

# **McAfee NGFW Reference Guide for Firewall/VPN Role 5.7**

NGFW Engine in the Firewall/VPN Role



## Legal Information

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 McAfee website:

<http://www.mcafee.com/us/about/legal/license-agreements.aspx>

# TABLE OF CONTENTS

## INTRODUCTION

---

### CHAPTER 1

<b>Using SMC Documentation.</b>	13
How to Use This Guide	14
Documentation Available	15
Product Documentation	15
Support Documentation	16
System Requirements	16
Supported Features	16
Contact Information	16

### CHAPTER 2

<b>Introduction to Firewalls</b>	17
The Role of the Firewall	18
Firewall Technologies	19
Packet Filtering	19
Proxy Firewalls	19
Stateful Inspection	20
Multi-Layer Inspection	20
Additional Firewall Features	21
Authentication	21
Deep Packet Inspection and Unified Threat Management	21
Integration With External Content Inspection	22
Load Balancing and Traffic Management	22
Logging and Reporting	23
Network Address Translation (NAT)	23
VPNs	23
Firewall Weaknesses	24
Complexity of Administration	24
Single Point of Failure	24
Worms, Viruses, and Targeted Attacks	24

### CHAPTER 3

<b>Introduction to McAfee NGFW in the Firewall/VPN Role</b>	25
The McAfee NGFW Solution	26
SMC Components	27
Firewall/VPN Engines	28
Main Benefits of McAfee Firewall/VPN	29
Advanced Traffic Inspection	29
Built-in Clustering for Load Balancing and High Availability	29
Multi-Link Technology	30

Built-in Inbound Traffic Management	31
QoS and Bandwidth Management	31
Integration with IPS and Layer 2 Firewalls	31
Clustered Multi-Link VPNs	31

### CHAPTER 4

<b>NGFW Deployment in the Firewall/VPN Role</b>	33
Deployment Overview	34
Supported Platforms	34
General Deployment Guidelines	35
Positioning Firewalls	36
External to Internal Network Boundary	37
Internal Network Boundaries	38
DMZ Network Boundaries	39

## INTERFACES AND ROUTING

---

### CHAPTER 5

<b>Single Firewall Configuration</b>	43
Overview to Single Firewall Configuration	44
Configuration of Single Firewalls	44
Dynamic Firewall Interface Addresses	44
Internal DHCP Server	45
Security Strength of Management Connections	45
Configuration Workflow	45
Task 1: Create Single Firewall Elements	45
Task 2: Define Physical Interfaces	45
Task 3: Define VLAN Interfaces	46
Task 4: Define Tunnel Interfaces	46
Task 5: Define an ADSL Interface	46
Task 6: Define a Wireless Interface	46
Task 7: Define IP Addresses	46
Task 8: Define Loopback IP Addresses	47
Task 9: Define Modem Interfaces	47
Task 10: Install the Firewall Engine	47
Task 11: Install a Firewall Policy	48
Example of a Single Firewall Deployment	48
Setting up a Single Firewall	48
Adding a New Interface to an Existing Configuration	49

**CHAPTER 6**  
**Firewall Cluster Configuration** . . . . . 51

Overview to Firewall Cluster Configuration. . . . . 52

Benefits of Clustering . . . . . 52

Communication Between the Nodes. . . . . 52

Hardware . . . . . 53

Security Strength of Management Connections 53

Configuration of Firewall Clusters. . . . . 53

Load Balancing. . . . . 53

Standby Operation . . . . . 54

Network Interfaces and IP Addresses. . . . . 54

Clustering Modes . . . . . 55

How Packet Dispatch Works . . . . . 56

Configuration Workflow . . . . . 57

Task 1: Create a Firewall Cluster Element. . . . . 57

Task 2: Create Physical Interfaces. . . . . 57

Task 3: Define VLAN Interfaces . . . . . 57

Task 4: Define Tunnel Interfaces . . . . . 58

Task 5: Configure Physical or VLAN Interfaces . . . . . 58

Task 6: Define Loopback IP Addresses . . . . . 59

Task 7: Install the Firewall Engines . . . . . 59

Task 8: Install a Firewall Policy . . . . . 59

Using a Firewall Cluster. . . . . 60

Internal DHCP Server . . . . . 60

Node State Synchronization. . . . . 60

Security Level for State Synchronization . . . . . 61

Manual Load Balancing. . . . . 61

Examples of Firewall Cluster Deployment . . . . . 62

Setting up a Firewall Cluster . . . . . 62

Adding a Node to a Firewall Cluster . . . . . 63

**CHAPTER 7**  
**Master Engine and Virtual Firewall Configuration** . . . . . 65

Overview to Master Engine and Virtual Firewall Configuration . . . . . 66

Configuration of Master Engines and Virtual Firewalls . . . . . 66

Configuration Workflow . . . . . 67

Task 1: Create a Master Engine Element. . . . . 67

Task 2: Create Virtual Resource Element(s). . . . . 67

Task 3: Configure Master Engine Interfaces . . . . . 67

Task 4: Create a Virtual Firewall Element. . . . . 67

Task 5: Configure Virtual Firewall

Interfaces . . . . . 68

Task 6: Install a Firewall Policy . . . . . 68

Using Master Engines and Virtual Firewalls . . . 69

Moving a Virtual Firewall to a Different Master Engine . . . . . 69

Using Master Engines and Virtual Firewalls With Domains . . . . . 69

Example of Master Engine and Virtual Firewall Deployment. . . . . 70

Deploying Virtual Firewalls for MSSP Customers . . . . . 70

**CHAPTER 8**  
**Routing and Antispoofing** . . . . . 73

Overview to Routing and Antispoofing . . . . . 74

Configuration of Routing and Antispoofing . . . 74

Reading the Routing and Antispoofing Trees. . 74

Multi-Link Routing . . . . . 76

Default Elements . . . . . 77

Configuration Workflow. . . . . 77

Task 1: Add Router or NetLink . . . . . 77

Task 2: Add Network(s) . . . . . 77

Task 3: Refresh Firewall Policy . . . . . 77

Using Routing and Antispoofing. . . . . 78

Policy Routing . . . . . 78

Multicast Routing. . . . . 78

Modifying Antispoofing . . . . . 78

Monitoring Routing. . . . . 78

Examples of Routing . . . . . 79

Routing Traffic with Two Interfaces. . . . . 79

Routing Internet Traffic with Multi-Link . . . . . 79

Routing Traffic to Networks That Use Same Address Space . . . . . 80

**ACCESS CONTROL POLICIES**

---

**CHAPTER 9**  
**Firewall Policies** . . . . . 83

Overview to Firewall Policies . . . . . 84

Policy Hierarchy . . . . . 84

How the Engine Examines Traffic. . . . . 84

Configuration of Policy Elements . . . . . 87

Default Elements . . . . . 89

Configuration Workflow. . . . . 91

Task 1: Create a Template Policy . . . . . 91

Task 2: Create a Policy. . . . . 92

Task 3: Create a Firewall Sub-Policy . . . . . 92

Task 4: Install the Policy . . . . . 93

Using Policy Elements and Rules . . . . .	94	<b>CHAPTER 11</b>	
Validating Policies. . . . .	94	<b>Inspection Policies . . . . .</b>	<b>119</b>
User Responses. . . . .	94	Overview to Inspection Policies . . . . .	120
Connection Tracking vs. Connectionless Packet Inspection . . . . .	95	Configuration of Inspection Policies . . . . .	121
Policy Snapshots . . . . .	97	Verifying and Tuning Inspection . . . . .	122
Continue Rules. . . . .	97	Considerations for Designing Inspection Policies . . . . .	123
Adding Comments to Rules . . . . .	97	Exception Rule Cells. . . . .	124
Naming Rules. . . . .	98	Default Elements . . . . .	125
Examples of Policy Element Use . . . . .	98	Configuration Workflow . . . . .	126
Protecting Essential Communications . . . . .	98	Task 1: Create an Inspection Policy. . . . .	126
Improving Readability and Performance . . . . .	98	Task 2: Activate Deep Inspection in Firewall Policies . . . . .	126
Restricting Administrator Editing Rights . . . . .	99	Task 3: Activate the Relevant Inspection Checks . . . . .	126
<b>CHAPTER 10</b>		Task 4: Define the Exceptions . . . . .	127
<b>Access Rules . . . . .</b>	<b>101</b>	Task 5: Eliminate False Positives . . . . .	127
Overview to Access Rules . . . . .	102	Task 6: Add Custom Inspection Checks . . . . .	127
Configuration of Access Rules . . . . .	103	Using Inspection Policies . . . . .	128
Considerations for Designing Access Rules . . . . .	105	Setting Default Options for Several Inspection Exceptions . . . . .	128
Default Elements . . . . .	105	Importing Snort Rules Libraries. . . . .	128
Configuration Workflow . . . . .	106	Example of Inspection Rules . . . . .	129
Task 1: Define the Source and Destination . . . . .	106	Eliminating a False Positive. . . . .	129
Task 2: Define the Service . . . . .	106	<b>CHAPTER 12</b>	
Task 3: Select the Action and Action Options . . . . .	107	<b>Network Address Translation (NAT) . . . . .</b>	<b>131</b>
Task 4: Select Logging Options . . . . .	109	Overview of NAT . . . . .	132
Task 5: Add User Authentication Requirements . . . . .	109	Element-Based NAT . . . . .	132
Task 6: Restrict the Time When the Rule Is Enforced. . . . .	110	Static Source Translation . . . . .	133
Task 7: Restrict the Rule Match Based on Source VPN. . . . .	110	Dynamic Source Translation . . . . .	134
Using Access Rules . . . . .	110	Static Destination Translation . . . . .	134
Allowing System Communications . . . . .	110	Destination Port Translation . . . . .	135
Configuring Default Settings for Several Rules . . . . .	111	Configuration of NAT. . . . .	135
Using Continue Rules to Set Logging Options . . . . .	112	Considerations for Designing NAT Rules . . . . .	137
Using Continue Rules to Set the Protocol. . . . .	112	Default Elements . . . . .	137
Using Aliases in Access Rules. . . . .	113	Configuration Workflow . . . . .	137
Creating User-Specific Access Rules . . . . .	113	Task 1: Define Source, Destination, and Service . . . . .	137
Using Domain Names in Access Rules . . . . .	114	Task 2: Define Address Translation. . . . .	138
Interface Matching in Access Rules . . . . .	114	Task 3: Define the Firewall(s) that Apply the Rule . . . . .	138
Examples of Access Rules. . . . .	115	Task 4: Check Other Configurations . . . . .	138
Example of Rule Order . . . . .	115	Using NAT and NAT Rules . . . . .	139
Example of Continue Rules . . . . .	117	NAT and System Communications . . . . .	139
Example of User-Specific Rules . . . . .	118	Example of a Situation Where a Contact Address is Needed . . . . .	140
		Contact Addresses and Locations . . . . .	141
		Outbound Load-Balancing NAT. . . . .	141

Proxy ARP and NAT . . . . .	142
Protocols and NAT . . . . .	142
Examples of NAT. . . . .	142
Dynamic Source Address Translation . . . . .	142
Static Address Translation. . . . .	143
NAT with Hosts in the Same Network . . . . .	144
<b>CHAPTER 13</b>	
<b>Protocol Agents. . . . .</b>	<b>145</b>
Overview to Protocol Agents . . . . .	146
Connection Handling. . . . .	146
Protocol Validation . . . . .	146
NAT in Application Data. . . . .	147
Configuration of Protocol Agents . . . . .	147
Configuration Workflow . . . . .	147
Task 1: Create a Custom Service with a Protocol Agent. . . . .	147
Task 2: Set Parameters for the Protocol Agent . . . . .	148
Task 3: Insert the Service in Access Rules . . . . .	148
Using Protocol Agents. . . . .	149
FTP Agent . . . . .	149
GRE Agent . . . . .	149
GTP Agent . . . . .	149
H323 Agent . . . . .	150
HTTP Agent . . . . .	150
HTTPS Agent . . . . .	150
MGCP Agent. . . . .	150
MSRPC Agent. . . . .	150
NetBIOS Agent . . . . .	151
Oracle Agent . . . . .	151
RTSP Agent . . . . .	151
SCCP Agent . . . . .	151
Services in Firewall Agent . . . . .	151
Shell Agent . . . . .	151
SIP Agent. . . . .	152
SMTP Agent. . . . .	152
SSH Agent . . . . .	152
SunRPC Agent . . . . .	152
TCP Proxy Agent . . . . .	153
TFTP Agent. . . . .	153
Examples of Protocol Agent Use . . . . .	154
Preventing Active Mode FTP. . . . .	154
Logging URLs Accessed by Internal Users . . . . .	154

<b>CHAPTER 14</b>	
<b>TLS Inspection . . . . .</b>	<b>155</b>
Overview to TLS Inspection. . . . .	156
Configuration of TLS Inspection. . . . .	157
Default Elements . . . . .	158
Configuration Workflow. . . . .	158
Task 1: Create Server Credentials Elements . . . . .	158
Task 2: Create Client Protection Certificate Authority Elements. . . . .	158
Task 3: Exclude Traffic from Decryption and Inspection. . . . .	159
Task 4: Activate TLS Inspection . . . . .	159
Using TLS Inspection . . . . .	160
Security Considerations . . . . .	160
Virus Scanning of Decrypted TLS Traffic. . . . .	160
URL Filtering Decrypted TLS Traffic . . . . .	160
Examples of TLS Inspection . . . . .	161
Server Protection . . . . .	161
Client Protection . . . . .	161
<b>CHAPTER 15</b>	
<b>URL Filtering . . . . .</b>	<b>163</b>
Overview to URL Filtering . . . . .	164
Configuration of URL Filtering . . . . .	164
Default Elements . . . . .	165
Configuration Workflow. . . . .	165
Task 1: Prepare the Firewall . . . . .	165
Task 2: Create User Response Messages . . . . .	165
Task 3: Blacklist/Whitelist Individual URLs . . . . .	165
Task 4: Configure URL Filtering Rules . . . . .	166
Examples of URL Filtering. . . . .	166
Allowing a Blocked URL . . . . .	166
<b>CHAPTER 16</b>	
<b>Spam Filtering . . . . .</b>	<b>167</b>
Overview to Spam Filtering . . . . .	168
Configuring Spam Filtering . . . . .	168
Configuration Workflow. . . . .	168
Task 1: Define Spam Filtering for a Firewall . . . . .	168
Task 2: Select Traffic for Inspection with Access Rules . . . . .	168
Task 3: Select Traffic Not to Be Filtered . . . . .	168
Using Spam Filtering . . . . .	169
Anti-Spoofing and Anti-Relay Protection . . . . .	169
Handling E-mail Address Forgery . . . . .	169
Spam Filter Sensitivity Settings. . . . .	169
Spam Filtering Rules . . . . .	170

DNS-Based Blackhole Lists . . . . .	170
<b>CHAPTER 17</b>	
<b>Virus Scanning</b> . . . . .	171
Overview to Virus Scanning . . . . .	172
Configuration of Virus Scanning . . . . .	172
Configuration Workflow . . . . .	172
Task 1: Activate the Anti-Virus Feature for a Firewall . . . . .	172
Task 2: Select Traffic for Inspection with Access Rules . . . . .	172
Task 3: Define the Content Not to Be Scanned . . . . .	172
Using Virus Scanning . . . . .	173
Integrated Scanning vs. Content Inspection Server . . . . .	173
Limitations of Virus Scanning on Clusters . . .	173
<b>CHAPTER 18</b>	
<b>External Content Inspection</b> . . . . .	175
Overview to Content Inspection . . . . .	176
Configuration of Content Inspection . . . . .	177
Default Elements . . . . .	178
Configuration Workflow . . . . .	178
Task 1: Create a CIS Server Element . . . . .	178
Task 2: Create a Custom Service for Content Inspection Server Redirection . . . . .	178
Task 3: Define Access Rules for Redirection . . . . .	178
Task 4: Configure NAT Rules for Content Inspection Server Redirection . . . . .	179
Using Content Inspection . . . . .	179
Example of Content Inspection . . . . .	180
Inspecting Internal User's Web Browsing and File Transfers . . . . .	180
<b>CHAPTER 19</b>	
<b>Situations</b> . . . . .	183
Overview to Situations . . . . .	184
Configuration of Situations . . . . .	184
Situation Contexts . . . . .	185
Correlation Contexts . . . . .	185
Anti-Virus Contexts . . . . .	186
DoS Detection Contexts . . . . .	186
Scan Detection Contexts . . . . .	186
Protocol-Specific Contexts . . . . .	186
File Contexts . . . . .	187
System Contexts . . . . .	187
Default Elements . . . . .	187
Configuration Workflow . . . . .	187
Task 1: Create a Situation Element . . . . .	187
Task 2: Add a Context for the	

Situation . . . . .	188
Task 3: Associate Tags and/or Situation Types with the Situation . . . . .	188
Task 4: Associate the Situation with a Vulnerability . . . . .	188
Using Situations . . . . .	189
Example of Custom Situations . . . . .	189
Detecting the Use of Forbidden Software . . . .	189
<b>CHAPTER 20</b>	
<b>Applications</b> . . . . .	191
Overview to Applications . . . . .	192
Configuration of Applications . . . . .	192
Default Elements . . . . .	192
Configuration Workflow . . . . .	193
Task 1: Define TLS Matches . . . . .	193
Task 2: Create Access Rules . . . . .	193
Examples of Applications . . . . .	194
Blocking Application Use . . . . .	194
<b>CHAPTER 21</b>	
<b>Blacklisting</b> . . . . .	195
Overview to Blacklisting . . . . .	196
Risks of Blacklisting . . . . .	196
Configuration of Blacklisting . . . . .	197
Configuration Workflow . . . . .	198
Task 1: Define Blacklisting in Access Rules . . . . .	198
Task 2: Define Exceptions in the Inspection Policy . . . . .	198
Using Blacklisting . . . . .	198
Manual Blacklisting . . . . .	198
Monitoring Blacklisting . . . . .	198
Whitelisting . . . . .	198

## USERS AND AUTHENTICATION

---

<b>CHAPTER 22</b>	
<b>Directory Servers</b> . . . . .	201
Overview to Directory Servers . . . . .	202
Configuration of Directory Servers . . . . .	202
Internal User Database . . . . .	202
Authentication Server User Linking . . . . .	202
External Directory Server Integration . . . . .	203
User Agents for Active Directory . . . . .	203
Configuration Workflow . . . . .	204
Task 1: Create an LDAP Server or an	

Active Directory Server Element . . . . .	204
Task 2: Add an LDAP Domain . . . . .	204
Task 3: Add Users and User Groups or Link Users . . . . .	205
Task 4: Install and Configure the User Agent . . . . .	205
Examples of Directory Servers . . . . .	206
Using the Internal User Database . . . . .	206
Integrating a Microsoft Active Directory Server . . . . .	206

## CHAPTER 23

<b>User Authentication on the Firewall . . . . .</b>	<b>207</b>
Overview to User Authentication on the Firewall . . . . .	208
Configuration of User Authentication on the Firewall . . . . .	209
Default Elements . . . . .	209
Configuration Workflow . . . . .	209
Task 1: Define User Authentication in IPv4 Access Rules . . . . .	210
Task 2: Configure User Authentication Interfaces . . . . .	210
Example of User Authentication on the Firewall . . . . .	211
Authenticating VPN Client Users . . . . .	211

## CHAPTER 24

<b>External User Authentication . . . . .</b>	<b>213</b>
Overview to External User Authentication . . . . .	214
Configuration of External User Authentication . . . . .	215
Directory Servers for External User Authentication . . . . .	216
RADIUS Authentication . . . . .	216
TACACS+ Authentication . . . . .	217
Authentication Methods . . . . .	217
Federated Authentication . . . . .	218
Default Elements . . . . .	218
Configuration Workflow . . . . .	218
Task 1: Define Servers . . . . .	218
Task 2: Associate Authentication Methods with Servers . . . . .	219
Task 3: Define User Authentication in IPv4 Access Rules . . . . .	219
Task 4: Configure User Authentication Interfaces . . . . .	220
Examples of External User Authentication . . . . .	220
Integrating a Microsoft Active Directory Server . . . . .	220
Using SecurID Authentication with Stonesoft IPsec VPN Clients . . . . .	221

# TRAFFIC MANAGEMENT

## CHAPTER 25

<b>Outbound Traffic Management . . . . .</b>	<b>225</b>
Overview of Outbound Traffic Management . . . . .	226
Configuration of Multi-Link . . . . .	226
Load-Balancing Methods . . . . .	227
Standby NetLinks for High Availability . . . . .	227
Link Status Probing . . . . .	228
Configuration Workflow . . . . .	228
Task 1: Create NetLink Elements . . . . .	228
Task 2: Configure Routing for NetLinks . . . . .	228
Task 3: Combine NetLinks into Outbound Multi-Link Elements . . . . .	229
Task 4: Create NAT Rules for Outbound Traffic . . . . .	229
Using Multi-Link . . . . .	230
Multi-Link with a Single Firewall . . . . .	230
Multi-Link with a Firewall Cluster . . . . .	231
Using Multiple Outbound Multi-Link Elements . . . . .	231
Examples of Multi-Link . . . . .	232
Preparing for ISP Breakdown . . . . .	232
Excluding a NetLink from Handling a QoS Class of Traffic . . . . .	232
Balancing Traffic According to Link Capacity . . . . .	233
Balancing Traffic between Internet Connections . . . . .	233

## CHAPTER 26

<b>Inbound Traffic Management . . . . .</b>	<b>235</b>
Overview to Server Pool Configuration . . . . .	236
Configuration of Server Pools . . . . .	237
Multi-Link for Server Pools . . . . .	237
Default Elements . . . . .	238
Configuration Workflow . . . . .	238
Task 1: Define Hosts . . . . .	238
Task 2: Combine Hosts into a Server Pool Element . . . . .	238
Task 3: Configure the External DNS Server . . . . .	238
Task 4: Create an Inbound Load- Balancing Rule . . . . .	238
Task 5: Set up Server Pool Monitoring Agents . . . . .	239
Using Server Pools . . . . .	239
Dynamic DNS (DDNS) Updates . . . . .	239
Using Server Pool Monitoring Agents . . . . .	240
Examples of Server Pools . . . . .	242
Load Balancing for Web Servers . . . . .	242



Setting up Multi-Link and Dynamic DNS Updates . . . . .	243
<b>CHAPTER 27</b>	
<b>Bandwidth Management and Traffic Prioritization . . . . .</b>	<b>245</b>
Overview to Bandwidth Management and Traffic Prioritization . . . . .	246
Bandwidth Management . . . . .	246
Traffic Prioritization . . . . .	246
Effects of Bandwidth Management and Prioritization . . . . .	247
Configuration of Limits, Guarantees, and Priorities for Traffic . . . . .	247
Default Elements . . . . .	248
Configuration Workflow . . . . .	249
Task 1: Define QoS Classes . . . . .	249
Task 2: Define QoS Policies . . . . .	249
Task 3: Assign QoS Classes to Traffic . . . . .	251
Task 4: Define QoS for Interfaces and VPNs . . . . .	251
Using Bandwidth Management and Traffic Prioritization . . . . .	252
Designing QoS Policies . . . . .	253
Communicating DSCP Markers . . . . .	254
Managing Bandwidth of Incoming Traffic . . . . .	255
Collecting QoS Class-Based Statistics . . . . .	255
Examples of Bandwidth Management and Traffic Prioritization . . . . .	256
Ensuring Quality of Important Communications . . . . .	256
Preparing for ISP Breakdown . . . . .	257
Limiting the Total Bandwidth Required . . . . .	258

## VIRTUAL PRIVATE NETWORKS

<b>CHAPTER 28</b>	
<b>Overview to VPNs . . . . .</b>	<b>261</b>
Introduction to VPNs . . . . .	262
IPsec VPNs . . . . .	262
Tunnels . . . . .	262
Security Associations (SA) . . . . .	263
Internet Key Exchange (IKE) . . . . .	263
Perfect Forward Secrecy (PFS) . . . . .	264
Authentication Header (AH) and Encapsulating Security Payload (ESP) . . . . .	264
Authentication . . . . .	265
Tunnel and Transport Modes . . . . .	265
IPsec VPN Topologies for Policy-Based VPNs . . . . .	266

<b>CHAPTER 29</b>	
<b>Policy-Based VPN Configuration . . . . .</b>	<b>269</b>
Overview to Policy-Based VPN Configuration . . . . .	270
Configuration of Policy-Based VPNs . . . . .	271
Default Elements . . . . .	272
Configuration Workflow . . . . .	273
Task 1: Define the Gateway Settings . . . . .	273
Task 2: Define the Gateway Profile . . . . .	273
Task 3: Define the Gateways . . . . .	273
Task 4: Define the Sites . . . . .	274
Task 5: Create Certificates . . . . .	274
Task 6: Define the VPN Profile . . . . .	275
Task 7: Define the VPN Element . . . . .	275
Task 8: Modify the Firewall Policy . . . . .	276
Task 9: Configure VPN Clients and External Gateway Devices . . . . .	277
Using VPNs . . . . .	278
VPN Logging . . . . .	278
Using a Dynamic IP Address for a VPN End-Point . . . . .	278
Using a NAT Address for a VPN End-Point . . . . .	279
Supported Authentication and Encryption Methods . . . . .	279
FIPS Mode . . . . .	279
GOST-Compliant Systems . . . . .	280
Message Digest Algorithms . . . . .	280
Authentication Methods . . . . .	281
Encryption Algorithms . . . . .	281
Using Pre-Shared Key Authentication . . . . .	283
Using Certificate Authentication . . . . .	283
Validity of Certificates . . . . .	284
Internal VPN Certificate Authorities . . . . .	285
External Certificate Authorities . . . . .	286
Configuring Policy-Based VPNs with External Gateway Devices . . . . .	286
Clustering and Policy-Based VPNs . . . . .	287
Multi-Link and Policy-Based VPNs . . . . .	287
Providing Encryption for the Route-Based VPN in Tunnel Mode . . . . .	289
Examples of Policy-Based VPN Configurations . . . . .	289
Creating a Policy-Based VPN Between Three Offices . . . . .	289
Creating a Policy-Based VPN for Mobile Users . . . . .	291
Creating a Policy-Based VPN That Requires NAT . . . . .	292

**CHAPTER 30**  
**Route-Based VPN Configuration** . . . . . 295

Overview to Route-Based VPN Configuration . . . 296

Configuration of the Route-Based VPN . . . . . 296

Default Elements . . . . . 297

Configuration Workflow . . . . . 298

Task 1: Define Tunnel Interfaces . . . . . 298

Task 2: Configure Routing and Antispoofing for Tunnel Interfaces . . . . . 298

Task 3: Define the Gateways . . . . . 298

Task 4: Define the VPN Profile . . . . . 299

Task 5: Define Route-Based VPN Tunnels . . . 299

Task 6: Add Access Rules to Allow the Traffic . . . . . 300

Task 7: Refresh Firewall Policy . . . . . 300

Using the Route-Based VPN . . . . . 301

Configuring the Route-Based VPN with External Gateway Devices . . . . . 301

Using the Route-Based VPN in Tunnel Mode . . 301

Using the Route-Based VPN with Dynamic Routing . . . . . 302

Examples of Route-Based VPN Configurations . . 302

Protecting Dynamic Routing Communications . 302

Configuring a Route-Based VPN with an External Gateway . . . . . 303

**APPENDICES**

---

**APPENDIX A**  
**Command Line Tools** . . . . . 307

Security Management Center Commands . . . . 308

NGFW Engine Commands . . . . . 319

Server Pool Monitoring Agent Commands . . . . 327

**APPENDIX B**  
**Default Communication Ports** . . . . . 329

Security Management Center Ports . . . . . 330

Security Engine Ports . . . . . 333

**APPENDIX C**  
**Predefined Aliases** . . . . . 337

Predefined User Aliases . . . . . 338

System Aliases . . . . . 338

**APPENDIX D**  
**Situation Context Parameters** . . . . . 341

Correlation Context Parameters . . . . . 342

Regular Expression Parameter . . . . . 344

Other Context Parameters . . . . . 344

**APPENDIX E**  
**Regular Expression Syntax** . . . . . 345

SMC Regular Expression Syntax . . . . . 346

Special Character Sequences . . . . . 348

Pattern-Matching Modifiers . . . . . 349

Variable Expression Evaluation . . . . . 351

Stream Operations . . . . . 354

System Variables . . . . . 355

Independent Subexpressions . . . . . 356

Parallel Matching Groups . . . . . 357

Tips for Working With Regular Expressions . . . 357

**APPENDIX F**  
**Schema Updates for External LDAP Servers** . . . 359

**APPENDIX G**  
**SNMP Traps and MIBs** . . . . . 361

**APPENDIX H**  
**Multicasting** . . . . . 377

The General Features of Multicasting . . . . . 378

Multicasting vs. Unicast . . . . . 378

Multicasting vs. Broadcasting . . . . . 378

IP Multicasting Overview . . . . . 378

Multicasting Applications . . . . . 379

Internet Group Management Protocol . . . . . 379

Membership Messages . . . . . 379

Ethernet Multicasting . . . . . 380

Multicasting and McAfee Firewalls . . . . . 380

Unicast MAC . . . . . 381

Multicast MAC . . . . . 382

Multicast MAC with IGMP . . . . . 383

**Glossary** . . . . . 385

**Index** . . . . . 415

# INTRODUCTION

---

## **In this section:**

**Using SMC Documentation - 13**

**Introduction to Firewalls - 17**

**Introduction to McAfee NGFW in the Firewall/VPN Role - 25**

**NGFW Deployment in the Firewall/VPN Role - 33**



## CHAPTER 1

# USING SMC DOCUMENTATION

This chapter describes how to use this guide and related documentation. It also provides directions for obtaining technical support and giving feedback about the documentation.

The following sections are included:

- ▶ [How to Use This Guide](#) (page 14)
- ▶ [Documentation Available](#) (page 15)
- ▶ [Contact Information](#) (page 16)

# How to Use This Guide

The *McAfee NGFW Reference Guide for Firewall/VPN Role* provides information that helps administrators of McAfee® NGFW in the Firewall/VPN role to understand the system and its features. It provides high-level descriptions and examples of the configuration workflows.

This guide is divided into several sections. The chapters in the first section provide a general introduction to McAfee NGFW in the Firewall/VPN role. The sections that follow each include 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 15).

## 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 <b>bold-face</b> .
References, terms	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 <code>monospaced</code> .
User input	User input on screen is in <code>monospaced bold-face</code> .
Command parameters	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

Stonesoft technical documentation is divided into two main categories: [Product Documentation](#) and [Support Documentation](#) (page 16). Each SMC 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 the SMC comprehensively. Demonstrates the general workflow and provides example scenarios for each feature area. Available as separate guides for McAfee® Security Management Center and McAfee Firewall/VPN, and as a combined guide for McAfee IPS and McAfee Layer 2 Firewall.
Installation Guide	Instructions for planning, installing, and upgrading the SMC. Available as separate guides for McAfee Security Management Center and McAfee Firewall/VPN, and as a combined guide for McAfee IPS and McAfee Layer 2 Firewall.
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 Management Client and the Web Portal. An HTML-based system is available in the 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 McAfee Firewall/VPN, McAfee IPS, and McAfee Layer 2 Firewall, and as separate guides for the SSL VPN and the IPsec VPN Client.
User's Guide	Instructions for end-users. Available for the IPsec VPN Client and the Web Portal.
Appliance Installation Guide	Instructions for physically installing and maintaining McAfee NGFW appliances (rack mounting, cabling, etc.). Available for all McAfee NGFW appliances.

PDF guides are available at [https://www.stonesoft.com/en/customer\\_care/documentation/current/](https://www.stonesoft.com/en/customer_care/documentation/current/). The *McAfee SMC Administrator's Guide*, and the *Reference Guides* and *Installation Guides* for McAfee Security Management Center, McAfee Firewall/VPN, McAfee IPS, and McAfee Layer 2 Firewall are also available as PDFs on the Security Management Center DVD.

## Support Documentation

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

The latest technical documentation is available at <http://www.stonesoft.com/support/>.

## System Requirements

The certified platforms for running the McAfee Next Generation Firewall from Intel Security (NGFW) can be found at the product pages at [http://www.stonesoft.com/en/products/firewall\\_vpn/](http://www.stonesoft.com/en/products/firewall_vpn/).

The hardware and software requirements for the SMC and version-specific details for all software products can be found in the *Release Notes* available at [http://www.stonesoft.com/en/customer\\_care/kb/](http://www.stonesoft.com/en/customer_care/kb/).

## Supported Features

Not all features are supported on all platforms. See the [Appliance Software Support Table](#) for more information.

## Contact Information

---

For general information about SMC products, visit our web site at <http://www.mcafee.com/>.



## CHAPTER 2

# INTRODUCTION TO FIREWALLS

This chapter introduces and discusses the underlying security principles of firewalls in general. In this chapter we will discuss what firewalls are, which different types of firewalls there 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 the Firewall](#) (page 18)
- ▶ [Firewall Technologies](#) (page 19)
- ▶ [Additional Firewall Features](#) (page 21)
- ▶ [Firewall Weaknesses](#) (page 24)

# The Role of the Firewall

---

Firewalls are the primary tool for perimeter access control between networks with different security levels. Firewalls control the traffic between networks and deny access that does not look like acceptable business use as defined by the administrators.

The generally accepted principle of access control is *whatever is not expressly permitted is denied*. The most secure network is achieved when nobody and nothing is permitted entry to the protected network. In most cases, such a network is naturally too limited, so a firewall must be introduced to allow specific limited services to pass in a safe way. That means that in order for any traffic to be allowed into the network, it must first match an explicit “allow” rule.

There are three main types of platforms for running a firewall:

- Dedicated McAfee NGFW appliances.
- Firewall software installed on a server dedicated to be used as a firewall.
- Firewall software running as a virtual machine in a virtualized server environment.

McAfee Firewalls are available on all of these platform types.

Regardless of the type of platform, the network structure in which the firewalls are placed must be carefully designed so that there are no loopholes or back doors. Firewalls can only control traffic that actually passes through them; even the most carefully planned firewall system can be undermined by a single back door that allows traffic to circumvent the firewall.

In addition to access control, modern firewall devices often include a variety of additional integrated features, such as *intrusion prevention systems* (IPS), *content filtering*, *anti-virus*, and *anti-spam*. In this chapter, the additional features are discussed separately, and the discussion focuses on the primary role of access control. Such additional features in McAfee Firewalls are covered in more detail in section [Additional Firewall Features](#) (page 21).

# Firewall Technologies

This section presents an overview to the main firewall techniques, and explains how McAfee Firewalls use them. The discussion here is limited to the traditional firewall component of a firewall system; the various additional inspection features that modern firewalls often incorporate are discussed separately.

Traditional firewall features are commonly achieved through three main techniques:

- packet filtering
- proxy firewalls
- stateful inspection.

The next sections first discuss these techniques separately and then explains how they can be utilized together to achieve an optimal balance between performance and security.

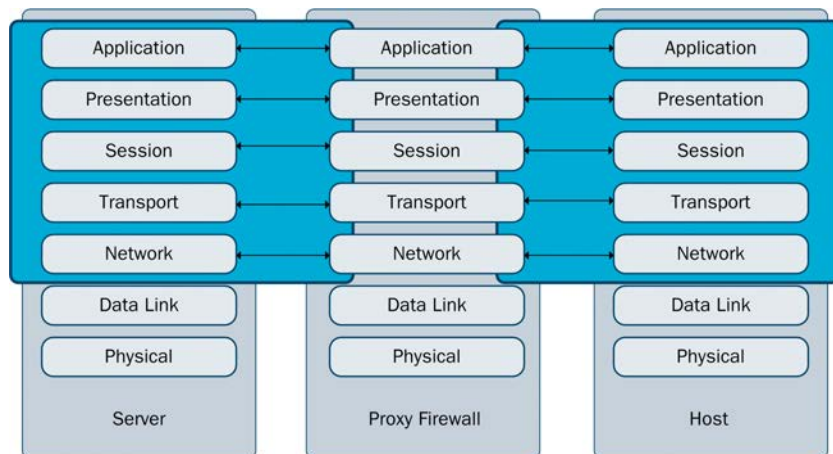
## Packet Filtering

Packet filtering examines the header information of packets and allows or stops each packet individually. In addition to firewalls, such simple *access control lists* (ACLs) are implemented on most common routing devices. Pure packet filters cannot protect against protocol misuse or other malicious contents in higher levels of the protocol stack. However, for some simple network protocols, packet filtering can be light on firewall resources and even provide an adequate level of protection.

## Proxy Firewalls

Proxy firewalls are firewalls running application proxy services. Proxies are a man-in-the-middle, and they establish their own separate connections to both the client and the server. This type of firewall is fully application-aware, and therefore very secure, but at the same time there's a trade-off in performance due to the inevitable increase in overhead.

**Illustration 2.1 Proxy Firewall Model**



# Stateful Inspection

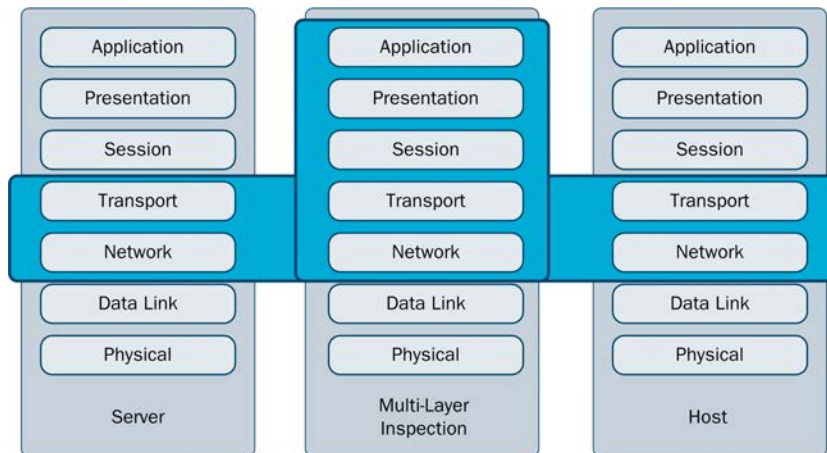
Stateful inspection firewalls are aware of basic networking standards and use historical data about connections in determining whether to allow or stop a packet. They track the established connections and their states in *dynamic state tables* and ensure that the connections comply with the security policies and protocol standards.

Since stateful inspection understands the context of connections (and therefore can relate the returning packets to appropriate connections), connections already determined to be “secure” can be allowed without full examination based on previous packets. This is especially important with services such as FTP, which can open several related connections that do not match a single basic profile. Even though Stateful inspection has some application awareness, it concentrates on protocols, not on inspecting data at the application layer.

## Multi-Layer Inspection

*Multi-Layer Inspection* combines application layer inspection, stateful inspection, and packet filtering technologies flexibly for optimal security and system performance. Like stateful inspection, the McAfee Firewall uses state tables to track connections and judge whether a packet is a part of an established connection or not. The McAfee Firewall also features application-layer inspection through specific *Protocol Agents*, when necessary, for enhanced security to inspect data all the way up to the application layer. The McAfee Firewall can also act as a packet filter for types of connections that do not require the security considerations of stateful inspection.

**Illustration 2.2 Multi-layer Inspection Model**



By default, all McAfee Firewall Access rules implement stateful inspection, but the administrator can flexibly configure rules with simple packet filtering or an additional layer of application level security as needed.

McAfee Firewalls apply application level inspection with or without proxying the connections, depending on what is required. Application level inspection can be selected to certain types of traffic by attaching a connection to a protocol-specific Protocol Agent.

Protocol Agents are also used to handle protocols that generate complex connection patterns, to redirect traffic to content inspection servers, and to modify data payload if necessary. For example, the FTP Protocol Agent, can inspect the control connection and only allow packets containing valid FTP commands. If an FTP data connection is opened using a dynamically assigned port, the Protocol Agent reads the port and allows the traffic. If NAT (network address translation) is applied to the connection, the Protocol Agent can also modify the IP address and port transported in the packet payload to allow the connection to continue despite the NAT. The Protocol Agents are covered in more detail in [Protocol Agents](#) (page 145).

## Additional Firewall Features

---

A firewall can have several different functions on a network. Although a firewall's primary function is to control network access, they can be used in several complementary roles depending on the firewall product used. This section concentrates on the main features available in McAfee Firewalls.

### Authentication

The primary task of any firewall is to control access to data resources, so that only authorized connections are allowed. Adding an authentication requirement to firewall policies allows the firewall to also consider the user before access is granted.

For more information on authentication, see [User Authentication on the Firewall](#) (page 207) and [External User Authentication](#) (page 213).

### Deep Packet Inspection and Unified Threat Management

*Deep packet inspection* includes measures such as virus detection, web content filtering, intrusion detection, or some other check of the actual data being transferred. When several such features are combined together with a firewall, the solution is often called unified threat management (UTM). McAfee's UTM solution includes:

- Virus checking.
- URL filtering.
- Intrusion detection.

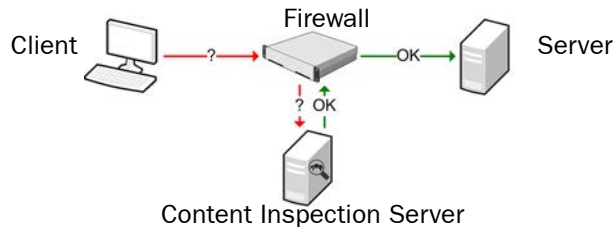
By combining several features, a UTM solution simplifies the physical network setup and makes the administration simpler. However, device performance limits can be quickly reached when several advanced inspection features are active. Therefore, UTM firewalls are generally used in environments where the traffic load stays relatively low even at peak times. When higher traffic volumes are needed, external content inspection servers and IPS devices are more often used for further inspecting the traffic.

For more information on the advanced traffic inspection features, see [Inspection Policies](#) (page 119), [Virus Scanning](#) (page 171), and [URL Filtering](#) (page 163).

## Integration With External Content Inspection

External *content inspection servers* (CIS) are a preferred choice in high traffic environments, as they offer better hardware optimization. Content inspection services can be run on a dedicated physical or virtual server that can be configured, scaled, and exchanged independently from the firewall. The firewall redirects the traffic to the CIS, which either strips anything deemed malicious from the packet or drops the packet altogether, according to what the security rules in force on the CIS define. Screened traffic continues to the destination.

**Illustration 2.3** Content Screening with CIS



For instance, incoming SMTP e-mail traffic could be forwarded from the firewall to the CIS for virus and content checking. The CIS removes suspicious content and the “scrubbed” packets are returned back to the firewall for routing to their final destination.

For more information on integrating a CIS with McAfee Firewalls, see [External Content Inspection](#) (page 175).

In addition to sending traffic to external content inspection, McAfee Firewalls also integrate with McAfee IPS and McAfee Layer 2 Firewalls. The Firewalls can accept blacklisting requests from the IPS and Layer 2 Firewalls. They can therefore stop traffic that the IPS engines or the Layer 2 Firewalls have detected to be harmful. For more information on McAfee IPS and Layer 2 Firewalls, see the *McAfee NGFW Reference Guide for IPS and Layer 2 Firewall Roles*.

## Load Balancing and Traffic Management

As an access controller with address translation duties, a firewall is also a natural point for affecting the distribution of traffic load. McAfee Firewalls utilize Multi-Link technology to flexibly use several standard network links to increase bandwidth and provide automatic failover when links go down.

For more information on traffic management, see [Outbound Traffic Management](#) (page 225) and [Inbound Traffic Management](#) (page 235).

Outbound bandwidth can be additionally managed through QoS measures by setting priorities, limits, and guarantees for different types of traffic.

For more information on the QoS features, see [Bandwidth Management and Traffic Prioritization](#) (page 245).

## Logging and Reporting

As a perimeter security device a firewall is a primary tool for logging the traffic that crosses or attempts to cross the network perimeter. Properly recorded log data can be used to monitor the capacity of networks, detect network misuse and intruders, and even to establish evidence to use against attackers.

Since a firewall operating in any corporate-type setting will quickly generate huge masses of log data, it is essential to have efficient tools to access and manage the logs in the form of filtered views, statistics, and reports. Consolidating logs from several sources is also vital in supporting the administrators in fully understanding the numerous network events.

For more information on logging, see the *McAfee SMC Reference Guide*.

## Network Address Translation (NAT)

Network address translation (NAT) modifies the IP headers of packets, changing IP address and port information for the source and/or destination.

Originally created to alleviate the problem of the rapidly diminishing IP address space, NAT has an added benefit; it can be used to conceal the private IP addresses of hosts and the structure of an internal network. In fact, NAT enables even hiding an entire network behind a single public IP address. As handy as NAT is, it is important to understand that NAT is not primarily a security feature. It simply a method of modifying packets that lends itself to security applications.

For more information on NAT, see [Network Address Translation \(NAT\)](#) (page 131).

## VPNs

VPNs (virtual private networks) conceal and encrypt traffic between end-points to establish a virtual, secure tunnel through an insecure network.

In IPsec VPNs, a firewall transparently encrypts and decrypts data exchanges at the network layer with some other IPsec VPN end-point on behalf of any number of computers. IPsec VPNs can also provide remote access to internal resources for individual client computers that have a VPN client application installed. IPsec VPNs are a good fit for VPN access that involves many communicating parties and/or many different applications.

SSL VPNs (secure socket layer virtual private networks) provide clientless access by utilizing the SSL encryption features included in web browsers. Users log in to a portal to access those resources that administrators have specifically configured. SSL VPNs are a good fit when there is a need to provide remote access to a few specific resources from various different types of devices and platforms.

The SSL VPN is available as a separate appliance product. For more information on the SSL VPN, refer to the *SSL VPN Administrator's Guide*.

IPsec VPN features are integrated in the Firewall. For more information on IPsec VPNs, see [Overview to VPNs](#) (page 261). For more information on how IPsec VPNs are configured, see [Policy-Based VPN Configuration](#) (page 269).

## Complexity of Administration

When a complex system is maintained with limited resources, the ease of administration becomes crucial. A great part of the benefits of a security system are wasted if administrators find it difficult to keep up with monitoring the system and the requests for adjusting its policies, if upgrades have to be postponed due to the effort required, or if there is no support for checking and finding errors in the configuration.

Ease of administration is central to the McAfee Security Management Center (SMC). The SMC's centralized management system provides the administrators more visibility into the whole network, simplifies and automates system maintenance tasks, and reduces the work required to configure the system.

## Single Point of Failure

As a network choke point, the failure of the firewall to pass traffic can mean that the network connectivity is completely cut off. In some environments, this small risk can be considered acceptable. However, an increasing number of organizations require network connectivity to conduct business, so a reliable high availability solution is required.

McAfee Firewalls have built-in support for clustering, which allows operating up to 16 physical Firewall devices as a single unit. All units can actively handle traffic at the same time. No special configuration is required in the surrounding network to achieve this, as the whole implementation is achieved through basic networking standards. Units can be plugged in, taken out, and replaced flexibly without cutting network connectivity from the users. For more information on clustering McAfee Firewalls, see [Firewall Cluster Configuration](#) (page 51).

## Worms, Viruses, and Targeted Attacks

As essential as a firewall is, it should not cause a false sense of being safe from all harm in the organization. There are many security threats that the firewall cannot stop, even if it incorporates several different kinds of additional inspection methods:

- Many virus and worm outbreaks and even many intentional attacks may start within an organization's internal network. Malicious code can be introduced to the network on removable media, unauthorized equipment attached to the network, or on the laptops of travelling users. The firewall has no way to detect and prevent something before it crosses a network boundary that the firewall enforces.
- Attackers may be able to bypass security measures by obtaining legitimate credentials of users or even administrators through spying and social engineering. If the firewall is not properly secured, it may itself be susceptible to a targeted attack. If the attacker gains remote administrator access or physical access to the firewall, the system can be covertly altered to allow and conceal further malicious activities.
- A denial-of-service attack may consume all of the inbound bandwidth before any of the organization's own security devices receive the traffic.

In many of these cases, the firewall may still be useful for containing damage and for collecting more information on what has taken place.



## CHAPTER 3

# INTRODUCTION TO McAfee NGFW IN THE FIREWALL/VPN ROLE

This chapter gives you an overview of the system architecture of McAfee Next Generation Firewall from Intel Security (NGFW) in the Firewall/VPN role, and how the engines inspect traffic.

The following sections are included:

- ▶ [The McAfee NGFW Solution](#) (page 26)
- ▶ [SMC Components](#) (page 27)
- ▶ [Main Benefits of McAfee Firewall/VPN](#) (page 29)

# The McAfee NGFW Solution

---

The McAfee NGFW engine in the Firewall/VPN role is part of the McAfee NGFW solution, which is especially well-suited to complex and distributed network environments. In addition to Firewalls and virtual private networking, the McAfee NGFW solution also provides intrusion detection and prevention.

The configuration, monitoring, and control of the system is done through a centralized Security Management Center (SMC) 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:

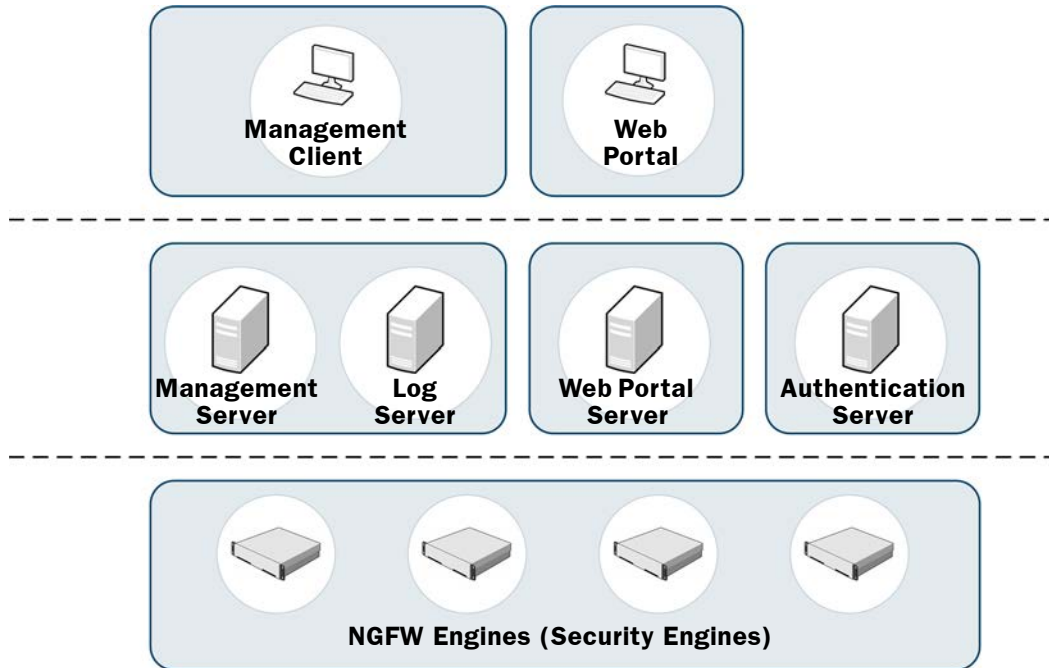
- Interaction between the McAfee Firewall/VPN, McAfee IPS, McAfee Layer 2 Firewall, Master Engine, and Virtual Security Engine components in the same SMC creates security benefits by allowing automatic coordinated responses when a security threat is detected, providing instant blocking of unwanted traffic, and reducing the need for immediate human intervention.
- Multiple administrators can log in at the same time to efficiently configure and monitor all McAfee NGFW engines. The SMC provides a single user interface that allows unified configuration, monitoring, and reporting of the whole McAfee NGFW solution with the same tools and within the same user session.
- The reuse of configuration information across components in the SMC allows you to avoid the duplicate work of configuring the same details for all components individually or exporting and importing the configurations between multiple SMCs.
- The SMC 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 without sacrificing its ease-of-use.

# SMC Components

---

The SMC components and their roles are illustrated below.

**Illustration 3.1 SMC Components**



One Security Management Center can manage a large number of Security Engines, Master Engines, and Virtual Security Engines. The distributed architecture allows deploying the components effectively in different network environments. You can flexibly add, remove, and reposition SMC components according to your needs.

**Table 3.1 SMC Components**

Component	Description
Management Clients	Provide a user interface for configuring, controlling, and monitoring the system. Connect to the Management Server.
Management Servers	Store all configuration data, relay commands to the engines, and notify administrators of new alerts in the system.
Log Servers	Store logs and correlate events detected by multiple Security Engines.
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.
NGFW Engines (Security Engines)	<p>Inspect and filter traffic. Correlate events in traffic inspected by the engine itself. NGFW engines are represented by Security Engine elements in the SMC.</p> <p>NGFW engines that have a license that allows the creation of Virtual Resources can be used as a Master Engine to provide resources for Virtual Security Engines. See <a href="#">Master Engine and Virtual Firewall Configuration</a> (page 65).</p>

All communications between SMC components are authenticated and encrypted. The engines work independently according to their installed configuration, so even if the connections to the Security Management Center are cut, traffic inspection continues without interruption.

## Firewall/VPN Engines

The term Firewall engine refers to the combination of the physical device and the McAfee NGFW software in the Firewall/VPN role. Firewall engines have the following representations in the SMC:

- The Single Firewall and Firewall Cluster elements are containers for the main configuration information directly related to the Firewall.
- The individual physical Firewall engines are shown as one or more *Nodes* under the main Firewall element in some views of the Management Client.

The Firewall engine includes 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 NGFW software upgrade.

# Main Benefits of McAfee Firewall/VPN

---

In addition to standard firewall features, the McAfee Firewall/VPN provides additional advanced features.

## Advanced Traffic Inspection

The Firewall's traffic inspection process is designed to ensure a high level of security and throughput.

The Firewalls' policies determine when to use stateful connection tracking, packet filtering, or application-level security. The Firewall uses the resources necessary for application-level security only when the situation demands it, and without unnecessarily slowing or limiting network traffic.

Some types of connections can be selected for inspection of the data content against harmful or otherwise undesired patterns in connections. The deep packet inspection features provide IPS-type capabilities right on the Firewall, and help in finding and stopping malicious or suspicious network activities. You can even inspect the content of encrypted HTTPS connections using the built-in deep packet inspection features.

An antivirus scanner complements the standard traffic inspection features when the Firewall is licensed for the UTM (unified threat management) feature.

## Built-in Clustering for Load Balancing and High Availability

The McAfee Firewall provides innovative built-in clustering and load-balancing features that provide several benefits over traditional solutions.

Traditionally, in order to achieve high availability on the Firewall itself, additional hardware switches, software clustering products, or special load-balancing devices have been added and maintained. This often results in the transfer of a *single point of failure* to another network component — typically the network link.

In the McAfee Firewall, however, the clustering of the Firewall engines is integrated in the product, thus introducing true *built-in high availability* and *load balancing*. The Firewall engines dynamically load-balance individual connections between the cluster nodes, transparently transferring connections to available nodes in case a node becomes overloaded or experiences a failure.

A Firewall Cluster can have a maximum of 16 nodes. With load balancing, the processing of network traffic is automatically balanced between the cluster nodes. This way, the performance of the Firewall can be upgraded by simply adding new nodes to the cluster when necessary. Individual nodes can also be taken offline during business hours for maintenance purposes; connections that were handled by that particular engine are transparently redistributed to other online nodes.

The McAfee Firewall also comes with built-in technology for high availability and load balancing between different network connections as explained in the next section.

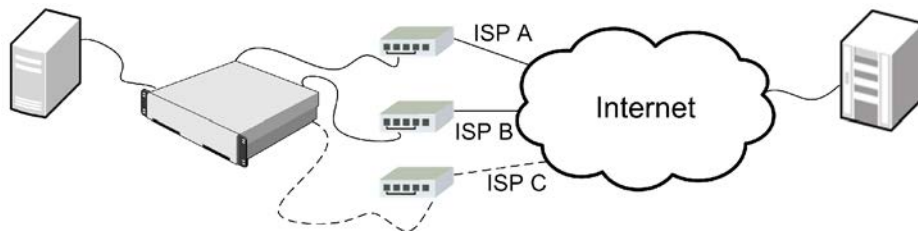
# Multi-Link Technology

Single-node and clustered Firewall installations both support *Multi-Link*, which ensures high availability for network connections by utilizing alternative network links.

Multi-Link allows configuring redundant network connections out of standard network connections without the complexity of traditional solutions that require redundant external routers and switches. In contrast to many alternative solutions, there is no need to use complex routing protocols, such as *Border Gateway Protocol* (BGP) and *Hot Standby Routing Protocol* (HSRP), and peering arrangements between the ISPs.

Any IP-based link with a dedicated IP address range can be used as part of a Multi-Link configuration. You can also define standby links that are used only when primary links fail. The illustration that follows shows a basic Multi-Link setup with a single Firewall that has two active and one standby network links to the Internet.

**Illustration 3.2 Multi-Link Technology**



Most often, multiple network links are used to ensure continuity of Internet access, but Multi-Link can be used to provide redundant links to internal networks as well. Traffic is dynamically balanced across the different links based on a performance measurement or based on the links' relative bandwidths. The traffic automatically fails over to other links when the Firewall detects that one of the links fails. The Firewall uses network address translation (NAT) to direct the traffic through the different links to make the source IP address valid for the link used.

Multi-Link technology provides highly available network connections for the following scenarios:

- Outbound connections: *Multi-Link routing* ensures that outbound traffic always uses the optimal link towards its destination and allows configuring standby links as backups. The traffic can be distributed across the links in several different ways. For more information, see [Outbound Traffic Management](#) (page 225).
- Inbound connections: the built-in inbound traffic management feature can utilize Multi-Link for ensuring continuity of services your company offers to external users. For more information, see [Multi-Link for Server Pools](#) (page 237).
- VPN connections: the Multi-Link tunnel selection for VPN traffic is done independently from other types of traffic. Standby links can also be selected independently for a VPN. For more information, see [Multi-Link and Policy-Based VPNs](#) (page 287).

## Built-in Inbound Traffic Management

The built-in Server Pool feature allows Firewalls to monitor a pool of alternative servers that offer the same service to the users. If one of the servers becomes unavailable or overloaded, the Firewall automatically redirects new connections to the alternative servers. Server pools can also interact with the Multi-Link feature for high availability of the incoming network connection.

For more information, see [Inbound Traffic Management](#) (page 235).

## QoS and Bandwidth Management

Quality of Service (QoS) Policies are interface-specific rules on a Firewall that help you ensure that important network services are given priority over less important traffic. Quality of Service and bandwidth management features are not supported for Modem interfaces of single Firewalls.

With QoS rules, you can set up a minimum guaranteed bandwidth and maximum bandwidth limit for traffic, and set a priority value for the traffic. You can optionally define settings for Active Queue Management (AQM) to queue and send traffic according to a scheduling algorithm. This reduces the volume of dropped or retransmitted packets when there is network congestion.

In addition to or instead of QoS rules, you can also create DSCP Match/Mark rules that read or write DiffServ Code Point (DSCP) type of service (ToS) field values. This allows you to integrate the Firewall with other network equipment that implements QoS management in your own or your ISP's network.

For more information, see [Bandwidth Management and Traffic Prioritization](#) (page 245).

## Integration with IPS and Layer 2 Firewalls

The interoperation between Firewall/VPN, IPS, and Layer 2 Firewall products makes the combination of these a very powerful network security solution. IP address blacklisting is a shared feature for Firewall/VPN, IPS, and Layer 2 Firewalls. This allows blocking harmful traffic not just at the component that detects it, but also on other engines on the connection path.

## Clustered Multi-Link VPNs

The Firewall provides fast, secure, and reliable IPsec VPN connections with the added benefits of the clustering and Multi-Link technologies that provide load balancing and failover for both the VPN gateways and the network connections. The system's scalability allows administrators full control over how many tunnels are created and used.

The VPN links can be in three different modes: *active*, *aggregate*, and *standby*. If there are multiple links in active mode, traffic is dynamically balanced across the different links based on a performance measurement or based on the links' relative bandwidths. If there are multiple links in aggregate mode, each connection is balanced between all the aggregate links in round robin fashion. The standby links are only used if the active or aggregate links fail.

For more information on VPNs, see [Overview to VPNs](#) (page 261) and [Policy-Based VPN Configuration](#) (page 269).





## CHAPTER 4

# NGFW DEPLOYMENT IN THE FIREWALL/VPN ROLE

This chapter provides general guidelines for deploying McAfee Next Generation Firewall from Intel Security (NGFW) in the Firewall/VPN role. It also illustrates a typical deployment with an example scenario.

The following sections are included:

- ▶ [Deployment Overview](#) (page 34)
- ▶ [Positioning Firewalls](#) (page 36)

## Supported Platforms

McAfee NGFW engines in the Firewall/VPN role can be run on the following general types of platforms:

- Purpose-built McAfee NGFW appliances.
- Standard Intel-compatible servers. Search for the version-specific *Hardware Requirements* at [http://www.stonesoft.com/en/customer\\_care/kb/](http://www.stonesoft.com/en/customer_care/kb/).
- Virtualization platforms that support the deployment of Open Virtual Format (OVF) templates. VMware is officially supported. Other virtualization platforms may also be supported. More information can be found in the *McAfee NGFW Installation Guide for Firewall/VPN Role*.

NGFW appliances can also be used as a Master Engine to provide resources for Virtual Security Engines. For more information about Master Engines and Virtual Security Engines, see [Master Engine and Virtual Firewall Configuration](#) (page 65). Master Engines can be run on the following general types of platforms:

- Purpose-built McAfee NGFW appliances with 64-bit architecture.
- Standard Intel-compatible servers with 64-bit architecture.

The NGFW engine software includes an integrated, hardened Linux operating system. This eliminates the need for separate operating system installation, configuration, and patching.

# General Deployment Guidelines

Table 4.1 summarizes the general deployment guidelines for Firewall/VPN engines, Master Engines, and the Security Management Center (SMC). The SMC deployment considerations are described in more detail in the *McAfee SMC Reference Guide*.

**Table 4.1 General Guidelines for Firewall/VPN Deployment**

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	Place 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.
Management Server	Position on a central site where it is physically accessible to the administrators responsible for maintaining its operation.
Firewalls	Position Firewall(s) at each location so that all networks are covered. Firewalls can be clustered. Functionally, the Firewall Cluster is equal to a single high-performance Firewall. Cluster deployment involves setting up a heartbeat link between the Firewalls that allows the devices to track each others' operating status, agree on the division of work, and exchange information on traffic.
Master Engines	Position Master Engine(s) at locations where Virtual Security Engines are needed, such as at a hosting location for MSSP services, or between networks that require strict isolation. Master Engines can be clustered. A clustered Master Engine provides scalability and high availability. In a Master Engine Cluster, the Virtual Resource is active in one Master Engine at a time. Cluster deployment involves setting up a heartbeat link between the engines that allows the devices to track each others' operating status, agree on the division of work, and exchange information on traffic.

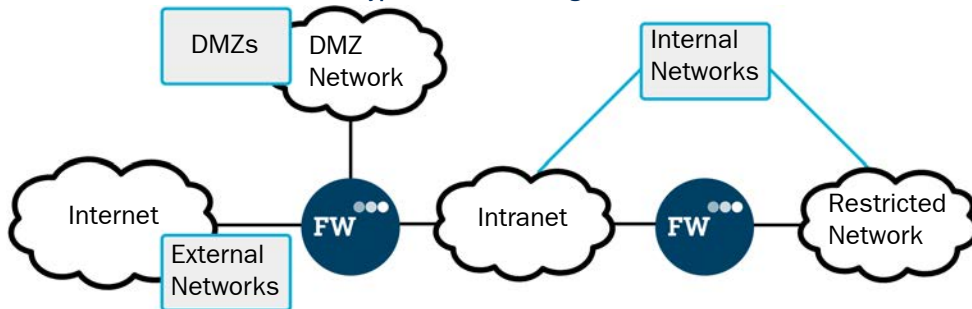
# Positioning Firewalls

The Firewall is a perimeter defense, positioned between networks with different security levels. Firewalls generally control traffic between:

- External networks (the Internet) and your internal networks.
- External networks (the Internet) and DMZ (demilitarized zone) networks.
- Between internal networks (including DMZs).

Firewalls separate the different networks by enforcing rules that control access from one network to another.

**Illustration 4.1** The Firewall in Different Types of Network Segments



Not all organizations necessarily have all of the types of networks that are shown here. One Firewall can cover all enforcement points simultaneously if it is practical in the network environment and compatible with the organization's security requirements.

The next few pages of this guide explain in more detail how Firewalls meet the particular requirements of each of the different types of networks (external, internal, and DMZ networks).

# External to Internal Network Boundary

The most common and most important use for a firewall is to separate internal networks from the public Internet.

Table 4.2 External Network Considerations for Firewalls

	Description	Implications on Firewalls
Main purpose	Connectivity between the protected and public networks.	The Firewall selects which traffic is permitted into and out of the internal networks and translates addresses between internal IP addresses and public IP addresses. The Firewall is typically also a VPN end-point.
Hosts	Only equipment that needs to be directly connected to the public network, such as routers and the Firewall.	The communicating hosts in external networks are unknown in many cases. IP address spoofing is a possibility. External hosts can be trusted if they are identified using VPN authentication mechanisms.
Users	Access to this network is open, but local access to the hosts is usually restricted to the administrative staff only.	Internal users are known and trusted. Users in public networks are unknown and untrusted. VPN authentication and encryption can be used to allow specific users access from external networks to internal resources.
Traffic volume	Varies from low to high, generally the full bandwidth of all Internet links combined.	Hardware requirements vary depending on the environment. Clustering allows flexible firewall throughput adjustments. Multi-Link allows high availability and load balancing for outbound connections. QoS Policies can control the bandwidth use.
Traffic type	Any type of traffic may be encountered, especially in incoming traffic, although some filtering may be done by the Internet service provider.	The Firewall controls which traffic is allowed access into your networks, but it is beyond the Firewall's control what and how much traffic it receives from the public networks. Advanced inspection checks can be activated on the Firewall and/or on an external content inspection server depending on the protocol.
Network security	Little or no access controls to pre-filter traffic arriving from the Internet. The hosts in this network should all be security-hardened and actively patched against known vulnerabilities.	The Firewall's policy should be as restrictive as possible. Generally, new connections are not allowed from the external to the internal networks (servers for external services are placed in DMZs). SSH access to the Firewall's command line from external networks should be disabled after use.

# Internal Network Boundaries

Internal networks are mixed environments with servers and end-user computers. Firewalls restrict traffic between the different internal networks, but traffic within each network is often not secured in any significant way.

**Table 4.3 Internal Network Considerations for Firewalls**

	Description	Implications on Firewalls
Main purpose	Network services and connectivity for authorized end-users. Back-end servers that serve other networks and user groups.	Internal networks transfer confidential data but can be very permissive for the traffic within the network. Firewalls can control access between different internal networks to enforce different security levels and prevent some types of network threats.
Hosts	Mixed environment consisting of servers, laptops, desktops, network printers, copiers, etc.	Network communications of the servers and the end user computers differ in characteristics. Hosts can be actively maintained and patched to reduce some types of risks. Access between networks can be restricted based on the type of host. Firewall logs provide a record of network use and alerts can be configured for unusual connection attempts.
Users	Authorized personnel.	Users can be considered trusted, but on various levels. The Firewall authenticates users for access between internal networks that have different security levels.
Traffic volume	Varies from low to high. Grows highest at network choke-points in large environments.	Installation at network choke-points often requires high-performance hardware. Clustering can provide load balancing and high availability in critical locations.
Traffic type	Diverse, with a large number of different applications communicating within and in/ out of the network.	The Firewall policy must balance users' demands for a wide range of different services with the need to keep the internal networks safe. Advanced inspection features further inspect permitted communications.
Network security	A "trusted network" where the users and the traffic are considered to be authorized.	The Firewall establishes boundaries between networks to protect sensitive data and essential services. Availability of network services sometimes overrides security.

# DMZ Network Boundaries

DMZ networks (demilitarized zone networks, also known as perimeter networks) are isolated environments for servers that offer services mainly for external access.

Table 4.4 DMZ Considerations for Firewalls

	Description	Implications on Firewalls
Main purpose	DMZs provide a limited number of services, mostly for external users. The services are often business-critical and open for completely public access.	The Firewall selects which traffic is permitted into and out of the DMZs. The Firewall typically also translates IP addresses from public IP addresses that are routable in the external networks to private addresses used in internal networks. VPNs may be used to provide services for partner-type users.
Hosts	A uniform environment consisting mainly of servers that often provide public or semi-public services.	A limited number of services are provided to an often large number of hosts. Some types of administrative access is allowed to a few specific trusted hosts.
Users	Mostly unknown, but some services may be for specific users. Administrators have wider privileges.	Users are often unknown or authenticated by the target servers themselves. Firewall authentication may be useful for restricting administrator access from internal networks.
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 depending on the environment. Clustering allows flexible adjustments to throughput. The inbound traffic management features can balance traffic between redundant servers.
Traffic type	Rather uniform traffic, with only specific applications and servers communicating within, into, and out of the networks.	The Firewall controls which traffic is allowed access in and out of each DMZ from external and internal networks. Usually, only a few specific services have to be allowed. Advanced inspection checks can be activated on the Firewall and/or on an external content inspection server depending on protocol.
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 target for attacks. Connections between the DMZ networks and to other internal networks facilitate further attacks, so these connections must be strictly controlled.





# INTERFACES AND ROUTING

---

## **In this section:**

**Single Firewall Configuration - 43**

**Firewall Cluster Configuration - 51**

**Master Engine and Virtual Firewall Configuration - 65**

**Routing and Antispoofing - 73**



## CHAPTER 5

# SINGLE FIREWALL CONFIGURATION

A Single Firewall is a firewall that has only one firewall engine (instead of a cluster of two or more engines).

The following sections are included:

- ▶ [Overview to Single Firewall Configuration](#) (page 44)
- ▶ [Configuration of Single Firewalls](#) (page 44)
- ▶ [Example of a Single Firewall Deployment](#) (page 48)

# Overview to Single Firewall Configuration

---

A Single Firewall can be deployed at sites where the performance benefits and high availability provided by a Firewall Cluster are not essential. High availability of network connections is still available, since Single Firewalls support Multi-Link. See [Outbound Traffic Management](#) (page 225) and [Inbound Traffic Management](#) (page 235) for information on using Multi-Link.

Single Firewalls also support the following features that are unavailable on Firewall Clusters:

- They can have dynamic IPv4 addresses on their interfaces.
- They support wireless links through 3G modems connected to the firewall engine's USB ports.
- They support ADSL connections. This feature is only available on specific pre-installed McAfee NGFW appliances that have an ADSL network interface card.
- They support wireless connections. This feature is only available on specific pre-installed McAfee NGFW appliances that have a wireless network interface card.

The Single Firewall engine software can run on a McAfee NGFW appliance, on a standard server with an Intel-compatible processor, or as a virtual machine on a virtualization platform. You can also configure anti-virus scanning and anti-spam filtering (requires separate licenses). See [Virus Scanning](#) (page 171) and [Spam Filtering](#) (page 167) for more information.

This chapter concentrates on network interface configuration, which is the only part of the basic firewall configuration that has major differences between a Single Firewall and a Firewall Cluster. Other features, including routing and antispoofing, are explained together for both types of installations in separate feature-specific chapters.

## Configuration of Single Firewalls

---

Firewalls are configured and managed centrally through the Management Server. The Single Firewall element represents the firewall's configuration on the Management Server. The main configuration options in the Single Firewall element are the settings related to network interfaces. This chapter concentrates on those settings.

### Dynamic Firewall Interface Addresses

Single Firewalls support the use of DHCP, PPPoA, and PPPoE to assign dynamic IPv4 addresses on the firewall's network interfaces. Typically, a dynamic IP address is used in smaller sites with xDSL connections that have no static IPv4 address available for the firewall. If you use a 3G modem with the firewall to provide a wireless link for connections to the Management Server or to the Internet, the Modem Interface has a dynamic IP address that is assigned automatically by a PPP daemon.

Instead of an IP address, each interface with a dynamic IP address is identified in the SMC by a DHCP Index, an arbitrary number of your choice that is used to distinguish different interfaces with dynamic IP addresses from one another.

When a firewall has a fixed IP address, the Management Server can contact the firewall whenever there is need. When the firewall has a dynamic IP address, the Management Server does not have a fixed IP address to contact, so the firewall contacts the Management Server instead. You can also define that a firewall that has a fixed IP address contacts the Management Server. The frequency of these contacts can be adjusted as necessary. If the

contact is lost (for example, the Internet connection goes down), the Management Server queues the commands you have made to the firewall and executes them when contact is re-established.

Dynamic IP addressing also affects the VPN configuration in much the same way as in management communications. The Firewall with the dynamic address always has to open the VPN tunnel. After the VPN tunnel is established, connections can be made in both directions as usual. VPN client connections can be forwarded through a gateway-to-gateway VPN from some gateway that has a static IP address (VPN hub configuration).

## Internal DHCP Server

Single Firewalls contain an internal DHCP server that can be used to assign IPv4 addresses to hosts in the protected network. This is meant for branch office type installations where it may be more practical to assign the IP addresses using the firewall rather than relay the DHCP requests from a separately maintained local DHCP server or from some other site's DHCP server through a VPN.

## Security Strength of Management Connections

You can optionally use 256-bit encryption for the connection between the engines and the Management Server. This requires both the engines and the Management Server to be version 5.5 or higher. When you create and use a new Internal ECDSA Certificate Authority to sign certificates for system communication, the Management Server and the engine re-establish their trust relationship and 256-bit encryption is enabled for the connection between the engines and the Management Server.

## Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create Single Firewall Elements

To introduce new Single Firewalls to the SMC, you must define Single Firewall elements that store the configuration information related to the firewall engines. You can define Single Firewall elements one-by-one, or define several Single Firewall elements at once using the Multiple Single Firewalls Wizard.

### Task 2: Define Physical Interfaces

A *Physical Interface* represents an actual network port on the engine. The number of defined Physical Interfaces can be lower than the number of network ports on the engine hardware. A *Normal* physical interface represents a single network port on the engine. An *Aggregated Link* represents two network ports on the engine. An Aggregated Link provides protection against hardware failure. You can also use an Aggregated Link in load-balancing mode to increase throughput.

By default, the Physical Interface definitions are mapped to the network ports on the engines one to one. If necessary, you can change the mapping using command line tools on the engine.

### Task 3: Define VLAN Interfaces

A *Virtual Local Area Network* (VLAN) is a logical grouping of hosts and network devices that allows creating several separated networks on a single physical link. To allow this separation, the Firewall supports *VLAN tagging* as defined in the IEEE 802.1q standard. One network interface can support up to 4094 VLANs.

The defined VLAN interfaces are displayed, for example, as “5.202” for network interface 5 with VLAN 202.

### Task 4: Define Tunnel Interfaces

*Tunnel Interfaces* are logical interfaces that are used as end-points for tunnels in the Route-Based VPN. Any traffic that is routed to a Tunnel Interface and allowed by the Access rules is automatically sent through the tunnel to the peer end-point defined in the Route-Based VPN. Defining Tunnel Interfaces is optional: Tunnel Interfaces are only used in the Route-Based VPN. See [Route-Based VPN Configuration](#) (page 295) for more information.

You can optionally add IPv4 and IPv6 addresses to a Tunnel Interface. Adding an IP address allows you to define the source IP address of traffic sent from the engine node itself.

### Task 5: Define an ADSL Interface

You can optionally configure one *ADSL Interface* for ADSL connections (ANSI T1.413 issue 2n, G.dmt, G.lite, ADSL2 DMT, ADSL2 G.lite, Annex A, and Annex B are supported). ADSL Interfaces are only available on specific pre-installed McAfee NGFW appliances that have an ADSL network interface card. Use the number of the ADSL port on the appliance as the Interface ID of the ADSL Interface.

### Task 6: Define a Wireless Interface

You can optionally define a *Wireless Interface* that allows you to use the firewall as a wireless access point. Wireless Interfaces are only available on specific pre-installed McAfee NGFW appliances that have a wireless network interface card. Use the number of the wireless port on the appliance as the Interface ID of the Wireless Interface.

You can define several wireless LANs for the Wireless Interface. A wireless LAN is defined by adding an SSID (service set identifier) interface for the Wireless Interface.

### Task 7: Define IP Addresses

You can define several IPv4 and/or IPv6 addresses for the same Physical Interface, VLAN Interface, or Aggregated Link Interface pair. You can define several IPv4 addresses for the optional ADSL Interface. Each SSID Interface defined for the optional Wireless Interface can have a single IPv4 or IPv6 address. Only IPv4 addresses are used in system communications.

If you want to configure multicast routing by using the firewall as an IGMP proxy, the IP addresses that you define for the downstream interfaces must be the lowest IP addresses among all the IGMP queriers in the local networks. For more information on IGMP-based multicast forwarding, see [Multicast Routing](#) (page 78).

## Task 8: Define Loopback IP Addresses

You can optionally define one or more *loopback IP addresses* for the firewall. Loopback IP addresses allow you to assign IP addresses that do not belong to any directly-connected networks to the firewall. Loopback IP addresses are not connected to any physical interface and they do not create connectivity to any network. Any IP address that is not used to route traffic on another interface can be used as a loopback IP address. The same IP address can be used as a loopback IP address and as the IP address of a Tunnel Interface.

## Task 9: Define Modem Interfaces

You can optionally define one or more *Modem Interfaces* on the firewall. A Modem Interface represents the settings of a 3G modem that provides a wireless link to the Internet or to the Management Server. The 3G modem is connected to a USB port on the firewall engine. Each Modem Interface is identified with a Modem Number in the Management Client. The Modem Number is mapped to the modem's IMEI (international mobile equipment identity) number, and each modem is assigned a unique ID when you connect the modem to the firewall engine. If necessary, you can change the mapping between the modem's IMEI number and the modem ID through the engine command line.

## Task 10: Install the Firewall Engine

During the engine installation, you map the physical network interfaces on the engine and the modem(s) connected to the USB port(s) on the engine to the Physical Interfaces, the ADSL Interface, the Wireless Interface, and the Modem Interfaces that you defined in the Management Client. The mapping of Tunnel Interfaces to physical network interfaces on the engine is done automatically by the engine operating system based on the routing configuration.

You can optionally include a policy in the initial configuration to transfer the complete configuration to the firewall with information on all interfaces, routing, policies, and other settings. The policy is automatically activated on the engine after initial contact is established with the Management Server. This allows the firewall to start processing traffic immediately.

You can configure the engine automatically using a configuration saved on a USB memory stick. If you use the automatic engine configuration, the Physical Interface, ADSL Interface, Wireless Interface, and Modem Interface definitions in the Management Client are automatically mapped to the physical network interfaces on the engine and to the modem(s) connected to the engine's USB port(s). If you configure the engine automatically through a Modem Interface, you must select the IPv4 address on Modem Interface 0 as the primary control IP address. If necessary, you can change the Modem number mapping of the Modem Interface and the Interface ID mapping of the other types of interfaces after the initial configuration using command line tools on the engine. See the *Appliance Installation Guide* delivered with the appliance for more information.

Alternatively, you can upload the initial configuration to the Installation Server for plug-and-play configuration. In plug-and-play configuration, the appliances automatically connect to the Installation Server, and the initial configuration is transferred from the Installation Server to the appliances. Only specific appliances are compatible with plug-and-play configuration.

During the installation, a basic initial configuration is activated and an initial contact between the Management Server and the engine is initiated. During the initial contact, the engine authenticates itself to the Management Server with a one-time password. When this initial contact succeeds, the engine receives a certificate from the SMC for authenticating all subsequent communications — a trust relationship between the engine and the Management Server is established. The one-time password expires at that time, since it is unnecessary from that point onwards.

## Task 11: Install a Firewall Policy

If you do not include a policy in the initial configuration, only the interface used for the control connection with the Management Server is configured after the engine establishes contact with the Management Server. You must install a Firewall Policy using the Management Client to transfer the complete configuration to the firewall. After this is done, the firewall is ready to start processing traffic.

## Example of a Single Firewall Deployment

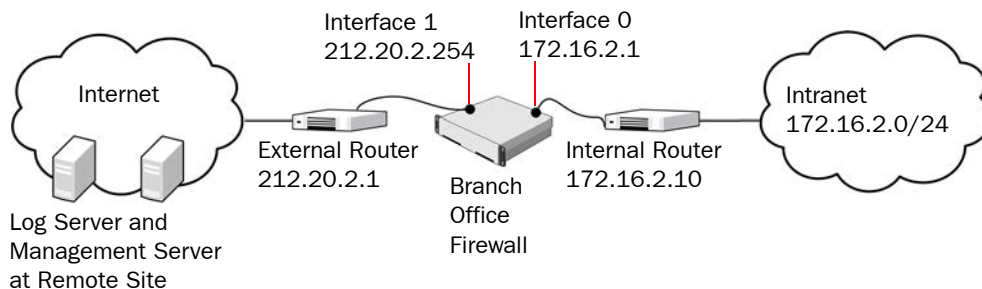
The examples in this section illustrate common deployment scenarios for a Single Firewall and general overviews to the steps on how each scenario is configured.

### Setting up a Single Firewall

Company A has just opened a new branch office. The administrator at the branch office is setting up a Single Firewall in the branch office network.

[Illustration 5.1](#) shows the network at the branch office.

**Illustration 5.1 Branch Office Network**



The Branch Office Firewall has two interfaces with internal and external routers:

- The internal router is connected to Interface ID 0.
- The external router is connected to Interface ID 1.



The SMC has already been installed at the remote site, and the branch office administrator is now ready to install and configure the Single Firewall. The administrator:

1. Creates a Single Firewall element (Branch Office Firewall) and defines the Log Server at the remote site as its Log Server.
2. Creates an interface for connecting to the internal router and gives it the following properties:
  - Interface ID: 0.
  - IP Address: 172.16.2.1.
3. Creates an interface for connecting to the external router and gives it the following properties:
  - Interface ID: 1.
  - IP Address: 212.20.2.254.
4. Saves the initial configuration of the Branch Office Firewall on a USB stick.
5. Installs the firewall engine in the server room.
6. Inserts the USB stick in the firewall, turns it on, and waits until the Management Client shows that contact is established between the engine and the Management Server.
7. Checks the routing configuration and adds the first few rules for allowing traffic through the firewall.
8. Installs a Firewall Policy using the Management Client to transfer the first working configuration to the firewall.

## **Adding a New Interface to an Existing Configuration**

In the previous example, the administrator initially configured the firewall at the company's new branch office with just two interfaces. Now the administrator decides to add a physically separated DMZ network for access to/from that office's mail server to properly control both internal and external traffic with this publicly exposed server. The administrator:

1. Creates an interface for the DMZ and gives it the following properties:
  - Interface ID: 2.
  - IP Address: 192.168.2.1.
2. Creates new rules in the firewall's policy to allow traffic to/from the DMZ and NAT rules to translate between the private and public IP address of the mail server.
3. Connects the new DMZ router to the firewall.
4. Installs a Firewall Policy using the Management Client to transfer the new working configuration to the Firewall.



## CHAPTER 6

# FIREWALL CLUSTER CONFIGURATION

A Firewall Cluster is a group of firewall nodes that work as a single logical entity to share the load of traffic processing and provide redundancy, ensuring the availability of network services to the users.

The following sections are included:

- ▶ [Overview to Firewall Cluster Configuration](#) (page 52)
- ▶ [Configuration of Firewall Clusters](#) (page 53)
- ▶ [Using a Firewall Cluster](#) (page 60)
- ▶ [Examples of Firewall Cluster Deployment](#) (page 62)

# Overview to Firewall Cluster Configuration

---

A *Firewall Cluster* consists of 2 to 16 nodes that function as a single entity. Clustering is a standard feature that you can activate on any McAfee Firewall/VPN installation if you have two or more licensed engines.

You can also configure Multi-Link on the Firewall Cluster to provide high-availability for inbound and outbound connections. See [Outbound Traffic Management](#) (page 225) and [Inbound Traffic Management](#) (page 235) for more information. You can optionally also configure anti-virus scanning and anti-spam filtering (separately-licensed features). See [Virus Scanning](#) (page 171) and [Spam Filtering](#) (page 167) for more information.

This chapter concentrates on the configuration of network interfaces, IP addresses, and clustering, which are the only parts of the basic configuration where there are major differences between a Single Firewall and a Firewall Cluster. The section [Using a Firewall Cluster](#) (page 60) describes other configuration tasks that you may do in the Firewall Cluster element properties. Other features, including routing and antispoofing, are explained together for both types of installations in separate feature-specific chapters.

## Benefits of Clustering

A Single Firewall can be a single point of failure. This can affect the availability of business critical applications and complicate the maintenance of the firewall equipment. Clustering firewall nodes can significantly reduce the risk of these problems.

The McAfee NGFW solution uses built-in clustering technology. No additional software or hardware is needed to cluster several nodes. If a node itself or the surrounding network equipment malfunctions, the other nodes in the cluster take over the traffic processing, minimizing any disruptions to the traffic flow. Similarly, maintenance is easier with a cluster, because individual nodes can be taken offline and even exchanged for new hardware without causing service outages.

Firewall Clusters also balance the load of traffic processing between the firewall nodes. You can flexibly add nodes to scale up the Firewall Cluster, improving the throughput and performance.

## Communication Between the Nodes

The Firewall Cluster nodes exchange information constantly. The state tables that list open connections (*state sync*) and the operating state of the other nodes (*heartbeat*) are exchanged. This exchange of information ensures that all nodes have the same information about the connections and that if a firewall node becomes unavailable, the other nodes of the cluster immediately notice this. The exchange of information between clustered Firewall nodes is synchronized through selected interfaces via a *heartbeat network* using multicast transmissions. The heartbeat messages are authenticated, and can also be encrypted if necessary (authentication is enabled by default).

## Hardware

The hardware the cluster nodes run on does not need to be identical. Different types of equipment can be used as long as all nodes have enough network interfaces for your configuration. Firewall Clusters can run on a McAfee NGFW appliance, on a standard server with an Intel-compatible processor, or as a virtual machine on a virtualization platform.

If equipment with different performance characteristics is clustered together, the load-balancing technology automatically distributes the load so that lower performance nodes handle less traffic than the higher performance nodes. However, when a node goes offline, the remaining node(s) must be able to handle all traffic on their own to ensure high availability. For this reason, it is usually best to cluster nodes with similar performance characteristics.

## Security Strength of Management Connections

You can optionally use 256-bit encryption for the connection between the engines and the Management Server. This requires both the engines and the Management Server to be version 5.5 or higher. When you create and use a new Internal ECDSA Certificate Authority to sign certificates for system communication, the Management Server and the engine re-establish their trust relationship and 256-bit encryption is enabled for the connection between the engines and the Management Server.

## Configuration of Firewall Clusters

---

Firewalls are configured and managed centrally through the Management Server. The Firewall Cluster element represents the Firewall Cluster's configuration on the Management Server. The main configuration options in the Firewall Cluster element include the settings related to network interfaces and clustering. This chapter concentrates on those settings.

### Load Balancing

In a Firewall Cluster 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.

The traffic arriving at the Firewall Cluster is balanced across the nodes according to the settings of the cluster's *load-balancing filter*. This filtering process distributes packets between the firewall nodes and keeps track of packet distribution. The Firewall determines the packet ownership of the nodes by comparing the incoming packet with node-specific values based on the packet headers. The load-balancing filter is pre-configured for optimal performance and is not meant to be adjusted independently by the system administrators.

The Firewall 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 does not usually affect the processing of the traffic.

## Standby Operation

In standby clustering, only one node at a time processes traffic, and other nodes wait on standby, ready to take over when the currently active node goes offline. Nodes that should not take over automatically can be set offline as usual. The drawback with standby mode is that there is no performance gain in clustering the firewalls.

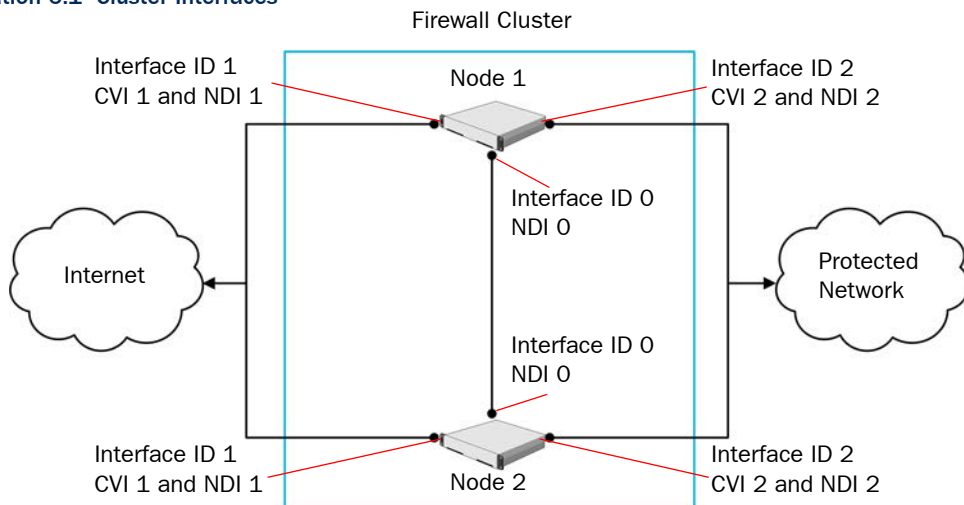
## Network Interfaces and IP Addresses

To work as replacements for each other, cluster members must have near-identical network interface configurations. A Physical Interface definition in the Management Client always represents a network interface definition on each node of the cluster. The table below explains the two types of IP addresses that you can add to a Physical Interface definition. You can add several IP addresses of each type to a single Physical Interface.

**Table 6.1 Firewall Cluster IP Address Types**

IP Address Type	Description
Cluster Virtual IP Address (CVI)	A CVI handles the traffic directed to the Firewall Cluster for inspection. The CVI is shared by all the nodes in the cluster. It depends on the selected clustering mode how this shared IP address is used. The CVI gives the cluster a single identity on the network, reducing the complexity of routing and network design.
Node Dedicated IP Address (NDI)	An NDI handles all communication for which the end-point is the node itself, including node-to-node, Management Server to node, and node-initiated connections. Each node in the cluster has its own dedicated IP address that is used as the NDI. In most cases, you must define an NDI for each physical interface. If necessary, you can define more than one NDIs for the same physical interface.

**Illustration 6.1 Cluster Interfaces**



The illustration above shows how CVIs and NDIs are used on a Firewall Cluster. In this example, the Interface ID 0 on each node has an NDI that is used for heartbeat traffic between the nodes in a dedicated network. There is no CVI on Interface ID 0, since it handles only node-to-node traffic. Interface ID 1 has a CVI that is used for Internet traffic (for example, web browsing), and it also has an NDI for traffic that the nodes send toward the Internet (for example, automatic tests the firewall uses to check connectivity). Interface ID 2 has a CVI for protected network traffic and an NDI for management connections to and from the nodes.

## Clustering Modes

There are several modes for how traffic can be directed to the cluster. The modes are explained in the table below. If necessary, refer to the documentation for the router, hub, or switch you are using for information on which mode is best in your environment.

**Table 6.2 Clustering Modes**

Mode	Description
Packet Dispatch	This is the recommended clustering mode. One node per physical interface is the dispatcher that handles the distribution of traffic between the different nodes for all CVIs on that physical interface. The assigned node handles the traffic processing. No additional switch configuration is needed. This mode can also be used with hubs but it is not the optimal clustering mode with hubs.
Unicast MAC	This is the recommended mode when hubs are used. This mode cannot be used with most switches. All the nodes in the cluster share the same unicast MAC address for the CVI. All the nodes in the cluster see all the packets.
Multicast MAC	The nodes share the same multicast MAC address for the CVI. All the nodes in the cluster see all the packets. Do not use this mode instead of the packet dispatch mode except in special cases (for example, if the nodes have uncertified network interface cards whose MAC address cannot be changed).
Multicast MAC with IGMP	The clustering works otherwise the same as in the Multicast MAC mode except that the engine answers to IGMP membership queries. This mode allows limiting multicast flooding when the switch does not support static MAC address forwarding tables.

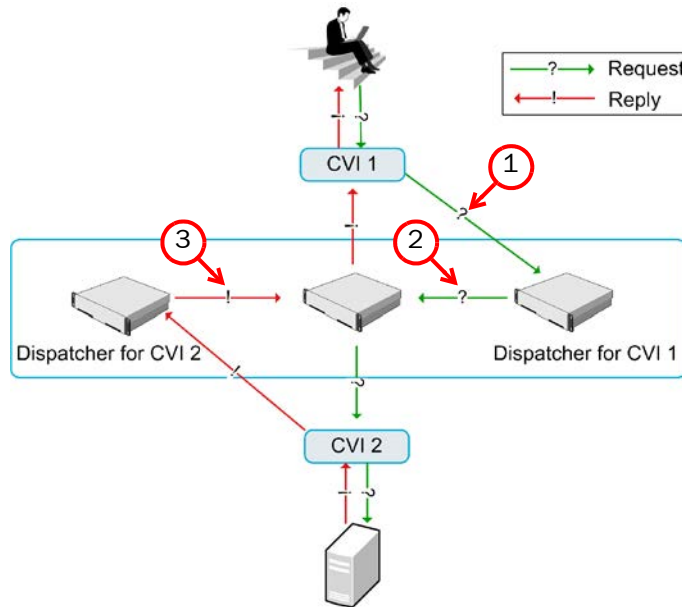
All CVIs on the same physical interface must use the same mode. It is possible to set different cluster modes for CVIs that are defined for different physical interfaces.

As Packet Dispatch mode is the recommended clustering mode, this section explains only the Packet Dispatch mode in more detail. For more information on the other clustering modes, see [Multicasting](#) (page 377).

# How Packet Dispatch Works

In Packet Dispatch mode, the node selected as the dispatcher on the physical interface assigns the packets to one of the nodes (to itself or to some other node). The assigned node then handles the actual resource-intensive traffic processing. The dispatcher attempts to balance the nodes' loads evenly, but assigns all packets that belong to the same connection to the same node. The node that acts as the packet dispatcher can be (and often is) different for CVIs on different physical interfaces. The illustration below shows an example of how packet dispatch handles a connection.

Illustration 6.2 Packet Dispatch CVI Mode



1. The dispatcher node for CVI 1 receives a new packet.
2. The dispatcher node either handles the packet itself or dispatches the packet to one of the other firewall nodes for processing according to the load-balancing filter. The packet is sent to the other node through the interface the packet arrived from.
3. The dispatcher node for CVI 2 forwards the replies within the open connection to the same node.

One node is responsible for handling each connection. The node responsible for the connection handles all resource-consuming tasks: it determines if the connection is allowed to continue, translates addresses as necessary, and logs the connection.

The dispatcher node controls the CVI's IP address and MAC address. The other nodes use their own physical interface's MAC address for the same CVI. When the dispatcher node goes offline, one of the other nodes becomes the dispatcher node. The new dispatcher node changes its interface's MAC address to the address defined for the Packet Dispatch CVI.



The network switch must update its address table without significant delay when the packet dispatcher MAC address is moved to another firewall node. This is a standard network addressing operation where the switch learns that the MAC address is located behind a different switch port. Then, the switch forwards traffic destined to the CVI address to this new packet dispatcher.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create a Firewall Cluster Element

To introduce a new Firewall Cluster to the SMC, you must define a Firewall Cluster element that stores the configuration information related to the firewall engines.

### Task 2: Create Physical Interfaces

Physical Interfaces represent the network ports on the engines. The number of defined Physical Interfaces can be lower than the number of network ports on the hardware. In a firewall cluster, a *Normal Interface* definition represents a single network port on all the nodes of the cluster. An *Aggregated Link* represents two network ports on one node that function as a single logical interface.

By default, the Physical Interface definitions are mapped to the actual network interfaces on the engines one to one. If necessary, you can change the mapping 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 interface that represents the same network interface on each node is correctly cabled to the same network.

### Task 3: Define VLAN Interfaces

A *Virtual Local Area Network* (VLAN) is a logical grouping of hosts and network devices that allows creating several separated networks on a single physical link. To allow this separation, the McAfee Firewall supports *VLAN tagging* as defined in the IEEE 802.1q standard. One network interface can support up to 4094 VLANs.

VLANs can also make it easier to deploy geographically distributed Firewall Clusters (for example, a cluster whose nodes are located in different buildings), as fewer physical interfaces and less cabling is needed.

When you create a VLAN interface, the CVI mode and MAC address are defined commonly for all virtual interfaces configured for the same Physical interface definition. The defined VLAN interfaces are displayed, for example, as “5.202” for network interface 5 with VLAN 202.

## Task 4: Define Tunnel Interfaces

*Tunnel Interfaces* are logical interfaces that are used as end-points for tunnels in the Route-Based VPN. Any traffic that is routed to a Tunnel Interface and allowed by the Access rules is automatically sent through the tunnel to the peer end-point defined in the Route-Based VPN. Defining Tunnel Interfaces is optional: Tunnel Interfaces are only used in the Route-Based VPN. See [Route-Based VPN Configuration](#) (page 295) for more information.

You can optionally add IPv4 and IPv6 addresses to a Tunnel Interface. Adding an IP address allows you to define the source IP address of traffic sent from the engine node itself.

## Task 5: Configure Physical or VLAN Interfaces

There are two types of IP addresses that you can define on Physical Interfaces or VLAN Interfaces:

- A *Cluster Virtual IP Address* (CVI) is used for traffic that is routed through the firewall for inspection. It is shared by all the nodes in the cluster.
- A *Node Dedicated IP Address* (NDI) is used for traffic that the nodes themselves send or receive (such as communication between the nodes and the Management Server or between the nodes in the cluster). Each node in the cluster has a specific IP address that is used as the Node Dedicated IP Address.

You can define more than one Cluster Virtual IP Address and/or Node Dedicated IP Address for the same physical interface or VLAN interface. If the Physical Interface is an Aggregated Link, both the interfaces that belong to the Aggregated Link share the same IP address definitions.

IPv6 addresses are supported on firewall clusters with dispatch clustering mode. IPv6 and IPv4 addresses can be used together on the same firewall cluster. Only IPv4 addresses are used in system communications.

Each CVI inherits the MAC address defined for the Physical Interface. The MAC/IP address pair remains the same all the time as only the location of the MAC address changes to the current dispatcher node (packet dispatch). This makes the external network equipment forward traffic to the correct node for dispatching. The CVIs on different Physical Interfaces cannot have duplicate MAC addresses.

The NDIs are used for node-to-node communications, for traffic between each individual node and the Management Server and Log Server, as well as communications with external components (such as authentication servers, or hosts that are probed in network connectivity tests). When you define NDIs, you must define both node-specific properties (such as the node's IP address) and properties that are shared by all the nodes in the cluster. All nodes must have the same netmask value for their NDI.

You also need at least one Node Dedicated IP Address. We recommend defining at least two NDIs (one for management connections and one for the heartbeat traffic between the nodes). Whenever the Firewall Cluster has a CVI address in a particular network segment, we recommend defining NDI addresses in the same network segment for each node. We recommend defining an NDI for all nodes on each Physical Interface, VLAN Interface, or Aggregated Link Interface pair. If there is a shortage of IP addresses, it is possible to leave some interface(s) without an NDI. If there is a CVI without a corresponding NDI from the same network segment, communications that require an NDI 'borrow' an IP address from another NDI on the same Physical Interface, VLAN Interface, or Aggregated Link interface pair. If there is no NDI on the same Physical Interface, VLAN Interface, or Aggregated Link interface pair, the

default IP address for outgoing traffic is used. The ‘borrowed’ IP address can be used without issues with routers that strictly follow the ARP standard. You may need to create a static ARP entry if some routers do not strictly follow the ARP standard.

If you want to configure multicast routing by using the firewall as an IGMP proxy, the IP addresses that you define for the downstream interfaces must be the lowest IP addresses among all the IGMP queriers in the local networks. For more information on IGMP-based multicast forwarding, see [Multicast Routing](#) (page 78).

## Task 6: Define Loopback IP Addresses

You can optionally define one or more *loopback IP addresses* for the firewall cluster. Loopback IP addresses allow you to assign IP addresses that do not belong to any directly-connected networks to the firewall. Loopback IP addresses are not connected to any physical interface and they do not create connectivity to any network. Any IP address that is not already used as a Cluster Virtual IP Addresses (CVI) or Node Dedicated IP Addresses (NDI) on another interface can be used as a loopback IP address. The same IP address can be used as a loopback IP address and as the IP address of a Tunnel Interface.

You can define both CVI and NDI loopback IP addresses for Firewall Clusters. Whether to define a CVI or NDI loopback address depends on the traffic for which the loopback IP address is used:

- A CVI loopback IP address is used for loopback traffic that is sent to the whole cluster. It is shared by all the nodes in the cluster.
- An NDI loopback IP address is used for loopback traffic that is sent to a specific node in the cluster.

## Task 7: Install the Firewall Engines

During the engine installation, you map the network interfaces on the engine to the Physical Interface definitions created in the Management Client. The mapping of Tunnel Interfaces to physical network interfaces on the engine is done automatically by the engine operating system based on the routing configuration. See the *Appliance Installation Guide* delivered with the appliance for more information.

During the installation, a basic initial configuration is activated and an initial contact between the Management Server and each engine is initiated. During the initial contact, each engine authenticates itself to the Management Server with its own single-use password. When this initial contact succeeds, the engine receives a certificate from the SMC for authenticating all subsequent communications - a trust relationship between that engine and the Management Server is established.

## Task 8: Install a Firewall Policy

The engines do not receive any clustering settings until the first time you install a policy on them and the working configuration is received from the Management Server. After the firewall engines have made initial contact with the Management Server, only the interface used for the control connection with the Management Server is configured. You must install a Firewall Policy using the Management Client to transfer the complete configuration to the firewall.

# Using a Firewall Cluster

The main points of Firewall Cluster configuration are explained in the preceding sections of this chapter. This section illustrates some additional concepts for working with Firewall Clusters:

- [Internal DHCP Server](#)
- [Node State Synchronization](#)
- [Security Level for State Synchronization](#) (page 61)
- [Manual Load Balancing](#) (page 61)



**Caution** – Do not modify the firewall's Advanced Settings without due consideration. An invalid configuration of the parameters may lead to system instability or malfunction.

## Internal DHCP Server

Firewall clusters contain an internal DHCP server that can be used to assign IPv4 addresses to hosts in the protected network. This is meant for small installations where it may be more practical to assign the IP addresses using the firewall rather than relay the DHCP requests from a separately maintained local DHCP server or from some other site's DHCP server through a VPN.

## Node State Synchronization

State synchronization is essential for the following features:

- Dynamic load balancing.
- Transparent switchover of nodes in case of failure or maintenance.
- Handling of related connections when a service (for example, FTP) opens multiple connections.

Regular, timer-launched synchronization events are needed to synchronize state data and to avoid cutting connections in case of node failure. Timed synchronization events are divided into full and incremental sync messages (see [Table 6.3](#) for details).

**Table 6.3** Sync Messages

Type	Explanation
Full Sync Messages	Contain all connection data about the traffic handled by a node at the time when the message was sent. When new data is received, it replaces the existing data. Full sync requires more bandwidth and processing time.
Incremental Sync Messages	Contain only data on connections that were created or changed since the last full or incremental sync message. Incremental sync needs less bandwidth and processing time. Since the incremental changes are sent only once, the system may lose connections if the data is lost. While able to produce accurate data with frequent updates, incremental sync requires full sync to provide reliable synchronization data.

By default, a combination of full and incremental sync messages is exchanged between nodes. This way, frequent updates on incremental changes and recurrent reports on existing connections are combined.

In cases where synchronization of connection information between nodes is causing a disturbance to specific traffic, you can optionally disable synchronization for the traffic using rule options in the Policy. Disabling synchronization reduces the traffic volume on the active heartbeat interface, but it also prevents transparent failover of connections to other nodes.

## Security Level for State Synchronization

Because synchronization controls the inter-node traffic within a heartbeat network, you must ensure the security of the heartbeat and synchronization data. The inter-node traffic can be authenticated, or both authenticated and encrypted. Inter-node traffic can also optionally be sent without authentication or encryption. However, this makes it possible to both sniff synchronization data and send fraudulent messages to open connections.



**Note – Independent of the security level, all critical information such as passwords and encryption keys are protected. They are never sent in plaintext.**

## Manual Load Balancing

The Firewall Cluster's load-balancing filter can be manually modified if there is a specific need for modifications. Any modified load-balancing parameters are combined with the automatically created filtering entries. However, modifying the load-balancing parameters of the Firewall Cluster without careful consideration can cause conflicts in filtering decisions.

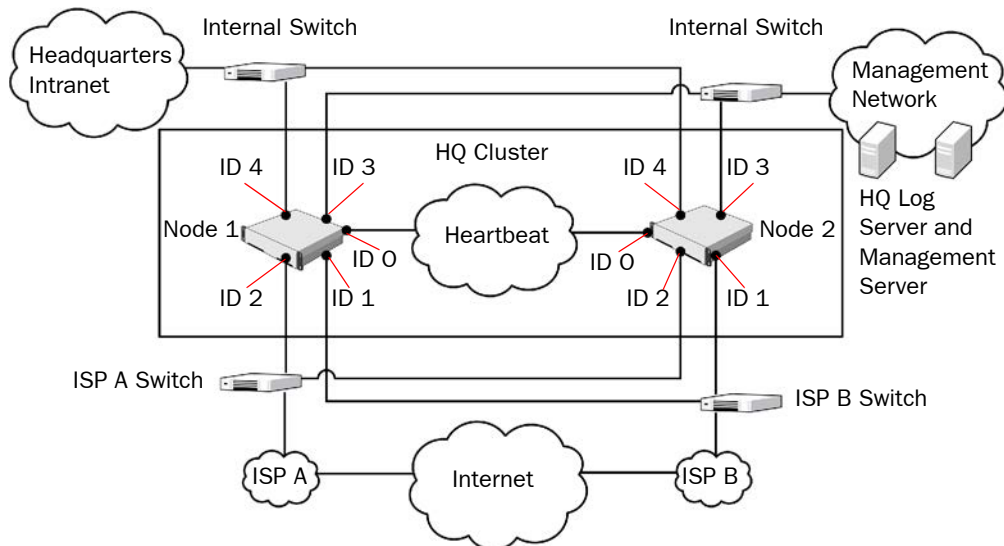
# Examples of Firewall Cluster Deployment

The examples in this section illustrate the configuration of a Firewall Cluster with general steps on how each scenario is configured.

## Setting up a Firewall Cluster

The administrators at the headquarters of Company A want to set up a Firewall Cluster. The cluster consists of two cluster nodes: Node 1 and Node 2. The HQ Cluster Firewall has a dedicated heartbeat network (10.42.1.0/24), and it is connected to two internal networks: Headquarters Intranet (172.16.1.0/24) and Management Network (192.168.10.0/24). It uses Multi-Link to ISP A and ISP B for its connection to the Internet.

**Illustration 6.3 Headquarters Network**



The administrators:

1. Create a Firewall Cluster element (HQ Cluster) and define HQ Log as its Log Server.
2. Define the physical interfaces 0-4.
3. Define the CVIs and NDIs for the physical interfaces. Except for the IP addresses, the node-specific properties for Node 1 and Node 2 are the same. See [Table 6.4](#).

**Table 6.4 Cluster Interfaces**

Interface ID	Type	IP Address	Comment
0	NDI for Node1	10.42.1.1	Heartbeat
0	NDI for Node2	10.42.1.2	Heartbeat
1	CVI	129.40.1.254	ISP B
1	NDI for Node1	129.40.1.21	ISP B
1	NDI for Node2	129.40.1.22	ISP B
2	CVI	212.20.1.254	ISP A
2	NDI for Node1	212.20.1.21	ISP A
2	NDI for Node2	212.20.1.22	ISP A
3	CVI	192.168.10.1	Management Network
3	NDI for Node1	192.168.10.21	Management Network
3	NDI for Node2	192.168.10.22	Management Network
4	CVI	172.16.1.1	Headquarters Intranet
4	NDI for Node1	172.16.1.21	Headquarters Intranet
4	NDI for Node2	172.16.1.22	Headquarters Intranet

4. Save the initial configuration of the engines in the Management Client.
5. Map the interface identifiers in the configuration to the physical interfaces on each engine's command line and establish contact between each engine and the Management Server.
6. Install a Firewall Policy on the Firewall Cluster in the Management Client to transfer the working configuration to the firewall engines. The nodes exchange authentication information and begin to work as a cluster.

## Adding a Node to a Firewall Cluster

Company A's Firewall currently consists of two nodes. However, the load on the Firewall is exceptionally high, so the administrator has decided to add another node to ensure continuity of network services even when one of the nodes is offline. The administrator does the following:

1. Adds a third node in the Firewall Cluster element's properties.
2. Defines the node-specific IP addresses for the NDI interfaces of the new node.
  - The cluster-level interface configuration does not need adjustments, since it is shared by all nodes.
3. Installs the new engine and performs the initial configuration.
4. Refreshes the security policy of the Firewall Cluster.





## CHAPTER 7

# MASTER ENGINE AND VIRTUAL FIREWALL CONFIGURATION

A *Virtual Security Engine* is a logically-separate engine that runs as a virtual engine instance on a physical engine device. A *Virtual Firewall* is a Virtual Security Engine in the Firewall/VPN role. A *Master Engine* is a physical engine device that provides resources for Virtual Security Engines.

The following sections are included:

- ▶ [Overview to Master Engine and Virtual Firewall Configuration](#) (page 66)
- ▶ [Configuration of Master Engines and Virtual Firewalls](#) (page 66)
- ▶ [Using Master Engines and Virtual Firewalls](#) (page 69)
- ▶ [Example of Master Engine and Virtual Firewall Deployment](#) (page 70)

# Overview to Master Engine and Virtual Firewall Configuration

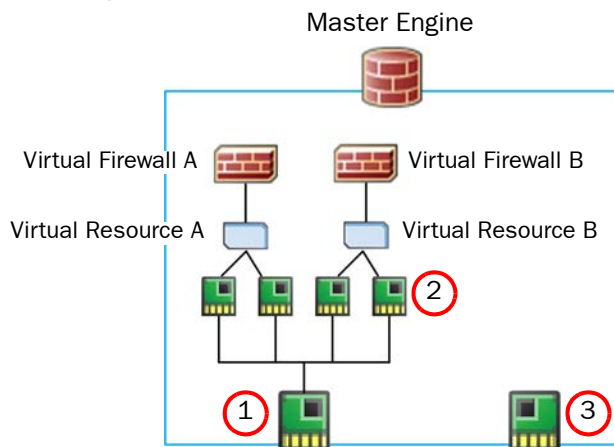
This chapter focuses on Virtual Security Engines in the Firewall/VPN role. Virtual Security Engines in the Firewall/VPN role are configured using Virtual Firewall elements in the Management Client.

Using Virtual Firewalls allows the same physical engine device to support multiple policies or routing tables, or policies that involve overlapping IP addresses. This is especially useful in a Managed Security Service Provider (MSSP) environment, or in a network environment that requires strict isolation between networks.

## Configuration of Master Engines and Virtual Firewalls

Any Security Engine that has a license that allows the creation of Virtual Resources can be used as a Master Engine.

**Illustration 7.1 Master Engine and Virtual Firewall Architecture**



One physical Master Engine can host multiple Virtual Firewalls. A *Virtual Resource* element defines the set of resources on the Master Engine that are allocated to a Virtual Firewall. Virtual Resource elements associate Virtual Firewalls with Physical Interfaces or VLAN Interfaces on the Master Engine. The license for the Master Engine defines how many Virtual Resources can be created. The number of Virtual Resources limits the number of Virtual Firewalls: one Virtual Firewall at a time can be associated with each Virtual Resource.

Master Engines can have two types of interfaces: interfaces for the Master Engine's own traffic, and interfaces that are used by the Virtual Firewalls hosted on the Master Engine. In the example above, the Master Engine has the following kinds of interfaces:

1. Physical Interface for hosted Virtual Firewall traffic.
2. VLAN Interfaces for hosted Virtual Firewall traffic.
3. Physical Interface for the Master Engine's own traffic.

# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Create a Master Engine Element

To introduce a new Master Engine to the SMC, you must define a Master Engine element that stores the configuration information related to the Master Engine and the Virtual Resources.

By default, Master Engine elements are created with two nodes. You can optionally add or remove nodes. Each Master Engine can consist of 1 to 16 nodes.

## Task 2: Create Virtual Resource Element(s)

Virtual Resource elements associate Virtual Firewalls with Physical Interfaces or VLAN Interfaces on the Master Engine. When you select the same Virtual Resource for a Physical Interface or VLAN Interface on the Master Engine and for a Virtual Firewall, the Virtual Firewall is automatically associated with the Master Engine.

## Task 3: Configure Master Engine Interfaces

You can add Physical Interfaces and VLAN Interfaces to a Master Engine. If you want to use a Physical Interface or VLAN Interface to host a Virtual Firewall, you must select a Virtual Resource for the interface. The same Virtual Resource can be used on more than one Master Engine interface to allocate multiple interfaces to the same Virtual Firewall. If you want the Virtual Firewall to have multiple interfaces, you must use the same Virtual Resource on more than one Master Engine interface.

If you want to use a Physical Interface or VLAN Interface for the Master Engine's own traffic, you must not select a Virtual Resource for the interface. You can configure several IPv4 addresses on each Physical Interface or VLAN Interface that does not have a Virtual Resource associated with it. Because the interfaces for the Master Engine's own traffic are only used for system communications, only IPv4 addresses are supported on interfaces for the Master Engine's own traffic.

By default, the Physical Interface definitions for the Master Engine are mapped to the actual network interfaces on the Master Engine hardware in numerical order. If necessary, you can change the mapping using command line tools on the Master Engine. This mapping can be done differently from one Master Engine node to another as long as you take care that the interface that represents the same network interface on each Master Engine node is correctly cabled to the same network.

## Task 4: Create a Virtual Firewall Element

Virtual Firewall elements store the configuration information related to the Virtual Firewalls. Selecting a Virtual Resource for the Virtual Firewall automatically associates the Virtual Firewall with the Master Engine where the Virtual Resource is used.

## Task 5: Configure Virtual Firewall Interfaces

Physical Interfaces in the properties of a Virtual Firewall represent interfaces allocated to the Virtual Firewall in the Master Engine. All communication between Virtual Firewalls and the SMC is proxied by the Master Engine.

Physical Interfaces for the Virtual Firewall are automatically created based on the interface configuration in the Master Engine properties. The number of Physical Interfaces depends on the number of interfaces allocated to the Virtual Firewall in the Master Engine. You can optionally modify the automatically-created Physical Interfaces.

In addition to the automatically-created Physical Interfaces, you can add the following types of interfaces to Virtual Firewalls:

- You can add VLAN Interfaces if the creation of VLAN Interfaces for Virtual Firewalls is enabled in the Master Engine Properties.
- You can optionally add Tunnel Interfaces for the Route-Based VPN.

Both IPv4 and IPv6 addresses are supported on Virtual Firewalls. You can define one or more static IP addresses for Virtual Firewall interfaces.

You can optionally add loopback IP addresses to the Virtual Firewall. Loopback IP addresses allow you to assign IP addresses that do not belong to any directly-connected networks to the Virtual Firewall. Loopback IP addresses are not connected to any physical interface and they do not create connectivity to any network. Any IP address that is not already used on another Physical or VLAN interface in the same Virtual Firewall can be used as a loopback IP address. The same IP address can be used as a loopback IP address and as the IP address of a Tunnel Interface. Loopback IP addresses can be used as the Identity for Authentication Requests, the Source for Authentication Requests, and the Default IP Address for Outgoing Traffic.

By default, the interface definitions for the Virtual Firewall are mapped to interfaces on the Master Engine in the order in which the interfaces are created on the Master Engine.

## Task 6: Install a Firewall Policy

After you have modified Physical Interfaces or VLAN Interfaces in the Master Engine properties, you must install or refresh the policy on the Master Engine to transfer changes. After you have modified Physical Interfaces or VLAN Interfaces in the Virtual Firewall properties, you must install or refresh the policy on the Virtual Firewall to transfer the changes.

There is no separate type of policy for Master Engines and Virtual Firewalls. Master Engines and Virtual Firewalls use Firewall Policies. You can install different Firewall Policies on the Master Engine and on the Virtual Firewalls. You can also install a different Firewall Policy on each Virtual Firewall.

## Using Master Engines and Virtual Firewalls

---

The main points of Master Engine and Virtual Firewall configuration are explained in the preceding sections of this chapter. This section illustrates some additional concepts for working with Master Engines and Virtual Firewalls:

- [Moving a Virtual Firewall to a Different Master Engine](#)
- [Using Master Engines and Virtual Firewalls With Domains](#)

### Moving a Virtual Firewall to a Different Master Engine

The Virtual Resource selected in the properties of a Virtual Firewall element determines the Master Engine to which the Virtual Firewall belongs. To move a Virtual Firewall to a different Master Engine, you select a Virtual Resource that is associated with a different Master Engine in the Virtual Firewall properties. The move becomes effective when you refresh the policy on the Master Engine.

### Using Master Engines and Virtual Firewalls With Domains

If there are multiple administrative Domains, Virtual Firewalls associated with the same Master Engine can belong to different Domains. However, the Master Engine must either belong to the Shared Domain or to the same Domain as the associated Virtual Firewalls. For example, the Master Engine can belong to the Shared Domain, while each associated Virtual Firewall belongs to a different Domain.

# Example of Master Engine and Virtual Firewall Deployment

The example in this section illustrates the configuration of Master Engines and Virtual Firewalls with general steps on how the scenario is configured.

## Deploying Virtual Firewalls for MSSP Customers

Company A is an MSSP (Managed Security Services Provider). Customer 1 and Customer 2 are customers of Company A. The customers each want one Virtual Firewall with two Physical Interfaces. The administrators at Company A decide to use their existing NGFW appliance as a Master Engine to host Virtual Firewalls for Customer 1 and Customer 2. Separate administrative Domains have already been configured for each customer. The engine already has a license that allows the creation of Virtual Resources.

The administrators at Company A:

- 1. Create a Master Engine element in the Shared Domain.
- 2. Create one Virtual Resource element for each customer’s Virtual Firewall and select the appropriate Domain for each Virtual Resource:

Virtual Resource Name	Domain
Customer 1 Virtual Resource	Customer 1 Domain
Customer 2 Virtual Resource	Customer 2 Domain

- 3. Create the following Physical Interfaces on the Master Engine:

Interface ID	Description
0	Physical Interface for the Master Engine’s own traffic.
1	Physical Interface for hosted Virtual Firewall traffic.

- 4. Add an IPv4 address for each Master Engine node to Physical Interface 0.

5. Add the following VLAN Interfaces to Physical Interface 1 and select the appropriate Virtual Resource for each VLAN Interface:

Interface ID	Virtual Resource	Description
VLAN 1.1	Customer 1 Virtual Resource	VLAN Interface for the first Physical Interface on the Virtual Firewall for Customer 1.
VLAN 1.2	Customer 1 Virtual Resource	VLAN Interface for the second Physical Interface on the Virtual Firewall for Customer 1.
VLAN 1.3	Customer 2 Virtual Resource	VLAN Interface for the first Physical Interface on the Virtual Firewall for Customer 2.
VLAN 1.4	Customer 2 Virtual Resource	VLAN Interface for the second Physical Interface on the Virtual Firewall for Customer 2.

6. Create a Virtual Firewall element for each customer and select the appropriate Virtual Resource for each Virtual Firewall:

Virtual Firewall	Virtual Resource
Customer 1 Virtual Firewall	Customer 1 Virtual Resource
Customer 2 Virtual Firewall	Customer 2 Virtual Resource

7. Add IP addresses to the Physical Interfaces on the Virtual Firewalls.  
8. Refresh the policy on the Master Engine.  
9. Refresh the policy on the Virtual Firewalls.





## CHAPTER 8

# ROUTING AND ANTISPOOFING

Routing is the process of defining through which network interface the engine forwards traffic from a source address to a destination address. Antispoofing is the process of defining which addresses are considered valid source addresses for the network(s) connected to each interface.

The following sections are included:

- ▶ [Overview to Routing and Antispoofing](#) (page 74)
- ▶ [Configuration of Routing and Antispoofing](#) (page 74)
- ▶ [Using Routing and Antispoofing](#) (page 78)
- ▶ [Examples of Routing](#) (page 79)

## Overview to Routing and Antispoofing

---

In addition to examining packets, Firewalls, Master Engines, and Virtual Firewalls must determine how to route packets. For the most part, the SMC automates the routing and antispoofing configuration. Much of the configuration is generated automatically based on the IP addresses of the network interfaces. Routing and antispoofing for Master Engines are configured in the same way as routing and antispoofing for Firewall Clusters. Routing and antispoofing for Virtual Firewalls are configured in the same way as routing and antispoofing for Single Firewalls.

## Configuration of Routing and Antispoofing

---

The routing and antispoofing information is displayed and configured graphically in interface-based trees in the Routing view and Antispoofing view. The routing information is stored on the Management Server. The information is transferred to the engines when the policies are installed or refreshed.

In addition to the routing information that is automatically generated based on the interface definitions, you must manually add any networks that are not directly connected to the engine to the routing tree. Similarly, you must also add a default route for packets whose destination IP address is not included anywhere else in the routing tree.

IP address *spoofing* is an attack where the source IP address in a packet is modified to gain unauthorized access or to cause a *denial-of-service* (DoS). Such attacks can be prevented with antispoofing rules. The antispoofing configuration is generated automatically based on the routing tree. Antispoofing is always enforced on all interfaces. You can change the antispoofing configuration, but in most environments, there is no need to do so.

New Networks are automatically added to routing when you change the engine's interface properties. However, none of the Networks are automatically removed, so you must check your Routing view for obsolete entries and clear them manually. This is to prevent the system from removing currently unused route definitions that you may want to reuse or keep for easy rollback to previous definitions. All additions and deletions in the Routing view are automatically reflected in the Antispoofing view. Manual definitions in the Antispoofing view are preserved regardless of routing changes.

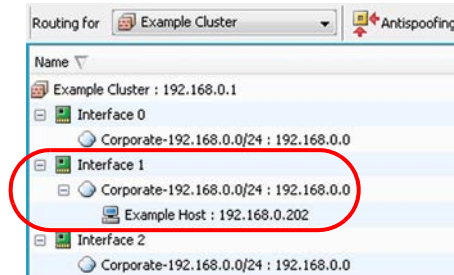
## Reading the Routing and Antispoofing Trees

The Routing view automatically displays the interfaces and a Network element for each network that is directly connected to the engine. The Network is created based on the IP address(es) that you define for each interface. Interfaces that belong to an aggregated link always share the same routing information. Only the first interface that belongs to the aggregated link is displayed in the Routing and Antispoofing views.

When the engine reads routing definitions, it selects the most specific route and antispoofing definition it finds for each packet. The engine first checks if there is a route defined for the specific destination IP address of the packet (a Host element), then checks routes to the defined networks (a Network element), and finally uses the default route (the Any Network element) if none of the other routes match the packet's destination address. If there are overlapping definitions, the more specific one is considered first.

**Example** Interface 1 has a Host element for 192.168.0.202 and Interface 2 has a Network element for 192.168.0.0/24, a packet to 192.168.0.202 is routed through Interface 1, and the Interface 2 definition is not considered for those packets at all.

**Illustration 8.1** Example: The More Specific Destination is Considered First in Routing

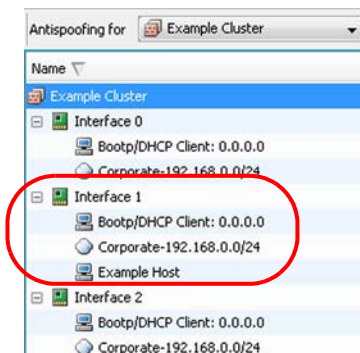


Interface 1 is always used for a destination of 192.168.0.202 because it has a Host element with the most specific address under it.

The Antispoofing view defines the source addresses of traffic that are considered valid (*not* spoofed) for each interface. If an interface receives a packet with a source address that is not a valid address for the network(s) connected to that interface, the packet is discarded. This is the case, for example, when an external interface receives a packet with an internal source. The engine selects the most specific antispoofing definition it finds for each packet.

**Example** In the Antispoofing tree automatically generated based on the routing example above, antispoofing discards any traffic with the address 192.168.0.202 if it originates from Interface 2, because it has the less specific definition for that address (the Network 192.168.0.0/24).

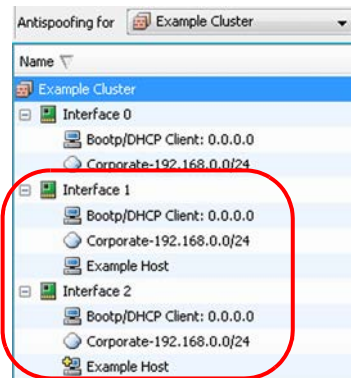
**Illustration 8.2** Only The Most Specific Destination is Considered Valid in Antispoofing



Packets from Example Host (192.168.0.202) are only considered valid if they originate from Interface 1, because it has the most specific route to the host's address.

**Example** To allow the traffic to originate from both interfaces, you would also have to add the Host element for 192.168.0.202 under Interface 2, so that the definitions are equally specific (both interfaces contain the Host element) and therefore both definitions are valid at the same time.

**Illustration 8.3 Both Interfaces are Considered Valid**

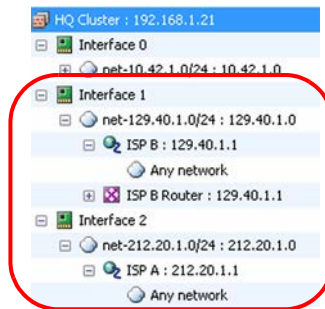


Both Interface 1 and Interface 2 are considered valid routes to Example Host (192.168.0.202) because the Host element is included under both interfaces.

## Multi-Link Routing

Multi-Link uses multiple network links (*NetLinks*) to balance the load of outbound traffic and ensure high availability of Internet connectivity. With each new outbound connection, the system selects the fastest route for the connection from the available NetLinks. Multi-Link routing can be used for both IPv4 and IPv6 traffic. See [Outbound Traffic Management](#) (page 225) for more information on Multi-Link for outbound connections.

**Illustration 8.4 NetLinks in Routing View**



In the illustration above, a Multi-Link configuration is used to define a default route to the Internet (to “Any network”) through the ISP A and ISP B NetLinks. Separate network interfaces can be used for each NetLink, which is recommended for the Multi-Link configuration. It is also possible to configure multiple NetLinks for a single network interface. However, this is not recommended, as configuring multiple networks on the same interface introduces a single point of failure.

For each NetLink, a range of IP addresses is defined for NATing the source IP address of an outbound connection that goes through the NetLink. A NAT rule in the Firewall Policy defines the Outbound Multi-Link element that is used for Multi-Link outbound connections.

## Default Elements

Networks that correspond to the IP addresses of each interface are automatically added to the routing and antispoofing configurations of Firewall, Master Engine, and Virtual Firewall elements.

The system includes a default network element called *Any network*, which is needed to define the default route. The system also includes a Host element for Bootp/DHCP clients in the Antispoofing view for Firewall, Master Engine, and Virtual Firewall elements.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Add Router or NetLink

A Router or a NetLink element represents a next-hop gateway that forwards packets to network(s) that are not directly connected to the engine.

You can use a single Router if a single route is enough for routing traffic to a network through an interface or an aggregated link. If you want to create separate routes for traffic to a network through two or more interfaces, you must use NetLinks. See [Outbound Traffic Management](#) (page 225) for more information on NetLinks and Multi-Link.

Tunnel Interfaces for the Route-Based VPN do not use Router or NetLink elements. Instead, networks that are reachable through the VPN tunnel are added directly to the Tunnel Interface as if they were directly connected networks.

### Task 2: Add Network(s)

A Network element represents the IP addresses of a network or a subnetwork to which the Router or the NetLink forwards the traffic, or the IP addresses of a network that is reachable through the VPN tunnel of the Route-Based VPN.

### Task 3: Refresh Firewall Policy

Changes to the routing or antispoofing configuration are only transferred to the engine when you refresh the Firewall Policy. When you change the routing or antispoofing for a Master Engine, you must refresh the policy on the Master Engine. When you change the routing or antispoofing for a Virtual Firewall, you must refresh the policy on the Virtual Firewall.

## Policy Routing

With *policy routing* you can route traffic through a selected gateway. These policy routing entries override other routing configuration defined in the Routing view.

A policy routing entry includes a source IP address, a destination IP address, netmasks or prefixes for the source and destination addresses, and the IP address of the selected gateway. Only IPv4 addresses are supported in policy routing.

Antispoofing configuration may need to be changed accordingly when using manual policy routing entries.

## Multicast Routing

IP multicasting is the transmission of an IP datagram to all hosts in a multicast host group, which is identified by a single destination IP address. See [Multicasting](#) (page 377) for more information. You can configure *static multicast* and *IGMP-based multicast forwarding* of IPv4 traffic. You can also configure *dynamic multicast* routing using the Quagga Routing Suite on the engine command line. See [NGFW Engine Commands](#) (page 319) for more information.

Static multicast does not rely on IGMP (Internet Group Management Protocol) messaging. Because of the static nature of the configuration, static multicast is suitable for enduring configurations, such as mutually agreed multicast traffic between organizations.

In IGMP-based multicast forwarding (IGMP proxying), the firewall maintains a list of subscriptions to the multicast host group and forwards multicast traffic to the subscribed hosts. The firewall periodically queries the downstream networks for hosts that want to join the multicast host group. The firewall also processes unsolicited IGMP join/leave requests received from downstream networks. As multicast traffic is only sent to the currently subscribed hosts, IGMP-based multicast forwarding can save bandwidth and provide faster service. IGMP-based multicast forwarding is only supported in tree topology networks. See RFC 4605 for more information.

If you use Multi-Link together with IGMP-based multicast forwarding, make sure that you do not create routing loops. If you add a NetLink to an engine's upstream interface, do not add a NetLink to any of the engine's downstream interfaces.

## Modifying Antispoofing

In rare cases you may need to modify the default antispoofing definitions to make exceptions to the antispoofing rules (for example, if you have defined policy routing manually). You can also disable antispoofing for an interface, if necessary.

## Monitoring Routing

You can monitor the currently active routing configuration of an engine in the Routing Monitoring view. The routing configuration can be viewed as a table or diagram. You can also save versions of the routing configurations as snapshots in the Routing Monitoring view.

## Examples of Routing

---

The examples in this section illustrate some common uses of routing and general steps on how each scenario is configured.

### Routing Traffic with Two Interfaces

Company A needs to route traffic to the Internet as well as to the internal Network B, which is not directly connected to the company's Branch Office Firewall. The company's administrators decide to create a separate route to the internal Network B and a default route for traffic to the Internet. The administrators:

1. Open the Routing view for the Branch Office Firewall.
2. Add a Router and Network B under Interface 0.
3. Add a Router and the default element "Any network" under Interface 1.
4. Refresh the Firewall Policy on the Branch Office Firewall.

### Routing Internet Traffic with Multi-Link

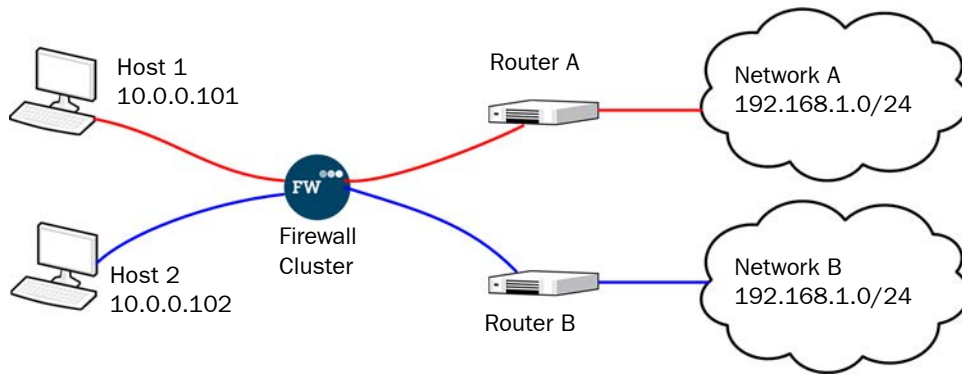
Company B wants to ensure high availability of Internet connections through the company's firewall. The company's administrators decide to use Multi-Link routing with two NetLinks to balance Internet connections. They:

1. Create two NetLinks.
2. Combine the NetLinks into an Outbound Multi-Link element to balance the connections.
3. Add one of the NetLinks under Interface 1 and the other NetLink under Interface 2 in the Routing view.
4. Add the default route "Any network" under the NetLinks.
5. Add a NAT rule to the Firewall Policy to balance the connections between the NetLinks.
6. Refresh the Firewall Policy.

# Routing Traffic to Networks That Use Same Address Space

Company C's network is connected to two partners: Network A and Network B. The Network A and the Network B partners use the same address space in their internal networks.

Illustration 8.5 Policy Routing in Company C



There are two hosts in Company C's network. Host 1 works only with the Network A partner and Host 2 only with the Network B partner. The administrators at Company C decide to use policy routing to route the traffic between Company C's network and the two partner sites. The administrators:

1. Create policy routing entries for Host 1 and Host 2 on the Firewall HQ Cluster as shown in [Illustration 8.6](#).

Illustration 8.6 Policy Routing Entries for Host 1 and Host 2

The screenshot shows a window titled 'Policy Routing' with a sub-tab 'IPv4 Policy Routes'. It contains a table with the following data:

Source IP Address	Source Netmask	Destination IP Address	Destination Netmask	Gateway IP Address	Comment
10.0.0.101	255.255.255.255	190.168.1.0	255.255.255.255	190.168.3.1	
10.0.0.102	255.255.255.255	190.168.1.0	255.255.255.255	172.16.1.1	

On the right side of the table, there are 'Up' and 'Down' buttons with arrows.

2. Modify the antispoofing rules so that they take into account the routing defined with the policy routing entries.
3. Refresh the Firewall Policy on the Firewall HQ Cluster.



# ACCESS CONTROL POLICIES

---

## In this section:

**Firewall Policies - 83**

**Access Rules - 101**

**Inspection Policies - 119**

**Network Address Translation (NAT) - 131**

**Protocol Agents - 145**

**TLS Inspection - 155**

**URL Filtering - 163**

**Spam Filtering - 167**

**Virus Scanning - 171**

**External Content Inspection - 175**

**Situations - 183**

**Applications - 191**

**Blacklisting - 195**



## CHAPTER 9

# FIREWALL POLICIES

Policy elements are containers for the lists of rules that determine how the Firewall, Master Engine, or Virtual Firewall examines traffic. The policy elements include Firewall Template Policies, Firewall Policies, Firewall Sub-Policies, and Inspection Policies.

The following sections are included:

- ▶ [Overview to Firewall Policies](#) (page 84)
- ▶ [Configuration of Policy Elements](#) (page 87)
- ▶ [Using Policy Elements and Rules](#) (page 94)
- ▶ [Examples of Policy Element Use](#) (page 98)

# Overview to Firewall Policies

---

The policy elements store rules according to which the engine examines the traffic. This chapter introduces how these elements are used by Firewall engines, Master Engines, and Virtual Firewalls. It also explains how to 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 separate chapters. See [Access Rules](#) (page 101), [Inspection Policies](#) (page 119), and [Network Address Translation \(NAT\)](#) (page 131).



**Note** – There is no separate type of policy for Master Engines and Virtual Firewalls. Master Engines and Virtual Firewalls use the same policies as Firewalls (Firewall Policies and Inspection Policies).

## Policy Hierarchy

The policy structure is a hierarchy 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 firewall.
- Make policies easier to read.

The template and policy hierarchy is flattened when the Firewall Policy is transferred to the engine, so the policy looks the same to the engine regardless of how it is organized on the Management Server (as long as the rules themselves are in the same order). You can also create sections of conditional rules that you can insert into the other policy elements. The engine 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 very simple and you do not need the benefits outlined above, you can create a very simple policy hierarchy. You can, for example, start with one Firewall Policy built on the provided Firewall Template. The same Firewall Policy can be used on more than one engine.

## How the Engine Examines Traffic

A Firewall, Master Engine, or Virtual Firewall passes through only traffic that is specifically allowed in the Firewall Policy. All other traffic is discarded. All connections are handled in exactly the same way, even connections that the engine itself opens, and the management connections that the engine is intended to receive.

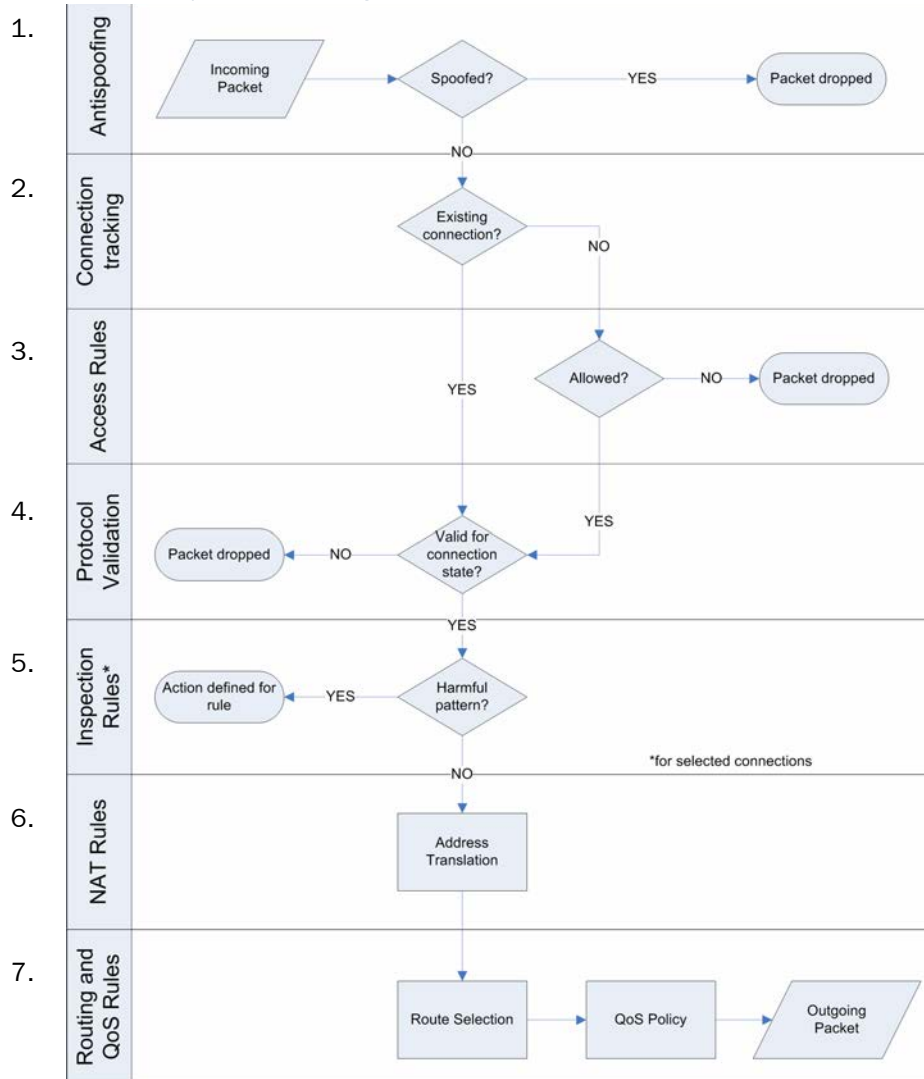
On Firewall Clusters and clustered Master Engines, the load-balancing filter first determines which node in the cluster actually processes the received packet. The processing then begins on the selected node.

Some clearly broken packets are dropped before any rule processing; the engine drops packets that contain no data, and ICMP error messages that are missing key information about the broken packet. You must activate diagnostic logging for packet filtering to log these invalid packets.

The engine checks a new connection against the policy rule by rule. The header on each packet arriving on an interface is examined for the source and destination IP address, and protocol-related information, such as the port. The authentication status of the user attempting a connection and the current date and time may also be included as parameters in the examination process.

The basic packet handling process is described in [Illustration 9.1](#).

**Illustration 9.1 Connection/Package Handling Process**



1. The engine checks that the traffic is coming in through the correct interface as defined in the Routing and Antispoofing trees.
2. The engine 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).

3. If the packet is not part of an existing connection, the packet is compared with the Access rules in the installed Firewall Policy. The processing continues until the packet matches a rule that tells the engine to allow or stop the packet.
  - If there is no rule match anywhere else in the policy, the packet is discarded when the engine reaches the end of the Firewall Policy.
4. If the packet is allowed as an existing connection or in an Access rule, the engine checks that the packet is valid for the state of the connection. If not, the packet is dropped.
  - For example, a TCP connection must always begin with a SYN packet (as defined in the protocol standards), so the engine checks that the first packet of a new connection is a valid SYN packet.
5. The engine applies Inspection rules to connections that are selected for deep packet inspection in the Access rules.
  - Inspection applies to all packets in a connection, so the Inspection rules are applied even if the packet is a part of an existing connection.
  - The Inspection rules are used to look for harmful patterns hiding in the legitimate-looking connections (that is, payload of packets that are a part of allowed connections).
6. Network Address Translation (NAT) rules are applied to IPv4 and IPv6 connections. The source and destination addresses are translated according to the first matching NAT rule (or not done at all, if a NAT rule so defines). If none of the NAT rules match, the packet continues with the original addresses.
7. A routing decision is made (using the translated address). If the destination of the packet is changed by a NAT operation, the packet is checked against the Access rules again. If the packet is still allowed by an Access rule, the packet is let through the engine according to its priority and any bandwidth limits or guarantees that may have been defined. If the packet no longer matches an Access rule, the packet is dropped.

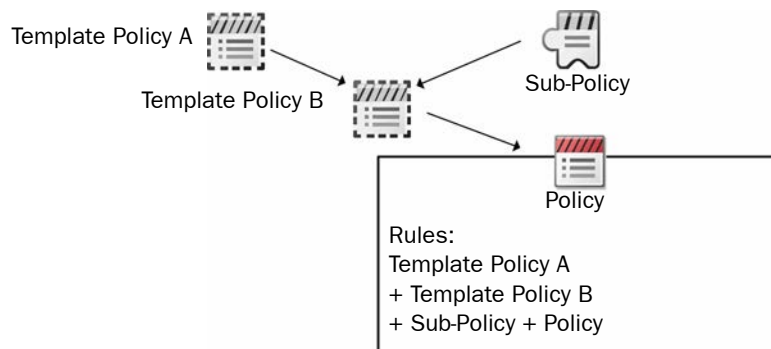
# Configuration of Policy Elements

Four kinds of policy elements are used in the policy configuration for Firewalls, Master Engines, and Virtual Firewalls:

- A *Firewall Template Policy* is a policy that is used as the basis for Firewall Policy and other Firewall Template Policy elements. A Template Policy may contain any number of rules. The rules in the Template Policy are copied as *inherited rules* into the Firewall Policy or the Template Policy which is based on the Template Policy. You can modify the inherited rules only by editing the original Template Policy from which the rules were inherited.
- An *Inspection Policy* element is a set of Inspection rules that are referenced from the Inspection tab of Firewall Policy and Firewall Template Policy elements. You can use the same Inspection Policy with multiple Firewall Policy and Template Policy elements.
- A *Firewall Sub-Policy* element is a section of rules that you can insert into Firewall Policy or Template Policy elements. The rules in the sub-policy are conditional rules that allow you to define matching criteria that determines whether the sub-policy applies to a connection. You can modify the rules by editing the sub-policy where the rules belong.
- A *Firewall Policy* element gathers together all the rules from the different policy elements (the rules added directly to the Firewall Policy, the rules from the Template Policy used as the basis of the policy, the Inspection rules from the Inspection Policy referenced from the policy's Inspection tab, and possibly conditional rules from one or more Sub-Policies added to the policy). A Firewall Policy is always based on a Firewall Template Policy element. The Firewall Policies are the only policy elements that can be installed on Firewalls, Master Engines, and Virtual Firewalls.

The hierarchy of how rules are inherited between the main policy elements is shown in the illustration below.

**Illustration 9.2 Rule Inheritance (without Inspection Rules Inherited from Inspection Policies)**



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

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

In addition to the inherited rules, the example policy also contains any rules that the administrators add to it directly. In the policy, the administrators can only edit the rules that were added directly to the policy. To change rules inherited from Template Policy A, Template Policy B, or the Sub-Policy, they must edit the policy in which the rules were originally defined.

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 Template Policy A is also added to any policy created based on that template. The next time the policies based on Template Policy A are installed on the engines, the new rule is used on all of those engines. There is no need to modify each individual policy separately.
- Restrict the editing rights of administrators. For example, administrators who are granted rights to only policies cannot edit the rules defined in the Template Policies on which the policies are based. Their actions have no effect on rules that are placed above the row where the Template Policy allows them to insert new rules. In the hierarchy shown in the illustration above, the insert point(s) for the Policy are defined in Template Policy B, which in turn can be edited only in the place where there is an insert point in Template Policy A.
- Reduce the likelihood of mistakes affecting important communications. Template Policies can be reserved for defining only the rules for essential communications, so that most daily editing is done in the lower-level policies. If the Template Policy is properly designed, the rules in the Template Policy cannot be overridden by any rules in the lower-level policy. Good organization also makes it easier to read policies, and reduces the risk of errors.
- Improve processing performance. With the help of 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 sub-policy. This reduces the processor load, which may lead to better throughput if the processor load is constantly very high.



## Default Elements

The default policy elements are introduced when you import and activate a recent dynamic update package (for example, during the installation). The elements may change when you install newer update packages.

None of the default policy elements can be modified. However, you can make copies of the default policies if you need to create a modified version.

The following table describes the default policy elements for Firewalls, Master Engines, and Virtual Firewalls.

**Table 9.1 Default Policy Elements for Firewalls, Master Engines, and Virtual Firewalls**

Element Type	Default Policy Name	Description
Firewall Template Policy	Firewall Template	A Template Policy that contains the predefined Access rules necessary for the engine to communicate with the SMC and some external components. The predefined Access rules are explained in <a href="#">Configuration of Access Rules</a> (page 103). The Firewall Template uses the Inspection rules defined in the No Inspection Policy. The Firewall Template provides access control without deep inspection.
	Firewall Inspection Template	A Template Policy that is based on the Firewall Template. It uses Inspection rules defined in the High-Security Inspection Policy. The Firewall Inspection Template enables deep inspection for all traffic.
Firewall Sub-Policy	DHCP Relay	A Sub-Policy that contains rules that allow the engine to relay DHCP requests from a host in one internal network to a DHCP server in a different network, as well as DHCP requests from VPN clients to an internal DHCP server.
Inspection Policy	No Inspection Policy	An Inspection Policy with a set of Inspection rules that do not enforce inspection.
	Medium-Security Inspection Policy	An Inspection Policy with a set of Inspection rules for detecting common threats. The Medium-Security Inspection Policy logs Situations categorized as Suspected Attacks but allows the traffic to pass. The Medium-Security Inspection Policy is suitable for Firewall, Layer 2 Firewall, Master Engine, and Virtual Firewall deployments. It is also suitable for inline IPS deployments in asymmetrically-routed networks and IPS deployments in IDS mode. The risk of false positives is low in production use.

**Table 9.1 Default Policy Elements for Firewalls, Master Engines, and Virtual Firewalls (Continued)**

Element Type	Default Policy Name	Description
Inspection Policy (cont.)	High-Security Inspection Policy	<p>An Inspection Policy with a set of Inspection rules for detecting common threats. The High-Security Inspection Policy terminates Suspected Attacks with an alert.</p> <p>The High-Security Inspection Policy is suitable for Firewall, Layer 2 Firewall, inline IPS, Master Engine, and Virtual Firewall deployments in which extended inspection coverage and strong evasion protection is required. The risk of false positives is moderate in production use. The High-Security Inspection Policy is suitable as the initial policy in most environments. The High-Security Inspection Policy terminates a connection if the engine cannot see the whole connection. It is recommended that you use the High-Security Inspection Policy as a starting point for your Inspection Policies.</p>
	Customized High-Security Inspection Policy	<p>An Inspection Policy that is based on the High-Security Inspection Policy and contains a set of customized Inspection rules.</p> <p>The High-Security Inspection Policy is an example of a highly customized Inspection Policy for network environments in which unconditional inspection coverage and evasion protection are required. The risk of false positives is high in production use.</p> <p>The Customized High-Security Inspection Policy was used when the IPS was tested at ICSA Labs and NSS Labs.</p>

Situations are the central elements in Inspection Policies. The Situation elements detect exploit attempts against known vulnerabilities and other commonly known security threats. Because dynamic updates include new and updated Situations, new patterns in traffic may begin to match when a new dynamic update is activated and you refresh the Inspection Policy.

In most environments we recommend using the High-Security Inspection Policy as the starting point for Inspection Policies. The High-Security Inspection Policy provides extended inspection coverage. It also protects the network against evasions, which are attempts to disguise attacks in order to avoid detection and blocking by network security systems. The only difference between the rules in the High-Security Inspection Policy and the Medium-Security Inspection Policy is in the way the Inspection rules handle Situations that are categorized as Suspected Attacks. The High-Security Inspection Policy terminates Suspected Attacks with an alert, whereas the Medium-Security Inspection Policy only logs Suspected Attacks.

Suspected Attacks also contain traffic patterns that may indicate malicious activities but are not any verified attack patterns. Suspected Attacks can catch zero-day attacks (attacks that are not yet publicly known), but may sometimes block some legitimate traffic if the traffic pattern happens to resemble malicious activities.

# 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* and the *McAfee SMC Administrator's Guide*.

Policy elements are only containers for the actual traffic handling rules. When you have decided on a policy hierarchy, you can populate the policy elements with the rules for handling the traffic. See [Access Rules](#) (page 101), [Inspection Policies](#) (page 119), and [Network Address Translation \(NAT\)](#) (page 131).

## Task 1: Create a Template Policy

*Firewall Template Policy* elements are used as a basis for Firewall Policies and other Firewall Template Policies. Every Firewall Policy and Firewall Template Policy that you create is based on a Firewall Template Policy. You can base several policies on the same Template Policy. The Firewall Template Policy or a customized copy of the Firewall Template Policy is always at the highest level of the policy hierarchy. It is not mandatory to create any custom Template Policies if you feel that it is not necessary in your environment.

When editing policies, the main difference between Template Policies and Firewall Policies are the special rows called *Insert Points*. Insert points are shown in both Template Policies and in Firewall Policies, but you can add them only to Template Policies. The Insert Points added to Template Policies mark the place where new rules can be added to policies that are based on the templates. If you create a new Template Policy and do not base the Template Policy on any pre-defined Template Policy, you must add insert points separately for Access rules and NAT rules.

**Illustration 9.3** Insert Point in a Template Policy and the Inheriting (Template) Policy

12	Local Cluster	ANY	IGMP
13	ANY	ANY	Default Services with Agents
Access rule : insert point			
15	ANY	12	Local Cluster(ND) ANY IGMP
16	ANY	13	ANY ANY Default Services with Agents
Access rule : insert point			
15	ANY	ANY	Ident
16	ANY	ANY	ANY

The illustration above shows what the same insert point looks like in a 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 Template Policy for inheritance to lower levels (orange) or whether it has been inherited from the higher-level Template Policy (green). Only the orange insert points are inherited to lower-level policy elements, so you must add at least one new insert point at each Template Policy level 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 Template Policy itself are not editable in lower-level policies that use the Template Policy. Such inherited rules are shown only on your request and they are displayed with a gray background. Only the actual rules are inherited from a higher-level Template Policy into the lower-level policies and Template Policies. The rights to edit policies and Template Policies are defined separately.

Together with the editing rights, insert points help ensure that important rules are not made void by configuration mistakes or modified by administrators who are not authorized to do so. Because the engine reads rules in order from the top down, any rules above the insert point in the higher-level Template Policy cannot be cancelled by anything a lower-level policy adds into the insert point.

## Task 2: Create a Policy

A Firewall Policy is the element that gathers together all the rules from the different policy elements: the rules inherited from the Template Policy that is used as the basis of the policy, rules from one or more Sub-Policies added to the policy, rules added directly to the policy, and rules from the Inspection Policy that is referenced from the Inspection tab in the policy. The Firewall Policy is the only policy element that you can transfer to a Firewall, Master Engine, or Virtual Firewall. The Firewall Policy is always based on a Template Policy, either a pre-defined Template Policy or a Template Policy that you have created.

## Task 3: Create a Firewall Sub-Policy

Firewall *Sub-Policies* and *IPv6 Sub-Policies* are sections of Access rules that you can insert in the IPv4 or IPv6 Access rules of Firewall Policies, Firewall Template Policies, or other Firewall Sub-Policies.

Using Sub-Policies can make the engine process traffic faster. You can also use Sub-Policies to organize rules. The Firewall Sub-Policies are not based on any Firewall Template Policy.

Sub-Policies may make it easier to read the policies and to assign editing rights to administrators. For example, you can give some administrators the rights to edit only certain Firewall Sub-Policies without giving them rights to edit Firewall Policies.

A Sub-Policy is inserted into some other policy element by adding a *Jump rule* to the policy element. The Jump rule directs connections that match the Jump rule for matching against the rules in the Sub-Policy.

**Illustration 9.4** A Sub-Policy in Use

A Jump rule inserts the 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 Sub-Policy are shown on a gray background, because they can be edited only within the Sub-Policy itself, not in the Firewall Policy that uses the rules.

You could use a Sub-Policy, for example, for examining 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 for matching against the Sub-Policy. Any connection that was destined to some other network would not get matched against any of the rules in the Firewall

Sub-Policy. This makes the matching process faster, because the engine can skip a whole Sub-Policy by comparing a connection to just one simple rule for any non-matching connection. If the Sub-Policy rules were inserted directly into the main Firewall Policy, the engine would have to compare all connections to all those rules individually (since that is the only way to find out whether the connection matches the rules). The performance benefit gained depends on the number and complexity of the rules that can be placed in a Sub-Policy and how heavy the engine load is to begin with.

## **Task 4: Install the Policy**

After creating or modifying a Firewall Policy, you must transfer the changes to the engine using the Management Client. You can either *install* the policy (transfers the policy you select) or *refresh* the policy (transfers the most recent version of the policy that the engine already uses). You can install the same Firewall Policy on several engines.

When you install a modified or a completely new Firewall Policy, any existing connections that are not allowed by the new Firewall Policy are dropped. The existing connections allowed by the new Firewall Policy are not affected; they continue uninterrupted. These include related connections and authenticated connections on the engines.

Policy installation transfers any new engine configuration information in addition to the Firewall Policy. Whenever you update the engine's configuration, you must reload the Firewall Policy on the engine to make the changes take effect. This includes, for example, changes in the routing configuration, the VPN configuration, and the properties of the Firewall, Master Engine, or Virtual Firewall element itself, even if the changes are not directly related to the rules in the Firewall Policy.

If the Firewall Policy installation fails, the system automatically rolls back to the previously installed configuration. By default, a rollback also occurs if the system detects that the new Firewall Policy or related configuration (such as routing configuration) does not allow the Management Server to connect to the engines. This safety feature prevents an administrator from inadvertently installing a configuration that would cause the critical management connections to fail.

# Using Policy Elements and Rules

---

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

- [Validating Policies](#)
- [User Responses](#)
- [Connection Tracking vs. Connectionless Packet Inspection](#) (page 95)
- [Policy Snapshots](#) (page 97)
- [Continue Rules](#) (page 97)
- [Adding Comments to Rules](#) (page 97)
- [Naming Rules](#) (page 98)

## Validating Policies

The number of rules in a Firewall Policy may grow quite large over time. It may become quite difficult, for example, to notice configuration errors in a policy. To make policy management easier and to make sure that the policy does not contain misconfigured rules, you can automatically validate the policy during editing as well as during the policy installation process. You can select 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 match traffic.

Additionally, the engines automatically count how many times each Access rule has matched. Engines also count the number of matches to NAT rules. You can run an analysis over a selected time frame in the policy editing view to display rule counter hits for each rule (in the Hits cell). This allows you to find otherwise valid rules that are unnecessary because they match traffic that does not appear in your networks.

## 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 Exceptions in the Inspection Policy. User Responses are not supported on Master Engines, or Virtual Firewalls.

# Connection Tracking vs. Connectionless Packet Inspection

Connection tracking means that the engine keeps a record of all currently open connections (stateful inspection). With connection tracking, the engine can verify that the connection proceeds according to the protocol standards. Connection tracking is also required for modifying addresses using NAT and enforcing some other connection-related settings. By default, connection tracking is on.

However, it is not necessary to keep track of the state of certain kinds of connections (for example, SNMP traps). Some applications may also communicate in a non-standard way that prevents them from being allowed through the engine when connection tracking is used. For those connections, you can disable connection tracking in the Access rules. This allows the engine to function as a simple packet filter for those connections. However, this also prevents the use of some features that require connection tracking.

When connection tracking is off, each packet that you want to allow must match an Access rule that allows the packet. This means that even if two-way communications are always opened one way, both directions of communication must be explicitly allowed in Access rules. Reply packets are not recognized, so they are not automatically allowed through. If done carelessly, turning connection tracking off reduces the benefits you gain from your engine and may even weaken security. You may have other options: in some cases using the correct Protocol Agent helps.



**Note** – Before disabling connection tracking, always check if there is a Protocol Agent for the protocol in question. The Protocol Agents can pass connections that require special handling when connection tracking is on, which is always a more secure option.

When connection tracking is enabled in an Access rule, the Service cell of the rule must contain one of the protocols supported for connection tracking (ICMP, TCP, UDP, or another protocol that belongs to the IP protocol suite). ICMP and UDP are stateless protocols that do not contain any connection data. However, ICMP and UDP packets contain data that the engine can use to build virtual connections. The engine can also build virtual connections based on the IP address and IP protocol data in other types of IP packets.

You can choose between several connection tracking modes, depending on the traffic type and how strictly you want the connections to be tracked. The effect of the connection tracking setting in the Access rules depends on the traffic type. The available options are explained in [Table 9.2](#).

**Table 9.2** Connection Tracking Modes in Access Rules

Mode	Explanation
Inherited from Continue Rule(s)	<p>The connection tracking setting defined in the Continue rule(s) higher up in the policy is used.</p> <p>The additional options available for connection tracking are explained in the next table.</p> <p><b>Note!</b> If connection tracking is disabled in Continue rule(s) higher up in the policy, the Firewall behaves as described in the Off (Not recommended) explanation below.</p>

**Table 9.2 Connection Tracking Modes in Access Rules (Continued)**

Mode	Explanation
Off (Not Recommended)	<p>Connection tracking is disabled. The Firewall operates as a simple packet filter and allows packets based on their source, destination, and port. You must add separate Access rules that explicitly allow the reply packets. NAT cannot be applied to traffic allowed without connection tracking.</p> <p><b>Note!</b> Turn off logging in the rule if you disable connection tracking. When connection tracking is off, a log entry is generated for each packet. This may put considerable strain on the engine, network, and the Log Server.</p>
Defined in Engine Properties	<p>The Firewall allows or discards packets according to the Connection Tracking mode selected in the Firewall properties.</p> <p>Protocols that use a dynamic port assignment must be allowed using a Service with the appropriate Protocol Agent for that protocol (in Access rules and NAT rules).</p> <p>The additional options available for connection tracking are explained in the next table.</p>
Normal	<p>The Normal mode is the default Connection Tracking Mode for Firewalls.</p> <p>The Firewall drops ICMP error messages related to connections that are not currently active in connection tracking (unless explicitly allowed by a rule in the policy). A valid, complete TCP handshake is required for TCP traffic. The Firewall checks the traffic direction and the port parameters of UDP traffic. If the Service cell in the rule contains a Service that uses a Protocol Agent, the Firewall also validates TCP and UDP traffic on the application layer. If a protocol violation occurs, the packet that violates the protocol is dropped.</p>
Strict	<p>The Firewall allows only TCP traffic that strictly adheres to the TCP standard as defined in RFC 793. The Firewall also checks the sequence numbers of the packets in pre-connection establishment states and for RST and FIN packets, and drops packets that are out of sequence. If the Service cell in the rule contains a Service that uses a Protocol Agent, the Firewall also validates the traffic on the application layer. If a protocol violation occurs, the packet that violates the protocol is dropped.</p>
Loose	<p>The Firewall allows some connection patterns and address translation operations that are not allowed in the Normal mode. This mode can be used, for example, if routing is asymmetric and cannot be corrected or if the use of dynamic routing protocols causes the Firewall to receive non-standard traffic patterns.</p>

If Connection Tracking is on, you can also set the **Idle Timeout** for connections. The timeout is meant for clearing the engine's records of old connections that the communicating hosts leave hanging. The timeout concerns only idle connections, so connections are not cut because of timeouts while the hosts are still communicating. The timeout defined for an Access rule overrides the default idle timeout value that is set for the protocol in the engine's properties.



**Caution – Setting excessively long timeouts for a large number of connections may consume engine resources and degrade engine performance and stability. Be especially careful when defining timeouts for ICMP and UDP. The ICMP and UDP virtual connections do not have closing packets, so the engine keeps the records for the ICMP and UDP connections until the end of the timeout.**



Connection Tracking options in Access rules also allow you to override the limit for concurrent connections from a single source and/or destination IP address defined on the Advanced tab in the Security Engine properties, in Virtual Resource properties, and in the properties of some interface types. When the set number of connections is reached, the next connection attempts are blocked by the engine until a previously open connection is closed.

Changes in the Connection Tracking mode affect how existing connections are handled when you install or refresh the policy. When you install or refresh the policy on an engine, the engine checks if the existing connections are still allowed in the policy. If the connection tracking mode changes from Loose to Strict, existing virtual ICMP connections are only allowed if they began with a valid packet (for example, not with a response packet). In addition, if the mode changes from Normal to Strict, existing TCP connections are only allowed if all the packets in the connection have been seen. In all other cases, changes in connection tracking mode do not affect existing ICMP, TCP, and UDP connections at policy installation.

## 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 Firewall 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.

## Continue Rules

The Continue action for a rule sets default options (such as logging options, idle timeout, etc.) 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. Firewall Template Policies are particularly convenient for setting options with a Continue rule, because all the Firewall Policies and Template Policies that use the Firewall Template Policy inherit the option settings you have specified. Continue rules are explained in detail in [Configuring Default Settings for Several Rules](#) (page 111).

## 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. You can also add a Rule Section that begins with a comment row and can contain one or more rules.

The Rule Section automatically contains all the rules below the Rule Section until the next Rule Section in the policy. You can expand and collapse the Rule Sections as necessary. The comment row in a Rule Section is displayed on a colored background (with configurable colors). This makes Rule Sections particularly useful in visually separating the sections of rules within a single policy.

## Naming Rules

In addition to comments, it is possible to specify an optional name or short description for Access rules and NAT rules in Firewall Policies, Exceptions in Inspection Policies, and rules in QoS Policies and Alert Policies. Names help administrators identify individual rules in large rule tables. You can also search for a rule by its name. If a rule has been named, the name is displayed in the Logs view as well.

## Examples of Policy Element Use

---

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

### Protecting Essential Communications

Company A has a firewall system administered by multiple administrators of various degrees of familiarity with networking, firewalls, and McAfee Firewalls. The administrators must often make very quick changes to respond to the needs of the company and attend to any problems detected.

In this situation, it is possible that someone may accidentally alter the Firewall Policy in such a way that important services are cut off. The administrators decide to separate the rules allowing the most important business communications from rules that deal with non-essential traffic to minimize this risk. The administrators:

1. Create a new Firewall Template Policy and select the pre-defined Firewall Template as the basis of the policy.
2. Cut and paste the rules allowing essential communications from their current Firewall Policy into the new Firewall Template Policy.
  - In this case, all administrators are allowed to edit the new Firewall Template Policy as well.
3. Add an insert point below the copied rules in the Firewall Template Policy.
  - Having the insert point below the essential rules prevents the rules added to the inheriting Firewall Policy from affecting the essential communications.
4. Reparent their current Firewall Policy to use the new template, moving it down a step in the policy hierarchy.
5. After validating the policy and making sure that the rules are correct, refresh the current Firewall Policy.
  - Since most daily editing is done in the Firewall Policy, there is less risk of someone accidentally changing the essential rules in the Firewall Template Policy, because the rules are not editable in the Firewall Policy.

### Improving Readability and Performance

Company B has two separate DMZs, one for the extranet and one for other web services. The number of services offered is quite large. The company also has a large number of partners and customers that have varying access rights to the different services. The administrators realize that a large number of the rules in their policies are related to the DMZ connections. The rest of the rules govern access to and from the company's internal networks. Many of the rules have

been entered over time by inserting them at the beginning of the rule table, so rules governing access to the different networks are mixed and finding all the rules that govern access to a particular network takes time.

The administrators decide that they want to make their Firewall Policy more readable and at the same time optimize the way the firewall handles traffic, so they:

1. Create two new Firewall Sub-Policies: one for each DMZ.
2. Cut-and-paste the rules from the current Firewall Policy into the correct Firewall Sub-Policy.
3. Add Jump rules into the Firewall Policy to direct the examination of traffic to/from the different networks into the correct Firewall Sub-Policy.
4. Refresh the Firewall Policy.

## Restricting Administrator Editing Rights

Company C is implementing a distributed network with multiple sites: one central office where most of the 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 administrators.

The administrators decide to limit the privileges of the branch office IT staff so that they are not able to edit the policies of the firewalls at any of the other sites. The administrators:

1. Create a new Firewall Template Policy and select the pre-defined Firewall Template as the basis of the policy.
2. Add rules to the Firewall Template Policy using Alias elements to cover the essential services that each of these sites has, such as the VPN connections to the central site.
  - Using a common Firewall Template Policy for all the branch offices also eliminates the need to make the same changes in several policies, easing the workload.
3. Create a new Firewall Policy based on the new Firewall Template Policy for each of the branch office sites.
  - Although a single Firewall 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 Firewall Template Policy, so rules can still be shared without duplicating them manually.
4. Grant each Firewall Policy to the correct Firewall element.
  - After this, only the correct policy can be installed on each firewall. No other policy is accepted.
5. Create new administrator accounts with restricted rights for the branch office administrators and grant the correct Firewall element and Firewall Policy to each administrator.
  - The branch office administrators are now restricted to editing one Firewall Policy and can install it on the correct firewall.
  - The branch office administrators are not allowed to edit the Firewall Template Policy the policy is based on, nor can they install any other policies on any other firewalls.

For detailed information on administrator rights, see the *McAfee SMC Reference Guide*.



## CHAPTER 10

# ACCESS RULES

Access rules are lists of matching criteria and actions that define how the engine treats different types of network traffic. They are your main configuration tool for defining which traffic is stopped and which traffic is allowed.

The following sections are included:

- ▶ [Overview to Access Rules](#) (page 102)
- ▶ [Configuration of Access Rules](#) (page 103)
- ▶ [Using Access Rules](#) (page 110)
- ▶ [Examples of Access Rules](#) (page 115)

# Overview to Access Rules

---

The IPv4 and IPv6 Access rules are traffic handling rules in which you define the details of how you want the traffic to be examined and which action you want to take when matching details are found. The Access rules are stored in policy elements, which are discussed in [Firewall Policies](#) (page 83).

The traffic matching is based on the information contained in the packets:

- Source and destination IP addresses.
- Protocol-specific information, such as the port information for protocols that use ports.

Additional matching criteria that is not based on information in the packets includes:

- The interface the traffic is coming from or going to. This allows you to restrict which traffic is allowed through which interfaces in more detail than basic antispoofing.
- The VPN the traffic is coming from (on an engine where that VPN terminates). This allows creating rules that apply to VPN traffic only, or rules that apply to all traffic except VPN traffic.
- (*IPv4 only*) User authentication (allowing you to create rules that define the end-users who are allowed to make connections and the authentication methods for the end-users).
- (*Firewalls only*) The User or User Group of a user who has logged in to an integrated Microsoft Active Directory domain (allowing you to create user-specific rules without configuring authentication).
- The day of the week and the time of day (allowing you to enforce rules only during certain times, such as working hours).

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

- Specifically allow the traffic.
- Specifically stop the traffic.
- (*IPv4 only*) Allow the traffic on the condition that the user has passed authentication.
- Allow the traffic on the condition that a VPN is established.
- Allow the traffic on the condition that the same source and/or destination IP address does not have an excessive number of connections already open (concurrent connection limit).

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

Additionally, Access rules select which allowed traffic is subjected to further inspection against the Inspection rules.

In addition to traffic allowed by the Access rules, Firewalls automatically allow the following types of traffic with specific configurations:

- DHCP requests and replies for an interface for which a DHCP server is enabled.
- DHCP requests and replies for an interface that has a dynamic IP address.
- State synchronization between cluster nodes.
- IPv6 Neighbor Discovery traffic.

# 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 Firewall Policy, Template Policy, and Sub-Policy elements. You can create new Access rules in the Policy Editing View. You can also create Access rules in the Logs view based on one or more selected log entries (*not available for IPv6 Access rules*).

**Illustration 10.1** Newly Inserted IPv4 Access Rule - Main Cells

ID	Source	Destination	Service	Action	Authentication	QoS ...	Logging	Time	Comment	Rule Name	Source VPN	Hits
15.1	<None>	<None>	<None>	Discard						@652.0		
Discard all												

Mandatory cells for matching traffic      Engine applies Action when it finds a match

[Illustration 10.1](#) above displays an Access rule that has just been inserted into the policy. The matching cells are set to **<None>** and the action is set to **Discard**, 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 **Protocol**) must be set to some value other than **<None>** for the rule to be valid. The **Source VPN** cell is also matched against traffic in the inspection process, but it can be left empty to match all traffic. The other editable cells specify further conditions and options, such as logging.

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 table below explains briefly what each Access rule cell does and which elements you can use in the rules. 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	( <i>Not editable</i> ) 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.
Source	A set of matching criteria that defines the IP addresses and interfaces 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, as well as User and User Group elements. Network Elements used in IPv4 Access Rules must contain IPv4 addresses, and Network Elements used in IPv6 Access Rules must contain IPv6 addresses.
Destination	

**Table 10.1 Access Rule Cells (Continued)**

Cell	Explanation
Service	A set of matching criteria that defines the service or application the rule matches. Services match a certain port, but they often also reference a Protocol for more advanced, application-layer inspection and traffic handling. The Service cell accepts Service and Service Group elements, URL Situations, Applications, and TLS matches.
Action	Command for the engine to carry out when a connection matches the rule. Also allows you to set options for anti-virus ( <i>IPv4 only, not supported on Virtual Firewalls</i> ), anti-spam ( <i>IPv4 only, not supported on Virtual Firewalls</i> ), blacklisting, connection tracking, deep inspection, rate-based DoS protection, scan detection, user responses, and VPN connections.
Authentication	( <i>IPv4 only</i> ) Defines whether the rule requires end-users to authenticate, which end-users the rule applies to when the rule requires authentication, and which authentication methods are valid for the rule. See <a href="#">Directory Servers</a> (page 201), <a href="#">User Authentication on the Firewall</a> (page 207), and <a href="#">External User Authentication</a> (page 213).
QoS Class	The QoS Class that the engine assigns to connections that match this rule. Used in traffic prioritization and bandwidth management. The QoS Class has effect only if you set up QoS Policies, see <a href="#">Bandwidth Management and Traffic Prioritization</a> (page 245).
Logging	The options for logging when traffic matches the rule. If no options are specified, the behavior is as explained in <a href="#">Task 4: Select Logging Options</a> (page 109).
Time	The time period when the rule is applied. If this cell is left empty, the rule applies at all times.
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. You can also add separate comment rows in between rules.
Rule Name	Contains a rule tag and optionally a rule name. Name: ( <i>Optional</i> ) Name or description for the rule. Displayed alongside the rule tag. Tag: (Not editable) An automatically assigned unique identifier 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, @180.2). 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.
Source VPN	Makes the rule match traffic based on whether it is coming from a specific VPN. If this cell is left empty, the rule matches both VPN and non-VPN traffic.
Hits	( <i>Not editable</i> ) Shows the number of connections that have matched the rule. This information is only shown if you have run a Rule Counter Analysis for the policy. The cell shows “N/A” if there is no information available about the rule.



# Considerations for Designing Access Rules

One of the crucial issues in designing policies is the order of the rules. The most important thing to keep in mind when editing Policies is that the rules are read from top down. The actions **Allow**, **Refuse**, and **Discard** and the action **Use IPsec VPN** with the option **Enforce** stop the processing from continuing down the 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.

**Example** 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.

## Default Elements

There are two predefined Firewall Template Policies called Firewall Template and Firewall Inspection Template. They contain the basic Access rules that allow communications between the engine on which the policy is installed and other SMC components. The Firewall Inspection Template is based on the Firewall Template.

You must use one of the predefined Firewall Template Policies as the basis for defining your own templates and policies, as it is not possible to create a new template at the highest level of the policy hierarchy. No changes can be made directly to the predefined Firewall Template Policies. However, you can create your own copies of the predefined Firewall Template Policies if you have a specific need to modify the rules in them.



**Note** – If you use a copy of a predefined Firewall Template Policy, you may have to adjust your rules manually when the system is upgraded to account for changes in system communications. Upgrades can change only the predefined Firewall Template Policies, not the copies.

There is a yellow row near the end of the list of rules on the IPv4 Access and IPv6 Access tabs in the predefined Firewall Template Policies. The yellow row marks the *insert point*, where rules can be added in the inheriting Firewall Policy and Firewall Template Policy elements.

The rules above the insert point detail the various kinds of system communications. Most of the IP addresses are defined using Aliases to make the rules applicable on any system where they are installed. These Aliases are default elements and they are listed in the appendix [Predefined Aliases](#) (page 337). The Service cell is the best starting point for understanding in greater detail what these rules do. See appendix [Default Communication Ports](#) (page 329) for a listing of the system communications and the Service elements that correspond to them.

There are two rules below the insert point. The rule directly below the insert point has the action **Refuse** for the Ident protocol traffic, which means that this traffic is stopped with an ICMP error message sent to the Ident request sender. This rule exists to prevent Ident requests from being silently dropped (as the next rule specifies for all other traffic), because silently dropping Ident requests causes delays in communications. The last rule before the end of the policy is a rule that discards all traffic and creates a log entry that is stored. This rule's purpose is to ensure that this connection dropping is logged, since the engine silently drops the connection without creating a log entry if the matching process reaches the end of the policy.

# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define the Source and Destination

The Source and Destination cells specify the IP addresses that are compared to the IP addresses in each packet's header. Based on these and other criteria, the rule is applied to matching packets.

You can add more than one element in each cell. You can optionally define detailed lists of matching criteria for each cell by combining Users (stored in an integrated Active Directory database), Network Elements, DNS Domain Names, and interface Zones. Each row of the list is combined with a logical AND: all items must match for the row to match. Groups, Aliases, Address Ranges, and Expressions are also useful for defining IP addresses in complex scenarios.

You can set the Source and Destination cells to ANY to make the rule match all possible source or destination IP addresses.

## Task 2: Define the Service

The Service cell defines the applications and protocols that are compared to the traffic. By default, the Service is set to <None>, and you must change the value to make the rule valid.

There are ready-made Services 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 several Services. You can optionally define detailed lists of matching criteria by combining URL Situations (for URL filtering), Applications, Services, and TLS matches. Each row of the list is combined with a logical AND: all items must match for the row to match.

Certain Services are associated with Protocols of the type Protocol Agent, which define more advanced inspection and handling for the connections. Additionally, the Protocol element identifies the protocol for the traffic for inspection against the Inspection rules. For more information on Protocols and the network protocols that require Protocols of the type Protocol Agent, see [Protocol Agents](#) (page 145).

You can set the Service to ANY to make the rule match traffic using any application or protocol. A previous Continue rule may define a Protocol of the type Protocol Agent for certain types of traffic that is allowed by rules with ANY as the Service (see [Configuring Default Settings for Several Rules](#) (page 111)). If there is no previous Continue rule that matches the same connection that would define a Protocol of the type Protocol Agent, a rule that allows ANY Service does not apply a Protocol of the type Protocol Agent to the traffic.



**Note – Firewalls cannot handle some types of traffic correctly if the traffic is allowed without the correct Protocol of the type Protocol Agent when connection tracking is on (stateful inspection).**

### Task 3: Select the Action and Action Options

The Action cell defines what happens when a packet matches the rule. The Action Options define additional action-specific options for connection tracking, deep inspection, anti-virus (*IPv4 only, not supported on Virtual Firewalls*), anti-spam (*IPv4 only, not supported on Virtual Firewalls*), user responses, and blacklisting. If no options are specified, the Action Option settings from the previous Continue rule are applied.

The available actions are explained in [Table 10.2](#).

**Table 10.2** Action Field Options

Action		Explanation
Allow		The connection is let through the engine.
Continue		Stores the contents of the Options and QoS Class cells and the Protocol (if defined in the Service used) in memory and continues the inspection process. Used for setting options for subsequent rules as explained in <a href="#">Configuring Default Settings for Several Rules</a> (page 111).
Discard		The connection is silently dropped. Optionally, a response message can be shown to the end-user ( <i>HTTP traffic only</i> ).
Refuse		The connection is dropped and an ICMP error message is sent in response to notify the packet's sender.
Jump		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.
Use IPsec VPN	Enforce	The connection is allowed if the specified VPN is used. Otherwise the connection is discarded.
	Apply	The connection is allowed if the specified VPN is used. Otherwise, the rule is considered as non-matching and the matching process continues to the next rule.
	Forward	The connection is forwarded from one VPN to another. For more information, see <a href="#">Policy-Based VPN Configuration</a> (page 269).
	The Selected IPsec VPN	Specifies a gateway-to-gateway or a client-to-gateway IPsec VPN.
	\$ Client-to-Gateway IPsec VPNs	Specifies any client-to-gateway IPsec VPNs.
Apply Blacklist		Checks the packet against the blacklist according to the options set for this rule. If the packet matches a blacklist entry, the connection is discarded.

The available action options are explained in [Table 10.3](#).

**Table 10.3 Action Options**

Action Option	Explanation
Anti-Virus ( <i>IPv4 only, not supported on Virtual Firewalls</i> )	Defines whether traffic is inspected against a virus signature database. By default, the anti-virus options are undefined, which means that the rule uses anti-virus options set in the previous matching Continue rule above. See <a href="#">Virus Scanning</a> (page 171) for more information. Used by rules with the Allow, Continue, or Jump action.
Anti-Spam ( <i>IPv4 only, not supported on Virtual Firewalls</i> )	Defines whether SMTP protocol traffic is filtered for spam. By default, the anti-spam options are undefined, which means that the rule uses anti-spam options defined in the previous matching Continue rule. See <a href="#">Spam Filtering</a> (page 167) for more information. Used by rules with the Allow, Continue, or Jump action.
Blacklisting	Defines which blacklist entries are enforced for connections that match the rule based on the component that added the blacklist entry to the blacklist. A restriction based on the blacklist sender may be necessary, for example, if the same IP address exists in two different networks. The default setting is to enforce all blacklist entries regardless of the component that created the entry. Used by rules with the Apply Blacklist action.
Connection Tracking	Defines whether the engine keeps a record of the currently open connections (stateful inspection). See <a href="#">Connection Tracking vs. Connectionless Packet Inspection</a> (page 95) for more information. Used by rules with the Allow, Continue, or Jump action.
Deep Inspection	Defines whether matching connections are also inspected against the Inspection rules. Used by rules with the Allow, Continue, or Jump action.
Rate-Based DoS Protection	Defines whether rate-based DoS protection is enabled for traffic that matches the rule. Used by rules with the Allow, Continue, or Jump action.
Scan Detection	Defines whether scan detection is enabled for traffic that matches the rule. Used by all rules.
User Response	Defines which automatic response is shown to the end-user when an HTTP connection is discarded. Used by rules with the Discard action.

## Task 4: Select Logging Options

By default, the rule's **Logging** options are undefined, which means that the rule uses logging options that have been set in the previous Continue rule above. If there is no previous Continue rule, a Stored-type log entry is created. Logging for the closing of the connection can be turned on or off, or on with accounting information. You must collect accounting information if you want to create reports that are based on traffic volumes.

The log levels are explained in [Table 10.4](#).

**Table 10.4 Log Levels in Access Rules**

Log Level	Explanation
None	Does not create any log entry.
Transient	Creates a log entry that is displayed in the Current Events mode in the Logs view (if someone is viewing it at the moment) but is not stored.
Stored	Creates a log entry that is stored on the Log Server.
Essential	Creates a log entry that is shown in the Logs view and saved for further use.
Alert	Triggers an alert entry.

When the Log Server is unavailable, log entries are temporarily stored on the engine. When the engine is running out of space to store the log entries, it begins discarding log data in the order of importance. Monitoring data is discarded first, followed by log entries marked as Transient and Stored, and finally log entries marked as Essential. The Alert entries are the last log entries to be discarded.

The settings for storing the logs temporarily on the engine are defined in the engine's log spooling policy. For more information, see the *McAfee SMC Administrator's Guide*.

## Task 5: Add User Authentication Requirements

(IPv4 only) The engine can enforce user authentication. A client-to-gateway VPN always requires some form of authentication, but you can also add the authentication requirement to non-VPN rules. For more information, see [User Authentication on the Firewall](#) (page 207) or [External User Authentication](#) (page 213).

The authentication requirements are configured in the **Authentication** cell. The cell accepts User and User Group elements to define the end-users who are allowed to make connections allowed by the rule, and Authentication Method elements to define the type of authentication required for connections that match the rule.

Authentication Server users cannot be used directly in rules. Instead, you must use the User element from the external directory server to which the Authentication Server user is linked. User Groups cannot be used with the Authentication Server.

If the authentication fails, the connection is discarded. If the authentication succeeds, the connection is allowed through.

## Task 6: 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 engine's local time zone. Also consider that UTC time does not adjust to daylight saving time (summer time).

## Task 7: Restrict the Rule Match Based on Source VPN

Optionally, you can match the rule based on whether the traffic is coming from a VPN. You can define that the rule matches only non-VPN traffic, or only traffic from a certain VPN. For more information, see [Overview to VPNs](#) (page 261).

# 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](#)
- [Configuring Default Settings for Several Rules](#) (page 111)
- [Using Aliases in Access Rules](#) (page 113)
- [Creating User-Specific Access Rules](#) (page 113)
- [Using Domain Names in Access Rules](#) (page 114)
- [Interface Matching in Access Rules](#) (page 114)

For general information on using rules, see [Using Policy Elements and Rules](#) (page 94).

## Allowing System Communications

The necessary communications between the engine and other SMC components are allowed in the predefined Firewall Template Policy. However, the predefined templates do not allow other SMC components to communicate through the engine to some third SMC component.

For example, in a configuration where you have a firewall and a Log Server at a remote site that are managed by a Management Server behind a firewall at a central site, you must create rules in the Firewall Policy at the central site to allow:

- Management and monitoring connections to/from the remote firewall.
- Monitoring and log browsing connections from the central site to the remote Log Server.
- Any remote-site Management Client connections to the Management Server at the central site.

If NAT is applied to the connections, Access rules alone are not enough. You must also create Location elements and add Contact Addresses for the elements to define which translated addresses are necessary for making contact (see [NAT and System Communications](#) (page 139)).

There are predefined Service elements for all system communications. You can use these to create Access rules. See [Default Communication Ports](#) (page 329) for more information on system communications.

## 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.

When a connection matches a rule with Continue as the action, some of 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. There are also default values that are used for rules that are set to use the values of a Continue rule, but there is no previous matching Continue rule.

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

- The Connection Tracking option (default is on):
  - Idle Time-out overrides also the global defaults set in the engine's properties.
  - The concurrent connection limits define the maximum number of connections allowed from a single source and/or destination IP address.
- The logging options (default is **Stored**).
- The Protocol inside the Service (used to apply a Protocol to rules with ANY as their Service, see [Using Continue Rules to Set the Protocol](#) (page 112)).
- The QoS Class (default is that no specific QoS Class is assigned).

Continue rules are defined the same way as other rules. However, you must keep in mind that when any of the options listed above is set in the Continue rule, many or even all rules below may be affected. The Continue rule options are used by the rules below, provided that the Source, Destination, Service, and the optional Source VPN definition 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.







Continue rules behave in the same way as any other rules. A Continue rule can be overridden by some subsequent Continue rule that has an identical scope (Source, Destination, etc.), 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.

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 Logging Options

One common use for the Continue action is to set the default log level for all subsequent rules. Instead of setting the log level for all rules individually, you can set a Continue rule in a template or in a policy to set the default log level. The log level for any subsequent matching rules can be left undefined. The rules trigger logging as defined in the Continue rule.

**Illustration 10.2** Setting the Default Log Level

IPv4 Access		IPv6 Access		Inspection		IPv4 NAT		IPv6 NAT	
ID	Source	Destination	Service	Action	Authentication	QoS ...	Logging		
15.1	 ANY	 ANY	 ANY	 Continue			 Transient	 No Closing	

In [Illustration 10.2](#), the default log level is set to Transient for any source, destination or service. All subsequent rules in this policy and any sub-policies log Transient by default. Individual rules can still override this option with specific log levels, such as Essential or Stored.

If logging is not defined for a rule and there is no prior Continue rule that sets logging options, the default log level is **Stored**.

## Using Continue Rules to Set the Protocol

The default Protocol can be set using the Continue action. This way, the correct Protocol is also used for traffic that is allowed by rules that are set to match any Service (this may be even be mandatory, for example, if you want to allow certain protocols that allocate ports dynamically). The Firewall Template includes one Continue rule that defines that Protocols of the type Protocol Agent are used for the Service Group **Default Services with Agents**.

**Illustration 10.3** Protocol Agent Rule in Firewall Template

14	ANY	ANY	Default Services with Agents	Continue
----	-----	-----	------------------------------	----------

If you want to set a Protocol for more types of protocols or override the Default rule shown in [Illustration 10.3](#) for some or all connections, place one or more Continue rules at the top or some other suitable place in your own template or policy. The Source and Destination can be some specific addresses if you want to limit the scope of your Continue rules.

For more information on using Protocols of the type Protocol Agent in rules, see [Protocol Agents](#) (page 145).



## Using Aliases in Access Rules

*Aliases* are one of the most useful tools for reducing the complexity of a policy. With Aliases, you can avoid creating multiple, near-duplicate rule sets when you have several engines. The Alias element is used like any other network element. However, the IP addresses that the Alias represents depends on the engine where the rules are installed. The IP address to engine mapping is defined in the Alias element.

For example, in a company that has offices in several different locations, each office can have its own web server. The web server rules could be put in a single Sub-Policy, but each location's web server has a different IP address. Normal rules would require allowing access to all IP addresses on all engines, which is not only unnecessary, but can also be considered a security risk. Using Aliases, the company can create a single set of rules that are still valid when applied to multiple engines, but which do not allow access to IP addresses that are not in use on a particular engine.

In this way, Aliases simplify policies without reducing security.

## Creating User-Specific Access Rules

The optional User Agent can be installed on a Windows Server that communicates with an Active Directory Domain Controller to associate users in an integrated Active Directory database with IP addresses. For more information, see [User Agents for Active Directory](#) (page 203). This makes it possible to use User and User Group elements as the source or destination of a rule to create user-specific rules without requiring user authentication.

Information about Users' IP addresses is cleared from the firewall's cache if the firewall becomes unable to contact the User Agent. This can prevent rules that block connections from matching. For this reason, it is recommended to use Users and User Groups only in rules that allow a connection.

Additionally, Users may be incorrectly removed from the list of IP addresses if the User's workstation does not respond to an ICMP echo (ping) request from the User Agent. Workstation monitoring can optionally be disabled in the Windows registry to prevent this. See the User Agent [Release Notes](#) for more information.

User-specific rules do not replace user authentication; they are a tool to simplify the configuration of access control, and improve the end-user experience by allowing transparent access to services. They are intended to be used for trusted users in a trusted environment where strong authentication is not required. User-specific rules can be used together with user authentication rules to allow some user groups to access a service, while otherwise requiring authentication for the same service.

## Using Domain Names in Access Rules

You can use *Domain Name* elements in Access rules to represent an Internet domain that may be associated with multiple IP addresses. If you have specified one or more DNS servers in the engine's properties, the engine periodically queries the DNS server to automatically resolve domain names to IP addresses. This makes it possible to create rules that are valid even if new addresses are added to the domain or the domain's IP addresses change. If the DNS server returns multiple IP addresses for the same domain name, the engine associates all of the IP addresses with the domain name. However, if there are a very large number IP addresses associated with the same domain name, the DNS server may only reply with a few of the IP addresses at a time. In this case, the engine may need to make additional queries to the DNS server to resolve all of the IP addresses for the domain name. By default, the engine queries the DNS server every six minutes. Resolved IP addresses are kept in the engine's DNS cache for a maximum of one hour by default.



**Note** – The DNS cache is not synchronized between nodes of a cluster. Each node separately queries the DNS server using the node's NDI address. It is possible that the DNS cache may be different on different nodes of a cluster.

## Interface Matching in Access Rules

*Zone* elements are interface references that can combine several network interfaces of a engine into one logical entity. Using Zones in the Source or Destination cells allows you to restrict traffic according to which interface(s) the traffic is travelling through. This can be useful, for example, when a certain type of traffic is only considered valid it when it travels through a specific interface, but basic Anti-Spoofing would have allowed the traffic on any interface.

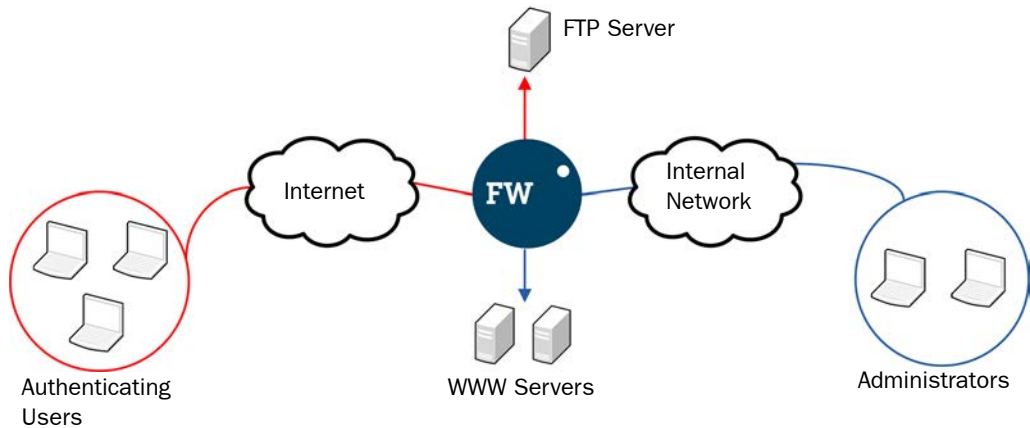
# Examples of Access Rules

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

## Example of Rule Order

Company A has an office network, a DMZ for WWW servers, and a second DMZ for an FTP server. At this point, the administrators only need to add rules for the DMZ traffic.

**Illustration 10.4** Company A's Communications of Special Interest



The WWW servers must be accessible to anyone from both internal and external networks. HTTP traffic will be inspected against the Inspection rules, excluding the administrators' own PCs (on the right in the illustration), since they often test the servers for vulnerabilities. The FTP server is accessible to all users in the general office network, but only to certain external users (on the left in the illustration) that authenticate using an external authentication server.

The administrators:

1. Create Host elements for the WWW servers, the FTP server, and the administrators' PCs.
2. Create a Group element that contains the WWW server Host elements.
3. Create a Group element that contains the administrator PCs' Host elements.
4. Configure an external authentication server for use with the Firewall.
5. Create User and User Group elements for the allowed external FTP users.

6. Add IPv4 Access rules with the Allow action for access to the DMZs:

**Table 10.5 Access Rules for the DMZ**

Source	Destination	Service	Authentication	Action
"Administrator PCs" Group	"WWW Servers" Group	"HTTP" Service		Allow (Deep Inspection Off)
ANY	"WWW Servers" Group	"HTTP" Service		Allow (Deep Inspection Off)
Network element for Office Network	"FTP Server" Host	"FTP" Service		Allow (Deep Inspection Off)
ANY	"FTP Server" Host	"FTP" Service	"External Users" User Group Require authentication with the external authentication method selected	Allow (Deep Inspection Off)

- As seen in the rule table, there are two rules for traffic to both the WWW servers and the FTP server.
- The rules are arranged so that the more specific rules are above the more general rules. For example, the rule allowing administrators to connect to the WWW servers without checking against the Inspection rules is above the more general rule that allows any connection to the servers as subject to the Inspection rules.
- If the first two rules were in the opposite order, the rule specific to administrators would never match, as the rule with the source as ANY would be applied first, the connection would be allowed according to that general rule, and the firewall would stop checking the rule table.

## Example of Continue Rules

Company B has decided to implement QoS Policies. The administrators want to set the QoS Class for traffic using a classification of high, medium, and low for all traffic depending on the sender. High priority is assigned to a small number of hosts in different networks, medium priority to one internal network, and low priority to all other hosts. The administrators want to follow how much traffic is allowed using the highest priority, so they also want to make sure that this traffic is logged with the accounting option turned on. They decide that the lower priorities of traffic need not be permanently logged at this point, so the administrators:

1. Configure the QoS features.
2. Create elements for all high-priority hosts.
3. Add the following Access rules to the top of their policy:

**Table 10.6 Continue Rules for Logging and QoS Class**

Source	Destination	Service	Action	Logging	QoS Class
Important Hosts	ANY	ANY	Continue	Stored with accounting	High priority
Network element for Important Network	ANY	ANY	Continue	Transient	Medium priority
All other Hosts	ANY	ANY	Continue	Transient	Low priority

- After adding these rules, individual rules can override the settings as needed, but most of the existing rules governing access from internal networks to the Internet now use the QoS Class and Logging options as set in these rules.
4. Transfer the policy to the firewall.

## Example of User-Specific Rules

Company C has an existing Microsoft Active Directory server that it uses for user accounts in its Windows domain. Users are divided into groups according to the department they work in. The administrators have already integrated the Active Directory user database with the SMC to be able to view and manage Users in the Management Client.

There is already an Access rule that blocks access to a popular video sharing site. However, the marketing team needs to be able to publish videos for its new marketing campaign on the site. The administrators want to allow users in the marketing group to access the site, but do not want to require user authentication.

Because the video sharing site has multiple servers with different IP addresses, the administrators decide to use a Domain Name element to dynamically resolve the IP addresses of servers in the video sharing site's Internet Domain.

The administrators:

1. Install the User Agent on a Windows server that communicates with the Active Directory Domain Controller.
2. Create a User Agent element and select it in the firewall's properties.
3. Add the following Access rule before the rule that blocks access to the video sharing site:

**Table 10.7 User-Specific Access Rule**

Source	Destination	Service	Action
Marketing user group	Domain Name element that represents the video sharing site	HTTP	Allow

4. Transfer the policy to the firewall.

## CHAPTER 11

# INSPECTION POLICIES

Inspection Policies define how the engines 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 Policies](#) (page 120)
- ▶ [Configuration of Inspection Policies](#) (page 121)
- ▶ [Using Inspection Policies](#) (page 128)
- ▶ [Example of Inspection Rules](#) (page 129)

# Overview to Inspection Policies

---

Inspection Policies define how the main traffic analysis is done for traffic that has been allowed and selected for inspection in the Access rules. The Inspection Policies are selected in Firewall Policy elements, which are discussed in [Firewall Policies](#) (page 83).

Inspection Policies examine the entire contents of the packets throughout whole connections to see if the data being transferred contains a pattern of interest. Dynamic update packages are the main source of these patterns, but you can also define new patterns as Situation elements, which are discussed in [Situations](#) (page 183).

Firewalls can inspect all protocols. Express Firewalls can inspect only the HTTP, POP, IMAP, SMTP, and DNS protocols. All inspection for Virtual Firewalls is handled by the Master Engine to which the Virtual Firewalls belong. Master Engines only inspect their own communications and process traffic for Virtual Firewalls.

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

- 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 Policy provides several different ways to react when some traffic is found to match a pattern of interest:

- Stop the traffic.
- Reset the connection.
- Blacklist the connection on one or more Security Engines.
- 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.



# Configuration of Inspection Policies

The engine inspects traffic based on Situation elements, which contain the information about traffic patterns. Patterns may trigger immediate responses or just be recorded. Detected events can be matched against Correlation Situations, which combine and further analyze the traffic-based findings to detect additional threats and produce an easy-to-read event stream.


























Inspection Policies are selected on the **Inspection** tab inside Firewall Policy and Firewall Template Policy elements. Sub-Policies cannot contain Inspection Policies. You can add new rules to the Inspection Policy in the Policy Editing View and also in the Logs view based on log entries.

The Inspection Policy has two parts:

- The **Inspection** tab contains the main rules for finding traffic patterns. The Rules tree is applied to all traffic that is not handled as Exceptions.
- The **Exceptions** tab 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 on the Inspection tab contains a tree of Situations, which are organized under Situation Types. This tree allows you to control which inspection checks trigger a reaction 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 to certain IP addresses or Logical Interfaces in the Rules tree.

















## Illustration 11.1 Inspection Tab - Rules Tree

Rules  Tools					
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. The most frequent use of Exceptions is to eliminate false positives. This typically requires permitting a pattern for some part of the traffic while the same pattern still triggers a reaction when it is encountered in any other traffic.

The illustration below shows the Exceptions tab with some rules.

**Illustration 11.2 Exceptions Tab**

Exceptions							
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 cell, which 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, the engine can 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 124).

## Verifying and Tuning Inspection

The most common way to introduce inspection is to start with a default Inspection Policy. Tuning the policy is important, since a general policy that is meant to work in all environments is not necessarily suited to your particular network scenario. A tuning period is needed to activate and deactivate inspection checks based on the findings and your particular needs. Tuning increases the relevancy and accuracy of the findings that the system generates.

To assist in policy tuning, you can use the *passive termination* feature. When passive termination is used, the engine creates a special log entry that notes that a certain connection would have been terminated, but the engine 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 engine's 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 use the Tags for the five most recent updates (**Situations→By Tag→By Situation Tag→Recent Updates**).

# Considerations for Designing Inspection Policies

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 detailed rules specific to some IP addresses and Protocols is defined as Exceptions. The general rules that are applied to remaining traffic are defined in the Rules tree on the Inspection tab.

The traffic matching in Inspection rules and exceptions 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 and exceptions are matched based on the Situation element that contains the detected pattern. Inspection rules and exceptions match certain patterns only. Non-matching traffic is passed through without any reaction.

The Situation-based configuration logic means that the behavior of the Inspection rules and exceptions can change without anyone editing the policy directly. Just creating a new Situation element may include the Situation in the policy if the Situation is associated with a Situation Tag or Situation Type grouping included in the policy. For more information about Situation elements, see [Situations](#) (page 183).

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.

The Permit action allows traffic that matches the traffic pattern and the Terminate action stops traffic that matches the pattern. A Permit action does not unconditionally allow the traffic because processing still continues to look for other patterns. However, 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 the logging level set to “None”. There is a second rule that contains Situation A below the first rule with Terminate as the action and the 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 a unique pattern (with the 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](http://www.example.com). A different rule specifies Termination for Situation B, which also matches [www.example.com](http://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.

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

Virtual Security Engines do not individually inspect traffic. One shared inspection process running on the Master Engine handles the inspection and correlation for all Virtual Security Engines associated with the Master Engine. To prevent excessive resource consumption on the Master Engine, take care when configuring Inspection policies for use on Virtual Security Engines.

## Exception Rule Cells

The table below explains briefly what each Exception rule cell does.

**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.
Severity	Limits the rule to 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	A list of matching criteria that defines the IP addresses and interfaces that the rule matches. The Source and Destination cells accept any elements in the Network Elements category, as well as User and User Group elements.
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 engine to carry out when a connection matches the rule. The action-specific Action Options define settings for anti-virus, anti-spam, connection termination and user responses. The Continue action can be used to set options for the Exceptions as explained in <a href="#">Setting Default Options for Several Inspection Exceptions</a> (page 128).
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	A free-form comment for the rule. You can also use comment rows to create sections in the rules.

**Table 11.1 Exception Rule Cells (Continued)**

Cell	Explanation
Rule Name	<p>Contains a rule tag and optionally a rule name.</p> <p>Name: <i>(Optional)</i> Name or description for the rule. Displayed alongside the rule tag.</p> <p>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.</p>

## Default Elements

There are several Inspection Policy elements, which are introduced when you import and activate a dynamic update package. The rules in the Inspection Policy templates may change when you activate new update packages. The table below lists the default Inspection Policy template elements.

**Table 11.2 Default Inspection Policy Elements**

Template	Description
No Inspection Policy	An Inspection Policy with a set of Inspection rules that do not enforce inspection.
Medium- Security Inspection Policy	<p>An Inspection Policy with a set of Inspection rules for detecting common threats. The Medium-Security Inspection Policy logs Situations categorized as Suspected Attacks but allows the traffic to pass.</p> <p>The Medium-Security Inspection Policy is suitable for Firewall, Layer 2 Firewall, Master Engine, and Virtual Security Engine deployments. It is also suitable for inline IPS deployments in asymmetrically-routed networks and IPS deployments in IDS mode. The risk of false positives is low in production use.</p>
High-Security Inspection Policy	<p>An Inspection Policy with a set of Inspection rules for detecting common threats. The High-Security Inspection Policy terminates Suspected Attacks with an alert. The High-Security Inspection Policy is suitable for Firewall, Layer 2 Firewall, inline IPS, Master Engine, and Virtual Security Engine deployments in which extended inspection coverage and strong evasion protection is required. The risk of false positives is moderate in production use.</p> <p>The High-Security Inspection Policy is suitable as the initial policy in most environments. The High-Security Inspection Policy terminates a connection if the engine cannot see the whole connection. It is recommended that you use the High-Security Inspection Policy as a starting point for your Inspection Policies.</p>
Customized High-Security Inspection Policy	<p>An Inspection Policy that is based on the High-Security Inspection Policy and contains a set of customized Inspection rules.</p> <p>The High-Security Inspection Policy is an example of a highly customized Inspection Policy for network environments in which unconditional inspection coverage and evasion protection are required. The risk of false positives is high in production use.</p> <p>The Customized High-Security Inspection Policy was used when the IPS was tested at ICSA Labs and NSS Labs.</p>

# Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.



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

## Task 1: Create an Inspection Policy

To customize inspection, you must have a custom Inspection Policy element. The pre-defined templates are a good starting point for your own customizations. Policy elements are discussed in [Firewall Policies](#) (page 83).

## Task 2: Activate Deep Inspection in Firewall Policies

Typically, you introduce deep inspection after creating and testing initial Access rules. You must specifically activate deep inspection for the portion of traffic that you want to deep inspect. This is done in the Access rules. See [Access Rules](#) (page 101) for more information. You also select which Inspection Policy is used for deep inspection on the Inspection tab of the Firewall Policy.

## Task 3: 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 and they are continuously updated through dynamic update packages.

The Rules tree on the Inspection tab 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 to prevent the definitions within the Rules tree from overlapping.

## Task 4: Define the Exceptions

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

- You can make exceptions to the general Rules tree definitions based on Source, Destination, and Protocol information.
- You can set options for connection termination (including User Responses) 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.
- You can create Continue rules to set Action Options and general rule 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 Inspection 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 Exceptions are those that you consider false positives in your environment (for example, Situations for exploit attempts against an operating system that is not used in your organization).

In the 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 malicious as successful attacks on a servers. This also creates unnecessary load on the engines, as a high number of Situations is checked in each matching connection.

## Task 5: Eliminate False Positives

As the Inspection rules and exceptions are matched to traffic, there are always some occurrences of false positives (matches that are incorrect or irrelevant in your environment). By tuning the Inspection Policy to the actual traffic and applications in your network environment, you can increase the relevance of inspection results greatly. To eliminate a false positive, you adjust either the Inspection 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 to use an existing log entry as the basis: you can create Exceptions through the right-click menu of log entries. See the [Eliminating a False Positive](#) (page 129) example for a practical overview of one approach to eliminating a false positive.

## Task 6: 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 184). You can add your custom Situations to the Rules tree by selecting a Situation Type for them.

### Setting Default Options for Several Inspection Exceptions

You may want to set default settings for some Exception 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 the same general way as in the Access rules. See [Configuring Default Settings for Several Rules](#) (page 111). In Exception rules, all settings in the Action Options and the Logging cell can be set using Continue rules. However, the Rules tree on the Inspection tab 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.

### Importing Snort Rules Libraries

You can import rule definitions from an existing Snort rules library (`.rules`) files. Importing a Snort rules library creates a new Inspection Policy. Each Snort rule is converted into a Situation element and an Exception rule in the Inspection Policy:

- The Action and Source/Destination Network parameters in the Snort rules are used to define the Exception rule.
- The Snort rule options are used to define the Situation element. The Situation element is used in the corresponding Exception rule.

The original Snort rule is included as a comment in the Context of the Situation. For more information about Situation elements, see [Situations](#) (page 183).



# Example of Inspection Rules

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

## Eliminating a False Positive

The administrators in this example have just started using inspection. They have installed a policy that includes only the rules defined in the Loose Inspection Policy. When they install the Firewall Policy, 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 specific server and a few clients in the internal network, so they quickly modify the Inspection Policies to exclude those particular hosts for the Situation in question. The administrators:

1. Create Host elements to represent the server and the clients.
2. Create a Group element that includes the client's Host elements.
  - The administrators name the Group so that it is immediately clear from the name that the Group contains those hosts that must contact the server running their custom-built application. This makes the new rule easier to read than if they included the hosts directly in the rule.
3. Add the following rule on the Exceptions tab in their Inspection Policy:

Table 11.3 Rule for Eliminating a False Positive

Situation	Source	Destination	Action	Logging
The Situation element that is mentioned in the alerts in the Logs view.	The Group defining the clients.	The Host for the internal server.	Permit	None

- If the Situation matches traffic between any other hosts than those included in the Group, the IP address does not match those defined in the new rule, so the processing will continue to the next rule, which terminates the traffic and triggers 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.
4. Refresh the policy on the Firewalls.



# NETWORK ADDRESS TRANSLATION (NAT)

Network address translation (NAT) means changing the IP address and/or port information in packets. Most often, NAT is used to allow internal hosts to communicate via networks where their actual address is not routable and to conceal the internal network structure from outsiders.

The following sections are included:

- ▶ [Overview of NAT](#) (page 132)
- ▶ [Configuration of NAT](#) (page 135)
- ▶ [Using NAT and NAT Rules](#) (page 139)
- ▶ [Examples of NAT](#) (page 142)

# Overview of NAT

---

*Network address translation (NAT)* changes the source or destination IP address or port for packets traversing the firewall. It is most often used to hide internal networks behind a single or just a few routable IP addresses on the external network. NAT is also often used to translate an external, routable destination address into the private internal address of a server. For destination NAT, port translation (sometimes referred to as PAT) is also possible when the protocol in question uses ports. Port translation can be used to redirect a standard service, such as HTTP (port 80/TCP), to a non-standard port (for example, port 8080/TCP). The NAT rules are stored in policy elements, which are discussed in [Firewall Policies](#) (page 83).

NAT is applied to traffic that has been already been allowed by Access rules that have connection tracking enabled. If you have Access rules that turn off connection tracking for some traffic, you cannot use address translation with those connections.

There are five possible methods for network address translation (these methods are explained in more detail in the next sections):

- *Static source translation*, which translates each single IP address to some other single IP address (one-to-one relationship).
- *Dynamic source translation*, which translates several IP addresses to a single IP address or a small pool of IP addresses (many-to-one/many-to-some relationship) differentiated by port. This method is not supported with Multi-Link if the Loose connection tracking mode is used.
- *Static destination translation*, which translates each single IP address to some other single IP address (one-to-one relationship).
- *Destination port translation*, which translates a port to a different one (one-to-one relationship).
- Both source translation and destination translation for the same connection.

*Dynamic destination translation* is done automatically as part of the Server Pool feature, see [Inbound Traffic Management](#) (page 235).

Additionally, when NAT is applied, *return address translation* is needed to allow reply packets to reach the correct sender or to show the source address that the destination host expects. However, return address translation does not normally need configuration as it is applied automatically with the help of connection tracking.

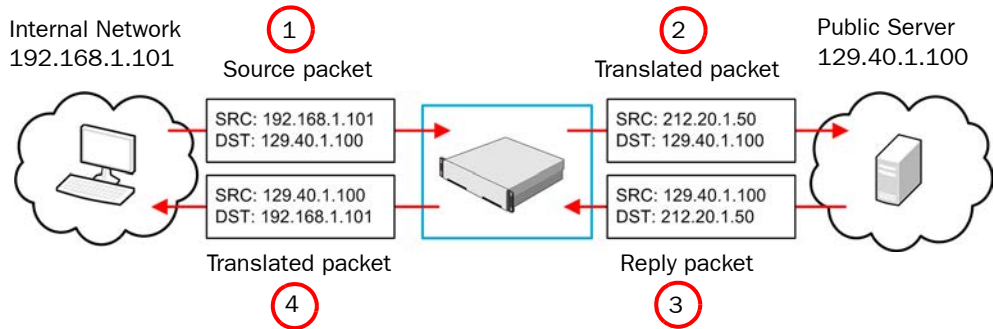
## Element-Based NAT

With element-based NAT, you select which elements have their own NAT address and define the NAT addresses for those elements. NAT rules are automatically generated in the Firewall Policy based on the NAT definitions created in the element properties.

# Static Source Translation

In static source translation (one-to-one source translation), the source IP address of a certain host is always translated using the same specific IP address. Often, the original source address is the actual assigned IP address for a device on an internal network or DMZ. The translation is then applied to a public IP address belonging to the public IP address range assigned by the Internet service provider (ISP).

**Illustration 12.1** Static Source Translation



In [Illustration 12.1](#), the packet starts out with the source (SRC) and destination (DST) addresses (1). The firewall replaces the source address of the packets with a new source address (2). Connection tracking information is used to automatically translate any reply packets: as the server responds, the destination address in the server's response (3) is replaced with the original address (4), ensuring that the responses find their way back to the host.

You can also define static translation using whole networks. There is still a fixed one-to-one relationship between each original and translated IP address, so the original and translated networks must be of the same size. The addresses map to their counterparts in the other network. For example, if you translate the network 192.168.10.0/24 to 212.20.1.0/24, the host 192.168.10.201 is always translated to 212.20.1.201.

## Dynamic Source Translation

Dynamic source translation allows translating a large number of original IP addresses to a much smaller pool of translated addresses, even a single IP address.

Dynamic source translation, sometimes referred to as *hide NAT*, is often used to mask the internal networks of a company behind one or a few public, routable IP addresses provided by an ISP.

**Illustration 12.2** Dynamic Source Translation

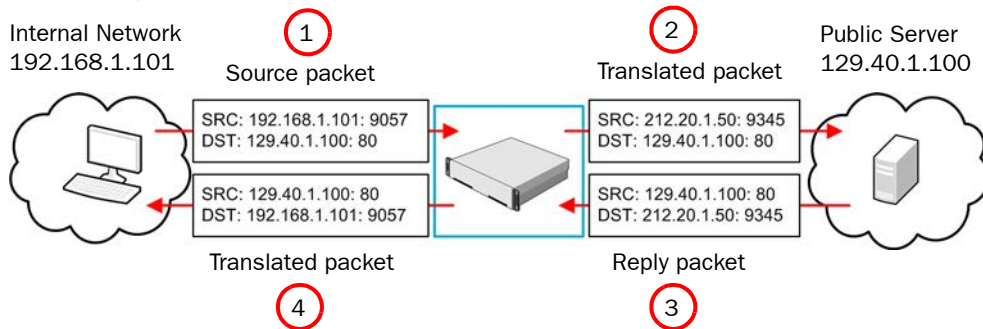
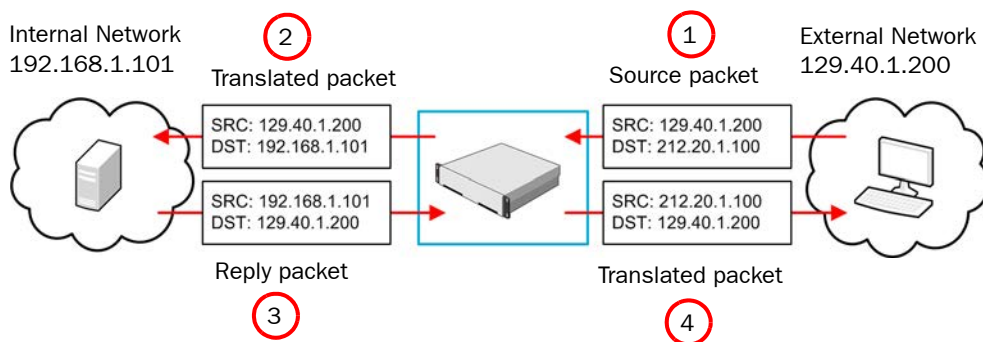


Illustration 12.2 shows the process for dynamic source translation. Because dynamic source translation involves multiple hosts using the same IP address (in a many-to-one or many-to-some relationship), the firewall needs some additional information to differentiate the connections when the reply packets arrive. For this, the firewall uses the source port: as each different host makes connections (1), it is assigned a unique port (one of the unreserved high ports) to track its connections (2). Based on the port used in the reply packets (3), the destination is translated to the original source address and port (4).

## Static Destination Translation

Most typically destination translation is needed when you have servers behind NAT to translate new incoming connections from the server's public IP address to the private one. You can use static destination translation for both IP addresses and ports.

**Illustration 12.3** Destination Translation



In the example in [Illustration 12.3](#), a host on the Internet connects to a server on the internal network. The host connects to the external, public IP address (1). The Firewall then translates the destination address to the private IP address of the server on the internal network (2). The server sends its response back (3), and the Firewall automatically translates the source address back to the external IP address (4).

You can also define static translation for whole same-size networks at once. This works in the same way as in static source translation.

## Destination Port Translation

Destination translation can also be used to translate ports. For example, web traffic to the corporate web servers on a DMZ would typically come in on port 80. However, an administrator may wish to run the web service on a non-standard port for security or network management reasons. The original destination port can be translated using static destination port translation with or without destination address translation.

## Configuration of NAT

Address translation is configured as part of the Firewall Policy using NAT rules. NAT rules are configured on the **IPv4 NAT** and **IPv6 NAT** tabs in Firewall Policy and Firewall Template Policy elements. Firewall Sub-Policies cannot contain NAT rules.



**Note – NAT rules are applied only after a packet matches an Access rule and is allowed by the firewall. The Access rule must have connection tracking enabled (default).**

**Illustration 12.4** Newly Inserted NAT Rule

ID	Source	Destination	Service	NAT	Used on	Comment	Rule Name	Hits
4.1	<None>	<None>	ANY		ANY		@653.2	
No NAT								

**Source, Destination, and Service** are used to match the rule to traffic. **NAT** cell defines how the translation is done. **Used on** makes the rule match on particular firewalls.

[Illustration 12.4](#) shows a NAT rule that has just been inserted into a policy. The **Source**, **Destination**, and **Service** cells are set to **<none>** and they must be changed to something else for the rule to match any traffic. The **Used on** cell is also used for traffic matching: you can add specific Firewall elements in this cell to make the rule valid only on those firewalls, or you can leave it to the default **ANY** to make the rule valid on all firewalls where the policy is installed.

The table below explains briefly what each NAT rule cell does (more information is included in the task flow later in this chapter). The columns are presented here in the default order, but you can drag and drop them to your preferred order in your own Management Client.

**Table 12.1 NAT Rule Columns**

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 4.3 is the third rule added in this Firewall Policy element to the insert point that is the fourth NAT rule in the upper-level Template Policy element.
Source	A list of matching criteria that defines the IP addresses and interfaces that the rule matches. The Source and Destination cells accept any elements in the Network Elements category, as well as User and User Group elements. Both the Source and the Destination defined must match the source and destination of a packet for the packet to match the rule. The addresses you insert must be valid for the address translation operation, for example, static source address translation requires that the Source cell contains a single continuous IP address space.
Destination	
Service	Allows limiting the rule's scope to a specific protocol (similar to Access rules). The Service cell accepts only Service elements.
NAT	The network address translation that is applied to connections that match the rule. You can also set outbound load-balancing parameters in this cell (see <a href="#">Outbound Load-Balancing NAT</a> (page 141)). If you leave this cell empty, address translation is not applied to matching connections, that is, the rule specifies that NAT is not to be applied to matching connections (to make an exception to the other NAT rules below).
Used on	The firewalls on which the NAT rule is applied. Used for creating NAT rules when a shared policy is used on several different firewalls. The Used on cell accepts only Firewall and Firewall Cluster elements.
Comment	Your free-form comment for this rule. Note that you can also add separate comment rows in between rules.
Rule Name	Contains a rule tag and optionally a rule name. Name: <i>(Optional)</i> Name or description for the rule. Displayed alongside the rule tag. Tag: <i>(Not editable)</i> An automatically assigned unique identifier 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, @180.2). 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.
Hit	<i>(Not editable)</i> Shows the number of connections that have matched the rule. This information is only shown if you have run a Rule Counter Analysis for the policy. The cell shows "N/A" if there is no information available about the rule.



# Considerations for Designing NAT Rules

The basic design principle of NAT rules is the same as in Access rules: the rules are read from the top down, and more specific rules must be placed above more general rules that match the same traffic. The traffic is matched based on the **Source**, **Destination**, **Service**, and **Used on** cells. The Source and Destination addresses in the cells must be valid for the address translation operation (the Source cell for source address translation and the Destination cell for destination address translation). When the first matching rule is found, the NAT defined for the rule is applied and the rest of the NAT rules are ignored. All NAT operations for the same connection must be defined in the same NAT rule (if you want to apply both source and destination translation to a connection).



**Note – NAT is applied after an Access rule has allowed the connection but before a routing decision is made. Make sure the Access rules allow the connection with the original (before NAT) addresses, and that the translated (after NAT) addresses are included under the correct interface in the Routing view, if necessary.**

If you use element-based NAT, the NAT rules generated from NAT definitions are applied only after the NAT rules that have been added manually to the policy. This means that the NAT rules that are generated from NAT definitions do not override the rules that you have manually added to the Firewall policy. Keep in mind, however, that a more specific NAT rule may prevent traffic from matching the automatically generated NAT rules.

## Default Elements

The Firewall Template contains NAT rules that exclude from address translation the communications between the firewall and the Management Server that controls it and the communications from the firewall to the Log Server where the firewall sends its log data. You must not use NAT rules to translate the addresses in these system communications, but define Locations and Contact Addresses instead. See [NAT and System Communications](#) (page 139).

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Define Source, Destination, and Service

**Source**, **Destination**, and **Service** are the main matching criteria that determine if a NAT rule is applied to a connection. These are defined in the same way as in Access rules.

## Task 2: Define Address Translation

If a connection matches the rule, the address translation defined in the NAT cell is applied. You can leave the NAT cell empty, if you do not want to apply NAT to any connections that match the rule. Otherwise, this cell allows you to define that the source address, the destination address, or both addresses are translated.

**Static** translation implies a one-to-one relationship. Therefore, in static source and destination translation, the translated address space must be as large as the original address space.

**Dynamic** source translation allows a many-to-one relationship, so that several hosts can use the same IP address. In dynamic translation, a port is reserved for each host that is communicating, so as many hosts can communicate simultaneously using a single IP address as there are ports in the port range you define. However, if the ports run out, translation cannot be done and some of the communications fail. If this happens, you may need to divide your dynamic translation rule and use an extra IP address for the translation. Naturally, dynamic translation can only be applied to communications that use ports (TCP and UDP based protocols).

## Task 3: Define the Firewall(s) that Apply the Rule

The **Used on** cell in the rule can be used to limit the scope of NAT rules based on the firewall where the rule is installed. This way, you can install the same policy on different firewalls and create NAT rules that take into account the different addressing in the different networks. If you leave the cell empty, the rule is applied on any firewall where the policy is installed.

## Task 4: Check Other Configurations

Translated IP addresses are used in routing, in VPN site definitions, and system communications. After adding or modifying NAT rules, you must consider how these areas of communications are affected and what changes are needed. If you are using Multi-Link, Outbound Multi-Links have their own NAT configurations that must not overlap with the NAT rules you define.

In particular, check that:

- Access rules and Inspection rules use the addresses that are seen in the packets as they arrive on the firewall (as they are before any NAT operation is done).
- Routing decisions are made after NAT, so the routing decision is made using the translated address. Make sure the translated address is included in the Routing view under the correct interface unless the packets are forwarded to the default gateway.
- If you translate addresses of communications going through VPN tunnels, the translated addresses must be included in the VPN site definitions.



**Note** – By default, NAT is disabled with connections traversing a VPN (NAT rules are completely ignored for VPN traffic). If you want the NAT rules to apply to connections traversing a VPN, enable NAT in the properties of the VPN element.

# Using NAT and NAT Rules

---

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

- [NAT and System Communications](#)
- [Outbound Load-Balancing NAT](#) (page 141)
- [Proxy ARP and NAT](#) (page 142)
- [Protocols and NAT](#) (page 142)

For general information on using rules, see [Using Policy Elements and Rules](#) (page 94).

## NAT and System Communications

If NAT is needed between SMC components, you must define *Contact Addresses* for the communications, so that the components use the correct (NATed) address for contact when needed.

Contact Addresses are used in a NAT environment for communications of SMC components with each other and with some external components that function as a part of the system (such as a RADIUS server used for authenticating Administrators). Contact Addresses may be needed also for gateway-to-gateway and client-to-gateway VPNs.

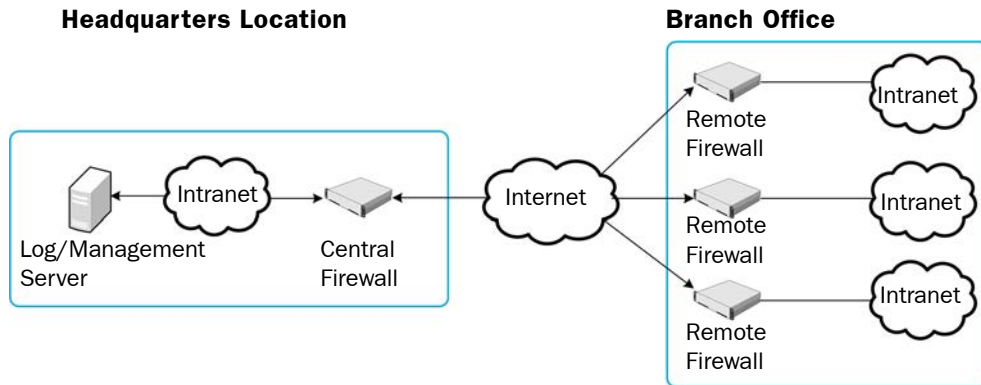
The Firewall Template includes NAT rules which define that NAT is not done for communications between the firewall where the policy is installed and the Management Server and Log Server that the firewall uses. If these connections require NAT, the configuration must be done as explained here. Other system communications traversing the firewall can be translated as any other connections (but Location and Contact Address definitions are usually still needed for those components so that they know the correct addresses to use with each other). See [Example of a Situation Where a Contact Address is Needed](#) (page 140) below.

Contact Addresses are defined for *Locations*, which is an element that represents all devices that are routable behind a particular interface of a NAT device. The components that need Contact Addresses are placed in the Locations according to the Contact Address they use.

## Example of a Situation Where a Contact Address is Needed

The following illustration demonstrates a scenario in which Contact Addresses are needed.

**Illustration 12.5 Contact Address Example**



In the illustration above, there are several remote firewalls that are managed through Management and Log Servers at a central site. NAT is typically applied at the following points:

- The central site firewall or an external router may provide the SMC servers external IP addresses on the Internet. The external addresses must be defined as Contact Addresses so that the remote firewalls can contact the servers across the Internet.
- The central firewall's IP address may be translated by an external router. The external IP address must be defined as a Contact Address to allow VPN connections from the remote firewalls to the central site using that address.
- NAT may also be applied at the remote sites (by external equipment) to translate the remote firewalls' IP address. In this case, you must define Contact Addresses for the remote firewalls so that the Management Server can contact them. The communications between the remote firewalls and the Management Server may also be reversed, so that the remote firewalls open the connections to the Management Server and maintain the connections open while waiting for commands.

When Contact Addresses are needed, a single Location to group all remote sites may be enough. The SMC servers' and the central firewall's definitions must include a Contact Address for the "Remote Firewalls" Location. However, if VPN communications between firewalls from different remote sites are allowed, it is necessary to create a Location for each remote firewall and to add further Contact Addresses for the firewalls.

## Contact Addresses and Locations

The Contact Address represents the translated address of a component. Contact Addresses are defined for each Location element. The Location element is a way to group components together, in effect telling them that there is no NAT device between them.

The SMC components on each side of a NAT device are grouped into two separate Location elements (if necessary, more Location elements can be used). The Contact Address is defined in each element's properties for the other Location. When contacting some other component in their own Location, the components always use the untranslated address. When contacting some component outside their own Location, the contacting component checks if the other component has a Contact Address defined for the contacting element's Location, and if found, it uses the Contact Address. If there is no Location-specific Contact Address defined, the contacting component checks if the element has a *Default Contact Address* that components belonging to any other Location use for contacting the element. If the element does not have a Default Contact Address, the connection is attempted using the element's untranslated address.

For example, when a Management Server contacts a firewall node through NAT, the Management Server uses the translated Contact Address instead of the firewall node's real Control IP address. The NAT device in between translates the NAT address to the firewall's real IP address as usual.

We recommend dividing elements into different Locations based on NAT and the communications the components have, and not just based on actual physical sites. For example, if there is one central site and several remote sites, and the system communications take place only from each remote site to the central site (not between the remote sites), only two Locations are needed no matter how many of the firewalls use a translated address.



**Note – If NAT is performed between a Log Server and a Management Client, you may need to select the correct Location for the Management Client as well.**

## Outbound Load-Balancing NAT

In addition to source and destination translation, another main use for NAT is *outbound load balancing*. It is used in a Multi-Link configuration where the Firewall balances outbound traffic between two or more network connections. To be able to direct traffic to the faster connection, the firewall translates outgoing connections to an address from a pool assigned to each available NetLink. In this case, the IP address(es) used for the NAT are defined in the properties of the Outbound Multi-Link element. Outbound traffic management using NAT with Multi-Link Technology is covered in detail in [Outbound Traffic Management](#) (page 225) and [Multi-Link Routing](#) (page 76).

## Proxy ARP and NAT

Proxy ARP (Address Resolution Protocol) is a specification that allows a device to respond to ARP requests on behalf of some other device on the network. When network address translation is used on a firewall, the firewall is typically configured to use proxy ARP so that it can accept packets for the translated addresses. The firewall uses proxy ARP instead of requiring the administrator to assign all of the translation addresses to the firewall's network interface.

In the Firewall, proxy ARP is handled automatically. Proxy ARP is enabled by default in the **NAT** cell in NAT rules for each translation type, although you have the option to uncheck it, if necessary. Automatic proxy ARP requires that the firewall is configured with an explicit route to the host in question (host/network added in the Routing view).

## Protocols and NAT

Protocols of the Protocol Agent type help with problems related to certain complex protocols and NAT. In some protocols, such as FTP, IP address information is included in the data payload of the packets, which do not normally include information for routing. Protocols of the Protocol Agent can examine the data payload of packets arriving to the firewall and also modify it. For example, when the source address is included in a packet's data, the firewall can translate the original source address and also the address embedded in the data. Protocols of the Protocol Agent are discussed in more detail in [Protocol Agents](#) (page 145).

## Examples of NAT

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

### Dynamic Source Address Translation

Company A uses private IP addresses that are not routable on the Internet in their internal network. The administrators need to translate the internal IP addresses to IP addresses that are routable on the Internet to make it possible to use external services. The administrators:

1. Create an Address Range element "External Addresses" for two consecutive IP addresses from the pool of addresses that they have been assigned by their Internet service provider.
2. Add a new NAT rule to their Firewall Policy:

**Table 12.2** Dynamic Translation Rule for Opening Connections to the Internet

Source	Destination	Service	NAT
"\$ Local Protected Sites" Alias	"NOT \$ Local Protected Sites" Expression	ANY	Source: Dynamic to External Addresses 1024-65535

- The administrators use the whole range of high ports (1024-65535) for translating the addresses in this case.
- Return address translation is done automatically, so a single rule suffices to cover all (client) hosts that only open connections themselves, and do not need to accept new connections coming from external networks.

3. Refresh the Firewall Policy. All internal addresses are now hidden behind two IP addresses and a range of ports.

# Static Address Translation

In the first example, Company A sets up the firewall to translate the IP addresses of all communications between the internal and the external network dynamically. However, the company also has a mail server, which must be able to accept connections from external networks. For this, it must have a fixed translated IP address. The administrators:

1. Create the Host element “Mail Server” to represent the mail server’s private IP address.
2. Create the Host element “Mail Server NAT” to represent the mail server’s public IP address.
3. Add two new NAT rules above the general dynamic translation rule.
  - In this case, new connections may be opened both from the mail server and from external hosts, so two rules are necessary.
4. Change the newly added NAT rules as follows:

Table 12.3 Static Translation Rules for Opening Connections Both Ways

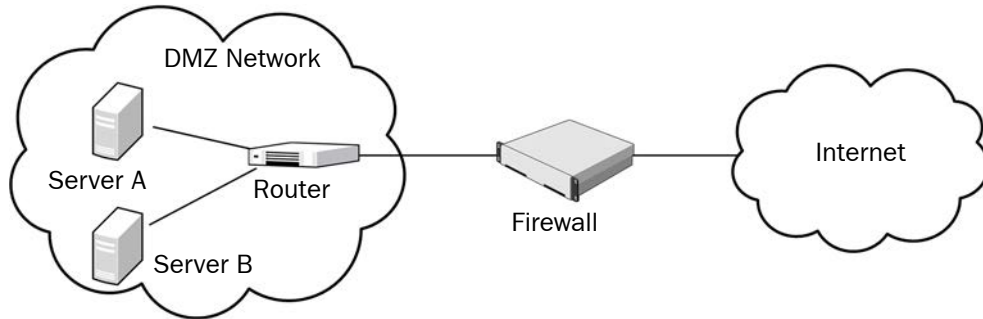
Source	Destination	Service	NAT
“Mail Server” Host element	“NOT \$ Local Protected Sites” Expression	“SMTP” Service element	Source: Static from Mail Server to Mail Server NAT
“NOT \$ Local Protected Sites” Expression	“Mail Server NAT” Host	“SMTP” Service element	Destination: Static from Mail Server NAT to Mail Server

- The first rule is for connections that the mail server opens to external hosts.
  - The second rule is for connections that external hosts open to the mail server.
  - Return address translation is done automatically, so if the connection would always be opened from one end, a single rule would suffice.
5. Refresh the Firewall Policy.

## NAT with Hosts in the Same Network

Company B has two servers running in the same DMZ network. The servers keep contact with each other to exchange some information. The administrators want to route the traffic through the firewall so that it is logged for reporting purposes instead of letting the servers communicate with each other directly.

**Illustration 12.6** Company B's Network Setup



The administrators first intend to just configure the servers to use the external (NAT) address of the other server as a destination and configure the related static destination NAT rule. However, they soon realize that the receiver would see the real source address in the communications and the replies would be sent directly, bypassing the firewall for the reply communications. This would obviously prevent the connections. A static source NAT is required in addition to the static destination NAT.

The administrators:

1. Create Host elements to represent the private addresses of the two servers.
2. Create Host elements to represent the public addresses of the two servers.
3. Add two new NAT rules before any other NAT rule that would match these connections:

**Table 12.4** Static Translation Rules for Opening Connections Both Ways

Source	Destination	Service	NAT
"Server A Private" Host	"Server B Public" Host	ANY	Source: Static from Server A Private to Server A Public Destination: Static from Server B Public to Server B private.
"Server B Private" Host	"Server A Public" Host	ANY	Source: Static from Server B Private to Server B Public Destination: Static from Server A Public to Server A private.

- When the servers are configured to contact each other using the public IP addresses, the communications are routed through the firewall.
- The Firewall translates the destination to the other server's private IP address and the private IP address of the source to the public IP address to "hide" the private source address from the receiving host. This way, the replies are sent to the public IP address and routed correctly through the firewall.



## CHAPTER 13

# 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 146)
- ▶ [Configuration of Protocol Agents](#) (page 147)
- ▶ [Using Protocol Agents](#) (page 149)
- ▶ [Examples of Protocol Agent Use](#) (page 154)

# Overview to Protocol Agents

---

Protocol Agents are software modules for advanced processing of protocols that require special handling on the Firewall due to their complexity, address information in the data payload, related connections, etc. Protocol elements also associate the traffic with a certain protocol for inspection against the Inspection Policy.

Protocol Agents on Firewalls can:

- Validate application-level protocol use (for example, FTP command syntax).
- Open related connections when required (for example, FTP data connections).
- Modify application data when required (for example, NAT in H.323 data payload).
- Redirect FTP, HTTP, and SMTP connections to content inspection servers.
- Proxy HTTP connections.

Some protocols always require the use of the correct Protocol Agent to pass inspection by the Firewall when the traffic is handled using stateful inspection.

## Connection Handling

When related new connections are opened based on information exchanged in an initial connection, Protocol Agents may be needed. Protocol Agents are provided to handle the following protocols:

- FTP with related active and passive data connections.
- H.323 conferencing protocol communications.
- Microsoft RPC (MSRPC) for Microsoft Exchange and Outlook communications.
- NetBIOS for the Windows NetBIOS datagram services.
- Oracle TNS protocol communications.
- Remote Shell protocol communications.
- SunRPC portmapper 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 firewall 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. Exactly how this works depends on the protocol in question.

A few examples:

- The FTP Protocol Agent can be set to strictly limit the allowed commands within the control connection to standard commands as listed in the FTP specifications. If other commands are sent in the control connection, the connection is dropped.
- The Oracle Protocol Agent can control the size of the Oracle TNS packets, or the location of the Listener service with respect to the database services.
- The SSH Protocol Agent can ensure that the SSH handshake is performed at the beginning of an SSH connection.

# NAT in Application Data

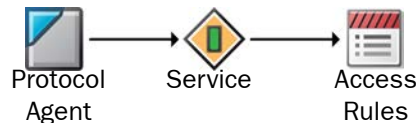
Protocol Agents on Firewalls can be used to assist with network address translation (NAT) in the application data. For example, the H.323 conferencing protocol includes the source and destination address information in the data payload of the packets (as opposed to 'normal' traffic where all IP address information relevant to the communications is in the spaces reserved for this in the packet headers). The H323 Protocol Agent can examine the data payload and modify the addresses according to the network address translation as needed. Therefore, when the source address is included in the protocol data, the source address is also translated in the data payload. The receiving machine then responds to the proper address.

## Configuration of Protocol Agents

---

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*.

**Illustration 13.1** Using Protocol Agents



Protocol Agents are not included directly in Firewall Policies. They are used inside custom Service elements that you create. The custom Service elements are used in Access rules. Whenever traffic matches a rule that contains a Service element with an associated Protocol Agent, the Protocol Agent is automatically activated.

All Protocol Agents are default elements, and you cannot change them or add any new ones. There are also default Service elements for most supported protocols that you can use to activate the Protocol Agents. However, some Protocol Agents have parameters and options you can set by creating customized Services as explained below.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create a Custom Service with a Protocol Agent

There are default Service elements 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 the way 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 that has configurable parameters, you can specify parameters for the Protocol Agent in the properties of the Service. The Protocol Agents are listed in [Using Protocol Agents](#) (page 149). See the *Management Client Online Help* or the *McAfee SMC Administrator's Guide* for information on the parameters for the Protocol Agents.

## 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 Firewall 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 cell is set to ANY. More specific Service definitions override the Continue rule, 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 the Connection Tracking option 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. The Firewall Template Policy contains a Continue rule that sets a Protocol Agent to be used with Services in the Service Group called *Default Services with Agents*.

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. 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.

## FTP Agent

One of the most common ways to transfer files across networks is using FTP. An FTP session starts with a control connection (by default, TCP port 21), and the data connection continues 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 Agent inspects protocol validity. There are two selectable levels of inspection: *loose* (default) and *strict*.

- In the loose mode, the Protocol Agent tries to identify information for opening the data connection. The loose mode is needed with some non-standard FTP applications.
- With the strict mode, protocol integrity can be enforced: all connections with commands that do not comply with the RFC 959 FTP standard are dropped.

The FTP Protocol Agent can modify payload data, if necessary. This may be required to handle NAT correctly.

The FTP Protocol Agent can also redirect traffic to an external content inspection server or to any proxy-type solution. For more information on how content inspection servers are used, see [External Content Inspection](#) (page 175).

The FTP Protocol Agent is platform-independent.

## GRE Agent

The Generic Routing Encapsulation (GRE) protocol is a tunneling protocol that allows the encapsulation of network layer packets inside IP tunneling packets. The GRE agent provides protocol inspection for tunneled GRE traffic. This agent specifies rematching parameters for GRE-encapsulated packets, and defines which traffic is tunneled. This agent has parameters you can set in the Service properties.

## GTP Agent

The GPRS Tunneling Protocol (GTP) is used to carry GPRS (general packet radio service) packets in GSM, UMTS, and LTE networks. The GTP agent provides protocol inspection for GTP traffic. There are no configurable parameters for this Protocol Agent.

## H323 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.

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

- H.323 may open several related connections, which places demands on access control and NAT. The H323 Protocol Agent's task is to keep track of the related connections that are opened within the same session. Particularly, if you want the Firewall to apply NAT to H.323 connections, you must ensure that the connections use this Protocol Agent.
- The H323 Protocol Agent examines the Call Signalling Channel (Q.931/H.225.0) connection and allows the related Control Channel (H.245) connection to open. It also examines the H.245 connection and allows further related connections (RTP and RTCP) to open, based on the port negotiations on the parent connection.
- When NAT is applied to the Q.931 connection, the Protocol Agent performs the same NAT to the related H.245 connection and modifies the payload data of the parent connection. The same NAT operation is performed also on the opened RTP and RTCP connections.

## HTTP Agent

The HTTP agent can be used for redirecting traffic to an external content inspection server and 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 decryption and inspection in the Access rules, and for identifying encrypted HTTPS traffic for inspection in the Inspection Policy. This agent has parameters you can set in the Service properties.

## MGCP Agent

The MGCP (Media Gateway Control Protocol) agent provides support for related RTP (Real-time Transport Protocol) connections in VoIP (Voice over IP) traffic. There are no configurable parameters for this Protocol Agent.

## MSRPC Agent

The MSRPC Protocol Agent is primarily meant for communications between Microsoft Outlook clients and a Microsoft Exchange server.

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. This 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.

If the traffic is Outlook/Exchange traffic, the Protocol Agent can also be used to support NAT for the related connections by modifying the payload data of the control connection accordingly.

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

## NetBIOS Agent

This Protocol Agent is used to allow Windows NetBIOS Datagram Service connections through the Firewall. 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 may go through a different route than the listener connection. You can create custom Oracle agents with different settings when required.

## RTSP Agent

The RTSP (Real Time Streaming Protocol) network control protocol is used for establishing and controlling media sessions between clients and media servers. The RTSP Protocol Agent allows RTP (Real-time Transport Protocol) and RTCP (Real-time Control Protocol) media streaming connections initiated with RTSP through the firewall and also handles NAT modifications to the protocol payload. This agent has parameters you can set in the Service properties.

## SCCP Agent

The SCCP (Skinny Call Control Protocol) provides support for related RTP (Real-time Transport Protocol) connections in VoIP (Voice over IP) traffic. There are no configurable parameters for this Protocol Agent.

## Services in Firewall Agent

This Protocol Agent is intended for system services running on Firewalls. It is only intended for the system's internal use. There are no configurable parameters for this Protocol Agent.

## Shell Agent

Remote Shell 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 on another computer. This agent has parameters you can set in the Service properties.

## SIP Agent

The Session Initiation Protocol (SIP) agent can be used with Firewalls to handle multimedia connections that use SIP as their transfer protocol (such as voice over IP).

Using the SIP agent allows SIP to be used across a Firewall that uses NAT. SIP uses TCP or UDP port 5060 to initiate the connection, after which the traffic is allocated a dynamically assigned port. The Protocol Agent keeps track of the actual ports used, so that the whole range of dynamic ports does not need to be allowed in the Firewall Policy.

The SIP agent can be configured to force the client and/or server address used within the SIP transport layer to also be used for the media stream carried over SIP (by default, this is enforced for both the client and the server).

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

## SMTP Agent

The Simple Mail Transfer Protocol (SMTP) Protocol Agent redirects connections to an external content inspection server. This agent has parameters you can set in the Service properties.

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

## SunRPC Agent

The Sun Remote Procedure Call (RPC) Protocol Agent only assists the Firewall in *Portmapper* connections. It makes the handling of RPC program numbers used in the Access rules faster. Only Portmapper connections going through the Firewall are assigned this Protocol Agent. This Protocol Agent is not intended for other communications.

The SunRPC Protocol Agent collects information about RPC services by interpreting the GET PORT and DUMP PORTS requests and their respective answers. All information it collects 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 reply information is stored in cache.

There are no configurable parameters for this Protocol Agent.



## TCP Proxy Agent

The TCP Proxy Agent is 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.

The handling of idle connections defined by the TCP Proxy agent is different from other connection handling on the Firewall, because without the Protocol Agent, idle connections are removed from the Firewall's records without sending any notices to the communicating parties (according to the general TCP timeout set in the Firewall properties, or an overriding timeout set in the rule that allowed the connection).

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

## 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). Apart from Access rules, the TFTP Protocol Agent is also useful in NAT operations.

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 in the control connection.

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

## Examples of Protocol Agent Use

The examples in this section illustrate some common uses for Protocol Agents 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 Firewall 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 the 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.
5. Refresh the policy on the IPS engine.

### 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:

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

- Using the “NOT Local Protected Sites” expression requires the Alias “Local Protected Sites” to be configured with a translation value for the Firewall in question.
2. Refresh the Firewall’s policy.

## CHAPTER 14

# TLS INSPECTION

The TLS Inspection feature decrypts TLS 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 156)
- ▶ [Configuration of TLS Inspection](#) (page 157)
- ▶ [Using TLS Inspection](#) (page 160)
- ▶ [Examples of TLS Inspection](#) (page 161)

## Overview to TLS Inspection

---

HTTPS is used to secure HTTP connections. When a browser connects to a server that uses HTTPS, the server sends a certificate that contains its public key and a digital signature from a certificate authority that verifies its identity to the browser. The browser and the server negotiate an encryption algorithm, which is used to create the encrypted connection.

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

The TLS Inspection feature consists of server protection, which inspects incoming connections to servers in the protected network, and client protection, which inspects TLS outgoing connections initiated by clients in the protected network. TLS Inspection requires two separate secure connections: one from the client to the firewall, and one from the firewall to the server.

When a TLS server in the internal network is the destination of an incoming connection, the firewall 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 TLS server, the firewall 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 a TLS server.

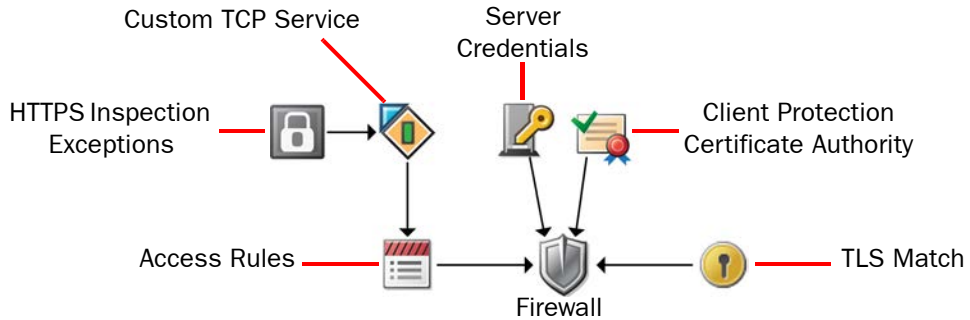
When a server's certificate is self-signed or has not been signed by a trusted certificate authority, the engine cannot trust the server certificate. In this case the engine makes a new self-signed certificate. This certificate is presented to the user in the internal network, and the user's browser shows the same warning it would show if it received a self-signed certificate directly from a TLS 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 SSL/TLS deep inspection. Substituting the original server certificate allows the firewall to decrypt and re-encrypt the traffic.

After decrypting the traffic, normal HTTP inspection and optionally virus scanning (requires separate license) are applied. If the traffic is allowed to continue, it is re-encrypted before forwarding it.

# Configuration of TLS Inspection

**Illustration 14.1** Elements in TLS Inspection



The Server Credentials and the Client Protection Certificate Authority are specified in the properties of the firewall that provides TLS Inspection. The firewall uses the private key and certificate stored in the Server Credentials to decrypt traffic to and from TLS servers in the protected network for inspection.

The Client Protection Certificate Authority contains a private key and a certificate. The firewall uses the private key stored in the Client Protection Certificate Authority to sign the certificates presented to the end-user, and the certificate to negotiate encrypted connections with TLS servers.

TLS Match elements define matching criteria for the use of the TLS protocol in traffic, and allow you to prevent specified traffic from being decrypted. TLS Matches that deny decrypting are applied globally, even if the TLS Match elements are not used in the policy.

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.

The Access rules define which traffic is decrypted and inspected. You can select specific traffic for decryption and inspection, or you can enable the decryption and inspection of all TLS traffic.

Once a certificate for client and/or server protection has been uploaded to the engine, it is possible to unintentionally enable TLS decryption for all traffic in one of the following ways:

- Adding an Application that allows or requires the use of TLS to an Access rule
- Enabling the logging of Application information in the Access rules
- Enabling Deep Inspection in an Access rule with the Service cell of the rule set to ANY.

## Default Elements

The **Default HTTPS Inspection Exceptions** element is an HTTPS Inspection Exceptions element that excludes domains used by the McAfee Security Management Center (SMC) and the engines from decryption and inspection. You cannot edit the Default HTTPS Inspection Exceptions element. If you need to make changes, you can duplicate the Default HTTPS Inspection Exceptions element and edit the copy.

The default **HTTPS (with decryption)** Service element enables the decryption of HTTPS traffic that uses the default port 443, excluding the domains that are specified in the Default HTTPS Inspection Exceptions. You cannot edit the default HTTPS (with decryption) Service element. If you need to make changes, you can duplicate the HTTPS (with decryption) Service element and edit the copy.

There are predefined Trusted Certificate Authority elements that represent the signing certificates of major certificate authorities. Default Trusted Certificate Authority elements are automatically added 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 Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create Server Credentials Elements

If you want to inspect TLS traffic for which an internal server is the destination, you must create a Server Credentials element to store the private key and certificate of the server. The private key and certificate allow the firewall to decrypt TLS 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 TLS traffic between a client in the internal network and an external 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 must configure users' 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: Exclude Traffic from Decryption and Inspection

Traffic to and from some servers that use TLS 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. You can optionally exclude traffic from decryption and inspection in two ways: globally with a TLS Match element, or for specific matching traffic with an HTTPS Inspection Exception element.

TLS Matches define matching criteria for the use of the TLS protocol in traffic, and allow you to prevent specified traffic from being decrypted. TLS Matches that deny decrypting are applied globally, even if the TLS Match elements are not used in the policy. However, TLS Match elements that are used in specific Access rules can override globally-applied TLS matches.

In most cases, TLS Matches are the recommended way to prevent traffic from being decrypted and inspected. Globally excluding domains from decryption may also prevent some Applications from being detected in encrypted connections. In this case, you can use HTTP Inspection Exceptions exclude the domain from TLS inspection.

HTTPS Inspection Exceptions are used in a custom HTTPS service to define a list of domains for which HTTPS traffic is not decrypted. The custom HTTPS service must be used in a rule, and only traffic that matches the rule is excluded from decryption and inspection. HTTPS Inspection Exceptions are primarily intended for backwards compatibility.

### Task 4: Activate TLS Inspection

In the Firewall properties, you specify the Client Protection Certificate Authority (if you want to inspect traffic between internal clients and external servers), and/or the Server 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.



**Note – Once a certificate for client and/or server protection has been uploaded to the engine, it is possible to unintentionally enable TLS decryption for all traffic by adding an Application that allows or requires the use of TLS to an Access rule, enabling the logging of Application information in the Access rules, or enabling Deep Inspection in an Access rule with the Service cell of the rule set to ANY.**

If the default HTTPS (with decryption) Service element meets your needs, you can use the default HTTPS (with decryption) Service element in the Access rules without modification. You must create a custom HTTPS Service in the following cases:

- You want to enable decryption for HTTPS traffic that uses a different port.
- You want to select a different HTTPS Inspection Exceptions element.
- You want to log the URLs in matching traffic.
- You want to modify any of the other settings in the Service Properties.

The Access rules define which traffic is decrypted and inspected. To select specific traffic for decryption and inspection, you create Access rules that enable Deep Inspection and use a custom HTTPS Service or the default HTTPS (with decryption) Service element. To enable the decryption and inspection of all TLS traffic, you enable Deep Inspection in an Access rule with the Service cell of the rule set to ANY. Traffic that matches the Access rule is decrypted and inspected in the same way as unencrypted HTTP traffic according to the Inspection rules. See [Access Rules](#) (page 129) 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 TLS communications mediated by the engine are decrypted for inspection, and because the private keys of the servers are stored in the Server 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 McAfee:

- 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 IP address is in a protected network.
- Save Management Server backups as encrypted files.

## Virus Scanning of Decrypted TLS Traffic

Once TLS traffic has been decrypted, virus scanning (separately-licensed feature) can be done in the same way as for regular traffic. Any traffic that is allowed to continue after virus scanning is re-encrypted and sent to its destination. For more information about how virus scanning works, see [Virus Scanning](#) (page 171).

## URL Filtering Decrypted TLS Traffic

Once TLS traffic has been decrypted, URL filtering (separately-licensed feature) can be done in the same way as for regular traffic. Any traffic that is allowed to continue after URL filtering is re-encrypted and sent to its destination. For more information about how URL filtering works, see [URL Filtering](#) (page 163).



# Examples of TLS Inspection

---

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

## Server Protection

Company A's server offers HTTPS services to their customers. The administrators want to be able to detect and block attacks targeting the HTTPS server, even if the attacks 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 Credentials element and import the private key and certificate of the HTTPS server.
2. Select the Server Credentials in the Firewall properties.
3. Create IPv4 Access rules with the default HTTPS (with decryption) Service as the Service.
4. Use the Medium-Security Inspection Policy to look for attacks in HTTP traffic, and check the HTTP traffic against the anti-virus signatures.
5. Save and install the policy.

## Client Protection

The administrators also want to detect and block -based attacks targeting the browsers of users in Company A's network to protect the workstations and internal networks. In addition to searching for attacks, the administrators also want to enable virus scanning. However, the 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 -based attacks that are encrypted inside an SSL tunnel, and use a TLS Match element to globally exclude the online banking domains from decryption and inspection.

The administrators do the following:

1. Create a Client Protection Certificate Authority element and generate a new certificate and private key. In their network environment, the administrators add the Client Protection Certificate Authority they created to the list of trusted certificate authorities in the users' browsers.
2. Select the Client Protection Certificate Authority in the Firewall properties.
3. Create a TLS Match element that prevents decryption when certificate validation succeeds for the domain names for the online banking sites that are excluded from decryption. Because the TLS Match is applied globally, the administrators do not have to use it in any specific rules.
4. Create IPv4 Access rules with the default HTTPS (with decryption) Service as the Service.
5. Use the Inspection rules from the Medium-Security Inspection Policy to look for attacks in HTTP traffic, and check the HTTP traffic against the anti-virus signatures.
6. Save and install the policy.



## CHAPTER 15

# URL FILTERING

URL filtering compares the URLs (uniform resource locators) 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 engine to respond in the various ways.

The following sections are included:

- ▶ [Overview to URL Filtering](#) (page 164)
- ▶ [Configuration of URL Filtering](#) (page 164)
- ▶ [Examples of URL Filtering](#) (page 166)

## Overview to URL Filtering

---

URL 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 URL filtering, the engines compare the URLs 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 otherwise want to block.

The URL categorizations are provided by the external BrightCloud service. 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. Category-based filtering with BrightCloud is a license-controlled feature.

The categories allow you to configure policies based on the types of sites to block instead of manually typing in URLs. The individual URLs included in the categories are updated continuously. The engines query the actual URLs from the external URL categorization service to access up-to-date URL listings. The individual URLs are not viewable in the Management Client except when a match is found in traffic and the match is logged.

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 firewall 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.

## Configuration of URL Filtering

---

**Illustration 15.1** Elements in the Configuration



The URL filtering feature is configured through McAfee-supplied URL Filtering Situations and/or manual URL lists. The Access rules and the Inspection Policy define how URL Filtering Situations are matched to traffic and what kind of reaction a match triggers. URL Filtering Situations can 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. The URL category names are updated through dynamic update packages.

## Default Elements

There are default elements for the categories you can use in URL filtering. These are represented by a specific type of Situation elements, which can be found under **Situations→By Type→URL 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 **Protocols→Application Protocols→HTTP** when selecting a Context for a Situation).

The Situations that represent URL 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 Situation icon.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Prepare the Firewall

URL filtering is part of the deep inspection features on firewalls. Category-based URL filtering requires that the engine is licensed to use the BrightCloud categorization service. You must also define DNS server addresses in the Firewall element so that the engines can contact the BrightCloud servers.

### Task 2: Create User Response Messages

Optionally, you can define customized User Responses for URL 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 Policy.

## Task 4: Configure URL Filtering Rules

The Access rules and the Inspection Policy define how URL Filtering Situations are matched to traffic and what kind of reaction a match triggers.

Category-based URL filtering can be configured in the IPv4 or IPv6 Access rules, or in the Inspection rules. In the Access rules, category-based URL filtering is configured as part of the matching criteria in the Service definition.

URL filtering based on URL lists can be configured in the Inspection Policy. Firewalls do not deep inspect traffic by default, so there must be an Access rule that matches HTTP traffic and has Deep Inspection enabled.

Different URL filtering features require you to adjust either the main Inspection Rules tree or the Exceptions. The URL 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. See [Inspection Policies](#) (page 119) for more information on Inspection rule configuration.

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 URL Filtering

---

### Allowing a Blocked URL

The company is using category-based URL 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 “URL 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 15.1** New Rule for Allowing a URL Above the Previously Added Category-Based Rule

Situation	Source	Destination	Action
Custom “URL Filtering Whitelist” Situation	Administrator’s workstations	ANY	Permit

## CHAPTER 16

# SPAM FILTERING

Spam filtering inspects incoming e-mail traffic for spam. If a spam e-mail is found, you can configure the system to mark the message as spam, or to reject or discard the message.

The following sections are included:

- ▶ [Overview to Spam Filtering](#) (page 168)
- ▶ [Configuring Spam Filtering](#) (page 168)
- ▶ [Using Spam Filtering](#) (page 169)

# Overview to Spam Filtering

---

Spam filtering allows a Firewall/VPN engine to inspect SMTP protocol traffic for spam to protect your mail servers from spam attacks. Only IPv4 traffic is supported in spam filtering. Spam filtering is available as a separately licensed feature on selected platforms. Spam filtering is not supported on Master Engines or Virtual Security Engines.

You can freely configure the criteria for filtering e-mail traffic. In the spam filtering process, each e-mail is assigned a spam score. If the spam score is low, an e-mail is considered to be legitimate. If the spam score is high, the e-mail can be marked as spam, rejected, or discarded depending on your settings.

## Configuring Spam Filtering

---

### 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* and the *McAfee SMC Administrator's Guide*.

#### Task 1: Define Spam Filtering for a Firewall

To define spam filtering, your firewall's license must include the feature. The anti-spam settings in the Firewall element properties allow you to define different spam filtering options.

#### Task 2: Select Traffic for Inspection with Access Rules

Activate spam filtering in the IPv4 Access rule Action Options. When you activate spam filtering, deep inspection is automatically activated for the same traffic (the traffic is also checked against the Inspection rules). You can deactivate spam filtering for trusted hosts in the IPv4 Access rule Action Options. You must define the SMTP service individually in the Service cell of each IPv4 Access rule in order to enable deep inspection and anti-spam filtering for this service.

#### Task 3: Select Traffic Not to Be Filtered

For some traffic spam filtering may not be feasible. For example, you may want to exclude traffic between your local mail servers from spam filtering to avoid unnecessary use of resources. To prevent spam filtering for certain destinations, you can create a more specific IPv4 Access rule before a more general one.



## Anti-Spoofing and Anti-Relay Protection

E-mail address spoofing is a technique used by spammers to obtain sensitive information. In e-mail address spoofing parts of the header of an e-mail are forged to make the message appear as though it originates from someone other than the original sender. Anti-spoofing and anti-relay options allow you to detect spammer activity and to stop suspicious e-mails. To protect your network from spoofing, you can specify your local network domains in the spam filtering settings. This allows the firewall to detect the messages that contain spoofed e-mail addresses. The firewall checks the domain information specified in the following parts of an e-mail message:

- Domain information in the HELO/EHLO command.
- Domain information in the MAIL FROM command.
- Domain information in the From field of an e-mail header.
- Relay information in the RCPT TO command.

If an external e-mail contains your local domain information it is considered to be spam. You can adjust the anti-spoofing and anti-relay options to discard, reject or score such messages.

## Handling E-mail Address Forgery

You can detect forgery of sender e-mail addresses by using *SPF* (Sender Policy Framework) and *MX* (Mail Exchanger) record matching. SPF protects the envelope sender address that is used for delivering e-mail messages. The method allows domain owners to specify in an SPF record a mail sending policy that indicates which mail servers they use to send e-mail from their domains. The SPF record is then published in the Domain Name System. Mail exchangers use SPF records to check if an e-mail is sent from a legitimate server. An MX record is a type of record published in the Domain Name System that specifies a mail server responsible for accepting e-mail messages for a certain domain. MX records are used to direct a domain's mail flow to the correct servers. We recommend that SPF and MX record matching is used when traffic is not routed through a proxy or a VPN gateway.

## Spam Filter Sensitivity Settings

Each incoming e-mail message that passes spam filter checks is assigned a spam score which determines the likelihood of its being spam. By default, a spam label is added to the headers of all e-mails with the score of 2 and above, and all e-mails with the score of 8 and above are rejected. You can adjust the score values that determine when e-mail messages are marked as spam or rejected in the scoring settings.

## Spam Filtering Rules

You can define separate spam filtering rules for different parts of an e-mail message:

- An *Envelope Rule* inspects data in the envelope of an e-mail.
- A *Header Rule* inspects data in the header of an e-mail.
- A *Content Rule* inspects content in the body of an e-mail.

The rules allow you to detect specific word patterns and regular expressions in e-mail messages, and to define how such messages are handled. You can create various rules to handle e-mails for different recipients differently. For example, you can create Envelope rules per recipient to have milder rules for marketing or PR divisions, and stricter rules for other employees. The table below shows an example of an Envelope rule. The rule increases the credibility of all e-mail that is sent to specified recipients. A negative score value decreases the overall spam score of an e-mail and makes the e-mail less likely to be spam.

**Table 16.1** Example Envelope Rule

Field	Value	Action
<Envelope>Rcpt To	E-mail addresses of employees working in marketing or PR divisions.	Score - 5

Spam filtering rules allow you to save system resources because if a message matches a specific rule, further processing may not be necessary. For example, you might create a Header Rule that blacklists e-mail messages if the content in the header is written in simplified Chinese.

**Table 16.2** Example Header Rule

Field	Value	Action
<Header Rules>Content-type	<Regular Expression>/gb2312/i	Blacklist

## DNS-Based Blackhole Lists

*DNS-based Blackhole Lists (DNSBLs)* are lists of IP addresses of computers or networks that are suspected of sending spam. They are published in the Domain Name System. There are two types of lists that you can define to be checked: *RBLs* and *URI DNSBLs*. A Real-Time Blackhole List (RBL) contains URLs of DNSBLs that list IP addresses of servers that are responsible for sending spam or that are hijacked for spam relay. A Uniform Resource Identifier DNSBL (URI DNSBL) contains URLs of DNSBLs that list domain names and IP addresses of links found in the body of spam e-mails.

## CHAPTER 17

# VIRUS SCANNING

A virus scanner compares network traffic against an anti-virus database to search for viruses. If a virus is found, infected traffic is stopped or infected content is stripped out.

The following sections are included:

- ▶ [Overview to Virus Scanning](#) (page 172)
- ▶ [Configuration of Virus Scanning](#) (page 172)
- ▶ [Using Virus Scanning](#) (page 173)

# Overview to Virus Scanning

---

A virus scanner is available as a separately licensed feature on selected platforms. Virus scanning is not supported on Master Engines or Virtual Security Engines.

Virus scanning is a resource-intensive activity and is practical mainly in branch-office-type settings, where there is a need to keep the physical setup as simple as possible with the minimum amount of equipment on-site.

The virus scanner can inspect IPv4 traffic. The supported protocols are HTTP, HTTPS, IMAP, POP3, and SMTP. If the virus scanner detects infected files, it strips them out. If an e-mail attachment is filtered out, a message is added to the e-mail notifying the recipient.

Virus scanning is alternatively available (on all Firewall/VPN engines) when you set up an external virus scanner and integrate it with the Firewall by configuring an external server as a content inspection server (CIS). See [External Content Inspection](#) (page 175) for more information.

## Configuration of Virus Scanning

---

### 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* and the *McAfee SMC Administrator's Guide*.

#### Task 1: Activate the Anti-Virus Feature for a Firewall

To activate virus scanning, your firewall's license must include this feature. The anti-virus settings in the Firewall element properties allow you to add mirrors for downloading updates to the anti-virus database, to set a schedule for downloading updates to the anti-virus database, and to change the settings for logging the viruses found in network traffic.

#### Task 2: Select Traffic for Inspection with Access Rules

Activate anti-virus scanning in the IPv4 Access rule Action Options. Activating the anti-virus scanning always also activates deep packet inspection for the same traffic (the traffic is also checked against the Inspection rules). You can deactivate the anti-virus scanning in the IPv4 Access rule Action Options if the download source is trusted and the download process appears to take too long. You must define the services individually in the Service cell to enable deep inspection and anti-virus scanning for them. To prevent anti-virus scanning for certain destinations you can create a more specific IPv4 Access rule before a more general one that does not have the anti-virus option defined.

#### Task 3: Define the Content Not to Be Scanned

For some content delivered through the HTTP or HTTPS protocol, the anti-virus scanning may not be feasible. For example, you may want to prevent videoconferences from being scanned for viruses to avoid any increase in latency. Exceptions to scanning can be made by matching the traffic with an Inspection rule that disables anti-virus in its options.

### Integrated Scanning vs. Content Inspection Server

In branch-office-type environments where there may be no skilled administrators, a centrally-managed virus scanning solution on the same hardware with the firewall makes maintenance easier than having separate equipment. Virus scanning is needed when there is direct Internet connectivity at the site (instead of only VPN connectivity to a central site where the traffic can be scanned centrally).

However, virus scanning directly on the firewall is not practical in high-traffic environments. The amount of data gathered for virus scanning is large, since files must be inspected as a whole to prevent any part of the infected content from passing through. Storing and scanning files significantly increases the demand for resources as the volume of traffic grows. In high-traffic environments, a separate content inspection server (CIS) integrated with the firewall is a more economical and flexible solution than a UTM device. For additional information, see [External Content Inspection](#) (page 175)

### Limitations of Virus Scanning on Clusters

Firewall/VPN clusters that are correctly licensed can be used for virus scanning. However, there are some restrictions that apply. Since the data being inspected is not synchronized between the nodes, connections that are undergoing virus scanning at the time of a fail-over are dropped when the fail-over occurs and must be reopened by the applications.



## CHAPTER 18

# EXTERNAL CONTENT INSPECTION

Content inspection means analyzing traffic for malicious content. You can integrate an external content inspection server with the firewall.

The following sections are included:

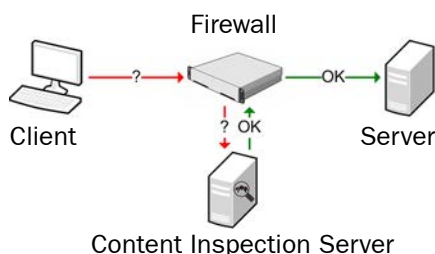
- ▶ [Overview to Content Inspection](#) (page 176)
- ▶ [Configuration of Content Inspection](#) (page 177)
- ▶ [Using Content Inspection](#) (page 179)
- ▶ [Example of Content Inspection](#) (page 180)

# Overview to Content Inspection

Content inspection allows you to inspect the FTP, SMTP, and HTTP protocols in IPv4 traffic for malicious content. Content inspection includes a wide range of ways to check traffic - many of which you can use in a simpler way by integrating an external content inspection server with your firewall. The integration involves setting up your firewall to redirect traffic to an external content inspection server. The main benefit in using the firewall to redirect traffic to a separate content inspection server is that the redirection works transparently: the communicating hosts need no additional proxy configuration when the redirection is done for them at the firewall.

Content inspection servers are most typically used for virus scanning and content filtering, but the available applications are not limited to those. Using an external content inspection server allows you to expand the capabilities of the firewall with virtually any type of content screening to perform tasks, for example, stripping certain types of attachments out of e-mail messages without blocking the message itself. This type of anti-virus checking is available directly on the Firewall as well, but an external content inspection server is a better option in medium to high throughput environments (see [Integrated Scanning vs. Content Inspection Server](#) (page 173) for some guidelines).

**Illustration 18.1** Content Inspection Server Redirection



The illustration above shows how a client's connection to a server is redirected from the firewall to the content inspection server. Connections arriving at the firewall are checked against the firewall's policy. Access rules determine which connections are redirected to the defined content inspection server for inspection. The content inspection server then handles the traffic according to its policies. Finally, the content inspection server opens a connection through the firewall and onwards to the original destination. Replies are received with the content inspection server's IP address, so those are also redirected to the content inspection server for screening.

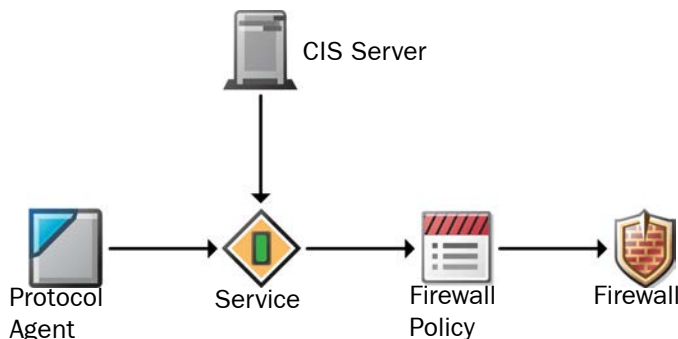
The content inspection server is used as a transparent proxy, so the client and server are not aware of the redirection and they require no additional configuration. The firewall uses NAT (network address translation) to forward the connections to the external content inspection server.



# Configuration of Content Inspection

FTP, SMTP, and HTTP traffic can be redirected to content inspection servers for inspection. This is done with the help of Protocol Agents.

**Illustration 18.2** Elements for Content Inspection Server Redirection



The illustration above shows how content inspection server redirection is configured. A custom Service element and a content inspection server (CIS) element are needed. A Service redirects connections for content inspection when one of the existing default Protocol elements of the type Protocol Agent is attached to the Service. The Service element contains a parameter that defines which content inspection server inspects the connection. The Service can be inserted to any number of Access rules in the Firewall Policy to select traffic to be redirected to the content inspection server. There can be several different Services for content inspection server redirection, if you have several content inspection servers.

The Protocol Agent redirection allows using the content inspection server as a transparent proxy, thus requiring no additional configuration on the client machines. The redirection is fully transparent to both the client and the server. The Protocol Agent translates the destination address automatically to the content inspection server's address to redirect the traffic. The content inspection server then functions as a proxy by establishing the forward connection to the actual destination address.

In addition to translating the real destination address to the content inspection server address for redirection, the source address is also translated for handling the content inspection server redirection on the firewall. The translated source address can be any address that is routed back from the content inspection server to the firewall (so that replies are correctly handled). Further address translation can be applied to the connection from the content inspection server to the communications destination.

## Default Elements

A Protocol of the type *Protocol Agent* is needed for content inspection server redirection. All Protocol Agents are always default elements. There are three Protocol Agents that can redirect connections to a content inspection server:

- **FTP** for file transfer protocol file transfers.
- **HTTP** for hypertext transport protocol connections used in web browsing.
- **SMTP** for simple mail transfer protocol for e-mail.

Additionally, the default Services for the three supported protocols can be used when allowing connections that the content inspection server opens for the redirected traffic.

## Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create a CIS Server Element

The CIS Server element defines the content inspection server's IP address and the services and ports that the content inspection server handles.

### Task 2: Create a Custom Service for Content Inspection Server Redirection

A custom Service element is always needed for content inspection server redirection. When you attach a Protocol that supports content inspection server redirection, you can activate the feature by attaching the correct CIS Server element to the Service.

The (mandatory) source address translation for the redirected connections is defined in the Service - these addresses are what the content inspection server sees as the connection originator. Any additional address translation for the traffic must be configured for the connections from the content inspection server when the proxy connections are opened through the firewall.

### Task 3: Define Access Rules for Redirection

You define the connections that you want to redirect to a particular content inspection server in the IPv4 Access rules. To activate redirection, you use a custom Service element. The redirection rule must always have connection tracking on, since connection tracking is required for the mandatory address translation. Both incoming and outgoing connections can be redirected.

Two Access rules may be needed for content inspection server redirection:

- One rule for the original connection from the client.
- A second rule for the new connection that proxy-type content inspection servers open to the actual destination.

## Task 4: Configure NAT Rules for Content Inspection Server Redirection

NAT rules must usually be adjusted for content inspection server redirection. The Protocol Agent handles the translation of addresses when forwarding the traffic from the firewall to the content inspection server and there must not be overlapping address translation defined for these connections in the NAT rules. A matching rule with an empty NAT cell may be needed to ensure other NAT rules do not match to these connections.

Any NAT you want applied to proxy connections opened by the content inspection server must be done using normal NAT rules.

## Using Content Inspection

The main benefit in redirecting traffic to a separate content inspection server is that it works transparently: the communicating hosts need no additional proxy configuration when CIS redirection is used. However, the redirection uses NAT, which may sometimes cause problems if you want the CIS to treat traffic differently based on IP address. In such cases, it may be necessary to configure the CIS server traditionally on the clients instead of using the Firewall's redirection feature.

To illustrate how the connections are handled, the following table shows an example of the source and destination IP addresses that are used in redirected connections at different stages. This example may be useful when planning the Access rules. The client's translated IP address in the redirection must be different from the public translated IP address normally used for the client's Internet connections. There are two different connections: one connection between the client and the content inspection server, and a second connection between the content inspection server and the server that is the target of the client's connection.

**Table 18.1** Addresses Used in Content Inspection Server Redirection

Communication	Source IP Address	Destination IP Address
Client to firewall	Client's private IP address	Target server's public IP address
Firewall to CIS	Client's translated IP address	Content inspection server's private IP address
CIS to firewall	Content inspection server's private IP address	Target server's public IP address
Firewall to server	Content inspection server's public IP address	Target server's public IP address
Server to firewall	Target server's public IP address	Content inspection server's public IP address
Firewall to CIS	Target server's public IP address	Content inspection server's private IP address
CIS to firewall	Content inspection server's private IP address	Client's translated IP address
Firewall to client	Target server's public IP address	Clients private IP address

This scenario requires two Access rules (one for each connection) and one NAT rule (for the connection between the content inspection server and the target server). To see how these types of communications are reflected in firewall Access and NAT rules, see [Example of Content Inspection](#) (page 180).

Alternatively, anti-virus features (separately licensed feature) are available directly on the Firewall for low-traffic environments. See [Virus Scanning](#) (page 171) for more information. Also, limited URL filtering is available as part of the deep packet inspection features (hardware permitting).

## Example of Content Inspection

The example in this section illustrates a common use for content inspection and general steps on how the scenario is configured.

### Inspecting Internal User's Web Browsing and File Transfers

The example company has decided to screen out non-work-related connections using an external content inspection server that can screen the HTTP and FTP connections that the company's employees open to the Internet. The content inspection server acts as a proxy for these connections. The administrators have already installed the content inspection server and configured it to process HTTP and FTP traffic according to the company's policy. To configure the redirection, the administrators:

1. Create an element for their content inspection server.
2. Create a custom Service element for both HTTP and FTP that refer to the Protocol Agents for those protocols.
3. Add the content inspection server to the Protocol Agent parameters in the Service properties.
4. Create the Access rules for redirecting connections to the content inspection server and the connections that the proxy server then opens to the Internet or any other destination.

ID	Source	Destination	Service	Action
14.1	Internal Network element	ANY	HTTP-CIS-Redirect	Allow
14.2	CIS Server element	ANY	HTTP	Allow
14.3	Internal Network element	ANY	FTP-CIS-Redirect	Allow
14.4	CIS Server element	ANY	FTP	Allow

- The table above shows rules for redirecting outgoing HTTP and FTP traffic through a content inspection server. Connections opened from the corporate LAN are redirected to the content inspection server in rule 14.1 and 14.3. The content inspection server then connects to the actual destination, which is allowed in the rules 14.2 and 14.4.
- The FTP Service without redirection in rule 14.4 also uses a Protocol Agent, as this is required for the FTP connections to be correctly handled by the firewall. However, this is the default element that is not configured for content inspection server redirection.

5. Create the NAT rules for ensuring that NAT rules do not match to connections that are being redirected to the content inspection server and rules for translating the connections opened by the content inspection server for load-balancing the connections across the company's Multi-Link network connections.

ID	Source	Destination	Service	NAT
4.1	Internal Network element	ANY	HTTP-CIS-Redirect	
4.2	CIS Server element	ANY	HTTP	Dynamic Load Balancing: HQ Multi-Link
4.3	Internal Network element	ANY	FTP-CIS-Redirect	
4.4	CIS Server element	ANY	FTP	Dynamic Load Balancing: HQ Multi-Link

- Rules 4.1 and 4.3 ensure that there is no address translation for the traffic that is redirected to the content inspection server (NAT cell is empty, which means no NAT is done for matching connections).
- Rules 4.2 and 4.4 are used to NAT the forward connections from the content inspection server to the actual destination, in this case, dynamic source NAT for load-balancing the traffic across available ISP links (using the Outbound Multi-Link element, which in the example company has been given the name “HQ Multi-Link”).



## CHAPTER 19

# 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 184)
- ▶ [Configuration of Situations](#) (page 184)
- ▶ [Using Situations](#) (page 189)
- ▶ [Example of Custom Situations](#) (page 189)

# Overview to Situations

*Situations* define the traffic patterns and events you want to detect in the Inspection Policy. 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.

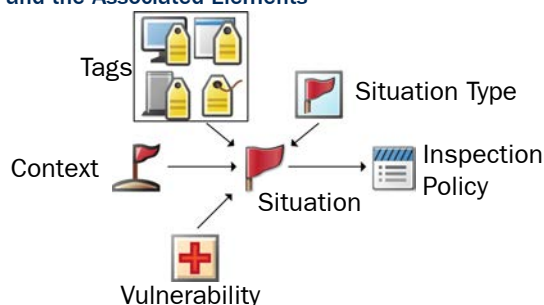
Situations also provide a description that is shown in the logs, and a link to relevant external information (CVE/BID/MS/TA) in the form of a *Vulnerability* element attached to the Situation.

The Inspection Policy defines how the Situations are matched to traffic and what kind of action the engine takes when a match to a particular Situation is found. Correlation Situations are a special type of Situations that group together event data to find patterns in that data.

## Configuration of Situations

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

**Illustration 19.1 A Situation and the Associated Elements**



The Situation element uses different elements to form a representation of the traffic that you want to detect in your Inspection Policy. The purpose of these elements is as follows:

- 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 the branch of the Rules tree under which the Situation appears (Attacks, Successful Attacks, etc.). One Situation Type can be associated with each Situation.
- The **Context** element defines the traffic patterns the Situation detects. 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 is recommended to consistently associate all relevant Tags with each custom 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

Context elements are protocol-specific, so they define what the Situation element matches. 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. The properties of each Context provide assistance on filling in the parameters for the Contexts.

The sections below explain the types of Context elements available and how they can be configured.



**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 for matching groups of related events in traffic. There are five types of Correlation Contexts:

**Table 19.1** Correlation Context Types

Correlation Context Type	Description
Compress	Combines repeated similar events into the same log entry, reducing clutter in the Logs view. <b>Example:</b> There is 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 how many times certain Situations occur within the defined period, so that action can be taken if the threshold values you set are exceeded. <b>Example:</b> 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. <b>Example:</b> 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 19.1 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.</p> <p><b>Example:</b> 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 may allow 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 341).

## Anti-Virus Contexts

The Anti-Virus Contexts are used to detect viruses in HTTP, HTTPS, IMAP, POP3, and SMTP protocol traffic. You must have a license that supports Anti-virus inspection to be able to use these Contexts for traffic inspection. Anti-virus inspection is not supported on Master Engines or Virtual Security Engines.

## 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 detecting attempts to scan which IP addresses are in use or which ports are open in your systems.

## Protocol-Specific Contexts

The protocol-specific Contexts are used 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 web sites.

For Contexts that have particular values to be filled in (instead of a regular expression), the parameters you define in the 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. In some cases, you may define what the Situation *does not* match.

Effective modifications to the protocol-specific Contexts require you to be 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 the McAfee Security Management Center (SMC), 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 changes whenever you update your system with new definitions. Both Situation elements 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.

## Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create a Situation Element

You can create new Situations in addition to using the predefined ones. You can create a Situation element to detect individual events or a Correlation Situation element to detect a group of related events. Situation elements can also be defined automatically based on Snort rules when you import a Snort rules library. See [Importing Snort Rules Libraries](#) (page 128) for more information.

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 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 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 SMC regular expressions is explained in [Regular Expression Syntax](#) (page 345).

The Correlation Situation parameters are explained in [Situation Context Parameters](#) (page 341).

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 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 Rules and Exceptions of the Inspection Policy. 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 Rules tree. See [Inspection Policies](#) (page 119) for more information.



**Note** – If a Tag or Situation Type you add to a Situation is in use in some Inspection Policy, the new Situation is automatically included in the policy when you save the Situation, and the engines 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 McAfee 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 Policy. Situations are generally used for:

- Detecting malicious patterns in traffic. The Situations supplied by McAfee in dynamic update packages concentrate on such known vulnerabilities and exploits.
- Reducing the number of alert and log entries you receive (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 Inspection 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.

## Example of Custom Situations

---

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

### Detecting the Use of Forbidden Software

Company A has a Firewall that inspects all outgoing web traffic against the Inspection Policy. The use of instant messaging clients across the Internet is forbidden in the company. The Inspection Policy is set to detect and log Situations with the **Instant Messaging** Tag.

The company's administrators have found out that some of the internal users have started chatting using a new little-known instant messaging client that does not have a default Situation yet. The communications seem to be standard HTTP directly from client to client. The administrators find one 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 Request URI** Context to the Situation and type in a regular expression that contains the address they want the Situation to find using the SMC regular expression syntax (see [Regular Expression Syntax](#) (page 345)).
3. Add the default system Tag **Instant Messaging** to the Situation.
4. Refresh the Firewall's policy.
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 to remove the software from the computers shown in the logs.



## CHAPTER 20

# APPLICATIONS

Application elements collect together combinations of identified characteristics and detected events in traffic to dynamically identify traffic related to the use of a particular application.

The following sections are included:

- ▶ [Overview to Applications](#) (page 192)
- ▶ [Configuration of Applications](#) (page 192)
- ▶ [Examples of Applications](#) (page 194)

# Overview to Applications

---

*Applications* are elements that provide a way to dynamically identify traffic patterns related to the use of a particular application. Applications allow you to more flexibly identify traffic beyond specifying a network protocol and ports for TCP and UDP traffic with a Service element. Matching is done based on the payload in the packets, making it possible to identify the protocol even when non-standard ports are used. Applications first identify the protocol, and then a protocol-specific pattern matching context is applied to identify the applications.

## Configuration of Applications

---

No configuration is required to be able to use Applications in Access rules. There are several predefined Application elements available that define the criteria for matching commonly-used applications. Creating new Applications or duplicating existing elements is not recommended. If you need to override the settings of a predefined Application, you can edit the Service Definition of the rule in which you use the Application.

### Default Elements

*Application Type* elements define general categories of applications. One Application Type can be associated with each Application. Application Types are predefined, and you cannot create new Application Types.

*Tags* help you to create simpler policies with less effort. Tag elements represent all Applications that are associated with that Tag. For example, the Media Tag includes several web-based image, music, and video applications. Several Tags can be associated with each Application.

*TLS Match* elements define matching criteria for the use of the TLS (transport layer security) protocol in traffic. When a connection that uses the TLS protocol is detected, the server certificate for the connection is compared to the TLS Match in the Application definition. TLS connections are allowed only to sites that have trusted certificates that meet the following criteria:

- The certificate domain name must match the domain name in the TLS Match element.
- The certificate must be signed by a valid certificate authority.
- The certificate must be valid (not expired or revoked).

The predefined elements are imported and updated from dynamic update packages. This means that the set of elements available changes whenever you update your system with new definitions. The Release Notes of each dynamic update package list the new elements that the update introduces.



# Configuration Workflow

The following sections provide an overview to the configuration tasks. Detailed step-by-step instructions can be found in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define TLS Matches

In addition to the predefined TLS Matches used in predefined Applications, you can optionally define your own TLS Matches.

TLS Matches can match traffic based on the following criteria:

- Whether certificate validation succeeded, failed, or was not performed.
- The server domain name in a valid certificate.
- Specific reasons a certificate is considered invalid if certificate validation failed.

TLS Matches also specify whether to decrypt TLS traffic to particular Internet domains for inspection. TLS Matches that deny decryption are applied globally. Even if the TLS Match element is not used in the properties of any Applications or in the Access rules, matching connections are never decrypted. Denying decryption in a TLS Match prevents Applications from being detected in encrypted connections to the specified domain(s). If the server certificate provides sufficient information to identify the Application without decrypting the client communications, you can alternatively specify that decryption is not necessary for application identification in the Application Properties.

An Application matches a TLS connection only if a TLS Match element in the Application also matches. However, TLS Matches used in Service Definitions override the TLS Match of an Application. In this case, the rule matches when the TLS Matches specified in the rule match.

## Task 2: Create Access Rules

To detect application use, you must create Access rules and use an Application in the Service cell. You can either use Applications directly in the Service cell, or as part of the Service Definition. Any other criteria in the Service Definition override the Application properties. For example, the predefined Google Application matches only TCP ports 80 and 443, but using the Any TCP Service allows the Application to match any traffic pattern that resembles the use of Google regardless of the port.

Alternatively, you can use Application Types and Tags directly in the Service cell to match any of the Applications that belong to the Application Type or Tag group.

Some Applications can open several related connections. If a related connection is identified by an Access rule that detects Application use, the related connection is matched against the Access rules again. If the rule that detected the Application use has Deep Inspection enabled and the related connection matches a rule that has Deep Inspection enabled, the related connection is matched against the Inspection Policy. No NAT payload modifications are done for the connection that matches the rule that detected the Application use. NAT payload modifications may be done for the related connections according to the policy.

## Examples of Applications

---

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

### Blocking Application Use

The administrators at Company A want to allow the use of HTTP in general, but block the use of social media applications from its corporate network. When social media use is detected, the administrators want to redirect users to the corporate security policy page on the company intranet.

The administrators:

1. Create a User Response to redirect dropped connections to the corporate security policy intranet page.
2. Add the following Access rules:

Source	Destination	Service	Action
Internal networks	Not internal networks expression	Social Media Application Tag	Discard Response: User Response to redirect connections to the intranet page
Internal networks	Not internal networks expression	HTTP	Allow

3. Refresh the firewall's policy.

## CHAPTER 21

# BLACKLISTING

*Blacklisting* is a way to temporarily block unwanted network traffic either manually or automatically with blacklist requests from a Security Engine or Log 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 196)
- ▶ [Configuration of Blacklisting](#) (page 197)
- ▶ [Using Blacklisting](#) (page 198)

# Overview to Blacklisting

Blacklisting makes it possible to block unwanted network traffic for a specified time. Engines can add entries to their own blacklists based on events in the traffic they inspect. Security Engines and Log Servers can also send blacklist requests to other Security Engines. 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 21.1 Risks of Blacklisting

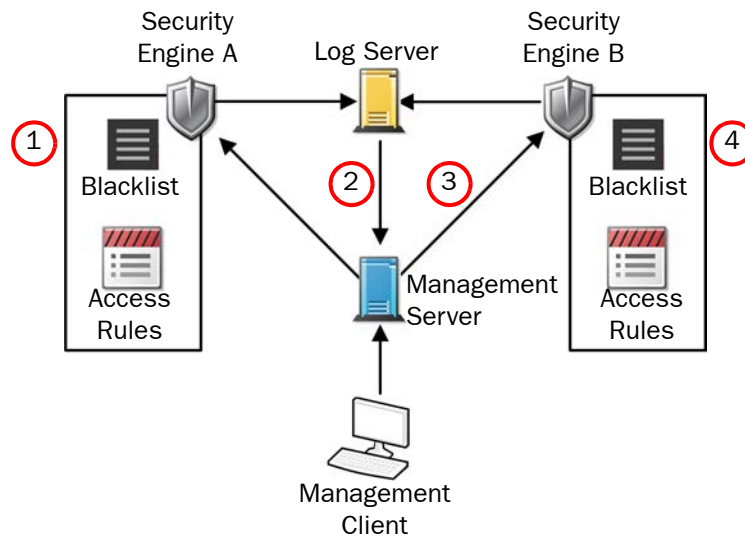
Risk	Explanation
Blacklisting legitimate connections (false positive)	If the defined pattern for detecting malicious traffic is inaccurate, legitimate traffic may sometimes be blacklisted. This causes service downtime for hosts that are incorrectly identified as a source of malicious traffic.
Causing 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 blacklisting must be defined only with good reasons.

# Configuration of Blacklisting

Blacklisting is executed as defined in the Access rules of the Firewall Policy, the Layer 2 Firewall Policy, or the IPS Policy. Automatic blacklisting requests are sent as defined in the Inspection Policy.

**Illustration 21.1 Blacklisting Process**



- Engines add entries to their own blacklists for traffic they inspect.
  - There is one blacklist for each Firewall, Layer 2 Firewall, IPS engine, or Virtual Security Engine.
  - In engine clusters, there is one blacklist for each cluster. The nodes in the cluster exchange blacklist information in their synchronization communications.
- Log Servers send blacklisting requests as a response to correlation of detected events. When one Security Engine sends a blacklisting request to another Security Engine, the Log Server relays the blacklisting request to the Management Server.
- Management Servers relay manual blacklisting commands from administrators, and blacklisting requests sent by Log Servers to the Security Engines.
  - There is no direct communication between different Virtual Security Engines or between Virtual Security Engines and the Management Server. This means that Virtual Security Engines cannot send blacklisting requests to other Virtual Security Engines.
- Engines enforce the entries on their blacklists according to their Access rules.
  - Each blacklist entry exists only for a defined time period after which the entry is cleared and matching connections are again allowed. The duration of the blocking is defined when the blacklist entry is created.
  - Access rules check connections against the blacklist. If the IP addresses and ports in one of the blacklist entries match, the connection is discarded.
  - If the connection does not match a blacklisting Access rule or its related blacklist entries, the next Access rule in the policy is checked as usual.

# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define Blacklisting in Access Rules

Access rules define which components are allowed to add entries to an engine's blacklist, and which connections are checked against the blacklist. Blacklisting is applied with Access rules that contain the Apply Blacklist action. Only connections that match the blacklisting Access rules are blacklisted.

No further configuration is needed if you want to blacklist connections manually. Task 2 explains the additional configuration needed for automatic blacklisting with the Inspection Policy.

## Task 2: Define Exceptions in the Inspection Policy

Blacklist scope options in the Exceptions of the Inspection Policy trigger automatic blacklisting for the detected events. You can define Blacklisting scope options for any type of Exception, including rules that use Correlation Situations. Automatic blacklist entries are created 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

---

### Manual Blacklisting

You can blacklist connections manually through the Management Client. 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 Security Engine element, or create new blacklist entries in the Blacklist view. Blacklist entries can be removed and added manually.

### Monitoring Blacklisting

The currently active blacklisting entries on the engine can be monitored in the Blacklist view. Blacklist monitoring does not show you which connections are actually dropped. Blacklist monitoring only shows you the IP addresses that are currently on the blacklist. 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 log entries.

### Whitelisting

Whitelisting means defining a list of IP addresses that must never be blacklisted. Whitelisting is implemented by following general Access rule design principles. Blacklisting applies only at the position of the blacklisting Access rule(s) in the policy. Connections that have already been allowed or discarded before the blacklisting rules is not affected by blacklisting. If an Access rule in the policy allows a connection, an Access rule that refers to the blacklist further down in the policy cannot blacklist the connection.

# USERS AND AUTHENTICATION

---

**In this section:**

**Directory Servers - 201**

**User Authentication on the Firewall - 207**

**External User Authentication - 213**





## CHAPTER 22

# DIRECTORY SERVERS

A directory server is a server that contains a database where information about user accounts is stored.

The following sections are included:

- ▶ [Overview to Directory Servers](#) (page 202)
- ▶ [Configuration of Directory Servers](#) (page 202)
- ▶ [Examples of Directory Servers](#) (page 206)

## Overview to Directory Servers

---

A directory server is a server that contains a user database that is queried during the user authentication process. You can store the user accounts in the Management Server's internal user database, or on an external directory server. Different users can be stored in different directories. Authentication is based on the user information, but is a separate operation and is not necessarily done on the same server that stores the user information. See [User Authentication on the Firewall](#) (page 207) and [External User Authentication](#) (page 213) for more information about authentication.

## Configuration of Directory Servers

---

User information is stored in an internal or an external LDAP (Lightweight Directory Access Protocol) directory. The standard LDAP user management model consists of three different levels: *LDAP domains*, *user groups*, and *users*. All three levels are represented as elements in the Management Client.

### Internal User Database

The Management Server includes an integrated LDAP directory for storing user information. The Management Server's internal user database can be used for authenticating users with passwords. Using an internal LDAP directory is the simplest choice when there is no specific need to have an external LDAP server.

When the Management Server's internal LDAP directory is used, the user and user group information is stored on the Management Server. Each firewall node stores a replica of the user database, and any changes to the main database are replicated immediately to the firewalls. This way, the firewalls can access their local directories instead of constantly communicating user information over the network.



**Note – It is not possible to give external components (such as external authentication servers) access to the Management Server's internal LDAP directory.**

If Domain elements have been configured, the Internal LDAP directory belongs to the Shared Domain. This means that the administrators who log in to some other Domain are allowed to view the contents of the Internal LDAP directory. If all user information should not be available to administrators in all Domains, you must use an external LDAP directory in each Domain. See the *McAfee SMC Reference Guide* for more information on Domains.

### Authentication Server User Linking

The optional Authentication Server component includes its own user database that is not based on LDAP. User and user group information is not replicated from the Authentication Server's user database to firewalls. See [Overview to External User Authentication](#) (page 214) for more information about the user authentication process with the Authentication Server.

# External Directory Server Integration

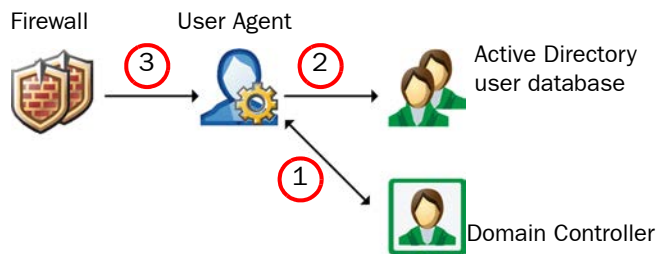
An external LDAP directory can be used instead of or in addition to the internal LDAP directory. The external directory server can be a general LDAP server, or a Microsoft Active Directory server providing LDAP services. The Management Server is defined as an LDAP client for the external server. User and User Group information is shown in the Management Client, and can be used for creating rules.

The external LDAP server's schema file defines the attributes (individual pieces of data) that a user account can contain. Extending the external server's schema with SMC-specific attributes is optional, but extending the schema allows you to also add SMC-specific information to User and User Group elements through the Management Client.

## User Agents for Active Directory

The User Agent is an optional software component that associates users from an integrated Active Directory server with IP addresses. This allows User and User Group elements to be used as the source and destination of rules without user authentication. For more information, see [Creating User-Specific Access Rules](#) (page 113).

**Illustration 22.1** User Agent Communications



1. The User Agent monitors the Domain Controller's Security Event Log to keep track of when a user logs on or logs off, a user's IP address changes, or a user acquires the same IP address that was previously associated with another user.
  - Users are associated with IP addresses based on logs collected by the Active Directory Domain Controller. For this reason, it is only possible to associate one user with each IP address from a client computer. It is not recommended to use the User Agent with terminal servers or other computers that have many users logged on at the same time.
  - The User Agent periodically sends ICMP echo (ping) requests to users' workstations to monitor which users are active. If a user's workstation does not respond, the user is removed from the list of IP addresses. In cases where ping requests to workstations are not allowed, users' connections may be incorrectly closed. Workstation monitoring can optionally be disabled in the Windows registry to prevent this. See the User Agent [Release Notes](#) for more information.
2. When the User Agent receives information about a particular user, the User Agent sends an LDAP query to the Active Directory server to look up the user's group membership from the user database.
3. The firewall periodically queries the User Agent to update the information about which IP address is associated with the user, and which User Group the user belongs to.

# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Create an LDAP Server or an Active Directory Server Element

If you want to use an external directory server, you must create an element to define the parameters for contacting the server and the LDAP object classes. You define the directory parameters in an LDAP Server element. If you are using an Active Directory server, you can define both the directory and the authentication parameters in the same Active Directory Server element. If administrative Domains have been created to separate configurations, use a separate external LDAP Server or Active Directory Server in each administrative Domain to create user accounts that are specific to an administrative Domain.

If you want to use authentication methods provided by the Authentication Server component, you must store users in an external directory and link users to the Authentication Server's internal user database.

## Task 2: Add an LDAP Domain

When using the internal user database, the users and user groups are always stored and managed in the default **InternalDomain** LDAP Domain.

If you are using an external LDAP directory for user management, you must create a new LDAP Domain. After the LDAP Domain is associated with the external server, the Management Server contacts the LDAP directory. You can then view and edit users and user groups through the Management Client.

You can select one LDAP Domain (excluding the Domain for Authentication Server) as the **Default LDAP Domain**. This allows users belonging to that LDAP Domain to authenticate without specifying the LDAP Domain information. Users in other LDAP Domains must specify their LDAP Domain whenever they authenticate themselves.

If there are administrative Domains to separate configurations, create a separate LDAP Domain in each administrative Domain to create user accounts that are specific to an administrative Domain. These can also point to different parts of the directory hierarchy in the same LDAP directory. The internal LDAP directory is always in the Shared Domain, which makes its contents visible in all administrative Domains. You can either select one Default LDAP Domain in each administrative Domain or select one of the LDAP Domains in the Shared Domain as the Default LDAP Domain for all administrative Domains.

### **Task 3: Add Users and User Groups or Link Users**

To implement user authentication, you must define user accounts either in the internal user database or on an external directory server. It is not possible to add users directly to the Authentication Server component's user database. Instead, users stored on an external directory server can be linked to the Authentication Server's database.

If you are using an external server for authentication and you do not need to define different access rights for different users, it is not necessary to integrate an external directory server with the SMC. In that case, you can create a special User element with the name `*external*` in the internal user database to represent any user that authenticates using the external authentication service.

Users are created as members of a User Group. This saves time and effort, as you do not have to specify all user parameters separately for each individual User. A User that is a member of a User Group can inherit, for example, the Authentication Method and account expiration time from the User Group. Each User Group must belong to an LDAP Domain. We recommend that each user have a separate user account used by that person. Each user can belong to several User Groups within the LDAP Domain. User-specific properties can override properties defined at the User Group level.

You can import and export Users and User Groups through an LDIF file to/from some other Management Server.

### **Task 4: Install and Configure the User Agent**

If you want to use Users and User Groups from an integrated Active Directory server as the source and destination of rules, you must install the User Agent software on a Windows server that communicates with the Active Directory Domain Controllers. You define the information about the Domain Controllers in the properties of the Active Directory Server element. You define the information about the User Agent as a User Agent element, and specify which User Agent each firewall communicates with in the properties of the firewall.

## Examples of Directory Servers

---

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

### Using the Internal User Database

Company A has a general office network and a separate HR network for servers that contain HR information, such as employee records and payroll information. The servers restrict which users have access, but for auditing reasons, the administrators want to separate the users into groups and require authentication to access the HR network. The administrators:

1. Create a User Group “HR Users” in the InternalDomain and assign one of the default internal authentication methods.
2. Create User elements for each person with access rights under the HR Users group.
3. Define Access rules for user authentication on the firewall.

### Integrating a Microsoft Active Directory Server

This example provides an overview to the configuration. For more information on configuring IAS, consult Microsoft’s documentation at <http://technet.microsoft.com/>.

Company B has an existing Microsoft Active Directory server that stores user information. They decide to integrate this existing server’s directory services.

The administrators:

1. Define an Active Directory Server element.
2. Add the SMC-specific classes and attributes into the Active Directory server’s configuration to be able to fully manage the user accounts through the Management Client.
3. Define the Management Server as an LDAP client for the Active Directory server.
4. Define the Firewall as an authentication client for the IAS.
5. Add a new LDAP Domain element for the Active Directory server in the Management Client.

# USER AUTHENTICATION ON THE FIREWALL

User authentication means requiring the users of services in your network to authenticate themselves before they are given access to some resources. User authentication on the firewall means that the firewall checks user credentials against its own replica of the user database.

The following sections are included:

- ▶ [Overview to User Authentication on the Firewall](#) (page 208)
- ▶ [Configuration of User Authentication on the Firewall](#) (page 209)
- ▶ [Example of User Authentication on the Firewall](#) (page 211)

## Overview to User Authentication on the Firewall

---

User authentication means requiring the users to prove their identity before giving access to a network resource. User authentication is mandatory with client-to-gateway VPNs, but you can require it for non-VPN connections as well. User authentication on the firewall is only supported for IPv4 traffic.

User authentication requires creating user accounts. See [Directory Servers](#) (page 201) for more information about user accounts. Different users can use different authentication methods. Storing the user information and authenticating the users are two separate concepts with separate options.

User authentication proceeds as follows:

1. The user opens an authentication connection to the firewall.
2. The firewall checks if the user exists and which authentication method the user should use.
3. The user-supplied credentials are verified.

If authentication succeeds, the firewall lists the user as an authenticated user, taking note of both username and authentication method.

When the user opens new connections, IPv4 Access rules that contain an authentication requirement may now match (in addition to other rules). The username and authentication method are both separately considered as matching criteria.

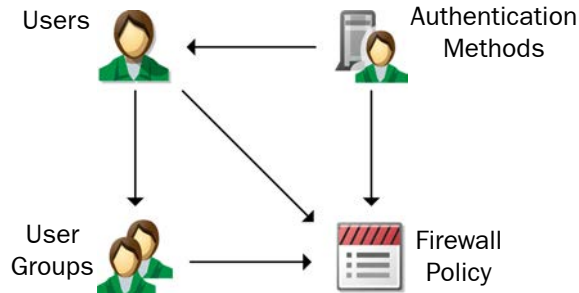
When the configured timeout is reached, the authentication expires and the user is removed from the list of authenticated users. Access rules that require authentication no longer match the user's connections.



# Configuration of User Authentication on the Firewall

*Authentication methods* define the authentication method used by particular users and user groups. The illustration below shows how elements are configured for user authentication on the firewall:

**Illustration 23.1** Configuring User Authentication



Authentication Methods define the allowed authentication methods for IPv4 Access rules and for the Users and User Groups. Both User and User Group elements can be used in IPv4 Access rules to define rules that only match connections from specific, successfully authenticated users. A specific Authentication Method definition is needed in each IPv4 Access rule especially when the Users and User Groups have several allowed Authentication Methods. Otherwise, the rules can allow any defined Authentication Method that is allowed for the included users.

## Default Elements

There are three predefined Authentication Methods for user authentication on the firewall:

- **IPsec Certificate** is for certificate-based authentication.
- **Pre-Shared Key Method** is for use with some third-party VPN clients.
- **User Password** is for simple password authentication against the internal LDAP database (including user authentication in IPsec VPN Client hybrid authentication).

To use the firewall for user authentication, you must use one of the predefined Authentication Methods.

## 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define User Authentication in IPv4 Access Rules

The IPv4 Access rules in a firewall policy can be configured to match only when the user is authenticated. With client-to-gateway VPNs, some form of authentication is always mandatory, but authentication can be required for non-VPN access as well. The authentication parameters are defined in the **Authentication** cell.

**Illustration 23.2 Authentication Field in the IPv4 Access Rules**

ID	Source	Destination	Service	Action	Authentication
14.1.1	ANY	net-192.168.1.0/24	ANY	Enforce VPN: \$ Client-to-gateway IPsec VPN	<div>Mobile VPN users</div> <div>Authorize client IP Client initiated</div> <div>IPsec Certificate Timeout = 3600</div>

An authentication method is activated when at least one rule that contains the corresponding Authentication Method element is installed on the firewall. The authentication is usually granted for a specific duration based on source IP address. Single connection authentication is an alternative with Telnet-based authentication.

No rules are needed to allow the authentication connection, except when browser-based user authentication is used. Any end-user with a valid user account for the active authentication methods is allowed to authenticate even if there are no rules that require authentication to access a particular service.

Once the user successfully authenticates, the firewall adds the user on a list of authenticated users. The next connection that the user opens can now match an Access rule that requires authentication if the user and authentication method match the parameters of the rule.

Note that the User, User Group, and Authentication Method elements are simply used as matching criteria, so any of the other rules above or below may also match the authenticated user's connections. This is especially important to consider when VPN client connections are concerned, since the IPsec VPN Client can be configured to receive an IP address from the organization's internal IP address space.

If necessary, you can define rules that discard connections from some combinations of Users and Authentication methods. The Source VPN cell in IPv4 Access rules can be used to match VPN traffic/non-VPN traffic as desired.

## Task 2: Configure User Authentication Interfaces

End-users usually authenticate through a VPN client, which requests the user to authenticate as needed. See [Overview to Policy-Based VPN Configuration](#) (page 270) for more information about VPNs. When the VPN client is used, successful authentication opens a VPN tunnel.

End-users can alternatively open an authentication page in a web browser. The end-users can authenticate using encrypted HTTPS connections as well as plain HTTP connections. Browser-based user authentication is configured in the properties of the firewall. The IPv4 Access rules for allowing authentication connections are not included in the Firewall Template Policy. You must add a rule that allows this traffic in the firewall's policy. Additionally, you must add IPv4 Access and Inspection rules to enable redirection of unauthenticated HTTP connections to the login page.



**Caution – Plain HTTP connections are unsecured and transfer user access credentials in cleartext. Use encrypted HTTPS connections to avoid loss of sensitive information.**

The end-users can also launch a separate Telnet authentication connection to the firewall. No special configuration is needed to use Telnet authentication.



**Caution – The Telnet method transfers the username and password in cleartext and does not provide any security in addition to the initial authentication of an IP address or a connection. Use a VPN client when a higher security level is required.**

## Example of User Authentication on the Firewall

The example in this section illustrates a common use for User Authentication on the firewall and general steps on how the scenario is configured.

### Authenticating VPN Client Users

Company A's employees include several consultants who frequently work at customer locations, but need to remotely access Company A's secure network. All of the company's users are stored in the Management Server's internal directory, and there is a separate User Group called Consultants for accounts belonging to the consultants. The administrators have set up a client-to-gateway VPN for remote access. They want to allow all users to establish a VPN tunnel to the office, but allow only users in the Consultants group to access the secure network.

The administrators:

1. Create a rule that establishes a VPN tunnel and allows users in the Consultants group to access the Secure Network after successful authentication:

Source	Destination	Service	Action	Authentication
DHCP address range for VPN clients Internal Networks	Secure Network	HTTP SSH FTP	Enforce VPN	Consultants User Group User Password Authentication

- This rule allows any users in any directory that is defined in the SMC to authenticate to a VPN client if their allowed authentication methods include User Password.
  - This rule allows any user whose account is stored in the internal directory to use a VPN client to establish a VPN tunnel to the office.
2. Create a rule to allow users who have established VPN tunnels to access the company's internal networks from the DHCP-assigned IP addresses for VPN clients:

Source	Destination	Service	Action	Authentication
DHCP address range for VPN clients	Internal Networks	ANY	Allow	

3. Transfer the policy to the firewall.



# EXTERNAL USER AUTHENTICATION

User authentication means requiring the users of services in your network to authenticate themselves before they are given access to some resources. In external user authentication, a separate server verifies the user credentials.

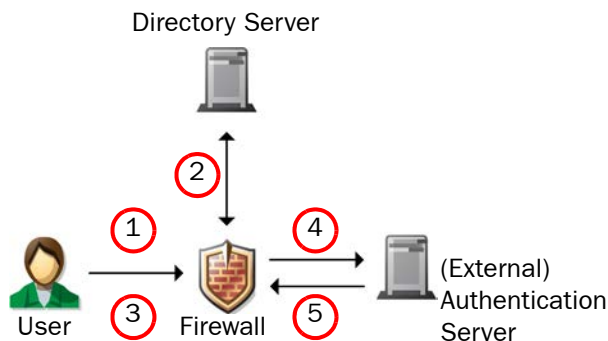
The following sections are included:

- ▶ [Overview to External User Authentication](#) (page 214)
- ▶ [Configuration of External User Authentication](#) (page 215)
- ▶ [Examples of External User Authentication](#) (page 220)

# Overview to External User Authentication

External user authentication means that authentication services are provided by an authentication server external to the firewall, with no replication of user information to the firewall. The external authentication server can be the optional Authentication Server component, or a third-party authentication server. You can use external authentication services that support the RADIUS or TACACS+ protocol, such as RSA Authentication Manager (formerly known as ACE/Server) or the IAS (Internet authentication service) of a Windows (Active Directory) server. User authentication is only supported for IPv4 traffic.

**Illustration 24.1 External User Authentication Process**



External user authentication proceeds as follows:

1. The user opens an authentication connection to the firewall.
2. The firewall queries the directory server to check if the user exists and which authentication method the user should use.
3. The firewall prompts the user to authenticate, and the user enters the credentials required for the authentication method.
4. The firewall relays the user credentials to the authentication sever.
5. The authentication server verifies the user credentials and responds to the firewall whether authentication succeeds or fails.

If authentication succeeds, the firewall lists the user as an authenticated user, storing both the username and authentication method.

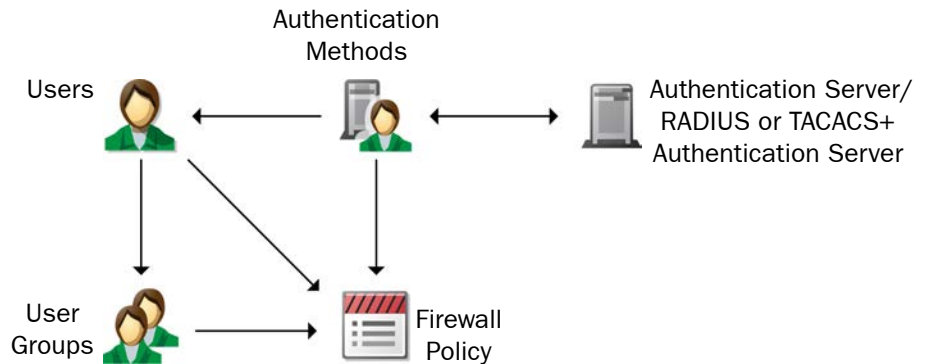
When the user opens new connections, IPv4 Access rules that contain an authentication requirement may now match (in addition to other rules). The username and authentication method are both separately considered as matching criteria.

When the configured timeout is reached, the authentication expires and the user is removed from the firewall's list of authenticated users. IPv4 Access rules that require authentication no longer match the user's connections.

# Configuration of External User Authentication

The illustration below shows how user authentication elements are configured.

**Illustration 24.2** Configuring User Authentication



- The optional Authentication Server component is installed as part of the McAfee Security Management Center (SMC) installation and configured as an Authentication Server element.
- External authentication servers are configured as RADIUS or TACACS+ Authentication Server elements.
- RADIUS or TACACS+ Authentication Servers and the Authentication Server component can be located in any network that allows them to communicate with the firewall that has an authentication rule in its policy. The Authentication Server component must also be able to communicate with the Management Server.
- Authentication Method elements are associated with authentication servers to define the allowed authentication methods for the server, or the servers that use a particular authentication method.
- Both User and User Group elements can be used in IPv4 Access rules to define rules that only match connections from specific, successfully authenticated users.
- A specific Authentication Method definition is needed in each IPv4 Access rule especially when the Users and User Groups have several allowed Authentication Methods. Otherwise, the rules allow any defined Authentication Method that is allowed for the included users.

## Directory Servers for External User Authentication

User authentication requires the creation of user accounts. You can use the same server for storing and authenticating the users, for example, when you use a Microsoft Active Directory server for both tasks. However, keep in mind that storing the user information and authenticating the users are two separate concepts with separate options. See [Directory Servers](#) (page 201) for more information about user accounts.

To be able to define different IPv4 Access rules for different users and user groups with external authentication, you must integrate an external directory server with the SMC. Users in the Authentication Server's directory cannot be used directly in rules. Instead, you must use the User element from the external directory server to which the Authentication Server user is linked. User Groups cannot be used with the Authentication Server.

It is also possible to use an external authentication server or the Authentication Server component without integration of an external directory server. In this configuration, it is not possible to add different IPv4 Access rules for different users and user groups, since the user information is not available to the firewall. Instead, you add the user *\*external\** with the correct external authentication method(s) into the internal user database, and use it to define which IPv4 Access rules require authentication.

## RADIUS Authentication

Remote Authentication Dial-in User Service (*RADIUS*) is a protocol for carrying authentication, authorization, and configuration information. RADIUS is a widely supported standard. For example, Microsoft IAS and RSA Authentication Manager (formerly known as ACE/Server) support the protocol and can be used for user authentication in the SMC. The authentication methods provided by the Authentication Server component are also based on the RADIUS protocol.

RADIUS uses UDP as its transport protocol. The exchanges between the client and the RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. User passwords transferred between the client and the RADIUS server are encrypted using the MD5 message digest algorithm. The rest of the RADIUS communications are in cleartext.

Servers that provide RADIUS-based authentication methods can also be used for authenticating administrators' Management Client logins (see the *McAfee SMC Reference Guide*) and wireless client connections to wireless interfaces on firewalls (see [Single Firewall Configuration](#) (page 43)).

Additionally, the Authentication Server component can optionally collect RADIUS accounting information and provide a summary of the data. RADIUS Accounting allows you to keep track of an authenticated user's session for billing, statistical, or network monitoring purposes. RADIUS Accounting packets contain information about when a user logs in and out of a system, and what resources the user accesses.



# TACACS+ Authentication

Terminal Access Controller Access Control System Plus (TACACS+) is a protocol used for similar purposes as RADIUS. In general, TACACS+ provides a more secure method of user authentication than RADIUS. This is mainly because TACACS+ uses TCP as the transport protocol instead of UDP. Consequently, transport is more reliable and less sensitive to disruption at the network layer.

TACACS+ also separates authentication, authorization, and accounting services, whereas RADIUS provides a user profile defining all the user-specific parameters with the authentication. This separation of services allows TACACS+ to use other forms of authentication, such as Kerberos, together with its own authorization.

TACACS+ uses a pre-shared key to authenticate exchanges. TACACS+ encrypts all traffic between the authentication server and the device requesting authentication. User information, such as IDs and passwords, are secured with the MD5 message digest algorithm.

## Authentication Methods

*Authentication methods* define the authentication method used by particular authentication servers, and by particular users and user groups. The following authentication methods can be used:

- User passwords stored in internal or external LDAP databases.
- IPsec certificates and passwords (for use with IPsec VPN Clients).
- Pre-shared keys (for use with some third-party VPN clients).
- Password, Mobile ID, and Mobile Text Authentication methods provided by the optional Authentication Server component.
- External authentication provided by servers that support the TACACS+ (Terminal Access Controller Access Control System Plus) protocol.
- External authentication provided by servers that support the RADIUS (Remote Authentication Dial-in User Service) protocol.

The SMC supports many third-party Authentication Methods. You can integrate third-party authentication products with the SMC through the RADIUS and TACACS+ protocols to provide simple password authentication, one-time passwords, or any other username/passcode-type authentication schemes.

The Authentication Server component supports the following authentication methods:

- **Password** is based on static password authentication. A static password is created and maintained for authenticating remote access.
- **Mobile ID - Synchronized** is for use with the Stonesoft Mobile ID client. During authentication, users enter their user ID and are prompted to enter a one-time password (OTP). Users enter their PIN in the Mobile ID client and the Mobile ID client software generates the OTP.
- **Mobile ID - Challenge** is for use with the Stonesoft Mobile ID client. During authentication, users enter their user ID, and are prompted with a challenge to provide the correct response. Users enter their PIN in the Mobile ID client, and the Mobile ID client software generates the response.
- **Mobile Text** is based on a combination of a PIN and one-time password (OTP) distributed by SMS to the end-user.

# Federated Authentication

In Federated Authentication, user identities and authentication are managed separately from services. This allows entities that provide services to delegate the authentication process and maintenance of user accounts to another entity. Federated Authentication also allows the user to use the same credentials for authentication in multiple security domains, optionally as part of a single-sign-on (SSO) configuration.

Entities in a Federated Authentication scenario have the following roles:

- **Subject:** the user who requests access to an application or service.
- **Identity Provider:** Verifies the credentials of the user and creates a unique, signed SAML assertion that contains the information about the user and the user's privileges. This assertion is then used to authenticate the user to the Service Provider.
- **Service Provider:** Provides applications or services. When a user requests an application or service, the Service Provider sends an authentication request to the Identity Provider. The Identity Provider authenticates the user and replies with an authentication response to the Service Provider.

The Authentication Server component can act as the Identity Provider. All Authentication Methods provided by the Authentication Server can be used to create SAML assertions for Federated Authentication.

## Default Elements

There are three predefined Authentication Methods for use with RADIUS or TACACS+ Authentication Server elements:

- **IAS Authentication** is for use with an external IAS (Active Directory) server.
- **Pre-Shared Key Method** is for use with some third-party VPN clients.
- **User Password** is for simple password authentication against the internal LDAP database.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Define Servers

The optional Authentication Server component is installed as part of the SMC installation.

Third-party external authentication servers are configured using RADIUS or TACACS+ Authentication Server elements. The server element is used for defining the information that SMC components need to contact the external authentication server, such as the IP address and the shared secret.

In addition to the server element configuration, you must configure the external authentication server to allow the firewalls to use the authentication services. The Authentication Server component automatically allows single firewalls with static IP addresses to use the authentication services.

## Task 2: Associate Authentication Methods with Servers

Authentication Methods represent third-party external authentication servers and the Authentication Server component in user properties and IPv4 Access rules.

There is a predefined Authentication Method for IAS that is automatically used for authentication with Microsoft IAS and Active Directory. To use some other external authentication server or the Authentication Server component, or to use an Active Directory server for RADIUS-based authentication, you must define Authentication Method elements.

Each Authentication Method element can be associated with one or more servers, but each RADIUS or TACACS+ Authentication Server can only be associated with one Authentication Method. When multiple servers are associated with the same Authentication Method, the servers are used as alternative servers if the first contacted server does not respond. All servers associated with the same Authentication Method must contain identical information on each authenticating user, since it is not possible for the user to determine which of the alternative servers is being contacted.

Authentication Methods for the Authentication Server component are based on the four predefined types of authentication methods. You can add one Authentication Method of each type to the Authentication Server. The Authentication Methods for the Authentication Server cannot be used by external authentication servers.

## Task 3: Define User Authentication in IPv4 Access Rules

The IPv4 Access rules in a firewall policy can be configured to match only when the user is authenticated. The authentication parameters are defined in the **Authentication** cell.

An authentication method is activated when at least one rule that contains the corresponding Authentication Method element is installed on the firewall. The authentication is usually granted for a specific duration based on source IP address. Alternatively, authentication can be granted only for the duration of a single connection with Telnet-based authentication.

No rules are needed to allow the authentication connection, except when browser-based user authentication is used. Any end-user with a valid user account for the active authentication methods is allowed to authenticate even if there are no rules that require authentication to access a particular service.

Once the user successfully authenticates, the firewall adds the user on a list of authenticated users. The next connection that the user opens can now match an Access rule that requires authentication if the user and authentication method match the parameters of the rule.

Note that the User, User Group, and Authentication Method elements are simply used as matching criteria, so any of the other rules above or below may also match the authenticated user's connections. This is especially important to consider when VPN client connections are concerned, since the VPN client can be configured to receive an IP address from the organization's internal IP address space.

If necessary, you can define rules that discard connections from some combinations of Users and Authentication methods. The Source VPN cell in IPv4 Access rules can be used to match VPN traffic/non-VPN traffic as desired.

## Task 4: Configure User Authentication Interfaces

End-users usually authenticate through a VPN client, which requests the user to authenticate as needed. See [Overview to Policy-Based VPN Configuration](#) (page 270) for more information about VPNs. When the VPN client is used, successful authentication opens a VPN tunnel.

End-users can alternatively open an authentication page in a web browser. The end-users can authenticate using encrypted HTTPS connections as well as plain HTTP connections. Browser-based user authentication is configured in the properties of the firewall. The IPv4 Access rules for allowing authentication connections are not included in the Firewall Template Policy. You must add a rule that allows this traffic in the firewall's policy. Additionally, you must add IPv4 Access and Inspection rules to enable redirection of unauthenticated HTTP connections to the login page.



**Caution – Plain HTTP connections are unsecured and transfer user access credentials in cleartext. Use encrypted HTTPS connections to avoid loss of sensitive information.**

The end-users can also launch a separate Telnet authentication connection to the firewall. No special configuration is needed to use Telnet authentication.



**Caution – The Telnet method transfers the username and password in cleartext and does not provide any security in addition to the initial authentication of an IP address or a connection. Use a VPN client when a higher security level is required.**

## Examples of External User Authentication

---

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

### Integrating a Microsoft Active Directory Server

This example provides an overview to the configuration. For more information on configuring IAS, consult Microsoft's documentation at <http://technet.microsoft.com/>.

Company B has an existing Microsoft Active Directory server that stores user information. They decide to use this existing information for user authentication.

The administrators:

1. Define an Active Directory Server element.
2. Add the SMC-specific classes and attributes into the Active Directory server's configuration to be able to fully manage the user accounts through the Management Client.
3. Define the Management Server as an LDAP client for the Active Directory server.
4. Define the Firewall as an authentication client for the IAS.
5. Add a new LDAP Domain element for the Active Directory server in the Management Client.

6. Add an IPv4 Access rule with authentication defined as shown below.

**Table 24.1 Example Access Rule for IAS Authentication**

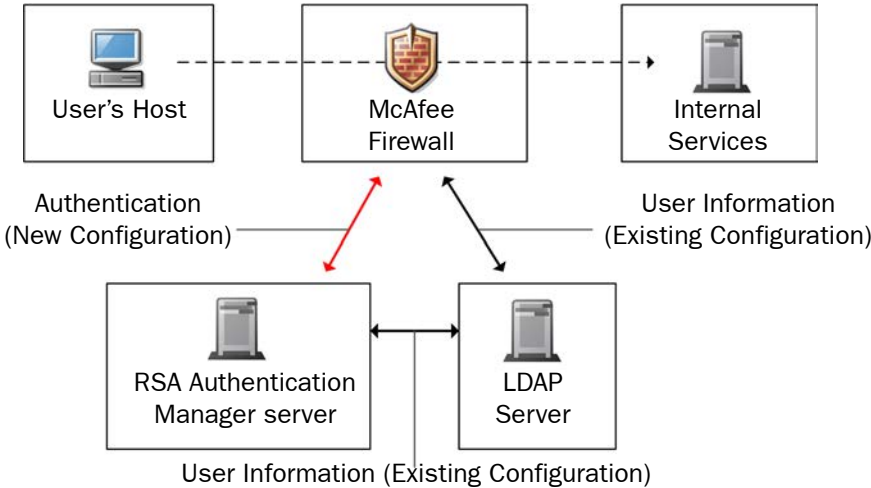
Source	Destination	Authentication
IP addresses of authenticated hosts.	IP addresses of network services that require authentication.	Some User or User Group elements from the AD's LDAP Domain. Require authentication with "IAS Authentication" Authentication Method.

# Using SecurID Authentication with Stonesoft IPsec VPN Clients

This example provides an overview to the configuration. For more information on using SecurID authentication, consult RSA's documentation at <http://www.emc.com/domains/rsa/index.htm?id=1850>.

Company C is about to introduce remote Stonesoft IPsec VPN Client access to their network. The administrators want to enhance the security of their authentication solution, as authentication is currently done using an external LDAP server and Telnet clients within the internal network. They decide to add the use of one-time passwords with SecurID cards with their existing RSA Authentication Manager server that already shares the user information with the company's LDAP server.

**Illustration 24.3 Company C's Authentication Scheme**



The administrators:

1. Create an Agent Host record for the Firewall in the RSA Authentication Manager server.
2. Create a VPN configuration in the Management Client with the default Hybrid Authentication selected as the authentication method for connecting clients.
  - Hybrid authentication, available for Stonesoft IPsec VPN Clients, requires that the VPN Gateway (the firewall) authenticates users using a certificate and that the users provide the correct Username/Password combination (validated by the RSA Authentication Manager server in this case).
3. Create a RADIUS Authentication Server element.
4. Create a custom Authentication Method element for the server and name it “SecurID”.
5. Add the “SecurID” Authentication Method in the correct User and User Group elements (stored on the existing external LDAP server).
6. Add IPv4 Access rules with both an authentication and a VPN requirement defined as shown below.

**Table 24.2 Example Access Rule for SecurID Authentication**

Source	Destination	Authentication	Action
The virtual IP address range used on VPN Clients’ virtual adapters.	IP addresses of network services that require authentication.	Some User or User Group elements. Require authentication with “SecurID” Authentication Method.	“Use IPsec VPN” with the “Enforce VPN” option.

# TRAFFIC MANAGEMENT

---

**In this section:**

**Outbound Traffic Management - 225**

**Inbound Traffic Management - 235**

**Bandwidth Management and Traffic Prioritization - 245**





# OUTBOUND TRAFFIC MANAGEMENT

You can use Multi-Link to distribute outbound traffic between multiple network connections and to provide high availability and load balancing for outbound traffic.

The following sections are included:

- ▶ [Overview of Outbound Traffic Management](#) (page 226)
- ▶ [Configuration of Multi-Link](#) (page 226)
- ▶ [Using Multi-Link](#) (page 230)
- ▶ [Examples of Multi-Link](#) (page 232)

# Overview of Outbound Traffic Management

---

A single connection to the Internet is a single point of failure: if the connection becomes unavailable, all outbound traffic is blocked. To prevent this, Multi-Link distributes *outbound traffic* and balances the load of outbound traffic between multiple network connections. Multi-Link ensures that Internet connectivity remains available even when one or more network connections fail. You can also use Multi-Link with aggregated links.

This chapter describes the use of Multi-Link for outbound connections by creating an Outbound Multi-Link element and by adding NAT rules manually to the Firewall Policy. For inbound connections, load balancing is provided by Server Pool elements. For information about using Server Pools and Multi-Link for inbound connections, see [Inbound Traffic Management](#) (page 235). For information on using Multi-Link with VPN, see [Multi-Link and Policy-Based VPNs](#) (page 287).

If you use element-based NAT and multiple external IP addresses, the default NAT address works like an Outbound Multi-Link and the NAT rules are automatically generated. See [Network Address Translation \(NAT\)](#) (page 131) for more information.

## Configuration of Multi-Link

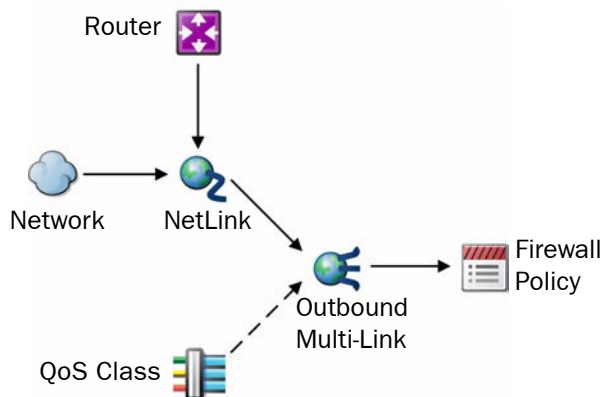
---

You can use Multi-Link on both Single Firewalls, Firewall Clusters, and Virtual Firewalls.

The network connections for Multi-Link are represented by *NetLink* elements in the SMC. In most cases, a NetLink element is used to represent an ISP connection. However, NetLinks can also represent a leased line, xDSL, or any other type of network connection mediated by your firewall.

There are two types of NetLinks: *static* and *dynamic* NetLinks. Static NetLinks are supported in the routing configuration for both IPv4 and IPv6 traffic. Dynamic NetLinks are used only with Single Firewalls, as dynamic IP addresses are not supported for Firewall Clusters. If you configure wireless Multi-Link on a Modem Interface of a single firewall, only Dynamic NetLinks are supported, as Modem Interfaces always have dynamic IP addresses.

**Illustration 25.1** Configuration of Multi-Link



The illustration above shows the elements that are used to configure Multi-Link. Each NetLink element contains a Router element that represents the router for that network connection, and a Network element that represents the set of public IP addresses allocated by the provider of the network connection. NetLinks are added to the Routing view under the Interface IDs and the Modem numbers that represent the physical interfaces or the 3G modems towards the routers used for the Internet connections.

Multiple NetLinks are combined into an Outbound Multi-Link element. Outbound Multi-Link elements are the central elements used to configure load balancing for outbound traffic. Outbound load balancing is implemented by using the Outbound Multi-Link elements in the Firewall Policy's NAT rules. Outbound Multi-Link elements are only supported for IPv4 traffic.

## Load-Balancing Methods

Load balancing can be based on either of two methods: *Round Trip Time* or *Ratio*.

When the Round Trip Time method is used, NetLink performance is measured for each new TCP connection by sending the initial request (SYN) to the destination through all the available NetLinks. When the destination host sends the reply (SYN-ACK), the NetLink that receives the reply first is used to complete the TCP connection establishment. The firewall cancels the slower connection attempts by sending a TCP Reset (RST) to the destination through the other NetLinks. This way, the fastest route is automatically selected for each connection. Information about the performance of each NetLink is cached, so no new measurement is made if a new connection is opened to the same destination within a short time period.

When the Ratio method is used, traffic is distributed between all of the available NetLinks according to the relative capacity of the links. The bandwidths of the other NetLinks are automatically compared to the bandwidth of the NetLink with the most bandwidth to produce a ratio for distributing the traffic. When the volume of traffic is low, the ratio of actual traffic distribution is approximate. When the volume of traffic is high, the ratio of traffic handled by each NetLink is closer to the ratio calculated from the link capacity.

## Standby NetLinks for High Availability

*Standby NetLinks* allow you to define a NetLink as a backup that is only activated when all primary NetLinks are unavailable. This minimizes the use of NetLinks that are more expensive or otherwise less preferable, while still ensuring the high availability of Internet connectivity. To test which NetLinks are available, the status of the NetLinks is monitored as described in [Link Status Probing](#) (page 228). As soon as one or more primary NetLinks become active again, the standby NetLinks are deactivated. Previously established connections continue to be handled by the deactivated NetLink, but new connections are no longer sent to the standby NetLink. You can define multiple active and standby NetLinks.

When load balancing is used with standby NetLinks, traffic is only distributed between the NetLinks that are currently active. Standby NetLinks are only activated when failure is detected, not to balance the load.

Using standby NetLinks on the same interface as other NetLinks affects the interface speed you enter in the configuration of QoS for firewall interfaces (see [Task 4: Define QoS for Interfaces and VPNs](#) (page 251)).

## Link Status Probing

To be able to monitor the status of the NetLinks, the firewall must be configured to probe each NetLink. Probing is always recommended, and it is mandatory with the following features:

- The Ratio-based load-balancing method.
- Failover to Standby NetLinks.
- Inbound traffic balancing with dynamic DNS updates.

Traffic is correctly balanced between active working links without probing if Round Trip Time balancing is used, since failed links are eliminated from use in the periodic round trip time checks. Although traffic is always allocated to a working link, information about the status of the links is not available unless you add NetLink probing to the configuration.

The probe is made using ICMP Echo Requests (ping) to IP addresses you define. Make sure the Probe IP Addresses you select produce a reliable measurement of the link performance. For example, probing the IP address of an ADSL router usually succeeds even if the ISP network is unreachable, and probing the default gateway provided to you by the ISP may succeed even if the ISP is not able to forward traffic anywhere outside the ISP's own network. Several alternative probe IP addresses can be added to avoid probing failures caused by a probed host going down.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Create NetLink Elements

To introduce a new Internet connection, you must define a NetLink element that represents the Internet connection in the SMC. The NetLink element contains the IP addresses that are used for translating source IP addresses (NAT) so that outgoing connections receive the correct IP address depending on the ISP. This allows the correct routing of the return packets. Each NetLink must have a unique IP address space.

A NetLink can be either static or dynamic. Dynamic NetLinks are supported only for single firewalls. In addition, only Dynamic NetLinks are supported if you configure Multi-Link using 3G modems with a single firewall. You can use the same NetLink with several firewalls. If you want to use NetLinks with a firewall that has several interfaces with dynamic IP addresses, you must create a separate Dynamic NetLink element for each interface with a dynamic IP address.

### Task 2: Configure Routing for NetLinks

The NetLinks must be added under the appropriate interfaces in the Routing tree to support Multi-Link. See [Multi-Link Routing](#) (page 76) for more information.

### Task 3: Combine NetLinks into Outbound Multi-Link Elements

To group the NetLinks together as a single entity, you must create an Outbound Multi-Link element that includes the NetLinks. When you create the Outbound Multi-Link element, you define which NetLinks are included, the load-balancing method for determining which link is selected for each new outbound connection, and whether each NetLink in the Outbound Multi-Link element is an Active or Standby NetLink.

You can create multiple Outbound Multi-Link elements, and each NetLink can belong to more than one Outbound Multi-Link element at the same time.

You can optionally assign QoS Classes to NetLinks in the Outbound Multi-Link element to specify which traffic is routed through which NetLink. NAT rules can alternatively be used to select a particular link, but if you use QoS Classes, traffic can still fail over to other links if the selected link fails. The same QoS class can be assigned to more than one NetLink in the same Outbound Multi-Link element to balance traffic through those selected NetLinks when those links are available. If you want to use QoS class to specify which traffic uses which NetLink, you must assign the QoS class to the traffic in an Access rule or with the QoS policies based on the DSCP codes in the traffic. For more information, see [Bandwidth Management and Traffic Prioritization](#) (page 245).

### Task 4: Create NAT Rules for Outbound Traffic

You activate Multi-Link for outbound connections in the Firewall Policy with NAT rules that match certain traffic for Outbound Multi-Link address translation. Other NAT rules may translate the source addresses in outbound connections to the IP address space of a particular ISP, so that the traffic is automatically routed through a particular link (even if the link fails). Only the part of traffic that matches a NAT rule with the Outbound Multi-Link element is balanced between different links.

Some protocols cannot use dynamic NAT based on IP/port translation. To achieve high availability and load balancing for connections that use these protocols, you can use static NAT with an Outbound Multi-Link element in an outbound load-balancing NAT rule. When static NAT is used, the size of the source network must be the same as the size of the Multi-Link network.

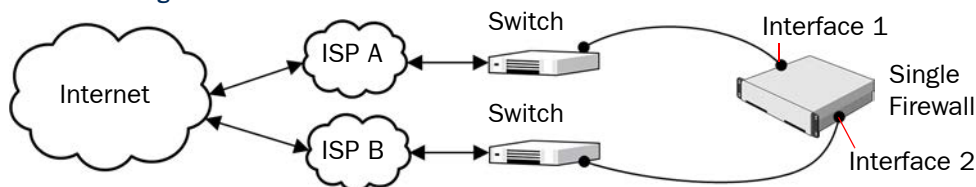
## Using Multi-Link

Multi-Link is mainly used for high availability to ensure that business-critical traffic gets through even when one or more Internet connections fail. Standby NetLinks act as backup Internet connections that are only activated if all the primary NetLinks fail. Using standby NetLinks provides high availability of Internet connectivity, but is less expensive than having multiple NetLinks active at the same time. Using Multi-Link for load balancing can also help reduce costs. Traffic can be balanced between two slower, less expensive, Internet connections instead of one faster connection.

### Multi-Link with a Single Firewall

[Illustration 25.2](#) shows how a Single Firewall's network interfaces are used for Multi-Link.

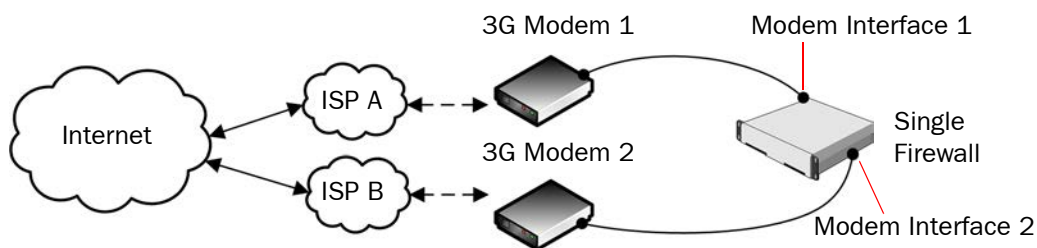
**Illustration 25.2 Single Firewall Interfaces with Multi-Link**



In this example, interface 1 is used as the network interface for Internet traffic that is routed through ISP A. Interface 2 is used as the network interface for Internet traffic that is routed through ISP B. It is also possible to configure Multi-Link by defining two or more IP addresses for a single physical interface - the router behind the interface then forwards the traffic to the different ISPs. However, this is not recommended, as it creates an additional single point of failure at the intermediate router, and the associated cabling and network cards.

You can also configure Multi-Link with single firewalls by replacing one or more physical interfaces with Modem Interfaces and 3G modems.

**Illustration 25.3 Multi-Link Configuration with Two Modem Interfaces on Single Firewall**

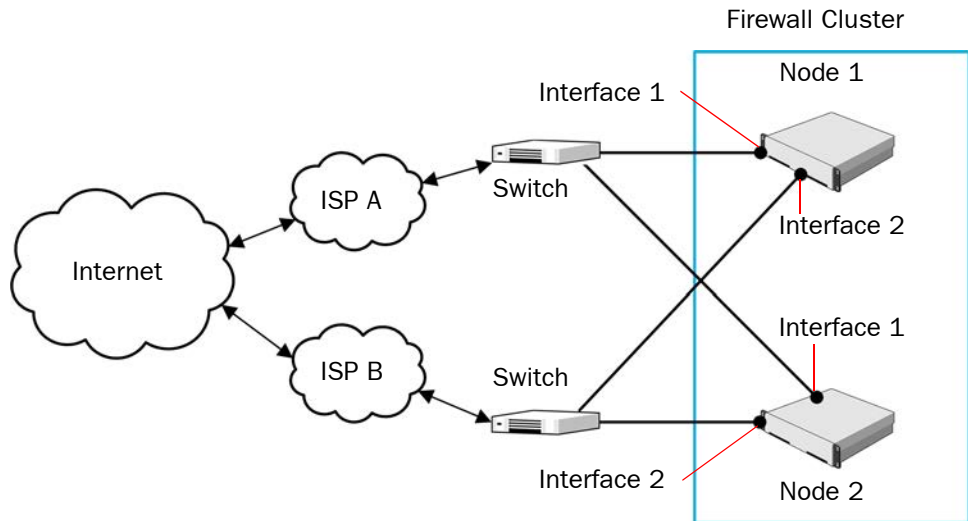


In this scenario, Modem Interface 1 is used for Internet traffic that 3G Modem 1 routes through ISP A. Modem Interface 2 is used for Internet traffic that 3G Modem 2 routes through ISP B.

# Multi-Link with a Firewall Cluster

Illustration 25.4 shows how Multi-Link works with the CVIs of a Firewall Cluster.

Illustration 25.4 Cluster Interfaces for Multi-Link



In this example, the firewall cluster consists of two nodes. On both nodes, Interface 1 is used as the CVI for Internet traffic that is routed through ISP A. Interface 2 is used as the CVI for Internet traffic that is routed through ISP B. Both nodes have one physical interface for each CVI, so that both nodes are physically connected to both routers leading to the Internet.

It is also possible to configure Multi-Link by connecting two CVIs to a single router, which in turn connects to both ISPs. However, this configuration is not recommended, as it creates a single point of failure.

## Using Multiple Outbound Multi-Link Elements

You can create multiple Outbound Multi-Link elements, and each NetLink can belong to more than one Outbound Multi-Link element at the same time. This can be useful, for example, when you want a certain type of traffic to be balanced only between some of the NetLinks, and another type of traffic to be balanced between all of the NetLinks.

## Examples of Multi-Link

---

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

### Preparing for ISP Breakdown

Company A wants to make sure their Internet connection remains available even when one ISP connection fails. The company has subscribed to one Internet connection each from ISP A and ISP B. The administrators decide to use Multi-Link to ensure high availability of Internet connectivity.

The administrators do the following:

1. Create NetLink elements to represent connections to ISP A and ISP B.
2. Place the ISP A and ISP B NetLinks under the correct interfaces in the Routing view.
3. Create an Outbound Multi-Link element and add the ISP A and ISP B NetLinks to it.
4. Define the following NAT rule in the Firewall Policy so that traffic from the internal network (Internal Network element) to destinations that are not internal (Not Internal expression) is handled by the Outbound Multi-Link element (My Multi-Link):

Source	Destination	Service	NAT
Internal Network	Not Internal	ANY	Dynamic load balancing: My Multi-Link

### Excluding a NetLink from Handling a QoS Class of Traffic

Company B has three Internet connections: IPS A, ISP B, and ISP C, which is a satellite link. Because of the long latency in satellite connections, the administrators do not want any VoIP traffic to be routed through ISP C. They decide to use QoS classes so that VoIP traffic is only routed through ISP A and ISP B.

To do this, the administrators:

1. Create NetLink elements to represent connections to ISP A, ISP B, and ISP C.
2. Place the ISP A, ISP B, and ISP C NetLinks under the correct interfaces in the Routing view.
3. Define a QoS class and assign it to VoIP traffic.
4. Create an Outbound Multi-Link element and add the ISP A, ISP B, and ISP NetLinks to it.



5. Select the QoS class for the ISP A NetLink and the ISP B NetLink in the Outbound Multi-Link element properties. No QoS class is assigned to ISP C.
6. Define the following NAT rule for outbound load balancing in the Firewall Policy:

Source	Destination	Service	NAT
ANY	ANY	ANY	Dynamic load balancing: Multi-Link Element

## Balancing Traffic According to Link Capacity

Company B has three ISP connections that have different bandwidths:

- ISP A 20 Mbit/s
- ISP B 10 Mbit/s
- ISP C 4 Mbit/s

The administrators want the traffic to be divided between the NetLinks according to the ratio of their relative bandwidths. This means that ISP A handles twice as much traffic as ISP B and 5 times as much traffic as ISP C. The administrators have already created and configured NetLink elements to represent each ISP connection, so now they:

1. Combine the NetLinks for each ISP connection into an Outbound Multi-Link element and select the Ratio load-balancing method.
2. Define the following NAT rule for outbound load balancing in the Firewall Policy:

Source	Destination	Service	NAT
ANY	ANY	ANY	Dynamic load balancing: Multi-Link Element

## Balancing Traffic between Internet Connections

The administrator at Company B determines that a 4 megabyte Internet connection is needed to handle the volume of traffic their network receives. However, Company B is a small company on a tight budget, and the cost of a single 4 megabyte connection is too high. The administrator decides to subscribe to one 2 megabyte connection each from ISP A and ISP B, and use Multi-Link to balance the load of traffic between the two connections to reduce costs.

The administrator:

1. Creates NetLink elements to represent connections to ISP A and ISP B.
2. Places the ISP A and ISP B NetLinks under the correct interfaces in the Routing view.
3. Creates an Outbound Multi-Link element and adds the ISP A and ISP B NetLinks to it.
4. Defines the following NAT rule in the Firewall Policy so that traffic from the internal network (Internal Network element) to destinations that are not internal (Not Internal expression) is balanced by the Outbound Multi-Link element (My Multi-Link):

Source	Destination	Service	NAT
Internal Network	Not Internal	ANY	Dynamic load balancing: My Multi-Link



# INBOUND TRAFFIC MANAGEMENT

A *Server Pool* balances the load of incoming connections between a group of servers that function as a single entity. You can also use a Server Pool to send dynamic DNS (DDNS) updates to a DNS server to prevent incoming traffic from attempting to use a non-functioning NetLink in a Multi-Link configuration.

The following sections are included:

- ▶ [Overview to Server Pool Configuration](#) (page 236)
- ▶ [Configuration of Server Pools](#) (page 237)
- ▶ [Using Server Pools](#) (page 239)
- ▶ [Examples of Server Pools](#) (page 242)

## Overview to Server Pool Configuration

---

The Server Pool is a built-in load balancer in the Firewall that can be used for distributing incoming traffic between a group of servers to balance the load efficiently and to ensure that services remain available even when a server in the pool fails. The Server Pool has a single external IP address that the clients can connect to. The Firewall uses NAT to distribute the incoming traffic to the different servers.

The server load is distributed to the Server Pool members based on each server's availability. *Monitoring Agents* installed on each server can be used to monitor server availability and load balancing. Alternatively, the server availability can be checked by periodically sending simple ICMP Echo Requests (ping) or by periodically sending TCP strings to check that the expected response is returned. The ping test only checks the server's connectivity, the TCP test checks that specific services are available, and the Monitoring Agents provide additional information about the server's load and functioning.

If the tests or the Monitoring Agent report a server failure, the server is removed from the Server Pool and the connections are distributed to the remaining servers. When a server is removed from the Server Pool, traffic from existing connections can still be sent to the server (since in typical use scenarios the other servers would not be able to handle them in any case) without sending new connections to the failed member. With Monitoring Agents, the server can be completely excluded from handling traffic.

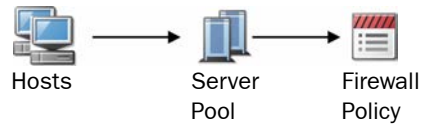
When a previously unavailable server comes back online, existing connections are not redistributed, but some of the new connections that are opened are again directed to the server that rejoins the pool.

Additionally, Multi-Link can be used with Server Pools to provide the connecting clients access to the Server Pool through multiple Internet connections, increasing Server Pool availability.

# Configuration of Server Pools

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

**Illustration 26.1 Server Pool Configuration**

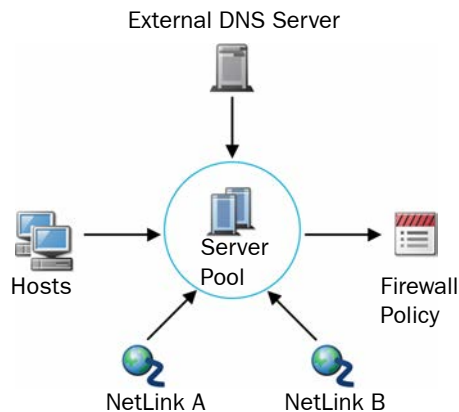


Host elements represent your servers in the SMC. One or more Host elements are added as Server Pool members to a Server Pool element. The Server Pool element must be used in an IPv4 Access rule in the Firewall Policy for incoming traffic to be routed to the pool members. There can be several Server Pools for different services. The Access rules define which traffic is directed to which pool.

## Multi-Link for Server Pools

If you have configured Multi-Link, it can be used to improve Server Pool availability. You can also use Multi-Link with just one server in the Server Pool to take advantage of dynamic DNS updates (as explained below).

**Illustration 26.2 Multi-Link Configuration for a Server Pool**



As an addition to the basic configuration, the NetLinks and (optionally) the external DNS Server are also specified for the Server Pool.

When dynamic DNS updates are not used, Multi-Link is based on assigning an IP address for the Server Pool in each NetLink. The Server Pool's DNS entry on the external DNS server must be configured with an IP address for each NetLink so that clients can access the servers through the different NetLinks. When the connecting client requests the IP address for the Server Pool's DNS name, the DNS server sends the Server Pool's DNS entry with the IP addresses on the different NetLinks. The client connects to one of these addresses and the Firewall allocates the connection to one of the Server Pool members. If the first Server Pool IP address is unreachable, the client can connect to the Server Pool's next IP address on a different NetLink (depending on the client application).

When dynamic DNS updates are used, the firewall updates the DNS entries automatically based on the availability of the NetLinks. When a NetLink becomes unavailable, the Server Pool's IP address for that link is automatically removed from the DNS entry on the external DNS server. When the NetLink becomes available, the IP address is again automatically added to the DNS entry (for more information, see [Dynamic DNS \(DDNS\) Updates](#) (page 239)).

## Default Elements

You can use Server Pools to balance the load between servers without using Multi-Link for inbound traffic management. To do this, you use the special “Not specified” default NetLink element.

## 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* and the *McAfee SMC Administrator's Guide*.

### Task 1: Define Hosts

First, you must introduce each server in the Server Pool into the SMC as a Host element so that the firewall has the IP addresses for sending the traffic. When you define a Host element, you enter the IP address that the firewall uses to contact the server. Only Hosts with IPv4 addresses can be used in the Server Pool.

### Task 2: Combine Hosts into a Server Pool Element

To allow the servers to function as a single entity, you must create a Server Pool element that includes the Host elements. If you have only one server and you want to balance the inbound traffic between your NetLinks, you can define a Server Pool with just one Host. This allows dynamic DNS update Information to be used to prevent contacting clients from attempting to use a NetLink that is out of service.

### Task 3: Configure the External DNS Server

When using static DNS entries, you must ensure that the IP address(es) for your Server Pool are properly entered into your DNS server's records. The Server Pool has one IP address per NetLink (Internet connection) in a Multi-Link configuration and a single IP address in a single-link configuration. If you decide to use dynamic DNS updates, you must configure the DNS server to accept the DDNS updates from the firewall.

### Task 4: Create an Inbound Load-Balancing Rule

Server Pool load balancing is applied to the firewall configuration by adding the Server Pool element in an IPv4 Access rule in the firewall policy (see [Access Rules](#) (page 101) for more information about firewall Access rules). When this rule matches traffic, the Server Pool uses NAT to change the destination IP address to the IP address of the server that the firewall selects for the connection. Reverse NAT (for the replies the server sends back to the client) is handled automatically. No separate NAT rule is required.

## Task 5: Set up Server Pool Monitoring Agents

Optionally, Server Pool Monitoring Agents can be installed to the servers to monitor the availability and load of the Server Pool members. The Monitoring Agent has various in-built tests and it can additionally run external scripts or programs on the server to check that the server is functioning properly.

## Using Server Pools

---

The main points of using Server Pool elements are explained in the preceding sections of this chapter. The sections below illustrate additional points that are useful to know when working with Server Pools.

### Dynamic DNS (DDNS) Updates

Optionally, the Firewall can automatically update dynamic DNS (DDNS) entries for the Server Pool according to the available NetLinks. The firewall engine removes the Server Pool IP addresses for NetLinks that are not available from the DNS entry, and adds the IP addresses back when the NetLink becomes available again. When the connecting client requests the Server Pool's IP address from the DNS server, the client receives a list of IP addresses that only contains IP addresses that work.



**Caution – DDNS can be a security risk because there is no access control. If you must use dynamic DNS updates, do so only after very careful research, planning, and testing.**

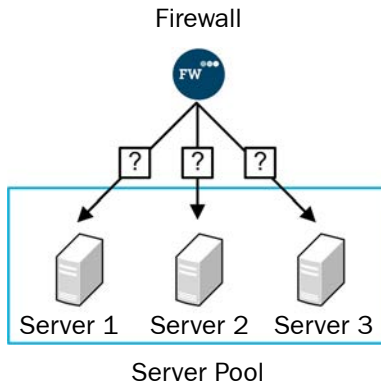
To improve the security of dynamic DNS updates:

- Always place the DNS server(s) behind the firewall (in a DMZ) for protection from IP spoofing.
- Use BIND or an equivalent DNS server that allows you to define which hosts are allowed to send dynamic updates.
- Always consider using static DNS entries instead, as DDNS is not necessarily needed with inbound load balancing. Obviously, in that case the DNS entries are not removed automatically from the DNS server if an ISP fails, but these problems can sometimes be solved otherwise: for example, some web browsers can automatically try other IP addresses if one address does not respond.

## Using Server Pool Monitoring Agents

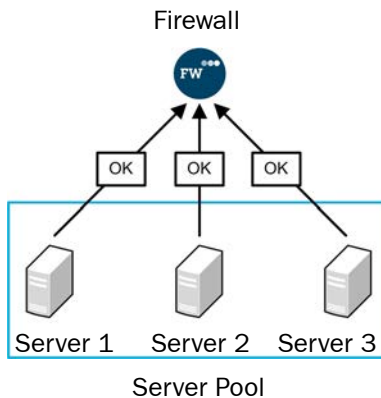
Server Pool Monitoring Agents provide advanced features for monitoring the server load and status. While the ping monitoring relies on checking the server availability and the TCP monitoring relies on checking the availability of specific services, Monitoring Agents run on each server to check the server status and load. The Monitoring Agents can also run system checks on the servers and send log messages to the SMC.

**Illustration 26.3 Firewall Queries Server Pool Monitoring Agents on Each Server**



A Monitoring Agent runs as a service (port 7777/UDP by default) on the Server Pool member. The firewall queries the Monitoring Agent on each Server Pool member to check the status and load of the server.

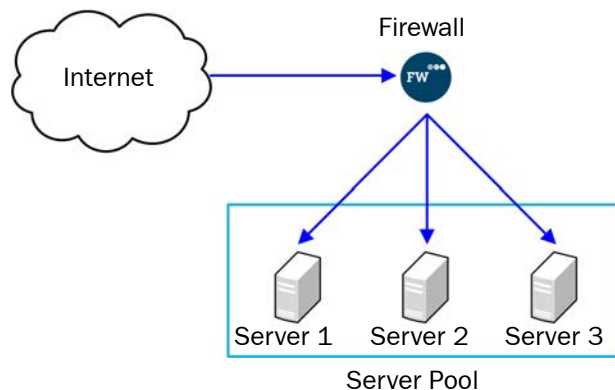
**Illustration 26.4 Server Pool Monitoring Agents Provide Status Information**



The Monitoring Agent on each Server Pool member provides information about the server load and status to the firewall.

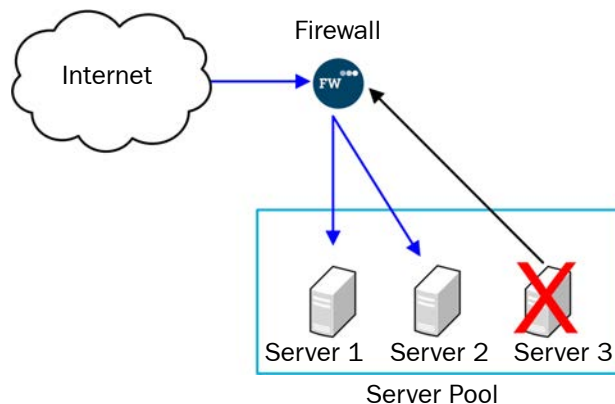


**Illustration 26.5 Firewall Balances Connections Between Server Pool Members**



The firewall balances the incoming connections between the Server Pool members according to the status and load information it receives from the Monitoring Agents.

**Illustration 26.6 Server Pool Monitoring Agents: Test Failure**



The Server Pool Monitoring Agent also has a tester that is configured to run predefined tests or user-defined programs. Automatic action can be configured based on the results of the test. When a test fails, an alert is sent to the Firewall engines. Optionally, the agent can also take the server out of the Server Pool by changing its status from “OK” to “Excluded”. When a server is excluded from the Server Pool, all new connections are directed to other available servers in the pool and the excluded server does not process any connections.

Server Pool Monitoring Agents are available for Windows, and Linux platforms. For more information, see the *McAfee SMC Administrator's Guide*.

## Examples of Server Pools

---

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

### Load Balancing for Web Servers

Company A has three web servers to handle the large volume of traffic its web site receives. The administrators have already created Host elements to represent their web servers and created NAT rules to assign an external address to each web server. Now the administrators want to distribute the load of the traffic between the servers. The administrators also want to monitor the status and the load of the servers, and receive an alert whenever one of the web servers in the Server Pool fails. The administrators decide to set up a Server Pool and install Monitoring Agents on the web servers. To do this, they:

1. Remove the existing NAT rules that translate the IP address of each server so that the Server Pool can do automatic NAT without conflicts.
2. Create a Server Pool element and add the Host elements to it. Because they are not balancing incoming connections to the Server Pool between multiple Internet connections, the administrators select the **Not Specified** NetLink.
3. Install a Server Pool Monitoring Agent on each server in the Server Pool and configure the Monitoring Agents to measure the load average on the server. They also set up a test that checks each server's connectivity every 60 seconds and sends an alert if the test fails.
4. Enable the Server Pool Monitoring Agents in the Server Pool element.
5. Add the following IPv4 Access rules in the Firewall policy to allow HTTP connections from addresses that are not internal (Not Internal expression) to the Server Pool:

**Table 26.1** Server Pool Access Rule

Source	Destination	Service	Action
Not Internal expression	Server Pool element	HTTP	Allow

## Setting up Multi-Link and Dynamic DNS Updates

The administrators at Company A have already configured a Server Pool (see the previous example) but now they want to ensure that web services remain available even when an Internet connection fails, and they want the Server Pool's NetLink addresses to be automatically updated on the DNS server based on the availability of the Internet connections.

The administrators have already configured Multi-Link routing with the necessary NetLink elements to represent each of their Internet connections. A DNS server has also already been set up in the network. The administrators decide to add the NetLinks to the Server Pool and set up Dynamic DNS (DDNS) updates.

The administrators do the following:

1. Configure the DNS server to accept DDNS updates from the firewall.
2. Edit the NetLink elements to add probe IP addresses for testing the NetLinks' status.
3. Edit the Server Pool element's properties and replace the default **Not Specified** NetLink with the NetLink element that represent their Internet connections.
4. Define an External DNS Server element to represent the DDNS-capable server in the SMC.
5. Enable dynamic DNS updates and configure the dynamic DNS settings in the Server Pool element's properties.



# BANDWIDTH MANAGEMENT AND TRAFFIC PRIORITIZATION

Bandwidth management involves defining how the available network link bandwidth is divided between different types of communications to ensure that important traffic is not delayed by less important traffic when the network is congested.

Traffic prioritization is used either independently or in addition to bandwidth management to ensure quick delivery of time-critical communications.

The following sections are included:

- ▶ [Overview to Bandwidth Management and Traffic Prioritization](#) (page 246)
- ▶ [Configuration of Limits, Guarantees, and Priorities for Traffic](#) (page 247)
- ▶ [Using Bandwidth Management and Traffic Prioritization](#) (page 252)
- ▶ [Examples of Bandwidth Management and Traffic Prioritization](#) (page 256)

# Overview to Bandwidth Management and Traffic Prioritization

---

This chapter explains the two Quality of Service (QoS) features: bandwidth management and traffic prioritization. Both features are configured using the same tools. You can use both bandwidth management and traffic prioritization together, or bandwidth management or traffic prioritization individually for any given type of traffic. Bandwidth management and traffic prioritization are not supported on Modem interfaces of single firewalls.

The Firewall can also read and write DiffServ Code Point (DSCP) markers in type of service (ToS) fields. This allows you to integrate the Firewall with other network equipment that implements QoS management in your own or your ISP's network.

## Bandwidth Management

Bandwidth management means giving a guaranteed minimum portion of the available bandwidth to some types of traffic and setting limits for how much of the total available bandwidth each type of traffic is allowed to consume at any given time. You can set a limit, a guarantee, or both for any given type of traffic.

Bandwidth management features can be used to ensure the quality of time-critical communications (even under normal network load), to prepare for malfunctions, and to restrict the total bandwidth needed.



**Note – Bandwidth management applies to outbound traffic only. The firewall can only indirectly limit the bandwidth use of incoming traffic. See [Managing Bandwidth of Incoming Traffic](#) (page 255).**

## Traffic Prioritization

Even under normal traffic conditions, temporary traffic peaks sometimes occur. With many communications, slight delays caused by queuing traffic are not noticeable to the user of the service. However, some connections, such as streaming audio or video, are extremely time-critical, and even relatively minor delays cause noticeable reduction in service quality.

Normally, when packets are queued, they are sent onwards in the same order in which the packets were received. To change this behavior, you can assign priority values to the traffic. For example, you can assign time-critical connections a high priority. High-priority packets are placed before any lower-priority packets in the queue, allowing the fastest possible delivery.

Active Queue Management (AQM) reduces the volume of dropped or retransmitted packets when there is network congestion. AQM monitors the average queue size and uses a scheduling algorithm to determine the statistical probability for dropping incoming packets. If the queue is almost empty, all packets are accepted. As the queue size increases, the probability for dropping incoming packets also increases. When the queue is full, all packets are dropped.

# Effects of Bandwidth Management and Prioritization

Bandwidth management and traffic prioritization improve the quality of service for important traffic, but this may decrease the quality of service for traffic that you define as less important.

Usually the traffic management process allows all connections to proceed, although some traffic may occasionally slow down when the bandwidth limits are reached. If there is prolonged congestion in the network, lower priority traffic eventually starts to time out. If you set priorities without setting any maximum limits or minimum guarantees for bandwidth, high-priority traffic may even use all available bandwidth, blocking all lower-priority traffic.

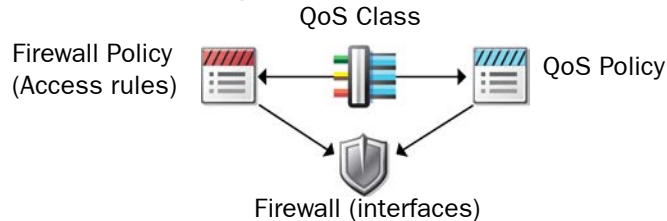
In most situations, the guaranteed minimum bandwidths given to important connections allow traffic to proceed. However, even traffic with bandwidth guarantees may not get through if the network links are not able to maintain their defined throughputs or if the volume of traffic continuously exceeds the throughput. Make sure that your bandwidth limits and guarantees are granular enough to account for situations where a part of the bandwidth is lost, for example, due to an ISP failure in a Multi-Link environment. Keep track of the total bandwidth use so that you can increase the throughput before problems appear.



**Caution** – Inappropriate bandwidth limits and guarantees only disturb traffic. Make sure the guarantees and limits you set are appropriate for the volume of each type of traffic.

## Configuration of Limits, Guarantees, and Priorities for Traffic

**Illustration 27.1** Elements in the QoS Configuration



Bandwidth management and traffic prioritization are configured in *QoS Policies*, which contain rules for the bandwidth guarantees, limits, and priorities you want to set for each type of traffic. The *QoS Policies* do not contain any traffic profile: to define which *QoS rule* affects which type of traffic, the same *QoS Class* element is used in the *QoS Policy* and in one or more *Access rules* to link them.

The *QoS Mode* for each interface defines how QoS is applied to the interface. You can select a *QoS Mode* and define a bandwidth for traffic in the properties of a *Physical*, *VLAN*, *ADSL*, *Tunnel*, or *SSID Interface*.

There are two ways the QoS Class can be applied to a packet:

- If the traffic contains a DSCP code when the traffic enters the firewall, and DSCP handling and throttling or full QoS are enabled, the firewall checks whether the interface that the packets use to enter the firewall has a QoS Policy. If the DSCP Match/Mark tab of the QoS Policy defines a QoS Class match for that code, the selected QoS Class is applied to the traffic. See [Communicating DSCP Markers](#) (page 254).
- When the traffic is inspected against the Firewall policy, the traffic may match a Firewall Access rule that includes a QoS Class. The QoS Class specified in the QoS Class cell is always applied to the traffic, overwriting any other QoS Class the traffic may have been assigned. Access rules are not needed if you only want to use DSCP handling and throttling.

The firewall uses the QoS Class as a matching criterion when it checks if the interface the packet uses to exit the firewall has a QoS Policy. If the QoS Policy contains a rule with the same QoS Class defined, the QoS rule is applied to the connection and the packets are dropped, sent directly, or sent into the queue, depending on the QoS rules and the current traffic situation.

## Default Elements

There are three default QoS Classes: High Priority, Normal Priority, and Low Priority. These are used in the default QoS Policy, **Prioritize**.

The Prioritize QoS Policy is a sample policy that contains simple rules for prioritizing traffic according to the three default QoS Classes. High Priority traffic is assigned the highest possible priority value of 1, the Normal Priority value is 8, and Low Priority is assigned the lowest possible value of 16. The default Prioritize policy does not provide any bandwidth guarantees or limits.



**Caution – If you set priorities without setting any bandwidth limits or guarantees, high-priority traffic may use all available bandwidth, blocking all lower-priority traffic.**

If the default Prioritize policy is sufficient for you, you can use the default QoS Classes and the Prioritize policy as they are. Just add the QoS Classes to Access rules as explained in [Task 3: Assign QoS Classes to Traffic](#) (page 251). Then, configure the interface(s) to use the Prioritize QoS Policy. See [Task 4: Define QoS for Interfaces and VPNs](#) (page 251).

If you want to define bandwidth guarantees or limits, or if you want to have more control over the traffic priorities, you must configure QoS as explained in the configuration workflow below.



# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define QoS Classes

QoS Classes can be used to collect QoS statistics about traffic, or to create a link between the Access rules and the QoS Policy. When traffic matches an Access rule, the QoS Class defined in the rule is applied to the traffic.

You can create as many QoS Classes as necessary. The QoS Policy must not have overlapping rules. For this reason, you must create one QoS Class for each rule you plan to add in the QoS Policy (each type of traffic must have its own QoS Class). The same QoS Class can be used in multiple firewall Access rules, so several Access rules can point matching traffic to the same QoS rule.

## Task 2: Define QoS Policies

QoS Policies are tables of QoS rules and DSCP Match/Mark rules. If you only want to collect QoS statistics about traffic, you do not need to define a QoS Policy.

Because the QoS rules are separate from the Access rules, you can flexibly design the rules. For example, you can create different QoS Policies for different interfaces of the same firewall.

All cells in the QoS rules are applied to outgoing packets. When Full QoS is used, packets that do not match a QoS rule are handled with priority 8 (middle of the scale) without any bandwidth guarantees or limits. The cells in the QoS rules are explained in the tables below.

**Table 27.1** QoS Rule Cells

Cell	Explanation
ID	An identifier that shows the order of the rules. The number changes as you add, remove, and move rules.
QoS Class	A list of the defined QoS Classes, which match the QoS rules to traffic. The QoS Class is assigned to traffic in Access rules or through the DSCP Match cell in the QoS Policy (see <a href="#">Table 27.2</a> ).
Guarantee	Sets the minimum bandwidth given to this type of traffic under any conditions. The guarantee can be set in kilobits per second or as a percentage of the available bandwidth.
Limit	Sets the maximum bandwidth that this type of traffic is allowed to consume at any single moment as kilobits per second or as a percentage of the available bandwidth.
Priority	Assigns this type of traffic a number that determines the order in which the firewall sends packets onwards if there is not enough bandwidth available to send all packets onwards directly, so that packets have to be queued. The priority is a number between 1 (highest priority) and 16 (lowest priority). Higher-priority packets are inserted in the queue ahead of any lower-priority packets already in the queue.

**Table 27.1 QoS Rule Cells (Continued)**

Cell	Explanation
Weight	The weight of the QoS Class controls the distribution of bandwidth between QoS Classes with the same priority after the Guarantees for the QoS Classes are reached. The weight of the QoS Class is entered as a value from 0 to 100. The relative weight of each QoS Class is displayed in parentheses as a percentage.
Latency	The average time packets are held in the queue for Active Queue Management (AQM). The engine makes a best effort to handle the packets within the specified time, but the Latency value is not a guarantee.
Comment	A short free-form comment for your reference.
Rule Name	Contains a rule tag and optionally a rule name. Name: <i>(Optional)</i> Name or description for the rule. Displayed alongside the rule tag. Tag: <i>(Not editable)</i> The unique identifier of the rule in this policy. It contains a static part that does not change when rules are added, removed, or moved, and a changing part that indicated the version of the rule.

All cells in the DSCP Match/Mark rules except the DSCP Match cell are applied to outgoing packets. If packets do not match any DSCP Match/Mark rule, any DSCP markers in the traffic are preserved, but have no effect on how the Firewall handles the traffic. The cells in the DSCP Match/Mark rules are explained below.

**Table 27.2 DSCP Match/Mark Rule Cells**

Cell	Explanation
ID	An identifier that shows the order of the rules. The number changes as you add, remove, and move rules.
QoS Class	A list of the defined QoS Classes, which match the QoS rules to traffic. The QoS Class is assigned to traffic in Access rules or through the DSCP Match cell (see below).
DSCP Match	Assigns the rule's QoS Class to traffic when the DSCP code (ToS field) defined in this cell is seen in traffic. The value specified in this cell is the only option that is applied on the interface that the packets use to enter the firewall.
DSCP Mark	Defines the DSCP code (ToS field) that is written to packets that match this DSCP Match/Mark rule when the packets exit the firewall. The DSCP Mark allows you to communicate the priority of this traffic to other devices that support QoS. You can also use the cell to clear the DSCP classification set by other devices by entering 0 as the value (shown in the policy as 0x00).
Comment	A short free-form comment for your reference.
Rule Name	Contains a rule tag and optionally a rule name. Name: <i>(Optional)</i> Name or description for the rule. Displayed alongside the rule tag. Tag: <i>(Not editable)</i> The unique identifier of the rule in this policy. It contains a static part that does not change when rules are added, removed, or moved, and a changing part that indicated the version of the rule.

### Task 3: Assign QoS Classes to Traffic

The same QoS Class can appear in several Access rules. You can insert a QoS Class in an Access rule that allows traffic or in an Access rule that uses the Continue action to set the same QoS Class for several rules. This way, you can assign a specific QoS Class to any traffic that you can match with a single Access rule. If you only want to collect QoS statistics about traffic, you only need to define Access rules to assign a QoS Class to the traffic.

If you only want to map existing DSCP codes in traffic to QoS Classes, no Access rules are required. However, the same traffic must not match an Access rule that sets a QoS Class, since the Access rule overwrites the QoS Class that is assigned based on the DSCP code.

### Task 4: Define QoS for Interfaces and VPNs

The QoS Mode for each interface defines how QoS is applied to the interface. By default, No QoS is selected. You can select a QoS Mode and define a bandwidth for traffic in the properties of a Physical, VLAN, ADSL, Tunnel, or SSID Interface. You can select different QoS Modes for each interface. It is not mandatory to use QoS on all interfaces of the same engine. The following table describes how QoS is applied for each QoS Mode.

Table 27.3 QoS Modes

QoS Mode	Explanation
No QoS	None of the QoS features are applied to the interface.
QoS Statistics Only	Allows the collection of QoS Class-based counters without activating any other QoS functionality. A QoS Policy is not required. The QoS Class for the packet comes from the QoS Classes that are applied in the Access rules. See <a href="#">Collecting QoS Class-Based Statistics</a> (page 255).
DSCP Handling and Throttling	QoS guarantees, limits, and priorities are applied to the traffic either according to the QoS Classes set by the Access rules or by a DSCP Match in the QoS Policy. DSCP markers are read and/or written for the traffic according to the QoS Policy. You must specify bandwidth guarantees and limits in the QoS Policy in kilobits per second. Only a QoS Policy is required.
Full QoS	QoS guarantees, limits, and priorities are applied to the traffic according to the QoS Policy. DSCP markers are read and/or written for the traffic according to the QoS Policy. You can specify bandwidth guarantees and limits in the QoS Policy in kilobits per second or as a percentage of the available bandwidth. A QoS Policy and bandwidth definitions for the interface are required.

When you use Full QoS on an interface, you must define the available throughput in the Firewall element's properties for each Physical, VLAN, ADSL, Tunnel, or SSID Interface whose throughput you want to manage. There is no automatic mechanism for finding out how much bandwidth each interface has. The throughput must always correspond to the actual throughput the interface offers to connecting clients, that is, the outbound bandwidth of an Internet link that is connected to the interface. If there are VLANs on a Physical Interface, the settings are only available in the properties of each VLAN.



**Note** – When you define the throughput of an interface, the Firewall always scales the traffic to this throughput. Take special care that you set the throughput value correctly.

If you are using Multi-Link in a configuration where more than one NetLink is connected to the same Physical Interface, the throughput setting depends on the selected Multi-Link configuration:

- If you are using load-balancing Multi-Link, set the throughput to the combined outbound bandwidth of all the Internet links behind the Physical Interface.
- If you are using standby NetLinks, set the throughput to the outbound bandwidth of the primary (active) NetLink. In cases where the bandwidth of the backup NetLink is lower, it is recommended to set the throughput to the speed of the primary NetLink to fully utilize its capacity, since the primary link is used most of the time.

Policy-based VPNs can optionally use a QoS Policy to define how DSCP matching and/or marking is done for VPN traffic. In policy-based VPN traffic, the DSCP mark for the encrypted ESP packet is normally inherited from the plaintext packet. Selecting a QoS Policy for the policy-based VPN makes it possible to mark the ESP packet after encryption. Since the total throughput is undefined, Guarantees and Priorities cannot be used for policy-based VPN traffic. For general information about IPsec VPNs, see [Overview to VPNs](#) (page 261). For more information about policy-based VPNs, see [Policy-Based VPN Configuration](#) (page 269).

## Using Bandwidth Management and Traffic Prioritization

---

Bandwidth management and traffic prioritization are generally used for the following purposes:

- To ensure the quality of service for time-critical communications during occasional traffic peaks. This may be necessary even if there is generally ample bandwidth available, since even very short periods of congestion may degrade the quality of some types of communications.
- To prepare for severe congestion caused by the loss of network links when there are technical problems. In a Multi-Link environment, you can have several NetLinks. Ideally, the throughput of these links is large enough that each link can alone handle the traffic. However, if this is not a viable option, it may become necessary to choose which connections are given priority if network connections are lost.
- To restrict non-essential services to reduce the total bandwidth needed if it is not possible to increase throughput of the network links or add new links. For example, important services (such as VPN connections to branch offices and clients' connections to the company extranet) can be given priority at the expense of employees' web surfing (either generally for all HTTP connections or based on the web servers' IP address).

## Designing QoS Policies

Each QoS Class can appear in only one (active) rule on each tab of a QoS Policy. The same QoS Class can be used on both the QoS tab and the DSCP Match/Mark tab of the same QoS Policy. The order of the QoS rules does not matter. The classification of the traffic is made using Access rules, so the purpose of the QoS Policy is to determine which limit, guarantee, or priority traffic marked with a certain QoS Class receives.

Except for the QoS Class, all other cells for rules on the QoS tab are optional, but at least one of the other cells must be filled for the rule to have any effect on the traffic. None of the cells exclude any of the other cells, so you are free to select which cells you want to use for any given QoS Class. It is not necessary to define the use of all available bandwidth in your QoS Policy. The bandwidth outside the guarantees as well as any bandwidth within the guarantees that is not used for actual traffic at any given time is used to handle the traffic that has no specific QoS rule on the normal first-come-first-served-basis using the medium priority of 8.



**Caution – If your guarantees are equal to the total throughput of an interface, any traffic without a guarantee is completely blocked if all guarantees are fully used.**

When you save the QoS Policy, the system checks if there are contradictions within each rule, such as a rule that sets a limit that is lower than the guarantee for the same rule. When you refresh the firewall's configuration, the QoS Policies defined for the firewall's interfaces are checked again, comparing the QoS rules to the throughput values you have set for the interfaces. At this point, the values may be automatically scaled down if the sum of all the guarantees in the QoS Policy exceeds the interface's throughput. However, the values are never scaled up.

The values for the bandwidth limits and guarantees can be entered either in kilobits or as percentages of the total throughput of the interface. Technically, nothing prevents you from using both ways of entering values even in the same rule. However, it is recommended to use one way of entering the values consistently throughout each QoS Policy. Using mixed methods of entering the values makes it more difficult for the administrators to read the QoS policy and may prevent the system from checking for issues when you save the QoS Policy, as the throughput(s) of the interfaces are not known at this point. If the QoS Policy cannot be checked when you save it, it is checked when the Firewall Policy is installed. Mistakes in the QoS Policy may prevent you from installing or refreshing the Firewall Policy.

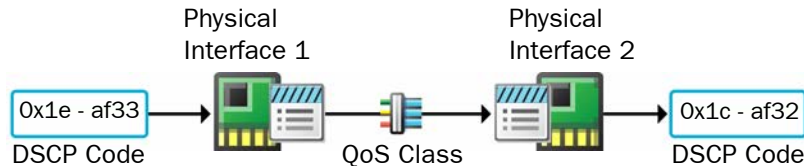
## Communicating DSCP Markers

You and your ISP may have routers that also make decisions on handling the packets based on the priority of the traffic. DSCP (DiffServ type of service field) markers in the traffic are a standard way to indicate priorities in network traffic. You can communicate traffic priorities between the Firewall and other network equipment using DSCP to ensure the best possible quality of service for important traffic.

It is possible to apply a DSCP mark to a particular type of traffic without configuring Access rules to apply a QoS Class to the traffic. The matching is done based only on the QoS Policy. When a packet that matches a particular protocol comes in, the DSCP markers are read and assigned a QoS Class according to the DSCP Match/Mark rules of the QoS Policy. When the packet is sent out, a DSCP mark is written in packets based on the QoS Class according to the DSCP Match/Mark rules of the QoS Policy on the interface through which the traffic leaves the firewall.

Two QoS Policies on two Physical Interfaces can be used together to “translate” between two different DSCP schemes as shown in [Illustration 27.2](#).

**Illustration 27.2 Translating Between Two DSCP Schemes**



In the illustration above, the packets arrive at Physical Interface 1. The firewall reads the existing DSCP value and compares it to the QoS Policy assigned to Physical Interface 1. The policy has a DSCP Match rule for the DSCP marker with an associated QoS Class, which is then assigned to this traffic.



**Note – The same traffic must not match any firewall Access rule with a QoS Class definition, because the QoS Class in the Access rule overrides the QoS Class that is assigned based on the DSCP marker.**

When the packets are sent out through Physical Interface 2, the Firewall checks the QoS Policy assigned to this Physical Interface. In this QoS Policy, there is a DSCP Match/Mark rule that defines that traffic with the assigned QoS Class will be marked with a DSCP marker specified in the rule, and the firewall overwrites the original DSCP marker before sending the packets onwards.

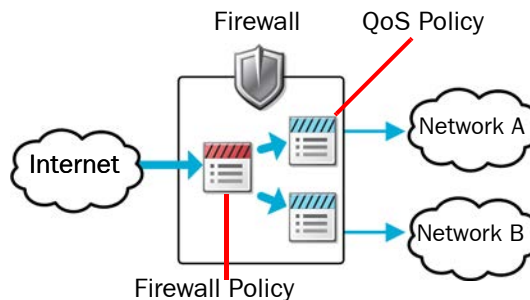
# Managing Bandwidth of Incoming Traffic

Bandwidth management and prioritization usually help manage the quality of service for traffic going out through Internet links, which are often the choke point in a corporate network due to the costs associated with increasing the bandwidth.

Outbound traffic can be controlled easily and accurately with QoS Policies, since the firewall has full control of what traffic it forwards. Controlling incoming traffic is more difficult, since by the time the firewall sees the traffic, the packets have already travelled through the congested links and taken their share of the limited bandwidth. Still, you may be able to limit some types of incoming traffic in a limited way. In this case, only limits apply. To set guarantees and priorities to traffic, you must consider other solutions, such as to arrange with your ISP(s) that they implement traffic management before the traffic is passed to your Internet links.

To limit the bandwidth incoming traffic consumes, you can apply the QoS Policy on the firewall's interfaces connected to the *internal* network. This arrangement is shown in [Illustration 27.3](#).

**Illustration 27.3** Applying QoS to Incoming Traffic



In the illustration above, traffic is checked against the Firewall Policy and allowed traffic is assigned a QoS Class. At the interfaces connected to the internal network, the QoS Policies limiting the bandwidth use are enforced as the traffic is sent onwards.

Limiting the bandwidth of incoming traffic in this way requires that the application that is the source of the traffic scales down the transmissions to match the available bandwidth. If an application does not scale down its bandwidth use, any limits you set have no effect, and the only option is to control the traffic before it reaches the firewall (by your ISP).

## Collecting QoS Class-Based Statistics

QoS Class-based statistics items are available even when QoS is not used for bandwidth management and traffic prioritization. Selecting the “QoS Statistics Only” QoS Mode for an interface allows the collection of QoS Class-based counters without activating any other QoS functionality. No QoS Policy is needed in this case, but you must define Access rules to apply QoS Classes to traffic. QoS Class-based statistics items can be used in Overviews and Reports.

# Examples of Bandwidth Management and Traffic Prioritization

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

## Ensuring Quality of Important Communications

In this example, Company A has two offices, one in Italy and one in France. The company has decided to replace phone lines with VoIP telephony.

**Illustration 27.4** Company A Networks



The illustration above shows the two offices and the traffic between them. Telephone and e-mail connections are both a very important tool for the employees, who use these services to communicate with team members at the other office. Additionally, employees at the Italian office must be able to use web-based tools at the French site. The administrators determine the priorities as follows:

- The VoIP streaming audio is not only important, but it is also a time-critical service. VoIP streaming audio must have the highest priority.
- Even though business e-mail is important, e-mail does not need to be delivered immediately after it has been sent. This traffic can be assigned a lower priority.
- The web-based services are not time-critical, but delays and time-outs may be annoying to the workers. The company decides to give these a lower priority than VoIP, but a higher priority than e-mail. It is not necessary to define a specific QoS Class for the medium-priority traffic because all traffic that is not specifically classified is assigned a medium priority.

The internal networks are so fast that there is no need to implement any QoS Policies for those interfaces, so only the interfaces connected to the Internet need a QoS Policy. The administrators decide that the same QoS Policy can be used at both sites, and that the default elements and the default Prioritize policy are suitable for use without customization. So now they:

1. Add the QoS Class “High Priority” to Access rules that permit VoIP traffic.
2. Add the QoS Class “Low Priority” to Access rules that permit e-mail traffic.
3. Define the QoS Policy Prioritize to be used on the interfaces connected to the Internet at both the Italian and the French firewall, along with the interface throughputs.
4. Refresh the policies of both firewalls.



## Preparing for ISP Breakdown

Company B decides to use Multi-Link to ensure high availability of network connections for important business communications. The company, an engineering subcontractor, is mainly concerned about two types of connections:

- A VPN connection they have for accessing the internal tools and resources of an important client when doing work for them.
- HTTPS connections to the extranet server that the company's clients use to check the status of projects.

The company is on a tight budget, and the cost of having enough bandwidth in both links to handle all traffic even during peak hours is deemed too high. Therefore, they decide that only the two most important types of traffic must get through if one ISP link goes down during peak hours. The company determines that 500 kbps is enough to handle those connections, so they decide to subscribe to two different ISPs for one 512 kbps link from each. None of the communications are especially time-critical, so the company decides not to prioritize the traffic. Then the administrators:

1. Create a new QoS Policy and two new QoS Classes, called VPN and Extranet.
2. Create the QoS rules for the important connections filling in the following cells:

**Table 27.4 QoS Rules in QoS Policy for Company B**

QoS Class	Guarantee
VPN	400
Extranet	100

3. Add the QoS Class "VPN" to the VPN rule for outbound traffic in the Firewall's Access rules.
4. Add the QoS Class "Extranet" to the Access rule that allows the outbound connections from the company extranet.
5. Define the throughputs and select the new custom QoS Policy to be used for both Physical Interfaces that correspond to the two ISP links on the firewall.
6. Refresh the policy of the firewall.

## Limiting the Total Bandwidth Required

Company C has experienced a radical increase in the amount of network traffic. It seems that many employees have started to use bandwidth-intensive services, downloading large files, and listening to Internet radio in high-quality mode while they work. The situation is getting to the point where business communications are starting to slow down. The management would rather prohibit connections that are not directly work-related than fund the required increase in bandwidth.

However, the administrators suggest a different approach: limiting the portion of the bandwidth that the non-essential traffic can use so at least some of the employees can still listen to Internet radio while the business communications are guaranteed the bandwidth they need. To assure the quick delivery of time-critical business communications, they also decide to prioritize the traffic using the three default QoS Classes. The administrators:

1. Create a new custom QoS Policy with the following rules:

**Table 27.5 QoS Rules in QoS Policy for Company C**

QoS Class	Priority	Guarantee	Limit
High Priority	1	35%	90%
Normal Priority	8	55%	90%
Low Priority	16	5%	50%

- Normal Priority traffic gets the largest guaranteed portion of the bandwidth because it has the largest volume.
  - High Priority and Normal Priority traffic can each use up to 90% of the bandwidth. Low Priority traffic cannot consume more than 50% of the available bandwidth even if there is more bandwidth available. In this configuration, there must be traffic in at least two of the classes for the bandwidth to be utilized up to 100%.
  - Even Low Priority traffic is given a guarantee of 5% of the bandwidth to avoid total loss of service, which would cause even more complaints from users than slowed-down service will.
2. Place a Continue rule at the top of the firewall Access rules that includes the Normal Priority QoS Class. This way, all traffic that is not specifically classified as High Priority or Low Priority is classified as Normal Priority.
  3. Edit the Access rules to assign QoS Classes to traffic:
    - Place the High Priority QoS Class into Access rules that permit important traffic.
    - Place the Low Priority QoS Class into Access rules that permit low-importance traffic.
  4. Define the throughputs and select the new custom QoS Policy to be used for the Physical Interfaces connected to the Internet on the firewall.
  5. Refresh the policy of the firewall.

# VIRTUAL PRIVATE NETWORKS

---

**In this section:**

[Overview to VPNs](#) - 261

[Policy-Based VPN Configuration](#) - 269

[Route-Based VPN Configuration](#) - 295



## CHAPTER 28

# OVERVIEW TO VPNs

This chapter provides an introduction to Virtual Private Networks (VPN) and the IPsec standards on which VPNs in McAfee Firewall/VPN are based.

The following sections are included:

- ▶ [Introduction to VPNs](#) (page 262)
- ▶ [IPsec VPNs](#) (page 262)
- ▶ [IPsec VPN Topologies for Policy-Based VPNs](#) (page 266)

# Introduction to VPNs

---

Virtual Private Networks (VPNs) allow secure communications over insecure networks. VPNs secure the communications through authentication, encryption, and integrity checking mechanisms:

- *Authentication* provides a way for the devices at both ends of the VPN to confirm the identity of the other device. This prevents malicious parties from obtaining confidential data or access by posing as a legitimate host.
- *Encryption* scrambles the transmissions to prevent anyone from viewing the content, providing privacy for the communications.
- *Integrity checking* is used to detect whether packets have been modified in transit, which could be a sign of malicious tampering (or simply transmission errors).

There are two types of VPNs in McAfee Firewall/VPN. The main difference between the two is how traffic is selected to be sent into the VPN:

- *Policy-based VPNs* are configured using **VPN** elements. The firewall IPv4 Access rules define which traffic is sent to the VPN and which traffic is allowed out of the VPN.
- *Route-based VPN tunnels* are configured using the **Route-Based VPN** element. Firewall interfaces can be designated as end-points for a Route-Based VPN tunnel. Any traffic that is routed to those interfaces is sent into the VPN tunnel.

Both types of VPNs are based on the *IPsec* (Internet Protocol security) standards.

This chapter concentrates on general VPN concepts. The next chapters, [Policy-Based VPN Configuration](#) (page 269) and [Route-Based VPN Configuration](#) (page 295), explain how VPNs are configured.

## IPsec VPNs

---

McAfee Firewall/VPN uses the IPsec protocol standards to implement VPNs at the IP network layer. This means that IPsec allows any IP traffic to be transported in the VPN regardless of which higher-level protocol the traffic uses on top of the IP protocol. Hosts can communicate through the VPN as if it were a normal link without the need for application-specific configurations on the gateway device. IPsec is part of both the IPv4 and IPv6 standards. IPsec is defined in RFC 4301.

Although this section introduces general concepts of the IPsec standard, most of these concepts are also directly present as options in the VPN-related dialogs and views in the Management Client.

## Tunnels

When there is traffic that needs to be sent through a VPN, the gateway or VPN client at the communication source contacts the gateway at the communication destination to establish a VPN tunnel. The original packets are encapsulated when they enter the tunnel, and de-encapsulated when they exit the tunnel at their destination. In between, only the encrypted packets can be detected in the traffic.

The hosts that communicate through the tunnel are not aware of the VPN. From the point of view of the communications going through the tunnel, the situation is no different than if the two gateways were connected directly to each other.

## Security Associations (SA)

For any communications to be able to use the VPN, the gateways must construct and maintain the VPN tunnels. To do this, the gateways must decide on which settings to use between each other and store this information so that it can be used for handling the traffic throughout the lifetime of the VPN tunnel.

The settings that are used for a tunnel are stored in *Security Associations (SA)*. There are two SAs for each VPN tunnel: one for outgoing traffic, and another one for incoming traffic.

The term SPI (security parameter index) is sometimes used in conjunction with SAs in IPsec VPNs. SPIs are used to identify the SAs.

For security reasons, each SA has an expiration time after which the gateways discard the old SAs and agree on new ones if there is still traffic going through the VPN.

## Internet Key Exchange (IKE)

The SAs are created in a process called the *Internet Key Exchange (IKE) negotiations*. During the IKE negotiations, the VPN gateways agree on the parameters to use, such as the encryption keys and the authentication methods. This information is then stored in the SAs. Both IKEv1 and IKEv2 are supported with McAfee Firewall/VPN.

The IKE negotiations consists of two phases:

- During the *IKE SA negotiations*, the gateways authenticate themselves to each other and establish a secure (encrypted) channel for the IPsec SA negotiations. Authentication in IKE SA negotiations can be done with signatures, or with pre-shared keys. These parameters are then stored in IKE SAs.
- During the *IPsec SA negotiations*, the gateways select parameters for encrypting the traffic going through the VPN tunnels. These parameters are then stored in IPsec SAs.

The IPsec SA negotiations are much faster than the IKE SA negotiations. Since IKE SA negotiations involve quite heavy computation, it is common to configure the IKE SAs to expire less frequently than the IPsec SAs.

IKEv2 also provides support for *IKEv2 Mobility and Multihoming Protocol (MOBIKE)* protocol. MOBIKE enables transparent recovery for VPN clients when the VPN clients change their IP addresses (for example, in roaming use when a laptop is connected to a different network while a VPN connection is open). MOBIKE also allows the IP addresses associated with IKE SAs and IPsec SAs to change in a VPN Multi-Link configuration. When a VPN client fails to connect to a gateway, it checks if another gateway address is available. If the VPN client can connect using the new gateway address, the gateway's IP address is updated in the IKE SAs and the IPsec SAs and VPN traffic can continue uninterrupted. There is no need to renegotiate the SAs.

## Perfect Forward Secrecy (PFS)

It is possible to configure the IKE SA negotiations to occur less frequently than IPsec SA negotiations to improve performance. However, this kind of arrangement is less secure than renegotiating both phases again, since the IPsec SA negotiations generate encryption keys based on information from the IKE SA negotiations. To counter this, you can activate Perfect Forward Secrecy (PFS), which ensures that the encryption keys for IPsec SA negotiations are created independently. When the negotiations are independent of each other, any one key that is compromised can only be used to decrypt communications sent between two IPsec SA negotiations instead of potentially breaching all communications between two IKE SA negotiations, which cover a longer period of time.

## Authentication Header (AH) and Encapsulating Security Payload (ESP)

Once a VPN tunnel is established, any traffic going through the tunnel is sent either as *Authentication Header (AH)* or *Encapsulating Security Payload (ESP)* packets:

- The IPsec AH protocol does not provide data encryption, so plain AH does not result in a VPN in the full meaning of the word. The transferred data can be seen by anyone who can intercept the packets in transit. AH can be used to provide authentication and data integrity in communications that do not need encrypting.
- The IPsec ESP protocol provides authentication, encryption, and integrity checking, providing secure data transfer. This protocol corresponds to what is usually meant with the term VPN, as the transferred data is hidden from outsiders.
- The standards also have a provision to use a combination of ESP and AH, but this option does not provide significant security improvements in the type of VPNs McAfee Firewall/VPN establishes, and is not supported in the current version of the firewall.

As a general guideline, use ESP for any normal VPN tunneling (data encapsulated in ESP payload). There is very rarely any need to use AH alone. AH alone may be used when no encryption is required for the data, but ESP with Null encryption can also be used to achieve the same purpose.



# Authentication

Authentication always requires exchange of information between the two authenticating parties. The information exchange must be done securely in such a way that the exchanged information cannot be used by others even if they are able to intercept it.

The confidentiality of authentication exchanges is most often achieved through digital signatures or through encrypting the authentication messages with a pre-shared key.

- Digital signatures use a public-private key pair to sign messages. This method requires that digital certificates signed by a mutually trusted Certificate Authority (CA) are present.
- VPN authentication with a pre-shared key does not require the presence of digital certificates. It requires the exchange of a secret encryption key that is known by both communicating parties.

Both methods can be secure enough for VPNs if used correctly, but the security of the pre-shared key method is much more dependent on administrator actions. If pre-shared keys are used for authentication, the keys must be long and random to be sufficiently secure. The pre-shared key must be kept absolutely secret, since the security of this setup completely relies on the assumption that nobody except the legitimate parties know the key.

The authentication and encryption methods available are listed in [Supported Authentication and Encryption Methods](#) (page 279).

# Tunnel and Transport Modes

IPsec supports two different modes for securing the traffic.

- *Tunnel mode* encapsulates the complete original packet into a new IPsec packet and is meant for gateway-to-gateway and client-to-gateway VPNs. Policy-based VPNs always use tunnel mode. You can optionally use the Route-Based VPN in Tunnel Mode. See [Using the Route-Based VPN in Tunnel Mode](#) (page 301).
- *Transport mode* does not encapsulate the packets into new IPsec packets. Instead, additional encapsulation, such as *Generic Routing Encapsulation (GRE)* or *IP in IP (IP-IP)*, is used to encapsulate the tunneled traffic. Transport mode is used in the Route-Based VPN.

# IPsec VPN Topologies for Policy-Based VPNs

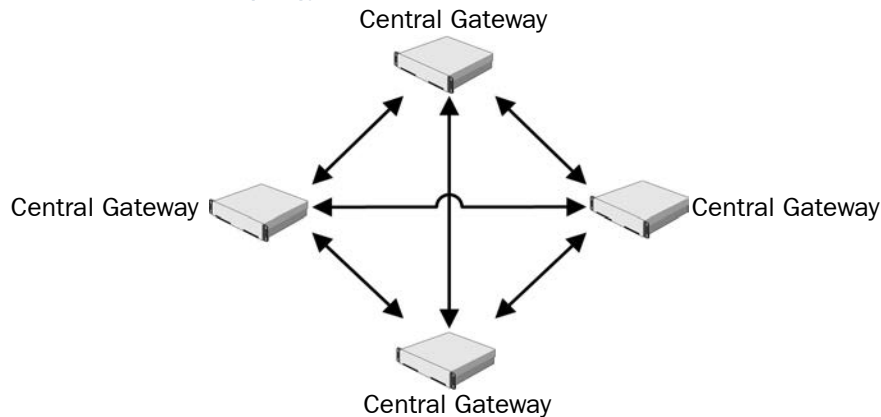
Policy-based VPN tunnels can be defined using different topologies:

- *Full-mesh topology* connects each site to every other site in the same VPN. All gateways are central gateways, which means that all gateways can establish tunnels with all other gateways in the VPN.
- *Star topology* connects sites behind satellite gateways to the sites behind central gateways. No VPN tunnels are established between the satellite gateways.
- *VPN hub topology* routes gateway-to-gateway or client-to-gateway connections through a central (hub) gateway to other sites through other gateway-to-gateway VPNs. The hub is usually a central gateway, but can also be a satellite gateway.

Usually, the VPN configuration of any organization is a mix of the different topologies, but the basic scenarios shown here are still a useful starting point for planning.

The full mesh topology, presented in [Illustration 28.1](#), is formed between sites that must all be able to connect to any other site.

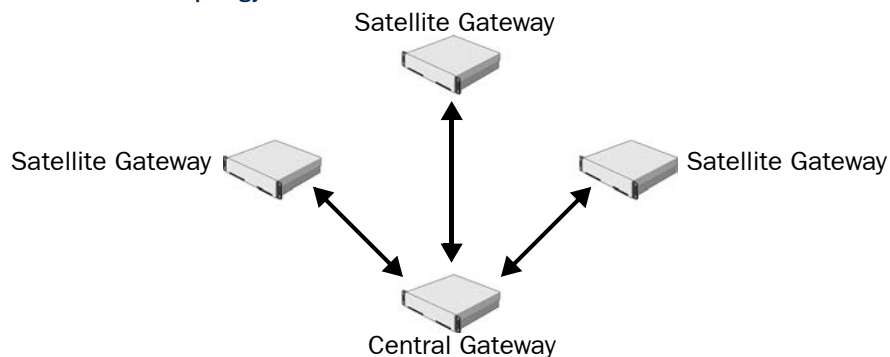
**Illustration 28.1** Full Mesh VPN Topology



In a full-mesh topology, all VPN gateways are defined as central gateways so that each gateway can establish, when needed, a VPN tunnel with the other gateways in the VPN. This allows VPN communications from any one site to any other site.

When VPNs are formed with partner organizations or small remote offices, VPN connectivity is needed between a number of remote sites and a main site, but not from one remote site to another. This results in a star topology ([Illustration 28.2](#)).

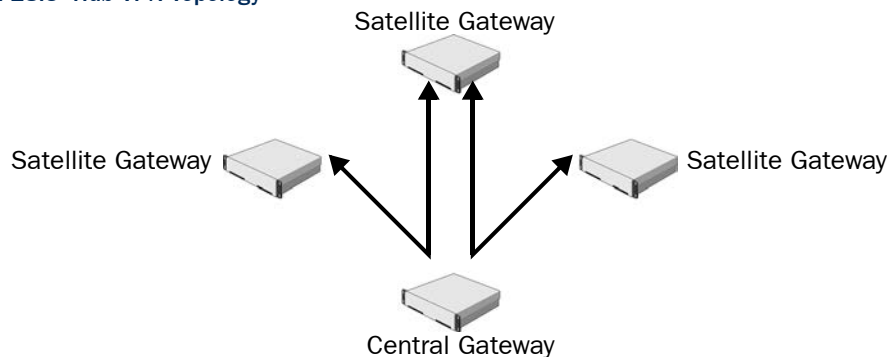
**Illustration 28.2 Star VPN Topology**



The star topology is defined with satellite gateways that connect only to the central gateway(s). There is no VPN between the satellite gateways. This reduces the number of VPN tunnels that the gateways need to maintain compared to full-mesh topology. Having less tunnels can therefore save resources on the remote gateways.

Sometimes the star topology is preferred even if there needs to be VPN connectivity between the remote offices. In this case, the central gateway can be used as a hub that relays traffic from one VPN tunnel to another. Traffic can be forwarded from either a gateway-to-gateway or a client-to-gateway tunnel.

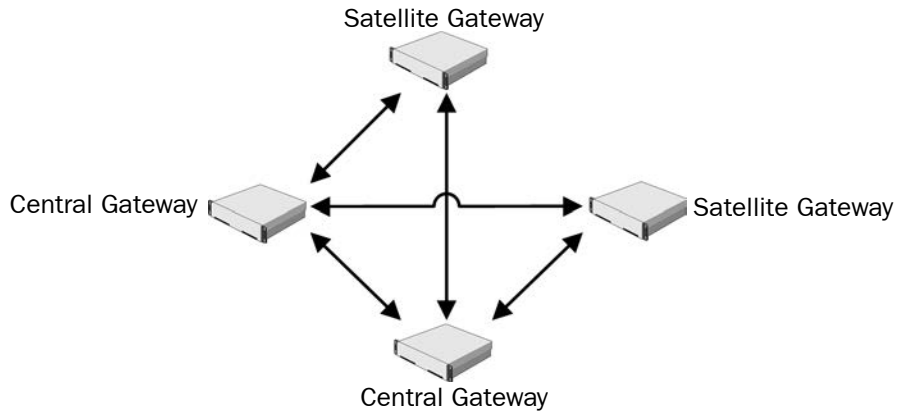
**Illustration 28.3 Hub VPN Topology**



The hub topology simplifies VPN client use if the clients connect to several gateways. It can also make setting up gateway-to-gateway VPNs easier, especially if the satellite gateways are 3rd party devices. VPN encryption/decryption requires heavy computing, so hardware performance should be considered before high volumes of traffic are concentrated at a hub gateway.

Different topologies are often combined in practice, since the connectivity needs usually vary from location to location. [Illustration 28.4](#) shows an example of a mixed topology.

**Illustration 28.4** Combination of Different Topologies



As seen here, replacing two of the central gateways from our full mesh example ([Illustration 28.1](#)) with satellite gateways actually results in a VPN where all but two gateways still have a VPN with each other.

# POLICY-BASED VPN CONFIGURATION

In policy-based VPNs, traffic is selected to be sent into the VPN tunnel according to the Access rules.

The following sections are included:

- ▶ [Overview to Policy-Based VPN Configuration](#) (page 270)
- ▶ [Configuration of Policy-Based VPNs](#) (page 271)
- ▶ [Using VPNs](#) (page 278)
- ▶ [Examples of Policy-Based VPN Configurations](#) (page 289)

# Overview to Policy-Based VPN Configuration

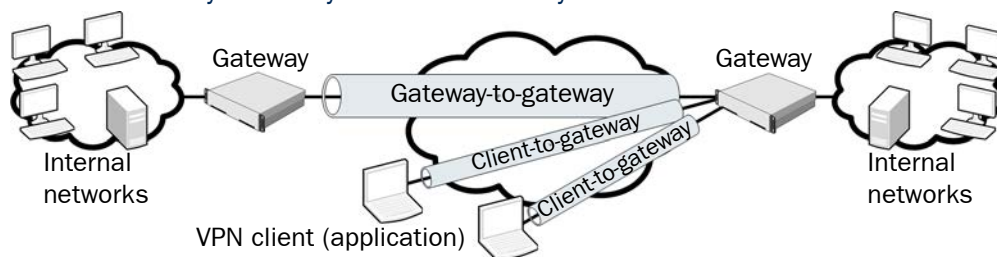
Policy-based VPNs are implemented according to the IPsec (Internet Protocol security) standards. This chapter assumes that you are already familiar with the basic concepts of building IPsec VPNs and concentrates on the features available in McAfee Firewall/VPN. Understanding basic IPsec concepts will greatly help you in configuring VPNs, so we recommend that you read the general overview to VPNs and IPsec before moving on to this section. See [Overview to VPNs](#) (page 261).

This chapter focuses on the configuration of policy-based VPNs using VPN elements. For information about the configuration of the Route-Based VPN, see [Route-Based VPN Configuration](#) (page 295).

You can create two main types of policy-based VPNs:

- *Gateway-to-gateway* VPNs are created between two or more gateway devices that provide VPN access to other hosts. Gateway-to-gateway VPNs are supported for IPv4 and IPv6 traffic.
- *Client-to-gateway* VPNs are created between a gateway device and individual hosts running VPN client software, such as laptops of travelling users or a desktop PC at a home office. Client-to-gateway VPNs are supported only for IPv4 traffic.

**Illustration 29.1** Gateway-to-Gateway and Client-to-Gateway VPNs



Because the policy-based VPN follows the IPsec standards, you can create VPNs with gateway devices and VPN clients from many different manufacturers. This allows you to create VPNs with partner organizations that have an IPsec VPN gateway. See [Configuring Policy-Based VPNs with External Gateway Devices](#) (page 286) for more information.

For client-to-gateway VPNs, the recommended option is to use the Stonesoft IPsec VPN Client, which is available for Windows platforms. You can alternatively use a third-party IPsec-compatible VPN client, but third-party clients do not support all features offered by McAfee Firewall/VPN. VPN clients cannot connect directly to firewalls that have a dynamic IP address. Instead, VPN clients can connect through a central gateway that is used as a hub that forwards the connections to the non-compatible gateways using a gateway-to-gateway VPN.

Policy-based VPNs are recommended for the following uses:

- To use Multi-Link with the VPN.
- To create client-to-gateway VPNs.
- To create VPNs in which some gateways act as central gateways and other gateways act as satellite gateways (for example, star topology and VPN hub topology).

You can also use a policy-based VPN to provide encryption for the Route-Based VPN in Tunnel Mode. See [Providing Encryption for the Route-Based VPN in Tunnel Mode](#) (page 289).

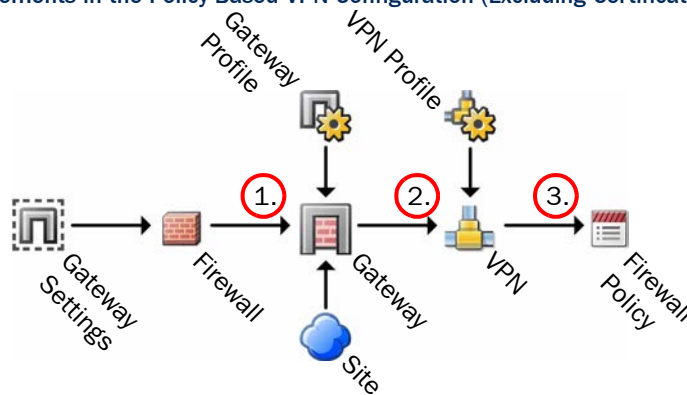
# Configuration of Policy-Based VPNs

Devices that provide VPN access to other computers are called *VPN gateways*. There are two general types of VPN gateways in McAfee Firewall/VPN:

- *Internal VPN gateways* are McAfee Firewall/VPN engines that are managed by the Management Server (and administrative Domain) you are currently connected to with your Management Client.
- All other gateway devices are *external VPN gateways*, including McAfee Firewall/VPN engines that are managed by some different Management Server (or administrative Domain) than the one you are currently connected to with your Management Client.

Due to the various authentication and encryption methods that are supported in IPsec VPNs, the number of settings in policy-based VPNs is rather high. To save you from repeated configuration work, reusable profiles are used for storing different types of settings. These and other elements related to the configuration of policy-based VPNs are pictured in the illustration below, except the elements that are related to managing certificates. Certificate authentication is discussed in [Using Certificate Authentication](#) (page 283).

**Illustration 29.2** Elements in the Policy-Based VPN Configuration (Excluding Certificate-Related Elements)



The three main points of policy-based VPN configuration are (as indicated by the numbers in the illustration above):

1. The Gateway element sets VPN-related settings particular to one Firewall/VPN device. Each Gateway element can be used in several VPNs. The Gateway element refers to the following other elements:
  - The Firewall element contains some VPN client-related settings. The Firewall element always refers to a Gateway Settings element that defines settings for advanced VPN performance tuning, which are optional to adjust.
  - Gateway Profile elements contain information about the capabilities of different types of gateways, so that the system can disable unsupported options and find incompatible combinations of settings automatically. Gateway Profiles can be created and selected for External VPN Gateways. The Gateway Profiles of Internal VPN Gateways are selected based on the installed software version.
  - Site elements define those (real or translated) IP addresses that are routable through the policy-based VPNs. The system can add the IP addresses automatically from routing or you can adjust the Sites yourself.

2. The VPN element combines other elements together to define the settings used in one particular policy-based VPN and defines the topology for the VPN. The VPN element refers to a VPN Profile, which contains the IPsec authentication and encryption settings (IKE settings) that are essential in configuring a VPN.
3. The Firewall Policy controls VPN traffic in the same way as any other traffic.
  - The Access Rules determine which connections are directed out through each VPN and which traffic is allowed in from each VPN.
  - The NAT Rules define what kind of address translation is done for VPN connections. The VPN communications between the gateway devices are always subject to NAT as usual, but the traffic that uses the tunnels is subject to NAT only if address translation is specifically enabled for the VPN.

With the exception of End-Point IP addresses, the same elements used in the configuration of policy-based VPNs can also be used in the configuration of the Route-Based VPN.

## Default Elements

There are several default elements for policy-based VPN configuration:

**Table 29.1 Default Elements for Policy-Based VPN Configuration**

Element Type	Default Elements
Gateways	There is a predefined IPsec Client gateway element that is used to represent VPN clients, including any Stonesoft and third-party VPN clients. Note that you can change the Gateway Profile associated with this default element.
Certificates	The Internal RSA CA for Gateways VPN Certificate Authority represents the Management Server's internal RSA Certificate Authority. You can use the element to define certificate trust relationships if you configure other CAs in the SMC.
Gateway Profiles	Several different Gateway Profiles are included for different Firewall/VPN and Stonesoft IPsec VPN Client versions. With third-party VPN devices, you can use the Default (All Capabilities) profile, which enables all options, or create a more restrictive profile yourself for better automatic configuration validation.
Gateway Settings	Gateway Default Settings is a predefined Gateway Settings element that contains the default recommended settings for most environments.
VPN Profiles	<p>The predefined VPN Profiles (VPN-A Suite, Suite-B-GCM-128, and Suite-B-GCM-256) are provided to allow you to quickly try out VPNs without creating a VPN Profile yourself.</p> <ul style="list-style-type: none"> <li>- The <i>VPN-A Suite</i> VPN Profile contains the VPN settings specified for the cryptographic suite "VPN-A" in RFC 4308.</li> <li>- The <i>Suite-B-GCM-128</i> VPN Profile contains the VPN settings specified for the cryptographic suite "Suite-B-GCM-128" in RFC 6379.</li> <li>- The <i>Suite-B-GCM-256</i> VPN Profile contains the VPN settings specified for the cryptographic suite "Suite-B-GCM-256" in RFC 6379.</li> </ul> <p>The predefined VPN Profiles also allow you to change settings that are not specified in RFC 4308 and RFC 6379, so you may need to adjust the settings to achieve a valid VPN in some configurations.</p>



# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define the Gateway Settings

Each firewall has settings that are common to all VPNs the firewall establishes, set in the Gateway Settings element. These settings are mostly for performance tuning, and in most cases, there is no need to change them at all. If there is some particular need to change the settings, you must create a new Gateway Setting element, since the Gateway Default Settings system element cannot be edited.

## Task 2: Define the Gateway Profile

The Gateway Profile introduces information about the features and options available so that the VPN configuration can be automatically validated. The general settings directly affect the settings used in VPNs. The authentication and encryption settings defined in the Gateway Profile do not directly influence which of the displayed settings are used for any VPNs; it is more of a configuration aid and a reference for checking that the settings defined for the VPNs correspond to the options supported by the gateway devices involved.

For Internal VPN Gateways, the correct Gateway Profiles are automatically selected according to the software version, and you cannot change the selection. If you use a McAfee Firewall/VPN engine managed by a different Management Server or administrative Domain as an External VPN Gateway, you must select the correct Gateway Profile according to the software version. If you use a third-party device as an External VPN Gateway, you can either use the Default (All Capabilities) profile (which allows any of the options to be selected for the Gateway) or define a new Gateway Profile yourself. For Stonesoft IPsec VPN Clients, there are predefined Gateway Profiles, but you can also create custom Gateway Profiles yourself.

## Task 3: Define the Gateways

The Internal and External VPN Gateway elements define settings for the internal and external gateway devices in their role as VPN gateways. 'Internal' and 'external' refer to whether the device is managed by the Management Server (and under the same administrative Domain) where the element is defined.

Usually, there is just one Gateway element per firewall, because you can use the same Gateway in several different VPNs, possibly overriding some of the Gateway's settings as necessary. It is possible to create several Gateway elements to represent a single Firewall/VPN engine, but each Gateway element reserves a VPN end-point (IP address) that other Gateway elements cannot use. If you use the same Gateway element in both policy-based VPNs and the Route-Based VPN, you must define unique end-points for each type of VPN.

The special predefined IPsec Client Gateway element represents all types of VPN clients in client-to-gateway VPNs. When you set up a client-to-gateway VPN with Stonesoft IPsec VPN Clients, the IPsec Client element must always be used. In most cases, we recommend using the element with third-party VPN clients as well. However, it is possible to configure an individual third-party VPN client using an External VPN Gateway element instead if there is a specific need to do so, although this allows only one client at a time to connect to each gateway.

## Task 4: Define the Sites

The protected IP addresses behind each gateway are defined using Site elements. In the IPsec standard, these IP addresses are called *traffic selectors*. The IP addresses work like routing definitions when the gateway selects which VPN tunnel a packet is sent through. The Sites must contain the IP addresses of all protected hosts that potentially send or receive VPN traffic through any gateway-to-gateway or client-to-gateway VPN. IP addresses that are not included in the Sites are not allowed as source or destination addresses in the policy-based VPN. You cannot add or modify Sites under the IPsec Client Gateway element. The Sites are always added globally for all policy-based VPNs where a Gateway is used, but unnecessary Sites can be disabled in individual VPNs.

With Internal VPN Gateway elements, you have the option to include a Site that is automatically populated and updated according to the definitions of the Routing view. All interfaces and networks are included in the automatic Site, except the external interfaces (those with the Any Network element). If loopback IP addresses are defined for the Internal VPN Gateway, you can use a loopback IP address as an End-Point IP address.

The Sites must always contain the actual IP addresses that are used inside the VPN tunnel. If you enable NAT for the policy-based VPN and translate the local IP addresses, you must define the Sites using the translated (after NAT) addresses. The NAT addresses are not added to the Site automatically.

If you want to use a central gateway as a hub so that it forwards traffic from one VPN tunnel to another, all IP addresses that are accessible through the central gateway must be included in the central gateway's Sites.



**Note – An IP address must be included in a Site to be valid in the VPN, but Access rules define which connections are actually allowed to enter and exit a VPN tunnel.**

## Task 5: Create Certificates

Certificates can be used for authenticating gateways and VPN clients. In gateway-to-gateway VPNs, you can use both pre-shared keys and certificates as the authentication method. In client-to-gateway VPNs, certificates are always needed when Stonesoft IPsec VPN Clients are involved. However, if you use the hybrid authentication method with Stonesoft IPsec VPN Clients, only the gateway needs a certificate.

Certificates do not contain information that is specific to a particular VPN, and can be used with both policy-based VPNs and the Route-Based VPN. The same certificate can be used for any number of VPNs with any number of gateways and VPN clients. For internal VPN gateways, the certificate handling can be completely automatic if the certificate is signed using the Management Server's internal default Certificate Authority. See [Using Certificate Authentication](#) (page 283) for more information.

## Task 6: Define the VPN Profile

The VPN Profile elements contain settings related to authentication, integrity checking, and encryption. This is the main point of configuration for IKE and IPsec settings (the settings used or agreed on during IKE SA and IPsec SA negotiations). You are generally free to choose any combination of settings as long as all gateways and VPN clients involved support those settings and are configured to accept them. See [Supported Authentication and Encryption Methods](#) (page 279) for more information.

The authentication methods for VPN clients are selected separately in the VPN Profile. A certificate-based method is always included in the VPN, but you can optionally add other authentication methods.

The same VPN Profile can be shared by several VPNs if the settings fit. You can use the same VPN Profile in both policy-based VPNs and the Route-Based VPN. You can also easily copy the element to create modified versions of the same basic settings.

## Task 7: Define the VPN Element

The VPN element collects together the Gateways and the VPN Profile and provides the settings for defining the topology and the tunnels of the policy-based VPN.

The topology is determined by selecting whether the Gateways are Central or Satellite in each particular VPN element:

- A Central Gateway establishes VPN tunnels with any other Central or Satellite Gateway in the VPN, unless you specifically disable the tunnels.
- A Satellite Gateway establishes VPN tunnels only with Central Gateways.
- You can also create a VPN hub by adding a gateway so that it is listed under some other (central or satellite) gateway in the topology (other Gateways connect to the higher-level gateway, which forwards the connections to the lower-level gateway).

The Sites and networks for each Gateway element can be adjusted in the VPN element, but for the most part, they are not VPN-specific. The only VPN-specific change is to disable some Site element in the VPN element, which excludes the IP addresses from that policy-based VPN only. Any other adjustments to the Sites and networks affect all other VPNs where the same Gateway element is used.

Tunnels are generated based on the overall topology (from each central gateway to all other gateways). You can further adjust the tunnels generated to limit which gateways and end-points form tunnels with each other, and change some of the properties for tunnels between two particular gateways or end-points, such as the VPN Profile used. You can also define Multi-Link VPN settings that allow you to select standby and active tunnels and to set tunnels to aggregate mode in which each connection is automatically balanced between the aggregate tunnels. Aggregate mode is likely to cause packet reordering that may actually decrease performance, depending on the TCP stacks in the connection end-points.



**Note** – Although the VPN end-points usually correspond to the NetLink interfaces in a Multi-Link configuration, the VPN end-point settings are separate from the NetLink and Multi-Link definitions. For example, a NetLink that is set to standby for cleartext traffic can still be used as an active end-point for VPN traffic.

Within the VPN element, you can also view warnings and errors regarding the configuration (indicated as Issues). Always check the Issues panel for any problems after you have edited the VPN.

## Task 8: Modify the Firewall Policy

No traffic is sent through the policy-based VPN until you direct traffic to the VPN in the IPv4 or IPv6 Access rules. The VPN must be referenced in at least one IPv4 Access rule or IPv6 Access rule for the VPN settings to be included in the firewall's configuration. The communications required to establish the VPN are allowed based on the VPN definitions and the rules in the Firewall Template, so you do not need to specifically include the gateway addresses in the Access rules (except possibly if you use your own customized top-level Template policy).

VPN Access rules behave basically the same as all other Access rules: you define certain matching criteria and all traffic that matches is then handled according to the Action set for the rule. The Use IPsec VPN rule action has three main options, which have different effects depending on the source and destination of the traffic:

**Table 29.2 Use IPsec VPN Action Options**

Option	Description
Apply VPN	Directs traffic from protected local networks into the VPN tunnel. Allows traffic that arrives through a VPN to proceed. The rule does not match non-VPN traffic from outside networks into the protected networks regardless of whether the other cells in the rule match; this allows handling special cases in which identical-looking VPN and cleartext traffic must be passed through the firewall.
Enforce VPN	Directs traffic from protected local networks into the VPN tunnel. Allows traffic that arrives through a VPN to proceed. The rule drops non-VPN connections from outside networks into the protected networks if the other cells in the rule match the connection.
Forward	Directs traffic from protected local networks or from a VPN tunnel into a VPN tunnel. Useful for forwarding connections from one VPN tunnel into another (VPN hub configuration) or connections from local networks to VPN client computers that are currently connected.

When traffic is sent out through a VPN, the correct VPN tunnel is selected based on the Sites of the Gateway elements. If a VPN Access rule matches a connection whose source or destination IP address is not included in the defined Sites, VPN tunnel selection fails and the connection is dropped.

Incoming connections that arrive through the VPN are matched just like connections that do not use a VPN. Incoming connections do not have to match a VPN Access rule to be allowed in through the VPN; any Access rule can match a VPN connection. Unintended matches can be avoided with the correct rule order, but as an additional tool, the Source VPN cell allows you to define rules that evaluate whether the incoming traffic is using a VPN.

NAT rules only apply to the encrypted packets (the VPN tunnel) by default. The addresses of the packets going through the VPN tunnel are translated if you specifically enable NAT for the VPN. With NAT, the traffic in the VPN tunnel uses the translated addresses, so you must define the Sites using the translated addresses.



**Note** – NAT is needed for the NAT Pool feature in VPN client communications and for the Server Pool feature in inbound traffic management. To use these features in a VPN, NAT must be specifically enabled in the VPN element.

## Task 9: Configure VPN Clients and External Gateway Devices

If you use the Stonesoft IPsec VPN Client, the policy-based VPN configuration you create in the Management Client is also used for creating the configuration for the VPN clients. The VPN clients then download the settings from the VPN gateway(s) either manually or automatically whenever there are relevant changes. All necessary IPsec and address management settings are included in the download (for example, information on which encryption methods are used and which internal networks clients can access through the gateway). The decision if a VPN tunnel is used is based on the IP addresses you have defined for the Sites of the Gateway element.

For third-party VPN clients and any external VPN gateways, you must duplicate the settings you defined for the Internal VPN Gateway in the configuration of the VPN client or gateway in question (including engines under a different administrative Domains). This includes all IPsec-related settings, such as the authentication, encryption, and integrity checking options as well as the encryption domain (the IP addresses that are allowed in the VPN as a source or destination IP address). For more information, see [Configuring Policy-Based VPNs with External Gateway Devices](#) (page 286).

# Using VPNs

---

This section covers the following topics:

- [VPN Logging](#).
- [Using a Dynamic IP Address for a VPN End-Point](#).
- [Using a NAT Address for a VPN End-Point](#) (page 279).
- [Supported Authentication and Encryption Methods](#) (page 279).
- [Using Pre-Shared Key Authentication](#) (page 283).
- [Using Certificate Authentication](#) (page 283).
- [Configuring Policy-Based VPNs with External Gateway Devices](#) (page 286).
- [Clustering and Policy-Based VPNs](#) (page 287).
- [Multi-Link and Policy-Based VPNs](#) (page 287).
- [Providing Encryption for the Route-Based VPN in Tunnel Mode](#) (page 289)

## VPN Logging

The VPN negotiations and new connections are logged as informational messages and can be viewed in the Logs view like any other logs. New connections that are allowed through the policy-based VPN (the actual traffic using the VPN) are logged just like any other traffic according to the logging options you set in Access rules.

If there are VPN-related problems, you can activate diagnostic logs for IPsec for the firewall in question to get more detailed information on the VPN negotiations. The Troubleshooting section in the Management Client *Online Help* and the *McAfee SMC Administrator's Guide* contains further information on problems you may encounter, including explanations for the most common messages you may see in the logs.

## Using a Dynamic IP Address for a VPN End-Point

If a VPN end-point has a Dynamic (assigned using DHCP, PPPoA, or PPPoE) IP address, some restrictions apply:

- Each VPN tunnel can have a dynamic IP address at only one end. The VPN can be established only one way: from the end-point with a dynamic IP address to the end-point with a static IP address. If there is not frequent enough traffic from the dynamic IP gateway to establish and keep up the VPN tunnel at all times, you can send some traffic through the VPN from the site with the dynamic IP gateway to ensure the tunnel is up and ready for traffic coming from the static IP gateway.
- It is not possible to configure a client-to-gateway VPN to end-points that have a dynamic IP address. To work around this, clients can connect to a gateway with a static IP address, which can then be configured to forward the traffic through a gateway-to-gateway VPN to the gateway with a dynamic IP address.
- When a dynamic IP address is used, the VPN gateway must be set up to use some other identifier than the IP address: either DNS name, e-mail address, or (if certificate authentication is used) the certificate's Distinguished Name (DN).
- IKEv1 main mode with pre-shared key authentication is not supported in VPNs involving dynamic IP gateways. Aggressive mode allows the use of pre-shared keys, but for security reasons certificate-based authentication is also recommended when IKEv1 is set in aggressive mode.

## Using a NAT Address for a VPN End-Point

If a gateway does not have a public IP address as a VPN end-point, you may have to configure NAT traversal and contact addresses for the VPN traffic.

VPN traffic is specifically protected against modifications, so specific provisions have to be taken when NAT is applied to the encrypted IPsec traffic. UDP encapsulation can be used to wrap the encrypted packets inside UDP packets to allow NAT modifications but prevent the encapsulated VPN traffic from being modified by NAT. Stonesoft IPsec VPN Clients also support TCP tunneling, which works similarly, but allows also forwarding the traffic through any port you define to allow the traffic to pass a filtering device that is not under your control.

You may also need to configure the policy-based VPN with contact addresses so that the gateways are aware of the NAT operation:

- Any firewall that is used as an Internal VPN Gateway in a NAT environment must have the Location and Contact Address definitions correctly defined for the end-point interface(s) involved (CVIs in clusters). If Contact Addresses have already been configured for non-VPN use, the same general configuration applies to VPN communications as well. Stonesoft IPsec VPN Clients download their configuration from the gateway (firewall), including any contact address configuration as needed.
- In most cases, any External VPN Gateway must be defined using its private IP address and the public IP address must be added as the Contact Address for the contacting McAfee Firewall's Location.

## Supported Authentication and Encryption Methods

Your company's security policy will typically contain guidelines for selecting authentication and encryption methods. The main consideration for selecting the authentication and encryption settings in your policy-based VPN configuration is that you fulfill the requirements of the security policy and other security requirements that concern your organization.

The tables in this section list the different message digest algorithms (for integrity checking), authentication methods, and encryption methods that are available in policy-based VPNs. The IPsec standards mandate support for some options, but also allow additional options to be provided by IPsec-compatible products. RFC 4305 lists the IPsec standard requirements that all IPsec-compliant products must follow.

The tables below contain estimates of how common support for the various algorithms is in IPsec-compatible products. This information may be of some use when deciding which methods to use when establishing a VPN with a third-party VPN device, even though no decisions can be made without referring to documentation that details the actual capabilities of the device in question.

### FIPS Mode

If your organization is required to follow FIPS encryption standards, some of the options presented in the tables that follow are not available in your system. See the *Common Criteria Certification User's Guide* for more information.

## GOST-Compliant Systems

If your organization is required to follow GOST encryption standards, the options in your system are mostly different from those presented in the tables that follow.

## Message Digest Algorithms

Message digest algorithms ([Table 29.3](#)) are used to ensure integrity of data (that the packets have not been changed in transit). These algorithms are often also referred to using the MAC or HMAC abbreviations (keyed-hash message authentication code).

**Table 29.3 Supported Message Digest Algorithms\***

Algorithm	Description
AES-XCBC-MAC	128-bit hash algorithm. Available only for checking the integrity of IPsec traffic. Many IPsec-compatible VPN devices do not support this algorithm, but support is becoming increasingly common. Reference: RFC 3566.
MD5	Message-Digest algorithm 5, a 128-bit hash algorithm (also referred to as HMAC-MD5). Available for checking the integrity of the IKE negotiations and IPsec traffic. Most IPsec-compliant VPN devices still support this algorithm, but support may become less common in the future. Reference: RFC 2403.
SHA-1	Secure Hash Algorithm with a 160-bit hash (sometimes referred to as HMAC-SHA-1). Available for checking the integrity of the IKE negotiations and IPsec traffic. All VPN devices must support this algorithm to be fully IPsec-compliant. Reference: RFC 2404.
SHA-2	Secure Hash Algorithm with 256-bit, 384-bit, and 512-bit hashes (it includes SHA-224, SHA-256, SHA-384, and SHA-512). Available for checking the integrity of the IKE negotiations and IPsec traffic. Most IPsec-compliant VPN devices support this method. Reference: RFC 4868.

\*The Russian product version has no strong encryption algorithms.



## Authentication Methods

Authentication ([Table 29.4](#)) ensures that the remote party is who they claim they are (for example, to prevent a man-in-the-middle attack).

**Table 29.4 Supported Authentication Methods**

Method	Description
Pre-Shared Key	<p>Also referred to with the acronym PSK. Authentication is done using a string of characters that is entered into the configuration of each VPN device.</p> <p>Security depends greatly on the complexity and length of the string of characters used, which is why an automatically generated, long string is strongly recommended. You can automatically generate the key in the Management Client or using some external tool. The key must also be periodically changed. Do not use a normal password, as that will make your VPN susceptible to brute-force attacks.</p> <p>It is more secure to use the main mode (instead of the aggressive mode) for IKE negotiations if you use pre-shared key authentication due to the pre-shared keys' susceptibility to brute-forced attacks in aggressive mode.</p> <p>All VPN devices must support this method to be fully IPsec-compliant.</p>
RSA Signatures	<p>Authentication is done using certificates and an RSA digital signature.</p> <p>Most IPsec-compliant VPN devices support this method.</p>
DSS Signatures	<p>Authentication is done using certificates and a DSA (Digital Signature Algorithm) digital signature (generated according to the DSS, Digital Signature Standard).</p> <p>Many IPsec-compatible VPN devices do not support this method.</p>
ECDSA Signatures	<p>Authentication is done using certificates and an ECDSA (Elliptic Curve Digital Signature Algorithm) digital signature (generated according to the DSS, Digital Signature Standard).</p> <p>Many older IPsec-compatible VPN devices do not support this method.</p>

## Encryption Algorithms

Encryption algorithms ([Table 29.5](#)) scramble the data so that it is not viewable while in transit.

**Table 29.5 Supported Encryption Methods\***

Method	Description
AES-128	<p>Advanced Encryption Standard (also referred to as Rijndael) with a 128-bit/192-bit/256-bit encryption key. The AES-128 option uses 128-bit keys by default, but accepts stronger 192-bit or 256-bit keys if requested by the other gateway.</p> <p>Most IPsec-compliant VPN devices support one or both key lengths of these methods, and support is becoming more common.</p> <p>Reference: RFC 4309.</p>
AES-256	

**Table 29.5 Supported Encryption Methods\* (Continued)**

Method	Description
AES-GCM-128	<p>Advanced Encryption Standard (also referred to as Rijndael) in GCM (galois/counter mode), uses a 16-octet ICV (integrity check value). AES-GCM-128 uses a 128-bit encryption key, and AES-GCM-256 uses a 256 bit encryption key. Provides both authentication and encryption. Note that these methods replace the selected message digest algorithm. In high-performance networks, these are the recommended encryption methods.</p> <p>Many IPsec compatible VPN devices do not support one or both key lengths of these methods, but support is becoming more common.</p> <p>Reference: RFC 4106.</p>
AES-GCM-256	
DES	<p>Data Encryption Standard (also referred to as the Data Encryption Algorithm, DEA), uses a 56-bit encryption key.</p> <p>Do not use DES if you can avoid it. DES has been largely abandoned because the short key makes it vulnerable to attacks. If you must use DES, make sure that PFS is enabled and that the encryption keys are frequently renegotiated.</p> <p>Many IPsec-compliant VPN devices still support this method, but support is becoming less common.</p> <p>Reference: RFC 2405.</p>
Blowfish	<p>Uses up to 448-bit keys. Policy-based VPNs use 128-bit keys by default, but accept up to 448-bit keys if requested by the other gateway.</p> <p>Many IPsec-compatible VPN devices do not support this method.</p> <p>Reference: RFC 2451.</p>
3DES	<p>Triple-DES (also referred to as TDES or TDEA, Triple Data Encryption Algorithm), uses 168-bit encryption achieved with three different 56-bit encryption keys.</p> <p>3DES is quite processor-heavy considering the level of protection it offers and is therefore not the most efficient method. It may not be optimal for VPNs that handle large traffic volumes or systems that otherwise have a high load.</p> <p>Most IPsec-compliant VPN devices support this method.</p> <p>Reference: RFC 2451.</p>
Null	<p>No encryption. Traffic is sent as cleartext just like any non-VPN traffic and can be viewed by anyone who intercepts it in transit.</p> <p>Null encryption is useful only in special cases. Do not select it for any VPN that requires protected data transfer.</p> <p>Most IPsec-compliant VPN devices support this method.</p> <p>Reference: RFC 2410.</p>

\*The Russian product version has no strong encryption algorithms.

## Using Pre-Shared Key Authentication

Pre-shared keys can be used for gateway-to-gateway VPN authentication and with third-party VPN clients. A pre-shared key is a string of characters that is used as an authentication key.

Both gateways create a hash value based on the pre-shared key and other information. The hash values are then exchanged and verified to authenticate the other party. As its name suggests, the pre-shared key has to be distributed beforehand to all devices that need to use it. Pre-shared keys must be transferred confidentially, since their security benefit is immediately lost if the key is exposed to unauthorized parties.

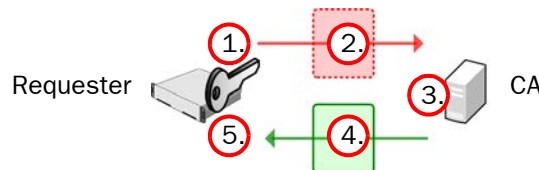
The pre-shared keys must also be long and random to be secure. Short or predictable pre-shared keys can be easily broken in brute-force attacks. Administrators must also remember to renew the pre-shared keys periodically to maintain a high level of security. McAfee Firewall/VPN includes tools for generating sufficiently long, random pre-shared keys for VPN components. The keys are automatically transferred to any internal VPN gateway devices that need them using the secure system communications channel.

## Using Certificate Authentication

Certificates can be used for authentication in any VPN, including the Route-Based VPN. In all gateway-to-gateway VPNs and in client-to-gateway VPNs with third-party VPN clients, you can select whether to use certificates or a pre-shared key for authentication. With Stonesoft IPsec VPN Clients, the authentication is always either a hybrid authentication that requires the presence of a valid certificate on the gateway and some other form of authentication from the VPN client user, or certificate exchange authentication that requires a certificate from both the gateway and the VPN client.

Certificates often provide better real-world security than pre-shared keys. Certificates only have to be renewed at an interval of a few years, and have an automatic expiration mechanism that makes sure the certificate is renewed. Certificate files cannot be compromised in transit, since they cannot be used without a private encryption key. The illustration below outlines the basics of how a certificate is generated.

**Illustration 29.3** VPN Certificate Creation



The following steps are completed when certificates are generated:

1. When a certificate request process is started, a private encryption key is generated and stored.
2. The certificate requester uses the private key to generate an encrypted certificate request that is transferred to the certificate authority (CA).
3. The CA signs the encrypted certificate request, which validates the certificate.
4. The signed certificate is transferred to the original certificate requester.
5. The requester uses its stored private key to access the certificate.

The certificate creation is either automatic or manual:

- For internal VPN gateways, all steps can be completely automatic if the internal certificate authority is used for signing the certificate. If some other certificate authority is used, the certificate request is exported from the SMC and the signed certificate is imported back into the SMC as a file.
- For VPN clients, the certificate request file is created manually in the VPN client and transferred manually to be signed by the internal or some other certificate authority. The signed certificate is then transferred manually into the VPN client computer.

Private keys are always generated automatically. If the private key is lost (for example, due to a hardware failure), any associated certificate becomes unusable and a new certificate has to be created. The private key is securely and automatically synchronized between clustered firewall nodes to allow all nodes to use the same certificate.

Unlike pre-shared keys, certificates do not need to be distributed to all gateways in the VPN. Instead, the other gateways are configured to trust the certificate issuer (the CA that signed the certificate), after which they trust all certificates from that issuer. This also allows renewing or recreating the certificate on one gateway without having to reconfigure the other gateways. Since only certificates from trusted sources are accepted in authentication, VPN gateways involved in the VPN must be specifically configured to trust the certificate authorities that have signed the certificates that the other gateways use for authentication.

## Validity of Certificates

Certificates are always created to be valid starting from a specific date and time and to expire at a certain date and time in the future. All components that use (or sign) certificates must have the correct time settings to avoid unexpected certificate rejections. The Internal RSA CA for Gateways and the Internal ECDSA CA for Gateways of the Management Server generate certificates that are valid starting immediately until three years from their creation.

Certificate revocation lists (CRL) can be used to cancel a certificate before it reaches its expiration, for example, if there is reason to suspect that unauthorized parties have obtained a copy of both the certificate and the associated private key. The Internal RSA CA for Gateways and the Internal ECDSA CA for Gateways do not support certificate revocation lists. If you want to use CRLs, you must use an external certificate authority (either one you maintain yourself or a commercial service). The CRL servers are accessed using LDAP or HTTP (depending on what the certificate specifies). If all defined CRL servers are unreachable, the certificates are considered invalid until the CRL can be checked. You can set up the Firewall/VPN engine to access CRL servers directly or use the OCSP protocol.

## Internal VPN Certificate Authorities

The Management Server includes a dedicated Internal RSA CA for Gateways and optionally an Internal ECDSA CA for Gateways for signing VPN certificates for creating self-signed certificates. You can use both an Internal ECDSA CA for Gateways and an Internal RSA CA for Gateways at the same time.

The internal Certificate Authorities run on the same computer as the Management Server. If you have both an Internal RSA CA for Gateways and an Internal ECDSA CA for Gateways, only one Internal CA for Gateways can be selected as the default Certificate Authority. Only the default CA is used in automated RSA certificate management. You must manually create and renew any certificates that are signed not signed by the default CA.

If you want to use the Internal RSA CA for Gateways or the Internal ECDSA CA for Gateways to sign certificates for VPN clients or external gateway devices, certificate requests and signed certificates must be exported, transferred, and imported manually. The Internal RSA CA for Gateways and the Internal ECDSA CA for Gateways do not support certificate revocation lists, so we do not recommend using them to sign certificates for components that are outside the control of your organization.

The Internal RSA CA for Gateways and the Internal ECDSA CA for Gateways are each valid for ten years. A new Internal RSA CA for Gateways or Internal ECDSA CA for Gateways is automatically created to replace the default Certificate Authority six months before the expiration date. The Internal CA for Gateways that is not selected as the default Certificate Authority is not automatically renewed.

If automatic RSA certificate management is activated for an internal VPN gateway, RSA certificates issued by the default Certificate Authority are renewed automatically as long as the certificate-related files are intact (including the private key stored on the engines). You must manually create and renew any certificates that are signed not signed by the default Certificate Authority. If certificates are used to authenticate VPN client users and the certificates have been signed by the expiring Internal CA for Gateways, you must manually create new certificates for the VPN clients. You must also create new certificates manually for any other external components that have certificates signed by the Internal RSA CA for Gateways or the Internal ECDSA CA for Gateways.

## External Certificate Authorities

External certificate authorities can create certificates for Internal or External VPN Gateways or VPN clients. All IPsec certificates follow the ITU-T X.509 standard (also used in TLS/SSL, HTTPS, etc.). External certificate authorities are especially useful when creating VPNs with partner organizations, since this allows both organizations to use their preferred certificate authority. Different gateways in a VPN can have certificates signed by different certificate authorities.

To make McAfee Firewall/VPN engines accept externally signed certificates of external components, you simply import the public key of the external certificate authority into the SMC.

To use an external certificate authority to create a certificate for McAfee Firewall/VPN engines or Stonesoft IPsec VPN Clients, you must generate a certificate request and have it signed by the certificate authority. The external certificate authority must support PKCS#10 certificate requests in PEM format and the signed certificates must also be in the PEM format.

Furthermore, the certificate authority must be able to copy all attributes from the certificate request into the certificate. Especially, the X.509 extension Subject Alternative Name must be copied into the certificate because its value is used for identification.

## Configuring Policy-Based VPNs with External Gateway Devices

In policy-based VPNs, the term ‘external’ refers to any VPN gateway that is not controlled by the same Management Server (and the same administrative Domain) on which you are configuring the gateway element. Often, external gateway devices are at a partner organization, not under your control, and not McAfee Firewall/VPN devices. Because IPsec is a networking standard, you can create a VPN between gateways of different brands by selecting the desired settings identically for both gateways. Any option that both gateways support is a valid option for the VPN.

The settings that must match are:

- The IKE SA settings.
- The IPsec SA settings.
- The Site definitions (IP addresses) defined for both gateways at both ends (possibly translated using NAT).
- The end-point identity type and value (this is often the IP address of each gateway, but other options are also possible).

When the listed settings are identical, the VPN works. Unfortunately, there are some practical problems that complicate matching the settings.

The first problem you may experience when you configure a VPN between different brands of gateways is how to agree on common settings. Each and every setting must match to produce a fully functional VPN, and the supported options may be partly different on the different gateways. The authentication and encryption methods that policy-based VPNs support are listed in [Supported Authentication and Encryption Methods](#) (page 279). A related problem is that there is no one common standard for naming the different options, so the two gateways may actually use a different name for the same authentication or encryption method. In the case of McAfee Firewall/VPN devices that are used as External Gateways, you can export and import some settings between the two Management Servers (or between administrative Domains), but many of the configurations must still be manually constructed.

The IP addresses accessible through each Gateway are a commonly mismatched setting. In Internal Gateways, the IP addresses included in the policy-based VPN are defined as separate Site elements. An associated setting is the SA (security association) granularity setting, which defines whether a new VPN tunnel is established per each communicating host or per each network. In most gateways there is a specific option for this setting, but some gateways may select it implicitly based on the type of IP address definition or even have a fixed setting.



**Note – Site definitions are always defined for the Gateway element and are therefore used in all policy-based VPNs where the same Gateway is used. If you add a new Site for the Gateway in one policy-based VPN, disable it in other policy-based VPNs where you do not want the Site to be included.**

A good resource for checking whether a VPN gateway is interoperable with other IPsec gateways is the IPsec consortium web site at [www.vpnc.org/](http://www.vpnc.org/). The IPsec consortium has also come up with two example scenarios (called *Interoperability Profiles*) for which different manufacturers have provided set-up instructions, allowing you to see how the same settings are configured in different products. McAfee has provided a profile for the first scenario (available at the IPsec Consortium web site).

## Clustering and Policy-Based VPNs

A McAfee Firewall/VPN cluster can be used for policy-based VPNs, and there are no additional configuration steps compared to a single firewall. Clustering provides additional high availability and load balancing at the VPN gateway with multiple nodes in a cluster. In case one of the nodes is put offline or fails, the remaining nodes in the cluster take over the VPN traffic that was handled by that node. To allow the nodes to use the same certificate, the associated private encryption key is exchanged securely through the heartbeat channel. To external VPN gateways, the cluster presents itself like a single device with a single end-point (CVI IP address) to contact.

## Multi-Link and Policy-Based VPNs

Using Multi-Link enhances the reliability of the VPN communications by offering network connection high availability. McAfee Firewall/VPN can balance the traffic load between multiple network links and fail over when a link goes down. This reduces the possibility of link congestion or ISP network connectivity breaks. Multi-Link is not a part of the IPsec standards.



**Note – Multi-Link is only with McAfee gateways at both ends. If an external gateway device allows configuring multiple VPN tunnels between two devices, you may still be able to enjoy some of the benefits Multi-Link offers, but all Multi-Link features will not be available.**

In a Multi-Link configuration, the traffic can use one or several alternative tunnels to reach the same destination. This ensures that even if one or more tunnels fail, the VPN service continues as long as there is some tunnel available.

Multi-Link can be used between two McAfee gateways when one or both gateways use multiple network connections. VPN traffic is balanced between the tunnels based on the link availability checks on each VPN tunnel. If one of the links fails or becomes congested, the VPN traffic is routed through the other tunnels.

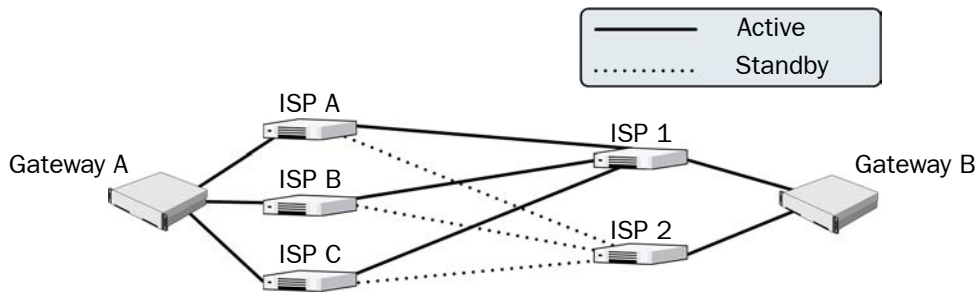
The VPN links can be in three different modes: active, aggregate, or standby. If there are multiple links in active mode, traffic is dynamically balanced across the different links based on a performance measurement or based on the links' relative bandwidths. In active mode, a single connection uses one of the active links. With multiple connections, all links are used. If there are multiple links in aggregate mode, each connection is balanced on a packet-by-packet basis between all the aggregate links in round robin fashion. Standby tunnels are used only if all active or aggregate tunnels become unavailable. Individual tunnels can also be completely disabled so that they are not used in the VPN under any conditions.



**Note – Aggregate mode in a Multi-Link VPN is likely to cause packet reordering due to different latencies of different links. This may decrease performance if the TCP stacks of the connection end-points do not handle reordering well. Use Active mode instead.**

Illustration 29.4 shows a Multi-Link VPN between two Gateways that both have multiple Internet connections. In this configuration, ISP 2 at Gateway B acts as a backup link for VPN traffic. The three tunnels (one from each ISP at Gateway A) with their end-points in the ISP 2 network have been set to standby, so that they are only used if ISP 1 fails. The standby setting is not tied to a particular ISP (NetLink), so it would be possible to set, for example, just the ISP A to ISP 2 tunnel back to active use while leaving the other tunnels in standby mode.

**Illustration 29.4 Example of a Multi-Link VPN with Standby Tunnels**



Stonesoft IPsec VPN Clients can also use Multi-Link. If one of the gateway's links fails, the client automatically connects to the next available NetLink.



## Providing Encryption for the Route-Based VPN in Tunnel Mode

When you use the Route-Based VPN in Tunnel Mode, the encapsulated tunnel is defined in the Route-Based VPN, and the encrypted tunnel is defined in a policy-based VPN. See [Using the Route-Based VPN in Tunnel Mode](#) (page 301) for information about creating the encapsulated Tunnel Mode tunnel(s) in the Route-Based VPN.

To configure a policy-based VPN to provide encryption for the Route-Based VPN in Tunnel Mode, you define a Host element to represent the End-Point IP address you use in the Route-Based VPN, and add the Host element to the Sites of an Internal VPN Gateway element. You use the Internal VPN Gateway in a policy-based VPN with a VPN profile that defines the encryption settings.

The Access rules that direct the Route-Based VPN traffic into the policy-based VPN are automatically generated for the Firewalls associated with the Internal VPN Gateway elements. The rules are not visible in the Firewall policy, and cannot be edited. If a policy that contains the automatically-generated rules for the VPN is installed on a Firewall that is not involved in the VPN, the rules are not included in the configuration that is transferred to the engine.

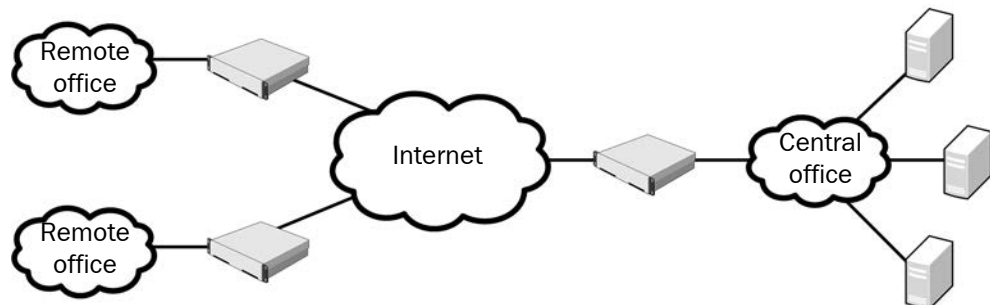
## Examples of Policy-Based VPN Configurations

The examples in this section illustrate some common uses for policy-based VPNs and general steps on how each scenario is configured.

### Creating a Policy-Based VPN Between Three Offices

Company A has a central office and two remote offices, each with their own McAfee Firewall/VPN device. The company needs secured communications links between the remote offices and the central office to allow access to various services, such as file servers, located at the central office.

**Illustration 29.5** Company A's Networks



There is no need for secure connectivity between the remote offices, since all shared servers are at the central office, and internal e-mails and other communications are also handled by the central servers.

All Firewall/VPN engines have a public IP address towards the Internet. The internal networks at each site use private IP addresses, but there is no need to translate the VPN traffic, since all offices use their own distinct address space.

The security policy of the company requires certificate-based authentication. The administrators decide to use the Management Server's Internal RSA CA for Gateways for issuing the VPN certificates.

The administrators:

1. Create Internal VPN Gateway elements for all three Firewall/VPN engines with each engine's public IP address as the VPN end-point and with automatic certificate management activated.
2. Add Site elements for all Gateways, and add the entire local internal network as the content for each Site.
3. Create a new VPN Profile and select RSA Encryption as the authentication method. RSA certificates are automatically generated for each Gateway.
4. Create a VPN element "Inter-Office VPN" that includes the central office Gateway as the Central Gateway and the two remote site Gateways as Satellite Gateways.
5. Add the following types of Access rules in the policy of the central office firewall:

Source	Destination	Action
Network elements for remote office 1 and remote office 2 internal IP addresses	Network elements for central office's internal networks	Use IPsec VPN - Enforce VPN "Inter-office VPN"
Network elements for central office's internal networks	Network elements for remote office 1 and remote office 2 internal IP addresses	Use IPsec VPN - Enforce VPN "Inter-office VPN"

6. Add the following types of Access rules in the policies of both remote office firewalls:

Source	Destination	Action
Network element for each remote office's internal IP addresses	Network elements for central office's internal networks	Use IPsec VPN - Enforce VPN "Inter-office VPN"
Network elements for central office's internal networks	Network element for each remote office's internal IP addresses	Use IPsec VPN - Enforce VPN "Inter-office VPN"

## Creating a Policy-Based VPN for Mobile Users

Company A from the example above has service technicians and salespeople at all offices who must be able to connect to their office network to access information when they are on customer visits. The administrators need to add VPN client access to the existing VPN infrastructure. The administrators decide to use the Stonesoft IPsec VPN Client, since all their users are running a compatible Windows operating system. As the authentication method, the administrators decide to use passwords stored in the Management Server's internal database.

The administrators also want to provide the service technicians and salespeople only one point of access so that they do not have to choose which gateway to connect to. That one point is the central office, which has gateway-to-gateway VPN tunnels to both remote offices that can be used for forwarding traffic to those sites as needed. The existing DHCP server at the central office can be used for assigning IP addresses to the VPN clients' Virtual Adapter, which is required for this kind of forwarding.

The administrators:

1. Edit the central office Internal VPN Gateway element and activate the Virtual Adapter method for VPN client address management.
2. Edit the VPN Profile to use Hybrid Authentication for authenticating the IPsec VPN Client users.
3. Create a VPN element "Remote User VPN" that includes the central office Gateway as the Central Gateway and the default Client Gateway element as a Satellite Gateway.
4. Create a new "Forward Addresses" Site element under the central office gateway and populate the site with the remote office networks to make those IP addresses route through the "Remote User VPN".
5. Disable the "Forward Addresses" Site in the existing "Inter-Office VPN" between the central office and the remote offices. Sites are global for all policy-based VPNs, so this Site must be specifically disabled to avoid a misconfiguration in the Inter-Office VPN.
6. Create User Group and User elements to define the user names and passwords for the IPsec VPN Client users.
7. Add the following Access rules in the policy of the central office firewall:

Source	Destination	Action	Users	Authentication	Source VPN
ANY	Central office internal networks	Use IPsec VPN - Enforce VPN "Remote User VPN"	"VPN Client Users" User Group	"User Password" Authentication Service	
VPN Client DHCP addresses	Remote offices' internal IP addresses	Use IPsec VPN - Forward VPN "Inter-Office VPN"			Match Any Mobile User VPN

8. Create a customized installation package of the Stonesoft IPsec VPN Client, so that the users can install using a silent installation package that does not require their input. The administrators include the Gateway contact information in the package so that the users do not need to enter it manually even when they use the Stonesoft IPsec VPN Client for the first time.

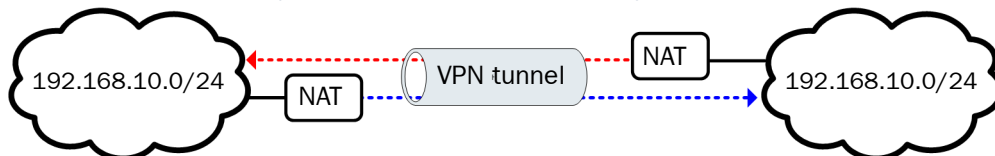
## Creating a Policy-Based VPN That Requires NAT

Company B has just decided to partner with Company C for a large project. Since the companies will need to exchange sensitive information, they decide to establish a VPN.

The external gateway device is behind a NAT device that translates between its internal and external IP address. This needs to be taken into consideration, since both addresses are needed in the policy-based VPN configuration.

Additionally, both companies use the same address space internally, so they must apply NAT for all connections through the policy-based VPN as well.

**Illustration 29.6 NAT for a Policy-Based VPN Between Two Gateways**



NAT is to be applied at both companies before traffic enters the VPN from each end. This way, routing problems caused by the same address space appearing in two different networks can be avoided, since traffic that is meant to be routed into the VPN uses unique translated addresses.

The administrators:

1. Create an Internal VPN Gateway element for their own Firewall/VPN engine with the engine's public IP address as the VPN end-point.
2. Create a new Location element and select it for their Firewall element.
3. Create an External VPN Gateway element "Partner Gateway" for the partner's VPN device using the internal IP address as the VPN end-point, and adding the external (translated) IP address as the Contact Address for the Location created in the previous step.
4. Create a Network element "HQ NAT Address Space" for the addresses that Company B plans to use for translating their internal IP addresses. They make sure these addresses are routable and unique in Company C's internal network.
5. Add only the Network created in the previous step in the Site for the internal gateway.
6. Create a Network element "Partner Network" for the addresses that Company C plans to use for translating their internal IP addresses. They make sure these addresses are routable and unique in Company B's internal network.
7. Add the "Partner Network" as the only network in the Partner Gateway's Site.
8. Create a new VPN Profile and select all settings so that they match those agreed with Company C.
9. Create a VPN element "Partner VPN" that includes the Internal VPN Gateway as the Central Gateway and the Partner Gateway as a Satellite Gateway.

10.Add the following types of Access rules in the policy of their firewall:

Source	Destination	Action
Network element “Partner Network”	Network element “HQ NAT Address Space”	Use IPsec VPN - Enforce VPN “Partner VPN”
Company B’s internal network (real IP addresses)	Network element “Partner Network”	Use IPsec VPN - Enforce VPN “Partner VPN”

11.Add the following types of NAT rules in the same policy:

Source	Destination	NAT
Company B’s internal network (real IP addresses)	Network element “Partner Network”	Static source translation to “HQ NAT Address Space”

- To make the static address translation work, the administrators take care that the translated address space is as large as the original address space.



# ROUTE-BASED VPN CONFIGURATION

A Route-Based VPN creates VPN tunnels between firewall interfaces that are designated as tunnel end-points. Any traffic that is routed to the specified interfaces is sent into the Route-Based VPN.

The following sections are included:

- ▶ [Overview to Route-Based VPN Configuration](#) (page 296)
- ▶ [Configuration of the Route-Based VPN](#) (page 296)
- ▶ [Using the Route-Based VPN](#) (page 301)
- ▶ [Examples of Route-Based VPN Configurations](#) (page 302)

## Overview to Route-Based VPN Configuration

---

The Route-Based VPN is implemented according to the IPsec (Internet Protocol security) standards. This chapter assumes that you are already familiar with the basic concepts of building IPsec VPNs and concentrates on the features available in McAfee Firewall/VPN. Understanding basic IPsec concepts will greatly help you in configuring VPNs, so we recommend that you read the general overview to VPNs and IPsec before moving on to this section. See [Overview to VPNs](#) (page 261).

The Route-Based VPN uses IPsec in *transport mode* between tunnel end-points. IPsec transport mode does not encapsulate the packets into new IPsec packets. Instead, the original headers of the packet are left intact, and the IP payload of the packet is encrypted. Before using IPsec transport mode to encrypt the packets, other encapsulation, such as *Generic Routing Encapsulation* (GRE) or *IP in IP* (IP-IP), must be used to add the tunnel end-point IP addresses in front of the original packet header. *Tunnel Interfaces* represent Route-Based VPN tunnel end-points on the Firewall/VPN engine. This allows routing information to be used to determine the correct VPN tunnel to use.

The Route-Based VPN is recommended for the following uses:

- To use VPN tunnels as paths in dynamic routing.
- To protect the integrity of dynamic routing communications that are sent through the Internet.
- To protect and route Multicast streams through the Internet.

## Configuration of the Route-Based VPN

---

Devices that provide VPN access to other computers are called *VPN gateways*. There are two general types of VPN gateways in the SMC:

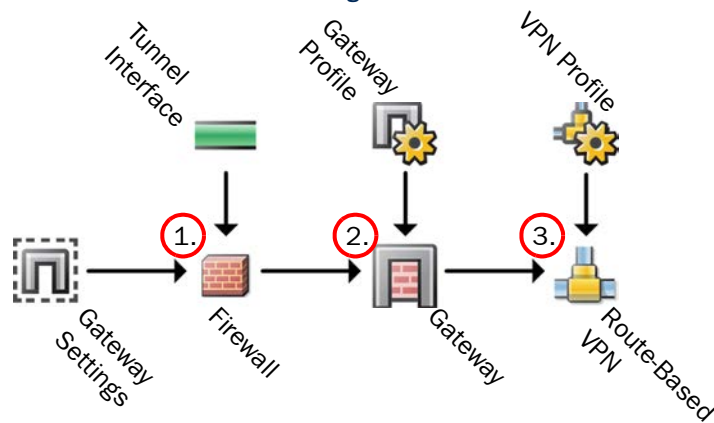
- *Internal VPN gateways* are McAfee Firewall/VPN engines that are managed by the Management Server (and administrative Domain) you are currently connected to with your Management Client.
- All other gateway devices are *external VPN gateways*, including McAfee Firewall/VPN engines that are managed by some different Management Server (or administrative Domain) than the one you are currently connected to with your Management Client.

With the Route-Based VPN, you can create only *gateway-to-gateway* VPN tunnels between gateway devices.

Due to the various authentication and encryption methods that are supported in IPsec VPNs, the number of settings is rather high. To save you from repeated configuration work, reusable profiles are used for storing different types of settings. These and other elements related to Route-Based VPN configuration are pictured in the illustration below.



**Illustration 30.1 Elements in the Route-Based VPN Configuration**



The main points of VPN configuration are (as indicated by the numbers in the illustration above):

1. The Firewall element contains the interface definitions for Tunnel Interfaces. The Firewall element refers to a Gateway Settings element that defines settings for advanced VPN performance tuning, which are optional to adjust.
2. The Gateway element sets VPN-related settings particular to one Firewall/VPN device. The Gateway element refers to a Gateway Profile element that contains information about the capabilities of different types of gateways. The Gateway Profiles of Internal VPN Gateways are automatically selected based on the installed software version.
3. The Route-Based VPN element combines the other elements together to define the settings used in each tunnel of the VPN. The Route-Based VPN element refers to a VPN Profile, which contains the IPsec authentication and encryption settings (IKE settings).

## Default Elements

The Route-Based VPN element is a predefined element. There can only be one Route-Based VPN element on each Management Server.

The following default VPN Profiles are available for use in the Route-Based VPN configuration:

- The VPN-A Suite VPN Profile contains the VPN settings specified for the cryptographic suite “VPN-A” in RFC 4308. It is provided to allow you to quickly try out VPNs without creating a VPN profile yourself. The profile also allows you to change settings that are not specified in RFC 4308 (such as the IKE mode for IKEv1), so you may need to adjust the settings to achieve a valid VPN in some configurations.

# 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* and the *McAfee SMC Administrator's Guide*.

## Task 1: Define Tunnel Interfaces

Tunnel Interfaces define end-points for tunnels in the Route-Based VPN, and allow non-routable packets, such as dynamic routing communications or Multicast streams, to travel between sites. Tunnel Interfaces encapsulate and encrypt outgoing traffic according to the VPN Profile selected for the Route-Based VPN tunnel, and de-encapsulate and decrypt incoming traffic. Any traffic that is routed to a Tunnel Interface is automatically sent through the tunnel to the peer end-point defined in the Route-Based VPN. The mapping of Tunnel Interfaces to physical network interfaces on the engine is done automatically by the engine operating system based on the routing configuration.

You can optionally add IPv4 and IPv6 addresses to a Tunnel Interface. Any IP address can be added to a Tunnel Interface, even if the same IP address is used on another interface or as a loopback IP address. Adding an IP address allows you to define the source IP address of traffic sent from the engine node itself. For example, an IP address is recommended to provide a source IP address for dynamic routing daemons, for IGMP proxy, and for Protocol Independent Multicast - Sparse-Mode (PIM-SM) configuration. If no IP address is added to the Tunnel Interface, the source IP address for traffic sent from the engine node is automatically selected according to the Bypass Default IP Address setting in the Interface Options for the firewall.

See [Single Firewall Configuration](#) (page 43) and [Firewall Cluster Configuration](#) (page 51) for more information about firewall interfaces.

## Task 2: Configure Routing and Antispoofing for Tunnel Interfaces

In the Route-Based VPN, the routing defines which traffic is sent through the VPN tunnel. The routing configuration also determines the physical network interfaces on the engine to which the Tunnel Interfaces are automatically mapped. You can either statically define which networks are reachable through each Tunnel Interface, or use dynamic routing to create the routes for traffic that is to be sent through the VPN tunnels.

The Antispoofing configuration is automatically generated for Tunnel Interfaces. See [Routing and Antispoofing](#) (page 73) for more information about configuring Routing and Antispoofing.

## Task 3: Define the Gateways

Internal VPN Gateway elements define settings for the Firewall/VPN devices in their role as VPN gateways. Usually, there is just one Gateway element per firewall, because you can use the same Gateway in several different Route-Based VPN tunnels. It is possible to create several Gateway elements to represent a single Firewall/VPN engine, but each Gateway element reserves a VPN end-point (IP address) that other Gateway elements cannot use. If you have both policy-based VPNs created using VPN elements and the Route-Based VPN, you cannot use the same end-points in both types of VPNs. In this case, you must create separate Gateway elements with different end-points for use in each type of VPN.

## Task 4: Define the VPN Profile

VPN Profile elements contain settings related to authentication, integrity checking, and encryption. This is the main point of configuration for IKE and IPsec settings (the settings used or agreed on during IKE SA and IPsec SA negotiations). If you want to use settings other than the ones defined by one of the default VPN Profile elements, you must define and customize a VPN Profile element.

You are generally free to choose any combination of settings as long as all gateways involved support those settings and are configured to accept them. The same VPN Profile can be used with several Route-Based VPN tunnels. You can use the same VPN Profile in both the Route-Based VPN and policy-based VPNs.

## Task 5: Define Route-Based VPN Tunnels

The Route-Based VPN element is a predefined element. There can only be one Route-Based VPN element on each Management Server. Rather than creating a new Route-Based VPN element for each tunnel, you edit the default Route-Based VPN to define all of the tunnels in the Route-Based VPN.

The configuration of each Route-Based VPN tunnel consists of the following settings:

1. Gateway A and Gateway B define the Internal VPN Gateway or External VPN Gateway element(s) that represent the peers at each end of the Route-Based VPN tunnel. Gateway A must always be an Internal VPN Gateway. Gateway B can be an Internal VPN Gateway or an External VPN Gateway.
2. End-Point A and End-Point B define the End-Point IP addresses to use for the tunnel. You cannot use the same End-Point in a Route-Based VPN tunnel and in a policy-based VPN tunnel. If loopback IP addresses are defined for the Internal VPN Gateway, you can use loopback IP addresses as End-Point IP addresses.
3. Tunnel Interface A and Tunnel Interface B define the Tunnel Interfaces on the firewalls through which Route-Based VPN traffic is routed. When a tunnel is created between an Internal VPN Gateway and an External VPN Gateway, only Tunnel Interface A needs to be defined in the tunnel configuration.
4. The Encryption cell defines the encryption mode for the tunnel. Different tunnels can use different encryption modes. The following encryption modes are supported:
  - Transport Mode: Defines an encapsulated tunnel with encryption settings defined by a VPN Profile.
  - Tunnel Mode: Defines an encapsulated tunnel with encryption settings defined in a policy-based VPN. See [Using the Route-Based VPN in Tunnel Mode](#) (page 301) for more information.
  - No Encryption: Defines a tunnel in which traffic is encapsulated but not encrypted. It is provided to allow encapsulation of traffic when the traffic does not need to be secured by a VPN. The No Encryption option is recommended only for the following uses:
    - creating unencrypted tunnels entirely within protected networks
    - testing and troubleshooting routing and connectivity without using VPN protection.

5. The Key defines the preshared key for the tunnel if a preshared key is used for authentication. A preshared key can only be used when Transport Mode is selected as the Encryption mode. In Tunnel Mode, the preshared key is defined in the policy-based VPN.
6. The Mode defines the encapsulation mode for the tunnel. Different tunnels can use different encapsulation modes. The following encapsulation modes are supported:
  - GRE (Generic Routing Encapsulation) is the industry standard, and is the recommended encapsulation mode in most cases.
  - IP-IP (IP in IP) is for use with third-party gateways that only support IP-IP.
  - SIT (Simple Internet Transition) is for use with IPv6 addresses.
7. (*Optional*) The TTL setting defines the initial time-to-live (TTL) value that is inserted into the encapsulation header of packets that enter the tunnel.
  - The TTL is needed when dynamic routing is used.
  - The default TTL is 64, but you can adjust this according to your needs.
8. (*Optional*) PMTU Discovery is used to automatically determine the largest common MTU that can be used for all the links in the network path without fragmentation.
9. (*Optional*) The Maximum Transmission Unit (MTU) setting specifies the largest unit of data that can be transmitted without fragmenting a packet. The MTU size should be as large as possible but not so large that it causes packets to be fragmented. You can usually use the default value, but you may need to configure the MTU setting according to your environment.

## Task 6: Add Access Rules to Allow the Traffic

Although the routing determines which traffic is sent into the Route-Based VPN tunnel, you must create Access rules to allow traffic between the internal network and the networks that are reachable through the Route-Based VPN. See [Access Rules](#) (page 101) for more information.

## Task 7: Refresh Firewall Policy

The interface configuration and the Route-Based VPN configuration are only applied when you refresh the Firewall Policy on the affected firewall(s).

## Configuring the Route-Based VPN with External Gateway Devices

In the Route-Based VPN, the term ‘external’ refers to any VPN gateway that is not controlled by the same Management Server (and the same administrative Domain) on which you are configuring the Route-Based VPN element. Often, external gateway devices are at a partner organization, not under your control, and are not McAfee Firewall/VPN devices.

Because the Route-Based VPN follows the IPsec standards, you can create VPNs between gateways of different brands by selecting the desired settings identically for both gateways. Any option that both gateways support is a valid option for the VPN.

The settings that must match are:

- The IKE SA settings.
- The IPsec SA settings.
- The Site definitions (IP addresses) defined for both gateways at both ends (possibly translated using NAT).
- The end-point identity type and value (this is often the IP address of each gateway, but other options are also possible).

You can also use the Route-Based VPN to create encrypted GRE tunnels with external gateways.

## Using the Route-Based VPN in Tunnel Mode

When you use the Route-Based VPN in Tunnel Mode, the encapsulation is provided by the Route-Based VPN, and the encryption is provided by a policy-based VPN. Before you define the encapsulated Tunnel Mode tunnel(s) in the Route-Based VPN, you must define the policy-based VPNs that provide the encryption. For more information, see [Providing Encryption for the Route-Based VPN in Tunnel Mode](#) (page 289).

Using the Route-Based VPN in Tunnel Mode allows you to do the following:

- Encrypt multiple encapsulated tunnels in the same VPN tunnel. This improves compatibility with third-party devices and cloud-based services that do not support multiple, separately encrypted tunnels, or that require the use of Tunnel Mode.
- Create multiple tunnels between remote and local sites when only one public IP address is available.
- Use Multi-Link with Route-Based VPN tunnels for high availability. See [Multi-Link and Policy-Based VPNs](#) (page 287).

To use the Route-Based VPN in Tunnel Mode, you create a Route-Based VPN tunnel that uses the same Internal VPN Gateway that you use in the policy-based VPN, and the same End-Point IP address that is included in the Site definition for the policy-based VPN. You select the policy-based VPN that provides the encryption as the Tunnel Mode Encryption for the tunnel.

The Route-Based VPN traffic is automatically directed into the policy-based VPN, but you must define Access rules to allow traffic between the internal network and the networks that are reachable through the Route-Based VPN. See [Access Rules](#) (page 101) for more information.

# Using the Route-Based VPN with Dynamic Routing

The Route-Based VPN can protect and route dynamic routing communications between sites to protect the confidentiality and integrity of the dynamic routing communications. Routing protocols, such as RIP, OSPF, and BGP, send non-routable multicast packets between routing devices, such as routers and firewalls. Because IPsec accepts only unicast traffic, these packets cannot be directly sent into IPsec VPN tunnels. Instead, dynamic routing communications are forwarded to Tunnel Interfaces, which encapsulate the traffic and send it into the Route-Based VPN tunnel. The following configuration considerations apply when the Route-Based VPN is used to protect dynamic routing protocols:

- A TTL value for the tunnel must be high enough to allow the packets to be routed through each hop in the route.
- If IP addresses are defined for the Tunnel Interfaces, the IP addresses must have a Netmask (*IPv4 addresses*) or Prefix Length (*IPv6 addresses*) that corresponds to a larger network space than the host network.

## Examples of Route-Based VPN Configurations

---

The examples in this section illustrate some common uses of the Route-Based VPN and general steps on how each scenario is configured.

### Protecting Dynamic Routing Communications

Company A is a large company with enterprise networks at multiple sites. The networks are currently connected with a private backbone network that is built with dynamic routing using OSPF. The administrators want to use public Internet networks for backup connectivity in case the private backbone fails. To be able to route the traffic, and to protect the confidentiality and integrity of the dynamic routing communications, the administrators decide to send dynamic routing communications through Route-Based VPN tunnels.

To configure the Route-Based VPN, the administrators:

1. Define Tunnel Interfaces on the firewalls that will act as Internal VPN Gateways at each site.
2. Add IP addresses to each Tunnel Interface.
3. Define Internal VPN Gateway elements to represent the Firewall/VPN devices.
4. Add a VPN tunnel to the Route-Based VPN between the appropriate Gateways, End-Points, and Tunnel Interfaces, and select the appropriate IPsec Profile and encapsulation Mode.
5. Define the following options for the tunnels:
  - TTL: Default.
  - MTU: Default.
  - PMTU Discovery: Enabled.
6. Save the changes to the Route-Based VPN.
7. Create Access rules that allow traffic between the internal networks and the networks that are reachable through the Route-Based VPN.
8. Refresh the policy on the firewalls that act as Internal VPN Gateways.
9. Configure dynamic routing on the command lines of the engines.

## Configuring a Route-Based VPN with an External Gateway

The administrators at Company B want to create a Route-Based VPN tunnel between their own network and a partner's network. The administrators:

1. Create a Network element to represent the partner's network.
2. Define a Tunnel Interface on the Company B firewall that will act as the Internal VPN Gateway.
3. Configure routing to define a route to the partner's network through the Tunnel Interface.
4. Define an Internal VPN Gateway element to represent the Company B Firewall/VPN device, and an External VPN Gateway element to represent the partner company's gateway device.
5. Add a VPN tunnel to the Route-Based VPN with the following settings:

Gateway A	End-Point A	Tunnel Interface A	Gateway B	End-Point B	Tunnel Interface B
Internal VPN Gateway element	End-Point IP address in the Internal Network	Tunnel Interface defined on the firewall	External VPN Gateway	End-Point IP address in the Partner Network	<Left empty>

6. Select an IPsec Profile and an encapsulation Mode that is compatible with the External VPN Gateway.
7. Save the changes to the Route-Based VPN.
8. Create an Access rule that allows traffic from the internal network to the partner network that is reachable through the Route-Based VPN.
9. Refresh the policy on the firewall that acts as the Internal VPN Gateway.





# APPENDICES

---

## **In this section:**

- Command Line Tools - 307**
- Default Communication Ports - 329**
- Predefined Aliases - 337**
- Situation Context Parameters - 341**
- Regular Expression Syntax - 345**
- Schema Updates for External LDAP Servers - 359**
- SNMP Traps and MIBs - 361**
- Multicasting - 377**
- Glossary - 385**
- Index - 415**



## APPENDIX A

# COMMAND LINE TOOLS

This appendix describes the command line tools for McAfee Security Management Center and the NGFW engines.



**Note** – Using the Management Client is the recommended configuration method, as most of the same tasks can be done through it.

The following sections are included:

- ▶ [Security Management Center Commands](#) (page 308)
- ▶ [NGFW Engine Commands](#) (page 319)
- ▶ [Server Pool Monitoring Agent Commands](#) (page 327)

# Security Management Center Commands

Security Management Center commands include commands for the Management Server, Log Server, Web Portal Server, and Authentication Server. Most of the commands are found in the *<installation directory>/bin/* directory. In Windows, the command line tools are \*.bat script files. In Linux, the files are \*.sh scripts.



**Note** – If you installed the Management Server in the C:\Program Files\McAfee\Security Management Center directory in Windows, some of the program data is stored in the C:\ProgramData\McAfee\Security Management Center directory. Command line tools may be found in the C:\Program Files\McAfee\Security Management Center\bin and/or the C:\ProgramData\McAfee\Security Management Center\bin directory.

Commands that require parameters must be run through the command line (cmd.exe in Windows). Commands that do not require parameters can alternatively be run through a graphical user interface, and may be added as shortcuts during installation.



**Caution** – login and password parameters are optional. Giving them as Command Line parameters may pose a security vulnerability. Do not enter login and password information unless explicitly prompted to do so by a Command Line tool.

Table A.1 Security Management Center Command Line Tools

Command	Description
<b>sgArchiveExport</b> [host=<Management Server Address [\Domain]>] [login=<login name>] [pass=<password>] [format=<exporter format: CSV or XML>] i=<input files and/or directories> [o=<output file name>] [f=<filter file name>] [e=<filter expression>] [-h   -help   -?] [-v]	Displays or exports logs from archive. This command is only available on the Log Server. The operation checks privileges for the supplied administrator account from the Management Server to prevent unauthorized access to the logs.  Enclose details in double quotes if they contain spaces.

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<b>sgArchiveExport</b> (continued)	<p><b>Host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username <b>root</b> is used.</p> <p><b>pass</b> defines the password for the user account.</p> <p><b>format</b> defines the file format for the output file. If this parameter is not defined, the XML format is used.</p> <p><b>i</b> defines the source from which the logs will be exported. Can be a folder or a file. The processing recurses into subfolders.</p> <p><b>o</b> defines the destination file where the logs will be exported. If this parameter is not defined, the output is displayed on screen.</p> <p><b>f</b> defines a file that contains the filtering criteria you want to use for filtering the log data. You can export log filters individually in the Management Client through <b>Tools→Save for Command Line Tools</b> in the filter's right-click menu.</p> <p><b>e</b> allows you to type in a filter expression manually (using the same syntax as exported filter files).</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p> <p><b>-v</b> displays verbose output on the command execution.</p> <p><b>Example</b> (exports logs from one full day to a file using a filter):  <code>sgArchiveExport login=admin pass=abc123  i=c:/mcafee/security_management_center/data/  archive/firewall/year2011/month12/. /sgB.day01/  f=c:/mcafee/security_management_center/export/  MyExportedFilter.flp format=CSV  o=MyExportedLogs.csv</code></p>
<b>sgBackupAuthSrv</b> <code>[pwd=&lt;password&gt;]  [path=&lt;destpath&gt;]  [nodiskcheck]  [comment=&lt;comment&gt;]  [-h   --help]</code>	<p>Creates a backup of Authentication Server user information. The backup file is stored in the <i>&lt;installation directory&gt;/backups/</i> directory. Backing up the Authentication only backs up Users, not the configuration of the Authentication Server. The Authentication Server configuration is included in the Management Server backup.</p> <p><b>pwd</b> enables encryption.</p> <p><b>path</b> defines the destination path.</p> <p><b>nodiskcheck</b> ignores free disk check before creating the backup.</p> <p><b>comment</b> allows you to enter a comment for the backup. The maximum length of a comment is 60 characters.</p> <p><b>-h or --help</b> displays information on using the script.</p> <p>Also see <b>sgRestoreAuthBackup</b>.</p>

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<b>sgBackupLogSrv</b> <b>[pwd=&lt;password&gt;]</b> <b>[path=&lt;destpath&gt;]</b> <b>[nodiskcheck]</b> <b>[comment=&lt;comment&gt;]</b> <b>[nofsstorage]</b> <b>[-h   --help]</b>	<p>Creates a backup of Log Server configuration data. The backup file is stored in the <i>&lt;installation directory&gt;/backups/</i> directory.</p> <p>Twice the size of log database is required on the destination drive. Otherwise, the operation fails.</p> <p><b>pwd</b> entering a password enables encryption.</p> <p><b>path</b> defines the destination path.</p> <p><b>nodiskcheck</b> ignores free disk check before creating the backup.</p> <p><b>comment</b> allows you to enter a comment for the backup. The maximum length of a comment is 60 characters.</p> <p><b>nofsstorage</b> creates a backup only of the log server configuration without the log data.</p> <p><b>-h</b> or <b>--help</b> displays information on using the script.</p> <p>Also see <b>sgRestoreLogBackup</b>.</p>
<b>sgBackupMgtSrv</b> <b>[pwd=&lt;password&gt;]</b> <b>[path=&lt;destpath&gt;]</b> <b>[nodiskcheck]</b> <b>[comment=&lt;comment&gt;]</b> <b>[-h   --help]</b>	<p>Creates a complete backup of the Management Server (including both the local configuration and the stored information in the configuration database). The backup file is stored in the <i>&lt;installation directory&gt;/backups/</i> directory.</p> <p>Twice the size of the Management Server database is required on the destination drive. Otherwise, the operation fails.</p> <p><b>pwd</b> entering a password enables encryption.</p> <p><b>path</b> defines the destination path.</p> <p><b>nodiskcheck</b> ignores free disk check before creating the backup.</p> <p><b>comment</b> allows you to enter a comment for the backup. The maximum length of a comment is 60 characters.</p> <p><b>-h</b> or <b>--help</b> displays information on using the script.</p> <p>Also see <b>sgRestoreMgtBackup</b> and <b>sgRecoverMgtDatabase</b>.</p>
<b>sgCertifyAuthSrv</b>	<p>Contacts the Management Server and creates a new certificate for the Authentication Server to allow secure communications with other SMC components. Renewing an existing certificate does not require changing the configuration of any other SMC components.</p>

**Table A.1 Security Management Center Command Line Tools (Continued)**

Command	Description
<b>sgCertifyLogSrv</b> <b>[host=&lt;Management Server Address</b> <b>[\Domain]&gt;]</b>	<p>Contacts the Management Server and creates a new certificate for the Log Server to allow secure communications with other SMC components. Renewing an existing certificate does not require changing the configuration of any other SMC components.</p> <p><b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>Domain</b> specifies the administrative Domain the Log Server belongs to if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used.</p> <p>The Log Server needs to be shut down before running this command. Restart the server after running this command.</p>
<b>sgCertifyMgtSrv</b>	<p>Creates a new certificate for the Management Server to allow secure communications between the SMC components. Renewing an existing certificate does not require changes on any other SMC components.</p> <p>The Management Server needs to be shut down before running this command. Restart the server after running this command.</p>
<b>sgCertifyWebPortalSrv</b> <b>[host=&lt;Management Server Address</b> <b>[\Domain]&gt;]</b>	<p>Contacts the Management Server and creates a new certificate for the Web Portal Server to allow secure communications with other SMC components. Renewing an existing certificate does not require changing the configuration of any other SMC components.</p> <p><b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>Domain</b> specifies the administrative Domain the Web Portal Server belongs to if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used.</p> <p>The Web Portal Server needs to be shut down before running this command. Restart the server after running this command.</p>
<b>sgChangeMgtIPOnAuthSrv &lt;IP</b> <b>address&gt;</b>	<p>Changes the Management Server's IP address in the Authentication Server's local configuration to the IP address you give as a parameter. Use this command if you change the Management Server's IP address.</p> <p>Restart the Authentication Server after running this command.</p>
<b>sgChangeMgtIPOnLogSrv &lt;IP address&gt;</b>	<p>Changes the Management Server's IP address in the Log Server's local configuration to the IP address you give as a parameter. Use this command if you change the Management Server's IP address.</p> <p>Restart the Log Server service after running this command.</p>

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<b>sgChangeMgtIPOnMgtSrv</b> <IP address>	Changes the Management Server's IP address in the local configuration to the IP address you give as a parameter. Use this command if you change the Management Server's IP address. Restart the Management Server service after running this command.
<b>sgClient</b>	Starts a locally installed Management Client.
<b>sgCreateAdmin</b>	Creates an unrestricted (superuser) administrator account. The Management Server needs to be stopped before running this command.
<b>sgExport</b> [host=<Management Server Address [\Domain]>] [login=<login name>] [pass=<password>] file=<file path and name> [type=<all/nw/ips/sv/rb/al> [name= <element name 1, element name 2, ...>] [recursion] [-system] [-h   -help   -?]	Exports elements stored on the Management Server to an XML file. Enclose details in double quotes if they contain spaces. <b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used. <b>Domain</b> specifies the administrative Domain for this operation if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used. <b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username root is used. <b>pass</b> defines the password for the user account. <b>file</b> defines the name and location of the export ZIP file. <b>type</b> specifies which types of elements are included in the export file: all for all exportable elements nw for network elements ips for IPS elements sv for services rb for security policies al for alerts vpn for VPN elements. name allows you to specify by name the element(s) that you want to export. <b>recursion</b> includes referenced elements in the export, for example, the network elements used in a policy that you export. <b>-system</b> includes any system elements that are referenced by the other elements in the export. <b>-h, -help, or -?</b> displays information on using the script.



Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<p><b>sgHA</b></p> <p>[<b>host</b>=&lt;Management Server Address [Domain]&gt;] [<b>login</b>=&lt;login name&gt;] [<b>pass</b>=&lt;password&gt;] [<b>master</b>=&lt;Management Server used as master server for the operation&gt;] [<b>-set-active</b>] [<b>-set-standby</b>] [<b>-check</b>] [<b>-retry</b>] [<b>-force</b>] [<b>-restart</b>] [<b>-h -help -?</b>]</p>	<p>Controls active and standby Management Servers. If you want to perform a full database synchronization, use the <code>sgOnlineReplication</code> command.</p> <p><b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>Domain</b> specifies the administrative Domain for this operation if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used.</p> <p><b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username <code>root</code> is used.</p> <p><b>pass</b> defines the password for the user account.</p> <p><b>master</b> defines the Management Server used as a master Management Server for the operation.</p> <p><b>-set-active</b> activates and locks all administrative Domains.</p> <p><b>-set-standby</b> deactivates and unlocks all administrative Domains.</p> <p><b>-check</b> checks that the Management Server's database is in sync with the master Management Server.</p> <p><b>-retry</b> retries replication if this has been stopped due to a recoverable error.</p> <p><b>-force</b> enforces the operation even if all Management Servers are not in sync. Note that using this option may cause instability if used carelessly.</p> <p><b>-restart</b> restarts the specified Management Server.</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>
<p><b>sgImport</b></p> <p>[<b>host</b>=&lt;Management Server Address [Domain]&gt;] [<b>login</b>=&lt;login name&gt;] [<b>pass</b>=&lt;password&gt;] <b>file</b>=&lt;file path and name&gt; [<b>-replace_all</b>] [<b>-h -help -?</b>]</p>	<p>Imports Management Server database elements from an XML file. When importing, existing (non-default) elements are overwritten if both the name and type match.</p> <p><b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>Domain</b> specifies the administrative Domain for this operation if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used.</p> <p><b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username <code>root</code> is used.</p> <p><b>pass</b> defines the password for the user account.</p> <p><b>file</b> defines the ZIP file whose contents you want to import.</p> <p><b>-replace_all</b> ignores all conflicts by replacing all existing elements with new ones.</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<pre>sgImportExportUser [host=&lt;Management Server Address [\Domain]&gt;] [login=&lt;login name&gt;] [pass=&lt;password&gt;] action=&lt;import export&gt; file=&lt;file path and name&gt; [-h -help -?]</pre>	<p>Imports and exports a list of Users and User Groups in an LDIF file from/to a Management Server's internal LDAP database. To import User Groups, all User Groups in the LDIF file must be directly under the <b>stonegate</b> top-level group (dc=stonegate).</p> <p><b>The user information in the export file is stored as plaintext. Handle the file securely.</b></p> <p><b>host</b> specifies the address of the Management Server. If the parameter is not defined, the loopback address (localhost) is used.</p> <p><b>Domain</b> specifies the administrative Domain for this operation if the system is divided into administrative Domains. If the Domain is not specified, the Shared Domain is used.</p> <p><b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username root is used.</p> <p><b>pass</b> defines the password for the user account.</p> <p><b>action</b> defines whether users are imported or exported.</p> <p><b>file</b> defines the file that is used for the operation.</p> <p><b>Example:</b> sgImportExportUser login=admin pass=abc123 action=export file=c:\temp\exportedusers.ldif</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>
<pre>sgInfo SG_ROOT_DIR FILENAME [fast] [-nolog] [-client] [-h -help -?]</pre>	<p>Creates a ZIP file that contains copies of configuration files and the system trace files. The resulting ZIP file is stored in the logged in user's home directory. The file location is displayed on the last line of screen output. Provide the generated file to support for troubleshooting purposes.</p> <p><b>SG_ROOT_DIR</b> Security Management Center installation directory.</p> <p><b>FILENAME</b> name of output file.</p> <p><b>-nolog</b> extended log server information is NOT collected.</p> <p><b>-client</b> collects traces only from the Management Client.</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<pre>sgOnlineReplication [login=&lt;login name&gt;] [pass=&lt;password&gt;] [active-server=&lt;name of active Management Server&gt;] [standby-server=&lt;name of additional Management Server&gt;] [standby-server-address=&lt;IP address of additional Management Server&gt;] [-nodisplay] [-h -help -?]</pre>	<p>Replicates the Management Server's database from the active Management Server to an additional Management Server. The Management Server to which the database is replicated must be shut down before running this command. Restart the Management Server after running this command.</p> <p><b>Note!</b> Use this script to replicate the database only if the additional Management Server's configuration has been corrupted, the additional Management Server's certificate has expired, or in new SMC installations if the automatic database replication between the Management Servers has not succeeded. Otherwise, synchronize the database through the Management Client. See the <i>McAfee SMC Administrator's Guide</i> for more information.</p> <p><b>login</b> defines the username for the account that is used for this operation. If this parameter is not defined, the username <b>root</b> is used.</p> <p><b>pass</b> defines the password for the user account.</p> <p><b>active-server</b> option specifies the IP address of the active Management Server from which the Management database is replicated.</p> <p><b>standby-server</b> option specifies the name of the additional Management Server to which the Management database is replicated.</p> <p><b>standby-server-address</b> option specifies the IP address of the additional Management Server to which the Management database is replicated.</p> <p><b>-nodisplay</b> sets a text only console.</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>
<pre>sgReinitializeLogServer</pre>	<p><b>Note!</b> This script is located in <i>&lt;installation directory&gt;/bin/install</i>.</p> <p>Creates a new Log Server configuration if the configuration file has been lost.</p>
<pre>sgRestoreArchive &lt;ARCHIVE_DIR&gt;</pre>	<p>Restores logs from archive files to the Log Server. This command is available only on the Log Server.</p> <p><b>ARCHIVE_DIR</b> is the number of the archive directory (0 – 31) from where the logs will be restored. By default, only archive directory 0 is defined. The archive directories can be defined in the <i>&lt;installation directory&gt;/data/LogServerConfiguration.txt</i> file:</p> <p><b>ARCHIVE_DIR_xx=PATH.</b></p>

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<b>sgRestoreAuthBackup</b> [-pwd=<password>] [-backup=<backup file name>] [-nodiskcheck] [-h -help]	Restores the Authentication Server user information from a backup file in the <installation directory>/backups/ directory. Apply the Authentication Server's configuration after this command. -pwd defines a password for encrypted backup. -backup defines a name for the backup file. -nodiskcheck ignores free disk check before backup restoration. -h or -help displays information on using the script.
<b>sgRestoreLogBackup</b> [-pwd=<password>] [-backup=<backup file name>] [-nodiskcheck] [-overwrite-syslog-template] [-h -help]	Restores the Log Server (logs and/or configuration files) from a backup file in the <installation directory>/backups/ directory. Apply the Authentication Server's configuration after this command. -pwd defines a password for encrypted backup. -backup defines a name for the backup file. -nodiskcheck ignores free disk check before backup restoration. -overwrite-syslog-template overwrites a syslog template file if found in the backup. -h or -help displays information on using the script.
<b>sgRestoreMgtBackup</b> [-pwd=<password>] [-backup=<backup file name>] [-nodiskcheck] [-h -help]	Restores the Management Server (database and/or configuration files) from a backup file in the <installation directory>/backups/ directory. -pwd defines a password for encrypted backup. -backup defines a name for the backup file. -nodiskcheck ignores free disk check before backup restoration. -h or -help displays information on using the script.
<b>sgRevert</b>	<b>Note!</b> This script is located in <installation directory>/bin/uninstall. Reverts to the previous installation saved during the upgrade process. The previous installation can be restored at any time, even after a successful upgrade.
<b>sgShowFingerPrint</b>	Displays the CA certificate's fingerprint on the Management Server.
<b>sgStartAuthSrv</b>	Starts the Authentication Server.
<b>sgStartLogSrv</b>	Starts the Log Server and its database.
<b>sgStartMgtDatabase</b>	Starts the Management Server's database. There is usually no need to use this script.

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<code>sgStartMgtSrv</code>	Starts the Management Server and its database.
<code>sgStartWebPortalSrv</code>	Starts the Web Portal Server.
<code>sgStopLogSrv</code>	Stops the Log Server.
<code>sgStopMgtSrv</code>	Stops the Management Server and its database.
<code>sgStopMgtDatabase</code>	Stops the Management Server's database. There is usually no need to use this script.
<code>sgStopWebPortalSrv</code>	Stops the Web Portal Server.
<code>sgStopRemoteMgtSrv</code> [ <code>host=&lt;Management Server Host Name&gt;</code> ] [ <code>login=&lt;login name&gt;</code> ] [ <code>pass=&lt;password&gt;</code> ] [ <code>-h -help -?</code> ]	Stops the Management Server service when run without arguments. To stop a remote Management Server service, provide the arguments to connect to the Management Server. <b>host</b> is the Management Server's host name if not localhost. <b>login</b> is an SMC administrator account for the login. <b>pass</b> is the password for the administrator account. <b>-h</b> , <b>-help</b> , or <b>-?</b> displays information on using the script.

Table A.1 Security Management Center Command Line Tools (Continued)

Command	Description
<b>sgTextBrowser</b> [ <b>host</b> =<Management Server address [\Domain]>] [ <b>login</b> =<login name>] [ <b>pass</b> =<password>] [ <b>format</b> =<CSV/XML>] [ <b>o</b> =<output file>] [ <b>f</b> =<filter file> ] [ <b>e</b> =<filter expression> ] [ <b>m</b> =<current/stored>] [ <b>limit</b> =<maximum number of unique records to fetch>] [-h -help -?]	<p>Displays or exports current or stored logs. This command is available on the Log Server.</p> <p>Enclose the file and filter names in double quotes if they contain spaces.</p> <p><b>host</b> defines the address of the Management Server used for checking the login information. If this parameter is not defined, Management Server is expected to be on the same host where the script is run. If Domains are in use, you can specify the Domain the Log Server belongs to. If domain is not specified, the Shared Domain is used.</p> <p><b>login</b> defines the username for the account that is used for this export. If this parameter is not defined, the username root is used.</p> <p><b>pass</b> defines the password for the user account used for this operation.</p> <p><b>format</b> defines the file format for the output file. If this parameter is not defined, the XML format is used.</p> <p><b>o</b> defines the destination output file where the logs will be exported. If this parameter is not defined, the output is displayed on screen.</p> <p><b>f</b> defines the exported filter file that you want to use for filtering the log data.</p> <p><b>e</b> defines the filter that you want to use for filtering the log data. Type the name as shown in the Management Client.</p> <p><b>m</b> defines whether you want to view or export logs as they arrive on the Log Server (current) or logs stored in the active storage directory (stored). If this option is not defined, the current logs are used.</p> <p><b>limit</b> defines the maximum number of unique records to be fetched. The default value is unlimited.</p> <p><b>-h, -help, or -?</b> displays information on using the script.</p>

# NGFW Engine Commands

The commands in the following two tables can be run on the command line on Firewall, Layer 2 Firewall, IPS engines and/or Master Engines.



**Note** – All command line tools that are available for single Security Engines are also available for Virtual Security Engines that have the same role. However, there is no direct access to the command line of Virtual Security Engines. Commands to Virtual Security Engines must be sent from the command line of the Master Engine using the `se-virtual-engine` command.

Table A.2 NGFW Engine Command Line Tools

Command	Engine Role	Description
<b>avdbfetch</b> [ <b>--dbzip</b> =<path to zip file>] [ <b>--proxy</b> =<proxy address>] [ <b>--proxy-pass</b> =<proxy password>] [ <b>--proxy-user</b> =<proxy user>] [ <b>--url</b> =<url path>]	Firewall	<p>If the separately-licensed anti-virus feature is enabled on a Firewall, use this command to manually update the anti-virus database.</p> <p><b>--dbzip</b> defines the location of the locally-stored database zip file. This option can be used when there is not an internet connection and you have manually copied the database to a folder on the engine. This parameter does not need to be defined if the zip file is stored in <code>/var/tmp</code>.</p> <p><b>--proxy</b> defines the address of an HTTP proxy if one is required to connect to the database mirror.</p> <p><b>--proxy-pass</b> defines the password (if required) for the HTTP proxy.</p> <p><b>--proxy-user</b> defines the username (if required) for the HTTP proxy.</p> <p><b>--url</b> defines the address of the database mirror. If not specified, the default address is <code>http://update.nai.com/Products/CommonUpdater</code>.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<pre> <b>sg-blacklist</b> <b>show</b> [-v] [-f <i>FILENAME</i>]   <b>add</b> [ [-i <i>FILENAME</i>]   [<b>src</b> <i>IP_ADDRESS/MASK</i>] [<b>src6</b> <i>IPv6_ADDRESS/PREFIX</i>] [<b>dst</b> <i>IP_ADDRESS/MASK</i>] [<b>dst6</b> <i>IPv6_ADDRESS/PREFIX</i>] [<b>proto</b> { <i>tcp udp icmp NUM</i> }] [<b>srcport</b> <i>PORT</i>{ -<i>PORT</i> }] [<b>dstport</b> <i>PORT</i>{ -<i>PORT</i> }] [<b>duration</b> <i>NUM</i>] ]   <b>del</b> [ [-i <i>FILENAME</i>]   [<b>src</b> <i>IP_ADDRESS/MASK</i>] [<b>src6</b> <i>IPv6_ADDRESS/PREFIX</i>] [<b>dst</b> <i>IP_ADDRESS/MASK</i>] [<b>dst6</b> <i>IPv6_ADDRESS/PREFIX</i>] [<b>proto</b> { <i>tcp udp icmp NUM</i> }] [<b>srcport</b> <i>PORT</i>{ -<i>PORT</i> }] [<b>dstport</b> <i>PORT</i>{ -<i>PORT</i> }] [<b>duration</b> <i>NUM</i>] ]   <b>iddel</b> <i>NODE_ID ID</i>   <b>flush</b> </pre>	Firewall, Layer 2 Firewall, IPS	<p>Used to view, add, or delete active blacklist entries. The blacklist is applied as defined in Access Rules.</p> <p><b>Commands:</b></p> <p><b>show</b> 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 <b>-f</b> option to specify a storage file to view (/data/blacklist/db_&lt;number&gt;). The <b>-v</b> option adds operation's details to the output.</p> <p><b>add</b> creates a new blacklist entry. Enter the parameters (see below) or use the <b>-i</b> option to import parameters from a file.</p> <p><b>del</b> deletes the first matching blacklist entry. Enter the parameters (see below) or use the <b>-i</b> option to import parameters from a file.</p> <p><b>iddel</b> <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 <b>show</b> command).</p> <p><b>flush</b> deletes all blacklist entries.</p>



Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<b>sg-blacklist</b> (continued)	Firewall, Layer 2 Firewall, IPS	<p><b>Add/Del Parameters:</b></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><b>src</b> <i>IP_ADDRESS/MASK</i> defines the source IP address and netmask to match. Matches any IP address by default.</p> <p><b>src6</b> <i>IPv6_ADDRESS/PREFIX</i> defines the source IPv6 and prefix length to match. Matches any IPv6 address by default.</p> <p><b>dst</b> <i>IP_ADDRESS/MASK</i> defines the destination IP address and netmask to match. Matches any IP address by default.</p> <p><b>dst6</b> <i>IPv6_ADDRESS/PREFIX</i> defines the destination IPv6 address and prefix length to match. Matches any IPv6 address by default.</p> <p><b>proto</b> { <i>tcp/udp/icmp/NUM</i> } defines the protocol to match by name or protocol number. Matches all IP traffic by default.</p> <p><b>srcport</b> <i>PORT</i> [-<i>PORT</i>] defines the TCP/UDP source port or range to match. Matches any port by default.</p> <p><b>dstport</b> <i>PORT</i> [-<i>PORT</i>] defines the TCP/UDP destination port or range to match. Matches any port by default.</p> <p><b>duration</b> <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><b>Examples:</b></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>
<b>sg-bootconfig</b> [--primary-console = <i>tty0/ttyS PORT,SPEED</i> ] [--secondary-console = [ <i>tty0/ttyS 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	Firewall, Layer 2 Firewall, IPS	<p>Used to edit boot command parameters for future bootups.</p> <p><b>--primary-console</b>=<i>tty0/ttyS PORT,SPEED</i> parameter defines the terminal settings for the primary console.</p> <p><b>--secondary-console</b>= [<i>tty0/ttyS PORT,SPEED</i>] parameter defines the terminal settings for the secondary console.</p> <p><b>--flavor</b>=<i>up/smp</i> [-<i>kdb</i>] parameter defines whether the kernel is uniprocessor or multiprocessor.</p> <p><b>--initrd</b>=<i>yes/no</i> parameter defines whether Ramdisk is enabled or disabled.</p> <p><b>--crashdump</b>=<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><b>--append</b>=<i>kernel options</i> parameter defines any other boot options to add to the configuration.</p> <p><b>--help</b> parameter displays usage information.</p> <p><b>apply</b> command applies the specified configuration options.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<code>sg-clear-all</code>	Firewall, Layer 2 Firewall, IPS	<p><b>Note! Use this only if you want to clear all configuration information from the engine.</b></p> <p>This command resets all configuration information from the engine. It does not remove the engine software. After using this command, you must reconfigure the engine using the <code>sg-reconfigure</code> command.</p>
<code>sg-cluster</code> <code>[-v &lt;virtual engine ID&gt;]</code> <code>[status [-c SECONDS]]</code> <code>[versions]</code> <code>[online]</code> <code>[lock-online]</code> <code>[offline]</code> <code>[lock-offline]</code> <code>[standby]</code> <code>[safe-offline]</code> <code>[force-offline]</code>	Firewall, Layer 2 Firewall, IPS	<p>Used to display or change the status of the node.</p> <p><b>-v &lt;virtual engine ID&gt;</b> (<i>Master Engine only</i>) option specifies the ID of the Virtual Security Engine on which to execute the command.</p> <p><b>status [-c SECONDS]</b> command displays cluster status. When <b>-c SECONDS</b> is used, status is shown continuously with the specified number of seconds between updates.</p> <p><b>versions</b> command displays the engine software versions of the nodes in the cluster.</p> <p><b>online</b> command sends the node online.</p> <p><b>lock-online</b> command sends the node online and keeps it online even if another process tries to change its state.</p> <p><b>offline</b> command sends the node offline.</p> <p><b>lock-offline</b> command sends the node offline and keeps it offline even if another process tries to change its state.</p> <p><b>standby</b> command sets an active node to standby.</p> <p><b>safe-offline</b> command sets the node to offline only if there is another online node.</p> <p><b>force-offline</b> command sets the node online regardless of state or any limitations. Also sets all other nodes offline.</p>
<code>sg-contact-mgmt</code>	Firewall, Layer 2 Firewall, IPS	<p>Used for establishing a trust relationship with the Management Server as part of engine installation or reconfiguration (see <code>sg-reconfigure</code> below). The engine contacts the Management Server using the one-time password created when the engine's initial configuration is saved.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<b>sg-dynamic-routing</b> <b>[start]</b> <b>[stop]</b> <b>[restart]</b> <b>[force-reload]</b> <b>[backup &lt;file&gt;]</b> <b>[restore &lt;file&gt;]</b> <b>[sample-config]</b> <b>[route-table]</b> <b>[info]</b>	Firewall	<p><b>start</b> starts the Quagga routing suite.</p> <p><b>stop</b> stops the Quagga routing suite and flushes all routes made by zebra.</p> <p><b>restart</b> restarts the Quagga routing suite.</p> <p><b>force-reload</b> forces reload of the saved configuration.</p> <p><b>backup &lt;file&gt;</b> backs up the current configuration to a compressed file.</p> <p><b>restore &lt;file&gt;</b> restores the configuration from the specified file.</p> <p><b>sample-config</b> creates a basic configuration for Quagga.</p> <p><b>route-table</b> prints the current routing table.</p> <p><b>info</b> displays the help information for the sg-dynamic-routing command, and detailed information about Quagga suite configuration with vtysh.</p>
<b>sg-ipsec -d</b> <b>[-u &lt;username[@domain]&gt;  </b> <b>-si &lt;session id&gt;  </b> <b>-ck &lt;ike cookie&gt;  </b> <b>-tri &lt;transform id&gt;</b> <b>-ri &lt;remote ip&gt;  </b> <b>-ci &lt;connection id&gt;]</b>	Firewall	<p>Deletes VPN-related information (use <b>vpninfo</b> command to view the information). Option <b>-d</b> (for delete) is mandatory.</p> <p><b>-u</b> deletes the VPN session of the named VPN client user. You can enter the user account in the form <b>&lt;username@domain&gt;</b> if there are several user storage locations (LDAP domains).</p> <p><b>-si</b> deletes the VPN session of a VPN client user based on session identifier.</p> <p><b>-ck</b> deletes the IKE SA (Phase one security association) based on IKE cookie.</p> <p><b>-tri</b> deletes the IPSEC SAs (Phase two security associations) for both communication directions based on transform identifier.</p> <p><b>-ri</b> deletes all SAs related to a remote IP address in gateway-to-gateway VPNs.</p> <p><b>-ci</b> deletes all SAs related to a connection identifier in gateway-to-gateway VPNs.</p>
<b>sg-logger</b> <b>-f FACILITY_NUMBER</b> <b>-t TYPE_NUMBER</b> <b>[-e EVENT_NUMBER]</b> <b>[-i "INFO_STRING"]</b> <b>[-s]</b> <b>[-h]</b>	Firewall, Layer 2 Firewall, IPS	<p>Used in scripts to create log messages with the specified properties.</p> <p><b>-f FACILITY_NUMBER</b> parameter defines the facility for the log message.</p> <p><b>-t TYPE_NUMBER</b> parameter defines the type for the log message.</p> <p><b>-e EVENT_NUMBER</b> parameter defines the log event for the log message. The default is 0 (H2A_LOG_EVENT_UNDEFINED).</p> <p><b>-i "INFO_STRING"</b> parameter defines the information string for the log message.</p> <p><b>-s</b> parameter dumps information on option numbers to stdout</p> <p><b>-h</b> parameter displays usage information.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<b>sg-raid</b> <b>[-status] [-add] [-re-add]</b> <b>[-force] [-help]</b>	Firewall, Layer 2 Firewall, IPS	<p>Configures a new hard drive. This command is only for McAfee NGFW appliances that support RAID (Redundant Array of Independent Disks) and have two hard drives.</p> <p><b>-status</b> option displays the status of the hard drive.</p> <p><b>-add</b> options adds a new empty hard drive.</p> <p>Use <b>-add -force</b> if you want to add a hard drive that already contains data and you want to overwrite it.</p> <p><b>-re-add</b> adds a hard drive that is already partitioned. This command prompts for the drive and partition for each degraded array.</p> <p>Use <b>-re-add -force</b> if you want to check all the arrays.</p> <p><b>-help</b> option displays usage information.</p>
<b>sg-reconfigure</b> <b>[--boot]</b> <b>[--maybe-contact]</b> <b>[--no-shutdown]</b>	Firewall, Layer 2 Firewall, IPS	<p>Used for reconfiguring the node manually.</p> <p><b>--boot</b> option applies bootup behavior. Do not use this option unless you have a specific need to do so.</p> <p><b>--maybe-contact</b> option contacts the Management Server if requested. This option is only available on firewall engines.</p> <p><b>--no-shutdown</b> 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.</p>
<b>sg-selftest [-d] [-h]</b>	Firewall	<p>Runs cryptography tests on the engine.</p> <p><b>-d</b> option runs the tests in debug mode.</p> <p><b>-h</b> option displays usage information.</p>
<b>sg-status [-l] [-h]</b>	Firewall, Layer 2 Firewall, IPS	<p>Displays information on the engine's status.</p> <p><b>-l</b> option displays all available information on engine status.</p> <p><b>-h</b> option displays usage information.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<b>sg-toggle-active</b> <i>SHA1 SIZE</i>   <b>--force</b> [ <b>--debug</b> ]	Firewall, Layer 2 Firewall, IPS	<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> (<b>ls -l /var/run/stonegate</b>).</p> <p>The <i>SHA1 SIZE</i> 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> <p><b>--debug</b> option reboots the engine with the debug kernel.</p> <p><b>--force</b> option switches the active configuration without first verifying the signature of the inactive partition.</p>
<b>sg-upgrade</b>	Firewall	Upgrades the node by rebooting from the installation DVD. Alternatively, the node can be upgraded remotely using the Management Client.
<b>sg-version</b>	Firewall, Layer 2 Firewall, IPS	Displays the software version and build number for the node.
<b>se-virtual-engine</b> <b>-l</b>   <b>--list</b> <b>-v</b> <virtual engine ID> <b>-e</b>   <b>--enter</b> <b>-E</b> "<command [options]>" <b>-h</b>   <b>--help</b>	Firewall (Master Engine only)	<p>Used to send commands to Virtual Firewalls from the command line of the Master Engine. All commands that can be used for the Firewall role can also be used for Virtual Firewalls.</p> <p><b>-l</b> or <b>--list</b> list the active Virtual Security Engines.</p> <p><b>-v</b> &lt;virtual engine ID&gt; specifies the ID of the Virtual Security Engine on which to execute the command.</p> <p><b>-e</b> or <b>--enter</b> enters the command shell for the Virtual Security Engine specified with the <b>-v</b> option. To exit the command shell, type <code>exit</code>.</p> <p><b>-E</b> "&lt;command [options]&gt;" executes the specified command on the Virtual Security Engine specified with the <b>-v</b> option.</p> <p><b>-h</b> or <b>--help</b> shows the help message for the <code>se-virtual-engine</code> command.</p>

Table A.2 NGFW Engine Command Line Tools (Continued)

Command	Engine Role	Description
<b>sginfo</b> [-f] [-d] [-s] [-p] [--] [--help]	Firewall, Layer 2 Firewall, IPS	Gathers system information you can send to McAfee support if you are having problems. Use this command only when instructed to do so by McAfee 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.

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

Table A.3 General Command Line Tools on Engines

Command	Description
<b>dmesg</b>	Shows system logs and other information. Use the -h option to see usage.
<b>halt</b>	Shuts down the system.
<b>ip</b>	Displays IP address information. Type the command without options to see usage. <b>Example:</b> type <b>ip addr</b> for basic information on all interfaces.
<b>ping</b>	Tests connectivity with ICMP echo requests. Type the command without options to see usage.
<b>ps</b>	Reports the status of running processes.
<b>reboot</b>	Reboots the system.
<b>scp</b>	Secure copy. Type the command without options to see usage.
<b>sftp</b>	Secure FTP. Type the command without options to see usage.
<b>ssh</b>	SSH client (for opening a terminal connection to other hosts). Type the command without options to see usage.
<b>tcpdump</b>	Gives information on network traffic. Use the -h option to see usage. You can also analyze network traffic by creating tcpdump files from the Management Client with the Traffic Capture feature. See the <i>McAfee SMC Administrator's Guide</i> for more information.
<b>top</b>	Displays the top CPU processes taking most processor time. Use the -h option to see usage.
<b>traceroute</b>	Traces the route packets take to the specified destination. Type the command without options to see usage.
<b>vpninfo</b>	Displays VPN information and allows you to issue some basic commands. Type the command without options to see usage.

# Server Pool Monitoring Agent Commands

You can test and monitor the Server Pool Monitoring Agents on the command line with the commands described in the table below.

Table A.4 Server Pool Monitoring Agent Commands

Command	Description
<b>agent</b> [-v <i>level</i> ] [-c <i>path</i> ] [test [ <i>files</i> ]] [syntax [ <i>files</i> ]]	<p>(Windows only) Allows you to test different configurations before activating them.</p> <p>-v <i>level</i> Set the verbosity level. The default level is 5. Levels 6-8 are for debugging where available.</p> <p>-c <i>path</i> Use the specified path as the first search directory for the configuration.</p> <p>test [<i>files</i>] Run in the test mode - status queries do not receive a response. If you specify the files, they are used for reading the configuration instead of the default files. The output is directed to syslog or eventlog instead of the console where the command was run unless you use the -d option.</p> <p>syntax [<i>files</i>] Check the syntax in the configuration file. If no files are specified, the default configuration files are checked.</p>
<b>sgagentd</b> [-d] [-v <i>level</i> ] [-c <i>path</i> ] [test [ <i>files</i> ]] [syntax [ <i>files</i> ]]	<p>(Linux only) Allows you to test different configurations before activating them.</p> <p>-d Don't Fork as a daemon. All log messages are printed to stdout or stderr only.</p> <p>-v <i>level</i> Set the verbosity level. The default level is 5. Levels 6-8 are for debugging where available.</p> <p>-c <i>path</i> Use the specified path as the first search directory for the configuration.</p> <p>test [<i>files</i>] Run in the test mode - status queries do not receive a response. If you specify the files, they are used for reading the configuration instead of the default files. The output is directed to syslog or eventlog instead of the console where the command was run unless you use the -d option.</p> <p>syntax [<i>files</i>] Check the syntax in the configuration file. If no files are specified, the default configuration files are checked. The output is directed to syslog or eventlog instead of the console where the command was run unless you use the -d option.</p>

Table A.4 Server Pool Monitoring Agent Commands (Continued)

Command	Description
<code>sgmon</code> <code>[status/info/proto</code> <code>]</code> <code>[-p port]</code> <code>[-t timeout]</code> <code>[-a id]</code> <code>host</code>	<p>Sends a UDP query to the specified host and waits for a response until received, or until the timeout limit is reached.</p> <p>The request type can be defined as a parameter. If no parameter is given, <code>status</code> is requested. The commands are:</p> <ul style="list-style-type: none"><li><code>status</code> - query the status.</li><li><code>info</code> - query the agent version.</li><li><code>proto</code> - query the highest supported protocol version.</li></ul> <p><code>-p port</code> Connect to the specified port instead of the default port.</p> <p><code>-t timeout</code> Set the timeout (in seconds) to wait for a response.</p> <p><code>-a id</code> Acknowledge the received log messages up to the specified id. Each response message has an id, and you may acknowledge more than one message at a given time by using the id parameter. Note that messages acknowledged by <code>sgmon</code> will no longer appear in the firewall logs.</p> <p><code>host</code> The IP address of the host to connect to. To get the status locally, you may give <code>localhost</code> as the host argument. This parameter is mandatory.</p>



## APPENDIX B

# DEFAULT COMMUNICATION PORTS

This chapter lists the default ports used in connections between SMC components and the default ports SMC components use with external components.

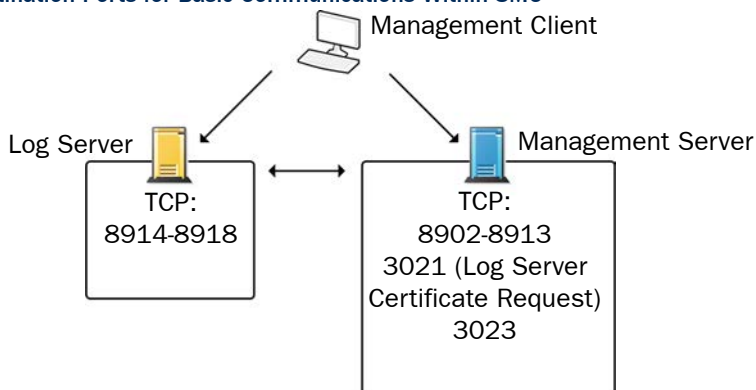
The following sections are included:

- ▶ [Security Management Center Ports](#) (page 330)
- ▶ [Security Engine Ports](#) (page 333)

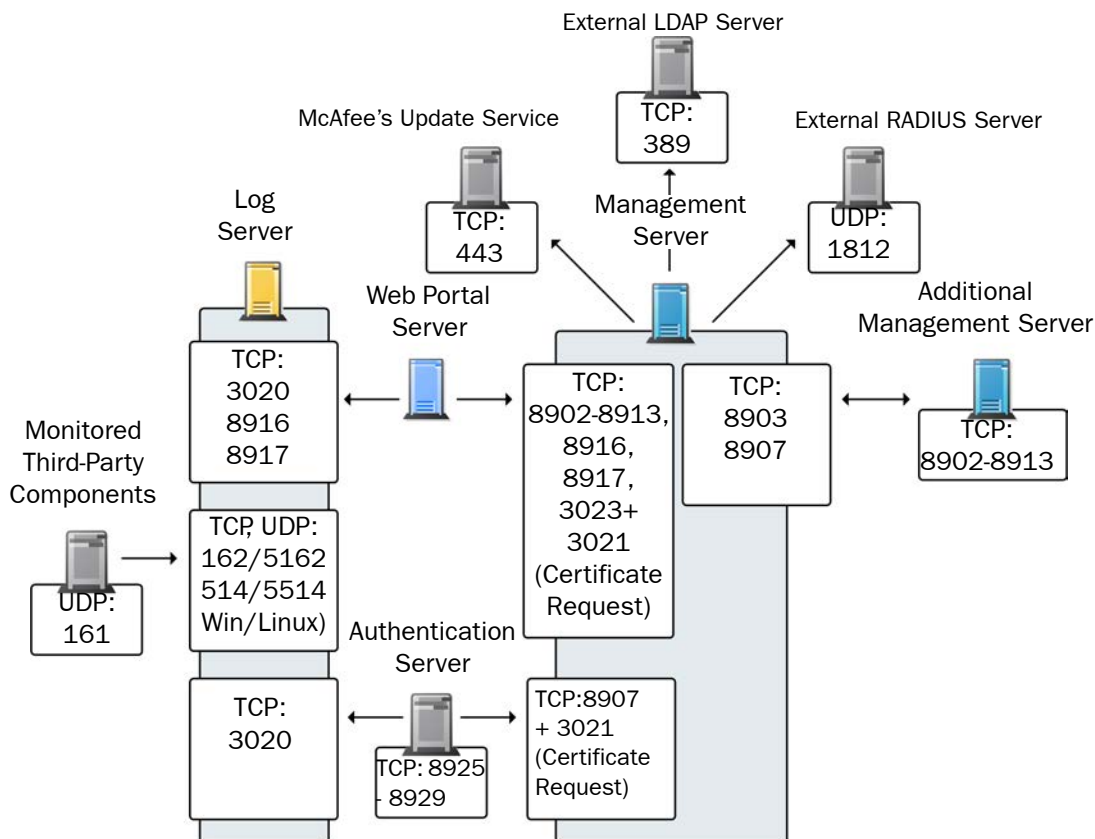
# Security Management Center Ports

The illustrations below present an overview to the most important default ports used in communications between the Security 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**



**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 Security Management Center Default Ports**

<b>Listening Host</b>	<b>Port/Protocol</b>	<b>Contacting Hosts</b>	<b>Service Description</b>	<b>Service Element Name</b>
Additional Management Servers	8902-8913/TCP	Management Server	Database replication (push) to the additional Management Server.	SG Control
Authentication Server	8925-8929/TCP	Management Server	Security 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	2055/UDP	Monitored third-party components	NetFlow or IPFIX reception from third-party components. Port 2055 is used in both Windows and Linux.	NetFlow (UDP)
Log Server	3020/TCP	Authentication Server, Log Server, Web Portal Server, Security Engines	Alert sending from the Authentication Server, Log Server, and Web Portal Server. Log and alert messages; monitoring of blacklists, connections, status, and statistics from Security Engines.	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)

**Table B.1 Security Management Center Default Ports (Continued)**

<b>Listening Host</b>	<b>Port/Protocol</b>	<b>Contacting Hosts</b>	<b>Service Description</b>	<b>Service Element Name</b>
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	3023/TCP	Additional Management Servers, Log Server, Web Portal Server	Log Server and Web Portal Server status monitoring. Status information from an additional Management Server to the active Management Server.	SG Status Monitoring
Management Server	8903, 8907/TCP	Additional Management Servers	Database replication (pull) to the additional Management Server.	SG Control
Management Server	8907/TCP	Authentication Server	Status monitoring.	SG Control
Monitored third-party components	161/UDP	Log Server	SNMP status probing to external IP addresses.	SNMP (UDP)
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)
SMC servers	443/TCP	Management Server	Update packages, engine upgrades, and licenses from update-pool.stonesoft.com and smc-pool.stonesoft.com.	HTTPS
Syslog server	514/UDP, 5514/UDP	Log Server	Log data forwarding to syslog servers. The default ports can be modified in the LogServerConfiguration.txt file.	Syslog (UDP) [Partial match]
Third-party components	2055/UDP	Log Server	NetFlow or IPFIX forwarding to third-party components. Port 2055 is used in both Windows and Linux.	NetFlow (UDP)

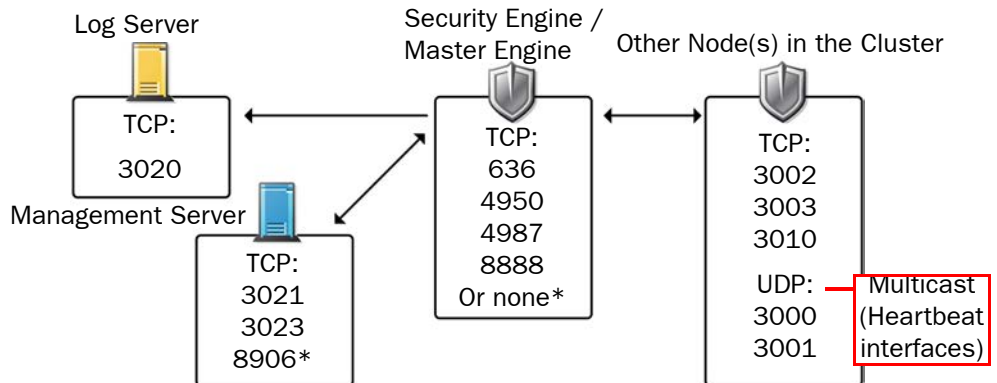
# Security Engine Ports

The illustrations below present an overview to the most important default ports used in communications between Security Engines and the SMC and between clustered Security Engine nodes. See the table below for a complete list of default ports for the engines.



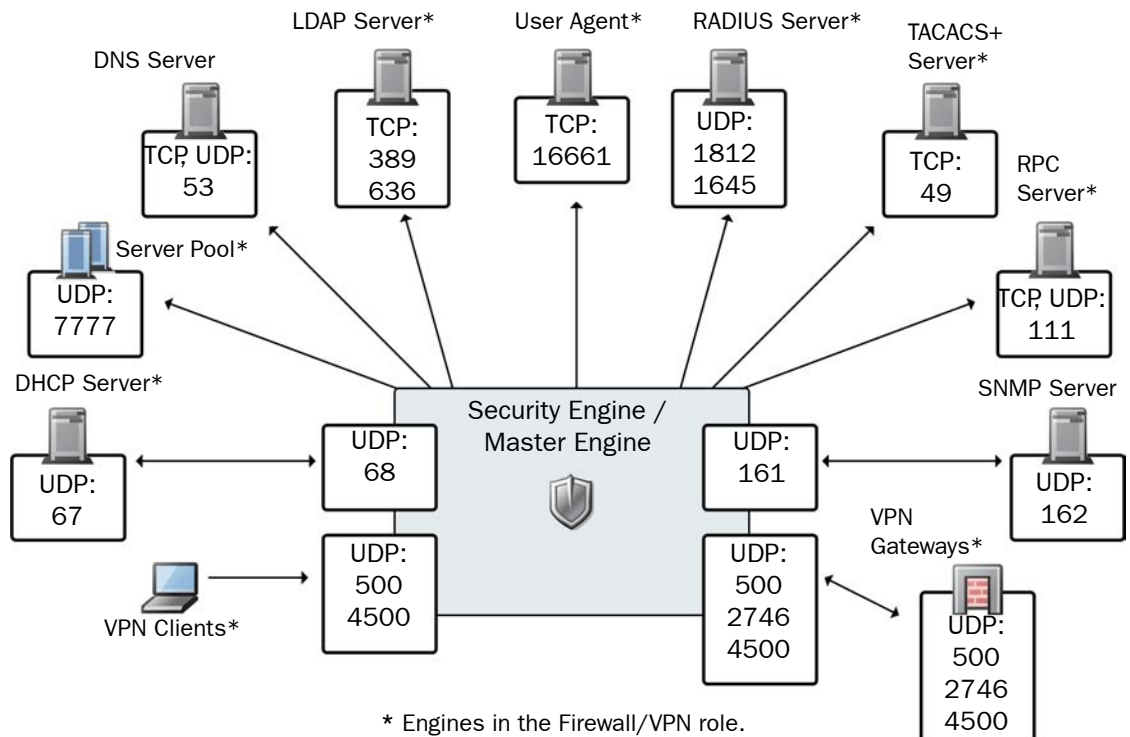
**Note – Master Engines use the same default ports as clustered Security Engines. Virtual Security Engines do not communicate directly with other system components.**

**Illustration B.3 Destination Ports for Basic Security Engine Communications**



\*Single engines with “Node-initiated Contact to Management Server” selected.

**Illustration B.4 Default Destination Ports for Security Engine Service Communications**



\* Engines in the Firewall/VPN role.

The table below lists all default ports the Security Engines use 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 Security Engine and Master Engine Default Ports**

<b>Listening Host</b>	<b>Port/Protocol</b>	<b>Contacting Hosts</b>	<b>Service Description</b>	<b>Service Element Name</b>
Anti-virus signature server	80/TCP	Firewall	Anti-virus signature update service.	HTTP
Authentication Server	8925-8929/ TCP	Firewall, Master Engine	User directory and authentication services.	LDAP (TCP), RADIUS (Authentication)
BrightCloud Server	2316/TCP	Firewall, Layer 2 Firewall, IPS, Master Engine	BrightCloud URL filtering update service.	BrightCloud update
DHCP server	67/UDP	Firewall	Relayed DHCP requests and requests from a firewall that uses dynamic IP address.	BOOTPS (UDP)
DNS server	53/UDP, 53/TCP	Firewall, Master Engine	Dynamic DNS updates.	DNS (TCP)
Firewall	67/UDP	Any	DHCP relay on firewall engine.	BOOTPS (UDP)
Firewall	68/UDP	DHCP server	Replies to DHCP requests.	BOOTPC (UDP)
Firewall, Master Engine	500/UDP	VPN clients, VPN gateways	VPN negotiations, VPN traffic.	ISAKMP (UDP)
Firewall, Master Engine	636/TCP	Management Server	Internal user database replication.	LDAPS (TCP)
Firewall, Master Engine	2543/TCP	Any	User authentication (Telnet) for Access rules.	SG User Authentication
Firewall	2746/UDP	McAfee VPN gateways	UDP encapsulated VPN traffic (engine versions 5.1 and lower).	SG UDP Encapsulation
Firewall, Master Engine	4500/UDP	VPN client, VPN gateways	VPN traffic using NAT-traversal.	NAT-T
Firewall Cluster Node, Master Engine cluster node	3000-3001/ UDP 3002-3003, 3010/TCP	Firewall Cluster Node, Master Engine cluster node	Heartbeat and state synchronization between clustered Firewalls.	SG State Sync (Multicast), SG State Sync (Unicast), SG Data Sync
Firewall, Layer 2 Firewall, IPS, Master Engine	4950/TCP	Management Server	Remote upgrade.	SG Remote Upgrade

**Table B.2 Security Engine and Master Engine Default Ports (Continued)**

<b>Listening Host</b>	<b>Port/Protocol</b>	<b>Contacting Hosts</b>	<b>Service Description</b>	<b>Service Element Name</b>
Firewall, Layer 2 Firewall, IPS, Master Engine	4987/TCP	Management Server	Management Server commands and policy upload.	SG Commands
Firewall, Layer 2 Firewall, IPS	8888/TCP	Management Server	Connection monitoring for engine versions 5.1 and lower.	SG Legacy Monitoring
Firewall, Layer 2 Firewall, IPS, Master Engine	15000/TCP	Management Server, Log Server	Blacklist entries.	SG Blacklisting
Firewall, Layer 2 Firewall, IPS, Master Engine	161/UDP	SNMP server	SNMP monitoring.	SNMP (UDP)
IPS Cluster Node	3000-3001/ UDP 3002-3003, 3010/TCP	IPS Cluster Node	Heartbeat and state synchronization between clustered IPS engines.	SG State Sync (Multicast), SG State Sync (Unicast), SG Data Sync
LDAP server	389/TCP	Firewall, Master Engine	External LDAP queries, including StartTLS connections.	LDAP (TCP)
Layer 2 Firewall Cluster Node	3000-3001/ UDP 3002-3003, 3010/TCP	Layer 2 Firewall Cluster Node	Heartbeat and state synchronization between clustered Layer 2 Firewalls.	SG State Sync (Multicast), SG State Sync (Unicast), SG Data Sync
Log Server	3020/TCP	Firewall, Layer 2 Firewall, IPS, Master Engine	Log and alert messages; monitoring of blacklists, connections, status, and statistics.	SG Log
Management Server	3021/TCP	Firewall, Layer 2 Firewall, IPS, Master Engine	System communications certificate request/renewal (initial contact).	SG Initial Contact
Management Server	3023/TCP	Firewall, Layer 2 Firewall, IPS, Master Engine	Monitoring (status) connection.	SG Status Monitoring
Management Server	8906/TCP	Firewall, Layer 2 Firewall, IPS	Management connection for single engines with "Node-Initiated Contact to Management Server" selected.	SG Dynamic Control
RADIUS server	1812, 1645/ UDP	Firewall, Master Engine	RADIUS authentication requests.	RADIUS (Authentication), RADIUS (Old)

**Table B.2 Security Engine and Master Engine Default Ports (Continued)**

<b>Listening Host</b>	<b>Port/Protocol</b>	<b>Contacting Hosts</b>	<b>Service Description</b>	<b>Service Element Name</b>
RPC server	111/UDP, 111/TCP	Firewall, Master Engine	RPC number resolve.	SUNRPC (UDP), Sun RPC (TCP)
Server Pool Monitoring Agents	7777/UDP	Firewall, Master Engine	Polls to the servers' Server Pool Monitoring Agents for availability and load information.	SG Server Pool Monitoring
SNMP server	162/UDP	Firewall, Layer 2 Firewall, IPS, Master Engine	SNMP traps from the engine.	SNMP Trap (UDP)
TACACS+ server	49/TCP	Firewall, Master Engine	TACACS+ authentication requests.	TACACS (TCP)
User Agent	16661/TCP	Firewall, Master Engine	Queries for matching Users and User Groups with IP addresses.	SG Engine to User Agent
VPN gateways	500/UDP, 2746/UDP (McAfee gateways only), or 4500 UDP	Firewall, Master Engine	VPN traffic. Ports 2746 and 4500 may be used depending on encapsulation options.	ISAKMP (UDP)



## APPENDIX C

# PREDEFINED ALIASES

This appendix lists the predefined Aliases in the SMC. The predefined Aliases are used in the default policies. Some of them may be useful when you create your own rules.

The following sections are included:

- ▶ [Predefined User Aliases](#) (page 338)
- ▶ [System Aliases](#) (page 338)

## Predefined User Aliases

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

**Table C.1 System-defined User Aliases**

Predefined 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 automatically created non-editable Aliases. 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 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)**

<b>System Alias</b>	<b>Description</b>
\$\$ 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:

- ▶ [Correlation Context Parameters](#) (page 342)
- ▶ [Regular Expression Parameter](#) (page 344)
- ▶ [Other Context Parameters](#) (page 344)

# Correlation Context Parameters

## Event Compress

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

Table D.1 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.2 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.

Table D.2 Event Compress Parameters (Continued)

Field	Option (if any)	Explanation
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.3 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 Policy but does not actually group the matching events into one. All the individual events are still available for further inspection, even though they have already triggered a response.
	No	Makes the Correlation Situation group the matching events together, so that only the response defined in the Inspection Policy 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 grouped.
Continuous Responses	Yes	Makes the Security Engine or Log Server respond as defined in the Inspection Policy to each occurrence of the defined event within the selected Time Window.
	No	Makes the Security Engine or Log Server 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.4 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.5 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 Policy 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 in the Inspection Policy 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 345).

## 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 Security Engine Configuration view).



## APPENDIX E

# REGULAR EXPRESSION SYNTAX

This section introduces SMC regular expression syntax. Regular expressions are used in Situations for matching network traffic. Situations are used in the Inspection rules on Security Engines.

The following sections are included:

- ▶ [SMC Regular Expression Syntax](#) (page 346)
- ▶ [Special Character Sequences](#) (page 348)
- ▶ [Pattern-Matching Modifiers](#) (page 349)
- ▶ [Variable Expression Evaluation](#) (page 351)
- ▶ [System Variables](#) (page 355)
- ▶ [Independent Subexpressions](#) (page 356)
- ▶ [Parallel Matching Groups](#) (page 357)
- ▶ [Tips for Working With Regular Expressions](#) (page 357)

# SMC Regular Expression Syntax

A regular expression is a sequence of characters that defines a matching pattern. These patterns are used for matching byte sequences in network traffic. The expression matching always starts from the beginning of the traffic stream, defined by the associated Situation Context. Depending on the context, this may mean the beginning of a TCP stream, the beginning of a UDP packet, or a protocol-specific field or header, such as the beginning of an HTTP request header or the beginning of an HTTP Request URI.

A regular expression consists of one or more branches that are separated by a logical OR symbol “|”. A Situation match occurs if any of the branches matches the traffic stream.

### Illustration E.1 Example: Regular Expression Matching

```
# This regular expression matches if any of the following patterns are seen
# at the beginning of the traffic stream: "aaa", "bbb", "ccc"
aaa|bbb|ccc
```

The basic sequences that can be used in an SMC regular expression are listed in the table below:

Table E.1 SMC Regular Expression Syntax

Sequence	Description	Example
<char>	Matches only the defined characters.	“2”, “A”, “foo” match exactly to the defined characters: “2”, “A”, and “foo” respectively.
• (dot)	matches any character, including the null character \x00 and a missing character. Matches also other than printable characters, such as the linefeed.  A missing character is a special character used by the engine to represent characters missing from a TCP connection. For example, in capture mode, the engine may not see all the traffic of a TCP connection.	“.” matches any single character or byte.
\x<hex>	Matches the hexadecimal byte value ranging from \x00 to \xFF.	“\x4d” matches hexadecimal value “4d” which represents the decimal value 77 and the ASCII character “M”.
[ <char> ]	Matches any single character in the list.	“[ 15aB]” matches when any of the characters “1”, “5”, “a”, or “B” in the matching location of the inspected string.
[ ^<char> ]	Matches any single character that is not on the list.	“[ ^aBc]” matches if none of the characters “a”, “B”, or “c” is present in the matching location of the inspected string.

Table E.1 SMC Regular Expression Syntax (Continued)

Sequence	Description	Example
[<char1>-<char2>]	Matches all the characters ranging from <char1> to <char2>, these two characters included.	"[a-f]" matches any character within the range from "a" to "f", with "a" and "f" included.
\<char>	Used for escaping special metacharacters to be interpreted as normal characters. The metacharacters are: \   ) ( [ ^ - * + ? . #	"\[ " matches the "[ " character instead of interpreting it as the regular expression class metacharacter.
#<text>	Anything starting with "# " up to the linefeed (\x0a) or the carriage return (\x0d) character is considered as a comment and not used in the matching process.	"# my comment." is not used in the matching process.
(<expr1> <expr2>)	Matches if either the expression <expr1> or <expr2> matches.	"a(bc de)" matches "abc" and "ade".

Illustration E.2 Example Regular Expression

```
# This regular expression matches any of the following strings:
# "login.php", "login1.php", "login2.php", "login_internal.php"
# Note: to match the "." character, the character must be escaped in the
# regular expression by prefixing the character with "\"
login\.php|login[12]\.php|login_internal\.php

# Alternatively, the branches of the above regular expression can be
# combined into one single branch as shown below
login([123]|_internal)?\.php
```

It is also possible to indicate repeated, consecutive characters or regular expressions using quantifiers. The quantifiers available in SMC regular expression syntax are listed in the table below:

Table E.2 SMC Regular Expression Quantifiers

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

The quantifiers always apply only to the single previous character (or special character sequence, see [Table E.3](#)), unless otherwise indicated by parentheses. For example, the regular expression “login\*” matches “logi”, “login” or “loginnnn”, whereas the regular expression “(login)\*” matches the empty string “”, “login” or “loginloginlogin”.

As the matching of a regular expression is always started from the beginning of the traffic stream, “. \*” (any character zero or more times) is often needed when writing SMC regular expressions. For example, the regular expression “. \*/etc/passwd” searches for the string “/etc/passwd” anywhere in the traffic stream.



**Note** – Use the wildcard characters ‘\*’ and ‘+’, as well as ‘<expr>{n,m}’ (where m has a large value) with care. If used in the middle of a regular expression, they may result in an expression that has a very large number of matching states and that is too complex for efficient use. It is recommended to use these wildcards only in the beginning of a branch.

## Special Character Sequences

Printable characters, such as “a” or “b”, are defined by simply typing them into a regular expression. In addition, there are some shorthands for common non-printable characters and character classes. Special character sequences are listed in the table below:

**Table E.3 Special Character Sequences**

Sequence	Description
\a	Bell (BEL) = \x07
\t	Horizontal tab (HT) = \x09
\n	Linefeed (LF) = \x0A
\f	Formfeed (FF) = \x0C
\r	Carriage return (CR) = \x0D
\e	Escape (ESC) = \x1B
\ooo	Octal code <i>ooo</i> of the character.
\xHH	Hexadecimal code <i>HH</i> of the character. Case-insensitive. For example, “\xaa” is considered to be the same as “\xAA”.
\c<char>	Control character that corresponds to Ctrl+<char>, where <char> is an upper-case letter.
\w	“word” class character = [A-Za-z0-9_]
\W	Non-“word” class character = [^A-Za-z0-9_]
\s	Whitespace character = [ \t\r\n\f]
\S	Non-whitespace character = [^ \t\r\n\f]
\d	Digit character = [0-9]

Table E.3 Special Character Sequences (Continued)

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

Illustration E.3 Example of Using Special Character Sequences

```
# This fingerprint matches HTTP content for which the length is >= 10000
# The situation context for this regular expression could be either "HTTP
# Request Header Line" or "HTTP Reply Header Line"
Content-Length: \d\d\d\d\d\d

# The regular expression could be also written as shown below
Content-Length: \d{5}
```

## Pattern-Matching Modifiers

The 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.

Illustration E.4 Example of Pattern Matching Modifiers

```
# This fingerprint is identical to the one in Illustration E.3, except for
# the (?i) modifier.
# HTTP Header names are case-insensitive. For this reason, case-
# insensitivity is enabled in this fingerprint.
(?i)Content-Length: \d\d\d\d\d\d
```

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>	“Case insensitive mode” When enabled, case insensitive matching is used for the uppercase and lowercase letters. Thus, a letter matches regardless of its capitalization. When disabled, the letters are matched case-sensitively so that capitalization is taken into account in the matching process.

Table E.4 Pattern-Matching Modifiers (Continued)

Sequence	Description
(?s)	<p>“Single line mode”</p> <p>When enabled, the dot character “.” matches also the null character \x00 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 (?-s) to disable it.</p>
(?x)	<p>“Extended readability mode”</p> <p>When enabled, equals to enabling (?C), (?L), and (?S). 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 (?C), (?L), and (?S). Comments, linefeeds and spaces are used in the matching process.</p>
(?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>
(?<modifiers> :<expr>)	<p>Applies the &lt;modifiers&gt; modifiers only to the expression &lt;expr&gt;. These modifiers are not used in other parts of the regular expression.</p>

# Variable Expression Evaluation

Variable expression evaluation is an extension to regular expression syntax that provides the ability to use variables, parse values from the traffic stream and perform arithmetic operations.

**Table E.5 Variable Expression Syntax**

Sequence	Description
(?(<expression>))	<expression> is one or more comma-separated expressions

**Illustration E.5 Example of Setting a Variable in a Variable Expression**

```
# This regular expression searches for "aaa" anywhere in the traffic stream,  
# and then sets the value of "parameter1" to 1  
  
.*aaa(?[parameter1=1])
```

The default variable size is one bit. Variable size can be changed by appending “@<size>” to the variable name. For example “parameter1@8” is an 8-bit variable. Possible variable sizes, in addition to 1, are 8, 16, 32 and 64 bits. By default variables are visible within a situation context. For example a variable used in a situation with context “HTTP Request URI” is visible to all other situations in that context. Prefixing the variable name with a dollar sign “\$” makes it a connection variable. A connection variable is visible in all the situations contexts for a single TCP connection, for example in both client and server stream contexts.

By default no situation match is created when the end of a variable expression is reached. To create a match when a variable expression is used, the “sid()” function must be called.

**Table E.6 Variable Expression Syntax**

Syntax	Description
<varexpr_a> -> <varexpr_b>	varexpr_b is executed only if varexpr_a is true

The following example shows a typical case where we want to search one string followed by another, for example “aaa” followed by “bbb”. An expression such as “.\*aaa.\*bbb” breaks the guideline of not using “.” in the middle of a regular expression. [Illustration E.6](#) shows how to circumvent this issue using variable expressions.

#### Illustration E.6 Example of Setting and Checking a Variable Value in a Variable Expression

```
# This regular expression searches for "aaa" anywhere in the traffic stream,
# and then sets the value of 'my_var' to 1.
# It also searches for "bbb", and checks whether "aaa" has already been
# seen earlier (i.e. the value of 'my_var' is one). If "aaa" has been seen
# already, a match is created using the "sid()" function.

# The following traffic matches this regular expression: "aaabbb",
# "xxaaaxxxxxbbbx", "aaaxbbb"
# The following traffic does not match this regular expression: "bbbbaaa",
# "aabbxxaaa"
(?x)
.*aaa(?:my_var=1) |
.*bbb(?:my_var==1 -> sid())
```

#### Illustration E.7 Example of Setting and Checking a Variable

```
# This regular expression matches when "login.php" is seen in the traffic
# stream before "user=admin"
# Situation Context e.g. "HTTP Request URI"
(?x)
.*login\.php(?:login_page_seen=1) |
.*user=admin(?:login_page_seen==1 -> sid())
```

All of the arithmetic operations that are available in SMC regular expressions are listed in the table below. Operator precedence is the same as in the C programming language, except that '>' is the 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>.



**Table E.7 Operations on Expression Results (Continued)**

Sequence	Description
<var> *= <expr>	Multiplies the value of <var> by 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 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.
<expr_a>    <expr_b>	Performs OR with if expressions <expr_a> and <expr_b> and returns the result.

Table E.7 Operations on Expression Results (Continued)

Sequence	Description
<var>++, ++<var>	Increase value of variable <var> by one.
<var>--, --<var>	Decrease value of variable <var> by one.
-<expr>	Negate the result of the expression <expr>.
~<expr>	Bitwise invert the result of the expression <expr>.
!<expr>	Perform NOT operation with the expression <expr>.



**Note** – In a regular expression such as “.\*aaa(?[var1=1])”, the starting of the variable expression “(?[var1=1])” is the most time-consuming operation, whereas setting or checking a variable value is a relatively fast operation. For example the regular expression “.\*/(?[parameter1=1])” in an HTTP context would cause the starting of a variable expression after every “/” character in the traffic stream. As this character is very common in HTTP protocol, the regular expression might degrade the system performance.

## Stream Operations

Stream operations can be used to read data from the traffic stream. The value returned by stream operations can either be written to a variable or used directly in an arithmetic operation. The stream operations are listed in the tables below:

Table E.8 ASCII Data Variable Expressions

Sequence	Description
parse_dec(<length>)	Parse ASCII decimal value. <length> is the maximum number of the characters to parse. The actual number of parsed digits is available in the variable \$parse_length@32. If no characters could be parsed, then the variable is set to zero.
parse_hex(<length>)	Parse ASCII hexadecimal value. <length> is the maximum number of the characters to parse. The actual number of parsed digits is available in the variable \$parse_length@32. If no characters could be parsed, then the variable is set to zero.
parse_int(<length>)	Parse ASCII value; parses hexadecimal if the string starts with “0x”, octal if the string starts with zero (“0”) and decimal otherwise. <length> is the maximum number of the characters to parse. The actual number of parsed digits is available in the variable \$parse_length@32. If no characters could be parsed, then the variable is set to zero.
parse_oct(<length>)	Parse ASCII octal value. <length> is the maximum number of the characters to parse. The actual number of parsed digits is available in the variable \$parse_length@32. If no characters could be parsed, then the variable is set to zero.

Table E.9 Miscellaneous Input Stream Operations

Sequence	Description
CRC(<length>)	Calculates a 32-bit CRC value starting from the current byte up to number of bytes specified by <length> parameter. This function can be used as a space optimizer for probabilistically matching against a specific large binary block by its CRC. The CRC used is the 32-bit CRC with polynomial 0x104C11DB7 (used for example in Ethernet).
skip(<length>)	Skip <length> number of bytes.
regex(<regexp>)	Launch independent subexpression. See section “Independent Subexpression” for more information.

Illustration E.8 Example of Parsing a Value From the Traffic Stream

```
# This regular expression finds the string "&parameter1=", parses the
# following three bytes as an ASCII decimal number, and writes the values
# to the "var1@8" variable
# The regular expression matches only if the number is greater than 100
(?x)
.*&parameter1=(?[var1@8=parse_dec(3), var1@8>100 -> sid()])
```

## System Variables

System variables are connection variables whose values are set by the Security Engine. A regular expression can only read the value of these variables. The two most commonly used variables are \$dport and \$offset. The \$dport variable contains the destination port of the connection/datagram, and it is useful especially in “Any Application Protocols” contexts, which receive all traffic (any TCP/UDP port), and in “Unknown Application Protocols” contexts, which receive traffic that does not have a dedicated, protocol-specific context (mostly high TCP/UDP ports). The \$offset variable contains the number of bytes that have been matched since the beginning of the traffic stream. The table below lists all system variables.

Table E.10 System Variables

Sequence	Description
\$major	The major version number of the NGFW engine.
\$minor	The minor version number of the NGFW engine.
\$patch	The patch level number of the NGFW engine.
\$build	The build number of the NGFW engine.
\$dport	The current destination port of the connection. For TCP, \$dport is the destination port of the SYN packet. For UDP, \$dport is the destination port of the first UDP packet sent between two hosts.

Table E.10 System Variables

Sequence	Description
\$offset	The byte that is under inspection when counted from the beginning of the traffic stream. For implementation-specific reasons, the value of \$offset is increased only after the first byte of a traffic stream (after the first byte, the value of \$offset is still 0). For this reason, the value of \$offset is actually the real offset minus one.
\$parse_length@32	Number of digits parsed by last parse_dec(), parse_hex(), parse_oct() or parse_in() expression. See <a href="#">Stream Operations</a> below.

**Illustration E.9 Example of System Variable Use**

```
# This regular expression matches if hexadecimal bytes "0x01", "0x02" and
# "0x03" are seen in port 5000
.*\x01\x02\x03(?:$dport==5000 -> sid())
```

## Independent Subexpressions

Independent subexpressions allow launching another regular expression from inside a variable expression. The function used for starting the subexpression is "regex()". The "cancel" function must always be called after a match in a subexpression. This stops the execution of the subexpression and frees resources. The "cancel" function is always called without parentheses "()" unlike other functions.

Subexpressions are useful for splitting a single complex regular expression into two. For example ".\*&filename=[^&]{256}" breaks the guideline of not using ".\*" or "<expr>{n,m}" with a large m in the middle of a regular expression. The following illustration shows how to circumvent this limitation by using an independent subexpression.

**Illustration E.10 Example of Independent Subexpression Use**

```
# This fingerprint detects an HTTP parameter filename with value longer than
# 256 bytes
(?x)
.*&filename=(?
    regex(
        [^&]{256}(?:[sid()],cancel))
    )
)
```

## Parallel Matching Groups

You can set different regular expressions to be matched in parallel groups within one Situation Context. Normally, manual Situation group definitions are not needed and the engine automatically compiles all your custom Situations in the same group (group 0). Manual group definitions are 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.11** 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.

## Tips for Working With Regular Expressions

- For more examples of regular expressions, you can view the Context tab of the Situation Properties dialog.
- When adding a new Situation to an Inspection rule, it is often useful to select the “Excerpt” logging option. This option includes an excerpt of the traffic that the regular expression matches and also the matching position (“Excerpt position”) in the log entry. This helps in verifying that the regular expression works as expected.
- Freely available tools, such as `wget`, can be used for generating traffic for testing regular expressions.
- If a policy upload fails with an error message such as “Fingerprint compilation failed”, it indicates that a regular expression is too complex. In this case, the regular expression should be modified. For example, use a variable expression or an independent subexpression. If it is not possible to modify the regular expression, the regular expression can be moved to a parallel matching group as explained in [Parallel Matching Groups](#).



## APPENDIX F

# SCHEMA UPDATES FOR EXTERNAL LDAP SERVERS

This section lists the SMC-specific LDAP classes and attributes that you add to the schema of external LDAP servers.

The SMC-specific attribute and class names start with “sg”. The classes are listed in the table below.

**Table F.1 SMC-Specific LDAP Classes**

Class	Description
sggroup	SMC user group
sguser	SMC user account

The SMC-specific attributes are listed in the table below.

**Table F.2 SMC-Specific LDAP Attributes**

Attribute	Related Classes	Description
sgactivation	sguser	Activation date for the user account.
sgauth	sggroup, sguser	Authentication service for the user or group.
sgdelay	sggroup, sguser	Number of days the user account is valid after the activation.
sgexpiration	sguser	Last day when the user account is valid and the user can log in.
sggrouptype	sggroup	Indicates the type of the group: a subtree or discrete group.
sgmember	sggroup	The Distinguished Name (DN) for the user member of this group.

**Table F.2 SMC-Specific LDAP Attributes (Continued)**

Attribute	Related Classes	Description
sgpassword	sguser	MD5 message digest hash of the user password.
sgpresharedkey	sguser	IPsec PreSharedKey for the user account.
sgsubjectaltnames	sguser	IPsec certificate SubjectAltNames for the user account.
sgvirtualip	sggroup, sguser	Virtual IP allocation allowed for the user.

In addition to updating the directory schema, there may be some server-specific requirements. For Netscape and OpenLDAP version 1.2.11 servers, you must configure the following lines to the LDAP server's `slapd.conf` configuration file after stopping the LDAP service:

**Illustration F.1 Additional Configuration for OpenLDAP v1.2.11 and Netscape Server**

```
include /etc/openldap/slapd.at.conf
include /etc/openldap/slapd.oc.conf
include /etc/openldap/sg-schema.conf
schemacheck on
```

For OpenLDAP server versions 2.0 and later, you must configure the following lines to the LDAP server's `slapd.conf` configuration file after stopping the LDAP service:

**Illustration F.2 Additional Configuration for OpenLDAP version 2.0 or later**

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/sg-v3.schema
```



## APPENDIX G

# SNMP TRAPS AND MIBs

Firewall/VPN, IPS, and Layer 2 Firewall engines can send SNMP traps on system events. The traps are configured using SNMP Agent elements. Additionally, Tester entries can be configured to send SNMP traps. The SNMP traps are listed in the table below.

**Table G.1 SNMP Traps for Firewall/VPN, IPS, and Layer 2 Firewalls**

Trap Name	Objects Included	Description
fwPolicyInstall	fwSecurityPolicy	(Firewall and Layer 2 Firewall) Policy was installed on the Firewall engine.
ipsPolicyInstall	ipsSecurityPolicy	(IPS) 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	(Firewall and Layer 2 Firewall) 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	(Firewall and Layer 2 Firewall) Logout on the firewall engine's console or through SSH.

The STONESOFT-SMI-MIB defines the top-level enterprise registrations for the NGFW products in the .iso.org.dod.internet.private.enterprises.stonesoft branch (OID .1.3.6.1.4.1.1369). The NGFW-specific MIB files can be downloaded at <http://www.stonesoft.com/>

The NGFW-specific MIBs are:

- STONESOFT-FIREWALL-MIB: see [Table G.2](#).
- STONESOFT-IPS-MIB: see [Table G.3](#).
- STONESOFT-NETNODE-MIB: see [Table G.4](#).

Security Engines in the Firewall/VPN and Layer 2 Firewall roles support objects in STONESOFT-FIREWALL-MIB. Security Engines in the IPS role support objects in STONESOFT-IPS-MIB. Security Engines in all roles support objects in STONESOFT-NETNODE-MIB.

Security Engines in the Firewall/VPN role also support objects in the following standard MIBs:

- IF-MIB (RFC 2863 and RFC 2233): see [Table G.5](#).
- IP-MIB (RFC 2011): see [Table G.6](#).
- SNMP-USER-BASED-SM-MIB (RFC 3414): see [Table G.7](#).
- SNMPv2 MIB (RFC 3418): see [Table G.8](#).

**Table G.2 STONESOFT-FIREWALL-MIB Objects**

Object Name	Object Description in MIB
fwPolicyTime	The time when the security policy was installed to the Firewall or Layer 2 Firewall
fwSecurityPolicy	Name of the current security policy on the Firewall or Layer 2 Firewall
fwSoftwareVersion	Version string of the Firewall or Layer 2 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 an 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 or Layer 2 Firewall rules
fwIfDroppedPkts	Number of dropped packets by Firewall or Layer 2 Firewall rules
fwIfForwardedPkts	Number of forwarded packets by Firewall or Layer 2 Firewall rules
fwIfLoggedPkts	Number of logged packets by Firewall or Layer 2 Firewall rules

**Table G.2 STONESOFT-FIREWALL-MIB Objects (Continued)**

<b>Object Name</b>	<b>Object Description in MIB</b>
fwIfRejectedPkts	Number of rejected packets by Firewall or Layer 2 Firewall rules
fwIfAccountedPkts	Number of accounted packets by Firewall or Layer 2 Firewall rules
fwIfAcceptedBytes	Number of accepted bytes by Firewall or Layer 2 Firewall rules
fwIfForwardedBytes	Number of forwarded bytes by Firewall or Layer 2 Firewall rules
fwIfDroppedBytes	Number of dropped bytes by Firewall or Layer 2 Firewall rules
fwIfLoggedBytes	Number of logged bytes by Firewall or Layer 2 Firewall rules
fwIfRejectedBytes	Number of rejected bytes by Firewall or Layer 2 Firewall rules
fwIfAccountedBytes	Number of accounted bytes by Firewall or Layer 2 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

**Table G.2 STONESOFT-FIREWALL-MIB Objects (Continued)**

<b>Object Name</b>	<b>Object Description in MIB</b>
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
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)
fwVpnEp4Local	Local IPv4 end-point address
fwVpnEp4Remote	Remote IPv4 end-point address
fwVpnEp4RemoteType	The type of remote VPN end-point (static, dynamic or mobile)
fwVpnEp4ReceivedBytes	Number of received bytes between the end-point pair
fwVpnEp4SentBytes	Number of sent bytes between the end-point pair
fwVpnEp4IpsecSa	Number of currently established IPsec SAs between the end-point pair
fwVpnEp6Local	Local IPv6 end-point address
fwVpnEp6Remote	Remote IPv6 end-point address
fwVpnEp6RemoteType	The type of remote VPN end-point (static, dynamic or mobile)
fwVpnEp6ReceivedBytes	Number of received bytes between the end-point pair
fwVpnEp6SentBytes	Number of sent bytes between the end-point pair
fwVpnEp6IpsecSa	Number of currently established IPsec SAs between the end-point pair
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

**Table G.2 STONESOFT-FIREWALL-MIB Objects (Continued)**

<b>Object Name</b>	<b>Object Description in MIB</b>
adsISynchroSpeedDown	The actual rate at which data is flowing downstream
adsIAttenuationUp	An estimate of the average loop attenuation upstream
adsIAttenuationDown	An estimate of the average loop attenuation downstream
adsINoiseMarginUp	This is a signal-to-noise ratio (SNR) margin for traffic going upstream
adsINoiseMarginDown	This is a signal-to-noise ratio (SNR) margin for traffic going downstream
adsIHecErrorsUp	The total number of header error checksum errors upstream
adsIHecErrorsDown	The total number of header error checksum errors downstream
adsIOcdErrorsUp	The number of out-of-cell delineation errors upstream
adsIOcdErrorsDown	The number of out-of-cell delineation errors downstream
adsILcdErrorsUp	The total of lost-cell-delineation errors upstream
adsILcdErrorsDown	The total of lost-cell-delineation errors downstream
adsIBitErrorsUp	The number of bit errors upstream
adsIBitErrorsDown	The number of bit errors downstream

**Table G.3 STONESOFT-IPS-MIB Objects**

<b>Object Name</b>	<b>Object Description in MIB</b>
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 G.4 STONESOFT-NETNODE-MIB Objects**

<b>Object Name</b>	<b>Object Description in MIB</b>
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

**Table G.4 STONESOFT-NETNODE-MIB Objects (Continued)**

Object Name	Object Description in MIB
nodeTestResult	The most recent result of the nodeTest
nodeTestResultTime	The timestamp of the most recent result of the nodeTest

**Table G.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).
ifAlias	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 or 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 or 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.
ifDescr	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 or software.
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 network 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 network 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>

**Table G.5 IF-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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 network 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 network 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>
ifHCOUcastPkts	<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 network 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 network 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 network management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>

**Table G.5 IF-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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 network 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 network 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>
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 network 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 or 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 or 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>



**Table G.5 IF-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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 network management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>
ifOutErrors	<p>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 network management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.</p>
ifOutOctets	<p>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 network 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 ifHCOutOctets counter's value.</p>
ifOutUcastPkts	<p>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 network 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 ifHCOutUcastPkts counter's value.</p>
ifPhysAddress	<p>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 that do not have such an address (for example, a serial line), this object must contain an octet string of zero length.</p>
ifPromiscuousMode	<p>This object has a value of false(2) if this interface only accepts packets or frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets or 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 or frames by the interface.</p>

**Table G.5 IF-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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 G.6 IP-MIB Supported Objects**

Object Name	Object Description in MIB
icmpInAddrMaskReps	The number of ICMP Address Mask Reply messages received.
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.

**Table G.6 IP-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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.
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.

**Table G.6 IP-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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, for example 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 (for example, 0.0.0.0) and addresses of unsupported Classes (for example, 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).
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.

**Table G.6 IP-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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.
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 G.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 did not 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.

Table G.7 SNMP-USER-BASED-SM-MIB Objects (Continued)

Object Name	Object Description in MIB
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 G.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.

**Table G.8 SNMPv2-MIB Supported Objects (Continued)**

Object Name	Object Description in MIB
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.
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 (for example, repeaters) 2 datalink or subnetwork (for example, bridges) 3 Internet (for example, supports IP) 4 end-to-end (for example, supports TCP) 7 applications (for example, supports 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 H

# MULTICASTING

This appendix describes the general principles of multicasting and how it can be used with CVIs (cluster virtual IP addresses) in Firewall Clusters.



**Note – The Packet Dispatch CVI mode should be used instead of multicast CVIs as it uses unicast and requires no additional switch or router configuration. The other CVI modes are provided mainly for backward compatibility.**

The following sections are included:

- ▶ [The General Features of Multicasting](#) (page 378)
- ▶ [IP Multicasting Overview](#) (page 378)
- ▶ [Internet Group Management Protocol](#) (page 379)
- ▶ [Ethernet Multicasting](#) (page 380)
- ▶ [Multicasting and McAfee Firewalls](#) (page 380)

# The General Features of Multicasting

---

Multicasting differs in certain important respects from unicasting and broadcasting as a transmission technique. A distinction can be made between multicasting traffic at the network layer (based on special class D IP addresses) and at the data link layer (based on multicast MAC addresses). The general differences how multicasting can be distinguished from unicasting and broadcasting are highlighted below.

## Multicasting vs. Unicasting

In unicasting, the transmitted datagrams are intended only for a single host having a unique address. In multicasting, the data is transmitted likewise to a single address (i.e., the multicast group address), but the actual data reaches all the hosts that belong to the group identified by the multicast address in question. This way the data needs only be sent once, and not separately to each host. This naturally saves bandwidth.

## Multicasting vs. Broadcasting

In broadcasting, the data is sent from a host to other hosts within a given network, and consequently, they must all use their resources to process the data. In contrast, in multicasting, the hosts that do not belong to a multicast group will not have to use their resources for multicast data. Moreover, multicasting is not restricted to a single network, and hosts on remote networks may receive IP multicast datagrams provided that they belong to a specific host group, and that there are multicast routers forwarding the traffic. Thus, IP multicasting can in principle be used globally whereas broadcasting is limited to a single network.

## IP Multicasting Overview

---

In the RFC 1112, IP multicasting is defined as the transmission of an IP datagram to a group of hosts identified by a single IP destination address. In addition to this common *multicast group address*, the hosts in the group all have separate and unique unicast addresses. The actual multicast host group may consist of any number of hosts, possibly even located in different networks. The number may vary over time, as hosts can join in and leave from a group at any time. Moreover, a particular host may belong to several groups simultaneously.

The multicast group addresses are class D addresses. They are identified by the high-order initial four bit sequence *1110*. In the dotted decimal notation, the multicast group address range runs from 224.0.0.0 to 239.255.255.255. There are certain special addresses:

- 224.0.0.0 is never assigned
- 224.0.0.1 is assigned to the permanent group of all hosts, including gateways, in the local network.
- 224.0.0.2 is assigned to all local multicast routers

Multicast IP addresses are not allowed to be used as source addresses. A multicast source address implies forging of an IP address.

The multicast groups are either permanent or transient. Permanent groups have administratively assigned IP addresses, while the addresses of the transient multicast groups can be assigned dynamically from the pool of multicast addresses not reserved for permanent groups. The IP

address of an established permanent group will persist even if the group would not have any members at a given time. The transient groups will cease to exist as soon as they no longer have member hosts, and the assigned multicast address is released.

See, for example, <http://www.iana.org/assignments/multicast-addresses> for a list of addresses registered with IANA.

## Multicasting Applications

On the basis of the comparisons above, multicasting may be considered a viable option for many types of transmissions. Multicasting is widely used in local area networks for various purposes. Moreover, multicasting can be used both for receiving a publicly transmitted session on an intranet, or for transmitting an internal communication to a public network (e.g., for announcing a product launch). Multicasting is particularly important solution for bandwidth-intensive applications, such as multimedia. The most typical protocol for multicast traffic is UDP.

Multicasting may be a suitable solution, for example, for the following applications:

- work groups, electronic whiteboards
- video/voice-over-IP conferences
- real-time streaming media (e.g., Internet radio)
- file transfer
- spreading of any information to certain selected destinations.

## Internet Group Management Protocol

---

*Internet Group Management Protocol* (IGMP) is an integral part of Internet Protocol. The IGMP messages are encapsulated in IP datagrams. IGMP is used both between hosts and multicast routers, and between multicast routers. It keeps multicast routers informed of the multicast group memberships on a given local network. Each host supporting multicasting must join the multicast group with the address 224.0.0.1 on each network interface at the initialization time. They shall remain members of this group as long as they are active. With IGMP, the hosts located on a LAN can inform the routers that they want to be able to receive multicast messages from external networks.

### Membership Messages

Multicast routers use IGMP for enquiring periodically which multicast groups have members in the connected local networks. This is carried out by sending *Host Membership Query* messages to the all-hosts address 224.0.0.1. The hosts receiving the query respond by sending *Host Membership Reports* to all neighboring multicast routers.

A host joining a new group will immediately transmit a report, instead of waiting for a query. When a host wishes to stop receiving a multicast transmission, it will send a *Leave Report* message with the destination address 224.0.0.2 to all subnet routers. A router receiving a *Leave Report* message will send in response a *Group Specific Query* to the multicast address in order to check whether there still are hosts in that group. In case no response is received, multicasting to that address will be stopped.

# Ethernet Multicasting

---

So far we have seen how multicasting is implemented at the network layer and how multicast IP addresses differ from other types of IP addresses. In addition, we must also distinguish multicasting at the data link layer where stations are identified, not only by their network level IP addresses, but also by their Media Access Control (MAC) addresses. As opposed to unicast and broadcast addresses, the relation of multicast addressing to IP addressing applies also at this level.

Most local area network (LAN) topologies allow for multicasting by using some kind of group addressing scheme. Some topologies offer better support for multicasting than others. In Ethernet (as defined in IEEE 802.3), all MAC addresses that have the least significant bit of the most significant byte as “1” are multicast addresses. Thus, for example, 01:00:00:00:00:00 and 49:aa:bb:cc:dd:ee are both multicast MAC addresses; while 02:00:00:00:00:00 and fe:fe:fe:fe:fe:fe are not. The devices with a given multicast MAC defined are able to listen to all traffic sent to that particular MAC address.

A specific subset of MAC addresses is reserved for mapping the IP multicasting addresses to data link layer addresses. In Ethernet, the multicast MAC addresses that correspond to multicast IP addresses range from 01:00:5e:00:00:00 to 01:00:5e:7f:ff:ff.

## Multicasting and McAfee Firewalls

---



**Note – The Packet Dispatch CVI mode should be used instead of multicast CVIs as it uses unicast and requires no additional switch or router configuration. The other CVI modes are provided mainly for backward compatibility.**

After distinguishing between network layer multicasting and data link layer multicasting, we can now have a look at how the firewall uses multicasting and unicasting.

When using clustering technology, the clustered firewall nodes share a common *unicast* IP address, which is called a CVI (cluster virtual IP address). This shared IP address is assigned to the node that receives traffic that arrives from the network for distributing and load-balancing between all nodes. Any traffic that has a specific node in the cluster as its final destination (such as management connections) is sent to *NDIs* (node dedicated IP addresses).

CVIs allow the cluster to appear as a single virtual entity to other network devices, rather than a group of individual nodes. Traffic addressed to CVIs is load-balanced between the nodes according to the cluster's load-balancing filters. The load-balancing filters determine which traffic is distributed to which individual nodes. This way, a specific node in a cluster handles all packets in the connection as long as the node stays online.

In addition to the shared unicast IP address, each node must also share a data link layer address (MAC) at the CVI. Only this way will each of the nodes be provided with the exact same traffic. There are different options for the cluster-wide MAC address, and the selection depends on the features of the other connected networking devices, such as switches and hubs. This document will not act as a definitive reference for different types of switch configurations, but it will give an overview of possible considerations when implementing firewall clusters in different types of network environments.

The method can be selected based on the surrounding network devices. Unicast MAC configuration can be used with hubs and with switches that support sending a specified unicast MAC address to several ports at the same time. When a layer2 network is not able to do this, multicast MAC can be used instead. Since send all packets to all ports anyway, unicast MAC mode gives better performance with hubs. However, in large networks with large amounts of traffic, the action of sending packets to all ports can create extra load. In that situation, static MAC address forwarding tables can be used to limit traffic to Cluster multicast MAC to cluster ports only. With switches that do not support static MAC address forwarding tables, IGMP snooping can be used for the same task. With switches, Packet Dispatch mode creates less load to switches than unicast MAC or multicast MAC modes.

The different configuration options are presented below. For further reference on different types of configurations, visit <http://www.stonesoft.com>.

## Unicast MAC

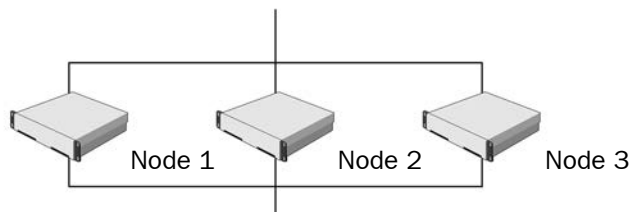
A common unicast MAC can be defined at the CVIs if the cluster is connected to hubs or switches that can forward frames with a unicast destination to multiple ports. This way the network devices will forward the same packets to each of the connected firewall nodes sharing this combination of unicast IP and MAC addresses. This mode is recommended whenever the networking devices support sending packets to a specified unicast MAC address to a predefined set of ports at the same time (as opposed to one port, which is typically the default). Hubs by default support this; however, with switches this is not as frequent, and they usually need additional configuration. With unicast MAC only the switches directly connected to the cluster need special configuration.



**Note – Unlike multicast MAC addresses, there can be only one unicast MAC address defined per Interface ID. Thus, all the NDIs and the unicast CVIs on the same physical interface use the same MAC address.**

In addition to the common CVI IP address, each node may optionally have unique unicast IP addresses defined at the same physical interface as the CVI. These unicast IP addresses are assigned to NDIs (node dedicated IP addresses), and used when an individual node is the end-point of a connection. Since there can only be one unicast MAC address at a given interface, also the node-specific NDI IP addresses are mapped to the common unicast MAC.

**Illustration H.1** exemplifies the IP and MAC address configuration of a cluster's interfaces that are connected to an external network. By default, the CVI of each node share one unicast IP address. The CVI is mapped to a common unicast MAC address. In addition, for each node, an NDI is defined at the same physical interface as the CVI. The NDI IP addresses are unique, but they all are mapped to the same unicast MAC as the CVI IP address, as there can be only one unicast MAC defined for a physical interface. Traffic directed from the Internet to the cluster's external CVI IP address is sent by the connected switch or hub to all nodes, since they all are identified by the same unicast MAC.



Interface (external)	Node 1	Node 2	Node 3
CVI IP Address	212.20.1.254	212.20.1.254	212.20.1.254
CVI Unicast MAC	08:08:08:08:08:08	08:08:08:08:08:08	08:08:08:08:08:08
NDI IP Address	212.20.1.21	212.20.1.22	212.20.1.23
NDI Unicast MAC	08:08:08:08:08:08	08:08:08:08:08:08	08:08:08:08:08:08

## Multicast MAC

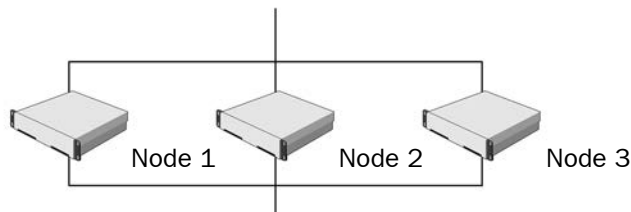
In case it is not feasible to use a switch that works in unicast mode with clusters, a shared multicast MAC may be defined for the cluster nodes. Most switches support this mode, however, not all switches in the same virtual LAN (VLAN) need to be configured. By default, most switches send packets with a multicast MAC address to all ports connected to the same VLAN. If the size of the VLAN is small, this type of flooding is acceptable. However, with larger VLANs performance problems may occur as the device needs to send each packet to each port connected to the same VLAN. In some switches it's possible to prevent this by statically restricting multicast traffic with a given MAC address to some predefined ports only.



**Note – Some networking devices discard ARP replies specifying a multicast MAC. In this case, static ARP entries must be used.**

Illustration H.2 presents an example where a common multicast MAC is configured for all cluster nodes. For instance, if a switch is not able to send packets with the same unicast MAC to multiple ports, this type of configuration may be used. Each node has also a unique unicast MAC address mapped to the corresponding IP addresses defined at the NDIs.

Illustration H.2 CVI with Multicast MAC



Interface (external)	Node 1	Node 2	Node 3
CVI IP Address	212.20.1.254	212.20.1.254	212.20.1.254
CVI Unicast MAC	09:08:08:08:08:08	09:08:08:08:08:08	09:08:08:08:08:08
NDI IP Address	212.20.1.21	212.20.1.22	212.20.1.23
NDI Unicast MAC	04:08:08:08:08:08	06:08:08:08:08:08	08:08:08:08:08:08

## Multicast MAC with IGMP

Internet Group Management Protocol (IGMP) can be used in combination with multicast MAC addresses to avoid flooding with switches that do not support statically defined destinations for multicast. In this mode, switches are configured to send multicast traffic only to the ports from which they have received IGMP *Host Membership Report* messages corresponding to the MAC address in question. Multicast with IGMP must be selected as the mode for the cluster, and IGMP snooping enabled on the switch. For the IGMP messaging, a common multicast *IP address* for the cluster nodes should be specified. The multicast *MAC address* is then computed automatically based on it. Do note, however, that the CVIs are still identified solely by the common *unicast IP address*; the multicast IP address is only used as the source address for the IGMP messages sent to the switch.



**Note –** Some routers that use router redundancy protocols such as HSRP or VRRP listen to all multicast traffic in addition to the routing related traffic. Thus, multicast packets are re-routed to the network. To prevent this, you can either configure the router to send this traffic only to the cluster ports or define the router's access control list (ACL) to drop all incoming packets with the cluster's multicast MAC.





# 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 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**

The Management Server that currently has control of all Domains in an environment that has at least one [Additional Management Server](#).

### **Additional Log Server**

A [Log Server](#) defined as a backup channel for components that primarily send their logs to some other Log Server.

### **Additional Management Server**

A redundant [Management Server](#) that replicates the configuration data from the [Active 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.

### **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 401). See also [Security Association \(SA\)](#) (page 407).

## AH (Authentication Header)

See [Authentication Header \(AH\)](#) (page 388).

## 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 SMC. 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 SMC 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 some [Situation](#) used in a policy or internally in the system, and 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](#).

## 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 legacy IPS device that analyzes the log information from [Sensors](#) according to its policy to find patterns, so that separate log entries can be combined together. See also [Log Server](#), [Security Engine](#).

2) The legacy [Element](#) that represents an Analyzer device in the SMC.

**Antispoofing**

Technique used to protect against malicious packages whose IP header information has been altered. See also [IP Spoofing](#) (page 398).

**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**

An [Action](#) in the Firewall Policy 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 394).

**ARP (Address Resolution Protocol)**

See [Address Resolution Protocol \(ARP\)](#) (page 385).

**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 410) and [Public-key Cryptography](#) (page 405).

**Auditing**

An SMC 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 407).

## **Authentication Server**

A component of the [Security Management Center \(SMC\)](#) that provides authentication services for end-user and [Administrator](#) logins.

## **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.

## **Authorization**

The process of giving someone or 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.

# **B**

## **Balancing Mode**

A [Security Engine](#) 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 410).

## **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 405) and [QoS Policy](#) (page 405).

## **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](#).

## **Border Routing**

Routing of connections between different autonomous systems.

## **BrightCloud**

A [URL 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.

# **C**

## **CA**

See [Certificate Authority \(CA\)](#) (page 389).

## **CAN**

A candidate for a [CVE](#) entry.

## **Capture Interface**

An [IPS Engine](#) or [Layer 2 Firewall](#) interface that can listen to traffic passing in the network, but which is not used for routing traffic through the engine. See also [Inline Interface](#).

## **Category**

A way of organizing elements and policies to display a specific subset at a time when configuring a large SMC environment in the Management Client to make it easier to find the relevant elements. 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 or what they claim to be. This is done through using public/private key pairs and digital signatures. Certificates are used in the SMC for authenticating communications between the SMC 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 391).

## 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 Credentials](#) (page 408).

## Client-to-Gateway VPN

A [Virtual Private Network \(VPN\)](#) between a software client and a [VPN Gateway](#). 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 the SMC, you can cluster Firewalls, IPS engines, and Layer 2 Firewalls 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) A legacy IPS device that has both [Sensor](#) and [Analyzer](#) engines running simultaneously on the same hardware.
- 2) The legacy [Element](#) that represents a Combined Sensor-Analyzer device in the SMC.

See also [IPS Engine](#).

## Connection Tracking

The set of data maintained for a connection. Used for relating incoming packets to existing connections. Connection tracking information also includes information 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 SMC 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 [Log Server](#) and/or the [Security Engines](#) look for when it examines event data produced by engines.

**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 an SMC administrator, as opposed to a ready-made [Default Element](#) created by the SMC.

**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 390).

### Default Element

An [Element](#) that is present in the SMC at installation, or is added to the SMC 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 389).

### 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 406).

### Dispatch Clustering

See [Packet Dispatch](#) (page 403).

### 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 subnet*.

### 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 SMC 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 McAfee 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**

An SMC object that represents 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 SMC, such as a Category. See also [Network Element](#) (page 402).

### **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 387).

## 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 [VPN Gateway](#) 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) An [Element](#) that represents the firewall device in the SMC. Either a [Single Firewall](#) or a [Firewall Cluster](#).
- 2) The device running the [Next Generation Firewall \(NGFW\)](#) engine software in the Firewall/VPN role.

## 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 the [Next Generation Firewall \(NGFW\)](#) engine software in the Firewall/VPN role. This can be a standard server, an engine installed on a virtualization platform, or a McAfee NGFW appliance. Represented by a [Firewall Node](#) in the Management Client.

## Firewall Node

An individual [Firewall Engine](#) in the Management Client, representing a device that runs the [Next Generation Firewall \(NGFW\)](#) engine software in the Firewall/VPN role 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 McAfee Firewall/VPN engines related to VPN performance.

## Gateway-to-Gateway VPN

In the SMC, a [Virtual Private Network \(VPN\)](#) element that 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 390), and [SHA-1](#) (page 408).

## 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.

# I

## 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 397) and [Security Association \(SA\)](#) (page 407).

## 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**

An [IPS Engine](#) or [Layer 2 Firewall](#) interface that combines together two physical interfaces, enabling the traffic to be routed through as if the engine were an extension of the network cable, but allowing the engine 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 in an Inspection Policy that define 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. See also [Action](#) (page 385).

## **Internal Gateway**

A McAfee Firewall/VPN engine that is managed by the same [Management Server](#) on which the [Virtual Private Network \(VPN\)](#) is being configured.

## **Internal Network**

The networks and network resources that the SMC is protecting.

## **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 that includes the information on the IP address of the component it licenses.

## **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 396).

## **IPS Cluster**

Group of two or more IPS engine nodes that work together as if they were a single IPS.

## **IPS Engine**

- 1) An 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.
- 2) The device that runs the [Next Generation Firewall \(NGFW\)](#) engine software in the IPS role. This can be a standard server, an engine installed on a virtualization platform, or a McAfee NGFW appliance.

## **IPS Policy**

The [Security Policy](#) for [IPS Engines](#) that contains the [Action](#) and [Rule](#) definitions that determine how traffic is inspected and how the engine 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 399).

### **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 385).

### **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 407).

### **ISP (Internet Service Provider)**

See [Internet Service Provider \(ISP\)](#) (page 397).

## **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**

### **Layer 2 Firewall**

- 1) A Layer 2 Firewall component that captures all the traffic from a physical network link, handles it according to its policy, and if installed inline, selects which connections are allowed to continue.
- 2) The device that runs the [Next Generation Firewall \(NGFW\)](#) engine software in the Layer 2 Firewall role. This can be a standard server, an engine installed on a virtualization platform, or a McAfee NGFW appliance.

### **License**

Files you install in the SMC to tell the [Management Server](#) that the components you have installed have been legally purchased. You generate the licenses at the License Center web page and install them on 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 SMC 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 SMC.

## **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 SMC log entries.

## **Log Server**

A component of the [Security Management Center \(SMC\)](#) responsible for storing and managing log (and alert) data, and analyzing and correlating events detected by multiple [Security Engines](#).

## **Log Spool**

A temporary storage area in an engine 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

### **Loopback IP address**

An optional type of IP address that allows you to assign IP addresses that do not belong to any directly-connected networks to a [Single Firewall](#), [Firewall Cluster](#), or [Virtual Firewall](#). Loopback IP addresses are not connected to any physical interface and they do not create connectivity to any network.

### **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 386) and [Perfect Forward Secrecy \(PFS\)](#) (page 404).

### **Malware**

Malicious software designed to infiltrate or damage a computer system.

### **Management Bound License**

A [License](#) file for McAfee NGFW engines that is based on information on the Management Server's [Proof of License \(POL\)](#) code.

### **Management Client**

A graphical user interface component that provides the tools for configuring, managing, and monitoring the engines, and other components in the SMC. 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**

An SMC component that stores all information about the configurations of all engines, and other components in the SMC, 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 SMC.

### **Master Engine**

A physical engine device that provides resources for [Virtual Security Engines](#).

### **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

An SMC 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 SMC 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 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 403).

# 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 403).

## NetLink

An [Element](#) used for implementing routing of [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 the SMC.
- 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).

### **Next Generation Firewall (NGFW)**

The device that runs NGFW software in [Firewall](#), [IPS Engine](#), or [Layer 2 Firewall](#) mode. Can be a standard server, an engine installed on a virtualization platform, or a McAfee NGFW appliance. Represented in the SMC by [Security Engine](#) elements.

### **Node**

The representation of an individual [Security Engine](#) in the SMC.

### **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, IPS engines, and Layer 2 Firewalls. Firewall Clusters use a second type of interface, [Cluster Virtual IP Address \(CVI\)](#), for operative traffic. IPS engines and Layer 2 Firewalls 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 409).

**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 397).

**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 402).

**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.

**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 SMC and NGFW software products. Used for generating [License](#) files.

### **Proof of Serial Number (POS)**

Identification code attached to McAfee NGFW 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 engines 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 engine, as well as assisting the engine with content filtering or network address translation tasks. See also [Connection Tracking](#) (page 390).

### **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 385).

### **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 engine in the same manner in which they were fragmented when the engine received them. See also [Virtual Defragmentation](#) (page 413).

**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 392).

**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 the SMC, 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 engine 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 the SMC, the routing table is represented graphically in the Routing view.

## **Rule**

An expression used to define the eventual outcome of packets arriving at the engine, 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 407).

## **Scan**

See [Network Scan](#) (page 402).

## **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.

## **Secret Key Cryptography**

See [Symmetric Encryption](#) (page 410).

## **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 VPN gateways. See also [Authentication Header \(AH\)](#) (page 388).

## **Security Engine**

A type of [Element](#) that represents an [Next Generation Firewall \(NGFW\)](#) engine device in the SMC. See also [Firewall](#), [IPS Engine](#), and [Layer 2 Firewall](#).

## **Security Management Center (SMC)**

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 [Security Engines](#), and to store and manage traffic and system-related data.

## **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 388).

## 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 [Security Engines](#), which then use their installed version of the policies to determine the appropriate action to take regarding packets in the network.

## Sensor

A legacy 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 387)).

## Sensor Cluster

Group of two or more legacy 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 [Security Management Center \(SMC\)](#) (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 Credentials

An element that stores the private key and certificate of an internal server. In TLS inspection, the private key and certificate allow the engine to decrypt TLS traffic for which the internal server is the destination. The certificate can also be used to secure a Web Portal Server's connections using HTTPS or to authenticate an Authentication Server to the client in CHAPv2 RADIUS authentication. See [Client Protection Certificate Authority](#) (page 390).

## 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 398).

## 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 396).

## 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 the SMC.

## 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 407).

## 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 the SMC, 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 Mode**

An operating state of a Security Engine 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 390).

## **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 [Multi-Link](#) environment in the SMC. They represent all possible routes that connect the end-points of the VPN 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 387).

## **Syslog**

A standard protocol for exchanging logs between network components. Defined in RFC 5424.

## T

### **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 [Security Engine](#) 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 [Security Engine](#) where the rule matching occurs is capable of stopping the connection.

### **Tester**

A tool that can automatically run tests on [Next Generation Firewall \(NGFW\)](#) 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 393), [QoS Class](#) (page 405) and [QoS Policy](#) (page 405).

## **Transparent Access Control Mode**

A [Security Engine](#) configuration in which the [IPS Engine](#) or [Layer 2 Firewall](#) 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 413).

## **URL 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 websites that are objectionable or potentially harmful.

## **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 385).

## **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 UTM solution comprises a Firewall, deep packet inspection (IDS), and anti-virus.

## **V**

### **Virtual Adapter**

A component of the Stonesoft IPsec 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 406).

### **Virtual Defragmentation**

A procedure in which incoming packet fragments are collected. The packet is defragmented for processing by the engine, and refragmented before it is transmitted again. See also [Fragmentation](#) (page 395).

### **Virtual Firewall**

A [Virtual Security Engine](#) in the Firewall role.

### **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 VPN 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 Stonesoft IPsec VPN Clients is always assigned by [DHCP \(Dynamic Host Configuration Protocol\)](#).

### **Virtual IPS**

A [Virtual Security Engine](#) in the IPS role.

### **Virtual Layer 2 Firewall**

A [Virtual Security Engine](#) in the Layer 2 Firewall role.

### **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.

### **Virtual Resource**

An element that defines the set of resources on the [Master Engine](#) that are allocated to each [Virtual Security Engine](#).

### **Virtual Security Engine**

Logically-separate engines that run as virtual engine instances on a [Master Engine](#). See also [Virtual Firewall](#), [Virtual IPS](#), and [Virtual Layer 2 Firewall](#).

### **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 Gateway**

A device, typically a firewall, that performs encryption or decryption on [Virtual Private Network \(VPN\)](#) packets sent between [Sites](#) through untrusted 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 the security of some system. Vulnerabilities are attached to [Situations](#) to provide you more information on what has happened when the Situation matches.

## **W**

### **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 [URL Filtering](#).

# INDEX

## Numerics

256-bit security strength, for management connections, 45, 53

## A

- access control by user, 205
- access rules, 101–118
  - action in, 104, 107
  - aliases in, 113
  - authentication in, 104, 109
  - comment in, 104
  - continue action in, 111
  - design issues in, 105
  - destination in, 103, 106
  - domain names in, 114
  - for allowing system communications, 110
  - for CIS redirection, 178
  - for route-based VPN, 300
  - for TLS inspection, 159
  - hit in, 104
  - inspection options in, 126
  - interface matching in, 114
  - logging options in, 104, 109
  - names for, 104
  - QoS classes in, 104
  - rule table for, 103
  - service in, 104, 106
  - source in, 103, 106
  - source VPN in, 104, 110
  - tags in, 104
  - time in, 104, 110
  - users in, 109
  - user-specific, 113
- ACE/Server, 216
- action
  - in access rules, 104, 107
  - in exception rules, 124
- action options
  - anti-spam, 108
  - anti-virus, 108
  - blacklisting, 108
  - connection tracking, 108
  - deep inspection, 108
  - rate-based DoS protection, 108
  - scan detection, 108
  - user response, 108
- activating inspection checks, 126
- active directory servers, 204
  - user agents in, 203
- active-active clustering, 53
- active-passive clustering, 54
- adding users, 205
- ADSL interfaces, 46
- aggregated links
  - for firewall clusters, 57
  - for single firewalls, 45
- AH (authentication header), 264
- aliases, 337
  - in access rules, 113
  - system aliases, 338
  - user aliases, system-defined, 338
- allow action, 107
- anti-relay protection, for spam filtering, 169
- anti-spam, 108, 167–170
  - anti-relay protection in, 169
  - anti-spoofing in, 169
  - DNSBLs (DNS-based blackhole lists), 170
  - sensitivity settings in, 169
  - SPF (Sender Policy Framework), 169
- anti-spam rules
  - content, 170
  - envelope, 170
  - header, 170
- antispoofing, 73–80
  - default elements, 77
  - for spam filtering, 169
  - modifying, 78
- anti-virus, 108, 171–173
  - contexts, 186
  - on clusters, 173
  - with TLS inspection, 160
- applications, 191–194
  - access rules for, 193
  - default elements, 192
- apply blacklist action, 107
- apply IPsec VPN action, 276
- apply VPN action, 107
- authenticating users
  - on the firewall, 207–211
  - with external servers, 213–222
- authentication, 214
  - access rules for, 109, 210, 219
  - ACE/Server, 216
  - federated, 218
  - in access rules, 104
  - in VPNs, 265
  - methods, 217
  - on the firewall, 208
  - one-time password, 217
  - RADIUS, 216
  - RSA SecurID, 216
  - simple password, 217
  - TACACS+, 217
- authentication methods for VPNs, 281
- authentication servers, 218

- user linking for, 205
- authentication services, 219

## B

- bandwidth management, 245–258
  - effects of, 247
  - guarantees for, 249
  - limits for, 249
  - priorities for, 249
- BGP (border gateway protocol), 30
- blacklisting, 195–198
  - in access rules, 108
  - manual, 198
  - monitoring of, 198
  - requests from other components, 197
  - risks of, 196
  - whitelisting, 198
- brightcloud, 164
- browser-based authentication, 210

## C

- category-based URL filtering, 164
- central gateways, 275
- centralized management, 26
- certificate authorities
  - client protection certificate authorities, 158
  - for VPNs, 285, 286
  - trusted certificate authorities, 158
- certificate revocation lists (CRL), for VPNs, 284
- certificates
  - for VPNs, 274, 283
  - in TLS inspection, 156
- CIS (content inspection server), 22
  - defining NAT for, 179
  - redirection to, 175–181
- client gateways in VPNs, 272
- client protection certificate authorities, 158
- client-to-gateway VPNs, *see* VPN
- cluster virtual IP addresses, *see* CVI
- clustering, 29
  - active-active, 53
  - active-passive, 54
  - benefits of, 52
  - firewalls, 51–63
  - modes, 53, 55
- command line tools, 307
- commands
  - for engines, 319
  - for log servers, 308
  - for management servers, 308
- comment
  - in access rules, 104
  - in exception rules, 124
  - in NAT rules, 136
- comments in rules, 97

- components of the SMC, 27
- compress situation context, 185
- connection tracking, 95–97
  - idle timeout in, 96
  - in access rules, 108
  - modes in access rules, 95
- connectionless packet inspection, 95
- contact addresses, 139–141
- contact information, 16
- content inspection server, *see* CIS
- contexts, 184
  - anti-virus, 186
  - protocol-specific, 186
  - system, 187
- continue action, 97, 107, 111, 112, 128
- correlation situations, 185
- count situation context, 185
- CVI (cluster virtual IP address), 54–55, 58

## D

- deep inspection, 21
  - in access rules, 108
- default contact addresses, 141
- default inspection policies, 125
- default policy elements, 89
- demilitarized zone, *see* DMZ
- deploying firewalls, 33–39
- designing
  - access rules, 105
  - inspection policies, 123
- destination
  - in access rules, 103, 106
  - in exception rules, 124
  - in NAT rules, 136, 137
- destination port translation, 135
- DHCP
  - index, 44
  - internal server, 45, 60
  - relay, 89
- differentiated services code point, *see* DSCP
- directory servers, 201–206
  - external, 203
  - users in, 205
- discard action, 107
- DMZ (demilitarized zone), 39
- DNSBLs (DNS-based Blackhole Lists), 170
- documentation available, 15
- domain names, in access rules, 114
- domains
  - for directory servers, 204
  - using master engines with, 69
  - using virtual security engines with, 69
- DoS detection situation context, 186
- DoS protection, rate-based, in access rules, 108
- DSCP, 254



- mark, 250
- match, 250
- rules
  - names for, 250
  - tags in, 250
- dynamic DNS (DDNS), 239
- dynamic IP addresses
  - distributing, 45, 60
  - for single firewall interfaces, 44
- dynamic IP multicast routing, 78
- dynamic netlinks, 226, 228
- dynamic source NAT, 134

## E

- ECDSA certificate authorities, 45, 53
- element-based NAT, 132
  - NAT rules from, 137
- eliminating false positives, 127
- encapsulation in route-based VPN, 300
- encryption algorithms for VPNs, 281
- end-points in policy-based VPNs, 275
- enforce IPsec VPN action, 276
- enforce VPN action, 107
- engines, 28
- ESP (encapsulating security payload), 264
- essential, log level, 109
- event compress, 185
- event correlation, 185
- event count, 185
- event group, 185
- event match, 185
- event sequence, 186
- exception rules, 124, 127
  - action in, 124
  - comment in, 124
  - destination in, 124
  - logging options in, 124
  - names for, 125
  - protocol in, 124
  - severity in, 124
  - situation in, 124
  - source in, 124
  - tags in, 125
  - time in, 124
- excluded server status, 241
- external certificate authorities, 286
- external content inspection, 175–181
- external gateways, 286, 301
- external LDAP, 203
  - classes and attributes in, 203
  - schema files, 359
- external network boundary, 37
- external security gateways, 271, 296
- external user authentication, 214

## F

- false positives, eliminating, 127, 129
- federated authentication, 218
- file detection situation context, 187
- filtering, URLs, 164
- fingerprint of certificates, 316
- fingerprint syntax, 345
- FIPS 140-2 compliant VPNs, 279
- firewall clusters, 29, 51–63
- firewall policies, 83–99
  - for master engines, 68
  - for virtual security engines, 68
  - inspection in, 126
  - installation of, 93
  - packet inspection with, 84
  - policy hierarchy of, 84
  - processing of, 84
  - sub-policies, 89, 92
  - templates, 89
  - types of, 87
  - user responses in, 94
  - validating, 94
- firewall template policies, 91
  - firewall inspection template, 89
  - firewall template, 89
- firewalls
  - authentication on, 21
  - benefits of, 29–31
  - clusters, 51–63
    - aggregated links for, 57
    - clustering modes for, 55
    - communications in, 52
    - CVIs in, 54–55
    - hardware for, 53
    - interfaces for, 57
    - IP address types in, 54
    - loopback IP addresses for, 59
    - manual load balancing for, 61
    - multi-link for, 231
    - NDIs, 54
    - synchronization settings for, 60
  - commands for, 319
  - content screening with, 21
  - deep packet inspection on, 21
  - deployment of, 33–39
  - engines, 28
  - functions of, 21
  - general principles of, 17
  - logging on, 23
  - packet filtering on, 19
  - role of, 18
  - routing configuration for, 74
  - single, 43–49
    - dynamic IP addresses for, 44
    - interface IP addresses for, 46
    - interfaces for, 45

- loopback IP addresses for, 47
  - with multi-link, 230
- stateful inspection on, 20
- technologies of, 19
- weaknesses of, 24

forward IPsec VPN action, 276

forward VPN action, 107

FTP protocol agent, 149

## G

gateway elements, 273, 298

gateway profile elements, 273

gateway settings elements, 273

gateway-to-gateway VPN, *see* VPN

general firewall principles, 17

GOST, 280

GRE protocol agent, 149

group situation context, 185

GTP inspection, 149

GTP protocol agent, 149

guarantees for bandwidth, 249

## H

H.323 protocol agent, 150

hardware requirements, 16

hardware support, 34

heartbeat, 52

high availability, 29, 30

hit

- in access rules, 104
- in NAT rules, 136

HSRP (hot standby routing protocol), 30

HTTP inspection exceptions, 159

HTTP protocol agent, 150

HTTP URL filter, 165

HTTPS inspection, *see* TLS inspection

HTTPS protocol agent, 150

## I

idle timeout, in connection tracking, 96

IEEE 802.1Q VLAN tagging, 46, 57

IGMP (internet group management protocol), 379

IGMP-based multicast forwarding, 78

IKE (internet key exchange), 263

- IKE SA negotiations, 263
- IPsec SA negotiations, 263
- MOBIKE, 263

inbound traffic management, 31, 235–243

inherited rules, 87

inspection, 29

inspection options in access rules, 126

inspection policies, 87, 119–129

- activating inspection checks in, 126
- continue action in, 128

customized high-security inspection policy, 90

design issues in, 123

exceptions in, 124, 127

high-security inspection policy, 90

medium-security inspection policy, 89

no inspection policy, 89

passive termination of, 122

rules tree in, 121

situations in, 90

snort rules in, 128

tuning of, 122

interface matching, 114

interfaces

- ADSL, 46
- for master engines, 66, 67
- for virtual security engines, 68
- IDs, 45, 57
- in firewall clusters, 57
- in routing, 74
- in single firewalls, 45
- modem, 47
- of firewall clusters, 51–63
- of single firewalls, 43–49
- physical, 45, 57
- speed setting for, 251
- SSID, 46
- tunnel, 46, 58
- VLAN, 46, 57
- wireless, 46

internal certificate authorities, 285

internal DHCP server, 45, 60

internal LDAP, 202

internal network boundaries, 38

internal security gateways, 271, 296

IP address spoofing, 74

IPS engines

- commands for, 319

IPsec overview, 262–265

issues in VPNs, 276

## J

jump action, 107

## L

layer 2 firewalls

- commands for, 319

LDAP

- schema updates, 359

LDAP (Lightweight Directory Access Protocol)

- classes and attributes in, 203

- domains, 204

- external directory, 203

- internal directory, 202

- servers, 204

limits for bandwidth, 249

- load balancing, 29
  - with ratio method, 227
  - with round trip time method, 227
- load-balanced clustering, 53
- locations, 139–141
- logging options
  - in access rules, 104, 109, 112
    - alert, 109
    - essential, 109
    - none, 109
    - stored, 109
    - transient, 109
  - in exception rules, 124
- loopback IP addresses
  - virtual security engines, 68

## M

- MAC addresses, for firewall cluster interfaces, 57
- malware, 164
- manual blacklisting, 198
- master engines, 65–71
  - commands for, 319
  - interfaces for, 66, 67
  - policies for, 68, 83–99
  - using with domains, 69
- match situation context, 185
- message digest algorithms for VPNs, 280
- MGCP protocol agent, 150
- MOBIKE, 263
- modem interfaces, 47
- monitoring agents, tester, 241
- MSRPC protocol agent, 150
- multicast MAC, 55
- multicast routing, 78
- multicasting, 377–383
  - ethernet, 380
  - IGMP in, 379
  - membership messages in, 379
  - with McAfee firewalls, 380
- multi-layer inspection, 29, 84
- multi-link, 22, 30, 225–233
  - for firewall clusters, 231
  - for server pools, 237
  - for single firewalls, 230
  - for VPNs, 287
  - in inbound traffic management, 237
  - routing, 76
  - standby netlinks in, 227

## N

- names
  - for access rules, 104
  - for DSCP match/mark rules, 250
  - for exception rules, 125
  - for NAT rules, 136

- for QoS rules, 250
- for rules, 98
- NAT (network address translation), 23, 131–144
  - contact addresses in, 139–141
  - destination port translation in, 135
  - dynamic source translation in, 134
  - element-based NAT, 132
  - for outbound load balancing, 141
  - locations in, 139–141
  - protocol agents with, 147
  - static destination translation in, 134
  - static source translation, 133, 229
- NAT rules, 135
  - comment in, 136
  - design issues in, 137
  - destination in, 136, 137
  - for outbound load balancing, 229
  - from element-based NAT, 137
  - hit in, 136
  - names for, 136
  - NAT cell in, 136, 138, 142
  - service in, 136, 137
  - source in, 136, 137
  - tags in, 136
  - used on cell in, 136, 138
- NDI (node dedicated IP address), 54, 58
- NetBIOS protocol agent, 151
- netlinks, 76, 77, 226
  - dynamic, 226, 228
  - probing settings for, 228
  - standby, 227
  - static, 226, 228
- network address translation, see NAT
- network boundary, 39
- network interfaces, 45, 57
- node dedicated IP addresses, see NDI
- non-decrypted domains, 159

## O

- operating system, 34
- oracle protocol agent, 151
- outbound load balancing, 141
  - NAT rules for, 229
- outbound traffic management, 225–233
  - multi-link in, 227
  - netlinks in, 226

## P

- packet dispatch, 55, 56
- packet filtering, 19
- packet inspection with firewall policy, 84
- passive termination, 122
- perimeter networks, 39
- PFS (perfect forward secrecy), 264
- phishing, 164

- physical interfaces, 45, 57
  - for master engines, 67
  - for virtual security engines, 68
- policies
  - default elements, 89
  - for master engines, 83–99
  - for virtual firewalls, 83–99
  - installation of, 93
  - processing of, 84
  - templates, 89
  - types of, 87
  - user responses in, 94
  - validating, 94
- policy routing, 78
- policy snapshots, 97
- policy-based VPNs, 269–293
  - access rules for, 276
  - end-points in, 275
  - topologies for, 266
- ports, 329
- positioning firewalls, 36
- predefined aliases, 338
- predefined template policies, 105
- priorities for bandwidth, 249
- profiles for VPNs, 272
- protocol agents, 142, 145–154
  - connection handling with, 146
  - FTP, 149
  - GRE, 149
  - GTP, 149
  - H.323, 150
  - HTTP, 150
  - HTTPS, 150
  - in access rules, 148
  - in rules, 112
  - MGCP, 150
  - MSRPC, 150
  - NetBIOS, 151
  - oracle, 151
  - protocol validation with, 146
  - remote shell, 151
  - RTSP, 151
  - SCCP, 151
  - services in firewall, 151
  - SIP, 152
  - SMTP, 152
  - SSH, 152
  - SunRPC, 152
  - TCP, 153
  - TFTP, 153
  - using with CIS, 180
  - with NAT, 147
- protocol independent multicast sparse mode (PIM-SM), 298
- protocol, in exception rules, 124
- protocol-specific contexts, 186

- proxy ARP, 142
- proxy firewalls, 19

## Q

- QoS, 22, 31, 245–258
  - classes, 247, 249
    - in access rules, 104
  - interface settings for, 251
  - policies, 247, 249
    - designing, 253
    - rule table, 249
  - rules
    - names for, 250
    - tags in, 250

## R

- RADIUS, 216
- ratio-based load balancing, 227
- reading DCSP codes, 250
- refuse action, 107
- regular expression syntax, 345
- release notes, 16
- remote shell protocol agent, 151
- requirements for hardware, 16
- risks of blacklisting, 196
- role of firewalls, 18
- round trip time-based load balancing, 227
- route-based VPNs, 295–303
  - access rules for, 300
  - dynamic routing with, 302
  - encapsulation in, 300
  - gateways in, 298
  - in tunnel mode, 301
  - profiles for, 297, 299
  - routing for, 298
  - security gateway types in, 296
  - third-party gateway devices in, 301
  - tunnel interfaces for, 46, 58, 298
  - tunnels in, 299
- routing, 73–80
  - default elements, 77
  - dynamic IP multicast routing, 78
  - IGMP-based multicast forwarding, 78
  - monitoring, 78
  - multi-link, 76
  - netlinks in, 77
  - policy routing, 78
  - static IP multicast routing, 78
- RSA SecurID, 216
- RTSP protocol agent, 151
- rule counter analysis, 94
- rules
  - comments in, 97
  - continue action in, 97, 111, 112, 128
  - inheritance of, 87

- logging options in, 112
- names for, 98
- NAT rules, 131–144
- protocol agents in, 112
- user-specific, 205
- validating, 94

rules tree, 121

## S

SA (security association), 263

satellite gateways, 275

scan detection

- in access rules, 108

scan detection situation context, 186

SCCP protocol agent, 151

security considerations for TLS inspection, 160

Sender Policy Framework, *see* SPF

sequence situation context, 186

server credentials, 158

server pools, 31, 235–243

- dynamic DNS (DDNS) for, 239
- excluded server status, 241
- monitoring agents for, 236, 239, 240
- multi-link for, 237

service

- in access rules, 104, 106
- in NAT rules, 136, 137

services

- for CIS redirection, 178
- in firewall protocol agent, 151

severity, in exception rules, 124

single firewalls, 43–49

- aggregated links for, 45

single point of failure, 29, 52

SIP protocol agent, 152

site elements, 274

situation contexts

- for correlation
  - event compress, 185
  - event count, 185
  - event group, 185
  - event match, 185
  - event sequence, 186
- for DoS detection, 186
- for file detection, 187
- for scan detection, 186
- parameters for, 341–344
  - event compress, 342
  - event count, 342
  - event group, 343
  - event match, 344
  - event sequence, 344

situation types, 184

situations, 183–189

- context definitions in, 185

- context parameters for, 341–344
- contexts for, 184, 185
- importing snort rules, 187
- in exception rules, 124
- in inspection policies, 90
- in inspection rules, 127
- severity of, 187
- tags for, 184
- vulnerabilities for, 184

SMC components, 27

SMTP protocol agent, 152

SNMP

- traps, 361

snort rules, importing, 128

source

- in access rules, 103, 106
- in exception rules, 124
- in NAT rules, 136, 137

source VPN, in access rules, 104, 110, 276

spam filtering, 167–170

SPF (Sender Policy Framework), 169

SPI (security parameter index), 263

SSH protocol agent, 152

SSID interfaces, 46

standby clustering, 54

standby netlinks, 227

state synchronization, 60

- security levels for, 61

stateful inspection, 20, 95

static destination NAT, 134

static IP multicast routing, 78

static netlinks, 226, 228

static source NAT, 133, 229

stored, log level, 109

sub-policies, 87, 89

SunRPC protocol agent, 152

supported platforms, 34

system aliases, 338

system contexts, 187

system requirements, 16

system-defined user aliases, 338

## T

TACACS+, 217

tags, 184

- in access rules, 104
- in DSCP match/mark rules, 250
- in exception rules, 125
- in NAT rules, 136
- in QoS rules, 250

TCP protocol agent, 153

template policies, 87

terminate (passive), 122

tester, 241

TFTP protocol agent, 153

- time
  - in access rules, 104, 110
  - in exception rules, 124
- timed synchronization settings, 60
- TLS inspection, 155–161
  - access rules for, 159
  - anti-virus with, 160
  - certificates in, 156
  - client protection certificate authorities for, 158
  - client protection in, 156
  - exceptions for, 159
  - in firewall properties, 159
  - non-decrypted domains for, 159
  - security considerations for, 160
  - server credentials for, 158
  - server protection in, 156
- TLS matches, 159
- topologies for VPNs, 266
- traffic inspection, 29
- traffic prioritization, 245–258
  - effects of, 247
- transient, log level, 109
- trusted certificate authorities, 158
- tuning inspection, 122
- tunnel interfaces, 46, 58, 298
  - for virtual security engines, 68
- tunnels
  - in policy-based VPNs, 275
  - in VPNs, 262
- typographical conventions, 14

## U

- unicast MAC, 55
- URL filtering, 163–166
  - preparing the firewall, 165
  - user response messages in, 165
- use IPsec VPN action, 107, 276
- used on cell, in NAT rules, 136, 138
- user agent, 205
- user aliases, system-defined, 338
- user authentication, 213–222
  - external, 214
  - interfaces for, 210, 220
  - on the firewall, 207–211
- user databases, 204
  - external, 203
  - internal, 202
- user groups, 205
- user responses, 127
  - in access rules, 108
- users (access rule field), 109
- users, adding, 205
- UTM (unified threat management), 21, 29
  - spam filtering, 167–170
  - virus scanning, 171–173

## V

- validating policies, 94
- virtual firewalls, see virtual security engines
- virtual resources, 66, 67, 69
- virtual security engines, 65–71
  - commands for, 319
  - interfaces for, 68
  - moving to different master engines, 69
  - policies for, 68, 83–99
  - using with domains, 69
- virus scanning, 171–173
  - on clusters, 173
- VLAN (virtual local area network), 46, 57
- VLAN interfaces
  - for master engines, 67
  - for virtual security engines, 68
- VPN (virtual private network), 23, 261–268
  - apply action for, 107
  - authentication in, 265
  - authentication methods for, 281
  - central gateways in, 275
  - certificate revocation lists for, 284
  - certificates for, 274, 283
  - client gateways in, 272
  - client-to-gateway, 270
  - clustering with, 287
  - dynamic IP addresses, as end-points for, 278
  - encryption algorithms for, 281
  - end-points in, 275
  - enforce VPN action for, 107
  - external certificate authorities for, 286
  - FIPS 140-2 mode for, 279
  - forward action for, 107
  - gateway profiles in, 273
  - gateway settings for, 273
  - gateways in, 273
  - gateway-to-gateway, 270, 296
  - GOST mode for, 280
  - IKE for, 263
  - internal certificate authorities for, 285
  - issues in, 276
  - logs for, 278
  - message digest algorithms for, 280
  - multi-link for, 31, 287
  - NAT addresses, as end-points for, 279
  - packet types in, 264
  - PFS in, 264
  - policy-based, 262, 269–293
    - access rules for, 276
    - default elements, 272
    - source VPN for, 276
    - topologies for, 266
    - tunnels in, 275
  - pre-shared key authentication in, 265, 283
  - profiles for, 272, 275
  - route-based, 262, 295–303

- access rules for, 300
- dynamic routing with, 302
- encapsulation in, 300
- gateways in, 298
- profiles for, 297, 299
- routing for, 298
- security gateway types in, 296
- third-party gateway devices in, 301
- tunnel interfaces for, 298
- tunnels in, 299
- SAs for, 263
- satellite gateways in, 275
- security gateway types in, 271
- sites in, 274
- SPI for, 263
- third-party gateway devices in, 286
- transport mode in, 265, 296
- tunnel mode in, 265, 301
  - encryption for, 289
- tunnels in, 262
- use IPsec VPN action for, 107
- VPN elements, 275

VPNs

- user aliases, 338

vulnerabilities, 184

## W

- web filtering, *see* URL filtering
- whitelisting, 198
- wireless interfaces, 46
- writing DSCP codes, 250

## Z

- zones, 114

Copyright © 2014 McAfee, Inc. Do not copy without permission.

McAfee and the McAfee logo are trademarks or registered trademarks of McAfee, Inc. or its subsidiaries in the United States and other countries. Other names and brands may be claimed as the property of others.

