



STONEGATE 5.2

IPS REFERENCE GUIDE

INTRUSION PREVENTION SYSTEM

STONESOFT

Secure Information Flow

Legal Information

End-User License Agreement

The use of the products described in these materials is subject to the then current end-user license agreement, which can be found at the Stonesoft website:

www.stonesoft.com/en/support/eula.html

Third Party Licenses

The StoneGate software includes several open source or third-party software packages. The appropriate software licensing information for those products at the Stonesoft website:

www.stonesoft.com/en/support/third_party_licenses.html

U.S. Government Acquisitions

If Licensee is acquiring the Software, including accompanying documentation on behalf of the U.S. Government, the following provisions apply. If the Software is supplied to the Department of Defense ("DoD"), the Software is subject to "Restricted Rights", as that term is defined in the DOD Supplement to the Federal Acquisition Regulations ("DFAR") in paragraph 252.227-7013(c) (1). If the Software is supplied to any unit or agency of the United States Government other than DOD, the Government's rights in the Software will be as defined in paragraph 52.227-19(c) (2) of the Federal Acquisition Regulations ("FAR"). Use, duplication, reproduction or disclosure by the Government is subject to such restrictions or successor provisions.

Product Export Restrictions

The products described in this document are subject to export control under the laws of Finland and the European Council Regulation (EC) N:o 1334/2000 of 22 June 2000 setting up a Community regime for the control of exports of dual-use items and technology (as amended). Thus, the export of this Stonesoft software in any manner is restricted and requires a license by the relevant authorities.

General Terms and Conditions of Support and Maintenance Services

The support and maintenance services for the products described in these materials are provided pursuant to the general terms for support and maintenance services and the related service description, which can be found at the Stonesoft website:

www.stonesoft.com/en/support/view_support_offering/terms/

Replacement Service

The instructions for replacement service can be found at the Stonesoft website:

www.stonesoft.com/en/support/view_support_offering/return_material_authorization/

Hardware Warranty

The appliances described in these materials have a limited hardware warranty. The terms of the hardware warranty can be found at the Stonesoft website:

www.stonesoft.com/en/support/view_support_offering/warranty_service/

Trademarks and Patents

The products described in these materials are protected by one or more of the following European and US patents: European Patent Nos. 1065844, 1189410, 1231538, 1259028, 1271283, 1289183, 1289202, 1304849, 1313290, 1326393, 1379046, 1330095, 131711, 1317937 and 1443729 and US Patent Nos. 6,650,621; 6,856,621; 6,885,633; 6,912,200; 6,996,573; 7,099,284; 7,127,739; 7,130,266; 7,130,305; 7,146,421; 7,162,737; 7,234,166; 7,260,843; 7,280,540; 7,302,480; 7,386,525; 7,406,534; 7,461,401; 7,721,084; and 7,739,727 and may be protected by other EU, US, or other patents, or pending applications. Stonesoft, the Stonesoft logo and StoneGate, are all trademarks or registered trademarks of Stonesoft Corporation. All other trademarks or registered trademarks are property of their respective owners.

Disclaimer

Although every precaution has been taken to prepare these materials, THESE MATERIALS ARE PROVIDED "AS-IS" and Stonesoft makes no warranty to the correctness of information and assumes no responsibility for errors, omissions, or resulting damages from the use of the information contained herein. All IP addresses in these materials were chosen at random and are used for illustrative purposes only.

Copyright © 2011 Stonesoft Corporation. All rights reserved. All specifications are subject to change.

TABLE OF CONTENTS

INTRODUCTION

CHAPTER 1

Using StoneGate Documentation	9
How to Use This Guide	10
Documentation Available	11
Product Documentation	11
Support Documentation	11
System Requirements	12
Contact Information	12
Licensing Issues	12
Technical Support	12
Your Comments	12
Other Queries	12

CHAPTER 2

Introduction to Intrusion Detection and Prevention	13
The Role of IDS and IPS	14
IDS and IPS Detection Methods	15
Challenges of Intrusion Detection	16
False Positives	16
False Negatives	16
Denial of Service	16
Overloaded Administrators	17

CHAPTER 3

Introduction to StoneGate IPS	19
The StoneGate Security Platform	20
StoneGate IPS System Components	21
Sensors	22
Analyzers	22
Sensor-Analyzers	22
Main Benefits of StoneGate IPS	23
Accuracy	23
Manageability	23
Scalability and High Availability	24
How StoneGate IPS Inspects Traffic	25
How StoneGate IPS Responds to Incidents	26

CHAPTER 4

StoneGate IPS Deployment	27
Overview to Sensor and Analyzer Deployment	28
Supported Platforms	28
General Deployment Guidelines	28
System Communication Volumes	29

Positioning Sensors	30
Internal Networks	31
DMZ Networks	32
External Networks	33
Positioning Analyzers	34
Positioning Management Center Components	34
High-Level Deployment Example	35
Example Large-Scale Installation	35
Example Medium Installation	36
Example Small-Scale Installation	37
Network Configuration Scenarios for Sensors	38
Deploying Sensors in IDS Configuration	38
Deploying Sensors in IPS Configuration	41
Cabling Guidelines	42
Cable Types	42
Speed and Duplex	43

SETTING UP STONEGATE IPS

CHAPTER 5

Sensor and Analyzer Configuration	47
Overview to Sensor and Analyzer Configuration	48
Configuration of Sensors and Analyzers	48
Heartbeat Network for Sensor Clusters	48
Network Interfaces	48
Configuration Workflow	49
Task 1: Create an Analyzer or Sensor Element	49
Task 2: Create Physical Interfaces	49
Task 3: Define VLAN Interfaces (<i>Sensor Elements Only</i>)	49
Task 4: Define Normal Interfaces	50
Task 5: Define Logical Interfaces (<i>Sensor Elements Only</i>)	51
Task 6: Define Capture Interfaces (<i>Sensor Elements Only</i>)	51
Task 7: Define Inline Interfaces (<i>Sensor Elements Only</i>)	52
Task 8: Select Interface Options	52
Task 9: Install the Sensor and Analyzer Engines	52
Task 10: Install an IPS Policy	52
Using Sensors and Analyzers	53
Contact Addresses for NATed Communications	53
Cluster Load Balancing	53

Processor Load Balancing	53
TCP Inspection Modes	54
Examples of Sensor and Analyzer Configuration	55
Configuring Capture Interfaces with SPAN	55
Configuring Capture Interfaces with TAP	56
Configuring Inline Interfaces	57
Setting up an Inline Serial Sensor Cluster	58

CHAPTER 6

Routing	59
Overview to Routing	60
Configuration of Routing	60
Default Elements	60
Configuration Workflow	60
Task 1: Add Router(s)	60
Task 2: Add Network(s)	60
Task 3: Refresh IPS Policy	60

TRAFFIC INSPECTION

CHAPTER 7

Situations	63
Overview to Situations	64
Configuration of Situations	64
Situation Contexts	65
Correlation Contexts	65
DoS Detection Contexts	66
Scan Detection Contexts	66
Protocol-Specific Contexts	66
File Contexts	67
System Contexts	67
Default Elements	67
Configuration Workflow	67
Task 1: Create a Situation Element	67
Task 2: Add a Context for the Situation	67
Task 3: Associate Tags and/or Situation Types with the Situation	68
Task 4: Associate the Situation with a Vulnerability	68
Using Situations	68
Examples of Custom Situations	69
Detecting the Use of Forbidden Software	69
Counting Events	69
Preventing Access to Forbidden Websites	70

CHAPTER 8

IPS Policies	71
Overview to IPS Policies	72
Policy Hierarchy	72
How StoneGate Examines the Packets	72
Configuration of Policy Elements	74
Default Elements	75
Configuration Workflow	76
Task 1: Create an IPS Template Policy	76
Task 2: Create an IPS Policy	77
Task 3: Create an IPS Sub-Policy	77
Task 4: Install the Policy	78
Using Policy Elements and Rules	79
Validating the Policy	79
Continue Rules	79
Policy Snapshots	79
Adding Comments to Rules	79
User Responses	80
Example of Policy Element Use	80
Restricting Administrator Editing Rights	80

CHAPTER 9

Ethernet Rules	83
Overview to Ethernet Rules	84
Configuration of Ethernet Rules	84
Considerations for Designing Ethernet Rules	86
Default Elements	86
Configuration Workflow	87
Task 1: Define the Source and Destination	87
Task 2: Define the Service	87
Task 3: Select the Action	87
Task 4: Select Logging Options	87
Using Ethernet Rules	88
Examples of Ethernet Rules	88
Logging Ethernet Protocol Use	88
Restricting the Use of Ethernet Protocols	89

CHAPTER 10

Access Rules	91
Overview to Access Rules	92
Configuration of Access Rules	92
Considerations for Designing Access Rules	94
Default Elements	94
Configuration Workflow	96
Task 1: Define the Source and Destination	96
Task 2: Define the Service	96

Task 3: Select the Action and Action Options	97	Using Protocol Agents	116
Task 4: Select Logging Options	98	FTP Agent	118
Task 5: Restrict the Time When the Rule Is Enforced	98	GRE Agent	118
Using Access Rules	98	H.323 Agent	118
Allowing System Communications	98	HTTP Agent	118
Configuring Default Settings for Several Rules	99	HTTPS Agent	118
Using Continue Rules to Set the Protocol	99	ICMP Agent	118
Rematching Tunneled Packets	100	IPv4 Agent	118
Using Aliases in Access Rules	100	IPv4 Encapsulation Agent	118
Examples of IPS Access Rules	101	IPv6 Agent	119
Exempting Traffic from Inspection	101	IPv6 Encapsulation Agent	119
Filtering Traffic on an Inline Sensor	101	MSRPC Agent	119
CHAPTER 11		NetBIOS Agent	119
Inspection Rules	103	Oracle Agent	119
Overview to Inspection Rules	104	Remote Shell (RSH) Agent	119
Configuration of Inspection Rules	105	Services in Firewall Agent	120
Verifying and Tuning Inspection	106	SMTP Agent	120
Considerations for Designing Inspection Rules	106	SSH Agent	120
Exception Rule Cells	108	SunRPC Agents	120
Default Elements	109	TCP Proxy Agent	120
Configuration Workflow	109	TFTP Agent	120
Task 1: Create a Policy	109	Examples of Protocol Agent Use	121
Task 2: Activate the Relevant Inspection Checks	110	Preventing Active Mode FTP	121
Task 3: Define the Exceptions	110	Logging URLs Accessed by Internal Users	121
Task 4: Eliminate False Positives	110	CHAPTER 13	
Task 5: Add Custom Inspection Checks	111	TLS Inspection	123
Using Inspection Rules	111	Overview to TLS Inspection	124
Setting Default Options for Several Inspection Rules	111	Configuration of TLS Inspection	125
Example of Inspection Rules	111	Default Elements	125
Eliminating a False Positive	111	Configuration Workflow	125
CHAPTER 12		Task 1: Create Server Protection Credentials Elements	125
Protocol Agents	113	Task 2: Create Client Protection Certificate Authority Elements	126
Overview to Protocol Agents	114	Task 3: Specify TLS Inspection Options in the Sensor Properties	126
Connection Handling	114	Task 4: Create an HTTPS Inspection Exceptions Element	126
Protocol Validation	114	Task 5: Create a Custom HTTPS Service	126
Configuration of Protocol Agents	115	Task 6: Create an Access Rule	126
Configuration Workflow	115	Using TLS Inspection	127
Task 1: Create a Custom Service with a Protocol Agent	115	Security Considerations	127
Task 2: Set Parameters for the Protocol Agent	115	Sensor Deployment	127
Task 3: Insert the Service in Access Rules	116	Examples of TLS Inspection	128
		Server Protection	128
		Client Protection	128

CHAPTER 14	
Web Filtering	129
Overview to Web Filtering	130
Configuration of Web Filtering	130
Default Elements	131
Configuration Workflow	131
Task 1: Prepare the Sensor.	131
Task 2: Create User Response Messages	131
Task 3: Blacklist/Whitelist Individual URLs.	131
Task 4: Configure Web Filtering Rules in the Policy	131
Examples of Web Filtering	132
Allowing a Blocked URL	132
CHAPTER 15	
Blacklisting	133
Overview to Blacklisting	134
Risks of Blacklisting	134
Whitelisting	134
Configuration of Blacklisting	135
Configuration Workflow	136
Task 1: Define Blacklisting in the Access Rules	136
Task 2: Define Analyzer-to-Firewall or Analyzer-to-Sensor Connections	136
Task 3: Define Exceptions in the IPS Policy	136
Using Blacklisting	137
Manual Blacklisting	137
Monitoring Blacklisting	137
Examples of Blacklisting	138
Blacklisting Traffic from a Specific IP Address Manually	138
Automatic Blacklisting with IPS	138

APPENDICES

APPENDIX A	
Command Line Tools	141
StoneGate-Specific Commands	142
General Tools	145
APPENDIX B	
Default Communication Ports	147
Management Center Ports	148
IPS Engine Ports	150
APPENDIX C	
Predefined Aliases	153
Pre-Defined User Aliases	154
System Aliases	154

APPENDIX D	
Situation Context Parameters	157
Port/Host Scan Detection Parameters	158
Correlation Context Parameters	159
Regular Expression Parameter	162
Other Context Parameters	162
APPENDIX E	
Regular Expression Syntax	163
Syntax for StoneGate Regular Expressions	164
Special Character Sequences	166
Pattern-Matching Modifiers	167
Bit Variable Extensions	168
Variable Expression Evaluation	170
Stream Operations	172
Other Expressions	173
System Variables	174
Independent Subexpressions	175
Parallel Matching Groups	176
APPENDIX F	
SNMP Traps and MIBs	177
APPENDIX G	
TCP/IP Protocol Headers	193
Internet Protocol (IP)	194
Internet Control Message Protocol (ICMP)	194
Transmission Control Protocol (TCP)	195
User Datagram Protocol (UDP)	195
APPENDIX H	
ASCII Character Codes	197
ASCII Character Codes	198
ASCII Control Codes	199
Glossary	201
Index	231

INTRODUCTION

In this section:

[Using StoneGate Documentation](#) - 9

[Introduction to Intrusion Detection and Prevention](#) - 13

[Introduction to StoneGate IPS](#) - 19

[StoneGate IPS Deployment](#) - 27

CHAPTER 1

USING STONEGATE DOCUMENTATION

Welcome to StoneGate™ IPS Intrusion Detection and Response System for Intelligent Analysis. This chapter describes how to use this Guide and related documentation. It also provides directions for obtaining technical support, and how to give feedback about the documentation.

The following sections are included:

- ▶ [How to Use This Guide](#) (page 10)
- ▶ [Documentation Available](#) (page 11)
- ▶ [Contact Information](#) (page 12)

How to Use This Guide

This *Reference Guide* provides information that helps administrators of StoneGate IPS installations to understand the system and its features. It provides high-level descriptions and examples of the configuration workflows.

This guide is divided into sections, which each include one to several chapters. The first section provides a general introduction to intrusion detection and StoneGate IPS. The sections that follow each include the chapters related to one feature area. The last section provides detailed reference information in tabular form, and some guideline information.

For other available documentation, see [Documentation Available](#) (page 11).

Typographical Conventions

The following conventions are used throughout the documentation:

Table 1.1 Typographical Conventions

Formatting	Informative Uses
User Interface text	Text you see in the User Interface (buttons, menus, etc.) and any other interaction with the user interface are in bold-face .
<i>References, terms</i>	Cross-references and first use of acronyms and terms are in <i>italics</i> .
Command line	File names, directories, and text displayed on the screen are monospaced.
User input	User input on screen is in monospaced bold-face .
<i>Command parameters</i>	Command parameter names are in <i>monospaced italics</i> .

We use the following ways to indicate important or additional information:



Note – Notes prevent commonly-made mistakes by pointing out important points.



Caution – Cautions prevent breaches of security, information loss, or system downtime. Cautions always contain critical information that you must observe.

Tip – Tips provide additional helpful information, such as alternative ways to complete steps.

Example Examples present a concrete scenario that clarifies the points made in the adjacent text.

Documentation Available

StoneGate technical documentation is divided into two main categories: [Product Documentation](#) and [Support Documentation](#). Each StoneGate product has a separate set of manuals.

Product Documentation

The table below lists the available product documentation.

Table 1.2 Product Documentation

Guide	Description
Reference Guide	Explains the operation and features of StoneGate comprehensively. Demonstrates the general workflow and provides example scenarios for each feature area. Available for StoneGate Management Center, Firewall/VPN, and StoneGate IPS.
Installation Guide	Instructions for planning, installing, and upgrading a StoneGate system. Available for StoneGate Management Center, Firewall/VPN, and IPS.
Online Help	Describes how to configure and manage the system step-by-step. Accessible through the Help menu and by using the Help button or the F1 key in any window or dialog. Available in the StoneGate Management Client and the StoneGate Web Portal. An HTML-based system is available in the StoneGate SSL VPN Administrator through help links and icons.
Administrator's Guide	Describes how to configure and manage the system step-by-step. Available as a combined guide for both StoneGate Firewall/VPN and StoneGate IPS, and as separate guides for StoneGate SSL VPN and StoneGate IPsec VPN Client.
User's Guide	Instructions for end-users. Available for the StoneGate IPsec VPN Client and the StoneGate Web Portal.
Appliance Installation Guide	Instructions for physically installing and maintaining StoneGate appliances (rack mounting, cabling, etc.). Available for all StoneGate hardware appliances.

PDF guides are available at <http://www.stonesoft.com/support/>. The *StoneGate Administrator's Guide*, and the *Reference Guides* and *Installation Guides* for StoneGate Management Center, Firewall/VPN, and StoneGate IPS are also available as PDFs on the Management Center CD-ROM.

Support Documentation

The StoneGate support documentation provides additional and late-breaking technical information. These technical documents support the StoneGate Guide books, for example, by giving further examples on specific configuration scenarios.

The latest StoneGate technical documentation is available on the Stonesoft website at <http://www.stonesoft.com/support/>.

System Requirements

The certified platforms for running StoneGate engine software can be found at the product pages at http://www.stonesoft.com/en/products/ips/Software_Solutions/.

The hardware and software requirements for the version of StoneGate you are running can also be found in the [Release Notes](#) available at the StoneGate Support Documentation pages.

Contact Information

For street addresses, phone numbers, and general information about StoneGate and Stonesoft Corporation, visit our website at <http://www.stonesoft.com/>.

Licensing Issues

You can view your current licenses at the License Center section of the Stonesoft website at <https://my.stonesoft.com/managelicense.do>.

For license-related queries, e-mail order@stonesoft.com.

Technical Support

Stonesoft offers global technical support services for Stonesoft's product families. For more information on technical support, visit the Support section at the Stonesoft website at <http://www.stonesoft.com/support/>.

Your Comments

We want to make our products fulfill your needs as well as possible. We are always pleased to receive any suggestions you may have for improvements.

- To comment on software and hardware products, e-mail feedback@stonesoft.com.
- To comment on the documentation, e-mail documentation@stonesoft.com.

Other Queries

For queries regarding other matters, e-mail info@stonesoft.com.

CHAPTER 2

INTRODUCTION TO INTRUSION DETECTION AND PREVENTION

This chapter introduces and discusses the underlying security principles of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) in general. In this chapter we will discuss what IDS and IPS are, how they are used, what they are capable of, as well as what their possible weaknesses are.

The following sections are included:

- ▶ [The Role of IDS and IPS](#) (page 14)
- ▶ [IDS and IPS Detection Methods](#) (page 15)
- ▶ [Challenges of Intrusion Detection](#) (page 16)

The Role of IDS and IPS

Intrusion detection systems (IDS) and intrusion prevention systems (IPS) are an important part of a comprehensive security solution. An IDS or IPS provides additional layers of defense to the existing security infrastructure by detecting and reacting to attacks and security breaches. Both the terms IDS and IPS are used here:

- An IDS system triggers an alarm whenever something defined as anomalous is detected on the monitored network.
- An IPS system can additionally block unwanted traffic.

StoneGate IPS can be configured either as an IDS or IPS or in *hybrid mode*, in which the system operates simultaneously in both IDS and IPS configuration.

To better understand the role of IDS/IPS, we can compare it to a firewall:

- A well-designed firewall stops everything that is not explicitly allowed, but IDS and IPS systems only react to unwanted traffic and allow everything else.
- As a rough generalization, firewalls mainly concentrate on the type, source, and destination of the traffic; the IDS/IPS system concentrates more on the information that is transferred.

The IDS/IPS system can look more deeply into traffic that has been allowed through the firewall or that is transferred between two hosts in the same network segment. For example, a connection from the Internet to a company's Web server is allowed through the firewall if it uses HTTP and proceeds according to the TCP protocol standards. However, this particular Web server contains an unpatched security vulnerability: an attacker in the know can include special code in a standard-looking URL. When this special code reaches the Web server, it causes an error condition that allows the attacker to execute some administrative commands. Even while the firewall allows this connection, an IPS system between the firewall and the Web server can stop it.

Generally speaking, there are two different classes of IDS/IPS: *host-based* and *network-based*. StoneGate is a network-based IDS and IPS system.

- Host IDS devices, such as Windows Defender, keep track of the system state changes on the hosts, alerting whenever something suspicious is detected. Host IDS is installed separately on each monitored host.
- Network-based IDS passively monitors network segments and notifies administrators when something out of the ordinary is detected. IDS is especially well suited for monitoring traffic that is sent and received within the same network segment so that it does not naturally flow through any single point of enforcement.
- Network-based IPS is located in the traffic path (known as *inline mode*) allowing an IPS device to actively filter out offending traffic. IPS allows direct blocking of attacks when you can identify a clear threat path. Most often, IPS devices are deployed in the traffic path between the Internet and internal network segments.

To an extent, the division between firewall and IPS systems is being blurred at the level of physical devices. A StoneGate firewall can be used to prevent some attacks, since StoneGate IPS product's inspection checks for a limited number of protocols have been integrated into the StoneGate Firewall/VPN product. On the other hand, StoneGate IPS can be used for access control, a traditional firewall task.

IDS and IPS Detection Methods

The following general techniques are employed in IDS and IPS systems:

- *Misuse detection* is based on signatures of well-known attacks. The monitored traffic is compared to these signatures and whenever a matching pattern is found, a response is triggered. The main limitation with this approach is that an ever increasing number of attacks must each be created a precise and unique signature that does not match other traffic. Any attack that does not have a specific fingerprint goes unnoticed. Also, the traffic stream may have been altered so that it cannot be compared directly with the signatures. *Traffic normalization* is needed to remove the ambiguities from the traffic before it can be compared with the signatures. Traffic normalization also makes it possible to detect *evasions*, attempts to bypass the IDS or IPS system, and to introduce attacks, exploits, or other malware into the network.
- *Vulnerability detection* is based on publicly known vulnerabilities that affect systems, network devices, and applications. Attackers may exploit vulnerabilities in many ways, for example, to gain unauthorized access to networks and to confidential information. There are several databases of currently known vulnerabilities (for example, Common Vulnerabilities and Exposures maintained by the MITRE Corporation). Vulnerabilities are closely connected to signatures of known attacks used in misuse detection. However, one vulnerability may be related to several known attacks and variations of attacks, which means that detecting vulnerabilities is more effective than inspecting traffic only for known attack patterns.
- *Anomaly detection* uses statistics to find abnormal network behavior. For example, a denial-of-service (DoS) attack is simply an abnormally high number of otherwise normal-looking connection attempts and anomaly detection is an effective way to detect it. However, this approach alone does not allow very comprehensive inspection, since it can be very hard to define just what is a normal and what is exceptional based on statistical information alone, especially considering that the ever-changing profile of network traffic makes ‘normal’ a moving target.
- *Protocol validation* checks whether traffic conforms to the protocol standards defined in the relevant *Request for Comments* (RFC). Because many current attacks are based on some sort of a protocol violation, any deviation from the defined protocol could be considered malicious. This approach is capable of detecting new types of attacks, also known as *zero-day attacks*. Unfortunately, sometimes there are also legitimate products that have been designed to violate the relevant standards either due to misinterpretation or purposely to circumvent a limitation.
- As an enhancement to protocol validation, *protocol anomaly detection* takes into account the fact that often attacks do not actually violate the protocol specifications as such, but take advantage of inaccuracies in standards. For example, a standard may not specify any limitations for the size of a given protocol field. In practice, there are practical limits to the size of the field in legitimate use, so an unusually large value can be considered to constitute a protocol anomaly. Naturally, this type of protocol validation also carries a risk of catching unusual but legitimate traffic.
- Protocol validation can also take the different states of connections into account, adding the principle of *stateful inspection* to the IDS/IPS. For example, what is normal—by the TCP protocol standards—for an opening SYN packet of a connection is not necessarily acceptable in an ACK packet within a TCP exchange. There is a low risk of disturbing any legitimate traffic, since communications must generally follow the rules of the basic protocols or they will either be refused or handled incorrectly by the recipient.

Given the advantages and disadvantages of the different methods, a comprehensive IDS or IPS system should be capable of more than one type of analysis. StoneGate IPS uses all of the methods described above to provide the best possible inspection coverage for the protected networks.

Challenges of Intrusion Detection

This page explains some of the negative effects IDS and IPS systems may produce, regardless of how well they work in principle. The key to avoiding most drawbacks is tuning the system to be relevant in the context of the installation environment. This is necessary in all environments. This becomes evident if you consider that even a real attempt to exploit a vulnerability may not be relevant when you consider the context of use: for example, an attempt to exploit a vulnerability in a Windows application is not a critical threat when launched at random against a Linux server.

False Positives

IPS can be an efficient tool that allows you to quickly receive notifications of events and even automatically block malicious traffic. If the patterns that the IPS is looking for are not accurate enough, also legitimate traffic may trigger alarms or be stopped.

False Negatives

False negatives occur when the IDS/IPS fails to react to malicious traffic. This can happen for many different reasons, for example:

- Skillful attackers may use techniques specifically designed to confuse the IDS/IPS.
- The inspection policies may be configured too permissive when trying to eliminate false positives.
- The IDS/IPS may not receive exactly the same traffic as the hosts on the monitored network due to network configuration issues.
- The lack of frequent updates may leave the IDS/IPS lacking the attack signatures that detect the latest publicly known exploits.

Denial of Service

IDS and IPS may be vulnerable to denial-of-service (DoS) attacks. If the sensors are busy processing a flood of packets, the attacker may succeed in slipping in some malicious traffic without the IDS/IPS being able to detect it.

In addition, an IDS/IPS configured with responses such as connection resetting or IP address blacklisting could be turned into an effective DoS tool by an attacker. If the administrators have not considered this possibility when designing the responses, an attacker could generate malicious traffic that makes the IDS/IPS stop legitimate traffic, for example, through IP address spoofing.

Overloaded Administrators

Some organizations, particularly MSSPs and other large organizations, will attract a high number of security incidents. Administrators may be engaged in investigating many events at the same time. If the IDS/IPS system is difficult to maintain and keep track of, the administrators will be overwhelmed and unable to respond to all incidents in a timely manner.

CHAPTER 3

INTRODUCTION TO STONEGATE IPS

This chapter gives you an overview of the StoneGate IPS system's architecture and how the system inspects traffic.

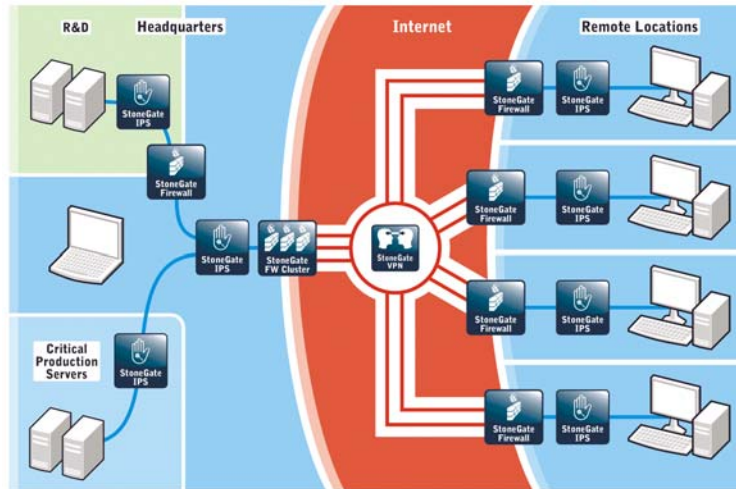
The following sections are included:

- ▶ [The StoneGate Security Platform](#) (page 20)
- ▶ [StoneGate IPS System Components](#) (page 21)
- ▶ [Main Benefits of StoneGate IPS](#) (page 23)
- ▶ [How StoneGate IPS Inspects Traffic](#) (page 25)
- ▶ [How StoneGate IPS Responds to Incidents](#) (page 26)

The StoneGate Security Platform

The StoneGate intrusion detection and prevention system is part of the StoneGate security platform, which is especially well-suited to complex and distributed network environments. In addition to IPS, the StoneGate security platform also provides clustered high availability firewalls and virtual private networking (VPNs).

Illustration 3.1 StoneGate Security Platform in Distributed Networks



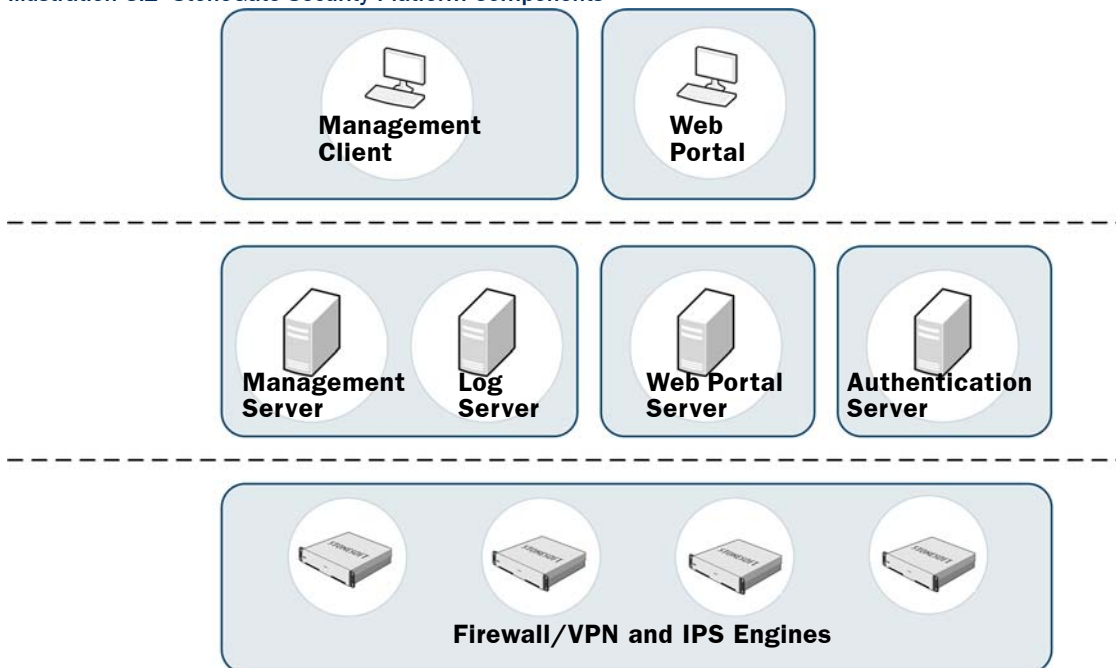
The configuration, monitoring, and control of the system is done through a centralized management system that provides a single point of contact for a large number of geographically distributed administrators. The unified management platform provides major benefits for organizations of all sizes:

- Interaction between the firewall and IPS components in the same system creates real security benefits by allowing automatic coordinated responses when a security threat is detected, providing instant blocking of unwanted traffic, and reducing the the need for immediate human intervention.
- Multiple administrators can log in at the same time to efficiently configure and monitor all StoneGate components. The system provides a single user interface that allows unified configuration, monitoring, and reporting of the whole StoneGate security platform with the same tools and within the same user session.
- The reuse of configuration information across components in the system allows you to avoid the laborious and error-prone duplicate work of configuring the same details for all components individually or exporting and importing the configurations between multiple separate systems.
- The system is designed to manage large installations and to be geographically distributed, so it is flexible and allows scaling up the existing components and adding new types of components to the system without sacrificing ease-of-use.

StoneGate IPS System Components

The StoneGate system components and their roles are illustrated below.

Illustration 3.2 StoneGate Security Platform Components



One StoneGate Management Center can manage a large number of both Firewall/VPN and IPS engines. The StoneGate distributed architecture allows deploying the system components effectively in different network environments. You can flexibly add, remove, and reposition StoneGate system components according to need.

The different system components are described in [Table 3.1](#).

Table 3.1 StoneGate System Components

Component	Description
Management Clients	Provide a user interface for configuring, controlling, and monitoring the system. Connects to the Management Server.
Management Servers	Store all configuration data and relay commands to the engines.
Log Servers	Store logs and perform alert escalation.
Web Portal Servers	Provide restricted viewing of configuration information, reports, and logs.
Authentication Servers	Provide user linking and user authentication services for end-user and administrator authentication.
Engines	Inspect and filter the traffic.

All communications between system components are authenticated and encrypted. The Sensors and Analyzers work independently according to their installed configuration, so even if the connections to the Management Center are cut, the IPS system continues its operation without interruption.

The IPS engines feature an integrated operating system (a specially hardened version of Linux). There is no need for separate operating system patches or upgrades; all software on the engines is upgraded during the IPS software upgrade.

Sensors

A StoneGate IPS Sensor is responsible for picking up and examining network traffic in real time. Based on this initial examination of packets and connections, each sensor creates event data that it sends to an analyzer for further processing. The sensors can also initiate immediate responses to any threats that they detect. Depending on how they are installed, sensors may also block traffic based on commands that other StoneGate components send.

Sensors can be flexibly scaled up to form clusters of up to 16 devices that work as a single virtual entity. Clustering provides additional performance and high availability for the traffic inspection service.

Analyzers

The StoneGate IPS analyzer correlates, processes, and further analyzes the event information that it receives from one or more sensors. This inspection is done out-of-band based on pre-analyzed data from the sensors instead of the actual network traffic. A single sensor may see only parts of a possible intrusion attempt, so it is the role of the analyzer to gather an overall picture of the connections in your network and to further analyze them for more complicated threats.

Analyzers also compress event information so that it is combined into fewer log entries that are based on security events rather than packets and connections, greatly clarifying the information provided to administrators.

Optionally, one or more backup analyzers can be installed for fault tolerance, allowing the inspection to continue and keeping the log stream flowing even if an analyzer goes down.

Sensor-Analyzers

The sensor and analyzer can also coexist in the same device. A combined sensor-analyzer works generally in the same way as separate components, with the following exceptions:

- It is not possible to send event data from other sensors to the analyzer component in a combined sensor-analyzer
- It is not possible to cluster sensor-analyzers.

Combined sensor-analyzers are especially well-suited as one-box IPS solutions for isolated, small network environments, such as remote branch offices.

Main Benefits of StoneGate IPS

Accuracy

Effective response to network security incidents requires the capability to recognize an enormous number of possible threats. On the other hand, the IPS system must not produce a high number of false alarms that engage the system administrators in needless investigations or automatically stop legitimate business communications.

To provide the best possible accuracy, StoneGate IPS provides multiple detection methods that complement each other. Attack signatures are supplemented with protocol-specific matching to produce accurate fingerprints of attacks. The observations on network traffic are not passed on to administrators directly, but instead collected together for further analysis and combined presentation.

What is considered as a serious threat against a crucial system in one environment may not be considered an event at all in some other network. There is no one set of traffic inspection policies that would work ideally in every environment, so StoneGate IPS provides detailed customization possibilities for the entire inspection process. The efficient configuration tools provide default policies that can be edited using drag-and-drop, while still allowing highly detailed controls for advanced configuration.

With accurate detection results, efforts can be concentrated on countering real threats instead of working on analyzing an endless stream of false alarms.

Manageability

Everyone recognizes the value of having the right tools for the right task, but unfortunately this is not always a given. StoneGate provides professional network administrators the tools they need to save time, reduce the likelihood of mistakes, and get the big picture of what is happening in the network.

While ease-of-use is one of the main goals for the product, StoneGate IPS does not achieve it by cutting the available features. The system provides extensive inspection process tuning possibilities, detailed information for monitoring, advanced automation, and tools for complete remote management of the IPS system (including all software upgrades). The distributed architecture allows components to be located on separate machines and in different networks, even in different countries and continents—and still be easily managed as a single system.

An easy-to-use system makes helps the administrators' concentrate on investigating the security threats instead of configuring the security systems.

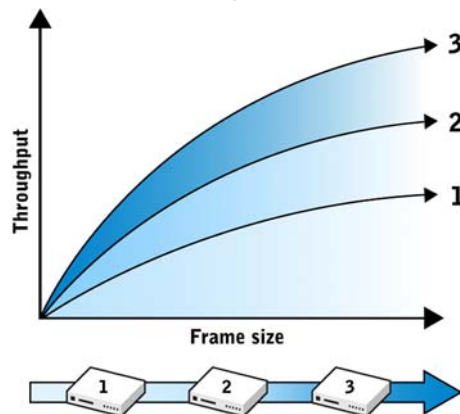
Scalability and High Availability

The constant introduction of new tools and services that rely on network access means that the volume of network traffic will keep growing over time even in companies that are not otherwise expanding. Also the IPS system must be scalable to meet the growing demands. Clustering answers this demand by combining devices together in a single, virtual entity.

Clustering also provides high availability for the intrusion detection services. If the IPS system is down when an attack is made, attackers have a head start, allowing them to penetrate deeper into the network and cover up their tracks. If a sensor in a cluster stops processing traffic, the inspection can switch over to the other remaining sensors transparently. This way, it is also possible to conduct online maintenance of individual sensors without corresponding interruptions in traffic inspection.

Sensor engines can be transparently clustered together into a single virtual entity in which the performance of each node contributes to the total throughput. Clustering allows new devices to be added flexibly as traffic volumes grow while retaining the existing equipment and configurations. Clustering improves performance by balancing the load intelligently between the clustered devices ([Illustration 3.3](#)).

Illustration 3.3 The Performance Benefits of Clustering



Analyzers cannot be clustered, but to provide high availability for the services they provide, it is possible to add a backup analyzer that is activated when a corresponding active analyzer becomes unavailable.

Optional high availability measures are also available for the StoneGate Management Center, see the *StoneGate Management Center Reference Guide* for more information.

Scalability and high availability ensure that the system can adapt to growing needs, simplifies planned maintenance, and protects against hardware failure.

How StoneGate IPS Inspects Traffic

Illustration 3.4 StoneGate IPS Traffic Inspection

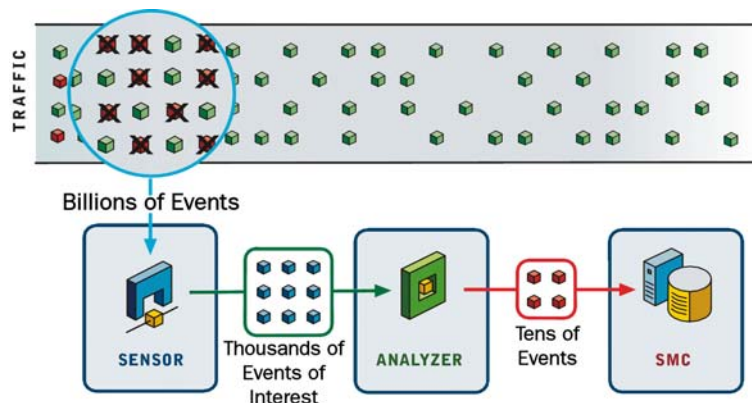


Illustration 3.4 shows the traffic inspection process in StoneGate IPS. First, the sensor inspects the network traffic for any anomalies. If the sensor detects ambiguities in the traffic, it normalizes the traffic before inspecting it further. Traffic normalization also helps the sensor in detecting attempts to use evasion techniques (for example, packet fragmentation, TCP segmentation, or combinations of evasions) to bypass inspection altogether. Then, the sensor informs the analyzer about interesting events. The analyzer further processes these events to find interesting patterns among the observations of one or more sensors. The analyzer adds its own findings to the information stream, but also combines related events together and discards unnecessary events. As a result, the information that the analyzer forwards to the StoneGate Management Center (SMC) for administrators to view is clear and concise.

In this process, known attacks are detected through attack signatures that are augmented with protocol awareness to form powerful attack *fingerprints*. Protocol awareness decreases the number of false positives compared to simple signatures. Each pattern is applied only to the correct type of traffic; for example, an attack that uses HTTP can be detected when the pattern is seen in HTTP traffic, but does not falsely match an e-mail message header transported over SMTP.

While fingerprinting accurately detects known attacks, it does not detect attacks that are not yet known. StoneGate IPS provides two types of anomaly detection to complement fingerprinting: *protocol analysis* and *statistical anomaly detection*.

- *Protocol analysis* identifies violations in network communications, such as unexpected data, wrong connection states, and additional or invalid characters. Detecting such violations is useful because many attacks purposely violate standards to trigger abnormal operating responses in vulnerable target systems.
- *Statistical anomaly detection* gathers traffic statistics to detect events such as slow scans, unusual number of connections, and so on. This method tracks patterns based on frequency and sequence of events, or the occurrence of sets of related events within a period of time. For example, many connection attempts from one host to many ports and IP addresses is certainly a network scan of some kind.

How StoneGate IPS Responds to Incidents

There are various automatic responses that StoneGate IPS can take when it detects traffic of interest ranging from logging the connection to actively filtering out the traffic.

Several responses are available on all sensors and analyzers:

- As the mildest response, an event can be logged. The log entries can be used, for example, for generating statistical reports. This may be appropriate, for example, for tracking trends in normal network traffic patterns.
- A step up from a log entry is to generate an alert entry that can be escalated to administrators through multiple configurable alert channels including e-mail, mobile phone text messaging (SMS), and SNMP, in addition to being used like log entries.
- Additionally, logs and alerts can record the full packet headers and data payload for further analysis.



Note – Storing or viewing the packets' payload may be illegal in some jurisdictions due to laws related to the privacy of communications.

- Blacklisting sends information about the traffic to one or more StoneGate firewalls or IPS-configured sensors that maintain a blacklist. Connections that match the blacklist are mainly stopped (depending on the enforcing component's policy).

Sensors have also some additional response methods. The availability of these responses depends on the sensor's physical configuration, both of which can be used on the same sensor at the same time (on different network interfaces):

- In an **IDS** (intrusion detection system) installation, external equipment duplicate the traffic flow for inspection, and the sensors just "listen in". The sensors do not have direct control over the traffic flow, but they can respond to selected threats by sending packets that reset the connections. IDS-only sensors can send blacklisting requests to other sensors and StoneGate firewalls, but they cannot enforce blacklisting requests from other components.
- In an **IPS** (intrusion protection system) installation, sensors are installed inline, so that traffic is routed directly through the Sensor. The Sensor has full control over the traffic flow and can be used to automatically block any traffic. Inline sensors can also enforce blacklisting commands received from other components. Fail-open network cards can be used to ensure traffic flow is not disrupted when the Sensor is offline. Inline sensors also provide transparent access control (TAC) and logging for any Ethernet traffic (layer 2).

CHAPTER 4

STONEGATE IPS DEPLOYMENT

This chapter provides general guidelines for the StoneGate IPS system deployment. It also illustrates some typical deployment scenarios.

The following sections are included:

- ▶ [Overview to Sensor and Analyzer Deployment](#) (page 28)
- ▶ [Positioning Sensors](#) (page 30)
- ▶ [Positioning Analyzers](#) (page 34)
- ▶ [Positioning Management Center Components](#) (page 34)
- ▶ [High-Level Deployment Example](#) (page 35)
- ▶ [Network Configuration Scenarios for Sensors](#) (page 38)
- ▶ [Cabling Guidelines](#) (page 42)

Overview to Sensor and Analyzer Deployment

Supported Platforms

Sensors and analyzers can be run on the following general types of platforms:

- Purpose-built StoneGate IPS appliances.
- Standard Intel-compatible servers. Search for the version-specific *Hardware Requirements* in the technical documentation search at <http://www.stonesoft.com/en/support/>.
- As a VMware virtual host. More information can be found in the StoneGate Technical Documentation database, see document 3210: [Installing and Activating StoneGate IPS in the VMWare ESX Server](#).

The sensors and analyzers have an integrated, hardened Linux operating system that is always a part of the StoneGate engine software, eliminating the need for separate operating system installation, configuration, and patching.

General Deployment Guidelines

[Table 4.1](#) summarizes the general deployment guidelines for StoneGate IPS and the Management Center. Naturally, there are valid reasons to make exceptions to these general rules depending on the actual network environment.

Table 4.1 General Guidelines for StoneGate IPS Deployment

System Component	General Guidelines
Management Server	Position on a central site where it is physically accessible to the administrators responsible for maintaining its operation.
Log Servers	Position the Log Servers centrally and/or locally on sites as needed based on log data volume, administrative responsibilities, etc.
Management Clients	Management Clients can be used from any location that has network access to the Management Server and the Log Servers.
Sensors	Position sensor(s) at each location so that all networks are covered. Sensors can be clustered. Functionally, the sensor cluster is equal to a single high-performance sensor. Cluster deployment involves setting up a heartbeat link between the sensors that allows the devices to track each others' operating status and agree on the division of work.
Analyzers	Position an analyzer at each location with the sensor(s).
Sensor-Analyzer	Position combined sensor-analyzers as a one-box solution in low-traffic environments, for example, at remote offices.

System Communication Volumes

One consideration for deploying the system components is the amount of data that flows between the components. The logs are generally the only type of system communications that produce a considerable volume of traffic. Other system communications require only minimal bandwidth (less than 1 Kbit/s on average).

The log data volume varies considerably based on the network environment, types of traffic, the rules in the policy, and many other factors. On average, a log entry is approximately 750 bytes in size. An analyzer can typically compress the volume down to 1/10th or even 1/20th of what is received from sensors. The volumes may be larger before you fine-tune the policies for the operating environment.

Table 4.2 provides rough estimates on the average transferred event data rates as they flow from the sensor through the analyzer to the Log Server.



Note – The event data transfer rates vary considerably depending on the type of traffic and generally follow the volume of inspected traffic.

Table 4.2 Estimated Average Rates of Event Data Transfer with the Strict Policy and System Policy

Captured Traffic	Average Event Transfer from Sensor to Analyzer	Average Event Transfer from Analyzer to Log Server
less than 2 Mbit/s	From ~1 Kbit/s for a protected office network to ~15 Kbit/s for an open Internet network.	Usually less than 1 Kbit/s.
2 – 10 Mbit/s	From less than 10 Kbit/s for a protected office network to ~60 Kbit/s for an open Internet network.	From less than 1 Kbit/s for a protected office network to less than 5 Kbit/s for an open Internet network.
10 – 100 Mbit/s	From ~60 Kbit/s for a protected office network to ~600 Kbit/s for an open Internet network.	From ~5 Kbit/s for a protected office network to ~30 Kbit/s for an open Internet network.
100 – 1000 Mbit/s	From ~600 Kbit/s for a protected office network to ~3000 Kbit/s for an open Internet network.	From ~50 Kbit/s for a protected office network to ~150 Kbit/s for an open Internet network.

Sensors can also record selected portions of traffic as configured. The recordings are transferred directly from the sensor to the Log Server. The traffic volume depends on the frequency and the size of the recordings.

For security, all connections between the system components are encrypted and authenticated using *Secure Sockets Layer/Transport Layer Security (SSL/TLS)*.

Positioning Sensors

Sensors pick up passing network traffic for inspection in real time. The positioning of sensors is the most critical part of the deployment. Each sensor can inspect the network traffic of one or more network segments in IDS and IPS configurations.

- In IDS (capture) mode, the sensors listen to network traffic that is replicated to the sensor through port mirroring (switch SPAN ports) or through dedicated network TAP devices. Hubs can also be used if network performance is not an issue.
- In IPS (inline) mode, the sensor actively filters traffic. The sensor is connected as a “smart cable” between two network devices, such as routers. The sensor itself does not route traffic: packets enter through one port, are inspected, and exit through the other port that makes up the inline pair. Fail-open network cards are recommended to allow network connectivity when the sensor is offline. Inline sensors can also transparently segment networks and control network access.
- The same sensor can be used for both IPS and IDS operation simultaneously. For example, a sensor can be deployed inline to examine traffic traversing from one network to another and additionally capture traffic that stays within each network.

The following can be used as general criteria for deciding where sensors are installed:

- Determine the critical assets to be protected and the potential attack paths.
- Determine the most suitable locations along the attack path for detecting and responding to attack attempts in order to protect the assets.
- Determine the volume and profile of traffic to be inspected at each location.

Illustration 4.1 Example of Positioning Sensors In Different Network Segments

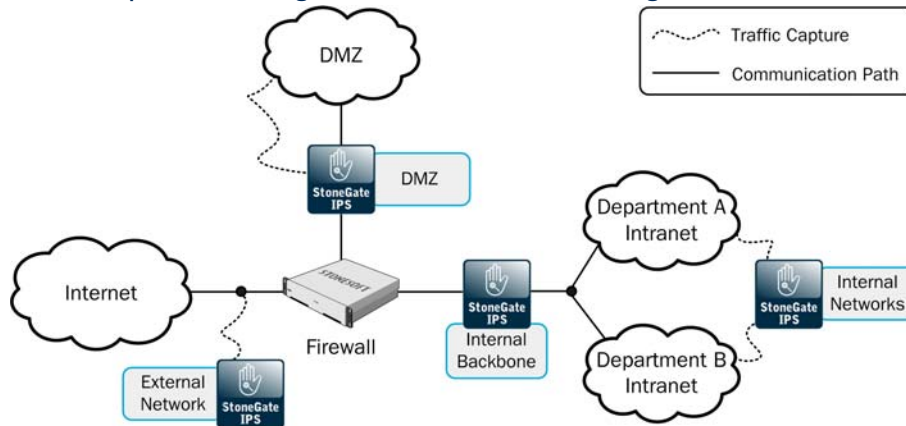


Illustration 4.1 outlines common sensor deployment scenarios; in general internal networks, in DMZ networks, and in the external network. The next few pages of this guide explain these scenarios in more detail. Sensors are not necessarily needed at each of these points in all environments. Also, one sensor could cover several or even all scenarios simultaneously if the physical setup makes it practical.

Internal Networks

In internal networks, access is generally quite permissive for purely internal communications, but there are strict controls at the perimeter firewall that separates the internal network from public networks. Inbound traffic from public networks is generally forbidden with few exceptions.

Table 4.3 Internal Network Considerations for Sensors

	Description	Implications on Sensors
Main purpose	Network services and connectivity for authorized end-users. Back-end servers that serve other networks and user groups.	IPS sensors can be used within and between different internal networks and for strengthening the perimeter defense with additional layers of inspection.
Hosts	Mixed environment consisting of servers, laptops, desktops, network printers, copiers, etc.	Sensors can control access between internal hosts that are not controlled by other devices. Connections between internal and external networks are of particular interest for inspection.
Users	Authorized personnel. Access in and out of the network is controlled by a firewall.	End-user-controlled devices can be distinguished from other hosts to create more accurate and fine-grained rules.
Traffic volume	Varies from low to high. Grows highest at network choke-points in large environments.	Installation at network choke-points where traffic levels are high requires high-performance hardware. Clustering and load-balancing can be applied to increase performance and provide high availability in critical locations.
Traffic type	Diverse with a large number of different applications communicating within and in/out of the network.	A wide range of permitted applications means that the policy has a wide scope. Access control and inspection can be fine-tuned based on the security levels of the different network segments. TLS inspection can be activated to inspect SSL/TLS encrypted traffic. The IPS can also detect certain prohibited applications.
Network security	A “trusted network” where the users and the traffic are considered to be authorized.	The primary line of defense is at the perimeter, but it is possible that authorized users in the trusted network become willingly or accidentally involved in a security incident.

DMZ Networks

DMZ networks (demilitarized zone networks, also known as perimeter networks) allow inbound access to a wide range of users, but are quite unified environments in terms of devices. The services offered are limited in number as well and their allowed usage is often quite strictly defined.

Table 4.4 DMZ Considerations for Sensors

	Description	Implications on Sensors
Main purpose	DMZs provide a limited number of services for external users. The services are often business-critical and open for public access.	DMZs are a tempting target for attacks because of their accessibility, importance, and visibility. Sensors provide a crucial line of defense both at the perimeter and within the DMZs.
Hosts	Often a uniform environment consisting mainly of servers. There are usually no end-user workstations.	Most sources are not trusted and IP address spoofing is a possibility. Internal networks may be considered more trustworthy if there is a firewall that prevents IP address spoofing.
Users	Most services are public, but some services may also be offered to specific users. Administrators have wider privileges.	If the users can be reliably recognized, allowed and forbidden activities can be specified in great detail for each type of access.
Traffic volume	Low to medium, generally the full bandwidth of all Internet links combined (shared with other local networks). Traffic to other local networks may be high in volume.	Hardware requirements vary greatly depending on the environment. Clustering allows flexible adjustments to the inspection performance.
Traffic type	Rather uniform traffic, with only specific applications and servers communicating within, into, and out of the networks.	The limited, well-defined set of protocols and applications means inspection can be tuned in great detail. If servers provide HTTPS services, decrypting the traffic for inspection may require heavy processing.
Network security	A network between the trusted and untrusted security zones allowing access for authorized and public use.	External access to services makes the servers in a DMZ a tempting target for attacks. Connections between the DMZs and other networks facilitate further attacks.

External Networks

External networks (the networks in front of the firewalls) provide the infrastructure to connect the “trusted” internal networks to the “untrusted” public networks. There is very little filtering between the external networks and the public Internet.

Table 4.5 External Network Considerations for Sensors

	Description	Implications on Sensors
Main purpose	Connectivity between the protected and public networks.	Traffic inspection can provide information about any attack attempts that originate in the public networks.
Hosts	Only equipment that needs to be directly connected to the public network, such as routers and the firewall. Possibly some secured hosts that need to be directly accessible from the public network.	Networking devices are generally very secure, but may still be vulnerable to some attacks that can be detected. In many cases, the communicating hosts are unknown. IP address spoofing is a possibility and the firewall's NAT configuration often hides internal network addresses.
Users	Access to this network is open, but local access to the hosts is usually restricted to the administrative staff only.	Administrative actions from unexpected sources may be of interest, but otherwise it is usually difficult to reliably classify users at this point.
Traffic volume	Varies from low to high, generally the full bandwidth of all Internet links combined. The volume of local traffic is usually low.	Hardware requirements may vary depending on the environment. Clustering allows flexible adjustments to the inspection performance.
Traffic type	Any type of traffic may be encountered, especially in incoming traffic, although some filtering may be done by the Internet service provider.	It is usually sensible to tune the sensor policy to be very specific rather than apply a broad policy such as the default Strict Policy or System Policy. Sensors in this insecure network should not decrypt HTTPS traffic for inspection. VPN tunnels are often terminated at the firewall, so that part of the traffic cannot be inspected due to encryption.
Network security	There is little or no access control to this network from the Internet. The hosts residing in this network should all be security-hardened and actively patched against known vulnerabilities.	The sensor also inspects traffic that is blocked by the firewall. Some detected events may not affect the networks protected by the firewall. Inspection serves mainly informational purposes.

Positioning Analyzers

You can position the analyzers quite freely, since analyzers do not examine the network traffic directly. However, the StoneGate IPS sensors and analyzers cooperate closely to inspect the captured network traffic, so we recommend positioning an analyzer at each geographical site that has sensors. There are some important reasons for this:

- The volume of transferred event data can be considerably larger between the sensor and analyzer than from the analyzer and sensor to other system components (see [System Communication Volumes](#) (page 29)).
- If the sensor to analyzer network connectivity is lost, but traffic keeps flowing through the sensor, the administrators are not alerted of any findings, and some events that the sensor cannot detect alone may not be noticed until long after they occur. The sensor may also eventually run out of space for storing the undelivered event data.
- Sensor to analyzer communications can be time-critical. For example, the analyzer may detect an attack pattern in the event stream and blacklist the attackers on the inline sensor. The longer it takes to transmit the event stream from the sensor to the analyzer, and the blacklist request from the analyzer to the sensor, the longer the attackers can continue with their activities.

A combined sensor-analyzer may be the best option for smaller environments as a one-box solution. It is also possible to set up the analyzer at a remote location in relation to the sensor as long as the factors listed above are considered.

Each analyzer can receive data from one or more single sensors or sensor clusters. However, combined sensor-analyzers cannot receive events from any external sensors.

Positioning Management Center Components

The StoneGate Management Center (SMC) consists of the Management Server and the necessary number of Log Servers and Management Clients. These can be positioned flexibly according to need. Each Management Server can remotely manage a high number of both StoneGate Firewall and IPS components. Optionally, you can install one or more Management Servers as backups and one or more Web Portal Servers for view-only access to the system. Sensors and analyzers are managed through the Management Server, so the Management Client never connects directly to the sensors or analyzers.

The general Management Center deployment steps are as follows:

1. Position the Management Server at a central location where it can access all the other components and so that the Management Clients can connect to it.
2. Position Log Servers centrally and/or locally based on the log data volume requirements.
3. Position Management Clients freely where they are needed.

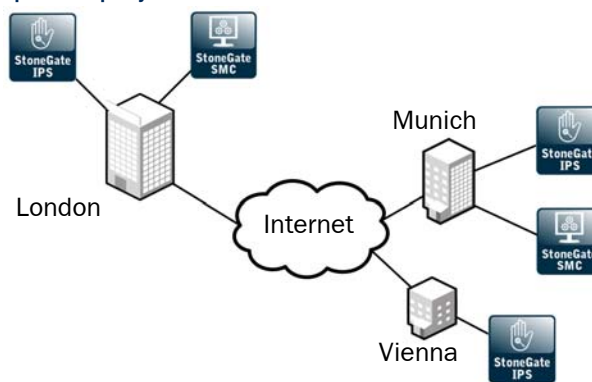
The SMC deployment considerations are described in detail in the *StoneGate Management Center Reference Guide*.

High-Level Deployment Example

This section shows one way to deploy StoneGate IPS in an organization. The scenario presented here is not meant to be representative of a typical installation. The main focus here is to highlight some of the criteria that can be used in planning deployment. The example covers considerations that affect most installations, but is not an exhaustive list of all factors that you may need to consider. Therefore, the IPS system could be deployed in alternative ways even in this example scenario, depending on issues that are not covered here, such as the physical layout of the individual local networks, the hardware available, and budget constraints.

This example explains the IPS deployment at a company that has three offices: headquarters in London, a branch office in Munich and a small satellite office in Vienna.

Illustration 4.2 The Example Company's Networks

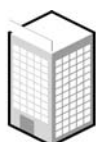


As shown in [Illustration 4.2](#), all offices have StoneGate IPS components. There are also Management Center components at the two larger sites. The example company has some critical assets to protect and some of the networks experience a heavy traffic load. With this in mind, the example company has decided on a high availability solution for most locations and acquired the following StoneGate IPS system components:

- 3 sensor clusters
- 3 analyzers
- 1 sensor-analyzer
- 1 Management Server
- 2 Log Servers

The next few pages show the placement of these components office by office and explain the criteria the example company used to select them.

Example Large-Scale Installation

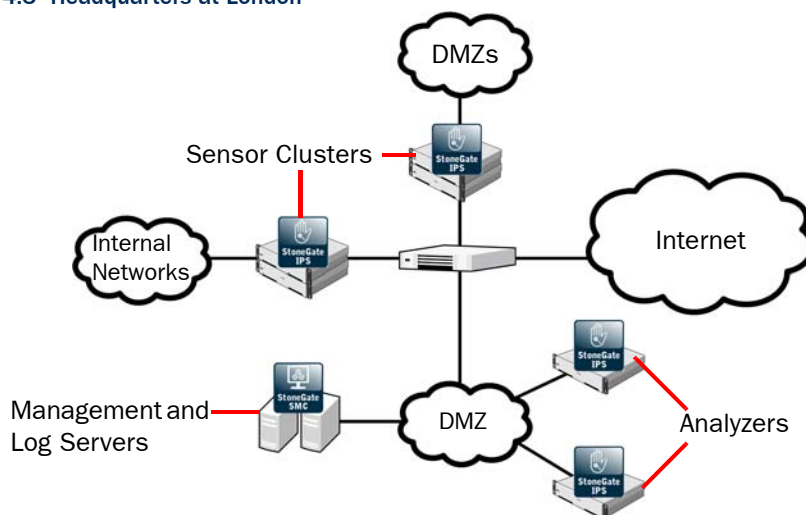


London

The example company's main office at London has a large number of end-users and servers. The servers host nearly all of the company's external services and receive a high volume of traffic. The large end-user base generates a high volume of network traffic as well. There are many different applications and protocols in use, resulting in a very diverse traffic pattern. The most important asset that the company wants to protect at its headquarters are the Web servers hosting the company's online store.

Since this is the main office, the main system administrators work at this site.

Illustration 4.3 Headquarters at London



In this case, the company has made the following decisions:

- Since most administrators are at this site, the Management Server that controls the whole distributed system is located here.
- There are many administrators and components, so there is also a Log Server here.
- Several DMZs for different services handle a high total volume of traffic. Part of the traffic is encrypted HTTPS, which uses significant processing power to decrypt for inspection. As the overall load is heavy, the company decided to protect the DMZs using a dedicated high-performance sensor that is clustered to ensure that these important communications are inspected even if one of the sensors is offline.
- A separate sensor is installed to protect the diverse high-volume communications of the internal networks. This sensor is also a cluster to ensure smooth operation in these business-critical networks.
- Two analyzers are installed at this office to provide a backup analyzer in case one analyzer goes offline. To better utilize both analyzers, each sensor cluster sends data to a different analyzer as long as both analyzers are operational.
- The Management Center and analyzers are placed in a dedicated DMZ for security.

Example Medium Installation

The example company's branch office at Munich has a moderate number of end-user clients. Although some services are only offered at the London headquarters and used remotely through a VPN, there are still quite many local servers, mostly for internal and partner use. There are also some administrators at this location who are responsible for the daily upkeep of the infrastructure at this office and the small satellite office in Vienna.

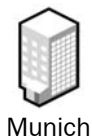
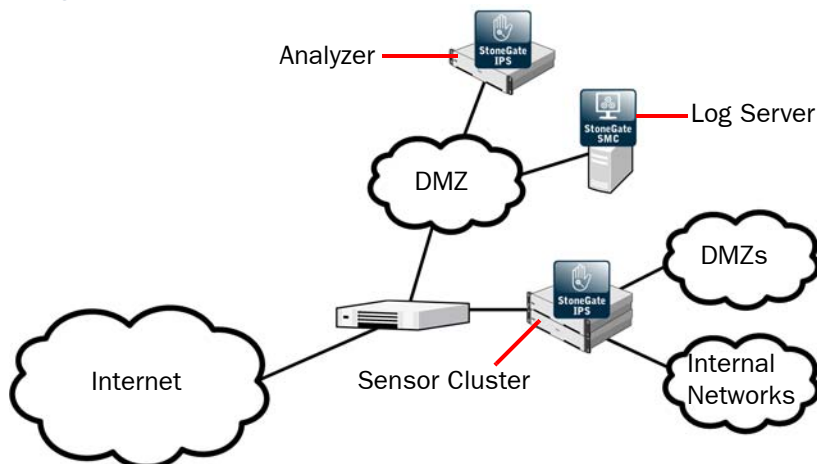


Illustration 4.4 Large Branch Office in Munich



In this case, the company has made the following decisions:

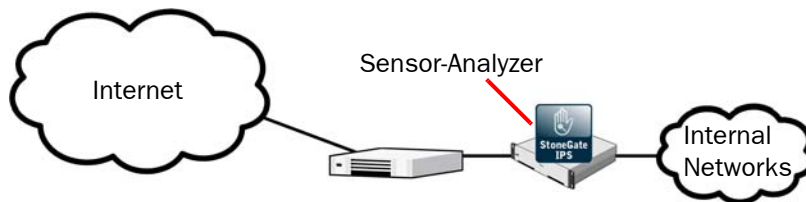
- Since there are administrators who browse logs extensively at this site, there is a dedicated Log Server here.
- One sensor is installed to protect the communications of both the internal networks and a DMZ that supports partner access. This sensor is also a cluster to ensure smooth operation.
- An analyzer is installed at this site for efficient and fast operation. There is no backup analyzer, since the company has decided that the chance of failure is not high enough to warrant a local backup analyzer in this case, since the Munich networks are not as business-critical as those in London.
- The Log Server and analyzer are placed in a dedicated DMZ for security.

Example Small-Scale Installation



The example company's small satellite office at Vienna has a relatively low number of end-user clients, and there are no servers of any major significance. Users rely mostly on the services at the Munich office, which they access through a VPN. Additionally, the users have direct Internet access for general Web browsing. There are no local administrators; systems are managed remotely by administrators in Munich.

Illustration 4.5 Small Satellite Office in Vienna



In this case, the company has decided to install a combined sensor-analyzer at the office to inspect the relatively low-volume traffic that the end-users' Internet and VPN-bound communications generate. Since there are no local administrators and the traffic volumes are low, the logs are sent to the Munich Log Server, making it quicker and easier for the responsible administrators there to view and manage the data.

Network Configuration Scenarios for Sensors

This section presents the most important scenarios for deploying a single or clustered sensor in different network configurations.

Deploying Sensors in IDS Configuration

One of the options in IDS mode is to use network TAP devices that copy packets for the sensors. In a cluster, all sensors must receive all packets and the sensors agree over the heartbeat link which sensor inspects which connections.

Illustration 4.6 Single Sensor (Left) and Sensor Cluster Capturing with Network TAPs

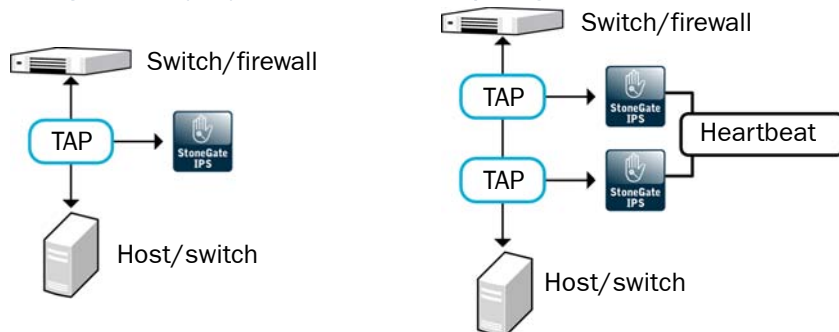


Illustration 4.7 Single Sensor Capturing with Network TAP and an Interface for Sending Resets

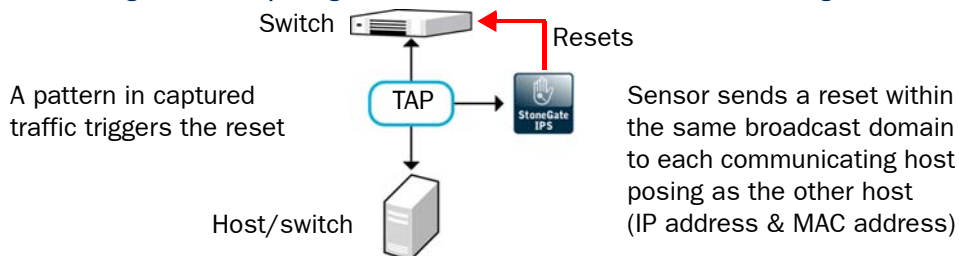
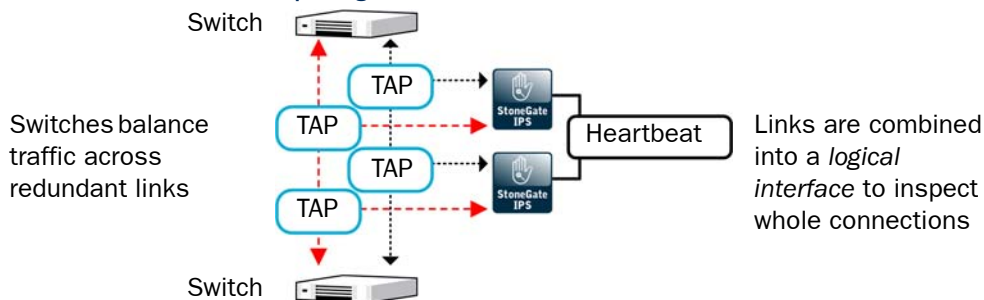
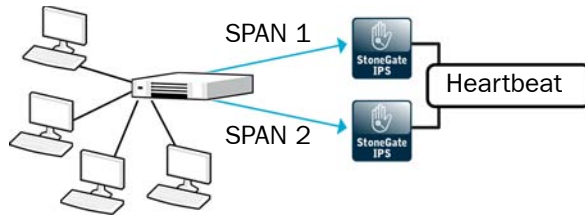


Illustration 4.8 Sensor Cluster Capturing with Network TAPs on a Redundant Link



Packets can also be duplicated for inspection through a SPAN or mirror port on a switch/router. In a sensor cluster, each node must be connected to a SPAN or mirror port of its own. A hub can be used to achieve a similar configuration when the low performance of a hub is not an issue (for example, in a basic testing environment), but hubs are generally not recommended.

Illustration 4.9 Sensor Cluster Capturing with SPAN/Mirror Ports



A sensor can be deployed alongside a clustered StoneGate firewall. In this configuration, the sensor is in the same broadcast domain as the firewall.

Illustration 4.10 Sensor Connected to Network TAPs Alongside a Dispatch-Mode Firewall Cluster

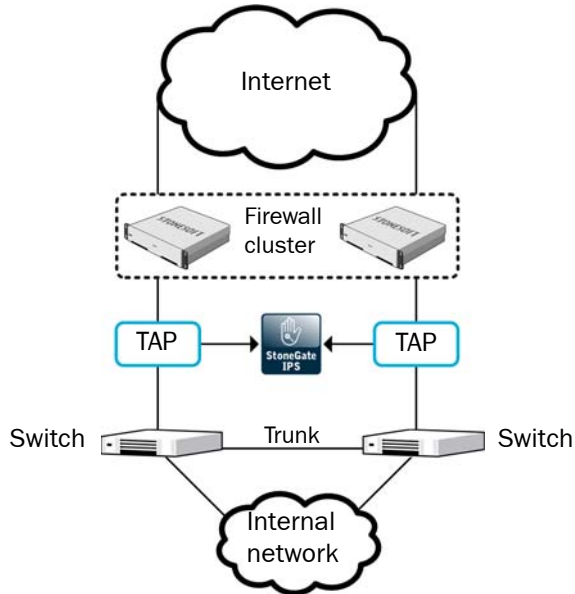
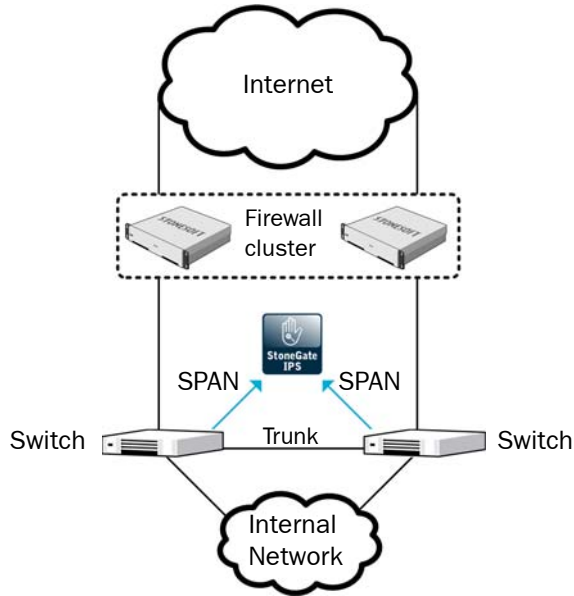
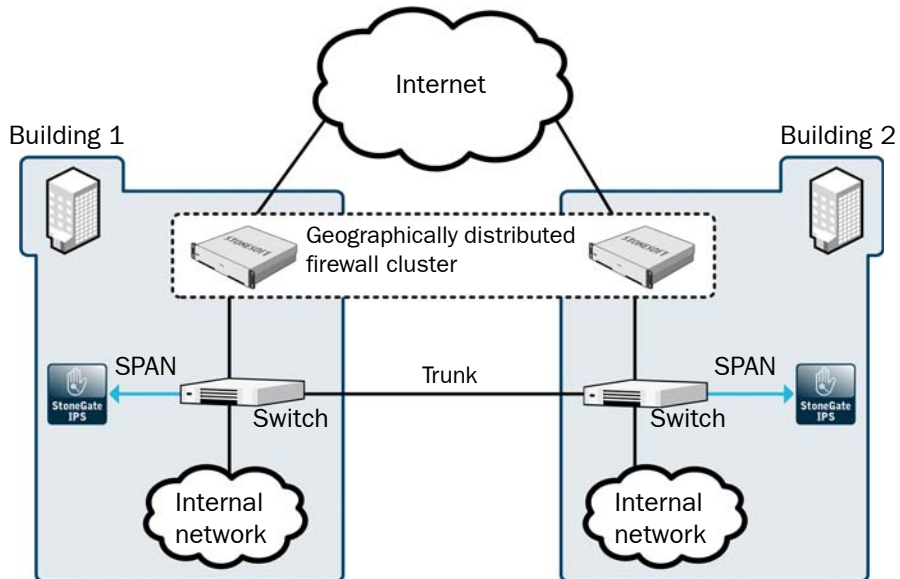


Illustration 4.11 Sensor Connected to SPAN Ports Alongside Redundant Switches



In a redundant disaster-recovery setup, firewall cluster nodes can be far apart. The sensors are not clustered in this configuration, but they have identical policies.

Illustration 4.12 Single Sensors in a Distributed Disaster-Recovery Environment



Deploying Sensors in IPS Configuration

In inline IPS configuration, the sensors are installed directly in the traffic path. Fail-open network cards are recommended to allow traffic flow when sensors are offline.



Caution - Always use standard cabling methods with inline IPS. Use crossover cables to connect the appliance to hosts and straight cables to connect the appliance to switches/hubs (see [Cabling Guidelines](#) (page 42)).

Illustration 4.13 Basic Inline Installations: Inline Single Sensor (Left) and Serial Sensor Cluster

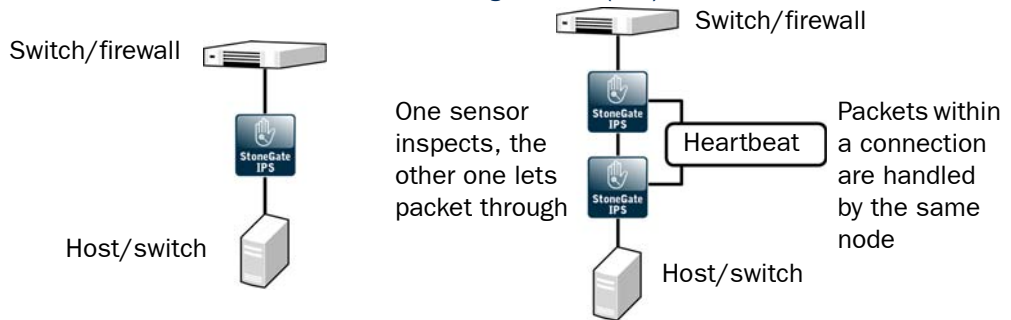


Illustration 4.14 Inline Sensors with External High-Availability/Load-Balancing

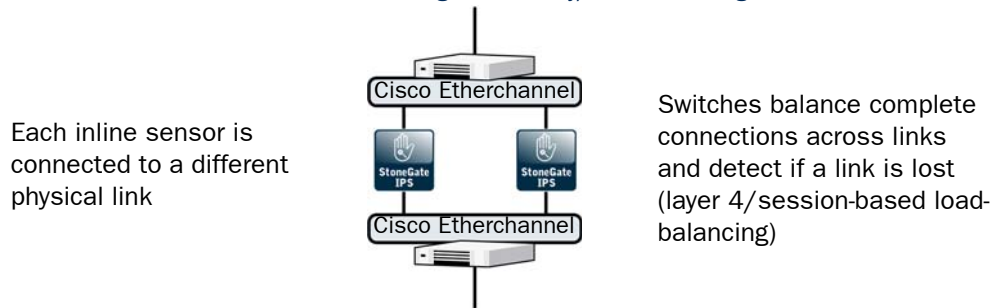
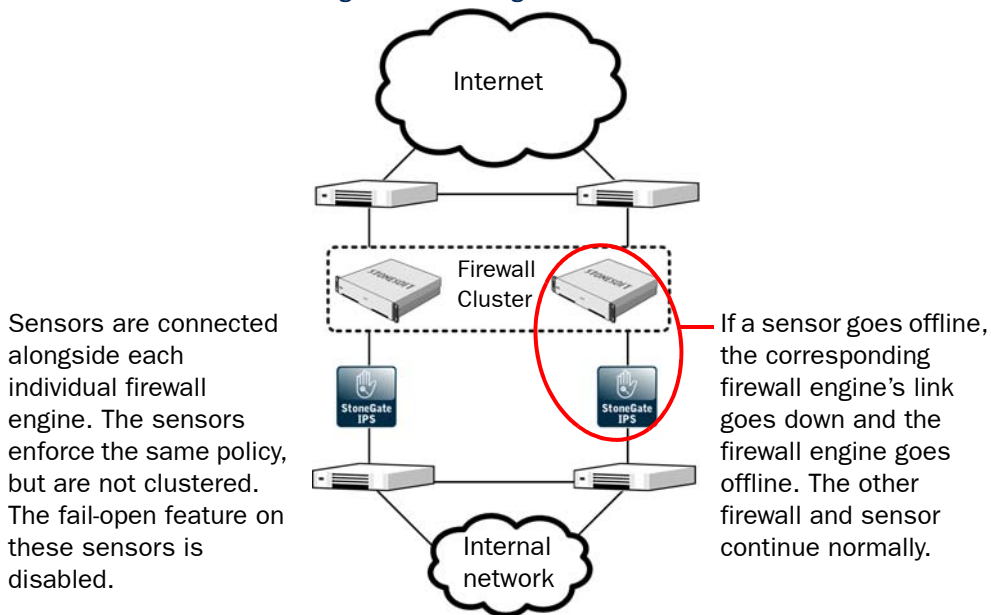


Illustration 4.15 Redundant Inline Single Sensors Alongside a Firewall Cluster



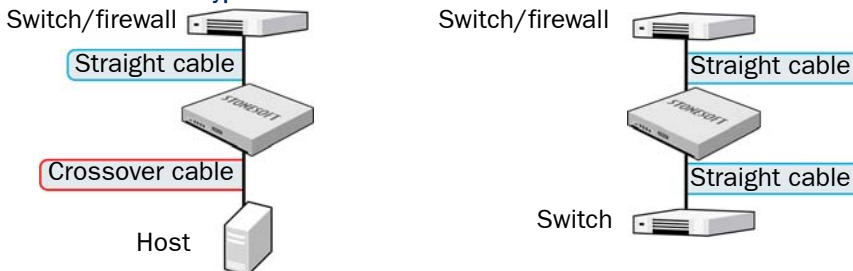
Cabling Guidelines

Cable Types

Follow standard cabling with inline IPS: use straight cables to connect the sensor to switches/hubs and crossover cables to connect the sensor to hosts. Both crossover and straight cables may work when the sensors are operating normally due to software-level correction, but only the correct type of cable allows traffic to flow when fail-open network cards must pass traffic without the help of higher-level features.

Also, make sure the copper cables are correctly rated (CAT 5e or CAT 6 in gigabit networks).

Illustration 4.16 Correct Cable Types

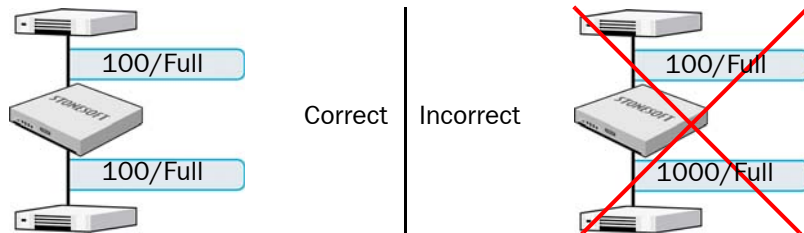


Speed and Duplex

Mismatched speed and duplex settings are a frequent source of networking problems. The basic principle for speed and duplex is simply that network cards at both ends of each cable must have identical settings. This principle also applies to the automatic negotiation setting: if one end of the cable is set to autonegotiate, the other end must also be set to autonegotiate and not to any fixed setting. Gigabit standards require interfaces to use autonegotiation—fixed settings are not allowed at gigabit speeds.

Inline interfaces of sensors require additional consideration: since the sensor is a “smart cable”, the settings must be matched on both links within each inline interface pair (identical settings on all four interfaces) instead of just matching settings at both ends of each cable (two + two interfaces). If one of the links has a lower maximum speed than the other link, the higher-speed link must be set to use the lower speed.

Illustration 4.17 Speed/Duplex Settings



SETTING UP STONEGATE IPS

In this section:

Sensor and Analyzer Configuration - 47

Routing - 59

CHAPTER 5

SENSOR AND ANALYZER CONFIGURATION

A Sensor Cluster is a group of sensor nodes that work as a single logical entity to share the load of traffic processing. A Single Sensor is a sensor that only has one sensor engine. A Combined Sensor-Analyzer consists of a single sensor engine and an analyzer engine in the same device.

The following sections are included:

- ▶ [Overview to Sensor and Analyzer Configuration](#) (page 48)
- ▶ [Configuration of Sensors and Analyzers](#) (page 48)
- ▶ [Using Sensors and Analyzers](#) (page 53)
- ▶ [Examples of Sensor and Analyzer Configuration](#) (page 55)

Overview to Sensor and Analyzer Configuration

This chapter concentrates on configuration of network interfaces and clustering, which are the only parts of the configuration where there are major differences between a single sensor and a clustered installation. The section [Using Sensors and Analyzers](#) (page 53) addresses other configuration tasks that you may do in the Sensor elements' properties.

Configuration of Sensors and Analyzers

StoneGate Sensors and Analyzers are configured and managed centrally through the Management Server. The Single Sensor, Sensor Cluster, Sensor-Analyzer and Analyzer elements represent the sensor and analyzer configuration on the Management Server. The main configuration options in the Sensor and Analyzer elements include the settings related to network interfaces and clustering (for sensors). This chapter concentrates on those settings.

Heartbeat Network for Sensor Clusters

The nodes in a sensor cluster exchange status information via a *heartbeat network* using multicast transmissions. If a sensor node becomes unavailable, the other nodes of the cluster immediately notice this, and connections are reallocated to the available nodes. A dedicated network is recommended for at least the Primary heartbeat communications. Unlike a firewall cluster, nodes in a sensor cluster *do not* exchange state synchronization information.

Network Interfaces

The network interfaces of a StoneGate Sensor or Analyzer are identified by *Interface IDs*. During the configuration in the Management Client, you define the network interfaces of the Sensor or Analyzer. During the engine configuration on the command line, the Interface IDs are mapped to the engine's physical interfaces. For each physical network interface, a unique Interface ID must be specified.

Depending on whether you are configuring a Single Sensor, a Sensor Cluster, a combined Sensor-Analyzer, or an Analyzer, you can configure the following types of interfaces for each Interface ID in use:

Table 5.1 Sensor and Analyzer Interface Types

Interface Type	Available on	Description
Capture Interface	Single Sensors, Sensor Clusters, Sensor-Analyzers	Capture interfaces are used for listening to traffic that is not routed through the sensor. They are dedicated for capturing network traffic, and cannot be used for other purposes. On combined Sensor-Analyzers, only the sensor uses the capture interfaces.

Table 5.1 Sensor and Analyzer Interface Types (Continued)

Interface Type	Available on	Description
Inline Interface	Single Sensors Sensor-Analyzers Inline Serial Clusters	Inline interfaces handle traffic that is routed through a Sensor in inline Sensor mode or Transparent Access Control mode, or Sensors in an Inline Serial Cluster. These inline interfaces cannot be simultaneously used for other purposes. On combined Sensor-Analyzers, only the sensor uses the inline interfaces.
Normal Interface	Single Sensors, Sensor Clusters, Sensor-Analyzers, Analyzers	Normal interfaces are used for Management communications, the Heartbeat between the nodes, sending event information to the Analyzer, and sending TCP Reset responses, as well as all communication for node-initiated connections.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Elements**.

Task 1: Create an Analyzer or Sensor Element

To introduce a new Analyzer, Single Sensor, Sensor Cluster or combined Sensor-Analyzer to the Management Center, you must create an element that stores the configuration information related to the engines.

During the Sensor element definition, you select the Analyzer to which the Sensor sends its event information. For this reason, you must define an Analyzer element before you begin defining Sensor elements. The only exception to this is when you are creating a combined Sensor-Analyzer. In that case, the Analyzer and the Sensor are defined at the same time.

Task 2: Create Physical Interfaces

Physical interfaces represent the actual network interfaces on the engines or the virtual interfaces on a virtual IPS appliance. In a Sensor cluster, each physical interface definition represents a network interface on all nodes of the cluster.

The network interface numbering in the configuration is independent from the numbering of the physical interfaces. By default, the two numbering schemes are mapped one to one, but you can change the mapping freely using command line tools on the engine. This mapping can be done differently from node to node as long as you take care that the same interface on each node is correctly cabled to the same network.

Task 3: Define VLAN Interfaces (*Sensor Elements Only*)

A *Virtual Local Area Network* (VLAN) is a logical grouping of hosts and network devices that appear as a single network segment regardless of its physical topology. StoneGate supports *VLAN tagging* as defined in the IEEE 802.1q standard. One network interface can support up to 4094 VLANs.

VLAN tagging can be used:

- to inspect VLAN tagged traffic (no VLAN interface configuration required on the Sensor)
- to define different inspection rules for different VLANs (requires defining VLAN interfaces for the Sensor)
- for the Sensor's own management and event logging communications when the control network interface is connected to a VLAN trunk.

By default, all captured VLAN traffic is inspected in the same way as non-VLAN traffic. You only need to configure VLAN interfaces for the Sensor capture interfaces if you want to customize traffic inspection for the different VLANs. The traffic inspection is customized for the VLANs by defining different Logical Interfaces for the different VLAN capture interfaces. The Logical Interface elements are then used in the IPS policy rules to define which rules are used for which VLANs.

When VLAN is used with inline interfaces, the interface numbers must be different and the VLAN identifier must be identical in both of the inline interfaces. For example, 3.101 and 4.101 would be a valid pair of VLAN inline interfaces. Also, when a VLAN interface is used for an inline interface, it cannot be simultaneously used for any other type of interfaces.

When VLAN is used with capture interfaces, the network interface used as the *Reset Interface* for sending TCP Reset responses must be defined in the capture interface's properties. The reset is automatically tagged for the same VLAN that triggers a reset. The reset interface must be connected to the same VLAN/Broadcast domain as the capture interface to reach the communicating hosts.

Task 4: Define Normal Interfaces

In a Single Sensor, normal interfaces are used as the Control Interface for communication between the sensor and the Management Server, for sending event information to analyzers, for sending traffic recordings to Log Servers, and as the Reset Interface for sending TCP Reset responses.

In a Sensor Cluster, the normal interfaces handle all traffic for which the end-point of the communication is a node itself. The normal interfaces are used as the Heartbeat Interface for communication between the nodes, as the Control Interface for communication between each individual node and the Management Server, for sending event information to analyzers, for sending traffic recordings to Log Servers, and for any other traffic between the node itself and some other host. Normal interfaces in a sensor cluster are also used as the Reset Interface for sending TCP Reset responses.

On an Analyzer, normal interfaces are used for communication between the analyzer and the Management Server, and for sending data to the Log Server. Normal interfaces are the only type of interfaces that can be defined for analyzers.

Each single sensor, sensor node in a cluster, and analyzer must have at least one normal interface defined. Multiple normal interfaces can be configured for the same physical network interface. It is recommended to create a separate normal interface that is used as the Control Interface for communication with the Management Server rather than using the same normal interface for sending event information to analyzers, sending traffic recordings to Log Servers, and for communication with the Management Server.

You can optionally assign an IP address to a normal interface. When a normal interface is used as the Control Interface for communication with the Management Server, the Heartbeat Interface in a sensor cluster, or for communication with the Log Server or Analyzer, an IP Address is needed. When the same normal interface that is used for communication with the Management Server, Log Server, and Analyzer is also used as a Reset interface for sending TCP Reset responses, it can have an IP address. When a normal interface is used *only* as a Reset Interface, it must not have an IP address.

All nodes in a sensor cluster must have the same netmask value for the IP address of their respective normal interfaces. The IP addresses specified for each node are used whenever the nodes need to be contacted individually.

The normal interface MAC addresses do not need to be modified unless you want to override the MAC address on the physical interface. Because of the limitation of only one unicast MAC address per physical interface, all the normal interfaces defined on the same physical interface use the same MAC address.

Task 5: Define Logical Interfaces (*Sensor Elements Only*)

A Logical Interface is used in the capture interface and inline interface configuration to represent one or more network interfaces. A Logical interface can represent any number or combination of physical interfaces or VLAN interfaces, except that the same Logical interface cannot be used to represent both capture interfaces and inline interfaces on the same Sensor.

Logical interfaces have one option, View interface as one LAN, which can be turned on or off. By selecting the option, you avoid the sensor seeing a single connection as multiple connections when a switch passes traffic between different VLANs and all traffic is mirrored to the sensor through a SPAN port.

Logical Interfaces can also be used in IPS policies to create rules that match based on which interface the traffic was picked up from. For example, you can create a different logical interface for each VLAN and use them to create rules that apply only to traffic from a specific VLAN.

Task 6: Define Capture Interfaces (*Sensor Elements Only*)

You must define capture interfaces if you want to use the Sensor to listen to traffic that is not routed through the Sensor.

Capture interfaces have definitions for the corresponding *Logical Interface* that this interface belongs to. The Logical Interface represents one or more network interfaces which capture the traffic for inspection:

- When a capture interface is connected to a switch SPAN port, each capture interface is bound to one Logical Interface. More than one capture interface can optionally be bound to the same Logical Interface.
- When a network TAP device is used, two capture interfaces are bound to the same Logical Interface, as the monitored traffic going to different directions is captured through these two related network interfaces and is then combined into a complete traffic flow on the Logical Interface.

You cannot select the same Logical interface for a capture and an inline interface on the same Sensor.

A Reset Interface can be selected for a capture interface to send TCP Reset responses for the traffic captured from this interface. The Reset Interface is a normal interface that can reach the communicating machines with the TCP Reset, for example, a normal interface connected to the monitored network.

Task 7: Define Inline Interfaces (*Sensor Elements Only*)

You must define inline interfaces if you want to place a Single Sensor, Sensor-Analyzer, or Inline Serial Cluster directly in the traffic path so that any traffic that is to be inspected goes through the Sensor. An inline interface is configured with two Interface IDs, representing two physical interfaces or two VLANs. Some StoneGate appliances use a fail-open network card, so the inline interfaces must be configured for those specific ports. Inline interfaces do not have an IP address or a MAC address visible to the network.

In addition to the Interface IDs, inline interfaces also have definitions for the corresponding Logical Interface that this interface belongs to. A single Logical interface can represent one or more pairs of inline interfaces. The Logical Interface element can be used to represent the interfaces in the IPS policy. You cannot select the same Logical Interface for a Capture and an inline interface on the same Sensor.

Task 8: Select Interface Options

The Interface Options allow you to define which physical interfaces are used as the primary and backup control interface, primary and backup heartbeat interface (*sensor cluster only*), and the log/analyzer communication source IP address. By default, the first normal interface with an IP address you create is automatically selected as the primary control interface, the primary heartbeat interface (*sensor cluster only*), and the log/analyzer communication source IP address. You can optionally change which physical interface is used for each of these purposes, and define a backup control interface and backup heartbeat interface (*sensor cluster only*).

Task 9: Install the Sensor and Analyzer Engines

During the engine installation, you map the physical interfaces to the interface IDs you define in the Management Client.

You can also install the engine automatically using a configuration saved on a USB memory stick. If you use the automatic engine configuration, interface IDs are automatically mapped to physical interfaces in sequential order. For example, Interface ID 0 is mapped to eth0, Interface ID 1 is mapped to eth1, and so on. The first physical interface (eth0) is always used as the Management interface. For this reason, Interface ID 0 must be defined as the Management interface in the Management Client when automatic engine configuration is used.

You also activate a basic IPS Policy (the initial configuration) that allows you to establish contact between the Management Server and the engine. After contacting the Management Server, the engine receives a certificate from the Management Center for identification, and a trust relationship between the engine and the Management Server is established.

Task 10: Install an IPS Policy

After the Sensor or Analyzer engine makes initial contact with the Management Server, only the primary control interface with the Management Server is configured. You must install an IPS Policy using the Management Client to transfer the complete interface configuration to the Sensor or Analyzer.

Using Sensors and Analyzers

The main points of Sensor and Analyzer configuration are explained in the preceding sections of this chapter, but this section illustrates some additional concepts that are useful when working with Sensors and Analyzers.

Contact Addresses for NATed Communications

In a situation where a device performs network address translation (NAT) between the communicating StoneGate components, you must specify contact addresses for the components. The *contact address* is the NATed address of the component that is contacted instead of the interface's real IP address.

The contact addresses for the system components on each "site" behind NAT are grouped into a *Location* element. The contact address for each component is defined in the element's properties based on the Location of the contacting component.

For example, when a Management Server contacts a sensor node through NAT, the Management Server uses the NATed contact address, not the sensor node's real IP address. The NAT device in between performs the address translation from the NATed address to the sensor's real IP address as usual.

You create the Locations and add elements in the Locations based on how your network is set up. Then you define the Contact Addresses for each element for each Location in the properties of the elements. All Management components in other Locations then use the addresses defined for their Location for contact.

Cluster Load Balancing

In a clustered sensor configuration, the recommended way to cluster the nodes is load-balanced clustering, where traffic is balanced between the nodes dynamically. Load-balanced clustering provides both fault tolerance and performance benefits.

If load-balanced clustering is used, the traffic arriving at the Sensor cluster is balanced across the nodes by means of a load balancing filter. This filtering process distributes packets between the sensor nodes and keeps track of packet distribution. The Sensor cluster determines the packet ownership of the nodes by comparing the incoming packet with node-specific values based on the packet headers.

The Sensor cluster keeps track of which node is handling each ongoing connection. As a result, all packets that are part of a given connection can be handled by the same node. Some protocols use multiple connections, which are sometimes handled by different nodes, but this usually does not affect the processing of the traffic.

Processor Load Balancing

In a situation where traffic comes from a small number of IP addresses but a large number of ports, such as when a sensor is deployed between two proxy servers, load balancing based only on IP addresses does not efficiently distribute the inspection of traffic between the available processors in the sensor hardware, resulting in reduced throughput. You can optionally configure the Sensor to take the port information into account when balancing the traffic between the processors to improve the inspection efficiency and increase throughput.

TCP Inspection Modes

IPS sensors and sensor-analyzers can handle TCP connections in two different modes: the *Normal* mode and the *Strict* mode.

Table 5.2 TCP Inspection Modes

Type	Description
Normal TCP	<p>This is the default mode for handling TCP connections. It is the only available inspection mode when the IPS is used as an intrusion detection system (IDS).</p> <p>In Normal TCP mode, the sensor or sensor-analyzer checks that the traffic proceeds according to the TCP protocol. The sensor or sensor-analyzer does not need to see all the packets in a TCP connection (for example, the packets that initiate a new TCP connection). The sensor or sensor-analyzer does not modify the packets and it does not enforce the packet direction. This means, for example, that SYN and SYN-ACK packets are allowed from the same interface.</p>
Strict TCP	<p>The Strict TCP inspection mode provides enhanced protection against TCP evasion attempts. It is only available with inline IPS. The Strict TCP mode is used by default for handling TCP connections with the TLS Inspection and Web Filtering features. Optionally, you can also enable the Strict TCP mode for all TCP traffic in the Sensor element properties.</p> <p>In Strict TCP mode, the sensor or sensor-analyzer controls the progress of a TCP connection and checks that the TCP handshake proceeds according to the TCP protocol. The same sensor or sensor-analyzer node must be able to see all the packets in the connection. The sensor or sensor-analyzer also enforces the packet direction (for example, SYN and SYN-ACK packets are not allowed from the same interface). In addition, the sensor or sensor-analyzer can also modify the TCP packets' header, especially the TCP window size, and timestamps. It can also remove certain TCP options from the packets. In TLS Inspection, the sensor or sensor-analyzer also modifies the TCP packet data (decrypts the packets for inspection and then encrypts the packets again).</p> <p>If a sensor or sensor-analyzer cannot inspect the whole TCP connection (for example, the sensor or sensor-analyzer goes offline), the connection is dropped if TLS inspection is applied to the connection. If Web Filtering is used, and the inspection of the connection starts in the middle of the connection (for example, the sensor goes from offline to online state), or the connection is transferred to another sensor or sensor-analyzer in a high-availability environment, the inspection is done as in the Normal TCP mode.</p> <p>When TLS inspection is first activated, the currently open TCP connections are inspected according to the Normal TCP mode and the Strict TCP mode is only applied to the new TCP connections. The same applies for all TCP connections if the Strict TCP mode has been manually enabled in the Sensor element properties.</p>

Examples of Sensor and Analyzer Configuration

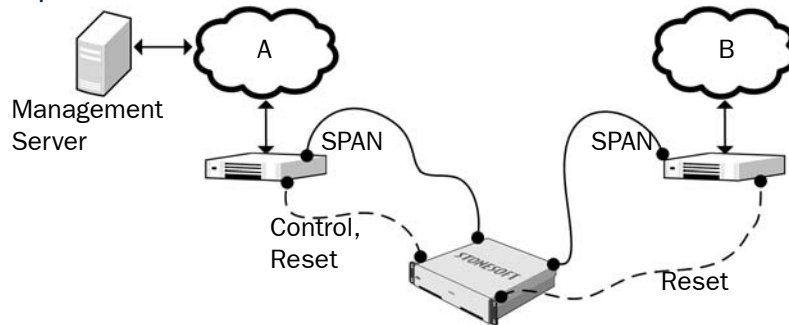
The examples in this section illustrate some common uses for Sensor and Analyzer configuration in StoneGate and general steps on how each scenario is configured.

Configuring Capture Interfaces with SPAN

The administrator at company A wants to set up a combined sensor/analyzer and deploy it in IDS configuration using SPAN ports on the switches to duplicate packets for inspection.

[Illustration 5.1](#) shows the interfaces of the sensor in IDS configuration.

Illustration 5.1 Capture Interfaces with SPAN



In this example, Interface ID 0 is a normal interface used for management connections, and sending TCP Reset responses for network segment A. Interface ID 1 is a capture interface for capturing network traffic from the network segment A switch for inspection. Interface ID 2 is a capture interface for capturing network traffic from the network segment B switch for inspection. Interface ID 3 is a normal interface used for sending TCP Reset responses for network segment B.

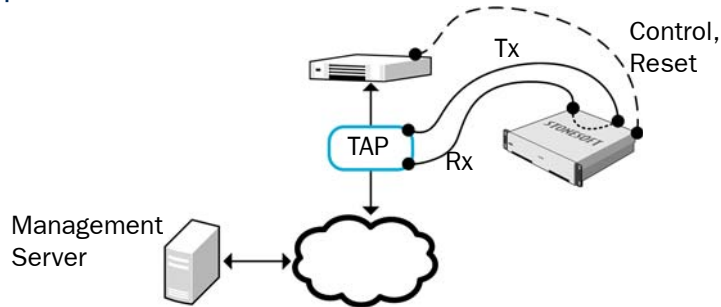
The administrator does the following:

1. Creates a Combined Sensor-Analyzer element and selects the Log Server, the Log Server for Alerts, and the Log Server to which traffic recordings are sent.
2. Defines Interface ID 0 as a normal interface and adds an IP address to it.
 - Interface ID 0 is automatically selected as the Primary Control Interface because it is the first normal interface with an IP address.
3. Defines Interface ID 3 as a normal interface without an IP address.
 - Because Interface ID 3 is used only as a reset interface, it must not have an IP address.
4. Defines Interface ID 1 as a capture interface and selects Interface ID 0 as the Reset Interface.
5. Defines Interface ID 2 as a capture interface and selects Interface ID 3 as the Reset Interface.
6. Saves the initial configuration of the engine in the Management Client.
7. Maps the interface IDs to the physical interfaces in the engine Configuration Wizard on the engine's command line and makes initial contact with the Management Server.
8. Installs an IPS policy in the Management Client to transfer the configuration to the engine.

Configuring Capture Interfaces with TAP

The administrator at company B wants to set up a combined sensor/analyzer and deploy it in IDS configuration using a network TAP device. WireTAP copies transmitted (Tx) and received (Rx) packets from the monitored cable and forwards them to separate links for further analysis in the sensor. [Illustration 5.2](#) shows the interfaces of the sensor in IDS configuration.

Illustration 5.2 Capture Interfaces with TAP



In this example, Interface ID 0 is a normal interface used for management connections, and sending TCP Reset responses. Interface ID 1 is a capture interface that listens to the received (Rx) packets from the network TAP. Interface ID 2 is a capture interface that listens to transmitted (Tx) packets from the network TAP. Interface IDs 1 and 2 share the same Logical Interface, which combines the traffic from both physical interfaces so that it can be inspected as a complete traffic flow.

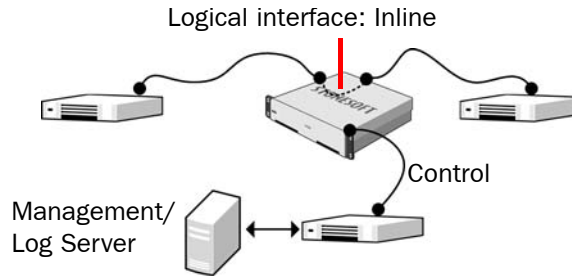
The administrator does the following:

1. Creates a Combined Sensor-Analyzer element and selects the Log Server, the Log Server for Alerts, and the Log Server to which traffic recordings are sent.
2. Creates a Logical Interface called Capture for the two capture interfaces.
3. Defines Interface ID 0 as a normal interface and adds an IP address to it.
4. Defines Interface ID 1 and Interface ID 2 as capture interfaces, selects Interface ID 0 as the Reset Interface, and selects the Logical Interface called Capture for both.
5. Saves the initial configuration of the engine in the Management Client.
6. Connects the network cables to the appropriate NICs.
7. Maps the interface IDs to the physical interfaces in the engine Configuration Wizard on the engine's command line and makes initial contact with the Management Server.
8. Installs an IPS policy in the Management Client to transfer the configuration to the engine.

Configuring Inline Interfaces

The administrator at Company C wants to set up a combined sensor-analyzer and deploy it in the traffic path. [Illustration 5.3](#) shows the interfaces of the inline sensor.

Illustration 5.3 Branch Office Inline Sensor



In this example, Interface ID 0 is configured as the Control interface for management connections. Interface ID 1 and Interface ID 2 are an inline interface pair that share the same logical interface, called Inline. Traffic comes in through Interface ID 1. Any traffic that is allowed by the sensor leaves through Interface ID 2.

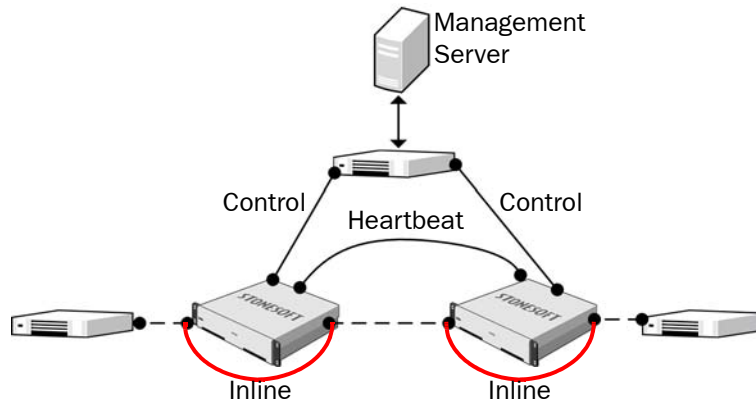
The administrator does the following:

1. Creates a Combined Sensor-Analyzer element and selects the Log Server, the Log Server for Alerts, and the Log Server to which traffic recordings are sent.
2. Creates a logical interface called Inline for the inline interface pair.
3. Defines Interface ID 0 as a normal interface and adds an IP address to it.
4. Defines Interface IDs 1 and 2 as an inline interface pair and selects the logical interface called Inline for the pair.
5. Saves the initial configuration of the engine in the Management Client.
6. Connects the network cables to the appropriate NICs.
7. Maps the interface IDs to the physical interfaces in the engine Configuration Wizard on the engine's command line and makes initial contact with the Management Server.
8. Installs an IPS policy in the Management Client to transfer the configuration to the engine.

Setting up an Inline Serial Sensor Cluster

The administrators at Company D want to set up an Sensor Cluster in a serial inline deployment. [Illustration 5.4](#) shows the interfaces of the inline sensor cluster.

Illustration 5.4 Inline Serial Sensor Cluster



In this example, the cluster consists of two sensor nodes. Interface ID 0 is a normal interface used for the heartbeat communication between the nodes. Interface ID 1 is a normal interface used as the control interface for communication with the Management Server. Interface ID 2 and Interface ID 3 are an inline interface pair that share one logical interface, called Inline. Traffic enters each sensor node through Interface ID 2 and leaves through Interface ID 3.

The administrators:

1. Define and configure an Analyzer element.
2. Create a Sensor Cluster element, select the Analyzer to which event data is sent, and the Log Server to which traffic recordings are sent.
3. Define Interface ID 0 as a normal interface and add IP addresses for each of the nodes. Interface ID 0 is automatically selected as the Primary Control Interface, the Primary Heartbeat Interface, and the Log/Analyzer communication source IP Address.
4. Define Interface ID 1 as a normal interface and add IP addresses for each of the nodes.
5. Define Interface IDs 2 and 3 as an inline interface pair and select the Logical Interface called Inline for the pair.
6. Select Interface ID 0 as the Primary Heartbeat Interface and Interface ID 1 as the Primary Control Interface in the Interface Options.
7. Save the initial configuration of the engine in the Management Client.
8. Connect the heartbeat and inline interfaces between the nodes with crossover cables, and the rest of the interfaces with straight cables.
9. Map the interface IDs to the physical interfaces in the engine Configuration Wizard on the engine's command line and makes initial contact with the Management Server.
10. Install an IPS policy on each of the nodes in the Management Client to transfer the configuration to the engine.

CHAPTER 6

ROUTING

Routing defines which network interface StoneGate selects to reach a particular destination address. Routes are only needed for the communications the engines initiate with other system components. StoneGate IPS components do not route other traffic.

The following sections are included:

- ▶ [Overview to Routing](#) (page 60)
- ▶ [Configuration of Routing](#) (page 60)

Overview to Routing

Routing information is used for deciding which network interface is used to reach any given destination address. The Sensors and Analyzers need modifications in their routing configuration only if they are connecting to some other network than the directly connected networks. The Sensors and Analyzers do not function as gateways. They do not forward traffic from one network to another.

Configuration of Routing

Each Sensor and Analyzer element has a list of network interfaces with the directly connected networks in the Routing view. The routes for the Sensors and Analyzers are defined in the Management Client using Network elements. Usually, only a default route is needed for the Sensors and Analyzers. These are used when the sensors and analyzers need to open connections to some network other than the directly connected networks. No routes need to be defined if a sensor or an analyzer communicates only in its local IP network.

Default Elements

The system includes a default network element called *Any network*, which is needed to define the default route for sensors and analyzers.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Routing**.

Task 1: Add Router(s)

A Router element represents the next-hop gateway device that forwards packets to the network(s) you define.

Task 2: Add Network(s)

The Network element represents the IP addresses to which the Router forwards the traffic.

- To add a default route, add the default **Any network** element to the Router you just created.
- If you need to forward traffic to a network that is not directly connected and it cannot be reached through the default gateway, you must define a Network element for this network and add it under the Router.

Task 3: Refresh IPS Policy

To transfer the routing changes, upload the policy on the sensor. The Management Server sends all necessary configuration information when uploading a policy.

TRAFFIC INSPECTION

In this section:

Situations - 63

IPS Policies - 71

Ethernet Rules - 83

Access Rules - 91

Inspection Rules - 103

Protocol Agents - 113

TLS Inspection - 123

Web Filtering - 129

Blacklisting - 133

CHAPTER 7

SITUATIONS

Situation elements collect together the information that identifies and describes detected events in the traffic (or in the operation of the system). Situations contain the context information, that is, a pattern that the system is to look for in the inspected traffic.

The following sections are included:

- ▶ [Overview to Situations](#) (page 64)
- ▶ [Configuration of Situations](#) (page 64)
- ▶ [Using Situations](#) (page 68)
- ▶ [Examples of Custom Situations](#) (page 69)

Overview to Situations

Situations are elements that provide a way to define which traffic patterns and events you want to detect in your system with the Inspection rules in IPS Policies. The patterns and events are defined by selecting a Context for the Situation. The Context contains the information on the traffic to be matched, and the options you can set for the matching process.

Situations also provide a description that is shown in the logs, and a link to relevant external information (CVE/BID/MS/TA) in the form of a *Vulnerability* element attached to the Situation.

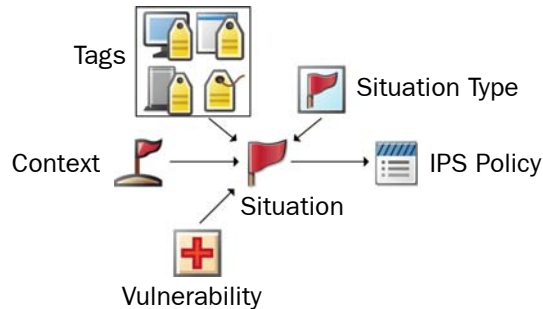
The Inspection rules in the IPS Policies define how the Situations are matched to traffic and what kind of action the system takes when a match to a particular Situation is found.

Correlation Situations are a special type of Situations that Analyzers use to group together event data sent by the Sensors and to find patterns in that data.

Configuration of Situations

The illustration below shows how Situations and the related elements are used together.

Illustration 7.1 A Situation and the Associated Elements



The Situation element uses different elements in the system to form a representation of the traffic that you want to detect in your IPS Policy.

- The **Tag** elements help you to create simpler policies with less effort. Tag elements represent all Situations that are associated with that Tag. For example, using the Tag “Windows” in a rule means that the rule matches all the Situations that concern Windows systems.
- The **Situation Type** elements define the general category of the Situation and includes the Situation in the Rules tree in Inspection rules. Each Situation can be associated with one Situation Type.
- The **Context** element allows you to define what you want your custom Situation to detect. The Context binds the Situation to a certain type of traffic and gives you a set of options or a field for entering a regular expression.
- The **Vulnerability** element associates your custom Situation with a commonly known vulnerability. It allows you to attach a description of the Vulnerability and references to public vulnerability databases (which are shown in the Logs view if a match is found).

The Context is the only mandatory element in a Situation. However, it will help in the long run if you consistently associate all relevant Tags with each Situation you create. The Vulnerability description is not mandatory, but it is helpful to have it for Situations that detect some publicly known issue.

Situation Contexts

The sections below explain the types of Context elements available and how they can be configured. They provide a framework for defining the parameters of each Situation. The parameters are entered as a regular expression or through a set of fields and options that you can adjust, depending on the Context element selected.



Note – The details related to the Contexts in your system may be different from what is described here, because the Contexts may have been updated through dynamic update packages after this guide was published. Read the Release Notes of each update package you import to see which elements are affected.

Correlation Contexts

Correlation Contexts define the patterns the Analyzers look for in traffic. There are five types of Correlation Contexts:

Table 7.1 Correlation Context Types

Correlation Context Type	Description
Compress	Combines repeated similar events into the same log entry, reducing clutter in the Logs view. For example, you could have a custom Situation for detecting suspicious access to a file server. An attacker is likely to browse through many files, triggering an alert entry for each file. An Event Compress Situation can be used to combine Situations together when the suspect's IP address is the same.
Count	Finds recurring patterns in traffic by counting the times certain Situations occur within the defined period, so that action can be taken if the threshold values you set are exceeded. For example, a Situation that detects access to a system could normally trigger just a log entry, but the Event Count Situation could be used to blacklist connections when access by any single host is too frequent.
Group	Finds event patterns in traffic by keeping track of whether all events in the defined set of Situations match at least once in any order within the defined time period. For example, individual attempts to exploit different vulnerabilities in a software product in use on your server may not be too alarming if you know that your system is patched against those vulnerabilities. However, when several such events are found in a short period of time, it becomes more likely that someone is trying to systematically attack the server and already knows that the server is running that particular piece of software. A Situation that belongs to the Group Context can detect this.
Match	Allows you to use Filters to filter event data produced by specific Situations.

Table 7.1 Correlation Context Types (Continued)

Correlation Context Type	Description
Sequence	<p>Finds event patterns in traffic by keeping track of whether all events in the defined set of Situations match in a specific order within the defined time period. For example, clients may use a certain type of request (e.g., “give file X”) to fetch a file from a file server. When administrators log in to the same server, a successful administrator login can be seen in the traffic as a certain type of response (e.g., “full access granted”). However, a vulnerability in the server software allows an attacker to send a specially crafted file fetch request that looks like a valid “give file x” command, but actually causes the server to give the attacker administrator access. This is seen as a normal-looking “full access granted” response from the server. The Event Sequence Situation can detect when a “give file X” Situation match is followed by a “full access granted” Situation match, which cannot be any legitimate traffic.</p>

Detailed descriptions of the parameters for each of the Correlation Contexts can be found in [Situation Context Parameters](#) (page 157).

DoS Detection Contexts

The DoS Detection Contexts provide parameters for detecting DoS (Denial of Service) events in network traffic.

Scan Detection Contexts

The Scan Detection Contexts provide parameters for adjusting attempts to scan which IP addresses are in use or which ports are open in your systems.

Detailed descriptions of the parameters for the Scan Detection Contexts can be found in [Situation Context Parameters](#) (page 157).

Protocol-Specific Contexts

The protocol-specific Contexts (the Contexts of the Application Protocols and Transport Protocols type) are used by Sensors to detect a particular characteristic in the network traffic. For example, you can detect a certain option number used in IP packets, or set the maximum length for particular arguments in FTP commands. You can also use the HTTP URL Filter to allow or deny access to specific websites.

For those Contexts that have particular values to be filled in (instead of a regular expression), the parameters you define in these contexts often actually determine what is considered normal, so that anything above/below/outside/not matching these values is considered a match for the Situation. So in other words, you may define what the Situation *does not* match.

Effective modifications to the protocol-specific Contexts require that you are familiar with the protocols in question and how the traffic in your network uses those protocols.

File Contexts

The File Contexts are used to detect malicious or suspicious content in transferred files regardless of the transport protocol used. When a file is detected, the file is inspected to identify the file type. Once the file type is identified, more specific inspection can be applied to the file.

System Contexts

The System Contexts are used for errors and other system events. They are internal to StoneGate, and they cannot be modified in any way.

Default Elements

There are many predefined Contexts, Situations, Tags, and Vulnerabilities available, which are imported and updated from dynamic update packages. This also means that the set of elements available in your system changes whenever you update your system with new definitions. Both Situations and Context elements have a comment and a longer description that you can view in the Management Client (in the Info panel or in the Properties dialog for the element) to see what each element is meant for.

The Release Notes of each dynamic update package list the new elements that the update introduces to your system. If your Management Server can connect to the Stonesoft website, you can view the release notes directly through the Management Client.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Create a Situation Element

You can create new Situations in addition to using the predefined ones. You can create a Situation element for Sensors or a Correlation Situation for Analyzers. A Situation element collects together the related elements and settings and sets the severity value for the Situation. The severity value can be set between Info (the least severe) to Critical (the most severe). You can use the severity value to restrict which Situations added to the Situations cell are considered in Inspection rule Exceptions and Alert Policies. For example, if a rule matches a large range of Situations you can create separate rules for less severe and more severe Situations.

Task 2: Add a Context for the Situation

Adding a Context to a Situation allows you to define what kinds of patterns you want to look for in the traffic. For example, you can specify that you want to look for a certain character sequence in an HTTP stream from the client to the server.



Note – With the exception of whitelisted URLs in URL Filtering, Situations are identified in the system only by the element name. Avoid matching the same pattern in different Situation elements. Situations with duplicate patterns can make the policy difficult to read and manage.

When you select a Context you get a set of options or a field for entering a regular expression as parameters for the Context. The parameters define the pattern you want to look for in the traffic. The syntax for StoneGate regular expressions is explained in [Regular Expression Syntax](#) (page 163). The context parameters for Port and Host Scan detection and the Correlation Situation parameters are explained in [Situation Context Parameters](#) (page 157).

Other types of context parameters are not listed in this guide. They concentrate on some aspect of a particular kind of network traffic, and using them requires that you have basic knowledge of the underlying network protocols. For more information on what a particular Context is used for, see the Properties dialog of the Context in question.

Task 3: Associate Tags and/or Situation Types with the Situation

You can use Tag elements to group Situations and Situation Types to classify Situations. You can use predefined Tags or create new ones according to any criteria (for example, create a Tag for grouping together related services). Situation Types are predefined, and you cannot create new Situation Types. You can associate multiple Tags with one Situation, but only one Situation Type can be associated with each Situation.

You can use the Tags and/or Situation Types to represent a group of Situations in the Inspection Rules. This allows you to match a rule to all Situations that contain the Tag or Situation Type. Situations that are associated with a Situation Type are automatically included in the Inspection Rules tree. See [Inspection Rules](#) (page 103) for more information.



Note – If a Tag or Situation Type you add to a Situation is in use in some IPS policy, the new Situation is automatically included in the policy when you save the Situation, and the IPS components start matching traffic to the Situation when you refresh the policy.

Task 4: Associate the Situation with a Vulnerability

Vulnerabilities provide a short description of the event that has matched. Vulnerability information is included in dynamic update packages, so all Situations provided by Stonesoft that are related to a known vulnerability are linked to a Vulnerability element. When you create your own Situations, you can associate them with an existing Vulnerability or a custom Vulnerability element.

You can add up to four references to public vulnerability databases to your custom Vulnerabilities (CVE/BID/MS/TA). System vulnerabilities can have an unlimited number of references to any reference system, and can have multiple references to the same reference system. The reference information is also shown in the Logs view.

Using Situations

Situations are used for defining what you want to detect with the Inspection rules. Situations are generally used for:

- Detecting malicious patterns in traffic. The Situations supplied by Stonesoft in dynamic update packages concentrate on such known vulnerabilities and exploits.
- Reducing the number of alert and log entries you receive from the system (using Correlation Situations).
- Detecting some other traffic patterns that you want to record, for example, you may be interested in the use of certain applications.

Although the general workflow requires ensuring that a Situation you want to use is included in the IPS policy, you may often not actually insert the Situation into the rule, but use a Tag or Situation Type element instead to represent a whole group of Situations.

Examples of Custom Situations

The examples in this section illustrate some common uses for Situations in StoneGate and the general steps on how each scenario is configured.

Detecting the Use of Forbidden Software

Company A has a Sensor deployed in between their internal network and the Internet. The Sensor uses a policy that is based on the IPS System Template.

The administrators find out that some of the internal users have installed a piece of software on their computers that the company's security policy forbids. They consider this software a security risk.

The administrators decide that they would like to detect the use of the software so that they can find out which users have installed it. The administrators find one simple but distinctive characteristic in the software: when launched, the software in question always connects to a particular address to check for updates using HTTP. The administrators:

1. Create a new custom Situation element with the name "Software X".
2. Add the **HTTP Client Stream** Context to the Situation and type in a regular expression that contains the address they want the Situation to find using the StoneGate regular expression syntax (see [Regular Expression Syntax](#) (page 163)).
3. Add one of the default Situation Types under **Traffic Identification** to the Situation.
4. Select the correct options for logging the traffic in the Rules tree in the IPS policy and install the policy on the Sensor.
5. Open the Logs view and filter the view using the "Software X" Situation as the filtering criteria.
6. See which computers use the forbidden software and take action based on which IP addresses are shown in the logs.

Counting Events

Company B has a StoneGate firewall and a sensor that monitor traffic to a DMZ network. The DMZ contains a server that provides information to Company B's partners. A while ago, users started complaining that the service had slowed down.

Upon investigation, Company B's administrators found out that the traffic had grown dramatically even though the number of users and the data on offer had stayed the same. They found out that one of the partners had made a misconfigured script that frequently copied

several large catalogs from Company B's server to their own server and had given the script to a few other partners as well. As a first step, the administrators decide to immediately stop excessive queries to the server. The administrators:

1. Create a custom Situation for detecting access to the catalog files.
2. Create a custom Correlation Situation, attach the **Count** Context to it, and define the settings for the Count Context to detect when there are more than 5 requests per minute to any of the files from the same source address.

Table 7.2 Context Settings for the Example Correlation Situation

Field	Option
Correlated Situations	Custom Situation
Time Window	60
Alarm Threshold	5
Log Fields Enabled	Select
Log Names	Src Addr

3. Insert the Correlation Situation in the IPS policy with blacklisting as the Action. The traffic from the offending hosts will be stopped at the StoneGate firewall.
4. Refresh the IPS policy on the Sensor.

Preventing Access to Forbidden Websites

The Administrators at Company C have noticed that employees frequently visit certain websites that are not related to their work. They want to block access to these websites to prevent employees from accessing them at work. To do this, they:

1. Create a new Situation element.
2. Add the **Website Access Control** Context to the Situation.
3. Specify the addresses they want to prevent access to. Access to the specified addresses will be blocked.
4. Refresh the IPS policy on the Sensor.

CHAPTER 8

IPS POLICIES

IPS policy elements are containers for the lists of rules that determine how the sensors and analyzers inspect traffic. The policy elements include IPS Template Policies, IPS Policies, and IPS Sub-Policies.

The following sections are included:

- ▶ [Overview to IPS Policies](#) (page 72)
- ▶ [Configuration of Policy Elements](#) (page 74)
- ▶ [Using Policy Elements and Rules](#) (page 79)
- ▶ [Example of Policy Element Use](#) (page 80)

Overview to IPS Policies

IPS policy elements store rules according to which the sensors and analyzers examine the traffic. This chapter introduces you to how these elements are used by the sensors and analyzers and explains how you can build a purposeful and efficient policy hierarchy using the different policy elements. The basics of building the actual traffic handling rules that are contained in the policy elements are discussed in the three chapters that follow (see [Ethernet Rules](#) (page 83), [Access Rules](#) (page 91), and [Inspection Rules](#) (page 103)).

Policy Hierarchy

The policy structure in StoneGate is a hierarchical structure based on templates, which allows you to:

- Reuse rules without duplicating them.
- Assign and enforce editing rights of different parts of a single policy to different administrators.
- Reduce the resource consumption of the sensors.
- Make policies easier to read.

The template and policy hierarchy is flattened when the IPS Policy is transferred to the sensors and analyzers, so the policy will look the same to the IPS components regardless of how it is organized on the Management Server (as long as the rules are in the same order). You can also create sections of conditional IPv4 Access rules that you can insert into the other policy elements. The sensor may skip the processing of a conditional block of rules based on whether or not certain common matching criteria is found in the packet being examined.

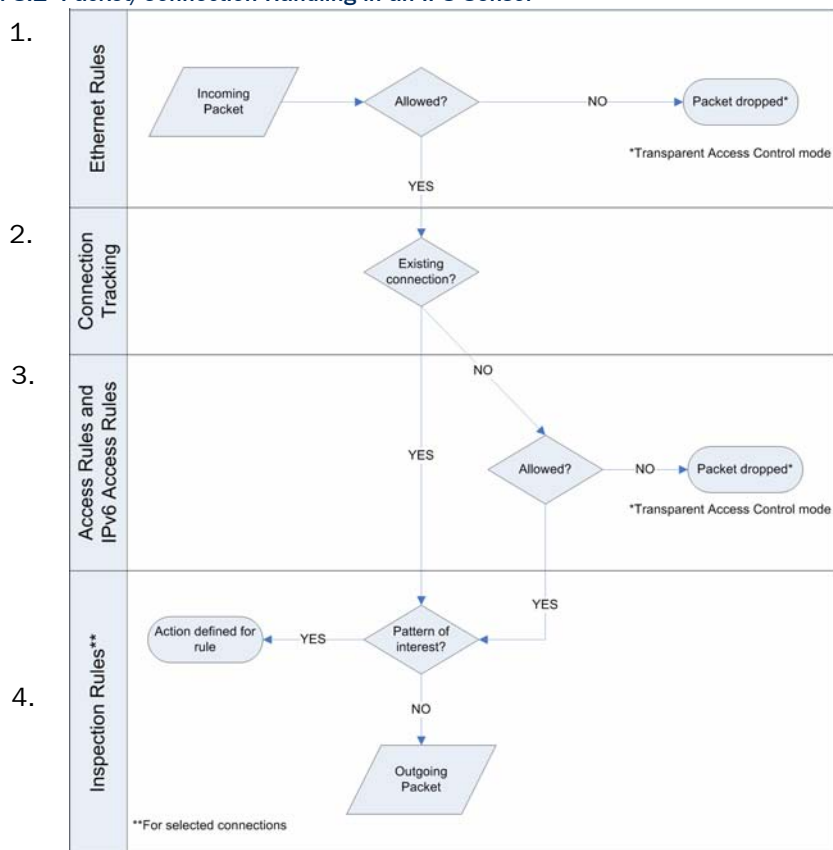
If your environment is simple and you do not see a need for the benefits outlined above, you have the option of creating a very simple policy hierarchy. You can start, for example, with a single custom IPS Policy built on one of the pre-defined Templates (the same policy can even be used on several sensors and analyzers).

How StoneGate Examines the Packets

The IPS system matches traffic to different protocols and then checks the definitions for known vulnerabilities and other threats for that protocol. The protocol is assigned first, before the deep inspection. The protocol assignment is done using Services, which match ports and protocol numbers to Protocol elements. Inline sensors can also filter traffic based on protocols, IP addresses, and the sensor interface that received the traffic without analyzing the traffic for threat patterns.

The packet handling process is shown in [Illustration 8.1](#).

Illustration 8.1 Packet/Connection Handling in an IPS Sensor



The illustration above shows how the packets are inspected:

1. An inline sensor or sensor-analyzer checks Ethernet frames against the Ethernet rules in the IPS policy. The packet is processed until it matches an Ethernet rule that tells whether to allow or to discard the packet. By default, all IP traffic is escalated to the Access rules.
2. The sensor or sensor-analyzer checks the current connection tracking information to see if the packet is part of an established connection (for example, a reply packet to a request that has been allowed). By default, all packets in established connections are escalated to the Inspection rules.
3. If the packet is not part of an existing connection, the packet is matched to IPv4 or IPv6 Access rules according to the IP protocol used.
 - If the traffic is tunneled using IP over IP or Generic Routing Encapsulation, the traffic can be checked against the IPv4 and/or IPv6 Access rules several times according to the number and type of layers in the tunnel and the settings of the engine.
 - The processing of the packet continues until the packet matches a rule that tells the sensor or sensor-analyzer to allow or discard the packet. Only inline sensors can drop

packets at this point in the inspection process. By default, all packets are escalated to the Inspection rules.

4. The sensor or sensor-analyzer matches Inspection rules for the connections that are selected for deep packet inspection in the IPv4 or IPv6 Access rules.
 - The Inspection rules are used to look for patterns of interest that are a part of allowed connections. The patterns may indicate potential attacks, exploits, or other possible threats. Alternatively, they can be any other patterns that might be of interest to the administrator (such as multiple login attempts, use of peer-to-peer or instant messaging software, or protocol violations in the traffic).
 - If a pattern in traffic matches a pattern defined in a rule, the action(s) defined in the rule are taken. Otherwise, no action is taken and the packet is allowed to pass through.

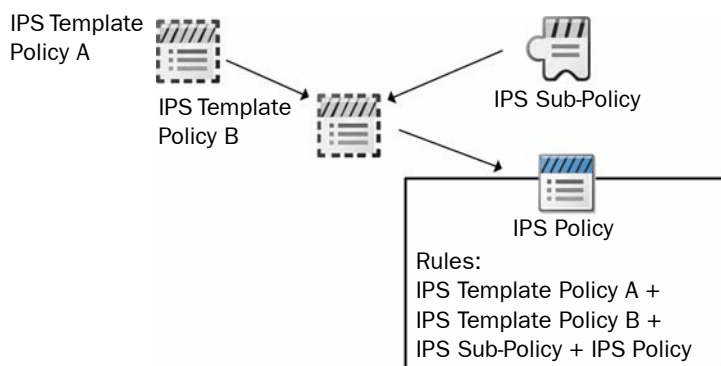
Configuration of Policy Elements

There are three kinds of IPS policy elements:

- An *IPS Template Policy* is a policy that can be used as the basis for IPS Policy and IPS Template Policy elements. The rules in the IPS Template Policy are copied as *inherited rules* into the IPS Policies and the IPS Template Policies that are based on the IPS Template Policy. Generally, you cannot edit the inherited rules directly in the IPS Policy or the IPS Template Policy to which the rules were inherited. To edit the rules, you must open the original IPS Template Policy for editing. However, the Rules tree in the Inspection rules can be edited regardless of changes made in the higher-level Template.
- An *IPS Policy* element is an element that gathers together all the rules from the different policy elements (the rules added directly to the IPS Policy, the rules from the higher-level IPS Template Policy, and possibly rules from one or more IPS Sub-Policies). IPS Policies are the only policy elements that can be installed on the IPS components.
- An *IPS Sub-Policy* element is a section of IPv4 Access rules that you can insert into IPS Policies and IPS Template Policies. The whole IPS Sub-Policy can be conditionally matched or skipped for each connection based on some basic matching criteria, which increases efficiency.

The hierarchy of how rules are inherited between different policy elements is shown in [Illustration 8.2](#).

Illustration 8.2 Rule Inheritance



In the illustration above, IPS Template Policy A is the basis for IPS Template Policy B, so IPS Template Policy B contains all the rules defined in IPS Template Policy A. Template Policy B also contains all the rules in an IPS Sub-Policy, as well as rules defined directly in IPS Template Policy B. The example IPS Policy inherits the following rules:

- All the rules in IPS Template Policy A.
- All the rules in IPS Template Policy B.
- All the rules in the IPS Sub-Policy.

Inherited rules cannot be edited in the policy that inherits the rules. For example, to change rules inherited from Template Policy A, administrators must have privileges to edit the Template Policy A in which the rules were originally defined. However, all definitions in the Rules tree in the Inspection rules are editable at all levels, even those overrides that are inherited from a Template Policy.

A hierarchy such as the one outlined above is useful to:

- Reduce the need for creating the same or similar rule in several policies. For example, any rule added to IPS Template Policy A is also added to any policy created based on IPS Template Policy A. The next time IPS Policies based on IPS Template Policy A are installed on IPS components, the new rule is used on all the components without the need to directly modify each individual IPS Policy.
- Restrict the editing rights of administrators. For example, administrators who are granted rights only to IPS Policy elements cannot edit the rules inherited from the IPS Policy Templates, with the exception on the Rules tree in Inspection rules. Their actions have no effect on rules that are placed above the row where the IPS Template Policy allows them to insert new rules. In the hierarchy shown in the illustration above, the insert point(s) for the IPS Policy are defined in IPS Template Policy B, which in turn can be edited only in the place where there is an insert point in IPS Template Policy A.
- Reduce the likelihood of mistakes affecting important communications. IPS Template Policies can be reserved for defining only the rules for essential communications, so that most daily editing is done in the lower-level IPS Policies. If the IPS Template Policy is properly designed, the rules in the IPS Template Policy cannot be overridden by any rules in the lower-level policy. Good organization also makes policies easier to read, further reducing the risk of errors.
- Improve processing performance. With the help of IPS Sub-Policies, whole blocks of rules may be skipped during processing when a connection does not match the rule that directs the traffic processing to the IPS Sub-Policy. This reduces the processor load, which may lead to improved throughput if the processor load is constantly very high.

Default Elements

The system has the following ready-made IPS Template Policies and IPS Policies:

- The IPS Strict Template
- The IPS System Template
- The Strict Policy
- The System Policy
- The Certification Policy

The IPS Strict Template and the IPS System Template contain a set of rules for detecting common threats. They provide an easy starting point for determining what kinds of rules your system needs. The IPS System Template contains rules that are suitable for the initial configuration of the IPS in most network environments. The rules in the IPS Strict Template are

suitable as the initial policy in high-risk environments, such as data centers. The only difference between the rules in the IPS Strict Template and the IPS System Template is in the way Inspection rules handle suspected attacks. In the IPS Strict Template, Suspected Attacks are terminated with an alert, whereas the IPS System Template only logs Suspected Attacks.

The Suspected Attacks are traffic patterns that are consistent with malicious activities, but are not any verified attack pattern. Suspected Attacks can catch zero-day attacks (attacks that are not yet publicly known), but may sometimes prevent some legitimate traffic from passing through if the traffic pattern happens to resemble malicious activities.

The Strict Policy and the System Policy do not add any rules to those defined in the IPS Strict Template and the IPS System Template. The Strict Policy and the System Policy exist by default so that you can install the predefined rules in the IPS Strict Template and the IPS System Template on the IPS engine right after installation (since Templates Policies cannot be installed on the engines).

The Certification Policy was used when StoneGate IPS was tested at NSS Labs. We recommend that you install either the Strict Policy or the System Policy as the initial policy on the IPS engines.

You can add new IPS Templates Policies freely to the system without basing them on any existing IPS Template Policy. However, in most cases the IPS Strict Template or the IPS System Template is the easiest starting point for your own customized IPS Template Policies and IPS Policies.

Situations are the central elements in the Inspection rules of your IPS Policies. The dynamic updates from Stonesoft contain the Situation elements for detecting exploit attempts against known vulnerabilities and other commonly known security threats. Because the dynamic updates include new and updated Situations, new patterns in traffic may begin to match when a new dynamic update is activated in your system and you refresh the IPS Policy.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF in the section called **Policies**.

Policy elements are merely containers for the actual traffic handling rules. When you have decided on a policy hierarchy, you can populate the policy elements with the actual rules for handling the traffic (see [Ethernet Rules](#) (page 83), [Access Rules](#) (page 91), and [Inspection Rules](#) (page 103)).

Task 1: Create an IPS Template Policy

IPS Template Policies are used as a basis for other IPS Policies and IPS Template Policies. Every IPS Policy that you create is based on an IPS Template Policy. You can also base several policies on the same IPS Template Policy. Although it is possible to create custom IPS Template Policies without basing them on either of the pre-defined templates, we recommend you to base your IPS Policy hierarchy on the IPS Strict Template or the IPS System Template. Updated versions of the Templates are included in dynamic update packages, so the policies that inherit rules from the pre-defined Templates are automatically updated when you activate a new dynamic update in your system. Policies based on a copy of the pre-defined templates or a completely different Template are not automatically updated.

When editing the policies, the main difference between IPS Template Policies and IPS Policies are the special rows called *Insert Points*. Insert points are shown in both IPS Template Policies and IPS Policies, but you can add them only to IPS Template Policies. The insert points added to templates mark the places where new rules can be added to policies that are based on the templates. When creating an IPS Template Policy, you must add insert points separately for all tabs in the Policy to be able to add rules on each tab.

Illustration 8.3 Insert Point in a Template and the Inheriting (Template) Policy

7	ANY	ANY	ANY	IP (IP in IP) with Rematch	IPv6 Encapsulation with Rematch
Access Rules					
9	ANY	ANY	ANY	IP (IP in IP) with Rematch	IPv6 Encapsulation with Rematch
10	ANY				
Access Rules					
9	ANY	ANY	ANY	ANY	ANY
10	ANY	ANY	ANY	ANY	ANY

Illustration 8.3 shows what the same insert point looks like in an IPS Template Policy and in the inheriting policy elements. The color of the insert point indicates whether the insert point has been added in the current IPS Template Policy for inheritance to lower levels (orange) or whether it has been inherited from the higher-level template (green). Only the orange insert points are inherited to lower-level policy elements, so you must add at least one new insert point to each template you create to make the lower-level policies editable. When you add the first new rule to the green insert point, the rule replaces the insert point. Any number of rules can then be added directly above and below that first rule.

Rules defined in the IPS Template Policy itself are not editable in lower-level policies that use the template, with the exceptions of the modifications done in the Rules tree in Inspection rules. Such inherited rules are shown only on your request and they have a grey background. Note that only the actual rules are inherited from a higher-level template into the lower-level policies and templates. The rights to edit policies and IPS Template Policies are defined separately.

Because the sensors and analyzers read rules in order from top down, rules above the insert point in the higher-level template cannot be cancelled by anything a lower-level policy adds into the insert point.

Task 2: Create an IPS Policy

The IPS Policy allows you to define rules directly, but it also gathers together all the rules from the other policy elements (the rules inherited from the IPS Template Policy that is used as the basis of the IPS Policy and rules from one or more IPS Sub-Policies added to the IPS Policy). The IPS Policy is the only policy element that you can transfer to the IPS components (by installing it on a sensor or sensor-analyzer).

Task 3: Create an IPS Sub-Policy

IPS Sub-Policies are sections of IPv4 Access rules that you can insert into IPS Policies, IPS Template Policies, and even other IPS Sub-Policies to make the sensors process traffic faster and to organize the rules. The IPS Sub-Policies are not based on any template. IPv6 Access rules, Inspection rules, and Ethernet rules cannot be inserted into IPS Sub-Policies.

The IPS Sub-Policies may make reading the policies and assigning administrator editing rights easier. For example, you can give some administrators the rights to edit only certain IPS Sub-Policies without giving editing rights to IPS Policies.

An IPS Sub-Policy is inserted into some other policy element by adding a *Jump rule* to the policy element. The Jump rule directs those connections that match the criteria defined in the Jump rule for inspection against the IPS Sub-Policy. When you have already added rules to the policy, one way to insert an IPS Sub-Policy is to select the rules for the IPS Sub-Policy and then an action for creating the IPS Sub-Policy directly in the policy. In that case, the system automatically adds a Jump rule into the policy.

Illustration 8.4 An IPS Sub-Policy in Use

A Jump rule inserts the IPS Sub-Policy, which is shown as an expandable section.

14.1	ANY	Guest-192.168.2.0/24	ANY	Jump My Sub-Policy
14.2	ANY	Boston WWW Server	HTTP	Allow
14.1	ANY	Guest-192.168.2.0/24	ANY	Jump My Sub-Policy
1	net-192.168.10.0/24	Guest-192.168.2.0/24	HTTP FTP	Allow
2	Guest-192.168.1.0/24	Guest-192.168.2.0/24	HTTP FTP	Allow
3	ANY	ANY	ANY	Discard
14.2	ANY	Boston WWW Server	HTTP	Allow

The illustration above shows the same Jump rule in a policy in the collapsed and the expanded state. The rules of the IPS Sub-Policy are shown on a grey background, because they can be edited only within the IPS Sub-Policy itself, not in the IPS Policy that uses the rules.

For example, you could create an IPS Sub-Policy for checking traffic destined to a group of servers located in one particular network. The Jump rule could then use the destination network as a criteria for directing connections to checking against the IPS Sub-Policy. Any connection that was destined to some other network would not get matched against any of the rules in the IPS Sub-Policy. This makes the Access rule processing faster, because the sensor can skip a whole IPS Sub-Policy by comparing a connection to just one simple rule for any non-matching connection. If the IPS Sub-Policy rules were inserted directly into the main IPS Policy, the sensor would have to compare all connections to all those rules individually (since that is then the only way to find out if the connection matches the rules or not). Naturally, the performance benefit gained depends on the number and complexity of the rules that can be placed in an IPS Sub-Policy and how heavily stressed the sensor is to begin with. Therefore, IPS Sub-Policies are mainly useful in the policies of inline sensors that are used extensively for IPv4 packet filtering.

Task 4: Install the Policy

After creating or modifying an IPS Policy, you must transfer the changes to the engines using the Management Client. The target of the action is always the Sensor element, because analyzers automatically receive the policy changes that are relevant to the sensors that send data to the analyzer. You can use the same IPS Policy on any number of sensors.

Each analyzer typically receives events from more than one sensor, so the analyzer configuration can contain rules from several IPS Policies. When processing the events, rules are applied to the correct traffic based on the sensor that sends the information.

Note that the policy transfer contains more information than just the rules in the IPS Policy. Whenever you update the sensor or analyzer configuration or the properties of an element used in the configuration, you must refresh the IPS Policy to make the changes take effect. This

includes, for example, changes in the routing configuration, the Situation elements, and the elements representing the IPS components, even if the changes are not directly related to the rules in the IPS Policy.

If the installation of a policy fails for some reason, the system automatically rolls back to the previously installed configuration.

Using Policy Elements and Rules

The main points of using policy elements are explained in the preceding sections of this chapter. This section illustrates additional points that are useful to know when working with policies and rules.

Validating the Policy

The number of rules in a policy may grow quite large over time. It may become quite difficult, for example, to spot duplicate rules in a policy. To make policy management easier and make sure that the policy does not contain misconfigured rules, you can automatically validate the policy while editing and during the policy installation. There are various different criteria for validating the policy. You can, for example, check the policy for duplicate and empty rules or check if there are rules that cannot ever match traffic.

Continue Rules

The Continue action for a rule sets default options (such as logging options) for the traffic matching process. Options set in Continue rules are used for subsequent rules that match the same criteria as the Continue rule, unless the rules are specifically set to override the options. Continue rules are also very useful in the hierarchical structure of the policies. IPS Template Policies are particularly convenient for setting options with a Continue rule, because all the IPS Policies and IPS Template Policies that use the template inherit the option settings you have specified. Continue rules are explained in detail in [Configuring Default Settings for Several Rules](#) (page 99).

Policy Snapshots

A *Policy Snapshot* is a stored record of a policy configuration. A policy snapshot is stored in an engine's upload history whenever a policy is installed or refreshed on the engine. The Policy Snapshots allow you to check which IPS Policies and other configuration information were uploaded, and when they were uploaded. You can also compare any two policy snapshots and see the differences between them highlighted.

Adding Comments to Rules

Each policy can contain a large number of rules. Adding comments provides administrators with useful information and also makes it easier to read policies. You can add comments to all types of rules. In rule tables, you can add comments in the rule's Comment cell or add dedicated Comment Rule rows between rules. In the Rules tree in the Inspection rules, you can add a comment to each branch of the tree.

When you add a Comment rule to a rule table, a new section is added to the policy. The new section automatically contains all the rules below the Comment Rule until the next Comment Rule in the policy. You can expand and collapse the sections as necessary. The Comment rules are displayed on a colored background (with configurable colors), so they are particularly good for visually separating sections of rules within a single policy.

User Responses

The User Response element allows you to send a configurable server reply to the client instead of just ending the TCP connection when an HTTP connection is terminated or blacklisted. The reply can be a custom error message, or an HTTP redirect to a specified URL. The User Response is selected in the Action options in Access rules and in the Exception in Inspection rules.

Example of Policy Element Use

The example in this section illustrates a common use for the different policy elements in StoneGate and general steps on how the scenario is configured.

Restricting Administrator Editing Rights

Company A is implementing a distributed network with multiple sites, one central office where most of the StoneGate administrators work, and a number of branch offices in different countries. The branch offices mostly have IT staff with only limited networking experience, but who are still responsible for the day-to-day maintenance of the network infrastructure at their site. They must be able to, for example, add and remove Access rules for testing purposes without always contacting the main StoneGate administrators.

In compliance with the company's practices, the StoneGate administrators decide to limit the privileges of the branch office IT staff so that they are not able to edit the policies of the sensors at any of the other sites. The administrators:

1. Create a new IPS Template Policy based on the IPS System Template.
2. Add rules using Alias elements to the template to cover their customizations at all sites with just one policy.
 - Using a common template for all the branch offices eliminates the need to make the same changes in several policies, easing the workload.
3. Create a new IPS Policy based on the new template for each of the branch office sites.
 - Although a single IPS Policy for all sites could work, in this case the administrators decide against it, since separate policies are needed for the separation of editing rights. The

policies are based on the same template, so rules can still be shared without duplicating them manually.

4. Grant each IPS Policy to the correct Sensor elements.
 - After this, only the correct IPS Policy can be installed on each Sensor. No other policy is accepted.
5. Create new accounts with restricted rights for the branch office administrators and grant the correct Sensor element and IPS Policy to each administrator.
 - The branch office administrators are now restricted to editing one IPS Policy and can install it on the correct Sensor.
 - The branch office administrators are not allowed to edit the template the IPS Policy is based on, nor can they install any other policies on any other Sensors.

Administrator rights are explained in more detail in the *Management Center Reference Guide*.

CHAPTER 9

ETHERNET RULES

Ethernet rules are lists of matching criteria and actions that define whether Ethernet protocol traffic is allowed or discarded.

The following sections are included:

- ▶ [Overview to Ethernet Rules](#) (page 84)
- ▶ [Configuration of Ethernet Rules](#) (page 84)
- ▶ [Using Ethernet Rules](#) (page 88)
- ▶ [Examples of Ethernet Rules](#) (page 88)

Overview to Ethernet Rules

Ethernet rules are used only by inline sensors or inline sensor-analyzers in Transparent Access Control mode. The role of the Ethernet rules in IPS is to define which Ethernet protocol packets sensors stop, and which packets are allowed through. Ethernet rules can also be used by sensors in capture mode to define how Ethernet protocol traffic is logged. The Ethernet rules are stored in policy elements, which are discussed in [IPS Policies](#) (page 71).

The traffic matching in Ethernet rules is based on the Source and Destination MAC Address in the packets. Any Ethernet network traffic, such as ARP, RARP, IPv6, Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (SPT), can be checked against the Ethernet rules. When some traffic is found to match an Ethernet rule, the traffic can be allowed or discarded. For sensors in Capture mode, only the Allow action is available.

Regardless of the action taken, a matching rule can also create a log or alert entry.

Configuration of Ethernet Rules

Ethernet rules are configured on the **Ethernet** tab inside IPS Policy and IPS Template Policy elements. IPS Sub-Policies cannot contain Ethernet rules.

Illustration 9.1 Newly Inserted Ethernet Rule - Main Cells

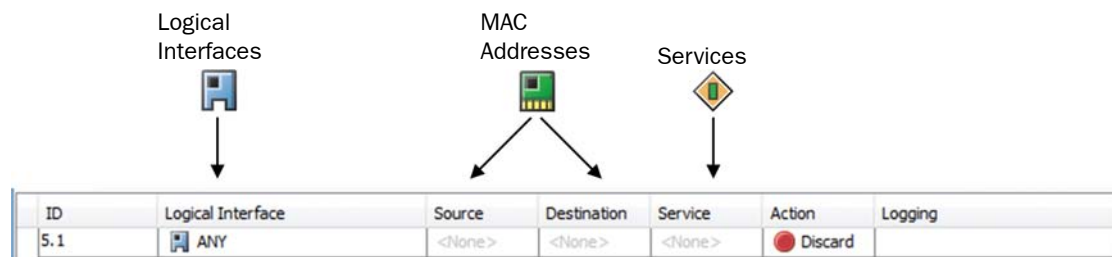
ID	Logical Interface	Source	Destination	Service	Action	Logging
Ethernet rule insert point (before)						
5.1	ANY	<None>	<None>	<None>	Discard	
Allow all						

Sensor applies Action when it finds a match

[Illustration 9.1](#) displays an Ethernet rule that has just been inserted into the policy. The **Source**, **Destination**, and **Service** cells are set to **NONE**, so this rule never matches until they are changed to **ANY** or some more specific value. The **Logical Interface** cell is also matched against traffic, but it is not mandatory to change it if you want the rule to apply regardless of the interface. The other editable cells specify further conditions and options for handling connections that match the cells that are used for traffic matching. It is not necessary to modify all cells in each rule.

Before starting to build policies, make sure you understand the network element types available and how you can use them efficiently to define the resources that you want to protect and control. The illustration below shows the types of elements that you can use in Ethernet rules.

Illustration 9.2 Elements in Ethernet Rules



The table below explains briefly what each Ethernet rule cell does and which elements you can use in the rules. More information on each cell is included in the task flow later in this chapter. The cells are presented in the default order, but you can drag-and-drop them to your preferred order in your own Management Client.

Table 9.1 Ethernet Rule Cells

Cell	Explanation
ID	(Not editable.) Automatically assigned ID number that indicates the order of the rules in the policy. The rules are matched against traffic in the order of the ID numbers.
Logical Interface	Matches the rule based on which interface the traffic is picked up from. The same logical interface may be assigned to one or several interfaces as configured in the properties of the Sensor. This cell accepts only Logical Interface elements.
Source	Elements containing the MAC addresses that the rule matches. Both the Source and the Destination defined must match the source and destination of a packet for the packet to match the rule. The Source and Destination cells accept MAC Address elements.
Destination	
Service	The Services match an Ethernet frame type. The Service cell accepts Ethernet Services elements.
Action	Command for the Sensor to carry out when a connection matches the rule.
Logging	The options for logging.
Comment	Your optional free-form comment for this rule. Note that you can also add separate comment rows in between rules.
Tag	(Not editable.) Automatically assigned unique identification for the rule. Works as a link between the log entries and the rule that has generated the log entries. The rule tag consists of two parts (for example, @20.1). The first part of the tag is permanent and belongs to only that rule. The second part changes when the rule is changed. The first part and the second part are separated by a period.

Considerations for Designing Ethernet Rules

Ethernet rules are read from the top down, and more specific rules must be placed above more general rules that match the same traffic. The actions **Allow** and **Discard** stop the processing from continuing down the rule table for any packet that matches the rule. Therefore, rules with any of these actions must be placed so that the more limited the rule is in scope, the higher up in the rule table it is. If the traffic does not match any of the Ethernet rules by the end of the policy, it is allowed by default.

Default Elements

The IPS Strict Template and the IPS System Template contain predefined Ethernet rules. Because the IPS Strict Template and the IPS System Template are added in the system and updated through dynamic update packages, the templates you currently have in your system may look slightly different from the one that is presented in this section. Newer versions of the templates work in the same way as described below. Any changes to the templates are documented in the Release Notes document for each dynamic update package.

The predefined Ethernet rules allow the most common types of Ethernet traffic.

Illustration 9.3 IPS Strict Template and IPS System Template - Ethernet Rules

ID	Logical Interface	Source	Destination	Service	Action	Logging
Ethernet rule insert point (before)						
2	ANY	ANY	ANY	ARP RARP STP (Span	Allow	
3	ANY	ANY	ANY	IPv4	Allow	
4	ANY	ANY	ANY	IPv6	Allow	
Ethernet rule insert point						
Allow all						

In the illustration above, you can see a green insert point at the top of the rule table, three default rules below it, and then another insert point below them.

- The first rule contains common Ethernet protocols and allows the matching traffic to pass through.
- The second rule contains the IPv4 protocol and allows IPv4 traffic with further inspection against the IPv4 Access rules.
- The third rule contains the IPv6 protocol and allows IPv6 traffic with further inspection against the IPv6 Access rules.

The two insert points indicate where you can add Ethernet rules to an IPS Policy that uses the IPS Strict Template or the IPS System Template. The first insert point above the default rules allows you to modify and make exceptions to the way traffic matching the three default rules is checked. For example, you could add a rule defining that no IPv4 or IPv6 traffic at all is allowed between certain MAC addresses.

The second insert point below the default rules allows you to define how traffic matching other protocols is checked. The final rule in the IPS Strict Template and the IPS System Template allows all traffic.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Define the Source and Destination

The source and destination MAC addresses specified in a rule are compared to the MAC addresses in each packet's header. Based on these and other criteria, the rule is applied to matching packets. By default, these cells are set to NONE, and you must change the value in both cells to make the rule valid.

The Source and Destination cells accept MAC address elements. You can set these cells to ANY to make the rule match all possible source or destination MAC addresses. Also, you can add more than one element in each cell to make the rule match multiple MAC addresses.

Task 2: Define the Service

The Service cell defines which protocol(s) the rule you design applies to. By default, the Service is set to NONE, and you must change the value to make the rule valid. The Service cell accepts only Ethernet Services elements. You can set the Service to ANY to make the rule match all protocols.

Task 3: Select the Action

The Action cell defines what happens when a packet matches the rule. The following actions are available in the Ethernet rules:

- Allow: the traffic is let through the Sensor.
- Discard: the traffic is silently dropped if going through an Inline interface.



Note - When the Allow action is used for IPv4 or IPv6 traffic in the Ethernet rules, the traffic is then checked against the Access rules (IPv4 traffic) or the IPv6 Access rules (IPv6 traffic). The final action for the IPv4 and IPv6 traffic is determined according to traffic type by the Access rules or the IPv6 Access rules.

Task 4: Select Logging Options

By default, the Logging options are undefined, which means that no log entry is issued when the rule matches.

The log levels are as follows:

- None.
- Stored (saved on the Log Server).
- Alert.



Note - A log entry is generated for each packet that matches an Ethernet rule. You must use careful consideration when setting the logging options to avoid producing an excessive amount of log data.

Using Ethernet Rules

You can validate Ethernet rules and add comments to the rules just like for any other types of rules. See [Using Policy Elements and Rules](#) (page 79) for more information.

Examples of Ethernet Rules

The examples in this section illustrate some common modifications to the default Ethernet rules in StoneGate and general steps on how each scenario is configured.

Logging Ethernet Protocol Use

The administrators at Company A have installed a Sensor in Transparent Access Control mode and they want to create some custom Ethernet rules. The administrators know that the majority of traffic uses the IPv4 protocol, but they do not have a clear idea of which other Ethernet protocols are being used in the company's network. They decide to temporarily log the usage of Ethernet protocols, excluding IPv4, to gather data for a report.

To do this, the administrators:

1. Create a new IPS policy based on the IPS System Template to replace the System Policy that they have currently installed.
2. Add a new rule in the Ethernet rules to exclude IPv4 traffic from logging:

Table 9.2 Ethernet Rule for Excluding IPv4 Traffic from Logging

Source	Destination	Service	Action	Options
ANY	ANY	IPv4	Allow	Logging: None

3. Add a rule to log the use of other Ethernet protocols:

Table 9.3 Ethernet Rule for Logging Ethernet Protocol Use

Source	Destination	Service	Action	Options
ANY	ANY	ANY	Allow	Logging: Stored

4. Save and install the policy on the Sensor.
5. View the logs generated by the matches to the Ethernet rules in the Logs view.
6. Create a report based on the log data to help them visualize the patterns of Ethernet protocol use (see the *Management Center Reference Guide* for more information on Reports).
7. Disable the logging Ethernet rule to prevent excess log data from being generated.

Restricting the Use of Ethernet Protocols

Now that the administrators at Company A from the previous example have a clear picture of which Ethernet protocols are being used in the company's network, they want to restrict which protocols are allowed. The administrators determine that ARP and Spanning Tree Protocol (SPT) must be allowed. Since the majority of traffic will use these protocols, the administrators do not want to log matches to the rules that allow specific protocols.

They decide to block the Cisco Discovery Protocol (CDP) on the logical interface named Inline interface because of the security problems it creates, and store log entries of detected CDP use.

To do this, the administrators:

1. Add a new rule in the Ethernet rules to allow ARP, Spanning Tree Protocol (SPT), and IPv4 without producing any logs:

Table 9.4 Ethernet Rule for Allowing ARP and SPT Use

Logical Interface	Source	Destination	Service	Action	Options
ANY	ANY	ANY	ARP SPT IPv4	Allow	Logging: None

2. Add another rule to block the use of Cisco Discovery Protocol (CDP) on the Inline interface, and produce logs that will be stored:

Table 9.5 Ethernet Rule for Blocking CDP Use

Logical Interface	Source	Destination	Service	Action	Options
Inline interface	ANY	ANY	CDP	Discard	Logging: Stored

3. Add a rule on the last line of the Ethernet rules to block the use of other Ethernet protocols without producing logs:

Table 9.6 Ethernet Rule for Blocking Other Ethernet Protocols

Logical Interface	Source	Destination	Service	Action	Options
ANY	ANY	ANY	ANY	Discard	Logging: None

4. Save and install the policy on the Sensor.

CHAPTER 10

ACCESS RULES

Access rules are lists of matching criteria and actions that define which traffic is allowed or discarded, as well as which allowed traffic is inspected against the Inspection rules.

The following sections are included:

- ▶ [Overview to Access Rules](#) (page 92)
- ▶ [Configuration of Access Rules](#) (page 92)
- ▶ [Using Access Rules](#) (page 98)
- ▶ [Examples of IPS Access Rules](#) (page 101)

Overview to Access Rules

The IPv4 and IPv6 Access rules are used only by sensors. Access rules define which traffic is inspected against the Inspection rules, which traffic inline sensors stop, and which traffic is passed through without inspection. The Access rules are stored in policy elements, which are discussed in [IPS Policies](#) (page 71).

The traffic matching in Access rules is based on source IP address, destination IP address, and port information included in the packets. Additional matching criteria that is not based on information in the packets includes the day of the week and the time of day (allowing you to enforce rules during specific times, such as working hours) and the physical network interfaces the traffic is picked up from.

The Access rules provide several different ways to react when some traffic is found to match a rule. You can:

- Allow the traffic with inspection against the Inspection rules.
- Allow the traffic without further inspection.
- Stop the traffic, if the traffic is traversing the inline interfaces of a sensor (requires that the sensor is licensed for the Transparent Access Control feature).

Regardless of which of the above actions is taken, a matching rule can also create a log or alert entry.

Configuration of Access Rules

IPv4 Access rules are configured on the **IPv4 Access** tab, and IPv6 Access rules are configured on the **IPv6 Access** tab inside IPS Policy and IPS Template Policy elements. You can also configure IPv4 Access rules on the **IPv4 Access** tab in IPS Sub-Policy elements. You can create new IPv4 and IPv6 Access rules in the Policy Editing View and also in the Logs view based on one or more selected log entries.

Illustration 10.1 Newly Inserted Access Rule - Main Cells

ID	Logical Interface	Source	Destination	Service	Action	Logging	Time	Comment	Tag
8.1	ANY	<None>	<None>	<None>	Allow				@150.1

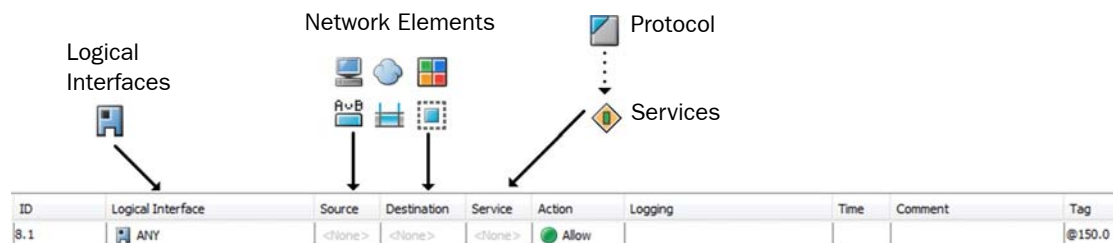
Mandatory cells for matching traffic

Sensor applies Action when it finds a match

The illustration above displays an Access rule that has just been inserted into the policy. The matching cells are set to **<None>** to prevent the rule from having any effect on traffic in case a new rule is added to the policy accidentally. It is not necessary to modify all cells in each rule, but the mandatory cells for traffic matching (**Source**, **Destination**, and **Service**) must be set to some value other than **<None>** for the rule to be valid. The **Logical Interface** cell is also matched against traffic, but it is not mandatory to change its value if you want the rule to apply regardless of the interface. The other editable cells specify further conditions and options for handling connections that match the cells that are used for traffic matching.

Before starting to build policies, make sure you understand the network element types available and how you can use them efficiently to define the resources that you want to protect and control. The illustration below shows the types of elements that you can use in IPv4 and IPv6 Access rules.

Illustration 10.2 Elements in Access Rules



The table below explains briefly what each Access rule cell does and which elements you can use in the rules. More information on each cell is included in the task flow later in this chapter. The cells are presented in the default order, but you can drag and drop them to your preferred order in your own Management Client.

Table 10.1 Access Rule Cells

Cell	Explanation
ID	(Not editable) Automatically assigned ID number that indicates the order of the rules in the policy. The rules are matched against traffic in the order of the ID numbers. For example, the rule 14.3 is the third rule added in this policy to the insert point that is the fourteenth rule in the upper-level template.
Logical Interface	Matches the rule based on which interface the traffic is picked up from. The same logical interface may be assigned to one or several interfaces as configured in the properties of the Sensor. This cell accepts only Logical Interface elements.
Source	Elements containing the IP addresses that the rule matches. Both the Source and the Destination defined must match the source and destination of a packet for the packet to match the rule. The Source and Destination cells accept any elements in the Network Elements category. Network elements used in IPv4 Access rules must contain IPv4 addresses, and Network elements used in IPv6 Access rules must contain IPv6 addresses.
Destination	
Service	The Services match a certain Protocol that defines the protocol for the traffic when it is further inspected against the Inspection rules.
Action	Command for the sensor to carry out when a connection matches the rule, and action-specific options for automatic responses, deep packet inspection (whether traffic is matched against Inspection rules), and blacklisting (<i>IPv4 only</i>).
Logging	The options for logging.
Time	The time period when the rule is applied. If this cell is left empty, the rule applies at all times.

Table 10.1 Access Rule Cells (Continued)

Cell	Explanation
Comment	Your optional free-form comment for this rule. If you add a rule from the Logs view, the Comment cell of the rule automatically includes information on the log entry which was used as the basis of the rule. Note that you can also add separate comment rows in between rules.
Tag	<i>(Not editable)</i> Automatically assigned unique identification for the rule. Works as a link between the log entries and the rule that has generated the log entries. The rule tag consists of two parts (for example, @20.1). The first part of the tag is permanent and belongs to only that rule. The second part changes when the rule is changed. The first part and the second part are separated by a period.

Considerations for Designing Access Rules

One of the crucial issues in designing any policies is the order of the rules. The most important thing to keep in mind when editing the IPS Template Policies, IPS Policies, and IPS Sub-Policies is that the rules are read from top down. The actions **Allow**, **Refuse**, and **Discard** stop the processing from continuing down the Access rule table for any connection that matches the rule. Therefore, rules with any of these actions must be placed so that the more limited the rule is in scope, the higher up in the rule table it is. At its simplest, this principle means, for example, that an Access rule that allows connections to the IP address 192.168.10.200 must be put above an Access rule that stops all connections to the network 192.168.10.0/24. See [Exempting Traffic from Inspection](#) (page 101) for a more detailed example. If the traffic does not match any of the Access rules by the end of the policy, it is allowed by default.

Default Elements

The IPS Strict Template and the IPS System Template have predefined Ethernet rules, IPv4 and IPv6 Access rules, and Inspection rules. Because the IPS Strict Template and the IPS System Template are added in the system and updated through dynamic update packages, the templates you currently have in your system may look slightly different from the one that is presented in this section. Newer versions of the templates work in the same way as described below. Any changes to the templates are documented in the Release Notes document for each dynamic update package.

The predefined IPv4 and IPv6 Access rules are the same in the IPS Strict Template and the IPS System Template. The templates include rules that do not stop any traffic by default, but direct all traffic to be checked against the Inspection rules. The predefined IPv4 Access rules also include a default rule for blacklisting connections that match the current blacklist.

Illustration 10.3 IPS Strict Template and IPS System Template - IPv4 Access Rules

Ethernet		IPv4 Access	IPv6 Access	Inspection					
ID	Logical Interface	Source	Destination	Service	Action	Logging			
1	ANY	D Class E Class Limited Br Loopback	ANY	ANY	Discard	Stored			
2	ANY	ANY	DHCP Bro E Class Loopback	ANY	Discard	Stored			
3	ANY	ANY	ANY	ICMP TCP UDP	Continue				
4	ANY	ANY	ANY	Any TCP Service Any UDP Service	Continue				
5	ANY	ANY	ANY	BrightStor Backup (TCP) BrightStor Engines (TCP) CCSO (TCP) Computer Associates ARCServe CVS DNS (TCP) Finger FTP HTTP HTTP proxy HTTPS IMAP IPP (TCP) LDAP (TCP) McAfee e-Business Server Microsoft Messaging Queuing Group Microsoft-OS MSRPC (TCP) MSSQL (TCP) MySQL NetBIOS-SSN NNTP Oracle TNS POP3 PPTP (TCP) Printer Spooler (TCP) RFB (TCP) Rlogin SIP (TCP) SMTP SNMP (TCP) SNMP Trap (TCP) Squid HTTP proxy SRP (TCP) SSH SVN Telnet WINS Replication X11	Continue				
6	ANY	ANY	ANY	BOOTPS (UDP) CCSO (UDP) DNS (UDP) LDAP (UDP) MSRPC (UDP) MSSQL (UDP) SIP (UDP) SNMP (UDP) SNMP Trap (UDP) TFTP	Continue				
7	ANY	ANY	ANY	GRE with Rematch IP (IP in IP) with Rematch IPv6 Encapsulation with Rematch	Continue				
Access Rules									
9	ANY	ANY	ANY	ANY	Apply Blacklist				
10	ANY	ANY	ANY	ANY	Allow	None			
Allow all									

Illustration 10.4 IPS Strict Template and IPS System Template - IPv6 Access Rules

Ethernet		IPv4 Access	IPv6 Access	Inspection					
ID	Logical Interface	Source	Destination	Service	Action	Logging			
1	ANY	IPv6 Multicast Localhost	ANY	ANY	Discard	Stored			
2	ANY	ANY	IPv6 Unspecified Localhost	ANY	Discard	Stored			
3	ANY	ANY	ANY	TCP UDP	Continue				
4	ANY	ANY	ANY	Any TCP Service Any UDP Service	Continue				
5	ANY	ANY	ANY	BrightStor Backup (TCP) BrightStor Engines (TCP) CCSO (TCP) Computer Associates ARCServe CVS DNS (TCP) Finger FTP HTTP HTTP proxy HTTPS IMAP IPP (TCP) LDAP (TCP) McAfee e-Business Server Microsoft Messaging Queuing Group Microsoft-OS MSRPC (TCP) MSSQL (TCP) MySQL NetBIOS-SSN NNTP Oracle TNS POP3 PPTP (TCP) Printer Spooler (TCP) RFB (TCP) Rlogin SIP (TCP) SMTP SNMP (TCP) SNMP Trap (TCP) Squid HTTP proxy SRP (TCP) SSH SVN Telnet WINS Replication X11	Continue				
6	ANY	ANY	ANY	BOOTPS (UDP) CCSO (UDP) DNS (UDP) LDAP (UDP) MSRPC (UDP) MSSQL (UDP) SIP (UDP) SNMP (UDP) SNMP Trap (UDP) TFTP	Continue				
7	ANY	ANY	ANY	GRE with Rematch IP (IP in IP) with Rematch IPv6 Encapsulation with Rematch	Continue				
IPv6 access rule insert point									
9	ANY	ANY	ANY	ANY	Allow	None			
Allow all									

The illustrations above show the IPv4 and IPv6 Access rules in the IPS Strict Template and the IPS System Template. There are several IPv4 and IPv6 Access rules with various Services defined with Continue as the action and a yellow insert point indicating the place where a Policy that uses the IPS Strict Template or the IPS System Template can be edited.

- The Continue rules above the insert point do not determine if traffic is allowed to pass (the Action is Continue). These rules set parameters for further matching rules. Their task is to ensure that unless otherwise defined in rules added to the insert point, all traffic is inspected against the Inspection rules.
- In the Access rules, the first rule below the Insert point checks the traffic against the current blacklist and terminates the matching connections.
- The last Access rule below the Insert point picks up the parameters of the Continue rules above and, using the parameters set in the Continue rules, directs the traffic to further examination against the Inspection rules.

The Access rules that you add at the insert point in custom policies based on the IPS Strict Template or the IPS System Template are (in most cases) quite specific exceptions to the rules explained above. For example, you could insert a rule there that allows a connection between two particular hosts to continue without any further inspection, or rules for inline sensors to always stop traffic between particular IP addresses and ports.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Define the Source and Destination

The source and destination IP addresses specified in a rule are compared to the IP addresses in each packet's header. Based on these and other criteria, the rule is applied to matching packets. By default, these cells are set to NONE, and you must change the value in both cells to make the rule valid.

The Source and Destination cells accept any of the elements in the Network Elements category. Any of the elements in the Network Elements category that have IPv6 addresses can be used in the Source and Destination cells of the IPv6 Access rules. Groups, Aliases, Address Ranges, and Expressions are especially useful for defining IP addresses in complex scenarios. You can set these cells to ANY to make the rule match all possible source or destination IP addresses. Also, you can add more than one element in each cell to make the rule match several IP addresses.

Task 2: Define the Service

The Service cell defines which protocol(s) the rule you design applies to, which also determines the protocol used in the Inspection rules for matching traffic (the protocol that is detected and selected for traffic by an Access rule is a matching criteria in the Inspection rules). By default, the Service is set to <None>, and you must change the value to make the rule valid.

The Service cell accepts only Service and Service Group elements. There are ready-made Services in the system that cover most needs, but you may also use your own customized versions, for example, to define a non-standard port. The Services available for rule design are categorized according to protocols. You can add more than one element in this cell to make the rule match different types of traffic.

Protocol Agent parameters are available for some Protocols in Service elements. The Protocol Agent parameters are primarily used with firewalls, as Sensors support only some of the Protocol Agents and only some of the parameters available for them.

You can set the Service to ANY to make the rule match all protocols. A previous Continue rule may define a Protocol for traffic allowed by rules that use ANY as the Service (see [Configuring Default Settings for Several Rules](#) (page 99)).

Task 3: Select the Action and Action Options

The Action cell defines what happens when a packet matches the rule. The available actions are explained in [Table 10.2](#).

Table 10.2 Action Field Options

Action	Explanation
Allow	The traffic is let through the sensor.
Continue	Stores the contents of the Options cell and the Protocol option (inside the Service used) in memory and continues the inspection process. Used for setting options for subsequent rules as explained in Configuring Default Settings for Several Rules (page 99).
Discard	The traffic is silently dropped if going through an inline interface. This action requires a that the sensor is licensed for the Transparent Access Control feature.
Refuse	The traffic is dropped if going through an inline interface and an ICMP error message is sent in response to notify the packet's sender. This action requires a that the sensor is licensed for the Transparent Access Control feature.
Jump (IPv4 only)	Matching is continued in the specified sub-policy until a match is found. If there is no matching rule in the sub-policy, the process is resumed in the main policy.
Apply Blacklist (IPv4 only)	Checks the packet against the blacklist according to the options set for this rule. If the packet matches a blacklist entry and is going through an inline interface, the sensor discards the connection.

Each IPv4 and IPv6 Access rule has action-specific options for response and deep packet inspection. IPv4 Access rules also have an option for blacklisting.

The **Response** options are used to define a customized reply to the user for any HTTP connection that is refused, terminated, or blacklisted according to a rule. A default response can be defined using Continue rules, or the response can be specified for an individual rule. The **User Response** element specified in the options allows you define a different response for different situations.

The **Inspection** option determines whether matching traffic is inspected against the Inspection rules (Deep Inspection).

The **Blacklisting** options are used in rules that have **Apply Blacklist** as the action (*IPv4 only*). The options allow you to choose which entries on the blacklist apply to connections that match the rule based on the component that added the blacklist entry on the blacklist. A restriction based on blacklist sender may be necessary, for example, if the same IP address exists in two different networks guarded by two different StoneGate components. The default setting is to take all blacklist entries into account from any other Firewall, Sensor, or Analyzer (that is managed by the same Management Server) regardless of the component that created the entry.

Task 4: Select Logging Options

By default, the **Logging** options are undefined, which means that no log entry is issued when the rule matches (note that this differs from firewall Access rules). If this is what you want, you do not need to define the Logging options.

The log levels are as follows:

- **None**
- **Stored** (saved on the Log Server)
- **Alert**

Task 5: Restrict the Time When the Rule Is Enforced

Optionally, you can set a specific time period when a rule is applied using the **Time** cell. The validity of the rule can be set by month, day of the week, and time of day. For example, you might have certain rules that allow access only during business hours on weekdays. If you leave the **Time** cell empty, the rule is always valid.



Note - The times are entered in Coordinated Universal Time (UTC), and you must adjust the times you enter to make them correspond to the sensor's local time zone. Also consider that UTC time does not adjust to daylight saving time (summer time).

Using Access Rules

The general configuration of Access rules is explained above. The sections below provide further information on configuring Access rules:

- [Allowing System Communications](#) (page 98)
- [Configuring Default Settings for Several Rules](#) (page 99)
- [Rematching Tunneled Packets](#) (page 100)
- [Using Aliases in Access Rules](#) (page 100)

For general information on using rules, see [Using Policy Elements and Rules](#) (page 79).

Allowing System Communications

If NAT is applied to StoneGate system communications, you must create Location elements and add Contact Addresses for the elements to define which translated addresses are necessary for making contact. Only IPv4 addresses are used in system communications.

If you have inline sensors, be careful that you do not define rules that would prevent other StoneGate components from communicating with each other through the sensor. The system communications are detailed in [Default Communication Ports](#) (page 1057).

There are predefined Service elements in the system for all system communications. You can use these in your Access rules as necessary.

Configuring Default Settings for Several Rules

You may want to set default values for some settings in rules to avoid defining the same settings for several rules individually. The **Continue** action is used to set such default values.

The options that can be set using Continue rules in Access rules includes:

- The Protocol option inside the Service used.
- The Logging options.

When a connection matches a rule with Continue as the action, the rule's settings are written in memory but the matching continues until another rule that matches is found. This matching rule uses the defaults set in the Continue rule unless the rule specifically overrides the defaults with different settings. This way, you do not have to define the settings for each rule separately.

You can use Continue rules to set default settings for a general type of traffic and define settings for individual rules only when specifically required. A Continue rule can be overridden by some subsequent Continue rule that has an identical scope (Source, Destination, and Service), or partially overridden by a Continue rule that partially overlaps with the previous Continue rule. When you define Continue rules with different matching criteria, you can have several Continue rules one after another without them interfering with each other in any way at all.

Continue rules are defined in the same way as other rules. However, you must keep in mind that many or even all rules below may be affected. Options in Continue rules are used by the rules below, provided that the Source, Destination, and Service match the same connection as the Continue rule. Continue rules are inherited from Template Policies into lower-level templates and policies like any other rules.

Sub-Policies may require special attention with Continue rules: the Sub-Policies may have different options when you insert them into different policies if the Sub-Policy rules do not override the options set by preceding Continue rules. Also, when a Sub-Policy contains a Continue rule, the options are then used for further matching in the higher-level policy (if the processing returns to the higher-level policy).

Using Continue Rules to Set the Protocol

Default Protocols are set using the Continue action. This way, the correct Protocol is used also for traffic that is allowed by rules that match any Service (and therefore have no particular Service element that would set the correct protocol). The Protocol is needed to associate the traffic with the correct protocol for further inspection and to handle some types of traffic, such as FTP, correctly. The IPS Strict Template and the IPS System Template include several Continue rules that associate all traffic with Protocols according to standard ports.

If you have TCP and UDP services set up in your network under non-standard ports, the traffic may not be associated correctly to the correct protocol and be therefore inspected at a more general (TCP or UDP) level. In this case, you can create your own custom Situation for the traffic and add it in your policy to have the traffic inspected with the correct protocol information. Only some protocols and some of their parameters are supported in the services that are used in IPS policies.

You can also add your own rules for the opposite purpose: to have some traffic not inspected as a particular protocol, but more generally as TCP or UDP traffic. In this case, you add a rule in your policy that includes the general TCP or UDP Service element from the **IP-PROTO** branch of the Services tree.

Rematching Tunneled Packets

If a Sensor inspects traffic that is tunneled using IP-in-IP tunneling or Generic Routing Encapsulation (GRE), the traffic can be checked against IPv4 Access rules and/or IPv6 Access rules several times according to the number and type of layers in the tunnel. For example, when an IPv4 datagram contains an IPv6 datagram, the IPv4 datagram is first matched according to Access rules. If the tunneling Service in the Access rule specifies that the encapsulated IPv6 datagram should be matched again, the contents are then matched against the IPv6 Access rules.

To limit the number of encapsulating layers, the Sensor engine properties define the maximum rematch count. By default, the maximum rematch count is 1. If this count is exceeded, the packet is allowed or discarded according to the setting specified in the Sensor properties and a log or an alert is generated.

Using Aliases in Access Rules

Aliases are one of the most useful tools for reducing the complexity of a policy. In a sense, Aliases are like variables in a mathematical equation—their value changes depending on the component on which they are installed. Because Aliases are able to change their meaning to adapt to local contexts, they can be used to create a single rule that changes in meaning depending on where it is installed. Thanks to Aliases, you can use a single rule to replace multiple, near duplicate rules created separately for each sensor.

To better understand this concept, let us consider an example company, which has its headquarters in Helsinki and branch offices in Atlanta, Munich, Tokyo, and Montreal. Each of these offices has its own Web server. In this scenario, it seems we would require a separate rule or set of rules for each location's Web server.

By using aliases, however, we can create a single rule or set of rules that is still valid in all parts when applied on different components.

The administrator of the example company can create a Web server alias, **\$WebServers**. In the Alias's properties, the administrator defines what **\$WebServers** means for each component. For the sensor in Helsinki, the Web server would be defined as 192.168.1.101, for the sensor in Tokyo as 192.168.2.101, and so on.

When the administrator installs a policy containing the Web server rules with the Alias, the addresses are translated to the correct address on that component. Therefore, when the policy is installed on the Helsinki sensor, the Alias translates to an IP address of 192.168.1.101. The other addresses are not included in the policy that is transferred to that particular engine.

Examples of IPS Access Rules

The examples in this section illustrate some common uses for Access rules and general steps on how each scenario is configured.

Exempting Traffic from Inspection

At Company A, there is a sensor deployed between the general office network and a subnetwork.

Illustration 10.5 Company A's Networks



In the subnetwork, there are several servers that provide services to the general office network as well as the StoneGate Management Server and Log Server. There is also a StoneGate firewall deployed between the internal and external networks. There is heavy traffic to the subnetwork where the internal servers are, so the administrators decide to exempt the log transmissions between the StoneGate firewall and the Log Server from being inspected against the IPS Inspection rules to reduce the Sensor's workload. The administrators:

1. Create a new IPS policy based on the IPS System Template to replace the System Policy that they have currently installed.
2. Add a new rule in the Access rules for their sensor:

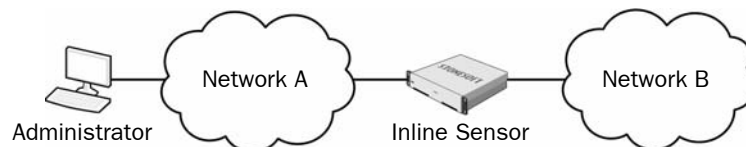
Table 10.3 Access Rule for Exempting Traffic from Inspection Against the Inspection Rules

Source	Destination	Service	Action	Options
Firewall	Log Server	SG Engine to Log	Allow	Deep inspection: Off

Filtering Traffic on an Inline Sensor

Administrators at company B decide that they want more control over which hosts and ports can be used between two networks.

Illustration 10.6 Company B's Network



Hosts in the two networks must be able to communicate between each other using certain specific ports. Additionally, one of the administrators has a workstation connected to Network A. The administrator's workstation must have unrestricted access to Network B. The administrators decide that the inline sensor provides an acceptable level of security at this point between two internal networks.

The administrators:

1. Create elements for network A, network B, and administration host.
2. Add new Access rules for their inline sensor:

Table 10.4 Access Rules for Filtering Traffic

Source	Destination	Service	Action	Options
Administrator Network B	Administrator Network B	ANY	Allow	Logging: Undefined Deep inspection: On
Network A Network B	Network A Network B	Service elements for allowed services	Allow	Logging: Stored Deep inspection: On
ANY	ANY	ANY	Refuse	Logging: Stored Deep inspection: (irrelevant, because dropped traffic is never inspected further)

- Each of the first two rules allows traffic between the Source and the Destination in both directions. The order of the elements within the Source, Destination, and Service cells makes no difference to the outcome of the matching process.
- The order of the rules is important. The rules above proceed correctly from most specific to the least specific. The two first rules must be in this order, because the administrators want all connections from the Administrator host (which is in Network A) to always match the first rule and never the second one, since the rules have different logging options.
- The last of the added rules stops all traffic that is not allowed in the rules above to prevent unauthorized traffic from passing.



Note – If the inline interfaces are on a fail-open network card, traffic passes freely whenever the sensor is offline regardless of what the Access rules state.

CHAPTER 11

INSPECTION RULES

Inspection Rules define how the sensors and analyzers look for patterns in traffic allowed through the Access rules and what happens when a certain type of pattern is found.

The following sections are included:

- ▶ [Overview to Inspection Rules](#) (page 104)
- ▶ [Configuration of Inspection Rules](#) (page 105)
- ▶ [Using Inspection Rules](#) (page 111)
- ▶ [Example of Inspection Rules](#) (page 111)

Overview to Inspection Rules

Inspection rules define how the main traffic analysis is done for traffic that has been allowed and selected for inspection in the IPv4 and IPv6 Access rules. The Inspection rules are stored in policy elements, which are discussed in [IPS Policies](#) (page 71).

Inspection rules examine the entire contents of the packets throughout whole connections to see if the data being transferred contains a pattern of interest. The main source of these patterns are the dynamic update packages that Stonesoft releases, but you can also define new patterns as Situation elements, which are discussed in [Situations](#) (page 63).

There are three general types of cases for using Inspection rules:

- You can detect attempts to exploit known vulnerabilities in your systems and prevent such attempts from succeeding if the system is not patched against it.
- You can monitor traffic that does not cause alarm on the surface, but when examined for certain patterns, may turn out to conceal actual threats. For example, you can detect if a series of occasional service requests are actually someone secretly scanning the network structure or if a spike in traffic is a denial-of-service attack under way.
- You can also detect other sequences in traffic, such as the use of certain applications or even access to a particular file.

Based on the detection results, the Inspection rules provide several different ways to react when some traffic is found to match a pattern of interest:

- Stop the traffic if it is going through an inline sensor.
- Reset the connection.
- Blacklist the connection on one or more StoneGate firewalls and sensors.
- Allow the traffic.

Regardless of which action is taken, a match can also create:

- A log entry with or without recording some of the detected traffic.
- An alert with or without recording some of the detected traffic.

The generated logs (including alert logs) can be monitored and further distilled into various statistical overviews and reports.

Configuration of Inspection Rules

Sensors inspect traffic based on Situation elements, which contain the information about traffic patterns. Patterns may trigger immediate responses or just be recorded. Events that the sensors detect are sent to analyzers for further processing. Analyzers use Correlation Situations, which combine and further analyze the traffic-based findings of sensors to detect additional threats and produce an easy-to-read event stream.

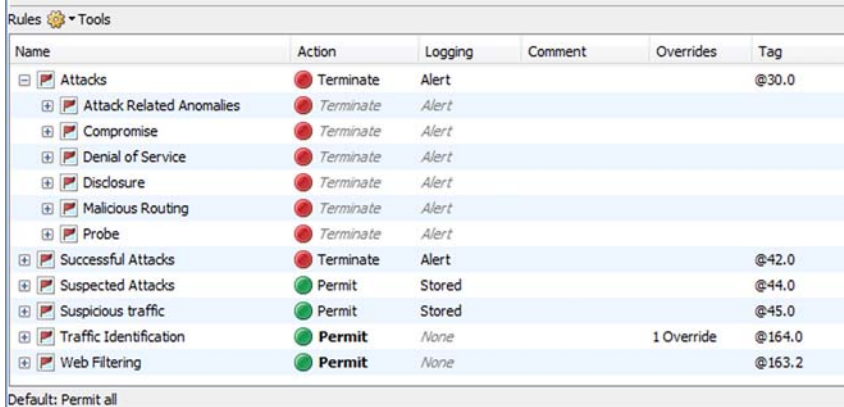
Inspection rules are configured on the **Inspection** tab inside IPS Policy and IPS Template Policy elements. Sub-Policies cannot contain Inspection rules. You can create new Inspection rules in the Policy Editing View and also in the Logs view based on one or more selected log entries.

The Inspection tab has two parts:

- The **Rules** tree contains the main rules for finding traffic patterns. The Rules tree is applied to all traffic that is not handled as Exceptions.
- The **Exceptions** table contains rules that match specific portions of the traffic based on Logical Interface, IP addresses, and Ports. Exceptions have some additional options, and can also set some of those options for the main Rules through the use of *Continue* rules.

The main Rules tree contains a tree of Situations, which are organized under Situation Types. This tree allows you to control which inspection checks trigger a reaction in the system and which checks are ignored. The Rules tree defines general checks that are applied to all patterns that are not handled by a more specific definition. It is not possible to limit the scope of the checks by IP addresses or Logical Interfaces in the Rules tree.

Illustration 11.1 Rules Tree



Name	Action	Logging	Comment	Overrides	Tag
Attacks	Terminate	Alert			@30.0
Attack Related Anomalies	Terminate	Alert			
Compromise	Terminate	Alert			
Denial of Service	Terminate	Alert			
Disclosure	Terminate	Alert			
Malicious Routing	Terminate	Alert			
Probe	Terminate	Alert			
Successful Attacks	Terminate	Alert			@42.0
Suspected Attacks	Permit	Stored			@44.0
Suspicious traffic	Permit	Stored			@45.0
Traffic Identification	Permit	None		1 Override	@164.0
Web Filtering	Permit	None			@163.2

Default: Permit all

The Exceptions are matched before the main rules, which is reflected in location of the Exceptions panel above the main Rules tree. The most frequent use of Exceptions is to eliminate false positives, which typically require permitting a pattern for some part of the traffic while it still triggers a reaction when encountered in any other traffic.

The illustration below shows the Exceptions panel with some rules.

Illustration 11.2 Exceptions Panel

ID	Situation	Severity	Source	Destination	Protocol	Action	Logging
1.1.1	False Positives	ANY	ANY	ANY	ANY	Permit	
1.1.2	URL Whitelist	ANY	net-192.168.1.0/24	ANY	HTTP	Permit	Stored
1.1.3	Web Filtering	ANY	net-192.168.1.0/24	ANY	HTTP	Continue Response: Custom User Responses	Stored

The main matching cell is the Situation that contains the actual patterns. The other matching cells are Logical Interface, Source, Destination, Protocol, and Time. The role of the other matching cells is to limit the scope of the rule to some specific traffic, for example, to take different action based on which host is the sender or receiver of traffic identified as malicious.

The cells are explained in more detail in [Exception Rule Cells](#) (page 108).

Verifying and Tuning Inspection

The most common way to introduce an IPS system in your network is to start with a default policy. Naturally, a general policy that is meant to work in all environments will not be perfectly suited to your particular network scenario, so a tuning period is needed to activate and deactivate inspection checks based on the findings and your particular needs. Tuning the policy is important, since with a small tuning effort, you can save a lot of time due to the increased relevancy and accuracy of the findings that the system generates.

To assist in policy tuning, you can utilize the *passive termination* feature of the sensors. When passive termination is used, the sensor creates a special log entry that notes that a certain connection was selected for termination, but the sensor does not actually terminate the connection. This allows you to check the logs and adjust your policy without the risk of cutting important business communications. There are two levels of activating this feature:

- Passive termination can be activated globally in the Sensor element properties for the initial policy tuning.
- Later on, you can test newly added Situations by setting individual Exception rules to passive termination mode.

For cautious introduction of new Situations introduced in dynamic update packages, you can utilize the Tags for the five most recent updates (**Situations**→**By Tag**→**By Situation Tag**→**Recent Updates**).

Considerations for Designing Inspection Rules

The basic design principle is the same as in other rules: the rules are read from top down, and more specific rules must be placed above more general rules that match the same traffic. The basic layout of the Inspection tab reflects this logic. The detailed rules specific to some IP addresses and Protocols is defined (as Exceptions) at the top, and the general rules that are applied to remaining traffic (the Rules tree) are at the bottom.

The traffic matching in Inspection rules is different from other types of rules, because it is done based on the traffic pattern definitions in Situation elements. The engines monitor the network for all patterns included in the policy. When a pattern is found, the Inspection rules are matched based on the Situation element that contained the found pattern. Inspection rules therefore match certain patterns and non-matching traffic is passed through without any reaction.

The Situation element based configuration logic means that the behavior of the Inspection rules can change without anyone editing the policy directly. Just creating a new Situation in the system may include the Situation in the policy if the Situation is associated with a Situation Tag or Situation Type grouping included in the policy.

The action Permit allows the traffic pattern and the action Terminate stops the traffic that matches the pattern. A Permit action does not unconditionally allow the traffic, because processing still continues to look for other patterns, but a Permit match does prevent the exact same Situation from matching again if it appears at any point further down in the policy.

Example Situation A matches a Permit rule with logging level set to “None”. A second rule that contains Situation A exists below the first rule in the policy with Terminate as the action and logging level set to “Stored”. The logs do not show any matches to Situation A and the traffic that matches the pattern continues uninterrupted.

Similarly, the Terminate action prevents the same Situation from matching again as the policy is processed to the end, but does not prevent other Situations from matching simultaneously.

It is important to note that for the purposes of configuring the system, each Situation element is considered as a unique pattern (with an exception that is discussed below). Avoid defining the exact same pattern in different Situation elements, because such duplicates in the policy can create unintended results and makes the policies difficult to manage.

Example A Continue rule sets a User Response for Situation A, which matches the URL www.example.com. A different rule specifies Termination for Situation B, which also matches www.example.com. When the users access the URL, their connections are terminated without a User Response, because the User Response is set for Situation A and the traffic is terminated by Situation B. The configuration handles these as two separate patterns.

An exception where one Situation is specifically used in the configuration to prevent a different Situation from matching is URL filtering. When you whitelist URLs, the special URL filtering Situations stop further URL-based matching.

Example A web filtering category defined in Situation A prevents users from accessing www.example.com (among other sites). The administrators add www.example.com to a custom Situation B that is permitted higher up in the policy. Users can now access www.example.com. With other types of Situations, matching connections would continue to be terminated if two different Situations were used.

Actual rules may look quite different even if they refer to the exact same Situation, since Situations have grouping mechanisms. However, it makes no difference in matching a pattern whether you add the Situation as a single element or together with other Situations through a Situation Tag or Situation Type.

Correlation Situations may influence the logging and detection results of other Situations. Analyzers use Correlation Situations to combine sensor-detected events and filter out some events. This is useful and necessary, but because some detected events are removed from the log stream, there is a risk that incorrect customizations remove more events than you intend.

Exception Rule Cells

The table below explains briefly what each Exception rule cell does. The columns are presented here in the default order, but you can rearrange them in your own Management Client.

Table 11.1 Exception Rule Cells

Cell	Explanation
ID	(Not editable.) Automatically assigned ID number that indicates the order of the rules in the policy. The rules are matched against traffic in the order of the ID numbers. For example, rule 1.3 is the third rule added in this policy to the insert point that is the first Inspection rule in the upper-level template.
Situation	Defines the patterns of traffic that the rule matches. In addition to individual Situation elements, this cell may contain Situation Type and Tag elements, which are shown as branches in the Situations tree and allow adding the whole branch of Situations at once to a rule. Analyzers use Correlation Situations.
Logical Interface	Limits the rule based on which interface the traffic is picked up from. The same logical interface may be assigned to one or several interfaces as configured in the properties of the Sensor. This cell accepts only Logical Interface elements.
Severity	Limits the rule to those matching Situations that have a severity value within a range you define. This is most useful with rules that include Situation Tags in the Situation cell.
Source	Limit the IP addresses that the rule matches, for example, to create different responses to the same pattern depending on the communicating hosts. The Source and Destination cells accept any elements in the Network Elements branch.
Destination	
Protocol	Limits the Protocols that the rule matches. The protocol is set for traffic in the Access rules in the Service cell of the rule that allows the traffic. The Protocol cell allows you to limit the scope of an Inspection rule based on the protocol that an Access rule has assigned.
Action	Command for the sensor or analyzer to carry out when a connection matches the rule. Action-specific options for blacklisting, connection termination, and user response. The Continue action can be used to set action-specific Action Options for Exceptions and (depending on the option) for the Rules tree as explained in Setting Default Options for Several Inspection Rules (page 111).
Logging	Options for logging.
Time	Limits the time period when the rule is applied. If the cell is empty, the rule applies at all times.
Comment	Your free-form comment for this rule. If you add a rule from the Logs view, the Comment cell of the rule automatically includes information on the log entry which was used as the basis of the rule. Note that you can also section the rules under comment rows.
Tag	(Not editable.) Automatically assigned unique identification for the rule. Works as a link between the log entries and the rule that has generated the log entries. The rule tag consists of two parts (for example, @20.1). The first part of the tag is permanent and belongs to only that rule. The second part changes when the rule is changed. The first part and the second part are separated by a period.

Default Elements

The IPS Strict Template and the IPS System Template contain slightly different definitions for Inspection (the other tabs are identical). Note that because the templates can be updated in dynamic update packages, the templates you currently have in your system may change over time. Changes to the templates are documented in the *Release Notes* for each dynamic update package.

If you base your IPS policies on the ready-made IPS Template Policies (which we recommend), the choice between the IPS Strict Template and the IPS System Templates depends on your network environment and the level of protection that you need. The IPS System Template is suitable in most network environments, whereas the IPS Strict Template is better suited for high-risk network environments, such as data centers.

The rules in the both template are similar. In both templates, Situations classified as Attacks and Successful Attacks are terminated (stopped) and trigger an alert. The difference between the templates is that in the IPS Strict Template, Suspected Attacks are also terminated with an alert, whereas the IPS System Template only logs Suspected Attacks.

Suspected Attacks are traffic patterns that are consistent with malicious activities, but do not match a specific pattern of a known attack. Suspected Attacks can catch zero-day attacks (attacks that are not yet publicly known), but may sometimes prevent some legitimate traffic from passing through if the traffic pattern happens to resemble malicious activities.

Note that rules set to terminate connections create a warning during the policy installation if you install the policy on a sensor that has no inline interfaces. This is only to notify you that matching connections cannot be terminated and does not prevent the system from carrying out the other parts of the rule.

Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.



Note – Keeping your system up-to-date with latest dynamic updates is an essential part of maintaining your Inspection rules. See the *Online Help* for information on dynamic updates and instructions for enabling automatic update download and activation.

Task 1: Create a Policy

To customize inspection, you must have a Custom Policy element. The pre-defined templates in the system are a good starting point for your own customizations. Even if you want to extensively customize the Inspection rules, the default rules on the other tabs are still very useful to keep, because they direct traffic to be inspected at the more advanced levels.

Policy elements, are discussed in detail in [IPS Policies](#) (page 71).

Task 2: Activate the Relevant Inspection Checks

Traffic patterns of interest are defined in Situations, so the inspection checks are based on selecting the desired reaction to the Situations when the pattern is found in network traffic. It is not mandatory to create any additional Situations to activate inspection checks, since there are many default Situation elements in the system and they are continuously updated through dynamic update packages that you can activate in your system.

The Rules tree is the main tool that allows you to select which traffic patterns are permitted and stopped, whether a log entry or an alert is triggered, and whether matching traffic is recorded. All Rules in the Rules tree can be edited, including overrides that have been set in a higher-level template. The Rules tree can contain a maximum of one instance of each Situation, so the definitions within the Rules tree do not overlap.

Task 3: Define the Exceptions

The Exceptions table allows you to create detailed rules, which are processed before the Rules tree definitions. The Exceptions have additional features compared to the Rules tree:

- You can make exceptions to the general Rules tree definitions based on Logical Interface, Source, Destination, and Protocol information.
- You can set options for connection termination (including resets and User Responses) and blacklist entry creation in addition to the options that are available in the Rules tree. The Response options define an automatic client notification for any HTTP connection that is terminated. The blacklisting options define how the blacklist entry is generated and to which inline sensor or firewall it is sent.
- You can create Continue rules to set Action options and Logging options for other Exceptions and the Rules tree. The Rules tree contains specific definitions for logging, so the logging options set with Continue rules do not affect traffic that matches the Rules tree.
- You can create rules in Policy Templates that cannot be changed in the inheriting policies.
- You can create rules that are applied only on certain days and/or times of day.

In addition to individual Situation elements, the Situation cell may contain Tag and Situation Type elements, which are shown as branches in the Situations tree and allow adding the whole branch of Situations at once to a rule. Most of the Situations you add to the policy are those that you consider to be producing false positives in your environment (for example, Situations for exploit attempts against an operating system that is not used in your organization).

In Exceptions, it is highly unusual to set the Situation cell to ANY. This is not useful in most cases because the patterns that Situations define range widely from Situations that detect something as benign as the use of particular applications to something as worrying as successful attacks on a servers. This also creates a lot of unnecessary load on the engines, as a high number of Situations is checked in each matching connection.

Task 4: Eliminate False Positives

As the Inspection rules are matched to traffic, there are always some occurrences of false positives (matches that are incorrect or irrelevant in your environment). By tuning the IPS policy to the actual traffic and applications in your network environment, you can increase the relevancy of inspection results greatly. To eliminate a false positive, you will need to further adjust either the Rules tree or the Exceptions depending on whether the change should be applied globally or to traffic between specific hosts. An easy way to create new Exceptions is

using an existing log entry as the basis (you can create Exceptions through the right-click menu of IPS log entries). See the [Eliminating a False Positive](#) example for a practical overview of one approach to eliminating a false positive.

Task 5: Add Custom Inspection Checks

If you want to detect some pattern in traffic that is not defined in the predefined Situations (for example, a particular internal file server in your network being accessed) or if you want to create a modified version of some existing Situation, you can create a new Situation element. This is explained in [Configuration of Situations](#) (page 64). You can add your custom Situations to the Rules tree by selecting a Situation Type for them.

Using Inspection Rules

For general information on using rules, see [Using Policy Elements and Rules](#) (page 79).

Setting Default Options for Several Inspection Rules

You may want to set default settings for some rules to avoid defining the same settings for several rules individually. The Continue action in Exception rules is used to set such default options. In Inspection rules, all settings in the Action Options and the Logging cell can be set using Continue rules. However, the Rules tree ignores any logging options set with Continue rules. In the Rules tree, the rules either inherit the logging settings from a higher level in the tree or define a specific logging option as an override.

Otherwise, the Continue rules in Inspection rules work in the same general way as those in Access rules. See [Configuring Default Settings for Several Rules](#) (page 99).

Example of Inspection Rules

The example in this section illustrates a common modification to the default Inspection rules in StoneGate and general steps on how the scenario is configured.

Eliminating a False Positive

The StoneGate administrators in this example have just installed a new IPS system with the predefined System Policy. When they command the Sensor online, they soon start receiving alerts.

After some investigation, the administrators realize that the alert is caused by a custom-built application, which communicates in a way that happens to match the pattern of how a certain exploit would be carried out by an attacker. The custom-built application is only used by a few specific servers in the internal network, so they decide to modify the IPS policy so that they no longer receive notifications if this exploit is detected in communications between those servers.

The administrators:

1. Create Host elements to represent the servers.
2. Create a Group element that includes the Host elements.
 - The administrators name the Group so that it is immediately clear from the name that the Group contains the servers that run their custom-built application. This makes the new rule easier to read than if they included the hosts directly in the rule.
3. Create a new custom policy based on the IPS System Template.
4. Add the following rule in the Exceptions on the Inspection tab of the new custom policy:

Table 11.2 Inspection Exception for a False Positive Between Specific Hosts

Situation	Source	Destination	Action	Options
The Situation element that is mentioned in the alerts in the Logs view.	The Group defining the internal servers.	The Group defining the internal servers.	Permit	Logging: None

- The Exceptions are processed before the main Rules tree. This rule can therefore override the rule that triggers the alerts.
 - If the Situation matches traffic between any other hosts than those included in the Group, the IP address does not match the ones defined in the new rule. The processing will continue to the Rules tree, in which a match will trigger an alert.
 - The logging would not have to be set to None, because it is the default option, but the administrators want to do so anyway to make sure any rules they add in the future cannot accidentally set logging on for this rule.
5. Install the new policy on the Sensors.

CHAPTER 12

PROTOCOL AGENTS

Protocols of the *Protocol Agent* type are special modules for some protocols and services that require advanced processing. Protocol Agents can enforce policies on the application layer.

The following sections are included:

- ▶ [Overview to Protocol Agents](#) (page 114)
- ▶ [Configuration of Protocol Agents](#) (page 115)
- ▶ [Using Protocol Agents](#) (page 116)
- ▶ [Examples of Protocol Agent Use](#) (page 121)

Overview to Protocol Agents

Protocol Agents are IPS modules for advanced processing of some protocols that require such handling. They can be used to extend the sensor's Access rules with proxy-like application layer features. Protocol Agents are also used to associate traffic with a certain protocol for inspection against the Inspection rules.

Protocol Agents can be used for:

- Opening related connections when required (for example, FTP data connections).
- Validating application-level protocol use (for example, FTP command syntax).

Some protocols always require the use of the correct Protocol Agent to pass the IPS inspection.

Connection Handling

Some protocols require special handling on the sensor due to their complexity, address information in the data payload, related connections, and so on. Protocol Agents are provided to handle related connections for the following protocols:

- FTP with related active and passive data connections.
- Microsoft RPC (MSRPC) for Microsoft Exchange and Outlook communications.
- Oracle TNS protocol communications.
- Remote Shell (RSH) protocol communications.
- TFTP file transfers.

Example File Transfer Protocol (FTP) uses two related connections: a control connection and a separately established data connection. If the control connection is allowed without the Protocol Agent, the sensor does not recognize that the data connection is part of an existing connection and handles it as a new connection (which usually leads to failed data transfer).

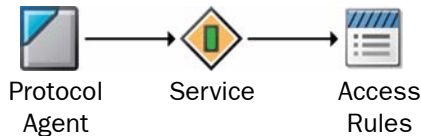
Protocol Validation

Protocol Agents can be used to validate communications against standards of specific protocols. How this exactly works depends on the protocol in question. Protocol Agents provide complete protocol decode and validation for the following protocols: FTP, HTTP, HTTPS, ICMP, IMAP, IPv4, IPv6, MSRPC, NBT, POP3, SIP, SMB, SMTP, SSH, TCP, and UDP.

Configuration of Protocol Agents

The Protocol Agents are represented in the Management Client by Protocol elements that have *Protocol Agent* as their type. Other Protocol elements are of the type *Protocol Tag*. The Protocol Agent elements represent software modules on the sensor that you activate by using a corresponding Protocol element in your configuration.

Illustration 12.1 Using Protocol Agents



As seen in [Illustration 12.1](#), Protocol Agents are not placed directly in IPS policies. They are used inside custom Service elements that you create and which can then be used in Access rules. Whenever a rule that contains a Service with an associated Protocol Agent matches, the Protocol Agent is automatically activated.

All Protocol Agents in the system are default elements and you cannot change them or add any new ones. To customize Protocol Agents for specific needs, you must use the Protocol Agent in a custom Service element you create and set parameters for the Protocol Agent in the properties of the Service. The Protocol Agent parameters can be accessed in the properties of Services that you have associated with a Protocol Agent.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* or the *Administrator's Guide* PDF, in the section called **Elements**.

Task 1: Create a Custom Service with a Protocol Agent

There are default Service elements in the system that refer to Protocol Agents. These default Services can be used without additional configuration in the Access rules. However, the default Services do not allow you to change the default parameters of Protocol Agents that have configurable parameters. If you want to modify how a Protocol Agent behaves, you must create a new custom Service of your own and attach the correct Protocol Agent to that Service.

The Service element contains the identifying information, such as a port number, that determines which traffic the Service matches. In most cases, this alone ensures that the Protocol Agent is not applied to the wrong type of traffic.

Task 2: Set Parameters for the Protocol Agent

If you create your own custom Service that uses a Protocol Agent, you can specify parameters for the Protocol Agent in the properties of the Service. The Protocol Agents and the parameters you can set are listed in [Table 12.1](#).

Task 3: Insert the Service in Access Rules

Whether you create a custom Service or use one of the predefined Services that have a Protocol Agent attached to them, you must define the traffic the Protocol Agent handles in the Access rules in your IPS Policies.

A Protocol Agent can be set either on a rule-by-rule basis, or you can create a rule with Continue as the rule's Action. When there is a Continue rule, rules further down in the rule table that match the same traffic (same source and destination) use the Protocol Agent defined in the Continue rule. With Protocol Agents, the Continue rule affects only rules where the Service is set to ANY. More specific definitions override the defaults set by Continue rules, as all Service elements specify that either some particular Protocol Agent or no Protocol Agent is used. Some protocols may require a Protocol Agent if Connection Tracking is enabled for the rule. Those protocols may not be allowed by a rule that has ANY as its Service unless a Protocol Agent is configured using a previous matching Continue rule.

Since Protocol Agents validate traffic against the specifics of a particular protocol, you must ensure that a Service with a Protocol Agent is not applied to traffic that does not use that protocol. Also, Protocol Agents are designed for particular types of uses, so they may not always be appropriate even if the protocol matches. See below for details of what each Protocol Agent does.

Using Protocol Agents

There are Protocol Agents for many different protocols, and the purpose of each Protocol Agent depends on the particular demands the protocol in question places on the sensor. This section describes the available Protocol Agents and lists the configurable parameters that they add to Services that use them. When the description below states "There are no configurable parameters for this Protocol Agent", the Protocol Agent does not have any options, but may still have a control for turning the Protocol Agent on/off in the Service (this control is meant for StoneGate IPS, which may require the Protocol element without the Protocol Agent features in some situations).

The following Protocol Agents and parameters are available on Sensors:

Table 12.1 Protocol Agents Used by Sensors

Name	Protocol Validation	Deep Inspection	Allow Related Connections	Protocol Agent Parameters
FTP , see FTP Agent (page 118)	yes	yes	yes	yes
GRE , see GRE Agent (page 118)	yes	yes	yes	yes
H.323 , see H.323 Agent (page 118)	no	yes	no	n/a
HTTP , see HTTP Agent (page 118)	yes	yes	n/a	yes
HTTPS , see HTTPS Agent (page 118)	no	yes	n/a	yes

Table 12.1 Protocol Agents Used by Sensors (Continued)

Name	Protocol Validation	Deep Inspection	Allow Related Connections	Protocol Agent Parameters
ICMP , see ICMP Agent (page 118)	yes	no	n/a	n/a
IPv4 , see IPv4 Agent (page 118)	yes	no	n/a	n/a
IPv4 Encapsulation , see IPv4 Encapsulation Agent (page 118)	yes	yes	n/a	yes
IPv6 , see IPv6 Agent (page 119)	yes	no	n/a	n/a
IPv6 Encapsulation , see IPv6 Encapsulation Agent (page 119)	yes	yes	n/a	yes
MSRPC , see MSRPC Agent (page 119)	yes	yes	yes	n/a
NetBIOS , see NetBIOS Agent (page 119)	no	yes	n/a	n/a
Oracle , see Oracle Agent (page 119)	no	yes	yes	yes
Remote Shell (RSH) , see Remote Shell (RSH) Agent (page 119)	no	yes	yes	yes
Services in Firewall , see Services in Firewall Agent (page 120)	n/a	n/a	n/a	n/a
SMTP , see SMTP Agent (page 120)	yes	yes	n/a	n/a
SSH , see SSH Agent (page 120)	yes	yes	n/a	n/a
SunRPC , see SunRPC Agents (page 120)	no	yes	no	n/a
TCP , see TCP Proxy Agent (page 120)	no	yes	no	n/a
TFTP , see TFTP Agent (page 120)	no	yes	n/a	n/a

FTP Agent

One of the most common ways to transfer files across networks is using the File Transfer Protocol (FTP). An FTP session starts with a control connection (by default, TCP port 21), and the communications continue using a dynamically allocated port. The Protocol Agent keeps track of the actual ports used so that ports can be opened only as needed for specific connections. This way, the whole range of possible dynamic ports does not need to be allowed in the policy. The FTP Protocol is platform-independent.

This agent has parameters you can set in the Service properties.

GRE Agent

The Generic Routing Encapsulation (GRE) protocol is a tunneling protocol that allows the encapsulation of network layer packets inside IP tunneling packets. This agent has parameters you can set in the Service properties.

H.323 Agent

H.323 defines a set of protocols as well as the components and procedures for real-time multimedia communication. H.323 consists of a series of different types of standards related to video and audio services, real-time transport, control channels, security, etc.

There are no configurable parameters for this Protocol Agent.

HTTP Agent

The **HTTP** agent can be used to log the URLs from HTTP requests. This agent has parameters you can set in the Service properties.

HTTPS Agent

The **HTTPS** agent can be used for identifying encrypted HTTPS traffic for HTTPS decryption and inspection in the Access rules, and for identifying encrypted HTTPS traffic for inspection in the Inspection rules. This agent has parameters you can set in the Service properties.

ICMP Agent

The Internet Control Message Protocol (ICMP) is used by the operating systems of networked computers to send error messages. There are no configurable parameters for this Protocol Agent.

IPv4 Agent

The IPv4 Agent provides protocol inspection for IPv4 traffic. There are no configurable parameters for this Protocol Agent.

IPv4 Encapsulation Agent

The IPv4 Encapsulation Agent provides protocol inspection for tunneled IPv4 traffic. This Protocol Agent specifies rematching parameters for IPv4 packets encapsulated in IPv6 packets.

This agent has parameters you can set in the Service properties.

IPv6 Agent

The IPv6 Agent provides protocol inspection for IPv6 traffic. There are no configurable parameters for this Protocol Agent.

IPv6 Encapsulation Agent

The IPv6 Encapsulation Agent provides protocol inspection for tunneled IPv6 traffic. This Protocol Agent specifies rematching parameters for IPv6 packets encapsulated in IPv4 packets.

This agent has parameters you can set in the Service properties.

MSRPC Agent

The MSRPC Protocol Agent supports the handling of related connections in MSRPC (Microsoft Remote Procedure Call) traffic.

The supported end-point mapper (EPM) connection method between the Outlook client and the Exchange server is TCP. By default, the Microsoft RPC/EPM service is available at port 135/TCP and the communications continue using a dynamically allocated port. The Protocol Agent keeps track of the actual ports used, so that the range of dynamic ports does not need to be allowed in the policy.

There are no configurable parameters for this Protocol Agent on the IPS.

NetBIOS Agent

This Protocol Agent provides deep inspection for Windows NetBIOS Datagram Service connections. There are no configurable parameters for this Protocol Agent.

Oracle Agent

This Protocol Agent handles Oracle Transparent Network Substrate (TNS) protocol-based SQL*Net, Net7, and Net8 connections. It is meant for cases where TCP port 1521 is used only for negotiating the port number for Oracle database connections, and the port number for the actual connection is assigned dynamically.

This Protocol Agent is needed only if the database is located on a different computer than the Oracle listener. The Oracle Protocol Agent does not modify payload data because the database service connections could go through a different route than the listener connection. You can create custom Oracle agents with different settings when required.

Remote Shell (RSH) Agent

Remote Shell (RSH) is a widely used remote management protocol. This Protocol Agent manages Remote Shell connections and RExec connections. RExec is a remote protocol with which it is possible to run commands in another computer.

This agent has parameters you can set in the Service properties.

Services in Firewall Agent

This Protocol Agent is used with services running on StoneGate firewall nodes in the same system as the sensor. It is only intended for the system's internal use. There are no configurable parameters for this Protocol Agent.

SMTP Agent

The Simple Mail Transfer Protocol (SMTP) Protocol Agent provides protocol validation and deep inspection. There are no configurable parameters for this Protocol Agent.

SSH Agent

Secure Shell (SSH) is an encrypted remote use protocol. This Protocol Agent validates the communications to make sure the protocol used really is SSH. The SSH Agent validates SSHv1 only. There are no configurable parameters for this Protocol Agent.

SunRPC Agents

StoneGate provides both UDP and TCP based Protocol Agents for Sun Remote Procedure Call (RPC) protocol. These Protocol Agents provide deep inspection.

The SunRPC Protocol Agents collect information about RPC services by interpreting the GET PORT and DUMP PORTS requests and their respective answers. All information they collect is stored in the Portmapper cache.

When the packet filter needs to evaluate RPC matches, it consults the Portmapper cache to check if the destination of the packet has the appropriate service defined in the rule. If the cache does not have the requested information available, the packet under evaluation is not let through and a query is sent to the destination host for RPC information. The information received is stored in cache.

There are no configurable parameters for these Protocol Agents.

TCP Proxy Agent

The TCP Protocol Agent is a proxy agent used for TCP connections that need to be closed after a certain amount of idle time. Certain TCP based applications do not properly handle the closing of connections, and leave them open for a long period of time, unnecessarily consuming resources. For such situations, the TCP proxy agent can be used to actively close the connections after a certain idle time. In addition, the TCP Proxy Agent may abort a connection if the closing of the connection does not complete in a specified period of time.

There are no configurable parameters for this Protocol Agent on the IPS.

TFTP Agent

The Trivial File Transfer Protocol (TFTP) Agent performs data transfer from a server to a client using dynamically selected ports. There are no specific limits to the port range in the TFTP protocol (RFC 1350).

A TFTP Agent is attached to a UDP connection established between the client and the server. The client opens the control connection from a dynamically selected source port to the fixed destination port 69/UDP on the server. A separate UDP data connection is established between the client and the server after the client has sent a read or write command to the server. The server opens a data connection from a dynamic source port to the client's destination port, which is the same as the one used as the source port of the control connection.

There are no configurable parameters for this Protocol Agent on the IPS.

Examples of Protocol Agent Use

The examples in this section illustrate some common uses for Protocol Agents in StoneGate and the general steps on how each scenario is configured.

Preventing Active Mode FTP

Company A has an FTP server that allows access from the Internet. According to company policy, the sensor must restrict users to passive mode FTP connections.

The administrators:

1. Create a new Service element for passive FTP
2. Attach the FTP Protocol Agent to the Service.
3. Change active mode FTP setting to **No** in the Service properties.
4. Create an Access rule that allows users to connect to the FTP server using their custom-made Service element.

Logging URLs Accessed by Internal Users

Company B has decided to keep track of which web pages the employees visit. In addition to logging the connections, the administrators also want to log URLs. The access is currently controlled by an Access rule that allows all outbound connections from the internal networks to the Internet regardless of the service, so the administrators decide to add the HTTP Protocol Agent in a Continue rule.

The administrators:

1. Add the Continue rule above the existing Access rule as shown in [Table 12.2](#):

Table 12.2 Example Rules for Company B

Source	Destination	Service	Action
Internal Networks	Expression "NOT Local Protected Sites"	"HTTP (with URL Logging)" Service	Continue
Internal Networks	Expression "NOT Local Protected Sites"	ANY	Allow

- Using the "NOT Local Protected Sites" expression requires that the Alias "Local Protected Sites" has been configured with a translation value for the sensor in question.
2. Refresh the policy on the sensor.

CHAPTER 13

TLS INSPECTION

The TLS Inspection feature decrypts HTTPS connections so that they can be inspected for malicious traffic, and then re-encrypts the traffic before sending it to its destination.

The following sections are included:

- ▶ [Overview to TLS Inspection](#) (page 124)
- ▶ [Configuration of TLS Inspection](#) (page 125)
- ▶ [Using TLS Inspection](#) (page 127)
- ▶ [Examples of TLS Inspection](#) (page 128)

Overview to TLS Inspection

HTTPS is used to secure HTTP connections. When a web browser connects to a server that uses HTTPS, the browser and the server negotiate an encryption algorithm, which is used to create the encrypted connection. The server sends a certificate that is signed by a certificate authority to authenticate its identity to the web browser.

However, the encrypted HTTPS connection can also be used to obscure web-based attacks. TLS Inspection allows you to decrypt HTTPS traffic so that it can be inspected.

Strict TPC inspection mode is automatically applied to TCP connections when TLS inspection is used. See [TCP Inspection Modes](#) (page 54) for more information.

The TLS Inspection feature consists of server protection, which inspects incoming connections to servers in the protected network, and client protection, which inspects HTTPS outgoing connections initiated by clients in the protected network.

When an HTTPS server in the internal network is the destination of an incoming connection, the sensor uses the server's credentials to decrypt and re-encrypt the traffic.

When a client in the internal network initiates a connection to an external HTTPS server, the sensor checks whether the server's certificate was signed by a certificate authority that is considered trusted. If the certificate was signed by a trusted certificate authority, the engine makes a new certificate that matches the server's certificate. From the point of view of a user in the internal network, the process is invisible: the connection is established in the same way as a connection made directly to an HTTPS server.

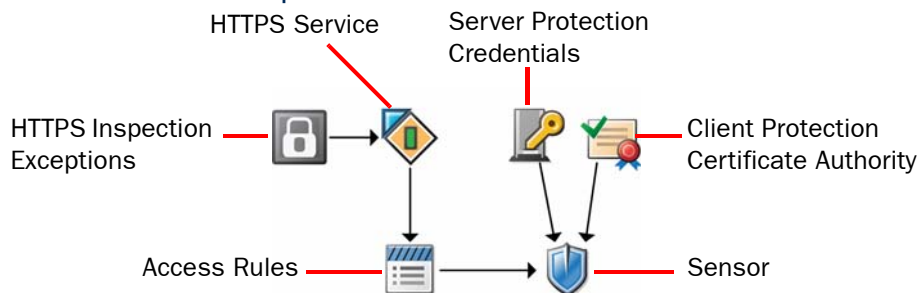
When a server's certificate is self-signed or has not been signed by a trusted certificate authority, the engine cannot trust the server certificate. In this case the engine makes a new self-signed certificate. This certificate is presented to the user in the internal network, and the user's browser shows the same warning it would show if it received a self-signed certificate directly from an HTTPS server. In this case, the user must decide whether or not to accept the certificate.

In both cases, the engine adds a Netscape Certificate Comment to the Extensions in the certificate to indicate that the certificate is a dynamically created certificate for StoneGate SSL/TLS deep inspection. Substituting the original server certificate allows the sensor to decrypt and re-encrypt the traffic.

After decrypting the traffic, normal HTTP inspection is applied, and if the traffic is allowed to continue, it is re-encrypted before forwarding it.

Configuration of TLS Inspection

Illustration 13.1 Elements in TLS Inspection



The Server Protection Credentials and the Client Protection Certificate Authority are specified in the properties of the sensor that provides TLS Inspection. The sensor uses the private key and certificate stored in the Server Protection Credentials to decrypt traffic to and from HTTPS servers in the protected network for inspection. The Client Protection Certificate Authority contains a private key and a certificate. The sensor uses the private key stored in the Client Protection Certificate Authority to sign the certificates presented to the user's Web browser, and the certificate to negotiate encrypted connections with HTTPS servers.

The HTTPS Inspection Exceptions element is a list of domains that are excluded from decryption and inspection. The HTTPS Inspection Exceptions can be specified in the Protocol Parameters of a custom HTTPS Service, which is used in the Access rules to select HTTPS traffic for inspection.

Default Elements

There are predefined Trusted Certificate Authority elements that represent the signing certificates of major certificate authorities. Default Trusted Certificate Authority elements are automatically added to the system from dynamic update packages and cannot be edited or deleted. When client protection is used, the engine checks whether the certificate of an external server was signed by one of the Trusted Certificate Authorities. You can also create your own Trusted Certificate Authority elements to represent other certificate authorities that the engine should consider trusted.

Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Create Server Protection Credentials Elements

If you want to inspect traffic for which an internal HTTPS server is the destination, you must create a Server Protection Credentials element to store the private key and certificate of the server. The private key and certificate allow the sensor to decrypt HTTPS traffic for which the internal server is the destination so that it can be inspected.

Task 2: Create Client Protection Certificate Authority Elements

If you want to inspect traffic between a client in the internal network and an external HTTPS server, you must create a Client Protection Certificate Authority element that contains the credentials the engine uses to sign the certificate it generates. You can import an existing private key and certificate, or generate a new private key and certificate. You can also configure the Client Protection Certificate Authority to check the certificate revocation list (CRL) on a CRL server.

You must configure users' web browsers to trust certificates signed using the credentials in the Client Protection Certificate Authority element to avoid excessive warnings or error messages about invalid certificates.

Task 3: Specify TLS Inspection Options in the Sensor Properties

In the Sensor properties, you specify the Client Protection Certificate Authority (if you want to inspect traffic between internal clients and external servers), and the Server Protection Credentials (if you want to inspect traffic for which an internal server is the destination). Depending on the options you specify, you can configure only client protection, only server protection, or both client and server protection.

Task 4: Create an HTTPS Inspection Exceptions Element

Traffic to and from some Web sites that use HTTPS may contain users' personal information that is protected by laws related to the privacy of communications. Decrypting and inspecting this traffic may be illegal in some jurisdictions. The HTTPS Inspection Exceptions optionally allow you to define a list of domains for which traffic is not decrypted. Connections to the specified non-decrypted domains are allowed to pass through the sensor without being decrypted.

Task 5: Create a Custom HTTPS Service

To inspect HTTPS traffic, you must create a custom HTTPS Service and enable HTTPS decryption and inspection in the Protocol Parameters. You also can optionally specify the HTTPS Inspection Exceptions in the Protocol Parameters.

Task 6: Create an Access Rule

The HTTPS traffic to be inspected is specified in the Access rules. Traffic that matches the Access rule is decrypted and inspected in the same way as regular HTTP traffic according to the Inspection rules.

To mark traffic for TLS inspection, you must use the custom HTTPS Service that you created, and deep inspection must be enabled in the rule. See [Access Rules](#) (page 91) for more information about the Access Rules.

Using TLS Inspection

The general configuration of TLS Inspection is explained above. This section provides further information on configuring TLS Inspection.

Security Considerations

Because the HTTPS communications mediated by the engine are decrypted for inspection, and because the private keys of the servers are stored in the Server Protection Credentials elements on the Management Server, you must carefully consider security precautions when using TLS Inspection. The following recommendations are general guidelines for ensuring the security of the engine and the Management Center:

- Run the Management Server on a hardened operating system.
- Disable SSH access to the engine's command line if it is not needed regularly.
- Ensure that the engine's Control interface is in a controlled network.
- Save Management Server backups as encrypted files.

Sensor Deployment

TLS Inspection requires two separate secure connections: one from the client to the sensor, and one from the sensor to the server. For this reason, sensors must be deployed in inline mode to use TLS Inspection. TLS Inspection cannot be done for traffic picked up through Capture interfaces.

TLS inspection cannot be used on redundant single inline sensors deployed alongside a firewall cluster using dispatch clustering. In dispatch clustering, traffic is received by one node in the firewall cluster. The node forwards the traffic to the other firewall node. This can result in a situation where one of the single inline sensors only receives one direction of the traffic and the other single inline sensor receives both directions of the traffic. If one sensor has created substitute certificates, and traffic is dispatched through a different sensor without passing through the sensor that created the substitute certificates, the connection fails.

For more information about sensor deployment, see [StoneGate IPS Deployment](#) (page 27).

Examples of TLS Inspection

The examples in this section illustrate some common uses for TLS Inspection in StoneGate and general steps on how each scenario is configured.

Server Protection

Company A's web server offers HTTPS services to their customers. The administrators want to be able to detect and block attacks targeting the HTTPS server that are encrypted inside an SSL tunnel. They decide to configure TLS Inspection to decrypt and inspect traffic to and from the HTTPS server.

The administrators do the following:

1. Create a Server Protection Credentials element and import the private key and certificate of the HTTPS server.
2. Select the Server Protection Credentials in the Sensor properties.
3. Create a custom HTTPS Service and enable HTTPS decryption and inspection in the Protocol Parameters.
4. Create Access rules with the custom HTTPS Service as the Service.
5. Use the Inspection rules from the IPS System Template to look for attacks in HTTP traffic.
6. Save and install the policy.

Client Protection

The administrators also want to detect and block Web-based attacks targeting the web browsers of users in Company A's network to protect the workstations and internal networks. However, employees at Company A often use online banking services that are secured with HTTPS, and these connections should not be inspected. The administrators decide to configure TLS Inspection to detect and block attacks that are encrypted inside an SSL tunnel, and use an HTTPS Inspection Exceptions list to exclude the online banking domains from decryption and inspection.

The administrators do the following:

1. Create Client Protection Certificate Authority elements and generate a new certificate and private key. Outside of StoneGate, the administrators add the Client Protection Certificate Authority they created to the list of trusted certificate authorities in the users' Web browsers.
2. Select the Client Protection Certificate Authority in the Sensor properties.
3. Create an HTTPS Inspection Exceptions element and add the domain names for the online banking Web sites that are excluded from decryption.
4. Create a custom HTTPS Service, and enable HTTPS decryption and inspection and specify the HTTPS Inspection Exceptions in the Protocol Parameters.
5. Create Access rules with the custom HTTPS Service as the Service.
6. Use the Inspection rules from the IPS System Template to look for attacks in HTTP traffic.
7. Save and install the policy.

CHAPTER 14

WEB FILTERING

Web filtering compares the URLs that end-users attempt to open to a list of URLs, which can be defined manually or through pre-analyzed and categorized addresses. When a match is found, you can configure the system to respond in the various ways allowed by Inspection rules.

The following sections are included:

- ▶ [Overview to Web Filtering](#) (page 130)
- ▶ [Configuration of Web Filtering](#) (page 130)
- ▶ [Examples of Web Filtering](#) (page 132)

Overview to Web Filtering

Web filtering can prevent end-users from intentionally or accidentally accessing most web sites that are objectionable (based on the content they contain) or potentially harmful (for example, phishing and malware sites). This type of content filtering can increase network security and enforce an organization's policy on acceptable use of resources.

In web filtering, the engines compare the URLs (uniform resource locators) in web browser page requests against a list of forbidden URLs. There are two ways to define the forbidden URLs:

- You can define a small number of blacklisted URLs manually according to your own criteria.
- You can filter access according to a supplied URL categorization scheme (for example, filter out 'adult content').

Both methods can be used together. You can also define whitelisted URLs manually if a useful site happens to be included in a category of URLs that you want to otherwise ban.

The URL categorizations are provided by an external service. At this time, the BrightCloud service is supported. BrightCloud provides categories for malicious sites as well as several categories for different types of non-malicious content you may want to filter or log. Using category-based filtering with BrightCloud is a license-controlled feature.

The categories allow you to configure policies based on the types of sites to ban instead of manually typing in URLs. The individual URLs included in the categories are updated continuously, so they are fetched dynamically from the categorization service. The individual URLs are not viewable in the Management Client except when a match is found in traffic and the match is logged. The engines query the actual URLs from the external URL categorization service to access up-to-date URL listings. Different responses can be taken when a URL match is found: for example, you can log the matches or block the traffic. If you decide to block traffic, the sensor can additionally notify the end-user with a custom message that the end-users see in their browsers instead of the page they tried to open.

When Web filtering is used, Strict TCP inspection is automatically applied to TCP connections. See [TCP Inspection Modes](#) (page 54).

Configuration of Web Filtering

Illustration 14.1 Elements in the Configuration



The Web Filtering feature is configured through Stonesoft-supplied Web Filtering Situations and/or manual URL lists. The Inspection rules are configured in the same basic way as other Inspection rules. However, Web Filtering Situations can uniquely be configured to directly override other Situations to whitelist some URLs manually (as explained further in this chapter).

Since the URLs that are included in category-based filtering are defined dynamically by an external service, it is not possible for you to manually add new categories or edit the existing ones. New web filtering situations are added through dynamic updates as necessary.

Default Elements

There are default elements for the categories you can use in web filtering. These are represented by a specific type of Situation elements, which can be found under **Situations**→**By Type**→**Web Filtering** in the element tree and in the corresponding branch of the Rules tree in the Inspection rules.

The Context for manually defining lists of URLs is **HTTP URL Filter** (under **Application Protocols**→**HTTP** when selecting a Context for a Situation).

The Situations that represent web filtering categories have a distinctive blue color so that you can easily spot them in the rules. URL lists that you create yourself carry the standard red icon.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Prepare the Sensor

Category-based web filtering requires that the engine is licensed to use the BrightCloud categorization service.

You must also define DNS server addresses in the Sensor elements so that the engines can contact the BrightCloud servers.

Task 2: Create User Response Messages

Optionally, you can define customized User Responses for web filtering matches, such as a custom HTML page that is displayed in the end-user's browser when a connection is blocked.

Task 3: Blacklist/Whitelist Individual URLs

The HTTP URL Filter Situation Context allows you to create Situations that blacklist or whitelist URLs that you manually define. There is only one type of list for both uses. Whether a particular list is a blacklist or a whitelist depends on the action you configure for it in the Inspection rule.

Task 4: Configure Web Filtering Rules in the Policy

The Inspection rules define how different categories and lists of URLs are matched to traffic and what kind of reaction a match sets off. Both manual lists and the dynamically-updated categories are introduced in the policy in the form of Situation elements. See [Inspection Rules](#) (page 103) for more information on Inspection rule configuration.

Different Web Filtering features require you to adjust either the main Rules tree or Exceptions. The Web Filtering branch in the Rules tree contains all category-based filters by default, making it easy to activate filtering for content categories and subcategories. Whitelists must be configured as Exceptions. Blacklists can be configured as parts of the Rules tree or as Exceptions depending on your needs. User Responses are configured in Exceptions. You can use the Continue action to set User Response options for other Exceptions and the Rules tree.

The available categories may change when you activate a new dynamic update package, and be automatically enforced after the next policy upload (depending on the Rules tree settings).

Examples of Web Filtering

Allowing a Blocked URL

The company is using category-based web filtering. Among other categories, the administrators have blocked end-users from viewing web sites categorized as “Questionable” in the Rules tree. However, now one of the network security administrators notices that they are blocked from accessing a hacker-oriented site that they have occasionally browsed to research new security threats. To make an exception for their own use, the administrators:

1. Create a new Situation called “Web Filtering Whitelist” with the Context “HTTP URL Filter” and type in the URL of the hacker site they want to access.
2. Add the following type of new Exception rule.

Table 14.1 New Rule for Allowing a URL Above the Previously Added Category-Based Rule

Situation	Source	Destination	Action
Custom “Web Filtering Whitelist” Situation	Administrators’ workstations	ANY	Permit

CHAPTER 15

BLACKLISTING

Blacklisting is a way to temporarily block unwanted network traffic either manually or with blacklist requests from a Sensor, Analyzer, or Management Server. Blacklisted connections are blocked for the duration of blacklist entries, after which the connections are again allowed.

The following sections are included:

- ▶ [Overview to Blacklisting](#) (page 134)
- ▶ [Configuration of Blacklisting](#) (page 135)
- ▶ [Using Blacklisting](#) (page 137)
- ▶ [Examples of Blacklisting](#) (page 138)

Overview to Blacklisting

Blacklisting makes it possible to block unwanted network traffic for a specified time. Sensors, Analyzers, and Management Servers can send blacklist requests to firewalls or Inline sensors with the Transparent Access Control (TAC) feature. Analyzers can add IP addresses to the blacklist based on detected events. You can also blacklist IP addresses manually.

Risks of Blacklisting

Blacklisting can have unintended consequences that could disrupt business-critical traffic. Use blacklisting with careful consideration. The following two categories represent the typical risks associated with blacklisting:

Table 15.1 Risks of Blacklisting

Risk	Explanation
Blacklisting legitimate connections (false positive)	If the defined pattern for detecting malicious traffic is inaccurate, legitimate traffic may sometimes be blacklisted. This causes service downtime for hosts that are incorrectly identified as malicious.
Causing self-inflicted denial-of-service (DoS)	When an attacker uses spoofed IP addresses, a different (legitimate) IP address may be blacklisted instead of the attacker's IP address. This may cause a self-inflicted denial-of-service on legitimate traffic.

These risks can be minimized with good planning. The threats must be identified and evaluated carefully, and the active responses must be defined only with good reasons.



Note – Use blacklisting with consideration. An attacker may use spoofed IP addresses, which may cause a self-inflicted denial-of-service on legitimate traffic.

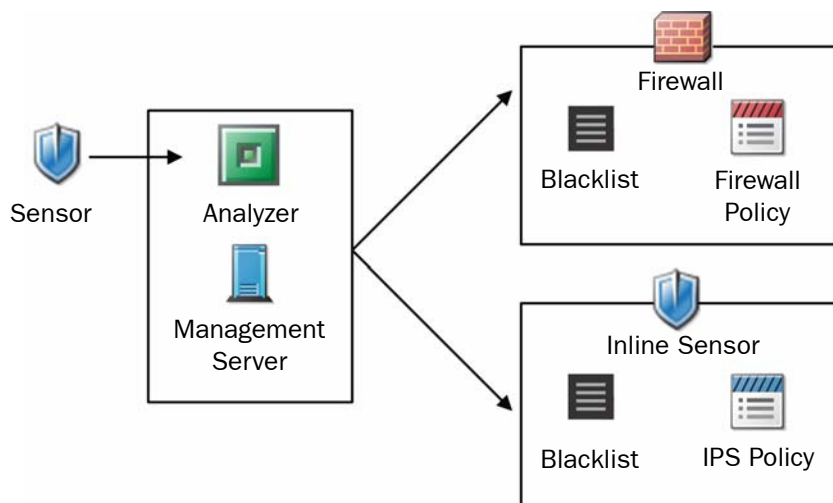
Whitelisting

Whitelisting means defining a list of IP addresses that must never be blacklisted. In Stonegate, whitelisting is achieved by following general Access rule design principles. Blacklisting applies only at the position of the blacklisting Access rule(s) in the policy. Traffic that has already been allowed or discarded before the blacklisting rules is not affected by blacklisting. Only traffic that matches the blacklisting Access rules is blacklisted. If an Access rule in the firewall's policy allows a connection, an Access rule that refers to the blacklist further down in the policy cannot blacklist the connection.

Configuration of Blacklisting

Blacklisting is executed as defined in the Access rules of the IPS or Firewall Policy, and automatic blacklisting requests are sent as defined in the Exceptions on the Inspection tab of the IPS policy.

Illustration 15.1 Blacklisting Configuration



In [Illustration 15.1](#), Sensors, Analyzers, and Management Servers send blacklist requests. When Sensors send blacklisting requests, analyzers relay the request to the component that enforces the blacklisting. When an Inline Sensor is both the sender and the executor of a blacklisting request, the request is sent through the analyzer back to the Inline Sensor. Manual blacklisting commands from the administrators are sent through the Management Server.

Firewalls or Inline sensors can execute blacklist requests generated automatically by sensors and analyzers that are managed by the same Management Server. The duration of the blocking is defined when the automatic blacklist entry is created. It is based on the value configured in the Exception on the Inspection tab that generates the blacklist entry (firewalls do not automatically create blacklist entries).

There is one blacklist per Firewall or Inline Sensor. When traffic matches a blacklisting Access rule, the current blacklist entries on the Firewall or Inline sensor are checked. The traffic is discarded if any of the current blacklist entries matches the traffic. If the traffic does not match the blacklisting Access rule or its related blacklist entries, the next Access rule in the policy is checked as usual. Each blacklist entry exists only for a defined time period after which the entry is cleared and matching traffic is again allowed.

Access rules in the firewall's or sensor's policy define which connections are matched to the blacklist.

Configuration Workflow

The following sections provide an overview of the configuration tasks. Detailed step-by-step instructions can be found in the *Online Help* of the Management Client and the *Administrator's Guide* PDF, in the section called **Policies**.

Task 1: Define Blacklisting in the Access Rules

Blacklisting is applied in the Firewall or IPS policy with Access rules that contain the Apply Blacklist action. By default, all Sensors, Analyzers, and Management Servers are allowed to send blacklist requests. You can also restrict the allowed blacklisters to certain elements in the Access rule's options.

No further configuration is needed for manual blacklisting. Tasks 2 and 3 explain the other steps needed for configuring automatic blacklisting with StoneGate IPS.

Task 2: Define Analyzer-to-Firewall or Analyzer-to-Sensor Connections

The Analyzer or Management Server connects to the StoneGate Firewall or Inline Sensor to send the blacklist requests. The IPS Strict Template and the IPS System Template contain an Access rule that allows the blacklisting connections to the Firewall or Sensor. If your policy is not based on the IPS Strict Template or the IPS System Template, or if you have deleted the rules that allow the Analyzer and the Management Server to send blacklist requests to the Firewall or Sensor, you may need to add Access rules that allow the blacklisting connections.

Task 3: Define Exceptions in the IPS Policy

Blacklist scope options in Exceptions on the Inspection tab trigger blacklisting for the detected events. Blacklisting scope options can be defined for any type of Exception, including rules that use Correlation Situations. Blacklisting is defined using the detected event's IP source and destination addresses, and optionally the TCP or UDP ports. If the event does not contain this information, a blacklist entry cannot be created. Netmasks can optionally be used to blacklist the detected event's network.

Using Blacklisting

Blacklisting is needed whenever the Firewall is unable to determine whether traffic is harmful and relies on a separate IPS to tell the difference. An inline Sensor combines these functions, and in many cases is able to block attack attempts alone without the need to use blacklisting. However, blacklisting is useful in the following cases:

- The traffic latency requirements are too strict for an Inline Sensor. A non-inline Sensor analyzes the traffic off-line and therefore does not cause any delays to legitimate traffic.
- The offending connection is not the only one that the administrators want to block. If the IPS detects that a business-critical application server has been compromised, the desired reaction may be to shut down the whole network until the intruder has been cleared out. This may require blacklist requests to several Firewalls.
- A Firewall is already in a suitable place, and therefore adding the non-inline Sensor is easier than implementing an inline Sensor.

Manual Blacklisting

You can blacklist traffic manually through the Management Client. This requires that the Firewall policy has an Access rule which applies blacklisting and allows Firewall(s) to accept blacklist requests from the Management Server. There are three ways to create new blacklist entries manually. You can blacklist a connection found in the log data, define a new blacklist entry for a Firewall element, or create new blacklist entries in the Blacklist view.

Monitoring Blacklisting

The currently active blacklisting entries on the Firewall can be monitored in the Blacklist view.

Blacklist monitoring does not show you which connections are actually dropped. Blacklist monitoring only shows you the addresses that are currently on the blacklist. The Access rule(s) that apply blacklisting in the Firewall policy determine if any of these connections are dropped. The Logs view can show which connections are actually dropped, depending on the logging options you have set. The blacklist can be sorted and filtered in the same way as logs. Blacklist entries can be removed and added manually.

Examples of Blacklisting

The examples in this section illustrate some common uses for blacklisting in StoneGate and the general steps on how each scenario is configured.

Blacklisting Traffic from a Specific IP Address Manually

Company A is using StoneGate Sensors and Analyzers. The company starts getting a large amount of spam from IP address X. The company's administrators decide to blacklist all traffic originating from that IP address for two hours. To do this, the administrators:

1. Create a new Access rule in the IPS policy. The Access rule applies blacklisting and allows the Management Server to send blacklist requests to the Sensor(s).
2. Refresh the Firewall policy on all Firewalls.
3. Open the Logs view.
4. Select one of the entries that originate from IP address X.
5. Create a new blacklist entry that sets two hours as the Duration of the blacklist entry and defines the Sensor(s) that will receive the blacklist request.

Automatic Blacklisting with IPS

Company B is using a Firewall and IPS managed by the same Management Server. The IPS has recently detected a large number of attempted attacks against several of the company's servers. The attempted attacks have come from multiple IP addresses. It is difficult to predict which server will be the target of a particular attack. The administrators want to automatically block traffic between the IP address that is the source of the attacks and the target server for one day whenever an attempted attack is detected.

There is already a default Situation for attempted attacks that defines the source IP address of matching traffic as the Attacker and the destination IP address as the Victim. To configure the automatic blacklisting for traffic from the attacker to the company's servers, the administrators:

1. Create a new Access rule in the Firewall policy. The rule applies blacklisting and allows any component to send blacklist requests to the Firewall.
2. Refresh the Firewall policy.
3. Create an Exception on the Inspection tab of the IPS policy that sends a blacklist request to the Firewall when traffic matches the Situation for attempted attacks.
4. Define the following Blacklist Scope properties in the options of the Exception:
 - Block traffic between endpoints
 - Duration: 1 day
 - Endpoint 1 Address: Attacker
 - Endpoint 2 Address: Victim
 - Blacklist Executor: Firewall
5. Refresh the IPS policy.

APPENDICES

In this section:

Command Line Tools - 141

Default Communication Ports - 147

Predefined Aliases - 153

Situation Context Parameters - 157

Regular Expression Syntax - 163

SNMP Traps and MIBs - 177

TCP/IP Protocol Headers - 193

ASCII Character Codes - 197

Glossary - 201

Index - 231

APPENDIX A

COMMAND LINE TOOLS

This appendix describes the command line tools available on StoneGate IPS engines. For instructions on how to access the command line, see the *Administrator's Guide* or the *Online Help* of the Management Client.

The following sections are included:

- ▶ [StoneGate-Specific Commands](#) (page 142)
- ▶ [General Tools](#) (page 145)

StoneGate-Specific Commands

StoneGate engine commands can be run from the command line on the sensors and analyzers. For a full list of command line tools for all types of components, see the *Command Line Tools* appendix in the *Administrator's Guide* or the *Online Help* of the Management Client.

Table A.1 StoneGate-specific Command Line Tools on Engines

Command	Description
<pre> sg-blacklist show [-v] [-f <i>FILENAME</i>] add [[-i <i>FILENAME</i>] [src <i>IP_ADDRESS/MASK</i>] [dst <i>IP_ADDRESS/MASK</i>] [proto {<i>tcp udp icmp NUM</i>}] [srcport <i>PORT</i>{-<i>PORT</i>}] [dstport <i>PORT</i>{-<i>PORT</i>}] [duration <i>NUM</i>]] del [[-i <i>FILENAME</i>] [src <i>IP_ADDRESS/MASK</i>] [dst <i>IP_ADDRESS/MASK</i>] [proto {<i>tcp udp icmp NUM</i>}] [srcport <i>PORT</i>{-<i>PORT</i>}] [dstport <i>PORT</i>{-<i>PORT</i>}] [duration <i>NUM</i>]] iddel <i>NODE_ID ID</i> flush </pre>	<p>Can be used to view, add, or delete active blacklist entries. The blacklist is applied as defined in Access Rules.</p> <p>Commands:</p> <p>show displays the current active blacklist entries in format: engine node ID blacklist entry ID (internal) entry creation time (internal) address and port match originally set duration (internal) (internal). Use the -f option to specify a storage file to view (<i>/data/blacklist/db_<number></i>). The -v option adds operation's details to the output.</p> <p>add creates a new blacklist entry. Enter the parameters (see below) or use the -i option to import parameters from a file.</p> <p>del deletes the first matching blacklist entry. Enter the parameters (see below) or use the -i option to import parameters from a file.</p> <p>iddel <i>NODE_ID ID</i> removes one specific blacklist entry on one specific engine. <i>NODE_ID</i> is the engine's ID, <i>ID</i> is the blacklist entry's ID (as shown by the show command).</p> <p>flush deletes all blacklist entries.</p> <p>Add/Del Parameters:</p> <p>Enter at least one parameter. The default value is used for the parameters that you omit. You can also save parameters in a text file; each line in the file is read as one blacklist entry.</p> <p>src <i>IP_ADDRESS/MASK</i> defines the source IP address and netmask to match. Matches any IP address by default.</p> <p>dst <i>IP_ADDRESS/MASK</i> defines the destination IP address and netmask to match. Matches any IP address by default.</p> <p>proto {<i>tcp udp icmp NUM</i>} defines the protocol to match by name or protocol number. Matches all IP traffic by default.</p> <p>srcport <i>PORT</i>[-<i>PORT</i>] defines the TCP/UDP source port or range to match. Matches any port by default.</p> <p>dstport <i>PORT</i>[-<i>PORT</i>] defines the TCP/UDP destination port or range to match. Matches any port by default.</p> <p>duration <i>NUM</i> defines in seconds how long the entry is kept. Default is 0, which cuts current connections, but is not kept.</p> <p>Examples:</p> <pre> sg-blacklist add src 192.168.0.2/32 proto tcp dstport 80 duration 60 sg-blacklist add -i myblacklist.txt sg-blacklist del dst 192.168.1.0/24 proto 47 </pre>

Table A.1 StoneGate-specific Command Line Tools on Engines (Continued)

Command	Description
<pre> sg-bootconfig [--primary-console=<i>tty0/ttyS</i> <i>PORT,SPEED</i>] [--secondary-console= [<i>tty0/ttyS</i> <i>PORT,SPEED</i>]] [--flavor=<i>up/smp</i>] [--initrd=<i>yes/no</i>] [--crashdump=<i>yes/no/Y@X</i>] [--append=<i>kernel options</i>] [--help] apply </pre>	<p>Can be used to edit boot command parameters for future bootups.</p> <p>--primary-console=<i>tty0/ttyS PORT,SPEED</i> parameter defines the terminal settings for the primary console.</p> <p>--secondary-console= [<i>tty0/ttyS PORT,SPEED</i>] parameter defines the terminal settings for the secondary console.</p> <p>--flavor=<i>up/smp [-kdb]</i> parameter defines whether the kernel is uniprocessor or multiprocessor.</p> <p>--initrd=<i>yes/no</i> parameter defines whether Ramdisk is enabled or disabled.</p> <p>--crashdump=<i>yes/no/Y@X</i> parameter defines whether kernel crashdump is enabled or disabled, and how much memory is allocated to the crash dump kernel (Y). The default is 24M. X must always be 16M.</p> <p>--append=<i>kernel options</i> parameter defines any other boot options to add to the configuration.</p> <p>--help parameter displays usage information.</p> <p>apply command applies the specified configuration options.</p>
<pre> sg-clear-all </pre>	<p>Use this only if you want to return a StoneGate appliance to its factory settings.</p> <p>Clears all configuration from the engine. You must have a serial console connection to the engine to use this command.</p>
<pre> sg-contact-mgmt </pre>	<p>Used for establishing a trust relationship with the Management Server as part of engine installation or reconfiguration (see <i>sg-reconfigure</i> below). The engine contacts the Management Server using the one-time password created when the engine's initial configuration is saved.</p>
<pre> sg-logger -f <i>FACILITY_NUMBER</i> -t <i>TYPE_NUMBER</i> [-e <i>EVENT_NUMBER</i>] [-i "<i>INFO_STRING</i>"] [-s] [-h] </pre>	<p>Can be used in scripts to create log messages with the specified properties.</p> <p>-f <i>FACILITY_NUMBER</i> parameter defines the facility for the log message.</p> <p>-t <i>TYPE_NUMBER</i> parameter defines the type for the log message.</p> <p>-e <i>EVENT_NUMBER</i> parameter defines the log event for the log message. The default is 0 (H2A_LOG_EVENT_UNDEFINED).</p> <p>-i "<i>INFO_STRING</i>" parameter defines the information string for the log message.</p> <p>-s parameter dumps information on option numbers to stdout</p> <p>-h parameter displays usage information.</p>

Table A.1 StoneGate-specific Command Line Tools on Engines (Continued)

Command	Description
<pre>sg-raid [-status] [-add] [-re-add] [-force] [-help]</pre>	<p>Configures a new hard drive on a StoneGate appliance. This command is only available for StoneGate appliances that support RAID (Redundant Array of Independent Disks) and have two hard drives.</p> <ul style="list-style-type: none"> -status option displays the status of the hard drive. -add options adds a new empty hard drive. Use -add -force if you want to add a hard drive that already contains data and you want to overwrite it. -re-add adds a hard drive that is already partitioned. This command prompts for the drive and partition for each degraded array. Use -re-add -force if you want to check all the arrays. -help option option displays usage information.
<pre>sg-reconfigure [--boot] [--no-shutdown]</pre>	<p>Used for reconfiguring the node manually.</p> <ul style="list-style-type: none"> --boot option applies bootup behavior. Do not use this option unless you have a specific need to do so. --no-shutdown option allows you to make limited configuration changes on the node without shutting it down. Some changes may not be applied until the node is rebooted.
<pre>sg-status [-l] [-h]</pre>	<p>Displays information on the engine's status.</p> <ul style="list-style-type: none"> -l option displays all available information on engine status. -h option displays usage information.
<pre>sg-toggle-active SHA1 SIZE --force [--debug]</pre>	<p>Switches the engine between the active and the inactive partition. This change takes effect when you reboot the engine. You can use this command, for example, if you have upgraded an engine and want to switch back to the earlier engine version. When you upgrade the engine, the active partition is switched. The earlier configuration remains on the inactive partition. To see the currently active (and inactive) partition, see the directory listing of <code>/var/run/stonegate</code> (<code>ls-l /var/run/stonegate</code>).</p> <p>The <code>SHA1 SIZE</code> option is used to verify the signature of the inactive partition before changing it to active. If you downgrade the engine, check the checksum and the size of the earlier upgrade package by extracting the signature and size files from the <code>sg_engine_[version.build]_i386.zip</code> file.</p> <ul style="list-style-type: none"> --debug option reboots the engine with the debug kernel. --force option switches the active configuration without first verifying the signature of the inactive partition.
<pre>sg-version</pre>	<p>Displays the software version and build number for the node.</p>

Table A.1 StoneGate-specific Command Line Tools on Engines (Continued)

Command	Description
sginfo [-f] [-d] [-s] [-p] [--] [--help]	Gathers system information you can send to Stonesoft support if you are having problems. Use this command only when instructed to do so by Stonesoft support. -f option forces sgInfo even if the configuration is encrypted. -d option includes core dumps in the sgInfo file. -s option includes slapcat output in the sgInfo file. -p option includes passwords in the sgInfo file (by default passwords are erased from the output). -- option creates the sgInfo file without displaying the progress --help option displays usage information.

General Tools

The table below lists some general operating system commands that may be useful in running your StoneGate engines. Some commands can be stopped by pressing Ctrl+c.

Table A.2 General Command Line Tools on Engines

Command	Description
dmesg	Shows system logs and other information. Use the -h option to see usage.
halt	Shuts down the system.
ip	Displays IP-address related information. Type the command without options to see usage. Example: type ip addr for basic information on all interfaces.
ping	Tool for sending ICMP echo packages to test connectivity. Type the command without options to see usage.
ps	Reports status of running processes.
reboot	Reboots the system. Upon reboot, you enter a menu with startup options. For example, this menu allows you to return the engine to the previous configuration.
scp	Secure copy. Type the command without options to see usage.
sftp	Secure FTP (for transferring files securely). Type the command without options to see usage.
ssh	SSH client (for opening a terminal connection to other hosts). Type the command without options to see usage.
tcpdump	Gives information on network traffic. Use the -h option to see usage.
top	Displays the top CPU processes taking most processor time. Use the -h option to see usage.

APPENDIX B

DEFAULT COMMUNICATION PORTS

This chapter lists the default ports used in connections between StoneGate components and the default ports StoneGate uses with external components.

The following sections are included:

- ▶ [Management Center Ports](#) (page 148)
- ▶ [IPS Engine Ports](#) (page 150)

Management Center Ports

The illustrations below present an overview to the most important default ports used in communications between the Management Center (SMC) components and from the SMC to external services. See the table below for a complete list of default ports.

ILLUSTRATION B.1 Destination Ports for Basic Communications Within SMC Management Client

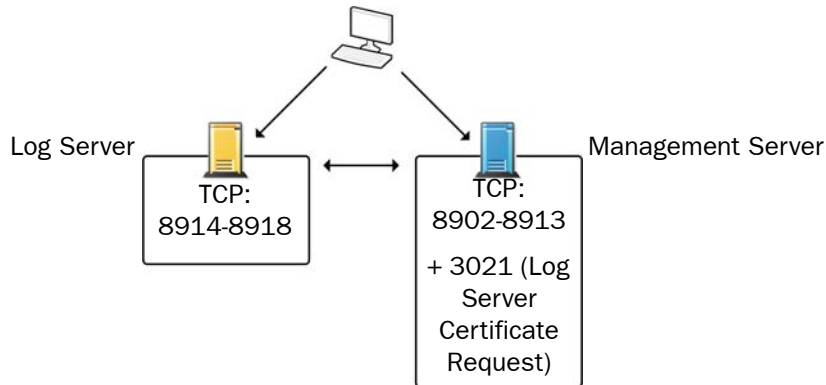
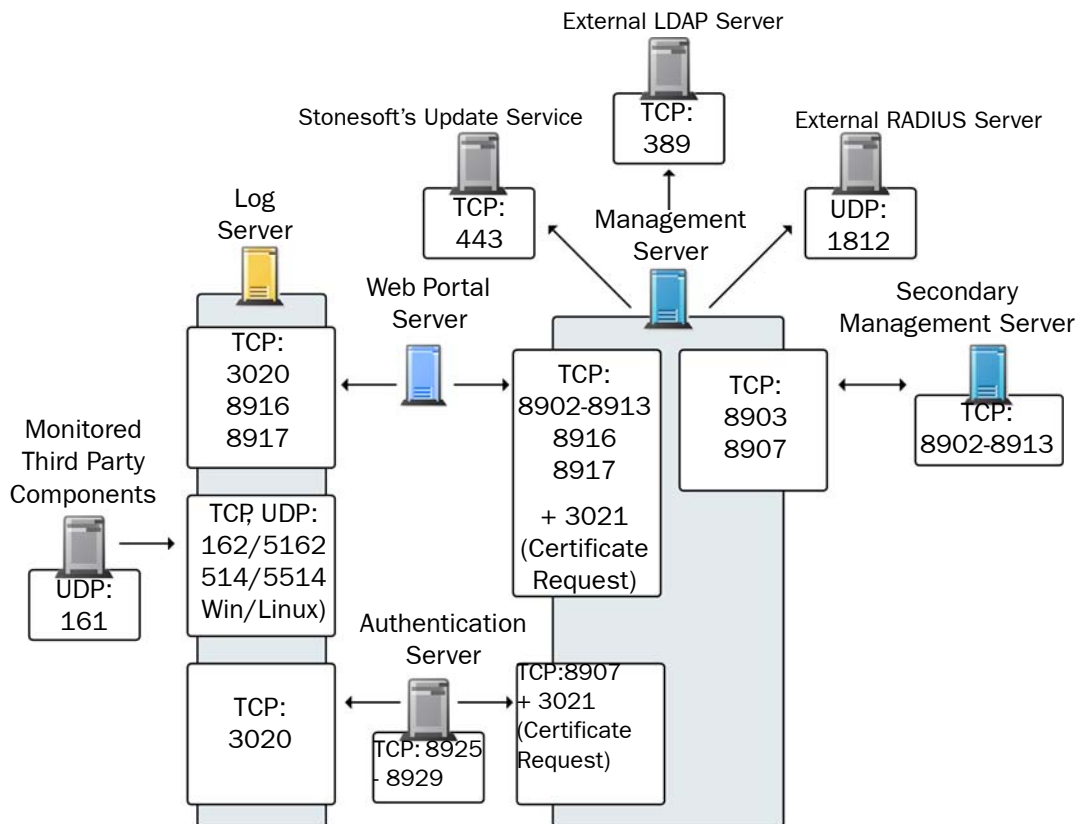


ILLUSTRATION B.2 Default Destination Ports for Optional SMC Components and Features



The table below lists all default ports SMC uses internally and with external components. Many of these ports can be changed. The name of corresponding default Service elements are also included for your reference. For information on communications between SMC components and the engines, see the separate listings.

TABLE B.1 Management Center Default Ports

Listening Host	Port/ Protocol	Contacting Hosts	Service Description	Service Element Name
Authentication Server	8925-8929/TCP	Management Server	StoneGate Management Server commands to Authentication Server.	SG Authentication Commands
Authentication Server node	8988-8989/TCP	Authentication Server node	Data synchronization between Authentication Server nodes.	SG Authentication Sync
DNS server	53/UDP, 53/TCP	Management Client, Management Server, Log Server	DNS queries.	DNS (UDP)
LDAP server	389/TCP	Management Server	External LDAP queries for display/editing in the Management Client.	LDAP (TCP)
Log Server	162/UDP, 5162/UDP	Monitored third party components	SNMPv1 trap reception from third party components. Port 162 is used if installed on Windows, port 5162 if installed on Linux.	SNMP (UDP)
Log Server	514/TCP, 514/UDP, 5514/TCP, 5514/UDP	Monitored third party components	Syslog reception from third party components. Port 514 is used if installed on Windows, port 5514 if installed on Linux.	Syslog (UDP) [Partial match]
Log Server	3020/TCP	Authentication Server, Log Server, Web Portal Server	Alert sending.	SG Log
Log Server	8914-8918/TCP	Management Client	Log browsing.	SG Data Browsing
Log Server	8916-8917/TCP	Web Portal Server	Log browsing.	SG Data Browsing (Web Portal Server)
Management Server	3021/TCP	Log Server, Web Portal Server	System communications certificate request/renewal.	SG Log Initial Contact
Management Server	8902-8913/TCP	Management Client, Log Server, Web Portal Server	Monitoring and control connections.	SG Control
Management Server	8907/TCP	Authentication Server	Status monitoring.	SG Control

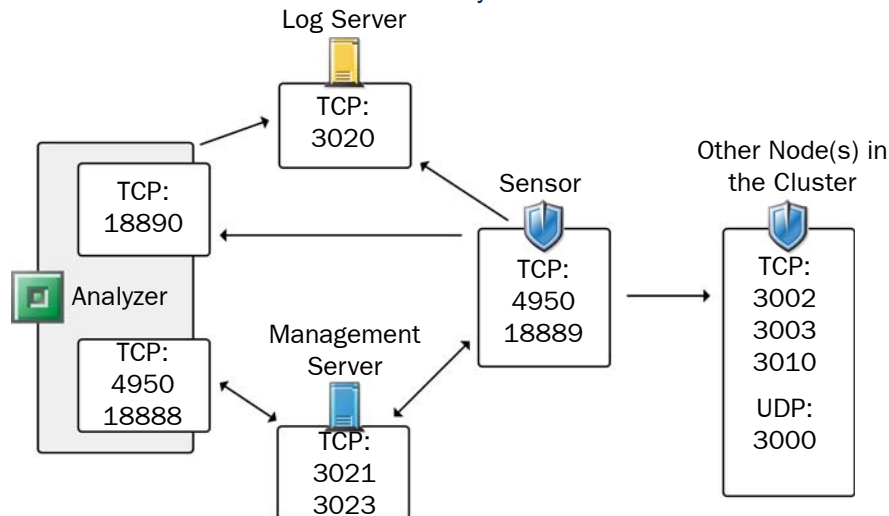
TABLE B.1 Management Center Default Ports (Continued)

Listening Host	Port/ Protocol	Contacting Hosts	Service Description	Service Element Name
Monitored Third Party Components	161/UDP	Log Server	SNMP status probing to external IP addresses.	SNMP (UDP)
Primary Management Server	8903, 8907/TCP	Secondary Management Servers	Database replication (pull) to the secondary Management Server.	SG Control
RADIUS server	1812/UDP	Management Server	RADIUS authentication requests for administrator logins. The default ports can be modified in the properties of the RADIUS Server element.	RADIUS (Authentication)
Secondary Management Servers	8902-8913/TCP	Primary Management Server	Database replication (push) to the secondary Management Server.	SG Control
Stonesoft servers	443/TCP	Management Server	Update packages, engine upgrades, and licenses from update.stonesoft.com and smc.stonesoft.com.	HTTPS
Syslog Server	514/UDP, 5514/UDP	Log Server	Log data export to syslog servers. The default ports can be modified in the LogServerConfiguration.txt file.	Syslog (UDP) [Partial match]

IPS Engine Ports

The illustration below presents an overview to the most important default ports used in communications between IPS engines and the SMC and between clustered sensor engines. See the table below for a complete list of default ports.

ILLUSTRATION B.3 Default Destination Ports for Basic IPS System Communications



The table below lists all default ports StoneGate IPS uses internally and with external components. Many of these ports can be changed. The name of corresponding default Service elements are also included for your reference.

TABLE B.2 IPS-Specific Ports

Listening Hosts	Port/Protocol	Contacting Hosts	Service Description	Service Element Name
Analyzer	514/UDP	Syslog server	Syslog messages forwarded to Analyzer.	Syslog (UDP)
Analyzer	4950/TCP	Management Server	Remote upgrade.	SG Remote-Upgrade
Analyzer	18889/TCP	Management Server	Management connection.	SG Commands (Analyzer)
Analyzer	18890/TCP	Sensor	Event data sent from the Sensors.	SG Event Transfer
BrightCloud Server	2316/TCP	Sensor	BrightCloud web filtering update service.	BrightCloud update
Log Server	3020/TCP	Analyzer, Sensor	Log and alert messages from Analyzers; recording file transfers from Sensors; and monitoring of blacklists, status, and statistics from Sensors.	SG Log
Management Server	3021/TCP	Sensor, analyzer	System communications certificate request/renewal (initial contact).	SG Initial Contact
Management Server	3023/TCP	Sensor, analyzer	Backup monitoring (status) connection.	SG Reverse Monitoring

TABLE B.2 IPS-Specific Ports (Continued)

Listening Hosts	Port/Protocol	Contacting Hosts	Service Description	Service Element Name
Sensor	3000-3001/ UDP 3002,3003, 3010/TCP	Sensor	Heartbeat between the cluster nodes.	SG State Sync (Multicast), SG State Sync (Unicast), SG Data Sync
Sensor	4950/TCP	Management Server	Remote upgrade.	SG Remote Upgrade
Sensor	18888/TCP	Management Server	Management connection.	SG Commands (Sensor)
Sensor, firewall	15000/TCP	Management Server, analyzer, sensor	Blacklist entries.	SG Blacklisting

APPENDIX C

PREDEFINED ALIASES

This appendix lists the predefined Aliases that exist in the system. The predefined Aliases are used in the Firewall's default system policies. Some of them may be useful when you create your own Firewall and IPS rules.

The following sections are included:

- ▶ [Pre-Defined User Aliases](#) (page 154)
- ▶ [System Aliases](#) (page 154)

Pre-Defined User Aliases

User Aliases are usually created by administrators, but there are also some pre-defined user aliases in the system. User Aliases are preceded with one \$ character. The table below lists all the editable user Aliases automatically created by StoneGate Management Center. These Aliases are used in the firewalls' default DHCP Relay Sub-Policy.

Table C.1 System-defined User Aliases

Pre-Defined User Alias	Description
\$ DHCP address pools	Addresses that can be allocated by DHCP server(s).
\$ DHCP address pools for IPsec VPN Clients	Address pools for assigning virtual IP addresses to VPN clients.
\$ DHCP servers	All DHCP servers defined for the firewall.
\$ DHCP servers for IPsec VPN Clients	The DHCP servers defined for assigning virtual IP addresses to VPN clients.

System Aliases

System Aliases are non-editable Aliases created by StoneGate. The System Aliases are preceded with two \$\$ characters. The table below lists the definitions of all the System Aliases. These Aliases are used in the firewall's default system policies.

Table C.2 System Aliases

System Alias	Description
\$\$ DHCP Enabled Interface Addresses	IP addresses (of CVIs on clusters) which have DHCP relay enabled.
\$\$ DHCP Enabled interface addresses for IPsec VPN clients	IP addresses (of NDIs on clusters) which have DHCP relay enabled for VPN Clients.
\$\$ DHCP Interface X.dns	IP address of the DHCP-assigned DNS server for interface number X.
\$\$ DHCP Interface X.gateways	IP address of the DHCP-assigned default router for interface number X.
\$\$ DHCP Interface X.ip	DHCP-assigned IP address for interface number X.
\$\$ DHCP Interface X.net	Network behind the dynamic IP interface number X.
\$\$ Interface ID X.ip	First IP address (CVI) of Physical Interface ID X.
\$\$ Interface ID X.net	Directly connected networks behind Physical Interface ID X.
\$\$ Local Cluster	All addresses of the cluster.
\$\$ Local Cluster (CVI addresses only)	All CVI addresses of the cluster.

Table C.2 System Aliases (Continued)

System Alias	Description
\$\$ Local Cluster (DHCP Interface Addresses)	All DHCP-assigned IP addresses of the engine.
\$\$ Local Cluster (NDI addresses only)	All NDI addresses of all nodes in the cluster.
\$\$ Local Cluster (NDI for heartbeat addresses only)	Heartbeat NDI addresses of all nodes in the cluster.
\$\$ Local Cluster (NDI for management addresses only)	Management NDI address(es) of all nodes in the cluster.
\$\$ Log Servers	IP addresses of all Log Servers.
\$\$ Management Servers	IP addresses of all Management Server.
\$\$ Valid DHCP Address Pools for IPsec VPN clients	Address pools defined for assigning virtual IP addresses to VPN clients.
\$\$ Valid DHCP Servers	All DHCP servers defined for the firewall.
\$\$ Valid DHCP Servers for IPsec VPN clients	The DHCP servers defined for assigning virtual IP addresses to VPN clients.

APPENDIX D

SITUATION CONTEXT PARAMETERS

This appendix describes the parameters you can define for Situation Contexts.



Note - The details related to the Contexts in your system may be different from what is described here, because the Contexts may have been updated through dynamic update packages after this guide was published. Read the release notes of each update package you import to see which elements are affected.

The following sections are included:

- ▶ [Port/Host Scan Detection Parameters](#) (page 158)
- ▶ [Correlation Context Parameters](#) (page 159)
- ▶ [Regular Expression Parameter](#) (page 162)
- ▶ [Other Context Parameters](#) (page 162)

Port/Host Scan Detection Parameters

The table below explains the parameters for the Scan Started Event Context.

Table D.1 Scan Detection Parameters

Parameter	Description
Port scan start period (seconds)	Port scan is reported if any of the thresholds is exceeded within this time limit.
Port scan idle timeout (seconds)	Port scan is assumed finished if the originator makes no scan attempts within this time limit.
Port scan status delay (seconds)	Defines how often an interim status of ongoing port scan is reported.
Maximum normal TCP connections	Defines how many TCP connections that proceed normally according to the protocol definitions are allowed before action is taken.
Maximum allowed open TCP ports	Number of SYN+ACK replies allowed during the tracking before action is taken.
Maximum unreplied TCP connections	Number of unreplied TCP connections during the tracking before action is taken.
Maximum allowed closed TCP ports	Number of RST replies allowed during the tracking before action is taken.
Maximum TCP segments with no SYN or ACK	Number of TCP segments with no SYN or ACK flag before action is taken.
Wait time for TCP connections	Seconds waited before considering TCP connection as successful port scan or unreplied connection attempt.
Maximum UDP packet destinations	Number of UDP destinations allowed per host during the tracking before action is taken.
Maximum bidirectional UDP transfers	Number of bidirectional UDP transfers allowed per host during the tracking before action is taken.
Maximum unidirectional UDP transfers	Allowed number of UDP destinations that have not replied or have replied with ICMP error during the tracking period before action is taken.
Maximum allowed closed UDP ports	Number of ICMP Port Unreachable replies allowed per host during the tracking before action is taken.
Maximum ICMP requests per host	Number of ICMP request destinations allowed per host during the tracking before action is taken.
Maximum unreplied ICMP request destinations	Number of ICMP request destinations that have not replied during the tracking allowed per host before action is taken.
Maximum ICMP Echo Request destinations	Number of ICMP Echo Request (ping) destinations allowed per host during the tracking before action is taken.
Maximum unreplied ICMP Echo Requests	Number of ICMP Echo Request (ping) destinations that have not replied during the tracking allowed per host before action is taken.

Table D.1 Scan Detection Parameters (Continued)

Parameter	Description
Maximum ICMP Timestamp Request destinations	Number of ICMP Timestamp Request destinations allowed per host during the tracking before action is taken.
Maximum unreplied ICMP Timestamp Requests	Number of ICMP Timestamp Request destinations that have not replied during the tracking allowed per host before action is taken.
Maximum ICMP Netmask Request destinations	Number of ICMP Netmask Request destinations allowed per host during the tracking before action is taken.
Maximum unreplied ICMP Netmask Requests	Number of ICMP Netmask Request destinations that have not replied during the tracking allowed per host before action is taken.

Correlation Context Parameters

Event Compress

Event Compress combines repeated similar events into the same log entry, reducing clutter in the Logs view.

Table D.2 Event Compress Parameters

Field	Option (if any)	Explanation
Correlated Situations		Situation(s) you want to compress.
Time Window		All the matches to the Situation(s) selected are combined to a common log entry when they are triggered within the defined time from each other.
Log Fields Enabled	Select	Events triggered by the selected Situations are considered the same when the values those entries have in the Log Fields you place in Lognames are identical.
	Ignore	Events triggered by the selected Situations are considered the same, except when the values those entries have in the Log Fields you place in Lognames are identical.
Lognames		The selected log fields are used by the matching option you selected in the previous step.
Location	Very Early	The execution order of the Compress operation in relation to other operations. Compress operations that share the same Location are executed in parallel; each compress operation receives the same events as the other compress operations in the same Location. “Very Early” and “Early” locations may effect the operation of other Correlations.
	Early	
	Late	
	Very Late	
Compress Filter		Filters in data for the compression.

Event Count

Event Count finds recurring patterns in traffic by counting the times certain Situations occur within the defined period, so that action can be taken if the threshold values you set are exceeded.

Table D.3 Event Compress Parameters

Field	Option (if any)	Explanation
Correlated Situations		Situation(s) you want to count.
Time Window		The period of time within which the matches to the Situation must occur the specified number of times.
Alarm Threshold		The number of times that the selected Situation(s) must occur for the Correlation Situation to match.
Log Fields Enabled	Select	Events triggered by the selected Situations are considered the same when the values those entries have in the Log Fields you place in Lognames are identical.
	Ignore	Events triggered by the selected Situations are considered the same, except when the values those entries have in the Log Fields you place in Lognames are identical.
Lognames		The selected log fields are used by the matching option you selected in the previous step.

Event Group

Event Group finds event patterns in traffic by following if all events in the defined set of Situations match at least once in any order within the defined time period.

Table D.4 Event Compress Parameters

Field	Option (if any)	Explanation
Member (column)	Event Match	Filter for grouping.
	Needed Number	How many occurrences of the Event selected for this Member are required for them to be included in the grouping.
	Binding	Log field used for the grouping.
Correlated Situations		Situation(s) you want to group.
Keep and Forward Events	Yes	Makes the Correlation Situation examine the events and trigger the desired response defined in the Inspection rule but does not actually group the matching events into one. All the individual events are still available for further inspection, event though they have already triggered a response.
	No	Makes the Correlation Situation group the matching events together, so that only the response defined for the inspection rule is triggered, and no further processing is done on the individual events.

Table D.4 Event Compress Parameters (Continued)

Field	Option (if any)	Explanation
Time Window Size		The period of time within which the Situation must occur for them to be grouped.
Continuous Responses	Yes	Makes the Analyzer respond as defined in the Inspection rule to each occurrence of the defined event within the selected Time Window.
	No	Makes the Analyzer respond only to the first occurrence of the defined event within the selected Time Window.

Event Match

Event Match allows filtering event data produced by specific Situations using Filter expressions.

Table D.5 Event Compress Parameters

Field	Explanation
Correlated Situations	Situation(s) you want the Correlation Situation to match.
Filter	Filter for finding a pattern in the event data.

Event Sequence

Event Sequence finds event patterns in traffic by following if all events in the defined set of Situations match in a specific order within the defined time period.

Table D.6 Event Compress Parameters

Field	Option (if any)	Explanation
Entry to/Exit from (columns)	Event Match	Filter for selecting data for the sequencing.
	Binding	Log field that the Correlation Situation traces to find a sequence.
Correlated Situations		Situation(s) from which you want to find sequences.
Keep and Forward Events	Yes	Makes the Correlation Situation examine the events and trigger the desired response defined in the Inspection rule but does not actually group the matching events into one. All the individual events are still available for further inspection, event though they have already triggered a response.
	No	Makes the Correlation Situation group the matching events together, so that only the response defined for the inspection rule is triggered, and no further processing is done on the individual events.
Time Window Size		The period of time within which the Situation must occur for them to be considered a sequence.

Regular Expression Parameter

See [Regular Expression Syntax](#) (page 163).

Other Context Parameters

See the properties dialog of the Context in question (the Contexts are shown as branches/sub-branches in the **Other Elements**→**Situations**→**By Context** tree in the IPS Configuration view).

APPENDIX E

REGULAR EXPRESSION SYNTAX

Regular expressions are used to define patterns in traffic for custom Situations, which can be used in Inspection rules in StoneGate Firewall and IPS.

The following sections are included:

- ▶ [Syntax for StoneGate Regular Expressions](#) (page 164)
- ▶ [Special Character Sequences](#) (page 166)
- ▶ [Pattern-Matching Modifiers](#) (page 167)
- ▶ [Bit Variable Extensions](#) (page 168)
- ▶ [Variable Expression Evaluation](#) (page 170)
- ▶ [System Variables](#) (page 174)
- ▶ [Independent Subexpressions](#) (page 175)
- ▶ [Parallel Matching Groups](#) (page 176)

Syntax for StoneGate Regular Expressions

StoneGate custom Situations are often defined as text strings using regular expression patterns for matching byte sequences in network traffic.

The expression matching starts always at the beginning of the traffic stream. For this reason, `.*` is usually necessary at the beginning of a regular expression to indicate that there can be an undefined number of bytes in the traffic stream preceding the expression.

The regular expression string consists of one or more *branches* that are separated by the `|` logical OR symbol as follows: `“branch-1|branch-2| . . .”`. A branch contains one or more regular expressions one after another. The Situation matches if any of the regular expression branches separated by `|` matches to the network traffic byte stream.



Note - Regular expressions are case sensitive. Space characters are included in the matching process unless the modifier (?S) or (?x) is used to ignore spaces.

The StoneGate regular expressions are described in the table below.

Table E.1 StoneGate Regular Expression Syntax

Sequence	Description	Example
<code><char></code>	Matches only the defined characters.	<code>'2', 'A', 'foo'</code> match exactly to the defined characters: <code>'2', 'A', and 'foo'</code> respectively.
<code>.</code> (dot)	matches any character, including the null character <code>\x00</code> and a missing character. Matches also other than printable characters, such as the linefeed.	<code>'.'</code> matches any single character or byte.
<code>\x<hex></code>	Matches the hexadecimal byte value ranging from <code>\x00</code> to <code>\xFF</code> .	<code>'\x4d'</code> matches hexadecimal value <code>'4d'</code> which represents the decimal value <code>77</code> and the ASCII character <code>'M'</code> .
<code>[<char>]</code>	Match any of the characters in the list.	<code>'[15aB]'</code> matches when any of the characters <code>'1', '5', 'a', or 'B'</code> in the matching location of the inspected string.
<code>[^<char>]</code>	Matches when none of the characters in the list is present.	<code>'[^aBc]'</code> matches if none of the characters <code>'a', 'B', or 'c'</code> is present in the matching location of the inspected string.
<code>[<char1>-<char2>]</code>	Matches all the characters ranging from <code><char1></code> to <code><char2></code> , these two characters included.	<code>'[a-f]'</code> matches any character within the range from <code>'a'</code> to <code>'f'</code> , with <code>'a'</code> and <code>'f'</code> included.

Table E.1 StoneGate Regular Expression Syntax (Continued)

Sequence	Description	Example
<code>[<class>:]</code>	Used in bracket expression to match any character of the defined class. The <i><class></i> can be: alnum [0-9A-Za-z], alpha [A-Za-z], ascii (ascii char), blank (space or tab), cntrl (control char), digit [0-9], graph (alnum or punct), lower [a-z], print (printable char), punct [.,"!?:;:], space (any space char), upper [A-Z], word (alnum + '_' char), xdigit [0-9A-Fa-f].	<code>'[[:digit:]]</code> matches any digit, e.g. 1, 5, or 7.
<code>\<char></code>	Used for escaping special metacharacters to be interpreted as normal characters, in this case as <i><char></i> . The metacharacters are: <code>\) ([^ - * + ? . #</code>	<code>'\[</code> matches the '[' character instead of interpreting it as the regular expression class metacharacter.
<code>#<text></code>	Anything starting with '#' up to the linefeed (<code>\x0a</code>) or the carriage return (<code>\x0d</code>) character is considered as a comment and not used in the matching process.	<code>'# my comment.'</code> is not used in the matching process.
<code>(<expr1> <expr2>)</code>	Matches if either the sub-expression <i><expr1></i> or <i><expr2></i> matches.	<code>'a(bc de)'</code> matches 'abc' and 'ade'.

It is also possible to indicate repeated, consecutive regular expressions by using quantifiers as described in the table below.

Table E.2 StoneGate Regular Expression Quantifiers

Quantifier	Description	Example
<code><expr>*</code>	Matches if there are zero or more consecutive <i><expr></i> strings.	<code>'a*'</code> matches '<empty>', 'a', 'aa' and so on.
<code><expr>+</code>	Matches if there are one or more consecutive <i><expr></i> strings.	<code>'a+'</code> matches 'a', 'aa', 'aaa' and so on, but not the empty string.
<code><expr>?</code>	Matches if there is zero or one <i><expr></i> string.	<code>'a?'</code> matches '<empty>' and 'a'.
<code><expr>{n,m}</code>	<code>{num}</code> matches exactly <i>num</i> times the expression. <code>{num,}</code> matches <i>num</i> or more times the expression. <code>{num,max}</code> matches at least <i>num</i> and no more than <i>max</i> times the expression.	<code>"a{5,}"</code> matches five or more consecutive 'a'. <code>"a{5,7}"</code> matches five, six, or seven consecutive 'a'.

The '*' and '+' wildcard characters in the middle of a regular expression pattern can easily result in an expression that has a very large number of different matching states. For this reason, they must be used with care. The computed matching pattern can grow exponentially, thus becoming too complex for efficient use on the Sensors.

Use the “`{num,max}`” quantifier where possible, instead of the ‘*’ and ‘+’ bounds. Variables can also be used to break down the pattern to smaller chunks as described in [Bit Variable Extensions](#) (page 168).

The illustration below provides an example regular expression.

Illustration E.1 Example regular expression

```
# This expression matches any of the following patterns in the traffic:
# '/bin/{ash|bash|csh|ksh|sh|tcsh}'

# First, match '/bin/sh' with zero or more characters in front of it:
./bin/sh|
# or match '/bin/' with zero or more characters in front of it,
# followed by 'ash', 'csh', or 'ksh':
./bin/[ack]sh|
# or match '/bin/' with zero or more characters in front of it,
# followed by 'bash' or 'tcsh':
./bin/(ba|tc)sh

# Alternatively, this expression with all the patterns can be integrated
# into one, for example: ./bin/(ba|tc|[ack])?sh
```

Special Character Sequences

The printable characters are defined simply by typing them in the regular expression. The hexadecimal values `\xHH` can also be used to match any byte value (e.g., ASCII character). In addition, there are some shorthands for common non-printable characters and character classes. The special character sequences are listed in the table below.

Table E.3 Special Character Sequences

Sequence	Description
<code>\a</code>	Bell (BEL) = <code>\x07</code>
<code>\t</code>	Horizontal tab (HT) = <code>\x09</code>
<code>\n</code>	Linefeed (LF) = <code>\x0A</code>
<code>\f</code>	Formfeed (FF) = <code>\x0C</code>
<code>\r</code>	Carriage return (CR) = <code>\x0D</code>
<code>\e</code>	Escape (ESC) = <code>\x1B</code>
<code>\OOO</code>	Octal code <i>OOO</i> of the character.
<code>\xHH</code>	Hexadecimal code <i>HH</i> of the character.
<code>\c<char></code>	Control character that corresponds to <code>Ctrl+<char></code>
<code>\w</code>	"word" class character = <code>[A-Za-z0-9_]</code>
<code>\W</code>	Non-"word" class character = <code>[^A-Za-z0-9_]</code>
<code>\s</code>	Whitespace character = <code>[\t\r\n\f]</code>

Table E.3 Special Character Sequences (Continued)

Sequence	Description
<code>\s</code>	Non-whitespace character = [^ \t\r\n\f]
<code>\d</code>	Digit character = [0-9]
<code>\D</code>	Non-digit character = [^0-9]
<code>\b</code>	Backspace (BS) = \x08 Note: allowed only in bracket expressions.
<code>\Q</code> <code><expr></code> <code>\E</code>	Quotes all metacharacters between the <code>\Q</code> and <code>\E</code> . Backslashes are considered as normal characters. For example, " <code>\QC:\file.exe\E</code> " matches the " <code>C:\file.exe</code> " string, not the " <code>C:\x0Cile.exe</code> " string where <code>\x0C</code> is the formfeed " <code>\f</code> ".

Pattern-Matching Modifiers

The StoneGate regular expression syntax has Perl-like extensions. The pattern-matching modifiers are extensions that can be used to control the matching process in more detail. The modifiers are enabled with `(?<modifiers>)` and disabled with a minus `(?-<modifiers>)`, where `<modifiers>` is a list of one or more modifiers.

The modifiers `(?C)`, `(?L)`, and `(?S)` are enabled by default. The pattern-matching modifiers are listed in the table below.

Table E.4 Pattern-Matching Modifiers

Sequence	Description
<code>(?i)</code>	<p>"Case insensitive mode"</p> <p>When enabled, case insensitive matching is used for the uppercase and lowercase letters. Thus, a letter matches regardless of its capitalization.</p> <p>When disabled, the letters are matched case-sensitively so that capitalization is taken into account in the matching process.</p>
<code>(?S)</code>	<p>"Single line mode"</p> <p>When enabled, the dot character <code>.</code> matches also the null character <code>\x00</code> and a missing character in addition to matching any character (including linefeed and other non-printable characters).</p> <p>When disabled, the linefeed or a missing character are not matched.</p> <p>This modifier is enabled by default. Use <code>(?-S)</code> to disable it.</p>
<code>(?x)</code>	<p>"Extended readability mode"</p> <p>When enabled, equals to enabling <code>(?C)</code>, <code>(?L)</code>, and <code>(?S)</code>. Comments, linefeeds and spaces are not used in the matching process, allowing to use them for readability of the expression.</p> <p>When disabled, equals to disabling <code>(?C)</code>, <code>(?L)</code>, and <code>(?S)</code>. Comments, linefeeds and spaces are used in the matching process.</p>

Table E.4 Pattern-Matching Modifiers (Continued)

Sequence	Description
(? C)	<p>“Allow comments mode”</p> <p>When enabled, anything after the hash character ‘#’ is considered as a comment and not included in the matching process.</p> <p>When disabled, the hash character ‘#’ and anything following are used in the matching process.</p> <p>This modifier is enabled by default. Use (?-C) to disable it.</p>
(? L)	<p>“Ignore linefeeds mode”</p> <p>When enabled, the linefeed and carriage return characters are not included in the matching process unless specifically defined (\x0A or \n for linefeed and \x0D or \r for carriage return).</p> <p>When disabled, the linefeeds and carriage returns are used in the matching process.</p> <p>This modifier is enabled by default. Use (?-L) to disable it.</p>
(? S)	<p>“Ignore spaces mode”</p> <p>When enabled, the space and horizontal tab characters are not used in the matching process unless specifically defined (\x20 for space and \x09 or \t for horizontal tab).</p> <p>When disabled, the space and horizontal tab characters are used in the matching process.</p>
(? <i>modifiers</i>): <i>sub-expr</i>)	Applies the <i>modifiers</i> modifiers only to the subexpression <i>sub-expr</i> . These modifiers are not used in other parts of the regular expression.

Bit Variable Extensions

Variables can be used to define regular expression patterns that are related to each other. These relations can be expressed with the variables so that the regular expression matches only when all the related patterns match. Complex matching with multiple Situations is also possible, as the variables and the variable values are shared with all the Situations in a Situation Group.

A variable extension can use the following expressions:

- A *value setting expression* defines the values for one or more variables when the corresponding top-level branch matches.
For example, (?{**var_a=1**}) sets the value 1 for the variable `var_a`.
- A *conditional expression* inspects the values defined for one or more variables so that the corresponding top-level branch matches, and the optional variable setting expressions are processed only if the conditional expression is true.
For example, (?{**var_b==1**}) matches when the variable `var_b` is equal to 1.
- When using both variable expression types for the same top-level branch, the implication operator ‘->’ must be used.
For example, (?{**var_a==1->var_a=0**}) matches when the variable `var_a` is equal to 1, and finally sets the value for this variable to be 0.

Each variable is unique within the Situation or Situation Group where the variable is used. The name of a Situation variable can be anything consisting of alphanumeric and underscore characters [A-Za-z_0-9]. The variable name must not begin with a digit. The variable has a boolean value that can be either 0 or 1. The variable values persist through each individual traffic stream.



Note - In variable expressions a single equal sign '=' sets a value for a variable, whereas two consecutive equal signs '==' evaluate the value of a variable.

Variables are defined with the expressions listed in the table below.

Table E.5 Variable Extensions

Sequence	Description
<code>(?{<var>=<value>})</code>	The expression matches and the <code><var></code> variable's value is set to <code><value></code> (0 or 1). Multiple value setting expressions can be defined by separating them with a comma <code>,</code> .
<code>(?{<var>=<value>,ignore})</code>	Sets the <code><var></code> variable's value to <code><value></code> (0 or 1). The <code>ignore</code> keyword is used to indicate a partial match that does not trigger response alone but requires another matching branch.
<code>(?{<var>==<value>})</code>	The expression matches only when the <code><var></code> variable's value is <code><value></code> . Multiple conditional expressions can be defined by separating them with <code>&&</code> .
<code>(?{<var1>==<value1>-><var2>=<value2>})</code>	The expression matches only when the <code><var1></code> variable's value is <code><value1></code> . When the condition is true, the <code><var2></code> variable's value is set to <code><value2></code> .

`(?{...})` can be used in the two top-level branches that are separated by the logical OR symbol `|`. A variable extension is processed only when its top-level branch matches.

Illustration E.2 Expression with Variables

```
# Expression matches only when 'POST /attack.asp?' string is followed
# by 'Content-Type: application/x-www-form-urlencoded' string
# with any number of bytes in between.

(?i)#case-insensitive mode

.*POST /attack.asp\?(?{match=1,ignore})|#does not trigger response alone
.*Content-Type: application/x-www-form-urlencoded(?{match==1->match=0})
```

The expression in the illustration below detects two different strings in the same connection. The variable is used so that the response is triggered only when the first branch matches, followed by the second branch match. Neither of the branches trigger the response alone.



Note - A '*' or '?' wildcard in a middle of an expression can result in a computed matching pattern that is too complex for efficient use on the Sensors. Therefore, it is recommended to divide the pattern into two branches as in the illustration above.

Variable Expression Evaluation

Variable expression evaluation is an extension to regular expression syntax that provides the ability to parse values from the input stream, perform arithmetic operations, detect large blocks of data, and use variable larger than one bit. This allows you to create more precise and reliable Situations in cases that are difficult with the traditional regular expression syntax.

Table E.6 Variable Expression Syntax

Sequence	Description
(?[<expression>])	<expression> is one or more comma-separated expressions

Variables can be 1, 8, 16, 32 or 64 bits long. By default, a variable is one bit (either 0 or 1). The default variable size in bits can be changed with a postfix that contains a "@" sign and the number of bits.

Example `test@32` means that the variable `test` is 32 bits long.

If the variable name is prefixed with a dollar sign (\$), the variable is matched against the current connection instead of the current stream. This matching in both client to server and server to client traffic.

Example `$command_seen@32` checks that a certain command has been issued by the client and the server has accepted the command without errors.

Each expression has a value after evaluation. The type of the value can be a 32-bit or 64-bit unsigned integer, or a void. The results of Expressions can be used to perform basic integer arithmetic, variable assignment, and comparisons. The order of operations is similar to that of the C programming language, for example $A + B * C$ is $A + (B * C)$, not $(A + B) * C$. The '-' is lowest in precedence. Statements inside parentheses () are always evaluated first, so the order of operations can be overridden with parentheses,

Table E.7 Operations on Expression Results

Sequence	Description
false	Always evaluates to a false.
true	Always evaluates to a true.
<number>	A literal number in decimal, octal and hexadecimal format, for example "32" or "0x20".
<var> = <expr>	Sets a value returned by expression <expr> to a variable <var>. See variable syntax below.
<var> += <expr>	Adds the value of variable <var> with the value returned by expression <expr> and sets the result to variable <var>.
<var> -= <expr>	Subtracts the value from variable <var> by the value returned by expression <expr> and sets the result to variable <var>.
<var> *= <expr>	Multiplies the value of <var> by the value returned by expression <expr> and sets the result to variable <var>.

Table E.7 Operations on Expression Results (Continued)

Sequence	Description
<var> /= <expr>	Divides the value of <var> with the value returned by expression <expr> and sets the result to variable <var>.
<var> %= <expr>	Divides the value of <var> with the value returned by expression <expr> and sets the modulo of result to variable <var>.
<var> <<= <expr>	Shifts the value of <var> to left by number of steps returned by expression <expr> and sets the result to variable <var>.
<var> >>= <expr>	Shifts the value of <var> to right by number of steps returned by expression <expr> and sets the result to variable <var>.
<var> &= <expr>	Performs bitwise AND with the value of variable <var> and the value returned by expression <expr> and sets the result to variable <var>.
<var> = <expr>	Performs bitwise OR with the value of variable <var> and the value returned by expression <expr> and sets the result to variable <var>.
<var> ^= <expr>	Performs bitwise XOR with the value of variable <var> and the value returned by expression <expr> and sets the result to variable <var>.
<expr_a> -> <expr_b>	Expression <expr_b> is evaluated only if <expr_a> is true.
<expr_a> ? <expr_b> : <expr_c>	Expression <expr_b> is evaluated only if <expr_b> is true and expression <expr_c> is evaluated if <expr_a> is false.
<expr_a> == <expr_b>	Test if expressions <expr_a> and <expr_b> return an equal value.
<expr_a> != <expr_b>	Test if expressions <expr_a> and <expr_b> do not return an equal value.
<expr_a> < <expr_b>	Test if expression <expr_b> returns higher value than expression <expr_a>.
<expr_a> <= <expr_b>	Test if expression <expr_b> returns higher or equal value than expression <expr_a>.
<expr_a> > <expr_b>	Test if expression <expr_a> returns higher value than expression <expr_b>.
<expr_a> >= <expr_b>	Test if expression <expr_a> returns higher or equal value than expression <expr_b>.
<expr_a> & <expr_b>	Performs bitwise AND with expressions <expr_a> and <expr_b> and returns the result.
<expr_a> <expr_b>	Performs bitwise OR with expressions <expr_a> and <expr_b> and returns the result.
<expr_a> ^ <expr_b>	Performs bitwise XOR with expressions <expr_a> and <expr_b> and returns the result.
<expr_a> && <expr_b>	Performs AND with expressions <expr_a> and <expr_b> and returns the result.

Table E.7 Operations on Expression Results (Continued)

Sequence	Description
<code><expr_a> <expr_b></code>	Performs OR with if expressions <code><expr_a></code> and <code><expr_b></code> and returns the result.
<code><var>++, ++<var></code>	Increase value of variable <code><var></code> by one.
<code><var>--, --<var></code>	Decrease value of variable <code><var></code> by one.
<code>-<expr></code>	Negate the result of the expression <code><expr></code> .
<code>~<expr></code>	Bitwise invert the result of the expression <code><expr></code> .
<code>!<expr></code>	Perform NOT operation with the expression <code><expr></code> .

Stream Operations

The binary data from the input stream can be read into variables with the following expressions:

Table E.8 Binary Data Variable Expressions

Sequence	Description
<code>parse_be@<size></code>	Parse big endian value. <code><size></code> is the size of the value to be read in bits, and it can be one of the following: 8, 16, 24, 32, 40, 48, 56 or 64.
<code>parse_le@<size></code>	Parse little endian value. <code><size></code> is the size of the value to be read in bits, and it can be one of the following: 8, 16, 24, 32, 40, 48, 56 or 64.

ASCII values can be read from the input stream with the following expressions:

Table E.9 ASCII Data Variable Expressions

Sequence	Description
<code>parse_dec(<length>)</code>	Parse ASCII decimal value. <code><length></code> is the maximum number of the characters to parse. The actual number of parsed digits in the variable <code>\$parse_length@32</code> . If no characters could be parsed, then the variable is set to zero.
<code>parse_dec(<is available>)</code>	The actual number of parsed digits is available in the variable.
<code>parse_hex(<length>)</code>	Parse ASCII hexadecimal value. <code><length></code> is the maximum number of the characters to parse. The actual number of parsed digits in the variable <code>\$parse_length@32</code> . If no characters could be parsed, then the variable is set to zero.
<code>parse_hex(<is available>)</code>	The actual number of parsed digits is available in the variable.
<code>parse_int(<length>)</code>	Parse ASCII value; parses hexadecimal if the string starts with "0x", octal if the string starts with zero ("0") and decimal otherwise. <code><length></code> is the maximum number of the characters to parse. The actual number of parsed digits in the variable <code>\$parse_length@32</code> . If no characters could be parsed, then the variable is set to zero.

Table E.9 ASCII Data Variable Expressions (Continued)

Sequence	Description
<code>parse_int(<is available>)</code>	The actual number of parsed digits is available in the variable.
<code>parse_oct(<length>)</code>	Parse ASCII octal value. <length> is the maximum number of the characters to parse. The actual number of parsed digits in the variable <code>\$parse_length@32</code> . If no characters could be parsed, then the variable is set to zero.
<code>parse_oct(<is available>)</code>	The actual number of parsed digits is available in the variable.

Miscellaneous operations with the input stream:

Table E.10 Miscellaneous Input Stream Operations

Sequence	Description
<code>CRC(<length>)</code>	Calculates a 32-bit CRC value starting from the current byte up to number of bytes specified by <length> parameter. This function is suitable to detect large binary blocks from the input stream.
<code>skip(<length>)</code>	Skip <length> number of bytes.
<code>regex(<regex>)</code>	Launch independent subexpression. See section "Independent Subexpression" for more information.

Other Expressions

Table E.11 Other Expressions

Sequence	Description
<code>sid()</code>	Generate a situation. This expression is used to generate a situation indicating a match.
<code>sid(<id>)</code>	Generate a specific situation specified by <id>.
<code>cancel</code>	Stop matching in the current level.
<code>cancel_fp_ctx</code>	Stop matching in the current fingerprinting context.

The illustration below provides an example of a regular expression that launches a variable expression.

Illustration E.3 Expression with Variable Expression

```
# Launches a variable expression when byte 20h (space character) is seen in
# the input stream. The evaluation first assigns variable aa@16 a value of
# 8, checks if the value of variable bb@32 is 47, and if so, generates a
# situation.
.*\x20(?:aa@16=8, bb@32==47 -> sid())
```

When `(?[])` is used (at the end of a regular expression branch or elsewhere), the situation is not reported automatically. `sid()` must be used explicitly. This is different from `(?{})`, where the situation is automatically reported if `ignore` is not used.

System Variables

The syntax of the system variables is the same as for other variables (see the table [Variable Extensions](#) (page 169), except that the variable's value is not user-changeable.

Table E.12 System Variables

Sequence	Description
<code>\$major</code>	The major version number of the StoneGate engine.
<code>\$minor</code>	The minor version number of the StoneGate engine.
<code>\$patch</code>	The patch level number of the StoneGate engine.
<code>\$build</code>	The build number of the StoneGate engine.
<code>\$fpver</code>	The version number of the current fingerprint matching engine.
<code>\$dport</code>	The current destination port of the connection. This value can be used to limit matching to traffic that is destined to a specific port.
<code>\$offset</code>	The byte that is under inspection when counted from the beginning of the traffic stream.
<code>\$parse_length@32</code>	Number of digits parsed by last <code>parse_dec()</code> , <code>parse_hex()</code> , <code>parse_oct()</code> or <code>parse_in()</code> expression. See Stream Operations below.

The illustration below provides example of an offset expression that matches when `'\xff\x53\x4d\x42\x25'` occurs in the stream, so that `\x25` is the 25th byte in the stream.

Illustration E.4 Expression with Variables

```
# Expression matches when '\xff\x53\x4d\x42\x25' occurs in the stream with
\x25 as the # 25th byte in the stream. The 25th byte in the stream has an
offset value
# of 24.

.*\xff\x53\x4d\x42\x25(?{$offset==24})
```

Independent Subexpressions

Independent subexpressions allow another regular expression to be launched independently from the main regular expression. The syntax for the independent subexpression is as follows:

Table E.13 Independent Subexpression Syntax

Sequence	Description
<code>(?><regular_expression>)</code>	<code><regular_expression></code> is a normal StoneGate regular expression launched independently from the main regular expression.
<code>(?>(?{<expression>}<regular_expression>)</code>	<code><expression></code> is a comparison expression that is evaluated before the independent subexpression <code><regular_expression></code> is launched. <code><regular expression></code> is launched only if <code><expression></code> evaluates to true.

Within `(?[...])` independent subexpressions can be launched by `regex(<regular_expression>)`.

The illustration below provides an example of a regular expression that launches independent subexpression.

Illustration E.5 Expression with Independent Subexpressions

```
# Main expression matches when a GET string has been seen in the input stream
# and launches an independent subexpression. The independent subexpression
# detects whether the parameter for the GET request is longer than 400
# characters.

*GET(?>\s*[^\\s]{400})
```

The illustration below provides an example of an independent subexpression that includes a precondition parameter check. The independent subexpression is launched only if the precondition expression evaluates to true.

Illustration E.6 Independent Subexpression with Precondition Parameter Check

```
# Launches independent subexpression for a Content-length: HTTP header only
# if it is seen in a POST request.

(?x)
(?i)
.*POST(?{post_seen=1,ignore})|
.*\nContent-length:(?>(?{post_seen==1})[^\n]{1024})
```

Parallel Matching Groups

StoneGate allows you to set different regular expressions to be matched in parallel groups within one Situation Context. Normally, manual situation group definitions are not needed and StoneGate automatically compiles all your custom Situations in the same group (group 0).

Manual group definitions is needed if the IPS policy upload fails due to fingerprint/DFA compilation problems that may occur with complex regular expressions.

To use grouping, add a new preprocessing tag to the beginning of the regular expression:

Table E.14 Preprocessing Tag for Setting a Group for Matching

Syntax	Description
<code>##!GROUP(X)</code> Comment <code>##!#</code>	'X' is the group number from 0 to 7. The comment is optional. If you do not specify the group with this tag, the Situation is processed in group zero.

Illustration E.7 Setting a parallel Matching Group

```
##!GROUP(1)
This heavy regular expression is matched in parallel matching group 1.
##!#

#Insert regular expression below
```

APPENDIX F

SNMP TRAPS AND MIBs

StoneGate Firewall/VPN and IPS engines can send SNMP traps on system events. The traps are configured using SNMP Agent elements. Additionally, the Tester entries can be configured to send SNMP traps. The SNMP traps are listed in the table below.

Table F.1 SNMP Traps for StoneGate Firewall/VPN and IPS

Trap Name	Objects Included	Description
fwPolicyInstall	fwSecurityPolicy	<i>(Firewall only)</i> Policy was installed on the firewall engine.
ipsPolicyInstall	ipsSecurityPolicy	<i>(IPS only)</i> Policy was installed on the IPS engine.
nodeBoot	-	Node bootup complete.
nodeHwmon	nodeHwmonEvent	Hardware monitoring system has detected problems.
nodeOffline	nodeOperState	Node changed to offline or standby state.
nodeOnline	nodeOperState	Node changed to online state.
nodeShutdown	-	Node is shutting down.
nodeTestFailure	nodeTestIdentity	Test subsystem reported a test failure on the node.
nodeFailedUserLogin	nodeLastLogin	<i>(Firewall only)</i> Login failed on the firewall engine's console or through SSH.
nodeUserLogin	nodeLastLogin	Login initiated on the engine's console or through SSH.
nodeUserLogout	nodeLastLogin	<i>(Firewall only)</i> Logout on the firewall engine's console or through SSH.

The STONESOFT-SMI-MIB defines the top-level enterprise registrations for the Stonesoft's products in the .iso.org.dod.internet.private.enterprises.stonesoft branch (OID .1.3.6.1.4.1.1369). The StoneGate specific MIBs are:

- STONESOFT-FIREWALL-MIB (table [STONESOFT-FIREWALL-MIB Objects](#) (page 178))
- STONESOFT-IPS-MIB (table [STONESOFT-IPS-MIB Objects](#) (page 181))
- STONESOFT-NETNODE-MIB (table [STONESOFT-NETNODE-MIB Objects](#) (page 181)).

The StoneGate Firewall/VPN MIB files can be downloaded from the Stonesoft website at <http://www.stonesoft.com/>.

The StoneGate Firewall/VPN also supports objects in the following standard MIBs:

- IF-MIB (RFC 2863 and RFC 2233) (table [IF-MIB Supported Objects](#) (page 181))
- IP-MIB (RFC 2011) (table [IP-MIB Supported Objects](#) (page 185))
- SNMP-USER-BASED-SM-MIB (RFC 3414) (table [SNMP-USER-BASED-SM-MIB Objects](#) (page 189)).
- SNMPv2 MIB (RFC 3418) (table [SNMPv2-MIB Supported Objects](#) (page 190))

Table F.2 STONESOFT-FIREWALL-MIB Objects

Object Name	Object Description in MIB
fwPolicyTime	The time when the security policy was installed to the firewall
fwSecurityPolicy	Name of the current security policy on the firewall
fwSoftwareVersion	Version string of the firewall software
fwConnNumber	Number of current connections
fwAccepted	Number of accepted packets
fwDropped	Number of dropped packets
fwLogged	Number of logged packets
fwAccounted	Number of accounted packets
fwRejected	Number of rejected packets
fwIfTable	This table contains an entry for each interface in system
fwIfStatsEntry	Row for a interface
fwIfStatsIndex	A unique value, greater than zero, for each interface or interface sub-layer in the managed system
fwIfName	Name of interface
fwIfAcceptedPkts	Number of accepted packets by firewall rules
fwIfDroppedPkts	Number of dropped packets by firewall rules
fwIfLoggedPkts	Number of logged packets by firewall rules
fwIfRejectedPkts	Number of rejected packets by firewall rules
fwIfAccountedPkts	Number of accounted packets by firewall rules

Table F.2 STONESOFT-FIREWALL-MIB Objects (Continued)

Object Name	Object Description in MIB
fwIfAcceptedBytes	Number of accepted bytes by firewall rules
fwIfDroppedBytes	Number of dropped bytes by firewall rules
fwIfLoggedBytes	Number of logged bytes by firewall rules
fwIfRejectedBytes	Number of rejected bytes by firewall rules
fwIfAccountedBytes	Number of accounted bytes by firewall rules
fwCpuTable	This table contains an entry for each CPU in a system and total usage of all CPUs
fwCpuStats	Row with information about CPU usage
fwCpuStatsId	A unique value, greater than zero, for each CPU in the managed system. First element with Id '0' is designed for total values
fwCpuName	Name of data current line concern
fwCpuTotal	The total CPU load percentage
fwCpuUser	The percentage of time the CPU has spent running users' processes that are not niced
fwCpuSystem	The percentage of time the CPU has spent running the kernel and its processes
fwCpuNice	The percentage of time the CPU has spent running user's processes that have been niced
fwCpuIdle	The percentage of time the CPU was idle
fwCpuIoWait	The percentage of time the CPU has been waiting for I/O to complete
fwCpuHwIrq	The percentage of time the CPU has been servicing hardware interrupts
fwCpuSoftIrq	The percentage of time the CPU has been servicing software interrupts
fwSwapBytesTotal	Total swap space
fwSwapBytesUsed	Used space of swap
fwSwapBytesUnused	Amount of unused space of swap
fwMemBytesTotal	Number of available bytes of physical memory
fwMemBytesUsed	Amount of memory being in use
fwMemBytesUnused	Amount of unused bytes of physical memory
fwMemBytesBuffers	Amount of memory used as buffers
fwMemBytesCached	Amount of memory used as cache
fwDiskSpaceUsageTable	Table contains an entry for each partition mounted in a system
fwDiskStats	Row of information concerning one partition
fwPartitionIndex	A unique value, greater than zero, for each partition

Table F.2 STONESOFT-FIREWALL-MIB Objects (Continued)

Object Name	Object Description in MIB
fwPartitionDevName	A unique name of a device
fwMountPointName	Name of a mount point
fwPartitionSize	Total size of the partition
fwPartitionUsed	Amount of used space of the partition (in kilobytes)
fwPartitionAvail	Information about amount of free space on partition (in kilobytes)
adslModulation	Modulation protocol
adslChannel	Channel type
adslConnStatus	The status of the DSL link or communication status with DSL modem in case of communication error
adslConnUptime	Uptime of current ADSL connection
adslLineStatus	Current status of DSL line
adslInOctets	Number of bytes received by ADSL interface
adslOutOctets	Number of bytes transmitted by ADSL interface
adslSynchroSpeedUp	The actual rate at which data is flowing upstream
adslSynchroSpeedDown	The actual rate at which data is flowing downstream
adslAttenuationUp	An estimate of the average loop attenuation upstream
adslAttenuationDown	An estimate of the average loop attenuation downstream
adslNoiseMarginUp	This is a signal-to-noise ratio (SNR) margin for traffic going upstream
adslNoiseMarginDown	This is a signal-to-noise ratio (SNR) margin for traffic going downstream
adslHecErrorsUp	The total number of header error checksum errors upstream
adslHecErrorsDown	The total number of header error checksum errors downstream
adslOcdErrorsUp	The number of out-of-cell delineation errors upstream
adslOcdErrorsDown	The number of out-of-cell delineation errors downstream
adslLcdErrorsUp	The total of lost-cell-delineation errors upstream
adslLcdErrorsDown	The total of lost-cell-delineation errors downstream
adslBitErrorsUp	The number of bit errors upstream
adslBitErrorsDown	The number of bit errors downstream

Table F.3 STONESOFT-IPS-MIB Objects

Object Name	Object Description in MIB
ipsPolicyTime	The time when the security policy was installed to the IPS engine
ipsSecurityPolicy	Name of the current security policy on the IPS engine
ipsSoftwareVersion	Version string of the IPS software

Table F.4 STONESOFT-NETNODE-MIB Objects

Object Name	Object Description in MIB
nodeClusterId	The identification number of the cluster this node belongs to
nodeCPULoad	The CPU load percentage on the node
nodeHwmonEvent	Reason for the hardware monitoring event
nodeLastLogin	The most recent login event on the node
nodeLastLoginTime	Timestamp of the most recent login event on the node
nodeMemberId	Node's member identification within the cluster
nodeOperState	The operative (clustering) state of the node
nodeTestIdentity	Identification string of a nodeTest
nodeTestResult	The most recent result of the nodeTest
nodeTestResultTime	The timestamp of the most recent result of the nodeTest

Table F.5 IF-MIB Supported Objects

Object Name	Object Description in MIB
ifAdminStatus	The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up(1) or testing(3) states (or remains in the down(2) state).

Table F.5 IF-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ifAlias	<p>This object is an 'alias' name for the interface as specified by a network manager, and provides a non-volatile 'handle' for the interface. On the first instantiation of an interface, the value of ifAlias associated with that interface is the zero-length string. As and when a value is written into an instance of ifAlias through a network management set operation, then the agent must retain the supplied value in the ifAlias instance associated with the same interface for as long as that interface remains instantiated, including across all re- initializations/reboots of the network management system, including those which result in a change of the interface's ifIndex value. An example of the value which a network manager might store in this object for a WAN interface is the (Telco's) circuit number/identifier of the interface. Some agents may support write-access only for interfaces having particular values of ifType. An agent which supports write access to this object is required to keep the value in non-volatile storage, but it may limit the length of new values depending on how much storage is already occupied by the current values for other interfaces.</p>
ifDescr	<p>A textual string containing information about the interface. This string includes the name of the manufacturer, the product name and the version of the interface hardware/software.</p>
ifHCInMulticastPkts	<p>The 64-bit wide number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>The 32-bit ifInMulticastPkts reports the low 32-bits of this counter's value.</p>
ifHCInOctets	<p>The 64-bit wide total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>The 32-bit ifInOctets reports the low 32-bits of this counter's value.</p>
ifHCInUcastPkts	<p>The 64-bit wide number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. This object is a 64-bit version of ifInUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>The 32-bit ifInUcastPkts reports the low 32-bits of this counter's value.</p>
ifHCOctets	<p>The 64-bit wide total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>The 32-bit ifOutOctets reports the low 32-bits of this counter's value.</p>

Table F.5 IF-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ifHCOutUcastPkts	<p>The 64-bit wide total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>The 32-bit ifOutUcastPkts reports the low 32-bits of this counter's value.</p>
ifHighSpeed	<p>An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object contains the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object must be zero.</p>
ifIndex	<p>A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re- initialization.</p>
ifInDiscards	<p>The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>
ifInErrors	<p>For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>
ifInMulticastPkts	<p>The 32-bit wide number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>This object reports the low 32-bits of the 64-bit ifHCInMulticastPkts counter's value.</p>
ifInOctets	<p>The 32-bit wide total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>This object reports the low 32-bits of the 64-bit ifHCInOctets counter's value.</p>

Table F.5 IF-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ifInUcastPkts	<p>The 32-bit wide number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p> <p>This object reports the low 32-bits of the 64-bit ifHCInUcastPkts counter's value.</p>
ifLastChange	<p>The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.</p>
ifLinkUpDownTrapEnable	<p>Indicates whether linkUp/linkDown traps are generated for this interface. By default, this object must have the value enabled(1) for interfaces which do not operate on 'top' of any other interface (as defined in the ifStackTable), and disabled(2) otherwise.</p>
ifMtu	<p>The size of the largest packet which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.</p>
ifName	<p>The textual name of the interface. The value of this object must be the name of the interface as assigned by the local device and must be suitable for use in commands entered at the device's `console`. This might be a text name, such as `le0` or a simple port number, such as `1`, depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. Note that for an agent which responds to SNMP queries concerning an interface on some other (proxied) device, then the value of ifName for such an interface is the proxied device's local name for it. If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string.</p>
ifNumber	<p>The number of network interfaces (regardless of their current state) present on this system.</p>
ifOperStatus	<p>The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) then ifOperStatus is down(2). If ifAdminStatus is changed to up(1) then ifOperStatus changes to up(1) if the interface is ready to transmit and receive network traffic; it changes to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it remains in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it remains in the notPresent(6) state if the interface has missing (typically, hardware) components.</p>
ifOutDiscards	<p>The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>

Table F.5 IF-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ifOutErrors	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
ifOutOctets	The 32-bit wide total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. This object reports the low 32-bits of the 64-bit ifHCOutOctets counter's value.
ifOutUcastPkts	The 32-bit wide total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. This object reports the low 32-bits of the 64-bit ifHCOutUcastPkts counter's value.
ifPhysAddress	The interface's address at its protocol sub-layer. For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object must contain an octet string of zero length.
ifPromiscuousMode	This object has a value of false(2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets/frames transmitted on the media. The value true(1) is only legal on certain types of media. If legal, setting this object to a value of true(1) may require the interface to be reset before becoming effective. The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface.
ifSpeed	An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object must contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object must report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object must be zero.
ifType	The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.

Table F.6 IP-MIB Supported Objects

Object Name	Object Description in MIB
icmpInAddrMaskReps	The number of ICMP Address Mask Reply messages received.

Table F.6 IP-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
icmpInAddrMasks	The number of ICMP Address Mask Request messages received.
icmpInDestUnreachs	The number of ICMP Destination Unreachable messages received.
icmpInEchoReps	The number of ICMP Echo Reply messages received.
icmpInEchos	The number of ICMP Echo (request) messages received.
icmpInErrors	The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).
icmpInMsgs	The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.
icmpInParmProbs	The number of ICMP Parameter Problem messages received.
icmpInRedirects	The number of ICMP Redirect messages received.
icmpInSrcQuenchs	The number of ICMP Source Quench messages received.
icmpInTimeExcds	The number of ICMP Time Exceeded messages received.
icmpInTimestampReps	The number of ICMP Timestamp Reply messages received.
icmpInTimestamps	The number of ICMP Timestamp (request) messages received.
icmpOutAddrMaskReps	The number of ICMP Address Mask Reply messages sent.
icmpOutAddrMasks	The number of ICMP Address Mask Request messages sent.
icmpOutDestUnreachs	The number of ICMP Destination Unreachable messages sent.
icmpOutEchoReps	The number of ICMP Echo Reply messages sent.
icmpOutEchos	The number of ICMP Echo (request) messages sent.
icmpOutErrors	The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value must not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.
icmpOutMsgs	The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.
icmpOutParmProbs	The number of ICMP Parameter Problem messages sent.
icmpOutRedirects	The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.
icmpOutSrcQuenchs	The number of ICMP Source Quench messages sent.
icmpOutTimeExcds	The number of ICMP Time Exceeded messages sent.
icmpOutTimestampReps	The number of ICMP Timestamp Reply messages sent.

Table F.6 IP-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
icmpOutTimestamps	The number of ICMP Timestamp (request) messages sent.
ipAdEntAddr	The IP address to which this entry's addressing information pertains.
ipAdEntBcastAddr	The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcasts addresses used by the entity on this (logical) interface.
ipAdEntIfIndex	The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of RFC 1573's ifIndex.
ipAdEntNetMask	The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.
ipAdEntReasmMaxSize	The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.
ipDefaultTTL	The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol.
ipForwarding	The indication of whether this entity is acting as an IP router in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP routers forward datagrams. IP hosts do not (except those source-routed via the host).
ipForwDatagrams	The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP routers, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route option processing was successful.
ipFragCreates	The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.
ipFragFails	The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set.
ipFragOKs	The number of IP datagrams that have been successfully fragmented at this entity.
ipInAddrErrors	The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
ipInDelivers	The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).

Table F.6 IP-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ipInDiscards	The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.
ipInHdrErrors	The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.
ipInReceives	The total number of input datagrams received from interfaces, including those received in error.
ipInUnknownProtos	The number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
ipNetToMediaIfIndex	The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of RFC 1573's ifIndex.
ipNetToMediaNetAddress	The IpAddress corresponding to the media-dependent `physical' address.
ipNetToMediaPhysAddress	The media-dependent `physical' address.
ipNetToMediaType	The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation- specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.
ipOutDiscards	The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.
ipOutNoRoutes	The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this `no-route' criterion. Note that this includes any datagrams which a host cannot route because all of its default routers are down.
ipOutRequests	The total number of IP datagrams which local IP user- protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams.
ipReasmFails	The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.
ipReasmOKs	The number of IP datagrams successfully re-assembled.

Table F.6 IP-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
ipReasmReqds	The number of IP fragments received which needed to be reassembled at this entity.
ipReasmTimeout	The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

Table F.7 SNMP-USER-BASED-SM-MIB Objects

Object Name	Object Description in MIB
usmStatsDecryptionErrors	The total number of packets received by the SNMP engine which were dropped because they could not be decrypted.
usmStatsNotInTimeWindows	The total number of packets received by the SNMP engine which were dropped because they appeared outside of the authoritative SNMP engine's window.
usmStatsUnknownEngineIDs	The total number of packets received by the SNMP engine which were dropped because they referenced an snmpEngineID that was not known to the SNMP engine.
usmStatsUnknownUserNames	The total number of packets received by the SNMP engine which were dropped because they referenced a user that was not known to the SNMP engine.
usmStatsUnsupportedSecLevels	The total number of packets received by the SNMP engine which were dropped because they requested a security Level that was unknown to the SNMP engine or otherwise unavailable.
usmStatsWrongDigests	The total number of packets received by the SNMP engine which were dropped because they didn't contain the expected digest value.
usmUserSpinLock	An advisory lock used to allow several cooperating Command Generator Applications to coordinate their use of facilities to alter secrets in the usmUserTable.
usmUserStatus	The status of this conceptual row. Until instances of all corresponding columns are appropriately configured, the value of the corresponding instance of the usmUserStatus column is 'notReady'. In particular, a newly created row for a user who employs authentication, cannot be made active until the corresponding usmUserCloneFrom and usmUserAuthKeyChange have been set. Further, a newly created row for a user who also employs privacy, cannot be made active until the usmUserPrivKeyChange has been set. The RowStatus TC [RFC2579] requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified: The value of this object has no effect on whether other objects in this conceptual row can be modified, except for usmUserOwnAuthKeyChange and usmUserOwnPrivKeyChange. For these 2 objects, the value of usmUserStatus MUST be active.

Table F.8 SNMPv2-MIB Supported Objects

Object Name	Object Description in MIB
snmpEnableAuthenTraps	Indicates whether the SNMP entity is permitted to generate authenticationFailure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authenticationFailure traps may be disabled. Note that it is strongly recommended that this object be stored in non-volatile memory so that it remains constant across re-initializations of the network management system.
snmpInASNParseErrs	The total number of ASN.1 or BER errors encountered by the SNMP entity when decoding received SNMP messages.
snmpInBadCommunityNames	The total number of SNMP messages delivered to the SNMP entity which used a SNMP community name not known to said entity.
snmpInBadCommunityUses	The total number of SNMP messages delivered to the SNMP entity which represented an SNMP operation which was not allowed by the SNMP community named in the message.
snmpInBadVersions	The total number of SNMP messages which were delivered to the SNMP entity and were for an unsupported SNMP version.
snmpInPkts	The total number of messages delivered to the SNMP entity from the transport service.
snmpProxyDrops	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity which were silently dropped because the transmission of the (possibly translated) message to a proxy target failed in a manner (other than a time-out) such that no Response-PDU could be returned.
snmpSetSerialNo	An advisory lock used to allow several cooperating SNMPv2 entities, all acting in a manager role, to coordinate their use of the SNMPv2 set operation. This object is used for coarse-grain coordination. To achieve fine-grain coordination, one or more similar objects might be defined within each MIB group, as appropriate.
snmpSilentDrops	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity which were silently dropped because the size of a reply containing an alternate Response-PDU with an empty variable-bindings field was greater than either a local constraint or the maximum message size associated with the originator of the request.
sysContact	The textual identification of the contact person for this managed node, together with information on how to contact this person. If no contact information is known, the value is the zero-length string.
sysDescr	A textual description of the entity. This value must include the full name and version identification of the system's hardware type, software operating-system, and networking software.
sysLocation	The physical location of this node (e.g., `telephone closet, 3rd floor'). If the location is unknown, the value is the zero-length string.

Table F.8 SNMPv2-MIB Supported Objects (Continued)

Object Name	Object Description in MIB
sysName	An administratively assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.
sysObjectID	The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining 'what kind of box' is being managed. For example, if vendor 'Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred Router'.
sysServices	A value which indicates the set of services that this entity may potentially offers. The value is a sum. This sum initially takes the value zero, Then, for each layer, L, in the range 1 through 7, that this node performs transactions for, 2 raised to (L - 1) is added to the sum. For example, a node which performs only routing functions would have a value of 4 ($2^{(3-1)}$). In contrast, a node which is a host offering application services would have a value of 72 ($2^{(4-1)} + 2^{(7-1)}$). Note that in the context of the Internet suite of protocols, values must be calculated accordingly: layer functionality 1 physical (e.g., repeaters) 2 datalink/subnetwork (e.g., bridges) 3 Internet (e.g., supports the IP) 4 end-to-end (e.g., supports the TCP) 7 applications (e.g., supports the SMTP) For systems including OSI protocols, layers 5 and 6 may also be counted.
sysUpTime	The time (in hundredths of a second) since the network management portion of the system was last re-initialized.

APPENDIX G

TCP/IP PROTOCOL HEADERS

This appendix is a brief overview of the common TCP/IP protocol headers.

The following sections are included:

- ▶ [Internet Protocol \(IP\)](#) (page 194)
- ▶ [Internet Control Message Protocol \(ICMP\)](#) (page 194)
- ▶ [Transmission Control Protocol \(TCP\)](#) (page 195)
- ▶ [User Datagram Protocol \(UDP\)](#) (page 195)

Internet Protocol (IP)

For the Internet Protocol (IP) specification, please refer to RFC791 available at <http://www.rfc-editor.org/>.

Table G.1 IP Datagram

bits 0 - 7		bits 8 - 15	bits 16 - 23	bits 24 - 31
Version (4 bits)	IP Header Length (4 bits)	Type of Service (8 bits)	Total Length (in number of bytes) (16 bits)	
IP Identification Number (16 bits)			Flags (3 bits)	Fragment Offset (13 bits)
Time to Live (8 bits)	Protocol Number (8 bits)		Header Checksum (16 bits)	
Source IP Address (32 bits)				
Destination IP Address (32 bits)				
Options (if any) + Padding (matching to 32-bit boundary)				
(data ...)				

Internet Control Message Protocol (ICMP)

For the Internet Control Message Protocol (ICMP) specification, please refer to RFC792 available at <http://www.rfc-editor.org/>.

Table G.2 ICMP Message

bits 0 - 7	bits 8 - 15	bits 16 - 23	bits 24 - 31
Type (8 bits)	Code (8 bits)	Checksum (16 bits)	
(data ...)			

Transmission Control Protocol (TCP)

For the Transmission Control Protocol (TCP) specification, please refer to RFC793 available at <http://www.rfc-editor.org/>.

Table G.3 TCP Segment

bits 0 - 7		bits 8 - 15						bits 16 - 23				bits 24 - 31			
Source Port Number (16 bits)						Destination Port Number (16 bits)									
Sequence Number (32 bits)															
Acknowledgement Number (32 bits)															
TCP Header Length (4 bits)	reserved (6 bits)	U	A	P	R	S	F	Window Size (16 bits)							
		R	C	S	S	Y	I								
		G	K	H	T	N	N								
Checksum (16 bits)						Urgent Pointer (16 bits)									
Options (if any) + Padding (matching to 32-bit boundary)															
(data . . .)															

User Datagram Protocol (UDP)

For the User Datagram Protocol (UDP) specification, please refer to RFC768 available at <http://www.rfc-editor.org/>.

Table G.4 UDP Datagram

bits 0 - 7		bits 8 - 15						bits 16 - 23				bits 24 - 31			
Source Port Number (16 bits)						Destination Port Number (16 bits)									
User Datagram Length (16 bits)						Checksum (16 bits)									
(data . . .)															

APPENDIX H

ASCII CHARACTER CODES

The decimal and hexadecimal values of the ASCII characters are presented for interpreting traffic captures and predefined Situation Contexts.

The following sections are included:

- ▶ [ASCII Character Codes](#) (page 198)
- ▶ [ASCII Control Codes](#) (page 199)

ASCII Character Codes

Table H.1 ASCII Character Codes

ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex
NUL	0	0x00	SPACE	32	0x20	@	64	0x40	`	96	0x60
SOH	1	0x01	!	33	0x21	A	65	0x41	a	97	0x61
STX	2	0x02	"	34	0x22	B	66	0x42	b	98	0x62
ETX	3	0x03	#	35	0x23	C	67	0x43	c	99	0x63
EOT	4	0x04	\$	36	0x24	D	68	0x44	d	100	0x64
ENQ	5	0x05	%	37	0x25	E	69	0x45	e	101	0x65
ACK	6	0x06	&	38	0x26	F	70	0x46	f	102	0x66
BEL	7	0x07	'	39	0x27	G	71	0x47	g	103	0x67
BS	8	0x08	(40	0x28	H	72	0x48	h	104	0x68
HT	9	0x09)	41	0x29	I	73	0x49	i	105	0x69
LF	10	0x0A	*	42	0x2A	J	74	0x4A	j	106	0x6A
VT	11	0x0B	+	43	0x2B	K	75	0x4B	k	107	0x6B
FF	12	0x0C	,	44	0x2C	L	76	0x4C	l	108	0x6C
CR	13	0x0D	-	45	0x2D	M	77	0x4D	m	109	0x6D
SO	14	0x0E	.	46	0x2E	N	78	0x4E	n	110	0x6E
SI	15	0x0F	/	47	0x2F	O	79	0x4F	o	111	0x6F
DLE	16	0x10	0	48	0x30	P	80	0x50	p	112	0x70
DC1	17	0x11	1	49	0x31	Q	81	0x51	q	113	0x71
DC2	18	0x12	2	50	0x32	R	82	0x52	r	114	0x72
DC3	19	0x13	3	51	0x33	S	83	0x53	s	115	0x73
DC4	20	0x14	4	52	0x34	T	84	0x54	t	116	0x74
NAK	21	0x15	5	53	0x35	U	85	0x55	u	117	0x75
SYN	22	0x16	6	54	0x36	V	86	0x56	v	118	0x76
ETB	23	0x17	7	55	0x37	W	87	0x57	w	119	0x77
CAN	24	0x18	8	56	0x38	X	88	0x58	x	120	0x78
EM	25	0x19	9	57	0x39	Y	89	0x59	y	121	0x79

Table H.1 ASCII Character Codes (Continued)

ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex
<i>SUB</i>	26	0x1A	:	58	0x3A	z	90	0x5A	z	122	0x7A
<i>ESC</i>	27	0x1B	;	59	0x3B	[91	0x5B	{	123	0x7B
<i>FS</i>	28	0x1C	<	60	0x3C	\	92	0x5C		124	0x7C
<i>GS</i>	29	0x1D	=	61	0x3D]	93	0x5D	}	125	0x7D
<i>RS</i>	30	0x1E	>	62	0x3E	^	94	0x5E	~	126	0x7E
<i>US</i>	31	0x1F	?	63	0x3F	_	95	0x5F	<i>DELETE</i>	127	0x7F

ASCII Control Codes

Table H.2 ASCII Control Codes

ASCII	Dec	Hex	Description
<i>NUL</i>	0	0x00	Null
<i>SOH</i>	1	0x01	Start of Heading
<i>STX</i>	2	0x02	Start of Text
<i>ETX</i>	3	0x03	End of Text
<i>EOT</i>	4	0x04	End of Transmission
<i>ENQ</i>	5	0x05	Enquiry
<i>ACK</i>	6	0x06	Acknowledge
<i>BEL</i>	7	0x07	Bell
<i>BS</i>	8	0x08	Backspace
<i>HT</i>	9	0x09	Horizontal Tabulation
<i>LF</i>	10	0x0A	Line Feed
<i>VT</i>	11	0x0B	Vertical Tabulation
<i>FF</i>	12	0x0C	Form Feed
<i>CR</i>	13	0x0D	Carrier Return
<i>SO</i>	14	0x0E	Shift Out
<i>SI</i>	15	0x0F	Shift In
<i>DLE</i>	16	0x10	Data Line Escape
<i>DC1</i>	17	0x11	Device Control 1

Table H.2 ASCII Control Codes (Continued)

ASCII	Dec	Hex	Description
<i>DC2</i>	18	0x12	Device Control 2
<i>DC3</i>	19	0x13	Device Control 3
<i>DC4</i>	20	0x14	Device Control 4
<i>NAK</i>	21	0x15	Negative Acknowledge
<i>SYN</i>	22	0x16	Synchronous Idle
<i>ETB</i>	23	0x17	End of Transmission Block
<i>CAN</i>	24	0x18	Cancel
<i>EM</i>	25	0x19	End of Medium
<i>SUB</i>	26	0x1A	Substitute
<i>ESC</i>	27	0x1B	Escape
<i>FS</i>	28	0x1C	File Separator
<i>GS</i>	29	0x1D	Group Separator
<i>RS</i>	30	0x1E	Record Separator
<i>US</i>	31	0x1F	Unit Separator

GLOSSARY

A

Access Control List

A list of Elements that can be used to define the Elements that an administrators with restricted permissions can access. See also [Administrator Role](#) and [Granted Element](#).

Action

What the firewall engine should do with a packet that matches the criteria for a particular rule in the security policy.

Action Option

Additional action-specific selections that affect how the traffic is handled set in the Action cell in rules.

Active Management Server

See [Primary Management Server](#) (page 219).

Address Range

A [Network Element](#) that defines a range of IP addresses. Use to avoid having to repeatedly type in the same IP addresses when defining address ranges that do not correspond to whole networks.

Address Resolution Protocol (ARP)

An Internet standard (RFC 826) protocol used to associate IP addresses with the media hardware address of a network interface card on a local area network (LAN).

Administrator

An [Element](#) that defines the details of a single person that is allowed to log on to the SMC using the Management Client. If used as a general term, Web Portal Users are also considered as administrators.

Administrator Role

An Element that defines which actions an [Administrator](#) with restricted permissions is allowed to take. See also [Granted Element](#) and [Permission Level](#).

Aggressive Mode

The authentication of two IPsec end-points with only three messages, as opposed to Main Mode's six. Aggressive mode also does not provide PFS support, and SA negotiation is limited. See [Main Mode](#) (page 216). See also [Security Association \(SA\)](#) (page 222).

AH (Authentication Header)

See [Authentication Header \(AH\)](#) (page 203).

Alert Chain

A list of rules defining which [Alert Channels](#) are used, and in which order, when an alert entry is directed to the Alert Chain from an [Alert Policy](#) to be escalated out from the Management Center. See also [Alert Escalation](#).

Alert Channel

A method of sending alerts out from the [Log Server](#). You can send alerts via SMTP (e-mail), SNMP, SMS text messages, or some other action you define in a custom script. Alert Channels are defined in the Log Server's properties, after which they can be used in [Alert Chains](#).

Alert Element

An [Element](#) that gives the name and description to an [Alert Event](#). The Alert element can be used as a matching criteria in the rules of an [Alert Policy](#).

Alert Entry

A log message with an alert status that has been raised based on some [Situation](#) (which you can see in the [Logs View](#)). Alert entries trigger [Alert Escalation](#).

Alert Escalation

Sending alerts out from the Management Center to administrators through [Alert Channels](#) (such as e-mail) according to a predefined [Alert Chain](#) until the original [Alert Entry](#) is acknowledged by some administrator in the [Logs View](#).

Alert Event

A pattern in traffic or a problem in the system's operation that matches to some [Situation](#) used in a policy or internally in the system, and thus triggers an [Alert Entry](#).

Alert Policy

A list of rules defining if an [Alert Entry](#) is escalated and which [Alert Chain](#) is used for escalating which type of alert entries. See also [Alert Escalation](#).

Alert Server

A StoneGate Management Center component that handles receiving and handling of Alerts. The Alert Server cannot be installed separately, it is integrated in the [Log Server](#) installation.

Alias

An [Element](#) that can be used to represent other network elements in configurations. It differs from a group element in that it does not represent all the elements at once: the value it takes in a configuration can be different on each engine where it is used.

Allow Action

An [Action](#) parameter that allows a connection matching that rule to pass through the firewall to its destination.

Analyzer

- 1) A device in the StoneGate IPS system that analyzes the log information from [Sensors](#) according to its policy to find patterns, so that separate log entries can be combined together.
- 2) The [Network Element](#) that represents an Analyzer device in the Management Center.

Antispoofing

Technique used to protect against malicious packages whose IP header information has been altered. See also [IP Spoofing](#) (page 214).

Application

A category of [Tags](#) for [Situations](#). Meant for grouping Situations that detect known vulnerabilities in a particular software application.

Application Layer Gateway; Application Level Firewall

A firewall system, or gateway, in which packets are examined based on the application protocol being used (e.g., telnet, FTP, SMTP). Proxies for each application-level service are installed on the gateway, and are often configured to relay a conversation between two systems. That is, a packet's destination is the gateway, which then establishes a separate connection to the other system to complete the connection.

Apply VPN Action

A Firewall [Action](#) parameter that directs traffic from protected local networks into the [Virtual Private Network \(VPN\)](#) tunnel and allows traffic that arrives through a VPN, but does not match non-VPN traffic from outside networks into the protected networks. See also [Enforce VPN Action](#) (page 209).

ARP (Address Resolution Protocol)

See [Address Resolution Protocol \(ARP\)](#) (page 201).

Asymmetric Encryption

A cryptographic technology that uses a pair of keys. The message is encrypted with the public half of a pair and can then be decrypted only with the matching private half of the key pair. Public key technology can be used to create digital signatures and deal with key management issues. Also referred to as public key encryption. See also [Symmetric Encryption](#) (page 225) and [Public-key Cryptography](#) (page 220).

Auditing

A Management Center feature that logs administrators' actions and allows administrators with unrestricted permissions to view and manage these logs to keep track of system changes.

Authentication

The process of proving that someone or something is who or what they claim to be. For example, typing a simple username-password combination is a form of authentication.

Authentication Header (AH)

A security protocol supported by the IPsec protocol to enhance traffic security. It enables the authentication and integrity of data against packet corruption or tampering. AH protocol can use SHA-1 or MD5 to generate a hash signature based on a secret component from the SA, the packet payload and some parts of the packet header. See also [Security Association \(SA\)](#) (page 222).

Authentication Token/Authenticator

A portable device for authenticating a user. Authentication tokens typically operate by challenge/response, time-based code sequences, or other techniques. One of the most commonly used tokens is the RSA SecurID card.

B

Authorization

The process of giving someone/something permission to do or have something. Usually related to authentication; once a user has authenticated (proved who they are), they are authorized (given permission) to perform certain actions.

Balancing Mode

A StoneGate cluster mode that attempts to divide the traffic as equally as possible between the online engines participating in the cluster. Confer to [Standby Mode](#) (page 225).

Bandwidth Management

The process of determining and enforcing bandwidth limits and guarantees for different types of traffic either together with [Traffic Prioritization](#) or on its own. Also see [QoS Class](#) (page 220) and [QoS Policy](#) (page 220).

Blacklisting

- 1) The process of blocking unwanted network traffic either manually or automatically.
- 2) Persistently blocking access to certain URLs manually.

Bookmark

A stored link to a view or layout in the [Management Client](#).

Bookmark Folder

A folder in the toolbar of the [Management Client](#) for storing and sharing [Bookmarks](#).

Boot Recovery

A StoneGate setting that brings the engines automatically back online after boot-up.

Border Routing

Routing of connections between different autonomous systems.

BrightCloud

A [Web Filtering](#) categorization service that provides categories for malicious sites as well as several categories for different types of non-malicious content that may be considered objectionable.

Buffer Overflow

When a program's data in the memory of a computer exceeds the space reserved for it (the buffer), data may in some circumstances be written on other parts of the memory area. Attackers may use buffer overflows to execute harmful program code on a remote system.

Bugtraq

A mailing list for discussing network security related issues, such as vulnerabilities.

Bulk Encryption Algorithm

Describes symmetric encryption algorithms which operate on fixed-size blocks of plaintext and generates a block of ciphertext for each.

CA

See [Certificate Authority \(CA\)](#) (page 205).

CAN

A candidate for a [CVE](#) entry.

Capture Interface

A [Sensor](#) interface that can listen to traffic passing in the network, but which is not used for routing traffic through the Sensor. See also [Inline Interface](#).

Category

A way of organizing elements and policies to display a specific subset at a time when configuring a large StoneGate system in the Management Client to make it easier to find the relevant elements when configuring the system. For example, a Managed Service Provider (MSP) who manages networks of several different customers can add a customer-specific category to each element and policy to be able to view one customer's elements and policies at a time.

Certificate

Electronic identification of a user or device. Certificates prove the user or device is who/what they claim to be. This is done through using public/private key pairs and digital signatures. Certificates are used in StoneGate for authenticating communications between the system components and for [Virtual Private Network \(VPN\)](#) authentication. Digital certificates are granted and verified by a [Certificate Authority \(CA\)](#), such as the internal CA included in the Management Server.

Certificate Authority (CA)

A trusted third-party organization or company that issues digital certificates, used to create digital signatures and public-private key pairs. The role of the CA in this process is to guarantee that the individual granted the unique certificate is, in fact, who he or she claims to be.

Challenge/Response

An authentication technique whereby a server sends an unpredictable challenge to the user, who computes a response using some form of authentication token, which can be an authenticator, or pre-shared keys used to encrypt random data.

Checksum

A one-way function applied to a file to produce a unique "fingerprint" of the file for later reference. File tampering can then be discovered by verifying the checksum value in the future.

CIS

See [Content Inspection Server \(CIS\)](#) (page 206).

Client

In a client-server architecture, a client is usually an application running on a computer or a workstation that uses services provided by a [Server](#).

Client Protection Certificate Authority

Contains the credentials that the engine uses to sign replacement server-side certificates the engine creates and presents to clients when inspecting the clients' HTTPS connections with external servers. Also see [Server Protection Credentials](#) (page 223).

Client-to-Gateway VPN

A [Virtual Private Network \(VPN\)](#) between a software client and a [Security Gateway \(SGW\)](#). Allows connecting mobile and home office workers safely to corporate resources using a secure (authenticated and encrypted) connection through insecure networks.

Cluster

A group of devices, or nodes, that share a given work load. In StoneGate, you can cluster Firewalls and Sensors to share the load and provide redundancy, allowing, for example, scheduled maintenance that takes one node out of service without interrupting services to the users.

Cluster Mode

Determines if all members of a cluster participate to traffic processing at all times ([Balancing Mode](#)) or if other members remain inactive until a traffic-processing member stops processing traffic ([Standby Mode](#)).

Cluster Virtual IP Address (CVI)

An IP and MAC address shared by all nodes in a cluster, which are used by every node in a cluster for communication. These interfaces give the cluster a single identity on the network, reducing the complexity of routing and network design. CVIs handle the traffic directed to the firewall for inspection in firewall clusters.

Combined Sensor-Analyzer

- 1) StoneGate IPS device that has both [Sensor](#) and [Analyzer](#) engines running simultaneously on the same hardware.
- 2) The [Network Element](#) that represents a Combined Sensor-Analyzer device in the Management Center.

Connection Tracking

The set of data maintained for a connection. Used for relating incoming packets to existing connections. Connection tracking information also includes information to necessary for [NAT \(Network Address Translation\)](#), [Load Balanced Routing](#) and [Protocol Agents](#). May also contain accounting information.

Contact Address

The IP address that is needed to contact a device performing a function in the StoneGate Management Center when there is [NAT \(Network Address Translation\)](#) being performed in between the two devices and thus the actual IP address assigned to the network interface cannot be used directly.

Content Inspection Server (CIS)

A server that performs detailed examination of a connection's data and assists in the determination to allow or discard packets. Common examples include virus scanning or filtering of Web URLs. Also known as *content screening*.

Continue Action

A policy parameter that sets default values to those used in the rule. The defaults are used in all subsequent rules except where specifically overridden until some other rule with the Continue action changes the values or the policy ends.

Context

An [Element](#) that is added to a [Situation](#) to define what the Situation should match. Provides a framework for defining parameters, which are most entered as a regular expression, or through a set of fields and options that the administrators adjust.

Correlation Situation

A [Situation](#) that defines the patterns that the [Analyzer](#) looks for when it examines event data produced by [Sensors](#).

CRL Server

A server that maintains a Certificate Revocation List (CRL), which can be used in [Authentication](#) to check if the certificate has been cancelled.

Custom Alert

An [Alert Element](#) that is defined by a StoneGate administrator, as opposed to a ready-made [Default Element](#) created by Stonesoft.

CVE

A dictionary that provides common names for publicly known information security vulnerabilities and exposures and thus a standardized description for each vulnerability that links the vulnerability information of different tools and databases.

CVI

See [Cluster Virtual IP Address \(CVI\)](#) (page 206).

D

Default Element

An [Element](#) that is present in the system at installation, or is added to the system during an upgrade or from a [Dynamic Update \(Package\)](#). Default elements cannot be modified or deleted by administrators, but they may be modified or deleted by dynamic update packages or upgrades.

Defragmentation

The process by which a large block of data is broken up into smaller pieces (datagrams), so that it can be packaged and transmitted by the underlying network technology ([Fragmentation](#)). Once the smaller pieces arrive at their destination, the datagrams are reassembled into the larger block of data (defragmentation).

DHCP (Dynamic Host Configuration Protocol)

A protocol for dynamically assigning IP addresses and other network information to an interface, based on BOOTP. A device on a network with no network information can broadcast a request for an IP address, subnet mask, default gateway and other information from a DHCP server on that same network. DHCP is defined in RFC 2131.

Diagram

An [Element](#) that contains one or more network diagrams created using the Diagram Editor.

Digital Certificate

See [Certificate](#) (page 205).

Discard Action

An [Action](#) parameter that stops all connections matching to the rule without sending any notification to the connecting host. Confer to [Refuse Action](#) (page 221).

Dispatch Clustering

See [Packet Dispatch](#) (page 218).

DMZ Network

A DMZ (DeMilitarized Zone Network) is a network separate from both internal and external networks, and connected through a gateway. Often used for isolating bastion hosts or publicly available machines, e.g., mail and HTTP servers are typically located on a DMZ network. Sometimes also referred to as a *screened subnetwork*.

DNS Spoofing

An attack method whereby the DNS name of a system is assumed by a malicious system, either by corrupting the name service cache of a victim, or by compromising a domain name server for a valid domain. The victim system is then directed to the malicious system instead of the original server.

Domain

Domains are administrative boundaries that allow you to separate the configuration details and other information in the system for the purpose of limiting administrator access.

DoS Attack (Denial of Service)

An attack with the objective of causing enough disruption in a computer system that its usability to legitimate users suffers. For example, an attacker may target a website so that it becomes overloaded, and slows down so much that it becomes unusable for people wishing to view it.

DSCP (DiffServ Code Point)

The Differentiated Services (DiffServ) Type of Service ([ToS Flag](#)) field added to packets in the network.

DSCP Mark

A field in [QoS Policy](#) rules that writes a particular [DSCP \(DiffServ Code Point\)](#) marker to the packets, if the QoS Policy is applied on the interface the packets use to exit the firewall.

DSCP Match

A field in [QoS Policy](#) rules that assigns the [QoS Class](#) specified in the rule to incoming packets that have a specific [DSCP \(DiffServ Code Point\)](#) marker set, if the QoS Policy is applied on the interface the packets use to enter of the firewall.

Dynamic IP address

An IP address that is assigned by using the [DHCP \(Dynamic Host Configuration Protocol\)](#).

Dynamic NAT

A way to translate network addresses, where for each original address, a translated address and possibly a port are selected dynamically from a predefined pool.

Dynamic Update (Package)

A file supplied by Stonesoft that provides updates to [Default Elements](#) and policies, most importantly to the [Situation](#) and [Vulnerability](#) information that is used for traffic inspection in [Inspection Rules](#).

E

Element

A StoneGate object representing the equipment in your physical networks or some area or concept of configuration. Elements may, for example, represent a single device such as a server, a range of IP addresses, or some configuration aid in the Management Center, such as a Category. Also see [Network Element](#) (page 218).

Encryption

Used for data security, encryption translates any data into a secret code. Public-key encryption and symmetric encryption are the main types of encryption. Decrypting ciphertext (encrypted data) into plaintext requires access to a secret key.

Encryption Domain

Networks that are defined to be behind a certain VPN gateway in a [Virtual Private Network \(VPN\)](#) configuration.

Encryption Key

The data that is used to convert plaintext to ciphertext. In symmetric algorithms, the same key is the decryption key as well. In public key algorithms, a different, but related key is used to convert the ciphertext back into plaintext.

Encryption Policy

Settings that define which encryption and authentication methods are used to establish a [Virtual Private Network \(VPN\)](#).

Enforce VPN Action

A firewall [Action](#) parameter that directs traffic from protected local networks into the [Virtual Private Network \(VPN\)](#) tunnel and allows traffic that arrives through a VPN, and drops any non-VPN traffic from external networks to the local network that matches the rule. See also [Apply VPN Action](#) (page 203).

Engine

The device that runs firewall, sensor, or analyzer software; a standard server or a StoneGate appliance. Represented by a [Node](#) in the Management Client.

Ethernet Rules

A set of rules in the [IPS Policy](#) that define which Ethernet traffic is allowed or discarded by a [Sensor](#) in [Transparent Access Control Mode](#).

Expression

An [Element](#) that can be used to accurately define a whole complex set of elements by including and excluding elements using logical expressions.

External Gateway

Any [Security Gateway \(SGW\)](#) that is managed by a different [Management Server](#) than the one on which the [Virtual Private Network \(VPN\)](#) is being configured.

F

Filter

A description of log fields and their values combined together using operators for the purpose of sorting in or out log, alert, and audit entries. Used, for example, to filter out logs from the display in the [Logs View](#) so that those entries that are interesting at the moment can be found more easily.

Firewall

- 1) A [Network Element](#) that represents the firewall device in the Management Center. Either a [Single Firewall](#) or a [Firewall Cluster](#).
- 2) The device running the StoneGate firewall software.

Firewall Cluster

A Group of two or more [Firewall Engines](#) that work together as if they were a single unit.

Firewall Engine

The device that runs firewall software; a standard server or a StoneGate appliance Represented by the [Firewall Node](#) in the Management Client.

Firewall Node

An individual [Firewall Engine](#) in the Management Client, representing a device that runs firewall software as part of a [Firewall Cluster](#) or a [Single Firewall](#).

Forward Action

A firewall [Action](#) parameter that directs traffic from protected local networks or from a [Virtual Private Network \(VPN\)](#) tunnel into another VPN tunnel.

Fragmentation

The process by which a large block of data is broken up into smaller pieces (datagrams), so that it can be packaged and transmitted by the underlying network technology (fragmentation). Once the smaller pieces arrive at their destination, the datagrams are reassembled into the larger block of data ([Defragmentation](#)).

G

Gateway

A device that provides VPN access for other devices.

Gateway Certificate

A [Certificate](#) used for authenticating a [Gateway](#) to other Gateways and [VPN Clients](#) in a VPN.

Gateway Profile

An element that defines a set of VPN-related capabilities that a VPN [Gateway](#) supports.

Gateway Settings

An element that contains general settings for StoneGate firewall/VPN engines related to VPN performance.

Gateway-to-Gateway VPN

In StoneGate, a [Virtual Private Network \(VPN\)](#) element which is set up so that the VPN is established between two gateway devices providing connectivity to networks behind the gateways.

Geolocation

Elements that define a geographical location of an IP address. Used for illustrating networks and network traffic on a map and other informative purposes in the [Management Client](#).

Granted Element

An [Element](#) or [Security Policy](#) that an administrator has been given permission to edit and install when their [Administrator Role](#) would otherwise prevent them from doing so.

Group

A [Network Element](#) that includes other elements and represents them all at once in policies and other parts of the configuration. For example, you can define a Group of several WWW-servers, and then use the Group element in policies when you need to make a rule that concerns all of the WWW-servers.

H

Hardware

A category of [Tags](#) for [Situations](#). Meant for grouping Situations that detect known vulnerabilities in applications that run on a particular hardware platform.

Hash Signature

A cryptography-related concept that refers to a digital fingerprint associated with a given message and computed with one-way algorithms. Hash signatures are used to secure the integrity of encrypted data, ensuring that no tampering has taken place during transmission. See also [Client-to-Gateway VPN](#) (page 206), and [SHA-1](#) (page 223).

Heartbeat

A protocol that the nodes of a [Firewall Cluster](#) or [Sensor Cluster](#) use to monitor each other and for other tasks that are needed for collaboration between each [Node](#).

High Availability

The implementation of clustering technology, hot standby technology, or general redundancy in a system to increase the availability of an application, service, or network beyond what a single system is capable of providing. Increased availability is achieved by eliminating all single points of failure, with clustering technology providing the highest level of availability.

Host

1) A [Network Element](#) that represents any single device that has an IP address.

2) Any device connected to a TCP/IP network, including the Internet, with one or more IP addresses. Hosts are distinguishable from gateways or routers, in that they do not forward, or route, packets to other networks.

Hot Standby

A solution where one node handles the work load with the support of a back-up node, which takes over connections in case of failure in the first node.

Hybrid Authentication

A system using both [Asymmetric Encryption](#) and [Symmetric Encryption](#). Asymmetric techniques are used for key management and digital signatures. The symmetric algorithms are used to encrypt the bulk of data with reduced strain on resources.

IKE Proposal

The suggested encryption algorithms, authentication methods, hash algorithms, and Diffie-Hellman information in the Security Association (SA) component of an IPsec VPN. The initiator of an IPsec tunnel can make multiple proposals, but the responder only sends one proposal in return. See also [Internet Key Exchange \(IKE\)](#) (page 213) and [Security Association \(SA\)](#) (page 222).

Incident Case

An [Element](#) that administrators can use to gather together all the data, actions, system configuration information, and files related to a specific incident of suspicious activity.

Incident History

A collection of all the logs and audit entries that track actions performed in a particular [Incident Case](#) window.

Info Panel

A tab in [Management Client](#) windows that shows information on the selected element or other object. The Info view shows, for example, the nodes belonging to a selected cluster.

Inherited Rule

A rule either hidden or shown on a grey background in a [Security Policy](#) or [Template Policy](#) which has been added in a template higher up in the policy hierarchy so that it has been passed down to the security policy or template policy. Inherited rules are enforced just as any other rules, but they can be edited only in the template where the rule was originally added.

Inline Interface

A Sensor interface that combines together two physical interfaces, enabling the traffic to be routed through as if the Sensor was an extension of the network cable, but allowing the Sensor to actively monitor packets and connections and stop them according to its [Actions](#) and [Inspection Rules](#).

Insert Point

The place in a [Security Policy](#) or [Template Policy](#) where new rules can be inserted when no rules have been inserted in that place yet (shown as a green row) or the place in a template policy where rules can be inserted in inheriting policies and template policies (shown as an orange row).

Inspection Rule

The definitions on the Inspection tab in a Firewall or IPS policy that defines options for deeper inspection and reactions to traffic accepted in [Actions](#). The matching in Inspection rules is done based on matching information provided by [Situation](#) elements. Confer to [Action](#) (page 201).

Internal Gateway

A StoneGate [Firewall](#)/VPN engine that are managed by the same [Management Server](#) on which the [Virtual Private Network \(VPN\)](#) is being configured.

Internal Network

The networks and network resources that StoneGate is protecting. In StoneGate, there is no concept of internal and external networks in the system.

Internet Key Exchange (IKE)

A protocol defined by the [IPsec \(IP Security\)](#) standard for securely exchanging key-related information between connecting hosts when establishing a [Virtual Private Network \(VPN\)](#).

Internet Service Provider (ISP)

A company that provides Internet connectivity to subscribers.

Intrusion Detection System (IDS)

A system that monitors network traffic for determining, and making administrators aware of data security exploits or attempts by providing logs or other network information. Confer to [Intrusion Prevention System \(IPS\)](#).

Intrusion Prevention System (IPS)

A system that monitors network traffic (like an [Intrusion Detection System \(IDS\)](#)) and has the capability of actively stopping traffic if it is deemed malicious or otherwise unwanted.

IP Address Bound License

A [License](#) file for the engines that includes the information on the IP address of the component it licenses. If you need to change the IP address of the component, you must request an IP address change at the Stonesoft Licensing website. On engines, an alternative to a [Management Bound License](#) (page 216).

IPComp (IP Payload Compression Protocol)

A protocol used to reduce the size of IP datagrams. Increases the overall communication performance between a pair of communicating gateways by compressing the datagrams, provided the nodes have sufficient computation power, and the communication is over slow or congested links. IPComp is defined in RFC 2393.

IP Splicing (or Hijacking)

An attack performed by intercepting and using an active, established session. Often occurs after the authentication phase of the connection is complete, giving the attacker the permissions of the original, authenticated user. Encryption at the session or network layer is typically the best defense from such an attack.

IP Spoofing

A technique used to obtain unauthorized access to computers by sending connection requests with tampered headers, simulating a trusted source.

IPsec (IP Security)

A set of protocols supporting secure exchange of packets. Used for the implementation of [Virtual Private Network \(VPN\)](#) solutions when high performance and/or support for a wide variety of protocols are needed. IPsec provides transport and tunnel encryption modes. IPsec is defined in RFC 2401.

IPsec Proposal

Suggested encryption algorithms, hash algorithms, authentication methods, etc. to be used for an [IPsec \(IP Security\)](#) tunnel. See also [IKE Proposal](#) (page 212).

IPS Policy

The [Security Policy](#) for IPS Sensors and Analyzers containing the [Action](#) and [Inspection Rule](#) definitions that determine how traffic is inspected and how the system reacts when a match is found.

IPv4 Access Rule

A row in a Firewall or IPS policy that defines how one type of IPv4 connection is handled by providing matching criteria based on the source, destination, and protocol information. Confer to [IPv6 Access Rule](#) (page 214).

IPv6 Access Rule

A row in an IPS policy that defines how one type of IPv6 connection is handled by providing matching criteria based on the source, destination, and protocol information. Confer to [Action](#) (page 201).

ISAKMP (Internet Security Association Key Management Protocol)

An open-ended encoding protocol necessary for IKE negotiation when establishing Security Associations. See also [Security Association \(SA\)](#) (page 222).

ISP (Internet Service Provider)

See [Internet Service Provider \(ISP\)](#) (page 213).

J

Journal

A tool in the [Incident Case](#) window that allows administrators to create a permanent record of their actions while investigating an incident.

Jump Action

A [Security Policy](#) parameter that directs the inspection to a [Sub-Policy](#), against which connections matching the rule with the Jump action are checked. Can be used to speed up traffic processing, as connections that do not match the Jump rules are not checked against rules in the sub-policies.

L

License

Files you import to the system to tell the [Management Server](#) that the components you have installed have been legally purchased. You generate the Licenses at the Stonesoft Licensing website and import them to the Management Server using the Management Client.

Lifetime

The interval at which the IPsec participants should begin to negotiate a replacement [Security Association \(SA\)](#) (soft lifetime) or the interval at which the current SA for an IPsec tunnel is no longer valid (hard lifetime) in a [Virtual Private Network \(VPN\)](#).

Load Balancing

A process for distributing work evenly across multiple, available devices to avoid overwhelming any single system.

Load Balancing Filter

A software component that determines which network connections should be handled by a particular node in a cluster, based on address information, current load, performance of individual machines, and other factors.

Load Balanced Routing

A method for choosing routes to destinations based on determining the fastest response time through multiple gateways. The application of [Multi-Link](#) technology to determine which network link provides the best round trip time.

Load Sharing

The distribution of work between multiple devices. Similar to [Load Balancing](#), but not as effective, since the techniques used do not ensure an *equal* distribution of the work load. Load sharing is typically a static method of distributing a load, whereas load balancing is often a dynamic method.

Location

An [Element](#) that groups together system components that are on the same side of a device doing [NAT \(Network Address Translation\)](#). Used to define [Contact Addresses](#) for components that communicate within the StoneGate Management Center.

Logging Options

A selection available in all rules in policies that determines if and how a record is created when the rule matches.

Logging Profile

Defines how the Log Server converts [Syslog](#) data received from a particular type of third-party component into StoneGate log entries.

Log Server

A component of the [Management Center](#) responsible for storing and managing log (and alert) data.

Log Spool

A temporary storage area in a firewall node for log data before it is sent to a [Log Server](#).

Logical Interface

An IPS [Element](#) used in the IPS policies to represent one or more physical network interfaces as defined in the [Sensor](#) properties.

Logs View

A tool that allows browsing logs, alerts, audit data, and connections each in an adapted version of the same user interface.

M

Main Mode

An IKE negotiation mode, which exchanges six messages between the end-points of an IPsec tunnel to complete the negotiation of authentication and keys for a [Virtual Private Network \(VPN\)](#). Optionally, Perfect Forward Secrecy (PFS) can be applied to protect further negotiations. See also [Aggressive Mode](#) (page 201) and [Perfect Forward Secrecy \(PFS\)](#) (page 219).

Malware

Malicious software designed to infiltrate or damage a computer system.

Management Bound License

A [License](#) file for StoneGate engines that is based on information on the Management Server's [Proof of License \(POL\)](#) code. An alternative to an [IP Address Bound License](#) (page 213).

Management Center

The system consisting of a [Management Server](#), one or more [Log Servers](#) and none to several Web Portal Servers that is used to manage the [Firewall Engines](#), and to store and manage traffic and system related data.

Management Client

A graphical user interface component that provides the tools for configuring, managing, and monitoring the firewalls, sensors, analyzers, and other components in the StoneGate system. The Management Client connects to the [Management Server](#) to provide these services based on the [Administrator](#) information that you use when launching the Management Client software.

Management Network

The network used for communication between firewalls, Management Servers, Log Servers and the Management Client.

Management Server

A system component that stores all information about the configurations of all firewalls, sensors, analyzers, and other StoneGate components in the system, monitors their state, and provides access for Management Clients when administrators want to change the configurations or command the engines. The most important component in the system.

Maximum Transmission Unit (MTU)

The largest physical size of a datagram that can be transmitted over a network without fragmentation. Often expressed in bytes, it can apply to frames, packets, cells or other media, depending on the underlying topology.

Modem Interface

A firewall interface that defines the settings of a 3G modem that provides a wireless outbound link for a [Single Firewall](#).

Monitored Element

A StoneGate server or engine component that is actively polled by the Management Server, so that administrators can keep track of whether it is working or not. All StoneGate system components are monitored by default.

Monitoring Agent

A software component that can be installed on servers in a [Server Pool](#) to monitor the server's operation for the purposes of [Traffic Management](#).

Multicast

A technique by which a set of packets are sent to a group of machines sharing a common address. Unlike broadcast, it does not include all machines, and unlike unicast, it usually has more than one member of the group.

Multi-Layer Inspection

A hybrid firewall technology that incorporates the best elements of application level and network level firewalls, with additional technology to enable the secure handling of many connection types.

Multi-Link

Patented Stonesoft technology to connect one site to another, or to the Internet, using more than one network link. Applications of Multi-Link technology include inbound and outbound traffic management for unencrypted as well as VPN traffic. See also [Outbound Multi-link](#) (page 218).

N

NAT (Network Address Translation)

A mechanism for assigning local networks a set of IP addresses for internal traffic and another for external traffic. It increases security by hiding internal IP addresses and enables hosts with "invalid" (non-routable) addresses to communicate on the Internet.

NDI

See [Node Dedicated IP Address \(NDI\)](#) (page 218).

NetLink

An [Element](#) used for implementing routing of StoneGate's [Multi-Link](#) features. NetLinks can represent any IP-based network links (such as ISP routers, xDSL, leased lines, dial-up modems). NetLinks are combined together into an [Outbound Multi-link](#).

Network Element

1) All [Elements](#) that represent one or more components that have an IP address, that is, a general category ('Network Elements') for those elements that represent physical devices and networks in StoneGate.

2) The Network Element called 'Network' that represents a (sub)network of computers. Used for rules and configurations that are common for all hosts in a specific (sub)network.

Network Scan

A stage of an attack in which the attacker scans the target to enumerate or map the directly-connected network(s).

Node

The representation of an individual firewall, sensor or analyzer [Engine](#) in the Management Client.

Node Dedicated IP Address (NDI)

A unique IP address for each machine. The only interface type for single firewalls. Not used for operative traffic in firewall clusters and sensors. Firewall clusters use a second type of interface, [Cluster Virtual IP Address \(CVI\)](#), for operative traffic. Sensors have two types of interfaces for traffic inspection, the [Capture Interface](#) and the [Inline Interface](#).

O

Operating System

A category of [Tags](#) for [Situations](#). Meant for grouping Situations that detect known vulnerabilities in a particular operating system or applications that run on that operating system.

Outbound Multi-link

An [Element](#) used for combining [NetLinks](#) for load balancing outbound traffic. The NetLinks included in a Outbound Multi-link element are frequently tested to determine which is the fastest NetLink for new outbound connections.

P

Packet

A segment of data sent across a network that includes a header with information necessary for the transmission, such as the source and destination IP addresses.

Packet Dispatch

A [Cluster Virtual IP Address \(CVI\)](#) mode in which only one node in the cluster receives packets. This dispatcher node then forwards the packets to the correct node according to [Load Balancing](#), as well as handles traffic as a normal node. The recommended cluster mode for new installations.

Packet Filtering

A method of controlling access to a network, or set of networks, by examining packets for source and destination address information, and permitting those packets to pass, or halting them based on defined rules.

Packet Sniffer

See [Sniffer](#) (page 224).

Perfect Forward Secrecy (PFS)

A property of IKE transactions that enhances the secrecy of keys, but requires additional processing overhead. PFS ensures that the distribution of key-related information remains independent from previously existing key material. See also [Internet Key Exchange \(IKE\)](#) (page 213).

Permission Level

The general level of rights that an [Administrator](#) has. Permissions are customized with [Administrator Roles](#) and [Granted Elements](#).

Permit Action

An [Inspection Rule](#) action that stops the inspection of all traffic that matches to the rule that uses the Permit action and lets the traffic continue to its destination.

Phishing

A [Social Engineering](#) attack in which a malicious e-mail or web page attempts to solicit sensitive information such as usernames, passwords, and credit card details by masquerading as coming from a trustworthy entity.

Player

Any element or IP address that was involved in an incident that is being investigated using the [Incident Case](#) element.

Policy

A container for the Access rules, Inspection rules, and NAT rules.

Policy Routing

User-defined routing based on information that is not normally used in routing, such as the source IP address, port information, or service type.

Policy Snapshot

A record of policy configuration that shows the configuration in the form that it was installed or refreshed, including the rules of the policy, the elements included and their properties, as well as the time when the policy was uploaded, and which administrator performed the upload. Helps in keeping track of configuration changes.

Port Address Translation (PAT)

A process, similar to [NAT \(Network Address Translation\)](#), where the source or destination port is changed to a different port. PAT is often used to disguise, or masquerade a service in place of another. See also [NAT \(Network Address Translation\)](#) (page 217).

Pre-shared Key

A string of characters that is stored on two (or more) systems and that is used for authenticating or encrypting communications between the systems.

Primary Management Server

The Management Server that is actively used for configuring the system in a system that has at least one [Secondary Management Server](#).

Probing Profile

Settings that define how a Log Server monitors third-party components.

Proof of License (POL)

A code used for verifying the legitimate purchase of StoneGate software products. Used for generating [License](#) files at the Stonesoft website.

Proof of Serial Number (POS)

Identification code attached to StoneGate appliances.

Protocol

An element that is used inside [Service](#) elements to specify a [Protocol Agent](#) for the Firewall [Actions](#) and the protocol of the traffic for the [Inspection Rules](#).

Protocol Agent

A process on the firewalls that assists the engine in handling a particular [Protocol](#). Protocol Agents ensure that related connections for a service are properly grouped and evaluated by the firewall engine, as well as assisting the engine with content filtering or network address translation tasks. See also [Connection Tracking](#) (page 206).

Protocol Tag

A type for [Protocol](#) elements that are only used to define the protocol of traffic for inspection against the inspection rules. Confer to [Protocol Agent](#).

Proxy ARP

Proxy ARP option on a device that does routing means that the device relays broadcast messages between two hosts that are in separate physical networks, but still have IP addresses from the same network. This proxy is needed for the ARP requests, as broadcast messages are not normally relayed from one network to another. See also [Address Resolution Protocol \(ARP\)](#) (page 201).

Pruning

Deleting log entries according to [Filters](#) either as the logs arrive on the Log Server or before they are stored (after displaying them in the current view in the Logs view).

Public-key Cryptography

A cryptographic system that uses a pair of keys: a public key, used to encrypt a message, and a private (secret) key that can decrypt the message. This is also called asymmetric encryption.

Q

QoS Class

An [Element](#) that works as a link between a rule in a [QoS Policy](#) and one or more firewall [Actions](#). The traffic allowed in the access rule is assigned the QoS Class defined for the rule, and the QoS class is used as the matching criteria for applying QoS Policy rules.

QoS Policy

A set of rules for [Bandwidth Management](#) and [Traffic Prioritization](#) for traffic that has a particular [QoS Class](#), or rules for assigning QoS Classes based on a [DSCP Match](#) found in the traffic.

Refragmentation

A technique to fragment outbound packets from the firewall in the same manner in which they were fragmented when the firewall received them. See also [Virtual Defragmentation](#) (page 228).

Refuse Action

An [Action](#) parameter that blocks the packet that matches the rule and sends an error message to the originator of the packet. Confer to [Discard Action](#) (page 208).

Regular Expression

A string that describes a set of strings. Used in many text editors and utilities to search for text patterns and, for example, replace them with some other string. In StoneGate, regular expressions are used, for example, for defining patterns in traffic that you want a certain [Situation](#) to match when you give the Situation a [Context](#) that calls for a Regular Expression.

Related Connection

A connection that has a relationship to another connection defined by a [Service](#). For example, the FTP protocol defines a relationship between a control connection, and one or more data connections at the application level. The firewall may be required to allow a connection that would otherwise be discarded, if it is related to an already allowed connection.

Request for Comments (RFC)

A document that outlines a proposed standard for a protocol. RFCs define how the protocol should function, and are developed by working groups of the Internet Engineering Task Force (IETF), and reviewed and approved by the Internet Engineering Steering Group (IESG). See <http://www.rfc-editor.org/>.

Retained License

A [Management Bound License](#) that has been used to install a policy on an engine and has then been unbound without relicensing or deleting the engine the license was bound to. Retained licenses cannot be bound to any engine before the engine the license was previously bound to is deleted or has a new policy refresh with a valid license.

RFC

See [Request for Comments \(RFC\)](#).

Rootkit

A set of tools that intruders to computer systems use for hiding their presence and the traces of their actions.

Route

The set of routers or gateways a packet travels through in order to reach its destination. In TCP/IP networks, individual packets for a connection may travel through different routes to reach the destination host.

Router

A [Network Element](#) representing a physical router in your network. Most often used to indicate next-hop routers in the Routing view and in Network Diagrams.

Routing Table

A database maintained on every router and gateway with information on paths to different networks. In StoneGate, the routing table is represented graphically in the Routing view.

Rule

An expression used to define the eventual outcome of packets arriving at the firewall, which match certain conditions (e.g., source and destination address, protocol, user).

Rules Tree

The main configuration tool for adjusting [Inspection Rule](#) definitions.

S

SA (Security Association)

See [Security Association \(SA\)](#) (page 222).

Scan

See [Network Scan](#) (page 218).

Secondary IP address

An IP address used for identifying an element with multiple addresses as a source or destination of traffic, defined in addition to a primary IP address.

Secondary Log Server

A [Log Server](#) defined as a backup channel for components that primarily send their logs to some other Log Server.

Secondary Management Server

A redundant [Management Server](#) that replicates the configuration data from the [Primary Management Server](#) under normal conditions so that the services offered by the Management Server can be used without interruption if components fail or are otherwise unavailable.

Secret Key Cryptography

See [Symmetric Encryption](#) (page 225).

Security Association (SA)

A unidirectional, logical connection established for securing [Virtual Private Network \(VPN\)](#) communications between two sites. A security association records the information required by one site to support one direction of the IPsec connection whether inbound or outbound. It uses transport mode for communications between two hosts and tunnel mode for communication between security gateways. See also [Authentication Header \(AH\)](#) (page 203).

Security Gateway (SGW)

A device, typically a firewall, that performs encryption/decryption on [Virtual Private Network \(VPN\)](#) packets sent between [Sites](#) through untrusted networks.

Security Parameter Index (SPI)

A value used by AH and ESP protocols to help the firewall cluster select the security association that will process an incoming packet. See also [Authentication Header \(AH\)](#) (page 203).

Security Policy

The set of templates, policies, and sub-policies together or individually that define what traffic is acceptable and what traffic is unwanted. Policies are defined using the Management Client, stored on the Management Server and installed on firewalls, sensors, and analyzers, which then use their installed version of the policies to determine the appropriate action to take regarding packets in the network.

Sensor

A StoneGate IPS component that captures all the traffic from a physical network link, inspects it according to its policy, and if installed inline, selects which connections are allowed to continue. Provides data for the Analyzer (see [Analyzer](#) (page 202)).

Sensor Cluster

Group of two or more IPS Sensor nodes that work together as if they were a single Sensor.

Server

1) A [Network Element](#) representing a physical server in your network. Generally, server elements are only defined to configure a specific server for use with the [Management Center](#) (such as a RADIUS server used for authenticating administrators), but generic Servers can be used in Network Diagrams instead of [Host](#) elements to better illustrate the network layout.

2) In a client-server architecture, a computer that is dedicated for running services used by [Client](#) computers. The services may include, for example, file storage, e-mail, or web pages.

Server Pool

A [Network Element](#) representing a group of [Servers](#). Used for inbound traffic management.

Server Protection Credentials

An element that stores the private key and certificate of an internal HTTPS server. The private key and certificate allow the engine to present itself as the server to clients so that the engine can decrypt and inspect incoming HTTPS traffic. Also see [Client Protection Certificate Authority](#) (page 206).

Service

An [Element](#) that is used for matching traffic to an application level protocol, for example, FTP, HTTP or SMTP. The TCP and UDP Services also determine the port number. Service elements are used in policies to make the rule match only a particular protocol, to enable [Protocol Agents](#), and select traffic to be matched against [Inspection Rules](#).

Session Stealing

See [IP Splicing \(or Hijacking\)](#) (page 214).

SHA-1

A cryptographic algorithm used for hash functions. It generates a 160-bit signature from an input of any length. See also [Hash Signature](#) (page 211).

Single Firewall

A firewall that has only one [Firewall Engine](#).

Single Point of Failure

The point at which the failure of a single device or component of a system will lead to either the failure of the entire system, or the inability to use services normally provided by that system. Redundant systems, using high availability technologies, eliminate single points of failure.

Site

A set of resources protected by StoneGate.

Situation

- 1) An [Element](#) that identifies and describes detected events in the traffic or in the operation of the system. Situations contain the [Context](#) information, i.e., a pattern that the system is to look for in the inspected traffic.
- 2) An [Inspection Rule](#) cell where Situation elements are inserted.

Situation Type

A category of [Tags](#) for [Situations](#). Meant for indicating what kind of events the associated Situations detect (for example, Attacks, Suspicious Traffic).

Sniffer

A device or program that captures data traveling over a network. Sniffers are often used for troubleshooting network problems, as they can show the packet flow taking place. They can also be used maliciously to steal data off a network.

SNMP Agent

A software component that sends SNMP traps when specific events are encountered.

Social Engineering

An attack involving trickery or deception for the purpose of manipulating people into performing actions or divulging confidential information.

SPI (Security Parameter Index)

See [Security Parameter Index \(SPI\)](#) (page 222).

SSH (Secure Shell)

A program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. Often used as a replacement for insecure programs such as `telnet` or `rsh`. In StoneGate, SSH can be used for remotely accessing the engine command line.

SSL VPN

A VPN technology that utilizes SSL encryption to secure users' remote access to specific applications. Allow authenticated users to establish secure connections to a limited number of specific internal services through a standard web browser ("clientless" access) or through a client application that allows a wider range of services.

Standby Management Server

See [Secondary Management Server](#) (page 222).

Standby Mode

An operating state of a StoneGate cluster that keeps one node online and the rest in standby, so that [State Synchronization](#) is done, but node does not process the traffic. If the online node is taken offline or fails, one of the standby nodes takes over the existing connections.

State Synchronization

The communication of connection tracking information between several firewall nodes in a cluster. Can be either a full synchronization, where all connection tracking information is transferred to the other nodes of a cluster, or an incremental synchronization, where only the information on connections changed after the last synchronization are transferred. See also [Connection Tracking](#) (page 206).

Static IP address

IP address that is typed in by a user or an administrator, and which does not change without their action.

Static NAT

[NAT \(Network Address Translation\)](#) where for each original address, there is a single, predefined translated address.

Static Routing

A form of routing that has permanent routes between networks programmed into every [Routing Table](#).

Sub-Policy

A set of rules that are separated from the main policy, based on some common category, such as the service or the destination IP address. In this way, related rules can be grouped together to make the entire policy easier to understand. Because subrules are only processed if the general rule in the main policy matches, the overall processing time is improved.

Subtunnel

The actual tunnels that are combined logically within a multi-route VPN tunnel in a StoneGate [Multi-Link](#) environment. They represent all possible routes that connect the end-points of the security gateways between which a [Virtual Private Network \(VPN\)](#) is formed. The individual subtunnels may connect the two gateways through different network links.

Symmetric Encryption

An Encryption mechanism that uses the same shared secret key for encrypting and decrypting messages. It is often referred to as symmetric bulk encryption since it processes large amounts of data rather quickly. Also known as conventional or secret key cryptography. There are two main types of symmetric encryption algorithms, bulk and stream encryption (also known as block ciphers and stream ciphers). Common symmetric algorithms are DES and 3DES. See also [Asymmetric Encryption](#) (page 203).

Syslog

A standard protocol for exchanging logs between network components. Defined in RFC 5424.

System Summary

A panel in the System Status view that provides a general summary view of the current status of the monitored elements according to the component type.

Tag

An [Element](#) for organizing [Situations](#). Tags can also be used in [Inspection Rules](#), in the Situation cell, to represent all Situations marked with that Tag.

Takeover Period

The time interval during which the active nodes in a firewall or sensor cluster collaborate to redistribute the work load of a failed node.

Task

An [Element](#) that allows you to schedule commands to run automatically at a convenient time.

Template Policy

A combination of rules and [Insert Points](#), which is used as a basis when creating policies or other template policies. Policies and template policies created from a particular template policy then inherit all the rules from that template policy and any of the template policies higher up in the inheritance hierarchy. The [Inherited Rules](#) cannot be edited within the inheriting policy. Used, for example, by high-privilege Administrators to restrict changes administrators with a lower [Administrator Role](#) can make to rules.

Temporary Filter

A log filter that is created from details of entries in the [Logs View](#) or the Connections view, and which is only available until the view is closed.

Terminate Action

An [Inspection Rule](#) parameter that stops or attempts to stop the connection matching to the rule according to the [Action Option](#) selected and the whether the [Engine](#) where the rule matching occurs is capable of stopping the connection.

Tester

A tool that can automatically run tests on StoneGate engines to check system or network operation and take action based on the results of those tests.

Timeline

A tool in the [Logs View](#) that allows you to select and change the time range for the logs that are displayed.

ToS Flag

A data field in IP packet headers that provides a number representing the type of the service the packet is a part of. The ToS flag is used for [Traffic Prioritization](#) and is also known as [DSCP \(DiffServ Code Point\)](#).

Traffic Handler

The set of [Network Elements](#) used for inbound and outbound traffic management. Includes [NetLinks](#), [Outbound Multi-links](#), and [Server Pools](#).

Traffic Management

The control, definition, and management of how packets or connections should flow through firewalls, routers, network links, VPNs or other gateway objects, based on load balancing, clusters, availability of links and more.

Traffic Prioritization

The process of assigning traffic a priority value, which is used to determine the order in which queued packets are sent forward, overriding the standard first-come-first-served operation of network devices. Used for assuring Quality of Service (QoS) for time-critical connections. Can be used together with [Bandwidth Management](#) or on its own. See also [DSCP \(DiffServ Code Point\)](#) (page 208), [QoS Class](#) (page 220) and [QoS Policy](#) (page 220).

Transparent Access Control Mode

A [Sensor](#) configuration in which the Sensor examines Ethernet traffic according to the [Ethernet Rules](#).

Transparent Proxy

A technique whereby a connection is routed to a proxy server, which then establishes a second connection to the original destination host, but the entire transaction takes place without notifying the user, or requiring the user to perform any additional actions.

Transport Protocol

Any protocol that communicates and functions on the transport layer of the TCP/IP protocol stack. These protocols function above the network layer, and are usually responsible for error correction, quality of service, and other characteristics not handled by the network layer. TCP, UDP, and IPsec are common examples of transport protocols.

Tunneling

A technology that enables one network to send its data through another, perhaps dissimilar, network. Tunneling works by encapsulating, or packaging, a network protocol within packets carried by the second network.

U

Use IPsec VPN Action

A firewall [Action](#) parameter that directs traffic matching to the rule to a VPN. Can be either an [Apply VPN Action](#) or an [Enforce VPN Action](#).

UDP Tracking

Information maintained by the firewall engines to group together UDP requests and replies, handling them as a single virtual connection. See also [Virtual Connection Tracking](#) (page 228).

User

An [Element](#) that defines an end-user in your network. Used for defining [Authentication](#) with or without [Client-to-Gateway VPN](#) access. Confer to [Administrator](#) (page 201).

User Response

Defines additional notification actions for rule matches, such as redirecting access to a forbidden URL to a page on an internal web server instead.

UTM (Unified Threat Management)

A device that combines different types of traffic filtering in one physical appliance. The features offered in a UTM device vary greatly from vendor to vendor. The StoneGate UTM comprises a firewall, deep packet inspection (IDS), and antivirus.

Virtual Adapter

A component of the StoneGate VPN Client, or a third-party VPN client, that allows using a second, [Virtual IP address](#) for [Virtual Private Network \(VPN\)](#) traffic. Shown as a network adapter in the operating system.

Virtual Connection Tracking

A superset of UDP tracking, ICMP tracking, etc. A technology that is used by the firewall engines for connectionless network protocols like UDP and ICMP. The firewall engines keep track of virtual connections by grouping together packets that are related, based on information in the packet headers. See also [Related Connection](#) (page 221).

Virtual Defragmentation

A procedure in which incoming packet fragments are collected. The packet is defragmented for processing by the firewall engine, and refragmented before it is transmitted again. See also [Fragmentation](#) (page 210).

Virtual IP address

A second IP address that is given to a [VPN Client](#) that has a [Virtual Adapter](#) enabled, and that is connecting to a security gateway using [Client-to-Gateway VPN](#). A virtual IP address enables the use of certain services that require the client to have an IP address belonging to a specific address range, while enabling it to retain its primary IP address for maintaining other connections. The Virtual IP address for StoneGate VPN Clients is always assigned by [DHCP \(Dynamic Host Configuration Protocol\)](#).

Virtual Local Area Network (VLAN)

A local area network which is defined through software in a switch or other networking device, rather than by the more traditional hardware division.

Virtual Private Network (VPN)

Refers to a confidential connection that is established through unsecured networks by the means of authentication, encryption, and integrity checking. The two major VPN technologies are [IPsec \(IP Security\)](#), which is better suited when a wide variety of network services and large traffic volumes are involved, and [SSL VPN](#), which is used to provide access to a limited number of services to individual users without client-side device configuration.

VPN Client

Software that can be used to establish a [Virtual Private Network \(VPN\)](#) with a VPN gateway device to securely access remote resources over insecure networks.

VPN Profile

An element that defines the [IPsec \(IP Security\)](#)-related settings for one or more VPNs.

Vulnerability

An IPS element that contains information on a publicly known flaw that affects security of some system. Vulnerabilities are attached to [Situations](#) to provide you more information on what has happened when the Situation matches.

W**Web Filtering**

A feature that compares the URLs that users attempt to open to a list of URLs to prevent users from intentionally or accidentally accessing most web sites that are objectionable or potentially harmful.

Web Portal

Browser-based service that allows users to view logs, [Policy Snapshots](#), and reports.

Whitelisting

The process of exempting specific traffic from being blocked by [Blacklisting](#) or [Web Filtering](#).

INDEX

A

- access rules, 91–102
 - aliases in, 100
 - allowing system communications, 98
 - apply blacklist action in, 97
 - continue action in, 97–99
 - for TLS inspection, 126
 - in IPS template policies, 94–96
 - rematching tunneled packets, 100
 - rule table for, 93–94
 - service in, 96
 - time in, 98
- activating inspection checks, 110
- aliases, 153
 - in access rules, 100
 - system aliases, 154
 - user aliases, system-defined, 154
- allow action, 87, 97
- analyzers, 22, 47–58
 - correlation situations for, 65
 - normal interfaces on, 49, 50
- anomaly detection, 15
- antievation, 15
- apply blacklist action
 - in access rules, 97

B

- benefits of IPS, 23
- blacklisting, 133–138
 - blacklist requests, 135
 - blacklists, 135
 - in access rules, 97
 - whitelisting, 134
- brightcloud, 130

C

- cabling guidelines, 42
- capture interfaces, 48
- CAT rating, 42
- category-based web filtering, 130
- certificate authorities
 - for client protection, 126
 - trusted, 125
- certificates in HTTPS, 124
- certification policy, 75–76
- clustering, 24
 - sensors, 47–58
- command line tools, 141
- comments in rules, 79
- components of the system, 21
- compress situation context, 65
- contact addresses, 53

- contact information, 12
- contexts, 64
- continue action, 79
 - in access rules, 97–99
 - in inspection rules, 111
- correlation situations, 65
- count situation context, 65
- crossover cables, 42
- customer support, 12

D

- deep packet inspection
 - of HTTPS, 123–128
 - options for access rules, 97
 - options for IPv6 access rules, 97
- default system policy
 - access rules in, 94–96
 - ethernet rules in, 86
 - inspection rules in, 109
- deployment
 - in IDS mode, 38
 - in IPS mode, 41
- designing inspection rules, 106
- detection
 - anomaly, 15
 - misuse, 15
- disaster recovery, 40
- discard action
 - in access rules, 97
 - in ethernet rules, 87
 - in IPv6 access rules, 97
- documentation available, 11
- DoS detection situation context, 66
- duplex, 43

E

- eliminating false positives, 110
- encapsulated packets, 100
- engine commands, 142
- ethernet rules, 83–89
 - action in, 87
 - in IPS strict template, 86
 - in IPS template policies, 86
 - logging options for, 87
 - rule table for, 85
 - service in, 87
- evasions, 15
- event compress, 65
- event correlation, 65
- event count, 65
- event group, 65
- event match, 65

event sequence, 66
exception rules, 108

F

false detection results, 16
false positives, 110
file detection situation context, 67
filtering, web content, 130
fingerprint syntax, 163
FTP protocol agent, 118

G

GRE protocol agent, 118
group situation context, 65
guidelines for cabling, 42

H

H.323 protocol agent, 118
hardware requirements, 12
heartbeat, 48
HIDS, 14
high availability, 24
how traffic is inspected, 25
HTTP protocol agent, 118
HTTP URL filter, 131
HTTPS inspection exceptions, 126
HTTPS inspection, *see* TLS inspection
HTTPS protocol agent, 118

I

ICMP protocol agent, 118
IDS
 host-based, 14
 network-based, 14
IDS mode deployment, 38
IEEE 802.1Q VLAN tagging, 49
inherited rules, 74
inline interfaces, 49, 52
inline mode, 14
inspection rules, 103–112
 activating inspection checks in, 110
 continue action in, 111
 design issues for, 106
 exceptions in, 108
 in IPS strict template, 109
 in IPS template policies, 109
 rules and exceptions in, 105
interfaces
 network interface cards, 48, 49
 speed and duplex for, 43
IP in IP tunneling, 100
IPS deployment, 27–43
IPS policies, 71–81
 certification policy, 75–76

installing, 78
packet inspection with, 72–74
policy hierarchy, 72
strict policy, 75–76
system policy, 75–76
types of, 74
user response in, 80
validating, 79

IPS strict template, 75–76
 access rules in, 94–96
 ethernet rules in, 86
 inspection rules in, 109

IPS sub-policies
 creating, 77–78

IPS system template, 75–76
 access rules in, 94–96
 ethernet rules in, 86
 inspection rules, 109

IPS template policies, 74
 access rules in, 94–96
 creating, 76–77
 ethernet rules in, 86
 inspection rules in, 109
 IPS strict template, 75–76
 IPS system template, 75–76

IPS, inline, 14

IPv4 access rules, 91–102

IPv4 Encapsulation protocol agent, 118

IPv4 protocol agent, 118

IPv6 access rules, 91–102
 rematching tunneled packets, 100

IPv6 Encapsulation protocol agent, 119

IPv6 protocol agent, 119

J

jump action, in access rules, 97

L

location, for NATed communications, 53

logging options

 for access rules, 98

 for ethernet rules, 87

 for IPv6 access rules, 98

logical interfaces, 51–52

M

MAC address, for cluster normal interfaces, 50
malware, 130

management center deployment, 34

match situation context, 65

mirror ports, 38

misuse detection, 15

MSRPC protocol agent, 119

N

- NATed addresses, 53
- NDI (node dedicated IP address), see normal interfaces
- netBIOS protocol agent, 119
- network interfaces, 48, 49
- network TAP, 38
- NIC (network interface card), 48, 49
 - speed and duplex for, 43
- NIDS, 14
- non-decrypted domains, 126
- normal interfaces, 49, 50
 - in analyzers, 50
 - in sensor clusters, 50

O

- oracle protocol agent, 119

P

- packet inspection, with IPS policy, 72–74
- passive termination, 106
- phishing, 130
- policy
 - certification, 75–76
 - predefined, 75–76
 - strict, 75–76
 - system, 75–76
- policy snapshots, 79
- positioning sensors, 30
- pre-defined aliases, 154
- protocol agents, 96, 113–121
 - connection handling with, 114
 - FTP, 118
 - GRE, 118
 - H.323, 118
 - HTTP, 118
 - HTTPS, 118
 - ICMP, 118
 - IPv4, 118
 - IPv4 Encapsulation, 118
 - IPv6, 119
 - IPv6 Encapsulation, 119
 - MSRPC, 119
 - netBIOS, 119
 - oracle, 119
 - protocol validation, 114
 - remote shell, 119
 - services in firewall, 120
 - SMTP, 120
 - SSH, 120
 - SunRPC, 120
 - TCP proxy, 120
 - TFTP, 120
- protocol validation, 15
- protocol-specific situation context, 66

R

- refuse action, in access rules, 97
 - regular expression syntax, 163
 - rematching tunneled packets, 100
 - remote shell protocol agent, 119
 - requirements for hardware, 12
 - reset interfaces, 50, 52
 - response options
 - for access rules, 97
 - routing, 59–60
 - any network element in, 60
 - default route in, 60
 - for analyzers, 60
 - for sensors, 60
 - rules
 - comments in, 79
 - continue action, 111
 - continue action in, 79
 - inheritance of, 74
 - protocol agents in, 96
 - validating, 79
 - rules tree, 105
- ## S
- scan detection situation context, 66
 - secure sockets layer, 29
 - security considerations for TLS inspection, 127
 - sensor cluster
 - communications in, 48
 - interfaces in, 48
 - sensor-analyzers, 22
 - sensors, 22
 - capture interfaces on, 48
 - clustering, 47–58
 - inline interfaces on, 49, 52
 - logical interfaces on, 51–52
 - normal interfaces on, 49, 50
 - reset interfaces on, 50, 52
 - transparent access control mode for, 26
 - sequence situation context, 66
 - server protection credentials, 125
 - service (access rule field), 96
 - service (ethernet rule field), 87
 - services in firewall protocol agent, 120
 - situation contexts
 - for correlation
 - event compress, 65
 - event count, 65
 - event group, 65
 - event match, 65
 - event sequence, 66
 - for DoS detection, 66
 - for file detection, 67
 - for scan detection, 66
 - protocol-specific, 66

- system, 67
- situations, 63–70
 - context definitions in, 65
 - contexts in, 64
 - severity of, 67
 - tags in, 64
 - type in, 64
 - vulnerabilities in, 64
- SMTP protocol agent, 120
- SPAN ports, 38
- speed and duplex, 43
- SSH protocol agent, 120
- SSL, 29
- straight cables, 42
- strict policy, 75–76
 - ethernet rules in, 86
 - inspection rules in, 109
- sub-policies, 74
- SunRPC protocol agent, 120
- support services, 12
- system aliases, 154
- system components, 21
 - analyzer, 22
 - sensor, 22
 - sensor-analyzer, 22
- system policy, 75–76
 - access rules in, 94–96
 - ethernet rules in, 86
 - inspection rules in, 109
- system requirements, 12
- system situation context, 67
- system-defined user aliases, 154

T

- tags, 64
- TAP devices, 38
- TCP inspection modes
 - normal, 54
 - strict, 54
- TCP proxy protocol agent, 120
- technical support, 12
- template policies, 74
- terminate (passive), 106
- TFTP protocol agent, 120
- time, in access rules, 98
- TLS, 29
- TLS inspection, 123–128
 - access rules for, 126
 - certificates in, 124
 - client protection certificate authorities for, 126
 - client protection in, 124
 - in sensor properties, 126
 - inspection exceptions for, 126
 - non-decrypted domains in, 126
 - security considerations for, 127

- server protection credentials for, 125
- server protection in, 124
- traffic inspection process, 25
- traffic normalization, 15
- transparent access control, 26
- transport layer security, 29
- trusted certificate authorities, 125
- tuning inspection, 106
- tunneled packets, rematching, 100
- typographical conventions, 10

U

- URL filtering, 130
- user aliases, system-defined, 154
- user responses, 80, 97, 110

V

- VLAN (virtual local area network), 49
- VPN (virtual private network)
 - user aliases, 154
- vulnerabilities, 15, 64
- vulnerability detection, 15

W

- web filtering, 130
- whitelisting, 134

StoneGate Guides

Administrator's Guides - step-by-step instructions for configuring and managing the system.

Installation Guides - step-by-step instructions for installing and upgrading the system.

Reference Guides - system and feature descriptions with overviews to configuration tasks.

User's Guides - step-by-step instructions for end-users.

For more documentation, visit
www.stonesoft.com/support/

Stonesoft Corporation

Itälahdenkatu 22 A
FI-00210 Helsinki
Finland

Tel. +358 9 476 711
Fax +358 9 4767 1349

STONESOFT

Secure Information Flow

Stonesoft Inc.

1050 Crown Pointe Parkway
Suite 900
Atlanta, GA 30338
USA

Tel. +1 770 668 1125
Fax +1 770 668 1131