor returns when an event matches the signature. Reference: For more information about using the Info window, see “Setting Up Information Fields for Windows Event Log User–Defined Signatures” on page 39.</td>
</tr>
<tr>
<td>Fusion Scripting</td>
<td>Displays Fusion scripts used by this signature. Reference: For more information about using Fusion Scripting, see Chapter 6, &quot;SecureLogic“ starting on page 95.</td>
</tr>
</tbody>
</table>

Table 20: 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 an event that the pre-defined 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 should be monitored.

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 52.

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, and then select the User Defined Events group.
   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.
   The new signature appears under the Log 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. 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 “Working with Responses” in the RealSecure Workgroup Manager User Guide or 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 55.
10. Do you want to generate a response only when certain text appears in a log?
    • 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 server sensor” on page 33 and “Specifying Exceptions when Monitoring Log Files” on page 56.
12. Do you want the server sensor to send certain attributes about the event to the console or to the database?
    • If yes, go to Step 13.
    • If no, go to Step 15.
13. In the left pane, select the signature you created (it appears under Log Rules), and then click **Info**.

The Info window appears.

14. Set the following values to retrieve certain information:

<table>
<thead>
<tr>
<th>To see this information when the event occurs...</th>
<th>Create an information field using this value...</th>
</tr>
</thead>
<tbody>
<tr>
<td>month</td>
<td>@ Field0</td>
</tr>
<tr>
<td>date</td>
<td>@ Field1</td>
</tr>
<tr>
<td>time</td>
<td>@ Field2</td>
</tr>
<tr>
<td>hostname</td>
<td>@ Field3</td>
</tr>
</tbody>
</table>

15. Click the **Save** icon.

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

**Reference:** For a procedure on applying a policy, see “Applying Policies in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.”
Chapter 4: User-Defined Signatures

Monitoring Local Syslog Events

Introduction

Before server sensor can monitor for local syslog events, you must configure the syslog.conf file. This topic describes how to configure the syslog.conf file.

Procedure

To edit the syslog.conf file:

1. Open the syslog configuration file, /etc/syslog.conf, using a text editor.
   
   **Example:** vi /etc/syslog.conf

   The syslog.conf file opens.

2. Add the following line:

   `*.info /path/messages_file`

   **Example:** `*.info /var/adm/messages`

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

3. To use the new syslog.conf file, do one of the following:

   - type the following commands to restart the syslog daemon, syslogd:
     
     `/etc/init.d/syslog stop`
     `/etc/init.d/syslog start`
   
   - type the following command to have the syslog daemon reread the syslog.conf file:
     
     `kill -HUP syslogd_process_id`
Monitoring the Wtmpx Binary Log File

Introduction

The server sensor can monitor the wtmpx binary log file on Solaris systems. This topic describes how to create a user-defined signature to monitor the wtmpx binary log and describes how to enable and disable the signature.

About the wtmpx log file

This version of RealSecure 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. RealSecure 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.

   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.
Chapter 4: User-Defined Signatures

10. Are you going to use Info pairs?
   - If yes, click Info and type the Name/Value pairs, and then click OK.
     Note: For more information about the Info window, see the RealSecure Help.
   - If no, go to Step 11.

11. Are you going to use Fusion scripts?
   - If yes, 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.
   - If no, go to Step 12.

12. From the File menu, click Save.

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 creating 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 ellipse \([\ldots]\) next to the Use logs box. The Logs window appears.
3. Click Add. The Add New Log window appears.
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.

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 19, “Field descriptions for custom syslog and generic log signatures” on page 49 and Table 20, “Button descriptions for custom syslog and generic log signatures” on page 49.
Specifying Exceptions when Monitoring Log Files

Introduction
An exception is a method of exempting certain data from being processed by the system. This method 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 two types of signatures support exceptions:

- Generic text log (ASCII)
- Windows Event Log

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

Reference: For more information about creating a generic text log signature, see “Monitoring Log Files” on page 46.

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

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

Prerequisites
Before creating an exception for a Windows event, you need to do the following:

- know the line in the event record that contains the information that will cause RealSecure 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 “Setting Up Information Fields for Windows Event Log User-Defined Signatures” on page 39.
- 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 server 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];
Specifying Exceptions when Monitoring Log Files

<table>
<thead>
<tr>
<th>Correlating new keys in the policy with string values</th>
</tr>
</thead>
<tbody>
<tr>
<td>The numbers 0 to 14 correspond to the index of string in event data. In other words, if we are looking for user name guest in event id 592, the user name is in string 3, so the matching expression would be:</td>
</tr>
<tr>
<td>IgnoreCase3 =B 1;</td>
</tr>
<tr>
<td>RegExp3 =S guest;</td>
</tr>
<tr>
<td>Except3 =B 0;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Creating an exception to a Windows NT Event Log signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> 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:</td>
</tr>
<tr>
<td>IgnoreCase0 =B 1;</td>
</tr>
<tr>
<td>RegExp0 =S administrator;</td>
</tr>
<tr>
<td>Except0 =B 1;</td>
</tr>
</tbody>
</table>
SECTION C: User–Defined Connection Signatures

Overview

Introduction
To monitor an event that pre–defined connection event signatures do not monitor for, create a user–defined connection event signature. This topic describes how to create user–defined connection event signatures.

Reference: For general information about connection events, see “Monitoring Suspect Connections” on page 28.

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.
   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.
   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 “Working with Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector 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 “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.
SECTION D: User–Defined Network Signatures

Overview

Introduction
Server sensor version 6.5 for Windows allows you to add user–defined network signatures to your system. This section describes how to edit server sensor files in order to add a user–defined network signature.

User–defined network signatures
User–defined network signatures allow you to define certain network events you want the server sensor to monitor for.

In this section
This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported User–Defined Network Signatures</td>
<td>62</td>
</tr>
<tr>
<td>Adding a User–Defined Network Signature</td>
<td>63</td>
</tr>
</tbody>
</table>
Chapter 4: User-Defined Signatures

Supported User-Defined Network Signatures

Introduction

This topic lists the categories of network signatures supported by server sensor version 6.5 for Windows platforms.

Supported user-defined network signatures

The following table lists the user-defined network signature categories supported by server sensor version 6.5 for Windows and their BlackICE issue numbers:

<table>
<thead>
<tr>
<th>Category</th>
<th>Issue Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-specified filename</td>
<td>2010000-2010999</td>
</tr>
<tr>
<td>User-specified URL</td>
<td>2011000-2011999</td>
</tr>
<tr>
<td>User-specified email recipient</td>
<td>2012000-2012999</td>
</tr>
<tr>
<td>User-specified email pattern</td>
<td>2013000-2013999</td>
</tr>
<tr>
<td>User-specified MIME-attached filename</td>
<td>2014000-2014999</td>
</tr>
<tr>
<td>User-specified TCP probe port</td>
<td>2015000-2015999</td>
</tr>
<tr>
<td>User-specified UDP probe port</td>
<td>2016000-2016999</td>
</tr>
<tr>
<td>User-specified registry key</td>
<td>2017000-2017999</td>
</tr>
<tr>
<td>User-specified TCP trojan response</td>
<td>2018000-2018999</td>
</tr>
<tr>
<td>User-specified IRC channel name</td>
<td>2019000-2019999</td>
</tr>
<tr>
<td>User-specified Java pattern</td>
<td>2020000-2020999</td>
</tr>
</tbody>
</table>

Table 21: User-defined signature categories and issue numbers

Reference

For more detailed information about category descriptions and configuration information, see Appendix C, "User-Defined Network Signature Category Descriptions and Configuration Information" starting on page 141.
Adding a User–Defined Network Signature

Introduction

Before server sensor version 6.5 for Windows platforms can monitor for a user–defined network event, you must add the user–defined network signature to the policy. To add a user–defined network signature to a policy, you must edit the policy. This topic describes how to edit the policy.

Caution: Do not add user–defined network signature information to the issuellist.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 default policy. You must derive a new policy or edit a customized policy to add a user–defined network signature.

```plaintext
\[Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\];
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;
\[Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\Response\];
\[Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\Response\BANNER\];
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 0;
\[Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\Response\RSKILL\];
Enabled =B 0;
\[Advanced\userdefinedsignatures\MicroAgent Rules\UDnetwork_template\Response\SUSPEND\];
```
To edit the policy to add the user-defined network signature:

1. Locate the policy you want to add the signature to.
   - For Workgroup Manager, locate the policy in the following directory:
     ISS/RealSecure 6.5 Console/Server Policies
   - For SiteProtector, locate the policy in the following directory:
     ISS/RealSecure 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 pre-defined policy. You must make an editable copy of a pre-defined policy and then customize the derived policy.

   **Reference:** For more information about making an editable copy of a pre-defined policy, see “Deriving New Policies from Existing Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

3. Click Open With ➔ Notepad.

4. Add the template information from “User-defined network signature template” on page 63 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.
   
   **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 BlackICE Issue ID.
   
   **Note:** In the template, the TID is -1. Make sure you replace the -1 with the appropriate BlackICE Issue ID.

8. Change the TValue to the appropriate configuration parameter information.
   
   **Note:** See Appendix C, "User-Defined Network Signature Category Descriptions and Configuration Information" for the following:

   ■ information about the categories of events you can monitor 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 65.

---

**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/:

```plaintext
[
Advanced]
userdefinedsignatures\MicroAgent Rules\userdefined_reg1\; 
Priority        =L      1; 
CheckDescription =S      userdefined network event for registry;
```
To configure the user–defined network event signature:

1. Open the policy that contains the user–defined network event 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 **FileÆSave**.
9. Apply the policy to the sensor or sensors that you want to use this signature.

**Reference:** For the procedure on applying a policy, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
SECTION E: User–Defined BSM Signatures

Overview

Introduction
This chapter describes how to create and configure user–defined server sensor signatures to monitor the Solaris Basic Security Module (BSM).

Prerequisite
Before using this section, you should have a basic understanding of the BSM. To learn how the BSM works, read the SunSHIELD Basic Security Module Guide. You can find this guide at the following location on the Sun Web site:


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

In this chapter
This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Solaris Basic Security Module Signatures</td>
<td>68</td>
</tr>
<tr>
<td>Task 1: Creating a User–Defined BSM Signature</td>
<td>69</td>
</tr>
<tr>
<td>Task 2: Generating the Event</td>
<td>71</td>
</tr>
<tr>
<td>Task 3: Using Praudit to Examine the Audit File</td>
<td>72</td>
</tr>
<tr>
<td>Task 4: Configuring the Server Sensor to Generate a Response</td>
<td>74</td>
</tr>
<tr>
<td>Task 5: Configuring the Information Fields Responses should Return</td>
<td>76</td>
</tr>
<tr>
<td>Task 6: Choosing Responses</td>
<td>77</td>
</tr>
<tr>
<td>Configuring BSM Audit Management</td>
<td>78</td>
</tr>
<tr>
<td>Examples of User–Defined BSM Information</td>
<td>79</td>
</tr>
</tbody>
</table>
Chapter 4: User-Defined Signatures

About Solaris Basic Security Module Signatures

Introduction

Solaris Basic Security Module (BSM) user-defined signatures allow you to take advantage of the auditing capability of the Solaris Basic Security Module. This topic outlines the tasks you must perform to configure a user-defined BSM signature.

Process overview

There are several steps involved in creating a BSM user-defined signature. The following table outlines this process. The remainder of this section describes this process in more detail.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a user-defined signature in the server sensor’s policy to record the event you want to monitor for.</td>
</tr>
<tr>
<td>2</td>
<td>Generate the event to create a record in the audit file.</td>
</tr>
</tbody>
</table>
| 3    | Use praudit to examine the record for the following:  
  • information you want to trigger the alert  
  • information fields in the audit record you want returned to you when the alert is triggered. |
| 4    | Configure the server sensor to trigger an alert whenever it detects the information you identified in Step 3. |
| 5    | Configure the server sensor to return the information fields you identified in Step 3 when an alert is triggered. |
| 6    | Choose how you want the server sensor to respond to the event. |

Reference: For examples that describe how to use BSM user-defined signatures, see “Examples of User-Defined BSM Information” on page 79.

What happens when I create a signature?

When you create a user-defined BSM signature, the server sensor configures the BSM to audit the events you specified in the signature. The server sensor uses its own audit flag to keep track of the events that are audited by the BSM for the server sensor.

What are 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. The rs flag is a custom flag created when you install the server sensor 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 update the server sensor’s policy, the server sensor associates these events with the rs flag in the audit_event file.

Example: The following text shows part of an audit_event file. RealSecure monitors the events that have and 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 BSM Signature

Introduction

To create a BSM user–defined signature you must first add the signature to the policy of the server sensor to record the events you want to monitor. You need to know the name of the events as recognized by the BSM.

In this task

This task explains 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
- how to set up the server sensor policy to monitor certain events by creating or modifying a BSM user–defined signature

Determining the name of an event

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

<table>
<thead>
<tr>
<th>If you...</th>
<th>Then...</th>
<th>And then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>can make a guess at the name</td>
<td>choose one or more events listed in the Event Log ID list in the console that appear to be the event you want to monitor</td>
<td>1. Generate the event (see “Task 2: Generating the Event” on page 71).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Look for the exact event that you want to audit in the log files (see “Task 3: Using Praudit to Examine the Audit File” on page 72).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Remove the other audit events you configured.</td>
</tr>
<tr>
<td>have no idea what the name of is</td>
<td>choose all the events listed in the Event Log ID list in the console, except for read and write calls (because they generate enormous audit logs)</td>
<td></td>
</tr>
</tbody>
</table>

Table 22: Options for choosing the correct event name

Kernel–level and user–level events and records

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:

<table>
<thead>
<tr>
<th>Auditing this event...</th>
<th>Monitors...</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUE_CREATE</td>
<td>the creation of new files</td>
</tr>
<tr>
<td>AUE_CHMOD</td>
<td>mode changes in files (for example, changes in read/write permissions)</td>
</tr>
<tr>
<td>AUE_at_create</td>
<td>at jobs</td>
</tr>
<tr>
<td>AUE_cron_invoke</td>
<td>cron jobs</td>
</tr>
</tbody>
</table>

Table 23: Examples of audit flags in the Solaris BSM
Chapter 4: User-Defined Signatures

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


Procedure

To create a user-defined BSM signature in the policy of the server sensor:

1. Open the policy you want to add this signature to.
2. Click the OS Events tab.
   All existing signatures appear in the Solaris BSM Rules group.
4. Click Add in the User-Defined BSM Rules pane.
   The Enter a name window appears.
5. Type a name for the user-defined BSM signature, and then click OK.
   The new signature appears under the Solaris BSM Rules group.
6. Select the signature that you just created in the left pane.
7. Click the … button, 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 page 69 for help.
8. Click OK, and then click the Save icon.
   A message tells you that you have modified the policy.
9. Click OK.
10. Click File ➔ Exit.
    The Policy Viewer closes.
11. Click Apply to Sensor.
Task 2: Generating the Event

Introduction
To test the user–defined BSM signature created in “Task 1: Creating a User–Defined BSM Signature” on page 69, you must generate an event to create a record in the audit file.

Procedure
To generate an event:

1. Generate the event so that you can examine the audit record it creates in the audit file.
2. Go to “Task 3: Using Praudit to Examine the Audit File” on page 72.

Example
If you want to monitor changes to a particular text file, you can generate the event by making a copy of the file and then saving the copy over the original file. Even though the file is the same, the change in the time and date stamp triggers an event that indicates that the file has changed.
Task 3: Using Praudit to Examine the Audit File

Introduction
Praedit is a command used to examine audit files. It turns the binary audit file into readable audit records. Before you use praedit, you should understand the components of audit records. This topic describes these components and also describes how to use praedit to examine your audit files.

Components of audit records
Each line of an audit record is called an audit token. Each token describes an attribute of the event recorded in the audit record. 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 more information about the kind of information tokens contain, see the “Audit Token Structure” section of the SunSHIELD Basic Security Module Guide, which is available online at the following Web site:


Procedure
To use praedit to examine the audit file:

1. Log on as root.
2. Type /etc/init.d/realsecure stop.
   The RealSecure server sensor stops.
3. Change to the /var/audit directory, and then type the following:
   cd /var/audit
   ls
   The system displays the contents of the /var/audit directory.
4. Identify the most recent audit file (under normal conditions it has not terminated in the name of the file), and then use the following command to view the last part of its contents:
   tail -f [audit file name] | praedit -s
5. Make a note of the following information from the audit record of the event you just created:
   - the name of the audit token
   - the exact data within the token that should trigger the server sensor and the name of the field that this data was in
   - the names of other fields that contain valuable information that you want to record whenever this event is found.

Note: If you do not know the name of the fields, look up the audit tokens in the SunSHIELD Basic Security Module Guide (http://docs.sun.com/db?q=SunSHIELD+Basic+Security+Module+Guide&p=). You will use this information in “Task 4: Configuring the Server Sensor to Generate a Response” on page 74 and in “Task 5: Configuring the Information Fields Responses should Return” on page 76.

6. Type /etc/init.d/realsecure start.
   The server sensor restarts.
7. Go to “Task 4: Configuring the Server Sensor to Generate a Response” on page 74.

Example: Audit record

The following audit record resulted from a person 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.cld.attacker.com
return,success,8
```

The kind of information you might want RealSecure 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 RealSecure 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 76.

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.
Chapter 4: User-Defined Signatures

Task 4: Configuring the Server Sensor to Generate a Response

Introduction

After you have examined the audit record, configure the server sensor to generate a response when it finds the information you identified.

Procedure

To configure the server sensor to generate a response:

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 → Solaris BSM Rules to open the list.
   All existing signatures appear in the Solaris BSM Rules group.
4. Click the signature that contains the events you set up in “Task 1: Creating a User-Defined BSM Signature” on page 69.
5. When you set up the signature, did you know the exact event name?
   ■ If no, go to Step 6.
   ■ If yes, go to Step 7.
6. Click the button, identify and clear the check boxes of the events you no longer want the server sensor to monitor for this signature, and then go to Step 7.
7. Click Match.
   The Match dialog box appears.
8. Add the information that you identified in Step 5 of “Task 3: Using Praudit to Examine the Audit File” on page 72.

To... Follow these steps...

Add information that the record should contain

1. In the Name box, select the token field that contains the information you want the record to contain.
   Note: If the token and field are not in the list, you can type them in using a Token.field format. Use the SunSHIELD documentation as a reference.
   Example: path
2. In the Value box, type the string of text that RealSecure should look for in that token field.
   Example: opt/webserver
   Caution: Wildcards and other metacharacters used in regular expressions do not work in the Value box. Create separate signatures if you want to use “or” logic in one or multiple tokens.
3. Click Add.
4. Repeat Steps 1 through 3 for each piece of information the record should contain.
   Note: 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.
### Task 4: Configuring the Server Sensor to Generate a Response

<table>
<thead>
<tr>
<th>To...</th>
<th>Follow these steps...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove information that you previously added</td>
<td>• Select the name of the information, and then click Remove.</td>
</tr>
</tbody>
</table>

When RealSecure finds a record that contains all the information you specify here, it generates a response to the event.

9. Click **OK**, and then click **Save**.
   
   Your changes are saved and you are prompted to apply the policy to the sensor before your changes can take affect.

10. Click **OK**, and then click **Close**.
    
    The system saves the policy.

11. Reapply the policy to the sensor.
    
    **Reference:** For more information on applying policies, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide*.
    
    The BSM is set to audit the flags you selected.

12. Go to “Task 5: Configuring the Information Fields Responses should Return” on page 76.
Task 5: Configuring the Information Fields Responses should Return

Introduction

After you specify the information that, if present, should generate a response, you must select the information in the record that you want RealSecure to log to the database and to return to you in a response.

Important

No matter what event you want to monitor, the identity of the source computer (the IP address or DNS name) is valuable information. In every audit record, this information is in the Machine ID field of the Subject token (Subject.Machine ID), so be sure this return value is 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 token field (token.field) you want RealSecure to return.
   Example: Subject.Audit User
3. In the Name box, give the token field a name.
   Example: Audit User ID
4. Repeat Steps 2 through 3 for each token field that contains useful information, and then click OK.
5. Go to “Task 6: Choosing Responses” on page 77.
Task 6: Choosing Responses

Introduction

After configuring the information fields you want included in RealSecure responses, you must choose which responses you want RealSecure to 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 “Working with Responses” in the RealSecure Workgroup Manager User Guide.

2. Click Save.

3. Close the Policy Editor, and then apply the policy you just modified to any Solaris server sensors that you monitor for the event.
Chapter 4: User-Defined Signatures

Configuring BSM Audit Management

Introduction

You can configure the server sensor to help manage the size of the BSM audit file in the following way:

- set a maximum BSM audit file size
- select a BSM audit reduction action for the server sensor to take if the file exceeds this file size

This topic explains how BSM audit reduction works and tells how to configure BSM audit management.

How BSM audit reduction works

When the BSM audit file grows larger than the maximum BSM audit file size setting, the server sensor invokes the BSM audit reduce policy. The policy takes action according to the setting in the BSM Audit Reduction field as follows:

- REMOVE unlinks the old audit file from the directory.
- REDUCE invokes the audit reduce program for each terminated audit file.
  
  Note: This action results in numerous 24-byte audit log files in the audit directory. Remove these files regularly.
- LEAVE does nothing to the audit file.
  
  Note: If you select LEAVE, you must manage the BSM audit files manually.

Procedure

To configure BSM audit management:

1. In the Managed Assets window, right-click the server sensor, and then select Properties.
   
   The Sensor Properties window appears.
2. On the Server Sensor tab, select the maximum file size in the Maximum BSM Audit Size field.
3. Select the action the server sensor should perform when the audit file reaches the specified size from the drop-down list in the BSM Audit Reduction field.
4. Click OK.
Examples of User–Defined BSM Information

Introduction

This topic discusses some of the information you can gather using user–defined BSM signatures. These examples include:

- logging source IP addresses and DNS names
- monitoring users
- monitoring the `su` command

Monitoring users

You can monitor a suspicious user’s activities 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 the `su` command

The `su` command enables a user to switch to another user account, which is root (the superuser), by default. An attacker can use `su` to gain the same privileges they would have if they logged in as root. Monitoring the use of `su` can help you watch for this type of attempted intrusion.

Example: If you suspected Bill, a user who has normal user ID privileges, of attempting to gain root access by using `su`, you could set up a user–defined signature to monitor his use of `su`. The AUE_su event logs the use of `su`. In an audit record, the `su` command shows up in the EventLog ID field of the Header token (Header.EventLog ID). The Return and the Text tokens let you know if the login succeeded or failed.

![Figure 2: Monitoring the su command](image-url)
**Audit record example:** If you set up a user–defined signature to detect when Bill uses su, records recording su failures and successes, like the following audit record that indicates a success, would cause the sensor to generate a response:

```
header,86,2,AUE_su,Thu Aug 26 16:53:54 1999, + 16246296 msec
subject,bill,root,nogroup,bill,nogroup,1551,1538,24 3 ip-100-195.cld.hacker.com
text,success for user root
return,success,0
```
Overview

Introduction

A firecell signature blocks or responds in other ways to packets that match a certain criteria. This section describes firecell signatures and explains how to create and use them.

In this chapter

This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Firecell Signatures</td>
<td>82</td>
</tr>
<tr>
<td>Understanding Firecell Signatures</td>
<td>83</td>
</tr>
<tr>
<td>Creating and Configuring Firecell Signatures</td>
<td>85</td>
</tr>
<tr>
<td>Disabling Firecell Signatures</td>
<td>89</td>
</tr>
<tr>
<td>Using Firecell Signatures</td>
<td>90</td>
</tr>
</tbody>
</table>
### About Firecell Signatures

**Introduction**

Firecell signatures are based on protocol and packet detection. They work like a firewall to ensure that only authorized clients can access the server.

**Firecell signatures**

Firecell signatures do the following:

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

**When to use firecell signatures**

Use firecell signatures to block certain packets all the time even if the packets match a default or other user-defined signature.

**Firecell signatures in server sensor for Windows**

In server sensor 6.5 for Windows, firecell signatures set by dynamic blocking stop functioning only after the dynamic block period expires.

**Note:** For more information about blocking, see the *RealSecure Workgroup Manager User Guide Version 6.7*.

**Limitation**

Server sensor version 6.5 does not support outbound firecell signatures. To use outbound firecell signatures, you must use Workgroup Manager version 6.0 to manage a server sensor version 6.0.
Understanding Firecell Signatures

Introduction

To use firecell signatures effectively, you must understand the parameters that define firecell signatures, the responses available to firecell signatures, and the relevance of the order of firecell signatures. This topic explains these concepts.

Parameters used to define firecell signatures

Use the following parameters to define firecell signatures:

- protocol type (IP, TCP, UDP, or ICMP)
- specific IP address or class of addresses
- port number (for TCP and UDP firecell signatures)

**Note:** When you define a port number, the following rules apply:

- you can monitor all ports by entering a 0 (zero) in the Port box
- you cannot specify a range of ports other than all ports
- you cannot enter more than one number in the Port box (0 for all ports, or the specific port number)

Allowing reply traffic

Server sensor version 6.5 for Windows platforms can differentiate inbound TCP connection requests from inbound TCP traffic. When you enable this feature, the sensor does the following:

- blocks incoming TCP connection requests but allows TCP traffic through the firewall if the connection was initiated from the server sensor

**Note:** In some situations it may appear that a connection was initiated from the computer on which the server 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 server sensor is installed. For example, because active mode FTP connections appear to originate from the computer where server sensor is installed but in fact are initiated from a remote computer, the sensor blocks active mode FTP connections.

- prevents firecell events from being triggered for any incoming reply packets on port 1024 or above

See Knowledge Base article 2144 for more information about how to enable this feature for Windows sensors.

Responses available to firecell signatures

You can configure the server sensor to generate any pre-defined or user-defined response when the sensor detects a packet that matches the criteria of the firecell signature. You can also configure the server sensor to allow the packet to pass through the network stack if you want to record and respond to the packet but do not want to stop it.

**Caution:** Be careful when assigning 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.

Order of firecell signatures

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.
Chapter 5: Firecell Signatures

**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 the order of firecell signatures**

You can rearrange the order of firecell signatures in the policy editor.

**Reference:** For more information about rearranging the order of firecell signatures, see “Firecell signature precedence” on page 87.
Creating and Configuring Firecell Signatures

**Introduction**
This topic describes how you create and configure firecell signatures.

**Prerequisite**
Before using firecell signatures that disable traffic, make sure you have signatures that specifically enable traffic that the server sensor needs to run properly.

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

**Reference:** For more information, see “Order of firecell signatures” on page 83 and “Firecell signature precedence” on page 87.

---

### 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.
   
   **Example:** If you want to block inbound UDP packets coming from a certain address and certain port number, select **UDP Inbound**.
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**.
   
   The new signature appears under the type of firecell signature you selected in Step 4.

---

### Configuring a firecell signature

To configure a firecell signature:

1. Open the policy that contains the signature you want to configure.
2. Select the **Protect** tab, open the **Protect** folder, and then open the **Firecell** folder.
3. In the left pane, select the signature you want to configure.
   
   The properties of the signature appear in the right pane.
4. Set the priority of the event in the **Priority** box.
5. Is this signature monitoring for TCP or UDP traffic?
   
   - If yes, go to Step 6.
   - If no, go to Step 7.
6. 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.
7. Click Add, and then specify the IP address(es) of the packets that you want to allow or not allow as described in the following table:

<table>
<thead>
<tr>
<th>If you want to specify...</th>
<th>Then...</th>
</tr>
</thead>
</table>
| one IP address           | 1. Select IP Address.  
2. Type the IP address in the Address box.  
3. Click OK to save your changes. |
| a range of IP addresses in a network class | 1. Select IP Address.  
2. Type an address in the Address box that is in the range you want to specify.  
3. Type a number in the Mask box to represent the level of network class.  
   Reference: For more information about using masks and other items in this window, click Help.  
4. Click OK to save your changes. |
| all IP addresses         | • Select Any Address, and then click OK. |
| a network asset          | Prerequisite: You must create an asset through ViewNetwork Assets in Workgroup Manager before you can specify an asset here.  
• Select Network Asset, choose an asset from the list on the right, and then click OK. |

8. In the Actions section, specify how you want the sensor to treat the packets that match the specified addresses.

9. Select the responses that you want the sensor to take when a match to this signature occurs.

   Reference: For more information about each response, see “Working with Responses” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

10. Click Save.

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

   Reference: For more information about applying policies, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.

Parameters used to block or allow packets

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.
Creating and Configuring Firecell Signatures

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

<table>
<thead>
<tr>
<th>Selected option</th>
<th>If the sensor detects a packet with a source IP of...</th>
<th>the sensor... and...</th>
<th>If the sensor detects a packet with a source IP of...</th>
<th>the sensor... and...</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the range of listed IP addresses, drop the packet and generate selected responses</td>
<td>196.131.16.2</td>
<td>drops the packet</td>
<td>generates responses.</td>
<td>208.21.28.3</td>
</tr>
<tr>
<td>In the range of listed IP addresses, accept the packet, but continue to generate responses</td>
<td>196.131.16.2</td>
<td>accepts the packet</td>
<td>generates responses.</td>
<td>204.20.3.41</td>
</tr>
<tr>
<td>Not in the range of listed IP addresses, drop the packet and generate selected responses</td>
<td>208.21.16.3</td>
<td>drops the packet</td>
<td>generates responses.</td>
<td>196.131.16.2</td>
</tr>
<tr>
<td>Not in the range of listed IP addresses, accept the packet, but continue to generate selected responses</td>
<td>104.20.3.41</td>
<td>accepts the packet</td>
<td>generates responses.</td>
<td>196.131.16.2</td>
</tr>
</tbody>
</table>

Table 24: Example of action options for firecell signatures

Specifying ports

When you specify a port number, the server sensor blocks or responds to packets on that port number only. The sensor ignores packets on other ports regardless of the packet IP addresses.

Firecell signature precedence

The first signature listed in a category, such as TCP Inbound, takes precedence over the next signature listed in a category. If the signatures are not in the order of precedence you want, you can rearrange them.

Rearranging firecell signatures

To rearrange firecell signatures:

1. Open the server sensor 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 type of 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, as shown in Figure 3 on page 88.
### Chapter 5: Firecell Signatures

#### Figure 3: Rearranging firecell signatures

<table>
<thead>
<tr>
<th>Enabled</th>
<th>Event</th>
<th>Priority</th>
<th>Response</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Block HTTP_Incom</td>
<td>High</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>Disallow all TCP Communication</td>
<td>High</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Allow port 80</td>
<td>High</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>Allow port 25</td>
<td>High</td>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>
Disabling Firecell Signatures

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

Procedure
To disable a firecell signature:

1. From the Protect tab in the policy editor, clear the check box for 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 “Applying Policies” in the RealSecure Workgroup Manager User Guide or the SiteProtector Help.
Using 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Addresses</td>
<td>172.25.50.0/24</td>
</tr>
<tr>
<td>Actions</td>
<td>#3 Not in the range of listed IP addresses, drop the packet and generate the selected responses</td>
</tr>
</tbody>
</table>

**Table 25: 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Addresses</td>
<td>172.25.50.0/24</td>
</tr>
<tr>
<td>Actions</td>
<td>#4 Not in the range of listed IP addresses, accept the packet but generate the selected responses</td>
</tr>
</tbody>
</table>

**Table 26: 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 27 on page 91.

**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 server sensor.
**Reference:** For more information about using multiple signatures, refer to “Order of firecell signatures” on page 83.

<table>
<thead>
<tr>
<th>Signature</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IP Addresses</td>
<td>Any address (0.0.0.0/0)</td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Actions</td>
<td>#3 Not in the range of listed IP addresses, drop the packet and generate the selected responses</td>
</tr>
<tr>
<td>2</td>
<td>IP Addresses</td>
<td>Any address (0.0.0.0/0)</td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Actions</td>
<td>#3 Not in the range of listed IP addresses, drop the packet and generate the selected responses</td>
</tr>
<tr>
<td>3</td>
<td>IP Addresses</td>
<td>Any address (0.0.0.0/0)</td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>0 (any port)</td>
</tr>
<tr>
<td></td>
<td>Actions</td>
<td>#1 In the range of listed IP addresses, drop the packet and generate the selected responses</td>
</tr>
</tbody>
</table>

*Table 27: Firecell example—blocking all TCP traffic except for typical Internet traffic*
Fusion Scripting
Chapter 6

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.

Note: For management consoles later than version 6.6 for Workgroup Manager, and later than version 2.0, Service Pack 1 for SiteProtector, the server sensor feature previously known as “SecureLogic” has been renamed “Fusion Scripting,” and the term “SecureLogic scripts” has been renamed “Fusion scripts.”

In this chapter

This chapter contains the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section A, &quot;Introduction to Fusion Scripting&quot;</td>
<td>97</td>
</tr>
<tr>
<td>Section B, &quot;Working with Fusion Scripting&quot;</td>
<td>105</td>
</tr>
</tbody>
</table>
SECTION A: Introduction to Fusion Scripting

Overview

Introduction
This section describes Fusion Scripting and tells how you might use it to enhance your system’s ability to monitor events.

In this section:
This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Fusion Scripting</td>
<td>98</td>
</tr>
<tr>
<td>Data Available to Fusion Scripting</td>
<td>100</td>
</tr>
<tr>
<td>Tcl Script Categories Used in Fusion Scripting</td>
<td>102</td>
</tr>
</tbody>
</table>
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 matched a certain pattern, the sensor took action depending on the responses set for that signature. Sometimes, however, 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 server sensor collects. This ability adds an extra layer of validation, reduces false positives, and prevents unwanted events from flooding the console.

Note: Although certain predefined signatures contain Fusion scripts, you can only associate Fusion scripts with user-defined signatures.

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 management 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:

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>When an event matches a signature that the sensor monitors for, the sensor saves the data from information fields to transient (temporary) memory. Fusion Scripting can then use the saved data associated with the signature.</td>
</tr>
<tr>
<td>3</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 4     | Fusion scripts then examine the data from the information fields and other system-specific data to validate the importance of the event.  
  - 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.  
  - 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. |
How to add Fusion Scripting to signatures

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 pre–defined signatures contain Fusion Scripting, however, while you can modify Fusion Scripting response scripts associated with pre–defined signatures, you cannot add new scripts to pre–defined signatures.

  **Reference:** For specific information about adding and modifying Fusion scripts, see “Adding or Modifying a Fusion Script” on page 108.

- 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 109.

Prerequisites

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

<table>
<thead>
<tr>
<th>Concept</th>
<th>References</th>
</tr>
</thead>
</table>
|                                                      | Practical Programming in Tcl and Tk, by Brent B. Welch
|                                                      | ISBN: 0130220280                                                          |
|                                                      | Tcl tutorial:                                                              |
|                                                      | [http://www.msen.com/~clif/TclTutor.html](http://www.msen.com/~clif/TclTutor.html) |
| Variables, info fields, and other data used by       | See “Data Available to Fusion Scripting” on page 100.                     |
| Fusion Scripting                                     |                                                                           |
| Tcl script categories used in Fusion Scripting        | See “Tcl Script Categories Used in Fusion Scripting” on page 102.         |

**Table 28:** Prerequisites to working with Fusion Scripting

**Caution:** Fusion Scripting is an advanced feature. The Tcl scripting language can be powerful, and if used improperly, it 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:

<table>
<thead>
<tr>
<th>Validation Script Result</th>
<th>Transient Variable Removed from Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>after the responses associated with the signature complete</td>
</tr>
<tr>
<td>False</td>
<td>immediately</td>
</tr>
</tbody>
</table>

Table 29: 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
Data Available to Fusion Scripting

- 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 114.

**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 115.

**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:

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

**Table 30:** Read-only event data values available to Fusion scripts

**Reference:** For more information about the GetData extension, see “Predefined Tcl Extensions” on page 106.

**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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>__iss_priority</td>
<td>The priority of the event.</td>
</tr>
</tbody>
</table>

**Table 31:** Read/write event data values available to Fusion scripts

**Reference:** For more information about the GetData extension, see “Predefined Tcl Extensions” on page 106.
Tcl Script Categories Used in Fusion Scripting

Introduction

The server 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 runs each type of script:

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

Table 32: 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 server sensor starts or loads a new policy (which causes the TCL process 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 begin to run at system startup, any signature can benefit from initialization script actions.

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 opens the file later 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:

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

Table 33: 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 and 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. 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 server 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 server 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 server sensor 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 options for working with Fusion Scripting.

In this section:
This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined Tcl Extensions</td>
<td>106</td>
</tr>
<tr>
<td>Adding or Modifying a Fusion Script</td>
<td>108</td>
</tr>
<tr>
<td>Configuring a Fusion Scripting Response</td>
<td>109</td>
</tr>
<tr>
<td>Configuring a Fusion Scripting SNMPv3 Response</td>
<td>110</td>
</tr>
<tr>
<td>Returning a True or False Result in a Validation Script</td>
<td>113</td>
</tr>
<tr>
<td>Using Fusion Scripts</td>
<td>114</td>
</tr>
<tr>
<td>Disabling Fusion Scripting</td>
<td>119</td>
</tr>
</tbody>
</table>
Predefined Tcl Extensions

Introduction

This topic describes the pre-defined Tcl extensions you can use with Fusion scripts.

Predefined Tcl extensions

RealSecure provides the following set of Tcl extensions:

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

Table 34: Predefined Tcl extensions
SaveArray Stores a Tcl array of values in persistent memory.  

Example: To create an array containing checksums of a set of files (using a chksum procedure that calculates the sum), use a script similar to the following:

```
# 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];
# Calculate 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;
```

RestoreArray Retrieves a Tcl array saved previously with the SaveArray command.  

Example: To retrieve the value of an array called checksum (previously saved using SaveArray) and compare its value to another variable, use the following command:

```
RestoreArray checksum
if { $checksum ([string tolower $fname] ) == $sum}
    {return 0;}
else
    {return 1;}
```

RemoveArray Deletes an entire array from persistent memory.  

Example: To delete an array called checksum, use the following command:

```
Removearray checksum
```

GetTid Returns a unique integer. You can use GetTid to distinguish multiple instances of the same event.  

Example: To create a unique name for a variable, use the following command:

```
set myUniqueName "myVar"
append myUniqueName [ GetTid ]
```

Table 34: 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**
You must create a signature before you add or modify Fusion Scripting. For more information about creating user-defined signatures, see Chapter 4, "User-Defined Signatures" starting on page 31.

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

1. Open the server sensor 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 102.
5. Click **OK**.
   The system saves the scripts.
6. Apply the modified policy to the sensor.
   **Reference:** For more information about applying policies, see “Applying Policies” in the *RealSecure Workgroup Manager User Guide* or the SiteProtector Help.
Configuring a Fusion Scripting Response

Introduction
A Fusion Scripting response allows you to use Tcl scripts to respond to events. With Fusion Scripting responses you can use and process data, such as 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configure the Fusion Scripting SNMPv3 response in the management console.</td>
</tr>
<tr>
<td>2</td>
<td>Install and configure a trap receiver that supports SNMPv3.</td>
</tr>
<tr>
<td>3</td>
<td>Enable the SNMPv3 response.</td>
</tr>
</tbody>
</table>

Task1: Configuring the SNMPv3 Fusion Scripting response

To configure the SNMPv3 Fusion Scripting response:

1. In the Sensor Responses window, select the Default response policy, and then click Derive New.
   
   The Choose Name window appears.
2. Type a name for the new response policy, and then click OK.
   
   The new policy appears on the Responses window.
3. Select the new response policy, and then click Customize.
4. Expand the Fusion Scripting response, and then select SNMPv3.
5. Modify the lines in the script according to the following table:

<table>
<thead>
<tr>
<th>Script line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># set trap receiver IP address or host name</td>
<td>Sets the IP address or host name of the trap receiver.</td>
</tr>
<tr>
<td>set host &quot;mytrapconsole.mydomain.com&quot;</td>
<td>Example: If the trap receiver is running on a host with an IP address of 172.40.50.30, change &quot;mytrapconsole.mydomain.com&quot; to &quot;172.40.50.30&quot;.</td>
</tr>
<tr>
<td># set the port number trap receiver is listening on</td>
<td>Sets the port number the trap receiver is listening on. The standard SNMP port is 162.</td>
</tr>
<tr>
<td>set port &quot;162&quot;</td>
<td>Example: If your trap receiver is listening on port 888, change &quot;162&quot; to &quot;888&quot;.</td>
</tr>
<tr>
<td># set the protocol to use when sending traps</td>
<td>Sets the protocol to use when sending traps. The standard SNMP protocol is UDP.</td>
</tr>
<tr>
<td>set protocol &quot;UDP&quot;</td>
<td>Example: If your trap receiver is using TCP, change &quot;UDP&quot; to &quot;TCP&quot;.</td>
</tr>
</tbody>
</table>
Caution: When using the SNMPv3 response, SNMPv3 passwords will be visible in text files on both the management console and the server sensor hosts.

6. Click OK.
   The Sensor Responses window closes.

7. Click Apply to Sensor.
   The system applies the response policy to the sensor.

8. Click OK.
   The Server Sensor Responses window closes.

<table>
<thead>
<tr>
<th>Script line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code># set SNMPv3 user name</code>&lt;br&gt;set userName &quot;mySNMPuser&quot;</td>
<td>Sets the user name for the SNMPv3 user.&lt;br&gt;&lt;br&gt;Note: 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.&lt;br&gt;&lt;br&gt;Example: If the user name you created at the trap receiver is &quot;testuser&quot;, change &quot;mySNMPuser&quot; to &quot;testuser&quot;.</td>
</tr>
<tr>
<td><code># set SNMPv3 authentication password - must be at least 8 characters long</code>&lt;br&gt;set authPass &quot;mySNMPpass&quot;</td>
<td>Sets the SNMPv3 authentication password. This password must be at least eight characters long.&lt;br&gt;&lt;br&gt;Example: If the authentication password you have set for your user is &quot;secure87Pass&quot;, change &quot;mySNMPpass&quot; to &quot;secure87Pass&quot;.&lt;br&gt;&lt;br&gt;Note: You can use different authentication and privacy passwords, but this is not required.</td>
</tr>
<tr>
<td><code># set SNMPv3 privacy password - must be at least 8 characters long</code>&lt;br&gt;set privPass &quot;mySNMPpass&quot;</td>
<td>Sets the SNMPv3 privacy password. This password must be at least eight characters long.&lt;br&gt;&lt;br&gt;Example: If the privacy password you have set for your user is &quot;secure87Pass&quot;, change &quot;mySNMPpass&quot; to &quot;secure87Pass&quot;.&lt;br&gt;&lt;br&gt;Note: You can use different authentication and privacy passwords, but this is not required.</td>
</tr>
<tr>
<td><code># set timeout in seconds</code>&lt;br&gt;set timeout &quot;10&quot;</td>
<td>Sets the number of seconds the console should wait for a &quot;message received&quot; confirmation from the trap receiver.&lt;br&gt;&lt;br&gt;Only change the default if the trap receiver you are using is slow to respond.&lt;br&gt;&lt;br&gt;Example: If you want to set the timeout period to 15 seconds, change &quot;10&quot; to &quot;15&quot;.</td>
</tr>
<tr>
<td><code># set retry count</code>&lt;br&gt;set retries &quot;1&quot;</td>
<td>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.&lt;br&gt;&lt;br&gt;Example: To resend the trap 5 times, change &quot;1&quot; to &quot;5&quot;.</td>
</tr>
</tbody>
</table>
Chapter 6: Fusion Scripting

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.

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 sensor(s) that you want to use this policy.

   **Reference:** For more information about applying policies, see “Applying Policies” in the RealSecure Workgroup Manager User Guide or 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:

<table>
<thead>
<tr>
<th>Result</th>
<th>Value to return</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1</td>
</tr>
<tr>
<td>False</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 35: 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.

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


## Using Fusion Scripts

### Introduction

You can use Fusion scripts in numerous ways. This topic describes several ways for you to 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 RealSecure 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 server sensor policy that contains the user-defined signature you want to modify.
Using Fusion Scripts

1. Select the signature from the appropriate folder.
   The properties of the signature appear in the right pane.

2. Click Fusion Scripting.
   A message box reminds you to use caution when using Fusion Scripting.

3. Click OK.
   The Fusion Scripting window appears.

4. Type or paste the scripts you want to use to calculate or create the baseline data in the Initial Script box.

5. Click OK.
   The system saves the scripts.

6. Click OK.
   The Fusion Scripting window appears.

7. Apply the modified policy to the sensor.

Reference: For more information about applying policies, see “Applying Policies” in the RealSecure Workgroup Manager User Guide.

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 server 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 server sensor changes the priority of an event, it sends an additional info field to indicate that the priority was changed. This info 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 117.

Correlating information between two 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 looks 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
Chapter 6: Fusion Scripting

```tcl
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**

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 RealSecure installation directory. This file contains a log that details how the Tcl subprocess interacts with the server sensor.

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

```tcl
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 server sensor would monitor for this type of attack:

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When the server 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.</td>
</tr>
<tr>
<td>2</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>The server 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.</td>
</tr>
</tbody>
</table>

**Table 36: Monitoring for an attack on a Web server**
### Example Initialization script:

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

### Example Validation script:

```bash
set fSrc [file size "c:/hidndir/backup.htm"]
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:

```bash
# 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.

```tcl
# 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.

Note: Disabling Fusion scripts disables the network monitoring component of server 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. The 5.5 version of the server sensor does not contain signatures based purely on Fusion Scripting, however, X–Press Updates may contain them. Review the Signatures Help or the ReadMe file in X–Press Updates for a list of signatures based on Fusion Scripting.

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 server sensor 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 to save the change.
6. Apply the modified policy to a sensor.
   Fusion scripts associated with the signature is disabled.

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 totally based on Fusion scripts
Disabling Fusion Scripting

To disable 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.
Troubleshooting
Chapter 7

Troubleshooting

Overview

Introduction
This chapter describes problems you may encounter while using the server sensor and it 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 xi for information about contacting Technical Support.

In this chapter
This chapter contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolating Policy Problems</td>
<td>124</td>
</tr>
<tr>
<td>Tcl Script Problems</td>
<td>131</td>
</tr>
<tr>
<td>No Communication Between the Sensor and the Console</td>
<td>132</td>
</tr>
</tbody>
</table>
Chapter 7: Troubleshooting

Isolating Policy Problems

Introduction

This topic provides information about isolating and correcting possible policy problems.

Error messages indicating policy problems

Error messages that might indicate policy file problems appear in the Managed Assets window. These messages include the following:

- “No such file or directory” displays in the Control Status column when connecting to a sensor
- “Failure to transfer current policy when the control channel opened” displays in the Control Status column
- “Failure to read/transfer common.policy” displays in the Control Status column
- “Unknown” displays in the Policy column

Troubleshooting techniques

When an error message indicates a possible policy file problem, you can try the following techniques to isolate and correct the problem:

- Inspect the issDaemon directory and sensor component directory. Look for things such as file sizes of zero bytes, or no policy files listed.
- Check the event log (Windows) or the syslog (Unix) of the sensor for any unusual events.
- Reapply a policy to the sensor. For more information about applying policies to sensors, see the “Working with Policies” chapter of the RealSecure Workgroup Manager User Guide.
- Revert the sensor to a default policy configuration by deleting the current.policy file, and then applying a new policy to the sensor. For more information, see “Deleting current.policy” on page 126.
- Determine whether the problem is the daemon. Start the daemon without automatically restarting the sensor to determine if the daemon loads correctly. For more information, see “Determining whether the problem may be the daemon” on page 127.
- Start the sensor manually to detect sensor component problems. For more information, see “Starting the sensor manually to detect sensor component problems” on page 129.
- If the problem persists, contact ISS Technical Support. For technical support contact information, see “Getting Technical Support” on page xi.

Important: To maintain the sensor’s intended configuration, change any newly applied policy to match the previous configuration.
Stopping the issDaemon

Some troubleshooting techniques for policy problems require stopping 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:

<table>
<thead>
<tr>
<th>For this operating system...</th>
<th>To use this method...</th>
<th>Do this...</th>
</tr>
</thead>
</table>
| Windows NT                   | Desktop               | To stop the issDaemon from the desktop:  
|                              |                       | 1. Select Start ➔ Settings ➔ Control Panel.  
|                              |                       |   The Control Panel appears.  
|                              |                       | 2. Double-click Services.  
|                              |                       |   The Services window appears.  
|                              |                       | 3. Double-click issDaemon.  
|                              |                       |   The issDaemon Properties window appears.  
|                              |                       | 4. In the Service Status section, click Stop.  
|                              |                       | 5. Click OK.  
|                              |                       |   The issDaemon stops.  
| Windows NT                   | Command line          | To stop the issDaemon from the command line, type:  
|                              |                       |   C:\>net stop issdaemon  
|                              |                       |   The issDaemon stops.  
| Windows 2000                 | Desktop               | To stop the issDaemon from the desktop:  
|                              |                       | 1. Select Start ➔ Settings ➔ Control Panel.  
|                              |                       |   The Control Panel appears.  
|                              |                       | 2. Double-click Administrative Tools.  
|                              |                       |   The Administrative Tools window appears.  
|                              |                       |   The Services window appears.  
|                              |                       | 4. Double-click issDaemon.  
|                              |                       |   The issDaemon Properties window appears.  
|                              |                       | 5. In the Service Status section, click Stop.  
|                              |                       |   The issDaemon stops.  
| Windows 2000                 | Command line          | To stop the issDaemon from the command line, type:  
|                              |                       |   C:\>net stop issdaemon  
|                              |                       |   The issDaemon stops.  
| Unix                         | Command line          | Log in as root.  
|                              |                       | From the command line, type:  
|                              |                       |   /etc/ini.d/realsecure stop  
|                              |                       |   The issDaemon stops.  

Note: This procedure assumes that the product is installed on the default directory and is the first or only network sensor installed on the machine.
Chapter 7: Troubleshooting

Starting the issDaemon

Some troubleshooting techniques for policy problems require starting 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:

<table>
<thead>
<tr>
<th>For this operating system...</th>
<th>To use this method...</th>
<th>Do this...</th>
</tr>
</thead>
</table>
| Windows NT Desktop          | Desktop               | To start the issDaemon from the desktop:  
2. Double-click **Services**.  
The Services window appears.  
3. Double-click **issDaemon**.  
The issDaemon Properties window appears.  
4. In the Service Status section, click **Start**.  
5. Click **OK**.  
The issDaemon starts. |
| Windows NT Command line     | Command line          | To start the issDaemon from the command line, type:  
C:\>net start issdaemon  
The issDaemon starts. |
| Windows 2000 Desktop       | Desktop               | To start the issDaemon from the desktop:  
2. Double-click **Administrative Tools**.  
The Administrative Tools window appears.  
3. Double-click **Services**.  
The Services window appears.  
5. Double-click **issDaemon**.  
The issDaemon Properties window appears.  
6. In the Service Status section, click **Start**.  
7. Click **OK**.  
The issDaemon starts. |
| Windows 2000 Command line  | Command line          | To start the issDaemon from the command line, type:  
C:\>net start issdaemon  
The issDaemon starts. |
| Unix                        | Command line          | 1. Log in as **root**.  
2. At the command line, type:  
/etc/init.d/realsecure start  
The issDaemon starts. |

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. In the Managed Assets window, select the sensor.
2. Select **Sensor ➔ Stop Managing Sensor.**
   A confirmation message appears.
3. Click **Yes** to confirm.
4. Stop the issDaemon.
   **Reference:** For the procedure on stopping the issDaemon, see “Stopping the issDaemon” on page 125.
5. Locate and then delete `current.policy`. The `current.policy` file is located in the directory where the sensor was installed.
6. Restart the issDaemon.
   **Reference:** For the procedure on starting the issDaemon, see “Starting the issDaemon” on page 126.
7. Using Windows Explorer, locate the transfer directory at the following location:
   `C:\Program Files\ISS\RealSecure 6.5 Console\Transfer`
8. Delete the files that contain the sensor’s IP address in the filename.
9. Did the sensor restart automatically?
   If *yes*, go to Step 10.
   If *no*, see “Determining whether the problem may be the daemon” on page 127.
10. In the Managed Assets window, select **Asset ➔ Manage.**
11. Select the sensor, and then click **OK.**
12. Apply a default, customized, or imported policy to the sensor.
   **Reference:** For more information about applying policies to sensors, see the “Working with Policies” chapter of the *RealSecure Workgroup Manager User Guide.*

**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. ISS recommends troubleshooting daemon problems with the help of Technical Support.

To determine if the problem was caused by the daemon:

1. Stop the issDaemon.
   **Reference:** For the procedure on stopping the issDaemon, see “Stopping the issDaemon” on page 125.
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.
4. Change the auto_recovery value for the selected sensor from 1 to 0.
   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: For the procedure on starting the issDaemon, see “Starting the
   issDaemon” on page 126.
7. Check the running processes to determine whether the daemon was loaded correctly.
   To check the running processes, use the following decision table to determine your
   next action:

<table>
<thead>
<tr>
<th>For this operating system...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT and 2000</td>
<td>1. Press CNTRL+ALT+DELETE.</td>
</tr>
<tr>
<td></td>
<td>The Windows NT Security window appears.</td>
</tr>
<tr>
<td></td>
<td>2. Click Task Manager.</td>
</tr>
<tr>
<td></td>
<td>The Windows NT Task Manager window appears.</td>
</tr>
<tr>
<td></td>
<td>3. Select the Processes tab, and ensure that the</td>
</tr>
<tr>
<td></td>
<td>issDaemon.exe process is running.</td>
</tr>
<tr>
<td>Unix</td>
<td>At the command line, type:</td>
</tr>
<tr>
<td></td>
<td>han[admin]# ps -ax</td>
</tr>
<tr>
<td>Solaris</td>
<td>At the command line, type:</td>
</tr>
<tr>
<td></td>
<td>han[admin]# ps -ef</td>
</tr>
</tbody>
</table>

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 129.
   - If no, then the policy problem may have been caused by the daemon. Contact ISS
     Technical Support for further assistance. See “Getting Technical Support” on
     page xi for contact information.
9. Stop the issDaemon.
   Reference: For the procedure on stopping the issDaemon, see “Stopping the
   issDaemon” on page 125.
10. Locate the issDaemon.policy file. The default location for this file is:
    C:\Program Files\ISS\IssDaemon
11. Open the issDaemon.policy file in a text editor such as Notepad.
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: For the procedure on starting the issDaemon, see “Starting the
   issDaemon” on page 126.
15. Did the sensor start automatically after starting the daemon?
   - If yes, determine whether the policy problem has been corrected. If the problem persists, see “If you need further help” on page 130.
   - 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 129.

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 starting the sensor manually. If you try to start the sensor manually without first shutting down the daemon, the following error message displays:

Error creating FirstInstance Mutex

To start the sensor manually and determine whether sensor component problems may exist:

1. Start the sensor, using the following decision table to determine your next action:

<table>
<thead>
<tr>
<th>To start the sensor from...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows command line</td>
<td>At the command line, type: issCSF “C:\Program Files\ISS\issSensors\network_sensor_1” network_sensor_1 NetworkEngine RealSecure NetworkEngine</td>
</tr>
<tr>
<td>Unix</td>
<td>At the command line, type: . /issCSF -d /opt/ISS/issSensors/ network_sensor_1/ -c /opt/ISS/issDaemon/ crypt.policy -n network_sensor_1 -e NetworkEngine -h NetworkEngine -g RealSecure</td>
</tr>
</tbody>
</table>

2. Did the sensor start correctly?
   - If yes, then you must set the auto-recovery flag back to its original value. Go to Step 3.
   - If no, then contact ISS Technical Support for further assistance. See “Getting Technical Support” on page xi for contact information.

3. Stop the issDaemon.
   Reference: For the procedure on stopping the issDaemon, see “Stopping the issDaemon” on page 125.

4. Locate the issDaemon.policy file.
   The Windows default location for this file is:
   C:\Program Files\ISS\IssDaemon
   The Solaris default location for this file is:
   /opt/ISS/issDaemon
Chapter 7: Troubleshooting

5. Open the issDaemon.policy file in a text editor such as Notepad.
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:** For the procedure on starting the issDaemon, see “Starting the issDaemon” on page 126.

If you need further help

If you still cannot isolate the problem, send copies of the sensor’s current.policy and common.policy files to ISS Technical Support for further assistance. See “Getting Technical Support” on page xi for contact information. Include the output from the following commands:

**Solaris**

tail /var/adm/messages

netstat -a

ps -ef

**Linux**

tail /var/log/messages

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 119 for more information.
No Communication Between the Sensor and the Console

Problem
You have applied a policy to the server sensor and now the sensor and the console cannot communicate.

Background
Because firecell signatures work like a firewall to prevent access to the server from unauthorized clients or to ensure that only authorized clients can access the server, you can easily disable all traffic to and from a system, including communication between the console and the server sensor.

Solution
To reestablish communication to the sensor:

1. Manually stop the issDaemon service.
   Reference: For the procedure on stopping the issDaemon, see “Stopping the issDaemon” on page 125.
2. Manually delete current.policy on the affected sensor.
   Reference: For the procedure on deleting current.policy, see “Deleting current.policy” on page 126.
3. Manually restart the issDaemon.
   Reference: For the procedure on starting the issDaemon, see “Starting the issDaemon” on page 126.
Appendixes
Appendix A

Default Block Response Settings in Server Sensor 6.5 for Windows

Introduction

Blocking behavior changed between the RealSecure server sensor version 6.0 and server sensor for Windows version 6.5 releases. In server sensor 6.0, the Block response controls both blocking and dynamic blocking behavior. In server sensor for Windows 6.5, the Block response controls dynamic blocking at the signature level and blocking is not user-configurable. Because the Block response behavior is different between versions, the default Block response setting was changed for some signatures. This appendix lists those changes.

Note: For more information about how the Block response works in server sensor for Windows 6.5, see the RealSecure Workgroup Manager User Guide.

Comparison

The following table shows how the Block response functions differently in server sensor versions 6.0 and earlier and in server sensor 6.5:

<table>
<thead>
<tr>
<th>To...</th>
<th>Server sensor 6.0...</th>
<th>Server sensor 6.5 for Unix...</th>
<th>Server sensor 6.5 for Windows...</th>
</tr>
</thead>
</table>
| Block | • enable the Block response  
|       | • set the Block Period to zero (0) | • enable the Block response  
|       |                                | • set the Block Period to zero (0) | this feature is not user-configurable. |
| Dynamic block | • enable the Block response  
|              | • set the Block Period | • enable the Block response  
|              |                                | • set the Block Period | • enable dynamic blocking for the sensor  
|              |                                |                                | • enable dynamic blocking for the signature |

Reference: For more information, see the RealSecure Workgroup Manager User Guide.
### Default Block response changes

The following table lists the default block response settings for server sensor for Windows 6.0 and server sensor 6.5:

**Note:** When the 6.5 default block response is set to off, the signature is not enabled to block dynamically; however, the signature itself may be set to block. This can be confusing, see Appendix B, “Signatures that Block by Default in Server Sensor 6.5 for Windows” for a list of signatures that block.

<table>
<thead>
<tr>
<th>Signature</th>
<th>Pre-6.0.1 default</th>
<th>6.5 default</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackOrifice</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Chargen_Denial_of_Service</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Echo_Denial_of_Service</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>All SMTP</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>Email_Qmail_Rcpt</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>All Finger</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>Finger_Bomb</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>All FTP</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>FTP_Bounce</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>FTP_PrivilegedBounce</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>All HTTP</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>HTTP_AnyForm</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_Cachemgr</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_Classifieds_Post</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_Cold_Fusion</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_EZMall2000</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_FaxSurvey</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_FormMail</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IIS$DATA</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IISHTR_Overflow</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IIS_Index_Server_Overflow</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IIS_ISAPI_Printer_Overflow</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IIS_Unicde_Translation</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IIS_URL_Decoding</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_IndexServer_Webhits</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>HTTP_JJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP_Orderform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP_PDGSofte</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 37:** Dynamic Block default setting changes
<table>
<thead>
<tr>
<th>Signature</th>
<th>Pre-6.0.1 default</th>
<th>6.5 default</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP_PHP_Read</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_QuickStore</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_Shells</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_SiteCsc_Access</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_Softcart</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_TestCgi</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_WebFinger</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_Website_Sample</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>HTTP_WebStore</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>IPFrag</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>IPHalfScan</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>IPProtocolViolation</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Land</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Land_UDP</td>
<td></td>
<td>off</td>
</tr>
<tr>
<td>Mstream_Zombie</td>
<td></td>
<td>on</td>
</tr>
<tr>
<td>Nmap_Scan</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>PingOfDeath</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Queso_Scan</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>TearDrop</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>TrinooDaemon</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>Win_IGMP_DOS</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Windows_OOB</td>
<td>on</td>
<td></td>
</tr>
</tbody>
</table>

Table 37: Dynamic Block default setting changes (Continued)
Appendix B

Signatures that Block by Default in Server Sensor 6.5 for Windows

Introduction

In server sensor 6.5 for Windows, only dynamic blocking is user-configurable. Blocking is not user-configurable but certain signatures are configured to block by default. This appendix lists these signatures.

Signatures that automatically drop attack packets

The following table lists the signatures in server sensor 6.5 for Windows that automatically block:

<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE Issue ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackOrifice</td>
<td>Back Orifice scan seen</td>
<td>2001501</td>
</tr>
<tr>
<td>BackOrifice</td>
<td>Back Orifice response</td>
<td>2001505</td>
</tr>
<tr>
<td>BackOrifice</td>
<td>Back Orifice ping</td>
<td>2001506</td>
</tr>
<tr>
<td>Chargen_Denial_of_Service</td>
<td>Chargen Denial of Service</td>
<td>2000211</td>
</tr>
<tr>
<td>Echo_Denial_of_Service</td>
<td>Echo Denial of Service</td>
<td>2000210</td>
</tr>
<tr>
<td>HTTP_Nimda_Riched20dll</td>
<td>Nimda RICHED20.DLL</td>
<td>2002710</td>
</tr>
<tr>
<td>IP_Unaligned_Timestamp</td>
<td>IP unaligned timestamp</td>
<td>2000017</td>
</tr>
<tr>
<td>IPFrag</td>
<td>IP fragment overlap</td>
<td>2000009</td>
</tr>
<tr>
<td>IPHalfScan</td>
<td>TCP xmas scan</td>
<td>2000308</td>
</tr>
<tr>
<td>IPHalfScan</td>
<td>TCP null scan</td>
<td>2000309</td>
</tr>
<tr>
<td>IPHalfScan</td>
<td>TCP SYN with URG flag</td>
<td>2000317</td>
</tr>
<tr>
<td>IPProtocolViolation</td>
<td>TCP OS fingerprint</td>
<td>2000313</td>
</tr>
<tr>
<td>Jolt2</td>
<td>Jolt2</td>
<td>2000018</td>
</tr>
<tr>
<td>Land</td>
<td>Land attack</td>
<td>2000001</td>
</tr>
<tr>
<td>Land_UDP</td>
<td>Snork attack</td>
<td>2000203</td>
</tr>
<tr>
<td>Nmap_Scan</td>
<td>NMAP OS fingerprint</td>
<td>2000314</td>
</tr>
<tr>
<td>Oshare_Attack</td>
<td>Oshare attack</td>
<td>2000024</td>
</tr>
<tr>
<td>PingOfDeath</td>
<td>Ping of death</td>
<td>2000012</td>
</tr>
</tbody>
</table>

Table 38: Signatures blocked in server sensor 6.5 for Windows
# Chapter B: Signatures that Block by Default in Server Sensor 6.5 for Windows

<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE Issue ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PingOfDeath</td>
<td>Jolt</td>
<td>2000019</td>
</tr>
<tr>
<td>Queso_Scan</td>
<td>Queso Scan</td>
<td>2000321</td>
</tr>
<tr>
<td>SMB_Malformed</td>
<td>SMB malformed</td>
<td>2000501</td>
</tr>
<tr>
<td>TearDrop</td>
<td>Teardrop attack</td>
<td>2000003</td>
</tr>
<tr>
<td>TearDrop</td>
<td>NewTear attack</td>
<td>2000004</td>
</tr>
<tr>
<td>TearDrop</td>
<td>SynDrop attack</td>
<td>2000005</td>
</tr>
<tr>
<td>TearDrop</td>
<td>TearDrop2 attack</td>
<td>2000006</td>
</tr>
<tr>
<td>TearDrop</td>
<td>Nestea attack</td>
<td>2000015</td>
</tr>
<tr>
<td>Win_IFMP_DOS</td>
<td>IGMP fragments</td>
<td>2002902</td>
</tr>
<tr>
<td>Windows_OOB</td>
<td>WinNuke attack</td>
<td>2000303</td>
</tr>
</tbody>
</table>

*Table 38: Signatures blocked in server sensor 6.5 for Windows (Continued)*
Appendix C

User-Defined Network Signature Category Descriptions and Configuration Information

Introduction
This topic describes the categories of events you can define a network signature for. This topic also describes the configuration information needed for the procedure “Editing the server sensor policy” on page 64.

User-specified filename

Issue numbers: 2010000-2010999

Description: Reserved for user-specified file names. By setting a configuration parameter, you can trigger an intrusion detection based on a file name.

Configuration Information: The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smb.filename.2010xxx path/</td>
<td>An intrusion with an id of 2010xxx is triggered if the SMB file name matches</td>
</tr>
<tr>
<td>name.type</td>
<td>the specified path/name.type. You can use a wildcard for the path, name, or type.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> To trigger detection number 2010023 whenever someone accesses</td>
</tr>
<tr>
<td></td>
<td>the file accounting.dat using SMB in any directory, add the following line to</td>
</tr>
<tr>
<td></td>
<td>the blackice.ini file:</td>
</tr>
<tr>
<td></td>
<td>smb.filename.2010023=*/)accounting.dat</td>
</tr>
<tr>
<td>ftp.filename.2010xxx path/</td>
<td>An intrusion with an id of 2010xxx is triggered if the FTP file name matches</td>
</tr>
<tr>
<td>name.type</td>
<td>the specified path/name.type.</td>
</tr>
<tr>
<td>tftp.filename.2010xxx path/</td>
<td>An intrusion with an id of 2010xxx is triggered if the TFTP file name matches</td>
</tr>
<tr>
<td>name.type</td>
<td>the specified path/name.type.</td>
</tr>
<tr>
<td>httpget.filename.2010xxx path/</td>
<td></td>
</tr>
<tr>
<td>name.type</td>
<td>An intrusion with an id of 2010xxx is triggered if the HTTP GET argument has</td>
</tr>
<tr>
<td></td>
<td>a file name that matches the one specified in path/name.type.</td>
</tr>
<tr>
<td>httppost.filename.2010xxx path/</td>
<td></td>
</tr>
<tr>
<td>name.type</td>
<td>An intrusion with an id of 2010xxx is triggered if the HTTP POST field has a</td>
</tr>
<tr>
<td></td>
<td>file name that matches the one specified in path/name.type.</td>
</tr>
</tbody>
</table>

Table 39: Configuring user-specified filenames
User–specified URL

Issue numbers: 2011000-2011999

Description: Reserved for user–specified URLs. By setting a configuration parameter, you can trigger 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.

Restrictions: There are two restrictions when creating user-specified URL signatures:

- you cannot specify more than one URL across the 2011000-2011999 category
- you cannot specify the following:
  - url.2011001.0 = http://www.networkice.com/sales/index.htm
  - url.2011032.0 = http://www.networkice.com/sales/index.htm

Example:

- To trigger on any reference to a file in the Admin_files directory, use the command
  url.2011001.0=*/Admin_files/*.*.

Example:

- To trigger on a CGI program named test.cgi with a POSTed field named joebob containing metacharacters, use url.2011001.101=/cgi-bin/test.cgi?joebob.

Example:

- You can map multiple URLs to the same signature number:
  url.2011021.0 = http://www.networkice.com/sales/index.htm
  url.2011021.3 = http://www.networkice.com/products/index.htm

Configuration Information: The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url.2011xxx.0</td>
<td>URL Triggers if the specified URL is seen.</td>
</tr>
<tr>
<td>url.2011xxx.1</td>
<td>URL Triggers if the specified URL is seen with any argument.</td>
</tr>
<tr>
<td>url.2011xxx.2</td>
<td>URL Triggers if the specified URL is seen with any argument</td>
</tr>
<tr>
<td>url.2011xxx.3</td>
<td>URL Triggers if the specified URL is seen with an argument</td>
</tr>
<tr>
<td>url.2011xxx.4</td>
<td>URL arg Triggers if the specified URL is seen with the argument</td>
</tr>
<tr>
<td>url.2011xxx.5</td>
<td>URL field=value Triggers if the specified URL is seen with a</td>
</tr>
<tr>
<td>url.2011xxx.101</td>
<td>URL field Triggers if the specified URL is seen with a POSTed</td>
</tr>
</tbody>
</table>

Table 40: Configuring user–specified URLs
**User–specified email recipient**

**Issue numbers:** 2012000-2012999

**Description:** Reserved for user–specified email recipients. By setting a configuration parameter, you can trigger an intrusion detection based on the recipient’s name.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recipient.2012xxx email address</td>
<td>An intrusion with an id of 2012xxx is triggered if the email address matches this parameter.</td>
</tr>
</tbody>
</table>

*Table 41: Configuring user-specified email recipients*

**User–specified email pattern**

**Issue numbers:** 2013000-2013999

**Description:** Reserved for user–specified email patterns. By setting a configuration parameter, you can trigger an intrusion detection based on a pattern match with a regular expression.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| emailpattern.2013xxx regular expression | An intrusion with an id of 2013xxx is triggered if an email line matches the specified regular expression.  
**Note:** This feature can slow frame processing considerably. |
| email.datascan off | Specifies whether the email data should be scanned.  
**Note:** For improved performance, the default setting is off and only email headers are scanned. To scan data within the email itself, set this to on. |

*Table 42: Configuring user-specified email patterns*

**User–specified MIME–attached filename**

**Issue numbers:** 2014000-2014999

**Description:** Reserved for user–specified MIME–attached filenames. By setting a configuration parameter, you can trigger an intrusion detection based on a MIME–attached filename. Both incoming and outgoing email is checked for the specified filenames.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mimefilename.2014xxx filename</td>
<td>An intrusion with an id of 2014xxx is triggered if the specified filename matches.</td>
</tr>
</tbody>
</table>

*Table 43: Configuring user-specified MIME-attached filenames*
User-specified TCP probe port

**Issue numbers:** 2015000-2015999

**Description:** Reserved for user-specified TCP probe port numbers. By setting a configuration parameter, you can trigger an intrusion detection based on a failed TCP connection attempt to a particular port number.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpprobe.2015xxx.port protocol</td>
<td>An intrusion with an id of 2015xxx is triggered if a connection is attempted on the specified port.</td>
</tr>
<tr>
<td>name</td>
<td></td>
</tr>
</tbody>
</table>

**Example:** To trigger a detection on a connection attempt to port 1234, use the command `tcpprobe.2015001.1234=something`

Table 44: Configuring user-specified TCP probe port numbers

User-specified UDP probe port

**Issue numbers:** 2016000-2016999

**Description:** 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.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>udpprobe.2016xxx.port protocol</td>
<td>An intrusion with an id of 2016xxx is triggered if a connection is attempted on the specified port.</td>
</tr>
<tr>
<td>name</td>
<td></td>
</tr>
</tbody>
</table>

**Example:** To trigger a detection on a connection attempt to port 1234, use the command `udpprobe.2016001.1234=something`

Table 45: Configuring user-specified UDP probe port numbers

User-specified registry key

**Issue numbers:** 2017000-2017999

**Description:** Reserved for user-specified registry keys. By setting a configuration parameter, you can trigger an intrusion detection based on remote access to a registry key.
**User-specified TCP trojan response**

**Issue numbers:** 2018000-2018999

**Description:** Reserved for user-specified TCP trojan horse strings. By setting a configuration parameter, you can trigger an intrusion detection based on a specific string seen in the first data frame of a connection.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>registrykey.2017xxx.1 key</td>
<td>An intrusion with an id of 2017xxx is triggered if remote access to the specified key is detected. <strong>Note:</strong> The registry key can contain a wildcard as part of the key path. <strong>Example:</strong> To trigger detection number 2017001 whenever the server sensor detects remote access to a registry key that includes the path /SOFTWARE/Microsoft/Windows/CurrentVersion/, use the command <code>registrykey.2017001.1=*/SOFTWARE/Microsoft/Windows/CurrentVersion/</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcptrojan.2018xxx.port trojan horse string</td>
<td>An intrusion with an id of 2018xxx is triggered if the specified string is seen on the first data frame on the specified TCP port.</td>
</tr>
</tbody>
</table>

**Table 47: Configuring user-specified TCP trojan responses**

**User-specified IRC channel name**

**Issue numbers:** 2019000-2019999

**Description:** Reserved for user-specified IRC channel names. By setting a configuration parameter, you can trigger an intrusion detection based on a specific IRC channel name seen.

**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ircchannel.2019xxx channel name</td>
<td>An intrusion with an id of 2019xxx is triggered if the specified channel name is used.</td>
</tr>
</tbody>
</table>

**Table 48: Configuring user-specified IRC channel names**

**User-specified Java pattern**

**Issue numbers:** 2020000-2020999

**Description:** Reserved for user-specified Java patterns. By setting a configuration parameter, you can trigger an intrusion detection based on a pattern match with a regular expression.
**Configuration Information:** The following table lists the parameters you can configure for this category of events:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>javapattern.2020xxx regular expression</td>
<td>An intrusion with an id of 2020xxx is triggered if a Java applet string matches the specified regular expression. <strong>Note:</strong> This feature can slow down Java applet processing considerably.</td>
</tr>
<tr>
<td>java.parsing on</td>
<td>Specifies whether the Java applets should be scanned.</td>
</tr>
</tbody>
</table>

*Table 49: Configuring user-specified Java patterns*
Appendix D

RealSecure to BlackICE Signature Mapping

Introduction

The RealSecure server sensor version 6.5 for Windows integrates BlackICE Agent 3.0 with the RealSecure server sensor for Windows NT and Windows 2000. This appendix provides a table that lists how the RealSecure signatures are mapped to BlackICE issues.

RealSecure and BlackICE signature relationship

The following table lists how RealSecure signatures and BlackICE issues are mapped for the server sensor 6.5 for Windows release:

<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admind</td>
<td>rpc.admind auth</td>
<td>2001704</td>
</tr>
<tr>
<td>Allaire_JRun_WebInf_DotSlash</td>
<td>HTTP URL contains ./</td>
<td>2000616</td>
</tr>
<tr>
<td>BackOrifice</td>
<td>Back Orifice scan seen</td>
<td>2001501</td>
</tr>
<tr>
<td></td>
<td>Back Orifice response</td>
<td>2001505</td>
</tr>
<tr>
<td></td>
<td>Back Orifice ping</td>
<td>2001506</td>
</tr>
<tr>
<td>Bind_Version_Request</td>
<td>DNS Chaos lookup</td>
<td>2000411</td>
</tr>
<tr>
<td></td>
<td>DNS BIND version request</td>
<td>2000417</td>
</tr>
<tr>
<td>Chargen_Denial_of_Service</td>
<td>Chargen Denial of Service</td>
<td>2000211</td>
</tr>
<tr>
<td>CyberCop_Scanner</td>
<td>Cybercop FTP scan</td>
<td>2001510</td>
</tr>
<tr>
<td>DNS_Hostname_Overflow</td>
<td>DNS name very long</td>
<td>2000403</td>
</tr>
<tr>
<td></td>
<td>DNS name overflow very long</td>
<td>2000423</td>
</tr>
<tr>
<td>DNS_Iquery</td>
<td>DNS I-Query</td>
<td>2000409</td>
</tr>
<tr>
<td>DNS_Length_Overflow</td>
<td>DNS Internet not 4 bytes</td>
<td>2000406</td>
</tr>
<tr>
<td>DNS_Zone_Transfer</td>
<td>DNS zone transfer</td>
<td>2000401</td>
</tr>
<tr>
<td>Echo_Denial_of_Service</td>
<td>Echo Denial of Service</td>
<td>2000210</td>
</tr>
<tr>
<td>Email_Debug</td>
<td>SMTP DEBUG command</td>
<td>2001002</td>
</tr>
<tr>
<td>Email_Decode</td>
<td>SMTP mail to decode alias</td>
<td>2001013</td>
</tr>
<tr>
<td></td>
<td>SMTP mail to uudecode alias</td>
<td>2001014</td>
</tr>
<tr>
<td>Email_ExchangeStore_DoS</td>
<td>SMTP MIME null charset</td>
<td>2001028</td>
</tr>
<tr>
<td>Email_Expn</td>
<td>SMTP EXPN command</td>
<td>2001004</td>
</tr>
</tbody>
</table>

Table 50: RealSecure signature to BlackICE issue mapping
### Chapter D: RealSecure to BlackICE Signature Mapping

<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email_Helo_Overflow</td>
<td>SMTP login name overflow</td>
<td>2001003</td>
</tr>
<tr>
<td>Email_Outlook_Date_Overflow</td>
<td>SMTP date overflow</td>
<td>2001025</td>
</tr>
<tr>
<td>Email_Pipe</td>
<td>SMTP pipe in mail address</td>
<td>2001001</td>
</tr>
<tr>
<td>Email_Qmail_Rcpt</td>
<td>SMTP Too many recipients</td>
<td>2001007</td>
</tr>
<tr>
<td>Email_Relay_Spam</td>
<td>SMTP relay attempt</td>
<td>2001011</td>
</tr>
<tr>
<td>Email_Vrfy</td>
<td>SMTP VRFY command</td>
<td>2001005</td>
</tr>
<tr>
<td>Email_WIZ</td>
<td>SMTP WIZ command</td>
<td>2001006</td>
</tr>
<tr>
<td>EvilFTP_Backdoor</td>
<td>EvilFTP trojan horse activity</td>
<td>2001522</td>
</tr>
<tr>
<td>Finger_Bomb</td>
<td>Finger forwarding</td>
<td>2001102</td>
</tr>
<tr>
<td>Finger_RTMT</td>
<td>Finger overflow</td>
<td>2001107</td>
</tr>
<tr>
<td>FTP_Args</td>
<td>FTP command line overflow</td>
<td>2001310</td>
</tr>
<tr>
<td>FTP_Bounce</td>
<td>FTP PORT bounce to other system</td>
<td>2001302</td>
</tr>
<tr>
<td>FTP_PrivilegedBounce</td>
<td>FTP Privileged Bounce</td>
<td>2001323</td>
</tr>
<tr>
<td>FTP_PrivilegedPort</td>
<td>FTP PORT restricted</td>
<td>2001303</td>
</tr>
<tr>
<td>FTP_Root</td>
<td>FTP CWD ~root command</td>
<td>2001304</td>
</tr>
<tr>
<td>FTP_Site_Cmd</td>
<td>FTP SITE EXEC command</td>
<td>2001305</td>
</tr>
<tr>
<td>FTP_Site_Exec_DotDot</td>
<td>FTP Site Exec DotDot</td>
<td>2001324</td>
</tr>
<tr>
<td>FTP_Site_Exec_Format_Attack</td>
<td>FTP SITE EXEC exploit</td>
<td>2001322</td>
</tr>
<tr>
<td>FTP_Site_Exec_Tar</td>
<td>FTP compress exec exploit</td>
<td>2001315</td>
</tr>
<tr>
<td>GateCrasher</td>
<td>GateCrasher trojan horse activity</td>
<td>2001520</td>
</tr>
<tr>
<td>HP_Openview_SNMP_Backdoor</td>
<td>SNMP backdoor</td>
<td>2002003</td>
</tr>
<tr>
<td>HTTP_AnyForm</td>
<td>CGI Anyform2</td>
<td>2002504</td>
</tr>
<tr>
<td>HTTP_Apache_DOS</td>
<td>HTTP URL has many slashes</td>
<td>2000606</td>
</tr>
<tr>
<td>HTTP_Cachemgr</td>
<td>Squid cachemgr.cgi</td>
<td>2002594</td>
</tr>
<tr>
<td>HTTP_Campas</td>
<td>CGI campas</td>
<td>2002506</td>
</tr>
<tr>
<td>HTTP_Cart32_ChangeAdminPassword</td>
<td>Cart32 ChangeAdminPassword URL</td>
<td>2002580</td>
</tr>
<tr>
<td>HTTP_Cart32_ClientList</td>
<td>Cart32_Clientlist URL</td>
<td>2002579</td>
</tr>
<tr>
<td>HTTP_Classifieds_Post</td>
<td>classifieds.cgi</td>
<td>2002590</td>
</tr>
<tr>
<td>HTTP_Cold_Fusion</td>
<td>Cold Fusion Sample URL</td>
<td>2002554</td>
</tr>
<tr>
<td>HTTP_Cold_Fusion_Cfcache</td>
<td>Cold Fusion cache file URL</td>
<td>2002567</td>
</tr>
<tr>
<td>HTTP_Count</td>
<td>HTTP GET data with repeated char</td>
<td>2000611</td>
</tr>
<tr>
<td></td>
<td>count.cgi</td>
<td>2002588</td>
</tr>
</tbody>
</table>

Table 50: RealSecure signature to BlackICE issue mapping (Continued)
<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP_DotDot</td>
<td>HTTP URL contains ../../../..</td>
<td>2000603</td>
</tr>
<tr>
<td></td>
<td>HTTP GET data contains ../../../..</td>
<td>2000609</td>
</tr>
<tr>
<td>HTTP_EZMail2000</td>
<td>EZMail data URL</td>
<td>2002551</td>
</tr>
<tr>
<td>HTTP_FaxSurvey</td>
<td>CGI faxsurvey</td>
<td>2002509</td>
</tr>
<tr>
<td>HTTP_FormMail</td>
<td>CGI formmail</td>
<td>2002511</td>
</tr>
<tr>
<td></td>
<td>CGI formmail.pl</td>
<td>2002512</td>
</tr>
<tr>
<td>HTTP_Glimpse</td>
<td>CGI glimpse</td>
<td>2002513</td>
</tr>
<tr>
<td>HTTP_Guestbook</td>
<td>CGI guestbook.cgi</td>
<td>2002514</td>
</tr>
<tr>
<td></td>
<td>CGI guestbook.pl</td>
<td>2002515</td>
</tr>
<tr>
<td>HTTP_HTMLScript</td>
<td>CGI htmlscript</td>
<td>2002517</td>
</tr>
<tr>
<td>HTTP_IE_BAT</td>
<td>bat URL type</td>
<td>2002501</td>
</tr>
<tr>
<td></td>
<td>cmd URL type</td>
<td>2002502</td>
</tr>
<tr>
<td>HTTP_IE3_URL</td>
<td>.url URL type</td>
<td>2002545</td>
</tr>
<tr>
<td></td>
<td>.lnk URL type</td>
<td>2002546</td>
</tr>
<tr>
<td>HTTP_IIS$DATA</td>
<td>HTTP URL with ::$DATA appended</td>
<td>2000607</td>
</tr>
<tr>
<td>HTTP_IIS_Index_Server_Overflow</td>
<td>ISAPI extension overflow</td>
<td>2002608</td>
</tr>
<tr>
<td>HTTP_IIS_ISAPI_Printer_Overflow</td>
<td>IIS printer overflow</td>
<td>2002607</td>
</tr>
<tr>
<td>HTTP_IIS_Obtain_Code</td>
<td>HTTP URL with +.htr append</td>
<td>2000633</td>
</tr>
<tr>
<td>HTTP_IIS_Showcode</td>
<td>IIS sample URL</td>
<td>2002557</td>
</tr>
<tr>
<td>HTTP_IIS_Unicode_Translation</td>
<td>HTTP UTF8 backtick</td>
<td>2000639</td>
</tr>
<tr>
<td>HTTP_IIS_URL_Decoding</td>
<td>HTTP double-encoded ./</td>
<td>2000645</td>
</tr>
<tr>
<td>HTTP_IIS3_Asp_Dot</td>
<td>HTTP asp with . appended</td>
<td>2000604</td>
</tr>
<tr>
<td>HTTP_IISHTR_Overflow</td>
<td>IIS malformed HTR request</td>
<td>2002559</td>
</tr>
<tr>
<td>HTTP_INDEXServer_Webhits</td>
<td>Index Server null.htw exploit</td>
<td>2002575</td>
</tr>
<tr>
<td>HTTP_Info2WWW</td>
<td>CGI info2www</td>
<td>2002518</td>
</tr>
<tr>
<td>HTTP_JJ</td>
<td>CGI jj exploit</td>
<td>2002589</td>
</tr>
<tr>
<td>HTTP_JSP_SourceRead</td>
<td>HTTP JavaServer URL in caps</td>
<td>2000632</td>
</tr>
<tr>
<td>HTTP_MachineInfo</td>
<td>CGI machineinfo</td>
<td>2002519</td>
</tr>
<tr>
<td>HTTP_MDAC_Access</td>
<td>IIS data service query</td>
<td>2002560</td>
</tr>
<tr>
<td>HTTP_NCSA_Buffer_Overflow</td>
<td>HTTP URL overflow</td>
<td>2000601</td>
</tr>
<tr>
<td></td>
<td>HTTP GET very long</td>
<td>2000608</td>
</tr>
<tr>
<td>HTTP_Netscape_SpaceView</td>
<td>HTTP URL with %20 appended</td>
<td>2000610</td>
</tr>
<tr>
<td>HTTP_Nimda_Riched20dll</td>
<td>Nimda RICHED20.DLL</td>
<td>2002710</td>
</tr>
<tr>
<td>HTTP_NphTestCgi</td>
<td>CGI nph-test-cgi</td>
<td>2002520</td>
</tr>
</tbody>
</table>

Table 50: RealSecure signature to BlackICE issue mapping (Continued)
### Table 50: RealSecure signature to BlackICE issue mapping (Continued)

<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP_Orderform</td>
<td>Order Form V1.2 data URL</td>
<td>2002549</td>
</tr>
<tr>
<td></td>
<td>Order Form data URL</td>
<td>2002550</td>
</tr>
<tr>
<td>HTTP_PDGSof</td>
<td>Shopping cart order URL</td>
<td>2002548</td>
</tr>
<tr>
<td>HTTP_PHPF</td>
<td>CGI phf</td>
<td>2002524</td>
</tr>
<tr>
<td>HTTP_PHP_Read</td>
<td>HTTP cgi starting with php</td>
<td>2000602</td>
</tr>
<tr>
<td>HTTP_QuickStore</td>
<td>QuikStore configuration URL</td>
<td>2002552</td>
</tr>
<tr>
<td>HTTP_Shells</td>
<td>CGI bash</td>
<td>2002505</td>
</tr>
<tr>
<td></td>
<td>CGI csh</td>
<td>2002508</td>
</tr>
<tr>
<td></td>
<td>CGI perl.exe</td>
<td>2002522</td>
</tr>
<tr>
<td></td>
<td>CGI rksh</td>
<td>2002527</td>
</tr>
<tr>
<td></td>
<td>CGI sh</td>
<td>2002528</td>
</tr>
<tr>
<td></td>
<td>CGI tcsh</td>
<td>2002529</td>
</tr>
<tr>
<td>HTTP_SiteCsc_Access</td>
<td>Site Server sample URL</td>
<td>2002556</td>
</tr>
<tr>
<td>HTTP_Softcart</td>
<td>SoftCart password URL</td>
<td>2002553</td>
</tr>
<tr>
<td>HTTP_TestCgi</td>
<td>CGI test-cgi.tcl</td>
<td>2002530</td>
</tr>
<tr>
<td></td>
<td>CGI test-cgi</td>
<td>2002531</td>
</tr>
<tr>
<td>HTTP_UNIX_Pwds</td>
<td>SMB passwd file</td>
<td>2002701</td>
</tr>
<tr>
<td></td>
<td>FTP passwd file</td>
<td>2003601</td>
</tr>
<tr>
<td></td>
<td>TFTP passwd file</td>
<td>2003701</td>
</tr>
<tr>
<td></td>
<td>HTTP GET passwd file</td>
<td>2003801</td>
</tr>
<tr>
<td></td>
<td>HTTP POST passwd file</td>
<td>2003901</td>
</tr>
<tr>
<td>HTTP_Webfinger</td>
<td>CGI finger</td>
<td>2002510</td>
</tr>
<tr>
<td>HTTP_Webgais</td>
<td>CGI webgais</td>
<td>2002534</td>
</tr>
<tr>
<td>HTTP_WebLogic_JavaInjection</td>
<td>HTTP path contains *.jhtml or *.jsp</td>
<td>2000634</td>
</tr>
<tr>
<td>HTTP_WebSendMail</td>
<td>CGI websendmail</td>
<td>2002535</td>
</tr>
<tr>
<td>HTTP_WebSite_Sample</td>
<td>CGI win-c-sample.exe</td>
<td>2002536</td>
</tr>
<tr>
<td>HTTP_WebSite_Uploader</td>
<td>CGI uploader.exe</td>
<td>2002538</td>
</tr>
<tr>
<td>HTTP_WebSphere_HeaderDoS</td>
<td>HTTP HOST: field overflow</td>
<td>2000636</td>
</tr>
<tr>
<td>HTTP_WebStore</td>
<td>WebStore admin URL</td>
<td>2002547</td>
</tr>
<tr>
<td>Ident_Error</td>
<td>identd invalid response</td>
<td>2001901</td>
</tr>
<tr>
<td>Ident_Newline</td>
<td>IDENT newline</td>
<td>2001905</td>
</tr>
<tr>
<td>IMAP Authenticate_OVERFLOW</td>
<td>IMAP4 authentication very long</td>
<td>2000803</td>
</tr>
<tr>
<td>IMAP_Overflow</td>
<td>IMAP4 user name overflow</td>
<td>2000801</td>
</tr>
<tr>
<td></td>
<td>IMAP4 password overflow</td>
<td>2000802</td>
</tr>
<tr>
<td>INN_Control</td>
<td>NNTP pipe seen</td>
<td>2002402</td>
</tr>
<tr>
<td>INN_Overflow</td>
<td>NNTP name overflow</td>
<td>2002401</td>
</tr>
<tr>
<td>RealSecure Signature</td>
<td>BlackICE Issue Name</td>
<td>BlackICE ID</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Innd_Cancel_Overflow</td>
<td>NNTP Message-ID too long</td>
<td>2002403</td>
</tr>
<tr>
<td>IP_Unaligned_Timestamp</td>
<td>IP unaligned timestamp</td>
<td>2000017</td>
</tr>
<tr>
<td>IPFrag</td>
<td>IP fragment overlap</td>
<td>2000009</td>
</tr>
<tr>
<td>IPHalfScan</td>
<td>TCP FIN scan</td>
<td>2000305</td>
</tr>
<tr>
<td></td>
<td>TCP xmas scan</td>
<td>2000308</td>
</tr>
<tr>
<td></td>
<td>TCP null scan</td>
<td>2000309</td>
</tr>
<tr>
<td></td>
<td>TCP SYN with URG flag</td>
<td>2000317</td>
</tr>
<tr>
<td>IPProtocolViolation</td>
<td>TCP OS fingerprint</td>
<td>2000313</td>
</tr>
<tr>
<td>ISS</td>
<td>ISS ping scan</td>
<td>2001508</td>
</tr>
<tr>
<td></td>
<td>ISS scan</td>
<td>2009201</td>
</tr>
<tr>
<td>Jolt2</td>
<td>Jolt2</td>
<td>2000018</td>
</tr>
<tr>
<td>Land</td>
<td>Land attack</td>
<td>2000001</td>
</tr>
<tr>
<td>Land_UDP</td>
<td>Snork attack</td>
<td>2000203</td>
</tr>
<tr>
<td>Loki</td>
<td>Loki</td>
<td>2000112</td>
</tr>
<tr>
<td>Mstream_Zombie</td>
<td>MStream agent activity</td>
<td>2001528</td>
</tr>
<tr>
<td>NetBus</td>
<td>NetBus response</td>
<td>2001502</td>
</tr>
<tr>
<td>NetSphere</td>
<td>NetSphere trojan horse activity</td>
<td>2001523</td>
</tr>
<tr>
<td></td>
<td>NetSphere client</td>
<td>2001527</td>
</tr>
<tr>
<td>Nmap_Scan</td>
<td>NMAP OS fingerprint</td>
<td>2000314</td>
</tr>
<tr>
<td></td>
<td>NMAP ping</td>
<td>2001526</td>
</tr>
<tr>
<td>Oshare_Attack</td>
<td>Oshare attack</td>
<td>2000024</td>
</tr>
<tr>
<td>PhaseZero</td>
<td>phAse Zero trojan horse activity</td>
<td>2001518</td>
</tr>
<tr>
<td>PingFlood</td>
<td>Echo storm</td>
<td>2000102</td>
</tr>
<tr>
<td>PingOfDeath</td>
<td>Ping of death</td>
<td>2000012</td>
</tr>
<tr>
<td></td>
<td>Jolt</td>
<td>2000019</td>
</tr>
<tr>
<td>PmapDump</td>
<td>rpc.portmap dump</td>
<td>2001705</td>
</tr>
<tr>
<td>PmapSet</td>
<td>rpc.portmap.set</td>
<td>2001708</td>
</tr>
<tr>
<td>PmapUnset</td>
<td>rpc.portmap.unset</td>
<td>2001709</td>
</tr>
<tr>
<td>POP_QPopAuth_Overflow</td>
<td>POP3 AUTH overflow</td>
<td>2000705</td>
</tr>
<tr>
<td>POP_QPopCommand_Overflow</td>
<td>POP3 command very long</td>
<td>2000704</td>
</tr>
<tr>
<td>Port_Scan</td>
<td>TCP port scan</td>
<td>2000301</td>
</tr>
<tr>
<td>Queso.Scan</td>
<td>Queso Scan</td>
<td>2000321</td>
</tr>
<tr>
<td>RPC_Cmsd_Overflow</td>
<td>rpc.cmsd overflow</td>
<td>2001717</td>
</tr>
<tr>
<td>Sadmind_Amslverify_Overflow</td>
<td>rpc.sadmind overflow</td>
<td>2001722</td>
</tr>
<tr>
<td>SMB_Malformed</td>
<td>SMB malformed</td>
<td>2000501</td>
</tr>
</tbody>
</table>

Table 50: RealSecure signature to BlackICE issue mapping (Continued)
<table>
<thead>
<tr>
<th>RealSecure Signature</th>
<th>BlackICE Issue Name</th>
<th>BlackICE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smurf</td>
<td>Possible Smurf attack initiated</td>
<td>2000103</td>
</tr>
<tr>
<td>SNMP_Delete_WINS</td>
<td>SNMP WINS deletion</td>
<td>2002006</td>
</tr>
<tr>
<td>SNMP_Suspicious_Set</td>
<td>SNMP SET sysContact</td>
<td>2002007</td>
</tr>
<tr>
<td>Statd_Automount_Exec</td>
<td>rpc.automountd overflow</td>
<td>2001701</td>
</tr>
<tr>
<td></td>
<td>rpc.statd with automount</td>
<td>2001716</td>
</tr>
<tr>
<td>Statd_Overflow</td>
<td>rpc.statd overflow</td>
<td>2001702</td>
</tr>
<tr>
<td>SubSeven</td>
<td>SubSeven trojan horse activity</td>
<td>2001519</td>
</tr>
<tr>
<td></td>
<td>SubSeven ICQ pager URL</td>
<td>2002572</td>
</tr>
<tr>
<td>Sun_SNMP_Backdoor</td>
<td>SNMP snmpdx attack</td>
<td>2002012</td>
</tr>
<tr>
<td>SYN_Flood</td>
<td>TCP SYN flood</td>
<td>2000302</td>
</tr>
<tr>
<td>TearDrop</td>
<td>Teardrop attack</td>
<td>2000003</td>
</tr>
<tr>
<td></td>
<td>NewTear attack</td>
<td>2000004</td>
</tr>
<tr>
<td></td>
<td>SynDrop attack</td>
<td>2000005</td>
</tr>
<tr>
<td></td>
<td>TearDrop2 attack</td>
<td>2000006</td>
</tr>
<tr>
<td></td>
<td>Nestea attack</td>
<td>2000015</td>
</tr>
<tr>
<td>Telnet_EnvAll</td>
<td>Telnet login name overflow</td>
<td>2000902</td>
</tr>
<tr>
<td></td>
<td>Telnet AUTH USER overflow</td>
<td>2000913</td>
</tr>
<tr>
<td>Telnet_EnvLong</td>
<td>Telnet ENV overflow</td>
<td>2000914</td>
</tr>
<tr>
<td>Telnet_EnvTermcap</td>
<td>Telnet bad TERMCAP</td>
<td>2000911</td>
</tr>
<tr>
<td>Telnet_Terminaltype</td>
<td>Telnet bad TERM</td>
<td>2000910</td>
</tr>
<tr>
<td>Telnet_TerminaltypeLong</td>
<td>Telnet terminal type overflow</td>
<td>2000904</td>
</tr>
<tr>
<td>ToolTalk_Overflow</td>
<td>rpc.tooltalkd overflow</td>
<td>2001703</td>
</tr>
<tr>
<td>Trace_Route</td>
<td>Traceroute</td>
<td>2000101</td>
</tr>
<tr>
<td>Tribe_Flood_Network</td>
<td>Echo reply without request</td>
<td>2000109</td>
</tr>
<tr>
<td>TrinooDaemon</td>
<td>Trinoo Daemon activity</td>
<td>2001525</td>
</tr>
<tr>
<td>TrinooMasterAttempt</td>
<td>Trinoo Master activity</td>
<td>2001524</td>
</tr>
<tr>
<td>UDP_Port_Scan</td>
<td>UDP port scan</td>
<td>2000201</td>
</tr>
<tr>
<td>VNC_Login_Failed</td>
<td>VNC login failed</td>
<td>2001612</td>
</tr>
<tr>
<td>VNC_NoAuthentication</td>
<td>VNC no authentication</td>
<td>2002905</td>
</tr>
<tr>
<td>Win_I_GMP_DOS</td>
<td>IGMP fragments</td>
<td>2002902</td>
</tr>
<tr>
<td>Win_IP_Src_Route</td>
<td>SSPING attack</td>
<td>2000021</td>
</tr>
<tr>
<td>Windows_OOB</td>
<td>WinNuke attack</td>
<td>2000303</td>
</tr>
<tr>
<td>Windows_Registry_Read</td>
<td>SMB winreg file</td>
<td>2002703</td>
</tr>
</tbody>
</table>

*Table 50: RealSecure signature to BlackICE issue mapping (Continued)*
Appendix E

Configuring the RealSecure Web Server Monitoring Component

Introduction

A server sensor full upgrade 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, it must be manually configured after the upgrade is installed. This appendix describes how to manually configure your Web server monitoring component.

Configuring the Apache Web server monitoring component manually

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](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 warning each time you restart the Web server that the module is not compiled with the EAPI flag and that the Web server may crash. This warning 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 Internet Security Systems to prevent this warning from appearing. See “Installing an EAPI flag compiled Apache module” on page 153 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.
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 the **Edit** button. The WWW Service Master Properties for Web Server window appears.
4. Select the **ISAPI Filter** tab, and then click the **Add** button.
5. In the **Filter Name** box, type `rsiisfilter`.
6. In the **Executable** box, type the full path to the `rsiisfilter.dll`. 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`
7. Click OK. The system applies your changes.
8. Restart IIS.
Index

symbols

@Field N 34
@StringN 34
@String * 34
__iss_attacktime 101
__iss_dstip 101
__iss_dstport 101
__iss_priority 101
__iss_rulename 101
__iss_scip 101
__iss_srcip 101
__iss_srcport 101
{!} 34

a

about server sensor 6
about user-defined signatures 33
adding
  fusion scripting to signatures 108
all files, monitoring 48
allowing
  only Internet traffic 90
  only local traffic 90

b

binary log monitoring 53
block response
  changes to default settings 136
blocking 6
  at the application layer 6
  before reaching the IP stack 7
definition 6

c

capturing packet information 26
changing event priority 115
CLI 4
Command Line Interface 4
configuring
  Apache Web server 153
  BSM audit management 78
  firecell signatures 85
  fusion scripting responses 109
  fusion scripting SNMPv3 response 110
  IIS Web server 154
  user-defined network signature 65
console 4
conventions, typographical
  in commands x
  in procedures x
  in this manual x
correlating information between signatures 115
creating
  firecell signatures 85
  user-defined binary log signatures 53

d

default block response
  changes 6.0 to 6.5 136
Default.policy 17
definition
  blocking 6
dynamic blocking 6
events 8
policies 8

external network traffic 90
blocking, example 86
BSM See Solaris Basic Security Module
policy files 8
signatures 8
deploying
  RealSecure 5
  sensors 5
disabling
  firecell signatures 89
  fusion scripting 119
  wtmpx log monitoring 54
dynamic blocking 6
definition 6

editing
  policy to add user-defined network signature 64
enabling
  fusion scripting SNMPv3 response 112
  wtmpx log monitoring 54
event data values 101
event detection
  at the application layer 6
  before reaching the IP stack 7
event priority
  changing 115
events 8
  monitoring syslog 50, 52
  priority 115
  priority of 101
  processing with fusion scripting 98
  processing without fusion scripting 98
  uniquely identifying 107
evidence logging 26
examples
  firecell signatures 90
extensions
  Tcl 106

file rotation 48
file switching 48
firecell signatures 81
  about 82
  allowing local subnet traffic 90
  allowing only Internet traffic 90
  allowing typical Internet traffic 90
  blocking traffic with 90
  configuring 85
  creating 85
  disabling 89
  examples 90
  monitoring
    external network traffic 90
    ip traffic 90
  order of 87
  prerequisites to using 85
  rearranging order of 87
  relevance of order of 83
  responses available to 83
  specifying ports 87
  using 90
  when to use 82
fusion scripting
  adding to signatures 108
  configuring responses 109
  correlating information between signatures 115
  disabling 119
  event data values and 101
  information fields and 100
  initialization scripts 102
  prerequisites 99
  priority level and 101
  procedure scripts 103
  process 102
  response scripts 103
  responses 109
  true and false values 113
  validation script values 113
  validation scripts 103
  when to use 98
fusion scripting response
  configuring 109
  configuring SNMPv3 110
  enabling SNMPv3 112
fusion scripts
  accessing information fields 114
  adding 108
  changing event priority 115
  creating new information fields 114
  modifying 108
  monitoring trusted users 117
  passing information to responses 115
  saving information fields 114
  saving variables to log files 116
  using initialization scripts 114
Index

**g**
-GetData 106
-GetTid 107
-global variables 100

**i**
- importing policies 13
  - to65policy.exe executable 15
- information fields 100
- initialization scripts 102
- Internet Security Systems
  - technical support xi
  - Web site xi
- intrusion prevention
  - block response 7
  - firecell signatures 7
- IP address
  - blocking options 86

**l**
- logging
  - evidence 26
  - packet 26

**m**
- monitoring
  - binary logs 53
  - external network traffic 90
  - for a text string 47
  - ip traffic using firecell signatures 90
  - log file 46
  - newest or all files 48
  - syslog events 50, 52
  - trusted users 117
  - wtmpx binary log file 53

**n**
- network event categories
  - user-specified email pattern 143
  - user-specified email recipient 143
  - user-specified filename 141
  - user-specified irc channel name 145
  - user-specified java pattern 145
  - user-specified MIME-attached filename 143
  - user-specified registry key 144
  - user-specified tcp probe port 144
  - user-specified tcp trojan response 145
  - user-specified udp probe port 144
  - user-specified url 142
- Network Events tab 22
- network signatures
  - user-defined 61
- newest files, monitoring 48
  - Newest Only flag 48

**o**
- order of firecell signatures 87
- ordering of firecell signatures 83
- OS Events tab 22

**p**
- packet logging 26
- persistent memory 100
- policies 8
  - implications when managing earlier version
    - sensors 16
  - importing 13
  - importing to 6.5 13
  - importing to manage earlier version sensors 16
  - isolating problems 124
- Maximum Linux 12
- Maximum Linux Apache 12
- Maximum Solaris 12
- Maximum Solaris Apache 12
- Maximum Windows 11
- Maximum Windows IIS 12
Index

Original Linux 12
Original Solaris 12
Original Windows 12
pre-defined 10
troubleshooting 124
user-defined 10
using to65policy.exe executable 15
policy	editing to add user-defined network signature 64
files 8
port number
blocking options 86
ports
25 and 80 90
Practical Programming in Tcl and Tk 99
pre-defined network signatures
TID setting 23
pre-defined policies 10
events monitored by 11
events not monitored by 11
pre-defined signatures 22
prerequisite
firecell signatures 85
fusion scripting 99
importing policies 13
monitoring log files 46
monitoring Unix syslogs 50
monitoring Windows event log 36
procedure scripts 103
Protect tab 22

S

SaveArray 107
SecureLogic See fusion scripting
sensors
deploying 5
server sensor
about 6
SetData 106
signatures 8
about 22
firecell 81
limitations 23
pre-defined 22
user-defined 23
user-defined BSM 67
user-defined network 61
SNMPv3 fusion scripting response 110
Solaris Basic Security Module
audit file 78
audit reduction 78
configuring audit management 78
creating user-defined signatures 69
specifying generic logs with wildcards 46
Store 106
syslog events
monitoring 50, 52

R

RealSecure 4
deploying 5
rearranging the order of firecell signatures 87
regular expression
monitoring for a string 47
Remove 106
RemoveArray 107
response scripts 103
RestoreArray 107
Retrieve 106
returning
true and false results in fusion scripts 113

T

tabs in the policy editor 22
Tcl
extensions 106
references for 99
tclproc1.log 116
troubleshooting 131
tutorial 99
technical support, Internet Security Systems xi
TID setting 23
to65policy.exe executable 15
transient
memory 100
variables 100
troubleshooting
policy problems 124
Tcl problems 131
trusted users
Index

monitoring 117

typographical conventions  x

u

UnsetData 106
user-defined
  binary log signatures  53
  network signature example 64
  network signature template 63
  network signatures 61
    supported 62
  policies 10
  signatures 23
user-defined binary log signatures
  creating 53
user-defined network signature
  configuring 65
user-defined signatures 33
users
  monitoring 117
  using
    firecell signatures 90
    wildcards to specify generic logs 46

v

validation script values 113
validation scripts 103
variables
  global 100
  transient 100
  types of 100

w

Web server

monitoring index.htm 116
Web site, Internet Security Systems  xi
Welch, Brent B. 99
wildcards 46
windows NT 4.0 start-up procedure 128
Windows NT Server 4.0 start-up procedure 128
wtmpx binary log file 53
wtmpx log monitoring
  disabling 54
  enabling 54

x

X-Press Update tab 22
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
</table>


Internet Security Systems, Inc. Software License Agreement

THIS SOFTWARE IS LICENSED, NOT SOLD. BY INSTALLING THIS SOFTWARE, YOU AGREE TO ALL OF THE PROVISIONS OF THIS SOFTWARE LICENSE AGREEMENT ("LICENSE"). IF YOU ARE NOT WILLING TO BE BOUND BY THIS LICENSE, RETURN ALL COPIES OF THE SOFTWARE AND LICENSE KEYS IN LIEU OF RETURN.

1. License - Upon payment of the applicable fees, Internet Security Systems, Inc. ("ISS") grants to you as the only end user ("Licensee") a nonexclusive and non-transferable license for the software associated herewith ("Software"). This license is subject to the terms and conditions of this License. Licensee may not copy, modify, rent, lease, disclose, distribute, re-sell, or sublicense the Software, except for associated license keys. This license shall automatically terminate if you breach any of the terms and conditions herein. Licensee may only use the Software to check systems or networked systems for vulnerabilities that are detectable by the Software.

2. Evaluation License - If ISS is providing Licensee with the Software and related documentation on an evaluation trial basis at no cost, such license Term is 30 days from issuance, unless otherwise agreed by the parties in writing. ISS reserves all intellectual property rights in the Software. Licensee agrees: (i) the Software is owned by ISS and/or its licensors, is a valuable trade secret of ISS, and is protected by copyright laws and international treaty provisions; (ii) to take all reasonable precautions to protect the Software from unauthorized reproduction, disclosure, copying, or use; (iii) not to modify, adapt, translate, reverse engineer, decompile, disassemble, or otherwise attempt to discover the source code of the Software; (iv) not to use ISS trademarks; (v) to reproduce all of ISS' and its licensors' copyright notices on any copies of the Software; and (vi) not to transfer, lease, assign, sublicense, or distribute the Software or make it available for time-sharing, service bureau, managed services offering, or resale.

3. Covenants - ISS reserves all intellectual property rights in the Software. Licensee agrees: (i) the Software is owned by ISS and/or its licensors, is a valuable trade secret of ISS, and is protected by copyright laws and international treaty provisions; (ii) to take all reasonable precautions to protect the Software from unauthorized reproduction, disclosure, copying, or use; (iii) not to modify, adapt, translate, reverse engineer, decompile, disassemble, or otherwise attempt to discover the source code of the Software; (iv) not to use ISS trademarks; (v) to reproduce all of ISS' and its licensors' copyright notices on any copies of the Software; and (vi) not to transfer, lease, assign, sublicense, or distribute the Software or make it available for time-sharing, service bureau, managed services offering, or resale.

4. Support and Maintenance - Maintenance for ISS Software includes technical support and electronic delivery to Licensee of software and security content updates as they become available. Licensee supports customers directly. Department of Commerce ("DoC") is not responsible for Software or for any Software products or services provided by ISS. DoC controls the use of Software and related documentation on an evaluation trial basis at no cost. Licensee may not use the Software to check systems or networked systems for vulnerabilities that are detectable by the Software. ISS reserves all intellectual property rights in the Software. Licensee agrees: (i) the Software is owned by ISS and/or its licensors, is a valuable trade secret of ISS, and is protected by copyright laws and international treaty provisions; (ii) to take all reasonable precautions to protect the Software from unauthorized reproduction, disclosure, copying, or use; (iii) not to modify, adapt, translate, reverse engineer, decompile, disassemble, or otherwise attempt to discover the source code of the Software; (iv) not to use ISS trademarks; (v) to reproduce all of ISS' and its licensors' copyright notices on any copies of the Software; and (vi) not to transfer, lease, assign, sublicense, or distribute the Software or make it available for time-sharing, service bureau, managed services offering, or resale.

5. Limited Warranty - The commencement date of this limited warranty is the date on which ISS furnishes to Licensee the license key for the Software. For a period of ninety (90) days after the commencement date or the Term if later, whichever is longer, if the Software is not in conformity with the applicable function specifications described in its then current documentation, Licensee may return the Software to ISS for repair, replacement or advanced exchange of the ISS hardware, and/or related Software security content updates for the pre-installed Software. During the period for which Licensee has paid the maintenance fees, ISS will provide maintenance in accordance with its prevailing Maintenance and Support Policy that is available at http://documents.iss.net/maintenance_policy.pdf.

6. Warranty Disclaimer - EXCEPT FOR THE LIMITED WARRANTY PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" AND ISS HEREBY DISCLAIMS ALL OTHER WARRANTIES AND IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, AND FREEDOM FROM VIRUSES. ISS EXPRESSLY DISCLAIMS ALL WARRANTIES TO THE EXTENT PERMITTED BY LAW. ISS SHALL HAVE NO LIABILITY UNDER ANY THEORY INCLUDING CONTRACT AND TORT (INCLUDING NEGLIGENCE AND STRICT PRODUCTS LIABILITY) FOR ANY SPECIAL, PUNITIVE, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF DATA, LOSS OF USE, OR COMPUTER HARDWARE MALFUNCTION, EVEN IF ISS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Proprietary Rights - ISS represents and warrants that ISS has the authority to license the rights to the Software that are granted herein. ISS shall defend and hold Licensee harmless from any final award of costs and damages against Licensee for any actions based on infringement of any U.S. copyright, trade secret, or patent as a result of the use or distribution of a current, unmodified version of this software, but only if ISS is promptly notified in writing of any such suit or claim. ISS shall have no liability under this License to Licensee for the Software or for any Software products or services provided by ISS to Licensee. In no event shall ISS be liable to Licensee under any theory including contract and tort (including negligence and strict products liability) for any special, punitive, indirect, incidental or consequential damages, including, without limitation, damages for loss of data, loss of use, or computer hardware malfunction, even if ISS has been advised of the possibility of such damages.

8. Limitation of Liability - ISS ENTIRE LIABILITY FOR MONETARY DAMAGES ARISING OUT OF THIS LICENSE SHALL BE LIMITED TO THE AMOUNT OF THE LICENSE FEES ACTUALLY PAID BY LICENSEE UNDER THIS LICENSE, PRORATED OVER A THREE YEAR TERM FROM THE DATE LICENSEE RECEIVED THE SOFTWARE. IN NO EVENT SHALL ISS BE LIABLE TO LICENSEE UNDER ANY THEORY INCLUDING CONTRACT AND TORT (INCLUDING NEGLIGENCE AND STRICT PRODUCTS LIABILITY) FOR ANY SPECIAL, PUNITIVE, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF DATA, LOSS OF USE, OR COMPUTER HARDWARE MALFUNCTION, EVEN IF ISS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

9. Termination - Licensee may terminate this License at any time by notifying ISS in writing. All rights granted under this License will terminate immediately, without prior notice from ISS, if the end user fails to comply with any terms or provisions of this License or if the end user has not paid all amounts due for the Software. Licensee may also terminate this License if such default has not been cured within ten (10) days following written notice of default to Licensee. Upon termination or expiration of a license for Software, Licensee shall cease all use of such Software, including Software pre-installed on ISS hardware, and destroy all copies of the Software and related documentation, including all copies of the source code of the Software; (iv) not to use ISS trademarks; (v) to reproduce all of ISS' and its licensors' copyright notices on any copies of the Software; and (vi) not to transfer, lease, assign, sublicense, or distribute the Software or make it available for time-sharing, service bureau, managed services offering, or resale.

10. General Provisions - This License, together with the identification of the Software, pricing and payment terms stated in the applicable Licensee purchase order as accepted by ISS, constitutes the entire understanding between the parties respecting its subject matter. No standard or form of agreement different from that established for the specific transaction between the parties shall be valid as an amendment to this License unless signed by an authorized officer of ISS. The terms, title, and manufacture of the Software furnished under this License is commercial computer software and any documentation is commercial computer technical data developed at private expense and is provided with RESTRICTED RIGHTS. Use, modification, reproduction, display, release, distribution or disclosure of this commercial computer software by the United States Government or its agencies is subject to the terms, conditions, and restrictions set forth in DFARS 252.227-7014 and FAR 52.227-14, as applicable. In the event of any conflict between the terms of this License and the terms of the DFARS or FAR, the terms of this License shall prevail.
12. Export and Import Controls; Use Restrictions - Licensee will not transfer, export, or reexport the Software, any related technology, or any direct product of either except in full compliance with the export controls administered by the United States and other countries and any applicable import and use restrictions. Licensee agrees that it will not export or reexport such items to anyone on the U.S. Treasury Department’s list of Specially Designated Nationals or the U.S. Commerce Department’s Denied Persons List or Entity List or such additional lists as may be issued by the U.S. Government from time to time, or to any country to which the United States has embargoed the export of goods, or for use with chemical or biological weapons, sensitive nuclear end-uses, or missiles. Licensee represents and warrants that it is not located in, under control of, or a national or resident of any such country or on any such list. Many ISS software products include encryption and export outside of the United States or Canada is strictly controlled by U.S. laws and regulations. Please contact ISS’ Customer Operations for export classification information relating to the Software (customer_ops@iss.net). Licensee understands that the foregoing obligations are U.S. legal requirements and agrees that they shall survive any term or termination of this License.

13. Authority - Because the Software is designed to test or monitor the security of computer network systems and may disclose or create problems in the operation of the systems tested, Licensee and the persons acting for Licensee represent and warrant that: (a) they are fully authorized by the Licensee and the owners of the computer network for which the Software is licensed to enter into this License and to obtain and operate the Software in order to test and monitor that computer network; (b) the Licensee and the owners of that computer network understand and accept the risks involved; and (c) the Licensee shall procure and use the Software in accordance with all applicable laws, regulations and rules.

14. Disclaimers - Licensee acknowledges that some of the Software is designed to test the security of computer networks and may disclose or create problems in the operation of the systems tested. Licensee further acknowledges that the Software is not fault tolerant and is not designed or intended for use in hazardous environments requiring fail-safe operation, including, but not limited to, aircraft navigation, air traffic control systems, weapon systems, life-support systems, nuclear facilities, or any other applications in which the failure of the Licensed Software could lead to death or personal injury, or severe physical or property damage. ISS disclaims any implied warranty of fitness for High Risk Use. Licensee accepts the risk associated with the foregoing disclaimers and hereby waives all rights, remedies, and causes of action against ISS and releases ISS from all liabilities arising therefrom.

15. Confidentiality - “Confidential Information” means all information proprietary to a party or its suppliers that is marked as confidential. Each party acknowledges that during the term of this Agreement, it will be exposed to Confidential Information of the other party. The obligations of the party (“Receiving Party”) which receives Confidential Information of the other party (“Disclosing Party”) with respect to any particular portion of the Disclosing Party’s Confidential Information shall not attach or shall terminate when any of the following occurs: (i) it was in the public domain or generally available to the public at the time of disclosure to the Receiving Party; (ii) it entered the public domain or became generally available to the public through no fault of the Receiving Party subsequent to the time of disclosure to the Receiving Party; (iii) it was or is furnished to the Receiving Party by a third party having the right to furnish it with no obligation of confidentiality to the Disclosing Party; or (iv) it was independently developed by the Receiving Party by individuals not having access to the Confidential Information of the Disclosing Party. Each party acknowledges that the use or disclosure of Confidential Information of the Disclosing Party in violation of this License could severely and irremediably damage the economic interests of the Disclosing Party. The Receiving Party agrees not to disclose or use any Confidential Information of the Disclosing Party in violation of this License and to use Confidential Information of the Disclosing Party solely for the purposes of this License. Upon demand by the Disclosing Party and, in any event, upon expiration or termination of this License, the Receiving Party shall return to the Disclosing Party all copies of the Disclosing Party’s Confidential Information in the Receiving Party’s possession or control and destroy all derivatives and other vestiges of the Disclosing Party’s Confidential Information obtained or created by the Disclosing Party. All Confidential Information of the Disclosing Party shall remain the exclusive property of the Disclosing Party.

16. Compliance - From time to time, ISS may request Licensee to provide a certification that the Licensed Software is being used in accordance with the terms of this License. If so requested, Licensee shall verify its compliance and deliver its certification within forty-five (45) days of the request. The certification shall state Licensee’s compliance or non-compliance, including the extent of any non-compliance, Licensee also, at any time, upon thirty (30) days prior written notice, at its own expense appoint a nationally recognized independent auditor, to whom Licensee has no reasonable objection, to audit and examine records at Licensee’s offices during normal business hours, solely for the purpose of confirming that Licensee’s use of the Licensed Software is in compliance with the terms of this License. ISS will use commercially reasonable efforts to conduct such audit conducted in a manner such that it will not unreasonably interfere with the normal business operations of Licensee. If such audit should reveal that use of the Licensed Software has been expanded beyond the scope of use and/or the number of Authorized Devices or Licensee certifies such non-compliance, ISS shall have the right to charge Licensee the applicable current list prices required to bring Licensee in compliance with its obligations hereunder with respect to its current use of the Licensed Software. In addition to the foregoing, ISS may pursue any other rights and remedies it may have at law, in equity or under this License.