

## **Oracle® Solaris 관리: IP 서비스**

Copyright © 1999, 2012, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록 상표입니다.

본 소프트웨어 혹은 하드웨어와 관련 문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

# 목차

---

머리말 .....	19
<b>제1부 TCP/IP 관리 .....</b>	<b>23</b>
<b>1 네트워크 배치 계획 .....</b>	<b>25</b>
네트워크 계획(작업 맵) .....	25
네트워크 하드웨어 결정 .....	26
네트워크에 대한 IP 주소 지정 형식 결정 .....	27
IPv4 주소 .....	27
DHCP 주소 .....	28
IPv6 주소 .....	28
개인 주소 및 설명서 접두어 .....	28
네트워크의 IP 번호 얻기 .....	29
네트워크의 이름 지정 엔티티 .....	29
호스트 이름 관리 .....	30
이름 서비스 및 디렉토리 서비스 선택 .....	30
서브넷 사용 .....	31
가상 네트워크 배치 .....	32
<b>2 IPv6 주소 사용 시 고려 사항 .....</b>	<b>33</b>
IPv6 계획(작업 맵) .....	33
IPv6 네트워크 토폴로지 시나리오 .....	34
IPv6에 대한 하드웨어 지원 확인 .....	36
IPv6 주소 지정 계획 준비 .....	37
사이트 접두어 획득 .....	37
IPv6 번호 지정 체계 만들기 .....	37
IPv6을 지원하도록 네트워크 서비스 구성 .....	38

▼ IPv6을 지원하도록 네트워크 서비스를 준비하는 방법 .....	39
▼ IPv6을 지원하도록 DNS를 준비하는 방법 .....	40
네트워크에서 터널 사용 계획 .....	40
IPv6 구현에 대한 보안 고려 사항 .....	41
<b>3 IPv4 네트워크 구성 .....</b>	<b>43</b>
네트워크 구성(작업 맵) .....	43
네트워크 구성을 시작하기 전에 .....	44
네트워크의 구성 요소 시스템 구성 .....	45
IPv4 자율 시스템 토폴로지 .....	45
▼ IP 인터페이스 구성 방법 .....	47
시스템 구성 모드 설정 .....	51
IPv4 라우터 구성 .....	56
▼ IPv4 라우터 구성 방법 .....	56
경로 지정 테이블 및 경로 지정 유형 .....	59
멀티홉 호스트 구성 .....	62
단일 인터페이스 시스템에 대한 경로 지정 구성 .....	64
네트워크에 서브넷 추가 .....	67
전송 계층 서비스 모니터 및 수정 .....	69
▼ 모든 수신 TCP 연결의 IP 주소 기록 방법 .....	70
▼ SCTP 프로토콜을 사용하는 서비스를 추가하는 방법 .....	70
▼ TCP 래퍼를 사용하여 TCP 서비스에 대한 액세스를 제어하는 방법 .....	73
<b>4 네트워크에서 IPv6 사용 .....</b>	<b>75</b>
IPv6 인터페이스 구성 .....	75
▼ IPv6에 대해 시스템을 구성하는 방법 .....	76
▼ IPv6 주소 자동 구성을 해제하는 방법 .....	77
IPv6 라우터 구성 .....	78
▼ IPv6 지원 라우터를 구성하는 방법 .....	78
호스트 및 서버에 대해 IPv6 인터페이스 구성 수정 .....	80
인터페이스에 대해 임시 주소 사용 .....	80
IPv6 토큰 구성 .....	83
서버에서 IPv6 지원 인터페이스 관리 .....	85
IPv6용 이름 서비스 지원 구성 .....	86
▼ DNS에 IPv6 주소를 추가하는 방법 .....	86

▼ IPv6 이름 서비스 정보를 표시하는 방법 .....	87
▼ DNS IPv6 PTR 레코드가 올바르게 업데이트되었는지 확인하는 방법 .....	87
▼ NIS를 통해 IPv6 정보를 표시하는 방법 .....	88
<b>5 TCP/IP 네트워크 관리 .....</b>	<b>89</b>
주요 TCP/IP 관리 작업(작업 맵) .....	90
netstat 명령으로 네트워크 상태 모니터링 .....	91
▼ 프로토콜별 통계를 표시하는 방법 .....	91
▼ 전송 프로토콜의 상태를 표시하는 방법 .....	92
▼ 네트워크 인터페이스 상태를 표시하는 방법 .....	93
▼ 소켓 상태를 표시하는 방법 .....	94
▼ 특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법 .....	95
▼ 알려진 경로의 상태를 표시하는 방법 .....	96
ping 명령으로 원격 호스트 확인 .....	97
▼ 원격 호스트가 실행 중인지 확인하는 방법 .....	97
▼ 원격 호스트가 패킷을 삭제하는 중인지 확인하는 방법 .....	98
네트워크 상태 화면 관리 및 기록 .....	99
▼ IP 관련 명령의 화면 출력을 제어하는 방법 .....	99
▼ IPv4 경로 지정 데몬의 작업을 기록하는 방법 .....	100
▼ IPv6 Neighbor Discovery 데몬의 작업을 추적하는 방법 .....	100
traceroute 명령으로 경로 지정 정보 표시 .....	101
▼ 원격 호스트에 대한 경로를 찾는 방법 .....	101
▼ 모든 경로를 추적하는 방법 .....	102
snoop 명령으로 패킷 전송 모니터링 .....	102
▼ 모든 인터페이스의 패킷을 확인하는 방법 .....	103
▼ snoop 출력을 파일로 캡처하는 방법 .....	103
▼ IPv4 서버와 클라이언트 간 패킷을 확인하는 방법 .....	104
▼ IPv6 네트워크 트래픽을 모니터링하는 방법 .....	105
IP 계층 장치를 사용하여 패킷 모니터링 .....	105
기본 주소 선택 관리 .....	108
▼ IPv6 주소 선택 정책 테이블을 관리하는 방법 .....	109
▼ 현재 세션에 대해서만 IPv6 주소 선택 정책 테이블을 수정하는 방법 .....	110
<b>6 IP 터널 구성 .....</b>	<b>111</b>
IP 터널 개요 .....	111

이 Oracle Solaris 릴리스에서 IP 터널 관리 .....	111
터널의 유형 .....	111
결합된 IPv6 및 IPv4 네트워크 환경에서의 터널 .....	112
6to4 터널 .....	113
터널 배치 .....	118
터널 만들기 요구 사항 .....	118
터널 및 IP 인터페이스 요구 사항 .....	118
dladm 명령을 통한 터널 구성 및 관리 .....	119
dladm 하위 명령 .....	119
터널 구성(작업 맵) .....	120
▼ IP 터널을 만들고 구성하는 방법 .....	120
▼ 6to4 터널을 구성하는 방법 .....	124
▼ 6to4 릴레이 라우터에 대한 6to4 터널을 구성하는 방법 .....	126
▼ IP 터널 구성을 수정하는 방법 .....	128
▼ IP 터널 구성을 표시하는 방법 .....	129
▼ IP 터널 등록 정보를 표시하는 방법 .....	129
▼ IP 터널을 삭제하는 방법 .....	130
<b>7 네트워크 문제 해결 .....</b>	<b>131</b>
일반 네트워크 문제 해결 팁 .....	131
기본 진단 검사 실행 .....	131
▼ 기본 네트워크 소프트웨어 검사를 수행하는 방법 .....	132
IPv6 배치 시 발생하는 일반적인 문제 .....	132
IPv4 라우터를 IPv6으로 업그레이드할 수 없음 .....	132
IPv6으로 서비스 업그레이드 후 발생하는 문제 .....	133
현재 ISP가 IPv6을 지원하지 않음 .....	133
6to4 릴레이 라우터로 터널링 시 발생하는 보안 문제 .....	133
<b>8 IPv4 참조 .....</b>	<b>135</b>
네트워크 구성 파일 .....	135
inetd Internet Services Daemon .....	136
name-service/switch SMF 서비스 .....	137
네트워크 데이터베이스에 대한 이름 서비스의 영향 .....	138
Oracle Solaris의 경로 지정 프로토콜 .....	139
RIP(Routing Information Protocol) .....	139

RDISC(ICMP Router Discovery) 프로토콜 .....	139
Oracle Solaris의 경로 지정 프로토콜 표 .....	140
<b>9 IPv6 참조 .....</b>	<b>141</b>
Oracle Solaris IPv6 구현 .....	141
IPv6 구성 파일 .....	141
IPv6 관련 명령 .....	145
IPv6 관련 데몬 .....	149
IPv6 Neighbor Discovery 프로토콜 .....	152
Neighbor Discovery에서 제공하는 ICMP 메시지 .....	153
자동 구성 프로세스 .....	153
이웃 요청 및 연결 불가 .....	155
중복 주소 감지 알고리즘 .....	155
프록시 알림 .....	156
인바운드 로드 균형 조정 .....	156
링크 로컬 주소 변경 .....	156
ARP 및 관련 IPv4 프로토콜과 Neighbor Discovery 비교 .....	156
IPv6 경로 지정 .....	158
라우터 알림 .....	158
Oracle Solaris 이름 서비스에 대한 IPv6 확장 .....	159
IPv6에 대한 DNS 확장 .....	159
이름 서비스 명령에 대한 변경 사항 .....	160
NFS 및 RPC IPv6 지원 .....	160
IPv6 Over ATM 지원 .....	160
<b>제2부 DHCP .....</b>	<b>161</b>
<b>10 DHCP 정보(개요) .....</b>	<b>163</b>
DHCP 프로토콜 정보 .....	163
DHCP 사용 시의 이점 .....	164
DHCP의 작동 방식 .....	165
ISC DHCP 서버 .....	168
레거시 Sun DHCP 서버 .....	168
DHCP 클라이언트 .....	169

<b>11</b>	<b>ISC DHCP 서비스 관리</b> .....	171
	DHCP 명령에 사용자 액세스 설정 .....	171
	▼ DHCP 명령에 사용자 액세스를 부여하는 방법 .....	171
	DHCP 서버 작업 .....	172
	▼ ISC DHCP 서버를 구성하는 방법 .....	172
	▼ DHCP 서비스의 구성을 수정하는 방법 .....	173
<b>12</b>	<b>DHCP 클라이언트 구성 및 관리</b> .....	175
	DHCP 클라이언트 정보 .....	175
	DHCPv6 서버 .....	176
	DHCPv4와 DHCPv6의 차이점 .....	176
	DHCP 관리 모델 .....	176
	프로토콜 세부 정보 .....	177
	논리적 인터페이스 .....	178
	옵션 협상 .....	178
	구성 구문 .....	179
	DHCP 클라이언트 시작 .....	179
	DHCPv6 통신 .....	180
	DHCP 클라이언트 프로토콜이 네트워크 구성 정보를 관리하는 방법 .....	180
	DHCP 클라이언트 종료 .....	182
	DHCP 클라이언트 사용 및 사용 안함 .....	182
	▼ DHCP 클라이언트를 사용으로 설정하는 방법 .....	183
	▼ DHCP 클라이언트를 사용 안함으로 설정하는 방법 .....	183
	DHCP 클라이언트 관리 .....	184
	DHCP 클라이언트와 함께 사용된 <code>ipadm</code> 명령 옵션 .....	184
	DHCP 클라이언트 구성 매개변수 설정 .....	185
	다중 네트워크 인터페이스의 DHCP 클라이언트 시스템 .....	186
	DHCPv4 클라이언트 호스트 이름 .....	186
	▼ DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법 .....	187
	DHCP 클라이언트 시스템 및 이름 서비스 .....	188
	DHCP 클라이언트 이벤트 스크립트 .....	189
<b>13</b>	<b>DHCP 명령 및 파일(참조)</b> .....	193
	DHCP 명령 .....	193
	DHCP 서비스에서 사용된 파일 .....	194



DHCP 서비스에서 사용된 SMF 서비스 .....	196
<b>제3부 IP 보안</b> .....	197
<b>14 IP 보안 아키텍처(개요)</b> .....	199
IPsec 소개 .....	199
IPsec RFC .....	201
IPsec 용어 .....	201
IPsec 패킷 흐름 .....	202
IPsec 보안 연결 .....	205
IPsec에서 키 관리 .....	205
IPsec 보호 방식 .....	206
AH(Authentication Header) .....	206
ESP(Encapsulating Security Payload) .....	207
IPsec의 인증 및 암호화 알고리즘 .....	208
IPsec 보호 정책 .....	209
IPsec의 전송 및 터널 모드 .....	209
VPN(Virtual Private Networks) 및 IPsec .....	211
IPsec 및 NAT 순회 .....	212
IPsec 및 SCTP .....	213
IPsec 및 Oracle Solaris 영역 .....	213
IPsec 및 논리적 도메인 .....	213
IPsec 유틸리티 및 파일 .....	214
<b>15 IPsec 구성(작업)</b> .....	217
IPsec를 사용하여 트래픽 보호 .....	217
▼ IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법 .....	218
▼ IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법 .....	221
▼ IPsec 정책을 표시하는 방법 .....	222
IPsec를 사용하여 VPN 보호 .....	223
터널 모드를 사용하여 IPsec로 VPN을 보호하는 예 .....	223
VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명 .....	225
▼ 터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법 .....	227
IPsec 및 IKE 관리 .....	230

▼ IPsec 키를 수동으로 만드는 방법 .....	231
▼ 네트워크 보안에 대한 역할을 구성하는 방법 .....	232
▼ IPsec 및 IKE 서비스를 관리하는 방법 .....	234
▼ IPsec로 패킷이 보호되는지 확인하는 방법 .....	235
<b>16 IP 보안 아키텍처(참조) .....</b>	<b>237</b>
IPsec 서비스 .....	237
ipseccnf 명령 .....	238
ipseccnf.conf 파일 .....	238
샘플 ipseccnf.conf 파일 .....	238
ipseccnf.conf 및 ipseccnf에 대한 보안 고려 사항 .....	239
ipseccalgs 명령 .....	240
IPsec에 대한 보안 연결 데이터베이스 .....	240
IPsec에서 SA 생성을 위한 유틸리티 .....	241
ipseckey에 대한 보안 고려 사항 .....	241
snoop 명령 및 IPsec .....	242
<b>17 Internet Key Exchange(개요) .....</b>	<b>243</b>
IKE로 키 관리 .....	243
IKE 키 협상 .....	244
IKE 키 용어 .....	244
IKE Phase 1 교환 .....	244
IKE Phase 2 교환 .....	245
IKE 구성 선택 .....	245
IKE와 미리 공유한 키 인증 .....	245
IKE와 공개 키 인증서 .....	246
IKE 유틸리티 및 파일 .....	246
<b>18 IKE 구성(작업) .....</b>	<b>249</b>
IKE 정보 표시 .....	249
▼ 1단계 IKE 교환에 사용 가능한 그룹 및 알고리즘 표시 방법 .....	249
IKE 구성(작업 맵) .....	251
미리 공유한 키로 IKE 구성(작업 맵) .....	251
미리 공유한 키로 IKE 구성 .....	252

▼ 미리 공유한 키로 IKE를 구성하는 방법 .....	252
▼ 새 피어 시스템에 대한 IKE 업데이트 방법 .....	254
공개 키 인증서로 IKE 구성(작업 맵) .....	256
공개 키 인증서로 IKE 구성 .....	257
▼ 자체 서명된 공개 키 인증서로 IKE를 구성하는 방법 .....	257
▼ CA가 서명한 인증서로 IKE를 구성하는 방법 .....	262
▼ 공개 키 인증서를 생성하여 하드웨어에 저장하는 방법 .....	266
▼ 인증서 해지 목록 처리 방법 .....	270
모바일 시스템에 대한 IKE 구성(작업 맵) .....	272
모바일 시스템에 대한 IKE 구성 .....	272
▼ 오프사이트 시스템에 대한 IKE 구성 방법 .....	272
연결된 하드웨어를 찾도록 IKE 구성 .....	279
▼ Sun Crypto Accelerator 6000 보드를 찾도록 IKE를 구성하는 방법 .....	279
<b>19 Internet Key Exchange(참조) .....</b>	<b>281</b>
IKE 서비스 .....	281
IKE 데몬 .....	282
IKE 구성 파일 .....	282
ikeadm 명령 .....	283
IKE 미리 공유한 키 파일 .....	284
IKE 공개 키 데이터베이스 및 명령 .....	284
ikecert tokens 명령 .....	284
ikecert certlocal 명령 .....	285
ikecert certdb 명령 .....	285
ikecert certrldb 명령 .....	286
/etc/inet/ike/publickeys 디렉토리 .....	286
/etc/inet/secret/ike.privatekeys 디렉토리 .....	286
/etc/inet/ike/crls 디렉토리 .....	286
<b>20 Oracle Solaris의 IP 필터(개요) .....</b>	<b>287</b>
IP 필터 소개 .....	287
오픈 소스 IP 필터에 대한 정보 소스 .....	288
IP 필터 패킷 처리 .....	288
IP 필터 사용 지침 .....	290
IP 필터 구성 파일 사용 .....	291

IP 필터 규칙 세트 사용 .....	292
IP 필터의 패킷 필터링 기능 사용 .....	292
IP 필터의 NAT 기능 사용 .....	295
IP 필터의 주소 풀 기능 사용 .....	296
패킷 필터 후크 .....	297
IP 필터용 IPv6 .....	297
IP 필터 매뉴얼 페이지 .....	298
<b>21 IP 필터(작업) .....</b>	<b>301</b>
IP 필터 구성 .....	301
▼ IP 필터를 사용으로 설정하는 방법 .....	302
▼ IP 필터를 다시 사용으로 설정하는 방법 .....	303
▼ 루프백 필터링을 사용으로 설정하는 방법 .....	304
IP 필터 비활성화 및 사용 안함으로 설정 .....	305
▼ 패킷 필터링 비활성화 방법 .....	305
▼ NAT 비활성화 방법 .....	306
▼ 패킷 필터링을 사용 안함으로 설정하는 방법 .....	306
IP 필터 규칙 세트 작업 .....	307
IP 필터에 대한 패킷 필터링 규칙 세트 관리 .....	308
IP 필터에 대한 NAT 규칙 관리 .....	314
IP 필터에 대한 주소 풀 관리 .....	316
IP 필터에 대한 통계 및 정보 표시 .....	318
▼ IP 필터에 대한 상태 테이블 확인 방법 .....	319
▼ IP 필터에 대한 상태 통계 확인 방법 .....	320
▼ IP 필터에 대한 NAT 통계 확인 방법 .....	321
▼ IP 필터에 대한 주소 풀 통계 확인 방법 .....	321
IP 필터 로그 파일 작업 .....	322
▼ IP 필터 로그 파일 설정 방법 .....	322
▼ IP 필터 로그 파일 확인 방법 .....	323
▼ 패킷 로그 파일을 비우는 방법 .....	324
▼ 기록된 패킷을 파일에 저장하는 방법 .....	325
IP 필터 구성 파일 만들기 및 편집 .....	326
▼ IP 필터에 대한 구성 파일을 만드는 방법 .....	326
IP 필터 구성 파일 예 .....	327

<b>제4부 네트워크 성능</b>	333
<b>22 통합된 로드 밸런서 개요</b>	335
ILB 용어	336
ILB의 기능	338
ILB 작동 모드	338
ILB 알고리즘	339
ILB 명령줄 인터페이스	339
ILB 서버 모니터링 기능	340
추가 ILB 기능	341
ILB 프로세스	342
ILB 사용 지침	343
ILB 및 서비스 관리 기능	343
ILB 명령 및 하위 명령	344
<b>23 통합 로드 밸런서 구성(작업)</b>	347
통합 로드 밸런서 설치	347
ILB 사용 및 사용 안함	348
▼ ILB를 사용으로 설정하는 방법	348
▼ ILB를 사용 안함으로 설정하는 방법	349
ILB 구성	349
DSR, Full-NAT, Half-NAT 토폴로지	349
Half-NAT 로드 균형 조정 토폴로지	351
Full-NAT 로드 균형 조정 토폴로지	352
ILB고가용성 구성(능동-수동 모드 전용)	353
DSR 토폴로지를 사용하여 ILB HA 구성	353
Half-NAT 토폴로지를 사용하여 ILB고가용성 구성	355
ILB 구성 하위 명령에 대한 사용자 권한 부여 설정	358
ILB 서버 그룹 관리	359
▼ 서버 그룹을 만드는 방법	359
▼ 서버 그룹을 삭제하는 방법	359
서버 그룹 표시	359
ILB에서 백엔드 서버 관리	360
▼ 서버 그룹에 백엔드 서버를 추가하는 방법	360
▼ 서버 그룹에서 백엔드 서버를 제거하는 방법	361

▼ 백엔드 서버를 다시 사용 또는 사용 안함으로 설정하는 방법 .....	362
ILB에서 건전성 검사 관리 .....	362
건전성 검사 만들기 .....	363
사용자 제공 테스트 세부 정보 .....	363
건전성 검사 삭제 .....	364
건전성 검사 나열 .....	364
건전성 검사 결과 표시 .....	365
ILB 규칙 관리 .....	365
▼ 규칙을 만드는 방법 .....	365
규칙 삭제 .....	366
규칙 나열 .....	366
ILB 통계 표시 .....	367
show-statistics 하위 명령을 사용하여 통계 정보 얻기 .....	367
NAT 연결 테이블 표시 .....	367
세션 지속성 매핑 테이블 표시 .....	368
Import 및 Export 하위 명령 사용 .....	368
 24 Virtual Router Redundancy Protocol(개요) .....	369
VRRP 용어 .....	370
VRRP 아키텍처 개요 .....	370
VRRP 라우터 .....	370
VRRP 프로세스 .....	371
VRRP 제한 사항 .....	373
배타적 IP 영역 지원 .....	373
다른 네트워크 기능과의 상호 작업 .....	374
 25 VRRP 구성(작업) .....	375
VRRP VNIC 만들기 .....	376
vrrpadm 구성 .....	376
vrrpadm create-router 하위 명령 .....	376
vrrpadm modify-router 하위 명령 .....	376
vrrpadm delete-router 하위 명령 .....	377
vrrpadm disable-router 하위 명령 .....	377
vrrpadm enable-router 하위 명령 .....	377
vrrpadm show-router 하위 명령 .....	377

보안 고려 사항 .....	379
<b>26 혼잡 제어 구현 .....</b>	<b>381</b>
네트워크 혼잡 및 혼잡 제어 .....	381
▼ TCP 및 SCTP 네트워크 혼잡 제어를 구현하는 방법 .....	382
<b>제5부 IPQoS(IP Quality of Service) .....</b>	<b>385</b>
<b>27 IPQoS 소개(개요) .....</b>	<b>387</b>
IPQoS 기본 .....	387
차별화 서비스란? .....	387
IPQoS 기능 .....	388
QoS(Quality-of-Service) 이론 및 실제에 대한 추가 정보를 얻을 수 있는 위치 .....	388
IPQoS에서 QoS 제공 .....	389
서비스 단계 계약 구현 .....	390
개별 조직에 대해 QoS 보장 .....	390
QoS 정책 소개 .....	390
IPQoS를 사용하여 네트워크 효율성 향상 .....	391
대역폭이 네트워크 트래픽에 미치는 영향 .....	391
서비스 클래스를 사용하여 트래픽 우선 순위 지정 .....	391
차별화 서비스 모델 .....	392
분류기(ipgpc) 개요 .....	392
측정기(tokenmt 및 tswtclmt) 개요 .....	394
표시기(dscpmk 및 dlcosmk) 개요 .....	394
흐름 계산(flowacct) 개요 .....	395
IPQoS 모듈을 통한 트래픽 흐름 방식 .....	395
IPQoS 사용 네트워크에서 트래픽 전달 .....	397
DS 코드 포인트 .....	397
흡별 동작 .....	397
<b>28 IPQoS 사용 네트워크 계획(작업) .....</b>	<b>401</b>
일반 IPQoS 구성 계획(작업 맵) .....	401
Diffserv 네트워크 토폴로지 계획 .....	402
Diffserv 네트워크에 대한 하드웨어 전략 .....	402

IPQoS 네트워크 토폴로지 .....	402
서비스 품질 정책 계획 .....	405
QoS 정책 계획 지원 .....	405
QoS 정책 계획(작업 맵) .....	406
▼ 네트워크에서 IPQoS를 준비하는 방법 .....	407
▼ QoS 정책에 대한 클래스 정의 방법 .....	407
필터 정의 .....	409
▼ QoS 정책에서 필터를 정의하는 방법 .....	410
▼ 흐름 제어 계획 방법 .....	411
▼ 전달 동작 계획 방법 .....	414
▼ 흐름 계산 계획 방법 .....	416
IPQoS 구성 예 소개 .....	417
IPQoS 토폴로지 .....	417
<b>29 IPQoS 구성 파일 만들기(작업) .....</b>	<b>421</b>
IPQoS 구성 파일에서 QoS 정책 정의(작업 맵) .....	421
QoS 정책을 만들기 위한 도구 .....	422
기본 IPQoS 구성 파일 .....	423
웹 서버에 대한 IPQoS 구성 파일 만들기 .....	423
▼ IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법 .....	425
▼ IPQoS 구성 파일에서 필터를 정의하는 방법 .....	427
▼ IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법 .....	429
▼ IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법 .....	432
▼ 최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법 .....	433
애플리케이션 서버에 대한 IPQoS 구성 파일 만들기 .....	436
▼ 애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법 .....	438
▼ IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법 .....	440
▼ IPQoS 구성 파일에서 흐름 제어를 구성하는 방법 .....	442
라우터에서 차별화 서비스 제공 .....	445
▼ IPQoS 사용 네트워크에서 라우터를 구성하는 방법 .....	445
<b>30 IPQoS 시작 및 유지 관리(작업) .....</b>	<b>447</b>
IPQoS 관리(작업 맵) .....	447
IPQoS 구성 적용 .....	448
▼ IPQoS 커널 모듈에 새 구성을 적용하는 방법 .....	448



▼ 재부트 때마다 IPQoS 구성이 적용되도록 하는 방법 .....	449
IPQoS 메시지에 대한 syslog 로깅 사용 .....	449
▼ 부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법 .....	449
IPQoS 오류 메시지를 사용하여 문제 해결 .....	450
<b>31 흐름 계산 및 통계 수집 사용(작업) .....</b>	<b>455</b>
흐름 계산 설정(작업 맵) .....	455
트래픽 흐름에 대한 정보 기록 .....	456
▼ 흐름 계산 데이터에 대한 파일을 만드는 방법 .....	456
통계 정보 수집 .....	458
<b>32 IPQoS 세부 정보(참조) .....</b>	<b>461</b>
IPQoS 아키텍처 및 Diffserv 모델 .....	461
분류기 모듈 .....	461
측정기 모듈 .....	463
표시기 모듈 .....	466
flowacct 모듈 .....	470
IPQoS 구성 파일 .....	473
action 명령문 .....	474
모듈 정의 .....	475
class 절 .....	475
filter 절 .....	476
params 절 .....	476
ipqosconf 구성 유틸리티 .....	477
용어집 .....	479
색인 .....	487



# 머리말

Oracle Solaris용 Oracle Solaris 관리: IP 서비스를 이용해 주셔서 감사합니다. 이 책은 Oracle Solaris 시스템 관리 정보의 중요한 부분을 다루고 있는 14권으로 구성된 세트의 일부입니다. 이 책은 사용자가 이미 Oracle Solaris를 설치했다고 간주합니다. 따라서 네트워크를 구성하거나 네트워크에 필요한 네트워킹 소프트웨어를 구성할 준비가 되어 있어야 합니다.

주 - 본 Oracle Solaris 릴리스는 프로세서 아키텍처의 SPARC 및 x86 제품군을 사용하는 시스템을 지원합니다. 지원되는 시스템은 [Oracle Solaris OS: Hardware Compatibility Lists](#)를 참조하십시오. 이 설명서에서는 플랫폼 유형에 따른 구현 차이가 있는 경우 이에 대하여 설명합니다.

## 시스템 관리 설명서의 구성

시스템 관리 설명서에서 설명하는 항목 목록은 다음과 같습니다.

설명서 제목	내용
<a href="#">SPARC 플랫폼에서 Oracle Solaris 부트 및 종료</a>	시스템 부트 및 종료, 부트 서비스 관리, 부트 동작 수정, ZFS에서 부트, 부트 아카이브 관리 및 SPARC 플랫폼에서 부트 문제 해결
<a href="#">x86 플랫폼에서 Oracle Solaris 부트 및 종료</a>	시스템 부트 및 종료, 부트 서비스 관리, 부트 동작 수정, ZFS에서 부트, 부트 아카이브 관리 및 x86 플랫폼에서 부트 문제 해결
<a href="#">Oracle Solaris 관리: 일반 작업</a>	Oracle Solaris 명령 사용, 시스템 부트 및 종료, 사용자 계정 및 그룹 관리, 서비스, 하드웨어 오류, 시스템 정보, 시스템 리소스 및 시스템 성능 관리, 소프트웨어 관리, 콘솔 및 터미널 인쇄, 시스템 및 소프트웨어 문제 해결
<a href="#">Oracle Solaris 관리: 장치 및 파일 시스템</a>	이동식 매체, 디스크 및 장치, 파일 시스템, 데이터 백업 및 복원
<a href="#">Oracle Solaris 관리: IP 서비스</a>	TCP/IP 네트워크 관리, IPv4 및 IPv6 주소 관리, DHCP, IPsec, IKE, IP 필터 및 IPQoS

설명서 제목	내용
<b>Oracle Solaris Administration: Naming and Directory Services</b>	DNS, NIS 및 LDAP 명명 규칙 및 디렉토리 서비스(NIS에서 LDAP으로의 전환 포함)
<b>Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화</b>	무선 WiFi를 포함한 자동 및 수동 IP 인터페이스 구성, 브릿지, VLAN, 집계, LLDP 및 IPMP 관리, 가상 NIC 및 리소스 관리
<b>Oracle Solaris 관리: 네트워크 서비스</b>	웹 캐시 서버, 시간 관련 서비스, 네트워크 파일 시스템(NFS 및 Autofs), 메일, SLP, PPP
<b>Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리</b>	응용 프로그램에서 사용 가능한 시스템 리소스가 사용되는 방식을 제어할 수 있는 리소스 관리 기능, 운영 체제 서비스를 가상화하여 응용 프로그램 실행을 위한 분리된 환경을 만드는 Oracle Solaris 영역 소프트웨어 분할 기술 및 Oracle Solaris 11 커널에서 실행되는 Oracle Solaris 10 환경을 호스트하는 Oracle Solaris 10 영역
<b>Oracle Solaris 관리: 보안 서비스</b>	감사, 장치 관리, 파일 보안, BART, Kerberos 서비스, PAM, Solaris 암호화 프레임워크, 권한, 키 관리, 권한, RBAC, SASL, 보안 셸 및 바이러스 검사
<b>Oracle Solaris Administration: SMB and Windows Interoperability</b>	SMB 클라이언트가 SMB 공유를 사용할 수 있도록 Oracle Solaris 시스템을 구성할 수 있는 SMB 서비스, SMB 공유에 액세스할 수 있도록 해주는 SMB 클라이언트, 사용자 및 그룹 ID를 Oracle Solaris 시스템과 Windows 시스템 간에 매핑할 수 있도록 해주는 기본 ID 매핑 서비스
<b>Oracle Solaris 관리: ZFS 파일 시스템</b>	ZFS 저장소 풀 및 파일 시스템 만들기/관리, 스냅샷, 복제, 백업, 액세스 제어 목록(ACL)을 통한 ZFS 파일 보호, 영역이 설치된 Solaris 시스템에서 ZFS 사용, 애플리케이션 볼륨, 문제 해결 및 데이터 복구
<b>Trusted Extensions 구성 및 관리</b>	Trusted Extensions에 대한 시스템 설치, 구성 및 관리
<b>Oracle Solaris 11 보안 지침</b>	Oracle Solaris 시스템 보안 및 보안 기능(예: 영역, ZFS 및 Trusted Extensions) 사용 시나리오
<b>Oracle Solaris 10에서 Oracle Solaris 11로 전환</b>	설치, 장치, 디스크 및 파일 시스템 관리, 소프트웨어 관리, 네트워킹, 시스템 관리, 보안, 가상, 데스크탑 기능, 사용자 계정 관리, 사용자 환경의 영역에서 Oracle Solaris 10에서 Oracle Solaris 11으로 전환하는 작업에 대한 시스템 관리 정보와 예제를 제공합니다.

## Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 전자 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

## 활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체	의미	예
AaBbCc123	명령 및 파일, 디렉토리 이름; 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오.  모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오.  <code>machine_name% you have mail.</code>
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	<code>machine_name% su</code>  Password:
aabbcc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.
AaBbCc123	책 제목, 장, 절	사용자 설명서의 6장을 읽으십시오.  캐시는 로컬로 저장된 복사본입니다.  파일을 저장하면 안 됩니다.  주: 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

## 명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 슈퍼유저 프롬프트가 나와 있습니다. 명령 예제에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 다릅니다.

표 P-2 셸 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	\$
슈퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
슈퍼유저용 C 셸	machine_name#



## 제 1 부

# TCP/IP 관리

이 파트에서는 TCP/IP 네트워크 구성, 관리 및 문제 해결 관련 작업과 개념을 다룹니다.





# 네트워크 배치 계획

이 장에서는 네트워크 설정을 계획할 때 고려해야 할 몇 가지 사항에 대해 간략하게 설명합니다. 이러한 고려 사항은 계획적이며 비용 효율적인 방식으로 네트워크를 배치하는 데 유용합니다. 네트워크 계획에 대한 자세한 내용은 본 설명서에서 다루지 않습니다. 여기서는 일반적인 지침만 제공합니다.

본 설명서에서는 사용자가 기본적인 네트워킹 개념 및 용어에 친숙한 것으로 간주합니다. 이러한 기본적인 개념에 대한 소개는 다음 자료를 참조하십시오.

- TCP/IP 프로토콜 제품군 및 OSI(Open Systems Interconnection) 모델 구현에 대한 개요는 [System Administration Guide: IP Services](#)의 1 장, “Oracle Solaris TCP/IP Protocol Suite (Overview)”을 참조하십시오.
- 이 Oracle Solaris 릴리스에서 TCP/IP 프로토콜 제품군을 구현하는 방법에 대한 간략한 설명은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 1 장](#), “네트워킹 스택 개요”를 참조하십시오.

소개 및 개요에 대한 추가 참조 자료는 다음에 나오는 해당 절에서 제공됩니다.

## 네트워크 계획(작업 맵)

다음 표에서는 네트워크 구성 계획과 관련된 다양한 작업을 나열합니다.

작업	설명	정보
계획된 네트워크 토폴로지의 하드웨어 요구 사항을 식별합니다.	네트워크 사이트에 필요한 장비의 유형을 결정합니다.	26 페이지 “네트워크 하드웨어 결정”  특정 장비 유형에 대한 자세한 내용은 장비 제조업체 설명서를 참조하십시오.

작업	설명	정보
등록된 IP 주소를 사용하고 얻는데 필요한 IP 주소의 유형을 결정합니다.	IPv4 네트워크와 IPv6 네트워크 중 하나만 배치할지 아니면 두 유형의 IP 주소를 모두 사용하는 네트워크를 배치할지 선택합니다. 인터넷의 공용 네트워크와 통신할 고유한 IP 주소를 얻습니다.	27 페이지 “네트워크에 대한 IP 주소 지정 형식 결정” 29 페이지 “네트워크의 IP 번호 얻기”
사용할 이름 서비스와 함께 네트워크의 호스트를 식별할 이름 지정 체계를 결정합니다.	네트워크의 시스템에 지정할 이름 목록을 만들고 NIS와 LDAP, DNS, 로컬 /etc 디렉토리의 네트워크 데이터베이스 중 사용할 데이터베이스를 결정합니다.	30 페이지 “호스트 이름 관리” 30 페이지 “이름 서비스 및 디렉토리 서비스 선택”
필요한 경우 관리 세분화를 설정하고 서브넷 전략을 설계합니다.	사이트에서 관리 세분화를 제공하기 위해 네트워크를 서브넷으로 구분해야 할지 여부를 결정합니다.	31 페이지 “서브넷 사용”
네트워크 설계 시 라우터를 배치할 위치를 결정합니다.	라우터가 필요한 만큼 네트워크가 큰 경우 라우터를 지원하는 네트워크 토폴로지를 만듭니다.	<b>System Administration Guide: IP Services</b> 의 “Planning for Routers on Your Network”
전체 네트워크 구성 체계에 가상 네트워크를 만들지 여부를 결정합니다.	네트워크의 하드웨어 메모리 단위를 줄이기 위해 시스템에 가상 네트워크를 만들어야 할 수도 있습니다.	<b>Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 제III부</b> , “네트워크 가상화 및 리소스 관리”

## 네트워크 하드웨어 결정

지원해야 할 시스템 수에 따라 네트워크 구성 방식이 달라집니다. 한 건물의 한 층에 수십대의 독립형 시스템이 배치되는 작은 규모의 네트워크가 조직에 필요할 수도 있고, 여러 건물에 1,000대 이상의 시스템이 배치되는 네트워크를 설정해야 할 수도 있습니다. 이 설정에 따라 **서브넷**이라는 세분화로 네트워크를 추가로 구분해야 할 수 있습니다.

하드웨어에 대해 결정해야 할 몇 가지 계획 요소는 다음과 같습니다.

- 네트워크 토폴로지, 레이아웃 및 네트워크 하드웨어 연결
- 필요한 서버를 비롯하여 네트워크가 지원할 수 있는 호스트 시스템의 유형 및 수
- 이러한 시스템에 설치할 네트워크 장치
- 사용할 네트워크 매체의 유형(예: 이더넷 등)
- 이 매체를 확장하거나 로컬 네트워크를 외부 네트워크에 연결할 브릿지 또는 라우터가 필요한지 여부

주 - 라우터 작동 방식에 대한 설명은 **System Administration Guide: IP Services**의 “Planning for Routers on Your Network”을 참조하십시오. 브릿지에 대한 개요는 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “브릿징 개요”**를 참조하십시오.

## 네트워크에 대한 IP 주소 지정 형식 결정

네트워크 주소 지정 체계를 계획할 때는 다음 요소를 고려하십시오.

- 사용할 IP 주소의 유형(IPv4 또는 IPv6)
- 네트워크의 잠재적 시스템 수
- 고유한 개별 IP 주소와 함께 여러 네트워크 인터페이스 카드(NIC)를 필요로 하는 멀티홈 또는 라우터 시스템 수
- 네트워크에서 개인 주소를 사용할지 여부
- IPv4 주소 풀을 관리하는 DHCP 서버를 사용할지 여부

다음은 IP 주소 유형을 요약한 것입니다.

### IPv4 주소

이러한 32비트 주소는 TCP/IP에 대한 원래 IP 주소 지정 형식입니다.

클래스 기반 IPv4 주소 지정에 대한 개요는 다음 자료를 참조하십시오.

- **System Administration Guide: IP Services**의 “Designing Your IPv4 Addressing Scheme”
- Internet Protocol DARPA Internet Program Protocol Specification (<http://tools.ietf.org/html/rfc791>)

IETF는 IPv4 주소 부족 및 전역 인터넷 경로 설정표의 제한적인 용량에 대한 중/단기적인 해결책으로 *CIDR(Classless Inter-Domain Routing)* 주소를 개발했습니다.

자세한 내용은 다음 자료를 참조하십시오.

- **System Administration Guide: IP Services**의 “Designing Your CIDR IPv4 Addressing Scheme”
- Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (<http://tools.ietf.org/html/rfc4632>)

다음 표에서는 서브넷을 CIDR 표기법과 점으로 구분된 십진수 형식으로 제공합니다.

표 1-1 CIDR 접두어 및 이와 동등한 십진수

CIDR 네트워크 접두어	동등한 점으로 구분된 십진수 서브넷	사용 가능한 IP 주소
/19	255.255.224.0	8,192
/20	255.255.240.0	4,096
/21	255.255.248.0	2,048
/22	255.255.252.0	1,024
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32

## DHCP 주소

DHCP(Dynamic Host Configuration Protocol) 프로토콜을 통해 시스템은 부트 프로세스의 일부로 DHCP 서버로부터 IP 주소 등의 구성 정보를 수신할 수 있습니다. DHCP 서버는 DHCP 클라이언트에 주소를 지정할 IP 주소의 풀을 유지 관리합니다. DHCP를 사용하는 사이트는 모든 클라이언트에 영구 IP 주소를 할당했을 때 필요한 것보다 작은 IP 주소 풀을 사용할 수 있습니다. DHCP 서비스를 설정하여 사이트의 IP 주소 또는 주소 일부를 관리할 수 있습니다. 자세한 내용은 10 장, “DHCP 정보(개요)”를 참조하십시오.

## IPv6 주소

128비트 IPv6 주소는 IPv4에서 사용할 수 있는 것보다 큰 주소 공간을 제공합니다. CIDR 형식의 IPv4 주소와 마찬가지로 IPv6 주소는 클래스가 없으며 접두어를 사용하여 사이트의 네트워크를 정의하는 주소 일부를 지정합니다.

IPv6 주소에 대한 자세한 내용은 다음 자료를 참조하십시오.

- **System Administration Guide: IP Services**의 “IPv6 Addressing Overview”
- **Internet Protocol, Version 6 (IPv6) Specification** (<http://tools.ietf.org/html/rc2460>)

## 개인 주소 및 설명서 접두어

IANA는 개인 네트워크에 사용하도록 IPv4 주소 블록 및 IPv6 사이트 접두어를 예약했습니다. 이러한 개인 주소는 개인 네트워크 내의 네트워크 트래픽에 사용되며, 설명서에서도 사용됩니다.

다음 표에서는 IPv4 주소 범위와 해당 넷마스크를 나열합니다.

IPv4 주소 범위	넷마스크
10.0.0.0 - 10.255.255.255	10.0.0.0
172.16.0.0 - 172.31.255.255	172.16.0.0
192.168.0.0 - 192.168.255.255	192.168.0.0

IPv6 주소의 경우 접두어 **2001:db8::/32**는 설명서 예에서 특별히 사용되는 특수한 IPv6 접두어입니다. 본 설명서의 예에서는 개인 IPv4 주소와 예약된 IPv6 설명서 접두어를 사용합니다.

## 네트워크의 IP 번호 얻기

IPv4 네트워크는 IPv4 네트워크 번호와 네트워크 마스크(**넷마스크**)의 조합으로 정의됩니다. IPv6 네트워크는 **사이트 접두어** 및 **서브넷 접두어**(서브넷으로 구분된 경우)로 정의됩니다.

개인 네트워크가 인터넷의 외부 네트워크와 통신할 수 있도록 하려면 해당 조직으로부터 네트워크에 대해 등록된 IP 번호를 얻어야 합니다. 이 주소가 IPv4 주소 지정 체계에 대한 네트워크 번호 또는 IPv6 주소 지정 체계에 대한 사이트 접두어로 사용됩니다.

인터넷 서비스 제공업체가 다양한 서비스 레벨을 기반으로 한 가격에 따라 네트워크에 대한 IP 주소를 제공합니다. 여러 ISP를 조사하여 네트워크에 가장 적합한 서비스를 제공하는 ISP를 결정하십시오. 일반적으로 ISP는 기업에 동적으로 할당되는 주소 또는 정적 IP 주소를 제공합니다. IPv4 주소와 IPv6 주소를 모두 제공하는 ISP도 있습니다.

사이트가 ISP인 경우 로케일에 적합한 인터넷 레지스트리(IR)로부터 고객의 IP 주소 블록을 얻습니다. 궁극적으로 IANA(Internet Assigned Numbers Authority)에서 등록된 IP 주소를 전세계의 IR로 위임합니다. 각 IR에는 IR이 제공하는 로케일에 적합한 템플릿과 등록 정보가 있습니다. IANA 및 IR에 대한 자세한 내용은 [IANA's IP Address Service 페이지](http://www.iana.org/ipaddress/ip-addresses.htm) (<http://www.iana.org/ipaddress/ip-addresses.htm>)를 참조하십시오.

## 네트워크의 이름 지정 엔티티

TCP/IP 프로토콜은 IP 주소를 사용하여 네트워크에서 시스템을 찾습니다. 하지만 호스트 이름을 사용하면 IP 주소보다 간편하게 시스템을 식별할 수 있습니다. TCP/IP 프로토콜(및 Oracle Solaris)의 경우 시스템을 고유하게 식별하는 데 IP 주소와 호스트 이름이 모두 필요합니다.

TCP/IP 관점에서 네트워크는 일련의 이름이 지정된 엔티티입니다. 호스트는 이름이 있는 엔티티입니다. 라우터도 이름이 있는 엔티티이며, 네트워크도 이름이 있는 엔티티입니다. 네트워크가 설치된 그룹 또는 부서가 사업부, 지역 또는 회사일 수 있으므로 해당 그룹 또는 부서에도 이름이 지정될 수 있습니다. 이론상 네트워크 식별에 사용될 수 있는 이름의 계층은 거의 제한이 없습니다. 도메인 이름은 **도메인**을 식별합니다.

## 호스트 이름 관리

네트워크를 구성할 시스템에 대한 이름 지정 체계를 계획합니다. 서버로 작동하며 NIC가 여러 개인 시스템의 경우 기본 네트워크 인터페이스의 IP 주소와 연관된 호스트 이름을 하나 이상 제공해야 합니다.

네트워크에 있는 두 시스템이 동일한 호스트 이름을 가질 수 없습니다. 따라서 각 호스트 이름은 각 시스템에 대해 고유해야 합니다. 하지만 고유한 이름이 지정된 호스트 또는 시스템의 IP 주소는 여러 개일 수 있습니다.

네트워크를 계획할 때는 설정 프로세스 중 간편하게 액세스할 수 있도록 IP 주소 및 연관된 호스트 이름 목록을 만드십시오. 이 목록을 통해 모든 호스트 이름이 고유한지 확인할 수 있습니다.

## 이름 서비스 및 디렉토리 서비스 선택

Oracle Solaris에서는 세 가지 유형의 이름 서비스(로컬 파일, NIS 및 DNS) 중에서 선택할 수 있습니다. 이름 서비스는 네트워크의 시스템에 대한 중요한 정보(예: 호스트 이름, IP 주소, 이더넷 주소 등)를 유지 관리합니다. 이름 서비스와 함께, 또는 이름 서비스 대신 LDAP 디렉토리 서비스를 사용할 수도 있습니다. Oracle Solaris의 이름 서비스 소개는 **Oracle Solaris Administration: Naming and Directory Services**의 제I부, “About Naming and Directory Services”를 참조하십시오.

OS 설치 중 서버, 클라이언트 또는 독립형 시스템의 호스트 이름과 IP 주소를 제공합니다. 설치 프로그램이 네트워크를 제공할 때 네트워크 서비스에 사용될 `hosts` 데이터베이스에 이 정보를 추가합니다.

네트워크 데이터베이스의 구성은 중요합니다. 따라서 네트워크 계획 프로세스의 일부로 사용할 이름 서비스를 결정해야 합니다. 또한 이름 서비스 사용 여부 결정에 따라 조직에서 네트워크를 관리 도메인으로 구성할지 여부가 달라집니다.

이름 서비스로 다음 중 하나를 선택할 수 있습니다.

- NIS 또는 DNS — NIS 및 DNS 이름 서비스는 네트워크에 있는 여러 서버의 네트워크 데이터베이스를 유지 관리합니다. **Oracle Solaris Administration: Naming and Directory Services**에서 해당 이름 서비스 및 데이터베이스 구성 방법에 대해 설명합니다. “네임스페이스” 및 “관리 도메인” 개념에 대해서도 자세히 설명합니다.

- 로컬 파일 — NIS, LDAP 또는 DNS를 구현하지 않을 경우 네트워크는 **로컬 파일**을 사용하여 이름 서비스를 제공합니다. “로컬 파일”이라는 용어는 네트워크 데이터베이스에 사용되는 /etc 디렉토리의 일련의 파일을 의미합니다. 본 설명서의 절차에서는 별도로 지정되지 않은 경우 로컬 파일을 이름 서비스로 사용 중인 것으로 간주합니다.

---

주 - 네트워크에 대한 이름 서비스로 로컬 파일을 사용하기로 결정할 경우 나중에 다른 이름 서비스를 설정할 수 있습니다.

---

## 도메인 이름

여러 네트워크는 호스트 및 라우터를 관리 도메인의 계층으로 구성합니다. NIS 또는 DNS 이름 서비스를 사용 중인 경우 조직에 대해 전세계적으로 고유한 도메인 이름을 선택해야 합니다. 도메인 이름이 고유하도록 하려면 InterNIC에 도메인 이름을 등록해야 합니다. DNS를 사용하려는 경우에도 InterNIC에 도메인 이름을 등록해야 합니다.

도메인 이름 구조는 계층 구조입니다. 일반적으로 새 도메인은 기존의 관련 도메인 아래에 배치됩니다. 예를 들어, 자회사의 도메인 이름은 모회사의 도메인 아래에 배치될 수 있습니다. 도메인 이름에 다른 관계가 없을 경우 조직에서는 기존의 최상위 레벨 도메인(예: .com, .org, .edu, .gov 등) 중 하나의 바로 아래에 도메인 이름을 배치할 수 있습니다.

# 서브넷 사용

서브넷 사용은 크기 및 제어 문제를 해결하기 위해 관리 세분화를 사용해야 하는 것과 관련이 있습니다. 네트워크에 있는 호스트 및 서버가 많을수록 관리 작업이 복잡해집니다. 관리 세분화를 만들고 서브넷을 사용하면 복잡한 네트워크 관리가 간편해집니다. 네트워크에 대한 관리 세분화를 설정하는 것은 다음 요소에 따라 결정됩니다.

- **네트워크 크기**  
서브넷은 광대한 지역에 세분화가 배치된 비교적 작은 네트워크에서도 유용합니다.
- **사용자 그룹의 공통 요구 사항**  
예를 들어, 한 건물에 국한되며 비교적 적은 수의 시스템을 지원하는 네트워크가 있을 수 있습니다. 이러한 시스템은 여러 하위 네트워크로 구분됩니다. 각 하위 네트워크는 요구 사항이 다른 사용자 그룹을 지원합니다. 이 예에서는 각 서브넷에 대해 관리 세분화를 사용할 수 있습니다.

일반적인 설명은 [System Administration Guide: IP Services](#)의 “What Is Subnetting?”을 참조하십시오.

## 가상 네트워크 배치

이 Oracle Solaris 릴리스에서는 가상 네트워크 카드(VNIC)와 영역을 구성하여 단일 네트워크에 여러 가상 네트워크를 만들 수 있도록 지원합니다. VNIC는 물리적 NIC 위에 만들어지는 네트워크 인터페이스입니다. 영역과 VNIC를 결합하면 많은 수의 물리적 시스템을 포함하는 거대한 데이터 센터를 적은 수의 시스템에 효과적으로 통합할 수 있습니다. 가상 네트워킹에 대한 자세한 내용은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화](#)의 제III부, “네트워크 가상화 및 리소스 관리”를 참조하십시오.



## IPv6 주소 사용 시 고려 사항

이 장은 1 장, “네트워크 배치 계획”의 내용을 보완하기 위해 네트워크에서 IPv6 주소를 사용하기로 결정한 경우 추가 고려 사항에 대해 설명합니다.

IPv4 주소와 IPv6 주소를 모두 사용하도록 계획한 경우 현재 ISP가 두 주소 유형을 모두 지원하는지 확인하십시오. 그렇지 않은 경우 IPv6 주소를 지원하는 별도의 ISP를 찾아야 합니다.

IPv6 개념에 대한 소개 내용은 다음 리소스를 참조하십시오.

- **System Administration Guide: IP Services**의 “IPv6 Addressing Overview”
- **Internet Protocol, Version 6 (IPv6) Specification** (<http://tools.ietf.org/html/rc2460>)

## IPv6 계획(작업 맵)

다음 표는 네트워크에서 IPv6을 구현하려고 계획한 경우 여러 고려 사항을 보여줍니다.

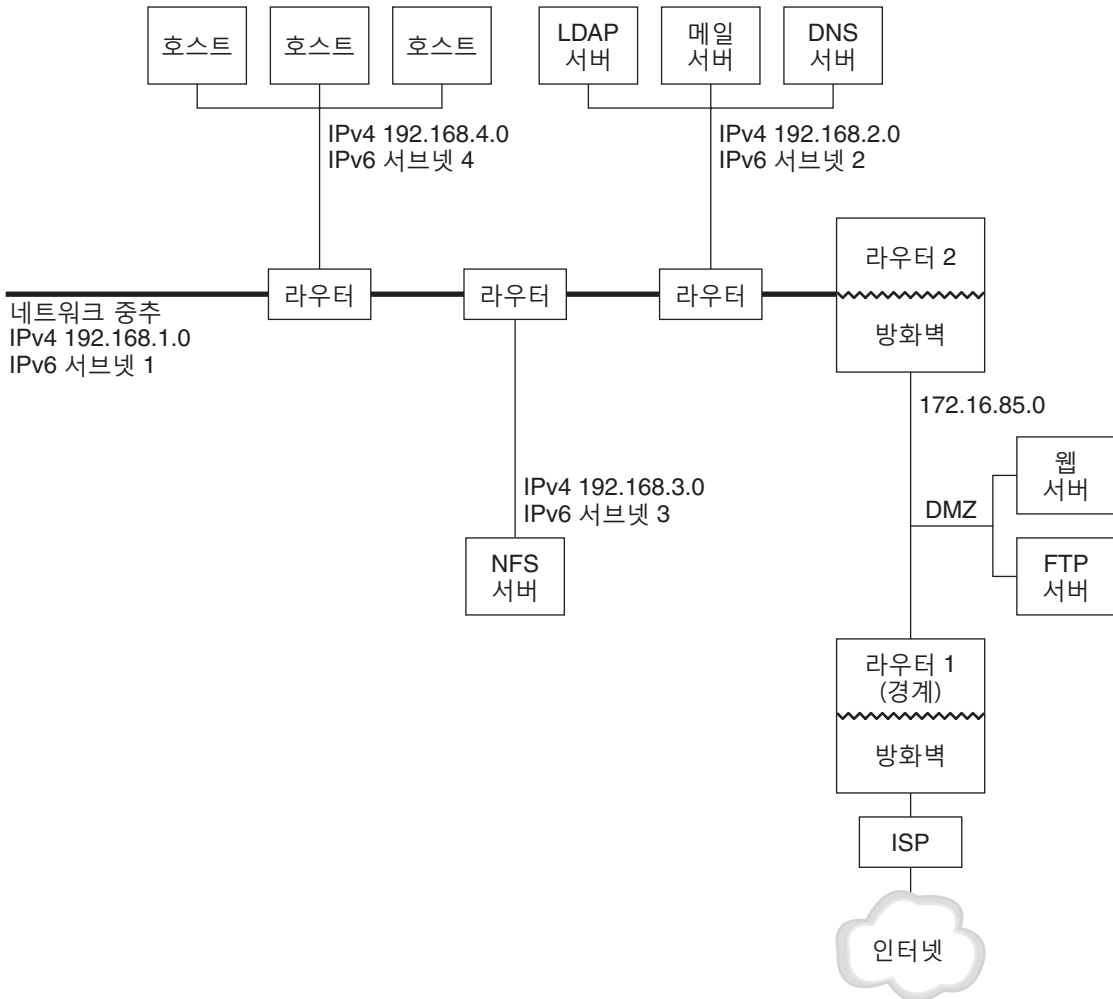
작업	설명	수행 방법
IPv6을 지원하도록 하드웨어 준비	하드웨어를 IPv6으로 업그레이드할 수 있는지 확인합니다.	36 페이지 “IPv6에 대한 하드웨어 지원 확인”
IPv6에서 응용 프로그램을 사용할 수 있는지 확인	IPv6 환경에서 응용 프로그램을 실행할 수 있는지 확인합니다.	38 페이지 “IPv6을 지원하도록 네트워크 서비스 구성”
터널 사용 계획 설계	다른 서브넷 또는 외부 네트워크에 대한 터널을 실행할 라우터를 결정합니다.	40 페이지 “네트워크에서 터널 사용 계획”

작업	설명	수행 방법
네트워크 보안을 설정하고 IPv6 보안 정책을 개발하는 방법 계획	보안을 위해 IPv6을 구성하기 전에 DMZ 및 해당 엔티티에 대한 주소 지정 계획이 필요합니다.  이 릴리스의 IP 필터, IP 보안 아키텍처(IPsec), IKE(Internet Key Exchange) 및 기타 보안 기능 사용과 같은 보안 구현 방식을 결정합니다.	41 페이지 “IPv6 구현에 대한 보안 고려 사항”  제3부
네트워크 시스템에 대한 주소 지정 계획 만들기	IPv6을 구성하기 전에 서버, 라우터 및 호스트에 대한 계획을 세워야 합니다. 이 단계에는 네트워크에 대한 사이트 접두어 얻기 및 IPv6 서브넷 계획(필요한 경우) 작업이 포함됩니다.	37 페이지 “노드에 대한 IPv6 주소 지정 계획 만들기”

## IPv6 네트워크 토폴로지 시나리오

일반적으로 IPv6은 다음 그림에 표시된 것과 같이 IPv4도 사용하는 혼합 네트워크 토폴로지에 사용됩니다. 이후 단원의 IPv6 구성 작업에 대한 설명에서 이 그림을 참조할 수 있습니다.

그림 2-1 IPv6 네트워크 토폴로지 시나리오



엔터프라이즈 네트워크 시나리오는 기존 IPv4 주소를 포함하는 5개의 서브넷으로 구성됩니다. 네트워크 링크는 관리 서브넷과 직접적으로 일치합니다. 네 개의 내부 네트워크는 RFC 1918 스타일의 개인 IPv4 주소로 표시되는데, 이는 IPv4 주소가 없는 경우의 일반적인 솔루션입니다. 이 내부 네트워크의 주소 지정 체계는 다음과 같습니다.

- 서브넷 1은 내부 네트워크 중추 192.168.1입니다.
- 서브넷 2는 LDAP sendmail 및 DNS 서버를 포함하는 내부 네트워크 192.168.2입니다.
- 서브넷 3은 엔터프라이즈의 NFS 서버를 포함하는 내부 네트워크 192.168.3입니다.
- 서브넷 4는 엔터프라이즈 직원에 대한 호스트를 포함하는 내부 네트워크 192.168.4입니다.

외부 공개 네트워크 172.16.85는 회사의 DMZ처럼 작동합니다. 이 네트워크에는 웹 서버, 익명 FTP 서버 및 엔터프라이즈가 외부에 제공하는 기타 리소스가 포함되어 있습니다. 라우터 2는 내부 중추와 구분된 공개 네트워크 172.16.85 및 방화벽을 실행합니다. DMZ의 다른 쪽 끝에 있는 라우터 1은 방화벽을 실행하며 엔터프라이즈의 경계 서버로 사용됩니다.

그림 2-1에서 공개 DMZ의 RFC 1918 전용 주소는 172.16.85입니다. 실제로 공개 DMZ에는 등록된 IPv4 주소가 있습니다. 대부분의 IPv4 사이트는 공개 주소 및 RFC 1918 개인 주소를 결합하여 사용합니다. 그러나 IPv6을 사용할 경우 공개 주소 및 개인 주소의 개념이 달라집니다. IPv6의 주소 공간은 훨씬 더 크므로 개인 네트워크 및 공개 네트워크 모두에서 공개 IPv6 주소를 사용하십시오.

Oracle Solaris 듀얼 프로토콜 스택은 동시 IPv4 및 IPv6 작업을 지원합니다. 네트워크에 IPv6을 배치하는 동안이나 배치한 후 IPv4 관련 작업을 성공적으로 실행할 수 있습니다. 이미 IPv4를 사용 중인 작동 중인 네트워크에 IPv6을 배치할 경우 진행 중인 작업이 중단되지 않습니다.

다음 절에서는 IPv6 구현을 준비할 때 고려해야 할 영역에 대해 설명합니다.

## IPv6에 대한 하드웨어 지원 확인

하드웨어의 다음 클래스와 관련하여 IPv6이 사용 가능한지 제조업체의 설명서를 확인하십시오.

- 라우터
- 방화벽
- 서버
- 스위치

---

주 - 이 설명서의 모든 절차는 장비 특히 라우터를 IPv6으로 업그레이드할 수 있다고 가정합니다.

---

일부 라우터 모델은 IPv6으로 업그레이드할 수 없습니다. 자세한 정보 및 해결 방법은 132 페이지 “IPv4 라우터를 IPv6으로 업그레이드할 수 없음”을 참조하십시오.

Neighbor Discovery 프로토콜을 사용하여 ID를 자동으로 얻는 대신, IPv6 서버의 NIC마다 IPv6 주소의 인터페이스 ID 부분을 수동으로 구성하십시오. 이 방식에서 NIC가 교체될 경우 동일한 인터페이스 ID를 대체 NIC에 적용할 수 있습니다. 다른 ID가 Neighbor Discovery 프로토콜을 통해 자동으로 생성될 경우 서버에서 예상치 않은 동작이 발생할 수 있습니다.

# IPv6 주소 지정 계획 준비

IPv4에서 IPv6으로 전환하는 데 있어 중요한 부분은 주소 지정 계획을 개발하는 것입니다. 이 작업은 다음과 같은 준비 작업과 관련됩니다.

- 37 페이지 “사이트 접두어 획득”
- 37 페이지 “IPv6 번호 지정 체계 만들기”

## 사이트 접두어 획득

IPv6을 구성하기 전에 사이트 접두어를 획득해야 합니다. 사이트 접두어는 IPv6 구현에서 모든 노드에 대한 IPv6 주소를 파생시키는 데 사용됩니다. 사이트 접두어에 대한 소개는 **System Administration Guide: IP Services**의 “Prefixes in IPv6”를 참조하십시오.

IPv6을 지원하는 ISP는 48비트 IPv6 사이트 접두어를 조직에 제공합니다. 현재 ISP가 IPv4만 지원할 경우 IPv4 지원을 위한 현재 ISP를 유지하면서 IPv6 지원을 위한 다른 ISP를 사용할 수 있습니다. 이 경우 여러 해결 방법 중 하나를 사용할 수 있습니다. 자세한 내용은 133 페이지 “현재 ISP가 IPv6을 지원하지 않음”을 참조하십시오.

소속된 조직이 ISP일 경우 적합한 인터넷 레지스트리에서 고객의 사이트 접두어를 획득합니다. 자세한 내용은 **Internet Assigned Numbers Authority (IANA)** (<http://www.iana.org>)를 참조하십시오.

## IPv6 번호 지정 체계 만들기

제안된 IPv6 네트워크가 완전히 새로운 네트워크가 아니라면 기존 IPv4 토폴로지를 기반으로 IPv6 번호 지정 체계를 만드십시오.

## 노드에 대한 IPv6 주소 지정 계획 만들기

대부분의 호스트에서는 인터페이스에 대한 IPv6 주소의 Stateless 자동 구성이 적합한 시간 절약 전략입니다. 호스트가 가장 가까운 라우터로부터 사이트 접두어를 받으면 Neighbor Discovery가 호스트에 있는 각 인터페이스에 대한 IPv6 주소를 자동으로 생성합니다.

서버는 정적 IPv6 주소를 사용해야 합니다. 서버의 IPv6 주소를 수동으로 구성하지 않은 경우, 서버에서 NIC 카드가 교체될 때마다 새 IPv6 주소가 자동 구성됩니다. 서버 주소를 만들 때 다음 사항에 유의하십시오.

- 서버에 의미 있고 안정적인 인터페이스 ID를 제공합니다. 한 가지 전략은 인터페이스 ID에 순차적 번호 지정 체계를 사용하는 것입니다. 예를 들어 **그림 2-1**에 표시된 LDAP 서버의 내부 인터페이스는 2001:db8:3c4d:2::2가 될 수 있습니다.

- IPv4 네트워크의 번호를 정기적으로 재지정하지 않는 경우, 라우터 및 서버의 기존 IPv4 주소를 인터페이스 ID로 사용합니다. [그림 2-1](#)에서 DMZ에 대한 라우터 1 인터페이스의 IPv4 주소는 123.456.789.111이라고 가정합니다. IPv4 주소를 16진수로 변환한 다음 그 결과를 인터페이스 ID로 사용할 수 있습니다. 새 인터페이스 ID는 ::7bc8:156f입니다.

ISP로부터 주소를 받은 것이 아니라 등록된 IPv4 주소를 소유한 경우에만 이 방법을 사용하십시오. ISP가 제공한 IPv4 주소를 사용하는 경우 종속성이 생기는데, 이 종속성으로 인해 ISP를 변경하면 문제가 발생할 수 있습니다.

IPv4 주소의 개수에는 제한이 있으므로 과거에는 네트워크 설계자가 등록된 전역 주소 및 개인 RFC 1918 주소를 사용할 위치를 고려해야 했습니다. 그러나 IPv6 주소에는 전역 및 개인 IPv4 주소의 개념이 적용되지 않습니다. 사이트 접두어를 포함하는 전역 유니캐스트 주소를 공개 DMZ를 비롯한 모든 네트워크 링크에 사용할 수 있습니다.

## 서브넷 번호 지정 체계 만들기

기존 IPv4 서브넷을 해당 IPv6 서브넷에 매핑하여 번호 지정 체계를 시작하십시오. 예를 들어 [그림 2-1](#)에 표시된 서브넷을 고려하십시오. 서브넷 1-4은 주소의 처음 16비트에 대해 RFC 1918 IPv4 개인 주소 지정을 사용합니다. 숫자 1-4는 서브넷을 나타냅니다. 설명을 위해 IPv6 접두어 `refix 2001:db8:3c4d/48`가 사이트에 지정되었습니다.

다음 표는 개인 IPv4 접두어가 IPv6 접두어에 매핑되는 방식을 보여줍니다.

IPv4 서브넷 접두어	해당 IPv6 서브넷 접두어
192.168.1.0/24	2001:db8:3c4d:1::/64
192.168.2.0/24	2001:db8:3c4d:2::/64
192.168.3.0/24	2001:db8:3c4d:3::/64
192.168.4.0/24	2001:db8:3c4d:4::/64

서브넷에 대한 자세한 설명은 [System Administration Guide: IP Services](#)의 “What Is Subnetting?”을 참조하십시오.

## IPv6을 지원하도록 네트워크 서비스 구성

현재 Oracle Solaris 릴리스에서 제공하는 다음과 같은 일반 IPv4 네트워크 서비스는 IPv6에서 사용할 수 있습니다.

- sendmail
- NFS
- HTTP(Apache 2.x 또는 r Orion)
- DNS

- LDAP

IMAP 메일 서버는 IPv4에서만 사용 가능합니다.

IPv6용으로 구성된 노드는 IPv4 서비스를 실행할 수 있습니다. IPv6을 설정할 경우 모든 서비스가 IPv6 연결을 수락하는 것은 아닙니다. IPv6으로 이식된 서비스만 연결을 수락합니다. IPv6으로 이식되지 않은 서비스는 계속 프로토콜 스택의 IPv4 절반에서 작동합니다.

서비스를 IPv6으로 업그레이드한 후 문제가 발생할 수 있습니다. 자세한 내용은 133 페이지 “IPv6으로 서비스 업그레이드 후 발생하는 문제”를 참조하십시오.

## ▼ IPv6을 지원하도록 네트워크 서비스를 준비하는 방법

### 1 IPv6을 지원하도록 다음 네트워크 서비스를 업데이트합니다.

- 메일 서버
- NIS 서버
- NFS

---

주 - LDAP은 IPv6 관련 구성 작업 없이 IPv6을 지원합니다.

---

### 2 IPv6에서 방화벽 하드웨어를 사용할 수 있는지 확인합니다.

지침은 해당 방화벽 관련 설명서를 참조하십시오.

### 3 네트워크에 있는 다른 서비스가 IPv6으로 이식되었는지 확인합니다.

자세한 내용은 소프트웨어의 마케팅 보조 자료 및 관련 설명서를 참조하십시오.

### 4 사이트에서 다음 서비스를 배치하는 경우 이러한 서비스에 대해 적절한 조치를 취했는지 확인합니다.

- 방화벽

IPv6을 지원하기 위해 준비된 IPv4의 정책을 강화합니다. 보다 자세한 보안 고려 사항은 41 페이지 “IPv6 구현에 대한 보안 고려 사항”을 참조하십시오.

- 메일

DNS용 MX 레코드의 경우 메일 서버의 IPv6 주소를 추가합니다.

- DNS

DNS 관련 고려 사항은 40 페이지 “IPv6을 지원하도록 DNS를 준비하는 방법”을 참조하십시오.

- IPQoS

IPv4에 사용된 것과 동일한 Diffserv 정책을 호스트에 대해 사용합니다. 자세한 내용은 [461 페이지 “분류기 모듈”](#)을 참조하십시오.

- 5 해당 노드를 IPv6으로 변환하기 전에 노드에서 제공하는 네트워크 서비스를 감사합니다.

## ▼ IPv6을 지원하도록 DNS를 준비하는 방법

현재 Oracle Solaris 릴리스는 클라이언트 측과 서버 측 모두에 대한 DNS 분석을 지원합니다. IPv6을 위해 DNS 서비스를 준비하려면 다음을 수행하십시오.

IPv6에 대한 DNS 지원과 관련된 자세한 내용은 [Oracle Solaris Administration: Naming and Directory Services](#)를 참조하십시오.

- 1 순환 이름 분석을 수행하는 DNS 서버가 듀얼 스택(IPv4 및 IPv6)인지 아니면 IPv4 전용인지 확인합니다.
- 2 DNS 서버에서 DNS 데이터베이스를 정방향 영역의 관련 IPv6 데이터베이스 AAAA 레코드로 채웁니다.

---

주 - 중요한 서비스를 여러 개 실행하는 서버의 경우 특별한 주의가 필요합니다. 네트워크가 제대로 작동하는지 확인하십시오. 또한 중요한 서비스가 모두 IPv6으로 이식되었는지도 확인하십시오. 그런 다음 서버의 IPv6 주소를 DNS 데이터베이스에 추가하십시오.

---

- 3 AAAA 레코드의 연관된 PTR 레코드를 역방향 영역에 추가합니다.
- 4 영역에 대해 설명하는 NS 레코드에 IPv4 전용 데이터 또는 IPv6 및 IPv4 데이터를 추가합니다.

## 네트워크에서 터널 사용 계획

사용자의 네트워크가 IPv4 및 IPv6으로 마이그레이션되므로 IPv6 구현은 전환 방식으로 사용될 여러 터널 구성을 지원합니다. 터널을 통해 분리된 IPv6 네트워크가 통신할 수 있게 됩니다. 대부분의 인터넷은 IPv4를 실행하므로, 사용자 사이트의 IPv6 패킷은 인터넷에서 터널을 통과하여 대상 IPv6 네트워크로 이동합니다.

다음은 IPv6 네트워크 토폴로지에서 터널을 사용하기 위한 몇 가지 주요 시나리오입니다.

- IPv6 서비스를 구매한 ISP는 사이트의 경계 라우터에서 ISP 네트워크로 연결되는 터널을 만들 수 있도록 해줍니다. [그림 2-1](#)은 이러한 터널을 보여줍니다. 이 경우 IPv4 터널을 통해 수동 IPv6을 실행합니다.



- IPv4 연결로 분산된 대형 네트워크를 관리합니다. IPv6을 사용하는 분산된 사이트를 연결하려면 각 서브넷의 에지 라우터에서 자동 6to4 터널을 실행하면 됩니다.
- 기반구조의 라우터를 IPv6으로 업그레이드할 수 없는 경우도 있습니다. 이 경우 두 개의 IPv6 라우터를 끝점으로 사용하여 IPv4 라우터를 통과하는 수동 터널을 만들 수 있습니다.

터널 구성 절차는 120 페이지 “터널 구성(작업 맵)”을 참조하십시오. 터널과 관련된 개념 정보는 111 페이지 “IP 터널 개요”를 참조하십시오.

## IPv6 구현에 대한 보안 고려 사항

IPv6을 기존 네트워크에 사용할 경우 사이트의 보안이 손상되지 않도록 유의해야 합니다. IPv6 구현을 도입할 때 다음 보안 문제에 유의하십시오.

- IPv6 패킷과 IPv4 패킷 모두에 대해 동일한 양의 필터링이 필요합니다.
- IPv6 패킷은 대개 방화벽을 통해 터널링됩니다. 따라서 다음 시나리오 중 하나로 구현해야 합니다.
  - 방화벽이 터널 내에서 콘텐츠를 검사를 수행하도록 합니다.
  - 반대쪽 터널 끝점에 동일한 규칙을 사용하는 IPv6 방화벽을 배치합니다.
- IPv6 - UDP - IPv4 터널을 사용하는 전환 방식이 존재합니다. 이러한 방식은 방화벽을 방해하므로 위험합니다.
- IPv6 노드는 엔터프라이즈 네트워크 외부에서 전역적으로 연결할 수 있습니다. 보안 정책이 공개 액세스를 금지하는 경우 방화벽에 대해 보다 엄격한 규칙을 설정해야 합니다. 예를 들어 Stateful 방화벽 구성을 고려하십시오.

이 설명서는 IPv6 구현 내에서 사용할 수 있는 보안 기능을 다룹니다.

- IP 보안 아키텍처(IPsec) 기능을 통해 IPv6 패킷에 대한 암호화된 보호를 제공할 수 있습니다. 자세한 내용은 14 장, “IP 보안 아키텍처(개요)”를 참조하십시오.
- IKE(Internet Key Exchange) 기능을 통해 IPv6 패킷에 대한 공개 키 인증을 사용할 수 있습니다. 자세한 내용은 17 장, “Internet Key Exchange(개요)”를 참조하십시오.



## IPv4 네트워크 구성

---

네트워크 구성은 하드웨어 어셈블 단계와 데몬, 파일 및 TCP/IP 프로토콜을 구현하는 서비스에 대한 구성 단계로 진행됩니다.

이 장에서는 IPv4 주소 지정 및 서비스를 구현하는 네트워크에 대한 구성 방법을 설명합니다.

이 장에서 설명되는 대부분의 작업은 IPv4 전용 및 IPv6 사용 네트워크에 모두 적용됩니다. IPv6 네트워크에만 적용되는 작업은 4 장, “네트워크에서 IPv6 사용”에서 설명됩니다.

---

주 - TCP/IP를 구성하기 전에 1 장, “네트워크 배치 계획”에 나열되는 다양한 계획 작업을 검토하십시오. IPv6 주소를 사용하려면 2 장, “IPv6 주소 사용 시 고려 사항”도 참조하십시오.

---

이 장은 다음 정보를 포함합니다.

- 43 페이지 “네트워크 구성(작업 맵)”
- 44 페이지 “네트워크 구성을 시작하기 전에”
- 45 페이지 “네트워크의 구성 요소 시스템 구성”
- 67 페이지 “네트워크에 서브넷 추가”
- 69 페이지 “전송 계층 서비스 모니터 및 수정”

## 네트워크 구성(작업 맵)

다음 표에서는 서브넷이 없는 네트워크 구성에서 서브넷을 사용하는 네트워크로 변경한 후 수행할 추가 작업을 나열합니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
시스템의 IP 인터페이스를 구성합니다.	시스템의 IP 인터페이스에 IP 주소를 지정합니다.	47 페이지 “IP 인터페이스 구성 방법”
로컬 파일 모드에 대한 시스템을 구성합니다.	시스템의 /etc 디렉토리에 있는 특정 구성 파일을 편집하고 nis/domain SMF 서비스를 구성합니다.	53 페이지 “로컬 파일 모드에 대한 시스템 구성 방법”
네트워크 구성 서버를 설정합니다.	in.tftpd 데몬을 사용하여 설정하고 시스템의 /etc 디렉토리에 있는 다른 구성 파일을 편집합니다.	55 페이지 “네트워크 구성 서버 설정 방법”
네트워크 클라이언트 모드에 대한 시스템을 구성합니다.	시스템의 /etc 디렉토리에 있는 구성 파일을 편집합니다.	54 페이지 “네트워크 클라이언트 모드에 대한 시스템 구성 방법”
네트워크 클라이언트에 대한 경로 지정 전략을 지정합니다.	정적 경로 지정 또는 동적 경로 지정을 사용하도록 시스템을 구성합니다.	64 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법” 및 66 페이지 “단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법”.

## 네트워크 구성을 시작하기 전에

이 Oracle Solaris 릴리스에서 시스템의 네트워크 구성은 활성 **네트워크 구성 프로파일(NCP)**을 통해 관리됩니다. 시스템의 활성 NCP가 automatic인 경우 네트워크 구성은 OS를 통해 자동으로 관리됩니다. 활성 NCP가 DefaultFixed인 경우 dladm 및 ipadm 명령을 사용하여 수동으로 네트워크 구성을 수행하십시오.

---

주 - 활성 NCP가 Automatic인 경우 dladm 및 ipadm 명령이 작동하지 않습니다.

---

시스템의 활성 프로파일을 확인하고 고정 NCP로 전환하는 절차는 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “프로파일 및 구성 도구”**를 참조하십시오.

NCP에 대한 자세한 내용은 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 제1부**, “Network Auto-Magic”을 참조하십시오.

본 설명서의 절차에서는 네트워크의 모든 시스템에 있는 활성 NCP가 DefaultFixed인 것으로 간주합니다.

## 네트워크의 구성 요소 시스템 구성

네트워크 시스템을 구성할 때는 다음 구성 정보가 필요합니다.

- 각 시스템의 호스트 이름
- 각 시스템의 IP 주소 및 넷마스크. 네트워크가 서브넷으로 세분화된 경우 개별 넷마스크를 비롯하여 서브넷 번호와 각 서브넷의 시스템에 적용할 IP 주소 스키마가 있어야 합니다.
- 각 시스템이 속한 도메인 이름
- 기본 라우터 주소

각 네트워크에 연결된 라우터가 하나뿐인 간단한 네트워크 토폴로지를 사용하는 경우 이 정보를 제공합니다. 라우터가 RDISC(Router Discovery Server Protocol), RIP(Router Information Protocol) 등의 경로 지정 프로토콜을 실행하지 않는 경우에도 이 정보를 제공합니다. 라우터에 대한 자세한 내용 및 Oracle Solaris에서 지원하는 경로 지정 프로토콜 목록은 [System Administration Guide: IP Services](#)의 “[Packet Forwarding and Routing on IPv4 Networks](#)”을 참조하십시오.

---

주 - Oracle Solaris를 설치하는 동안 네트워크를 구성할 수 있습니다. 지침은 [Oracle Solaris 11 시스템](#)를 참조하십시오.

본 설명서의 절차에서는 OS를 설치한 후 네트워크를 구성 중인 것으로 간주합니다.

---

네트워크의 구성 요소 시스템을 구성하려면 다음 절의 [그림 3-1](#)을 참조하십시오.

## IPv4 자율 시스템 토폴로지

일반적으로 라우터와 네트워크가 여러 개인 사이트에서는 네트워크 토폴로지를 단일 경로 지정 도메인 또는 **자율 시스템(AS)**으로 관리합니다.

그림 3-1 IPv4 라우터가 여러 개인 자율 시스템

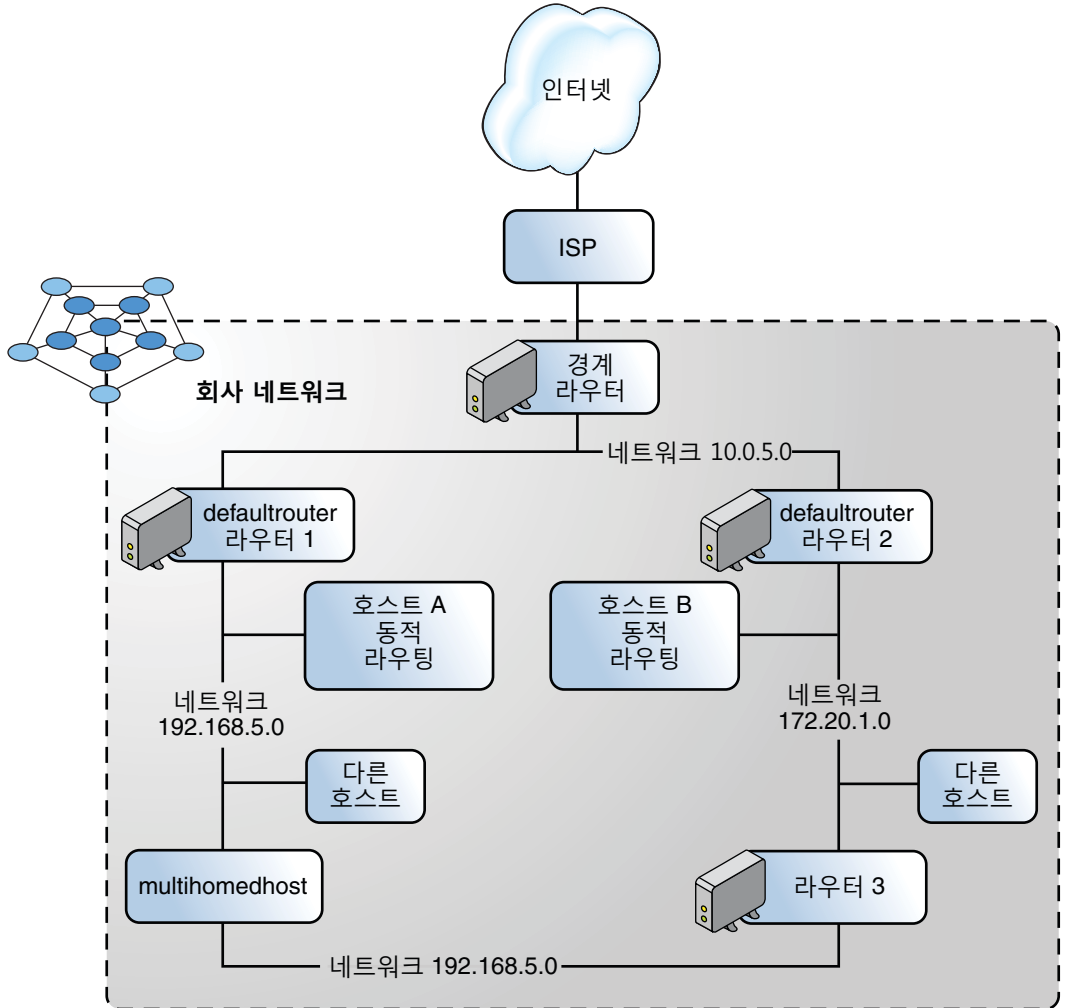


그림 3-1은 세 개의 로컬 네트워크(10.0.5.0, 172.20.1.0 및 192.168.5.0)로 구분된 AS를 보여 줍니다. 네트워크는 다음 유형의 시스템으로 구성됩니다.

- 라우터는 경로 지정 프로토콜을 사용하여 네트워크 패킷이 소스에서 로컬 네트워크 내의 대상 또는 외부 네트워크로 지정되거나 경로 지정되는 방식을 관리합니다. Oracle Solaris에서 지원되는 경로 지정 프로토콜에 대한 자세한 내용은 [140 페이지 “Oracle Solaris의 경로 지정 프로토콜 표”](#)를 참조하십시오.

라우터의 유형은 다음과 같습니다.

- **경계 라우터**는 외부적으로 로컬 네트워크(예: 10.0.5.0)를 서비스 공급자에 연결합니다.

- **기본 라우터**는 여러 로컬 네트워크를 자체적으로 포함할 수 있는 로컬 네트워크에서 패킷 경로 지정을 관리합니다. 예를 들어, **그림 3-1**에서 Router 1은 192.168.5에 대한 기본 라우터로 사용됩니다. 동시에 Router 1은 10.0.5.0 내부 네트워크에도 연결됩니다. Router 2의 인터페이스는 10.0.5.0 및 172.20.1.0 내부 네트워크에 연결됩니다.
- **패킷 전달 라우터**는 내부 네트워크 간에 패킷을 전달하지만 경로 지정 프로토콜을 실행하지 않습니다. **그림 3-1**에서 Router 3은 172.20.1 및 192.168.5 네트워크에 연결된 패킷 전달 라우터입니다.
- **클라이언트 시스템**
  - 멀티홈 시스템 또는 NIC가 여러 개인 시스템 Oracle Solaris에서 이러한 시스템은 기본적으로 패킷을 동일한 네트워크 세그먼트 내 다른 시스템으로 전달할 수 있습니다.
  - 단일 인터페이스 시스템은 패킷 전달 및 수신 구성 정보에 로컬 라우터를 사용합니다.

## ▼ IP 인터페이스 구성 방법

다음 절차에서는 기본적인 IP 인터페이스 구성을 수행하는 예를 제공합니다.

**시작하기 전에** 시스템에서 데이터 링크의 이름을 바꿀지 여부를 결정합니다. 기본적으로 데이터 링크에 지정된 일반 이름을 사용하는 것이 보통입니다. 링크 이름을 변경하려면 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “데이터 링크의 이름을 바꾸는 방법”**을 참조하십시오.

### 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”**을 참조하십시오.

### 2 (옵션) 현재 시스템에 있는 데이터 링크의 물리적 속성에 대한 정보를 표시합니다.

```
# dladm show-phys
```

이 명령은 시스템에 설치된 물리적 네트워크 카드 및 몇 가지 관련 등록 정보를 표시합니다. 이 명령에 대한 자세한 내용은 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “데이터 링크의 물리적 속성에 대한 정보를 표시하는 방법”**을 참조하십시오.

### 3 현재 시스템에 있는 데이터 링크에 대한 정보를 표시합니다.

```
# dladm show-link
```

이 명령은 데이터 링크 및 해당 데이터 링크에 대해 설정된 특정 등록 정보(링크가 만들어진 물리적 카드 등)를 표시합니다.

#### 4 IP 인터페이스를 만듭니다.

```
# ipadm create-interface-class interface
```

*interface-class* 만들 수 있는 다음과 같은 세 가지 인터페이스 클래스 중 하나를 나타냅니다.

- IP 인터페이스. 이 인터페이스 클래스는 네트워크 구성을 수행할 때 만드는 가장 일반적인 클래스입니다. 이 인터페이스 클래스를 만들려면 `create-ip` 하위 명령을 사용합니다.
- STREAMS 가상 네트워크 인터페이스 드라이버(VNI 인터페이스). 이 인터페이스 클래스를 만들려면 `create-vni` 하위 명령을 사용합니다. VNI 장치 또는 인터페이스에 대한 자세한 내용은 [vni\(7d\)](#) 매뉴얼 페이지를 참조하십시오.
- IPMP 인터페이스. 이 인터페이스는 IPMP 그룹을 구성할 때 사용됩니다. 이 인터페이스 클래스를 만들려면 `create-ipmp` 하위 명령을 사용합니다. IPMP 그룹에 대한 자세한 내용은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 14 장, "IPMP 소개"](#)를 참조하십시오.

*interface* 인터페이스의 이름을 나타냅니다. 이 이름은 인터페이스를 만들려는 링크의 이름과 동일합니다.

---

주 - IP 인터페이스를 만들어야만 여기에 IP 주소를 지정할 수 있습니다.

---

#### 5 유효한 IP 주소로 IP 인터페이스를 구성합니다.

다음 구문은 인터페이스에 정적 주소를 지정합니다. 기타 IP 주소 지정 옵션은 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
# ipadm create-addr -T address-type -a address/prefixlen addrobj
```

*-T address-type* 인터페이스에 지정된 IP 주소의 유형(static, dhcp, addrconf 중 하나)을 지정합니다. Addrconf는 자동으로 생성된 IPv6 주소를 나타냅니다.

*-a* 인터페이스에서 구성할 IP 주소를 지정합니다. 로컬 주소만 지정할 수도 있고, 터널 구성의 경우 로컬 주소와 원격 주소를 모두 지정할 수도 있습니다. 일반적으로 로컬 주소만 지정합니다. 이 경우 *-a* 옵션과 함께 직접 주소를 지정합니다(예: *-a address*). 주소는 자동으로 로컬 주소로 간주됩니다.

터널을 구성 중인 경우 시스템의 로컬 주소와 대상 시스템의 원격 주소를 모두 제공해야 할 수도 있습니다. 이 경우 *local* 및 *remote*를 지정하여 *-a local=local-addr, remote=remote-addr*과 같이 두 주소를 구분해야 합니다. 터널 구성에 대한 자세한 내용은 [Chapter 6, IP 터널 구성](#)을 참조하십시오.



숫자 IP 주소를 사용 중인 경우 CIDR 표기법으로 주소에 *address/prefixlen* 형식(예: 1.2.3.4/24)을 사용합니다. *prefixlen* 옵션에 대한 설명을 참조하십시오.

선택적으로 숫자 IP 주소 대신 *address*에 대한 호스트 이름을 지정할 수 있습니다. */etc/hosts* 파일에서 호스트 이름에 해당하는 숫자 IP 주소가 정의된 경우 호스트 이름을 사용하는 것이 유효합니다. 파일에서 숫자 IP 주소가 정의되지 않은 경우 *name-service/switch* 서비스에서 *host*에 대해 지정된 분석기 순서를 사용하여 숫자 값이 고유하게 지정됩니다. 지정된 호스트 이름에 대한 항목이 여러 개인 경우 오류가 발생합니다.

주 - 부트 프로세스 중 IP 주소 생성은 이름 지정 서비스를 온라인으로 전환하기 전에 진행됩니다. 따라서 네트워크 구성에서 사용되는 호스트 이름이 */etc/hosts* 파일에 정의되어 있는지 확인해야 합니다.

*/prefixlen*

CIDR 표기법을 사용하는 경우 IPv4 주소의 일부인 네트워크 ID의 길이를 지정합니다. 12.34.56.78/24 주소에서 24는 *prefixlen*입니다. *prefixlen*을 포함시키지 않을 경우 *name-service/switch* 서비스의 *netmask*에 대해 나열된 시퀀스에 따라 또는 클래스 기반 주소 의미를 사용하여 넷마스크가 계산됩니다.

*addrobj*

고유한 IP 주소 또는 시스템에서 사용되는 일련의 주소에 대한 식별자를 지정합니다. 주소는 IPv4 또는 IPv6 유형일 수 있습니다. 식별자는 *interface/user\_specified\_string* 형식을 사용합니다.

*interface*는 주소가 지정된 IP 인터페이스를 나타냅니다. *interface* 변수는 IP 인터페이스가 구성된 데이터 링크의 이름을 반영해야 합니다.

*user-specified-string*은 영문자로 시작하며 최대 길이가 32자인 영숫자 문자열을 나타냅니다. 나중에 시스템에서 주소를 관리하는 *ipadm* 하위 명령(예: *ipadm show-addr* 또는 *ipadm delete-addr*)을 사용할 때 숫자 IP 주소 대신 *addrobj*를 지정할 수 있습니다.

## 6 (옵션) 새로 구성된 IP 인터페이스에 대한 정보를 표시합니다.

확인할 정보에 따라 다음 명령을 사용할 수 있습니다.

- 인터페이스의 일반 상태를 표시합니다.

```
# ipadm show-if [interface]
```

인터페이스를 지정하지 않을 경우 시스템의 모든 인터페이스에 대한 정보가 표시됩니다.

- 인터페이스의 주소 정보를 표시합니다.

```
# ipadm show-addr [addrobj]
```

*addrobj*를 지정하지 않을 경우 시스템의 모든 주소 객체에 대한 정보가 표시됩니다.

`ipadm show-*` 하위 명령 출력에 대한 자세한 내용은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 및 주소 모니터링”](#)을 참조하십시오.

## 7 (옵션)/etc/hosts 파일에서 IP 주소에 대한 항목을 추가합니다.

이 파일의 항목은 IP 주소와 해당 호스트 이름으로 구성됩니다.

---

주 - 호스트 이름을 사용하는 정적 IP 주소를 구성 중인 경우에만 이 단계가 적용됩니다. DHCP 주소를 구성 중인 경우 `/etc/hosts` 파일을 업데이트하지 않아도 됩니다.

---

### 예 3-1 정적 주소로 네트워크 인터페이스 구성

```
# dladm show-phys
LINK      MEDIA      STATE      SPEED      DUPLEX      DEVICE
net3      Ethernet   up         100Mb      full        bge3

# dladm show-link
LINK      CLASS      MTU        STATE      BRIDGE      OVER
net3      phys       1500       up         --          --

# ipadm create-ip net3
# ipadm create-addr -T static -a 192.168.84.3/24 net3/v4static

# ipadm show-if
IFNAME    CLASS      STATE      ACTIVE      OVER
lo0       loopback   ok         yes         --
net3      ip         ok         yes         --

# ipadm show-addr
ADDROBJ    TYPE      STATE      ADDR
lo0/?      static    ok         127.0.0.1/8
net3/v4     static    ok         192.168.84.3/24

# vi /etc/hosts
# Internet host table
# 127.0.0.1          localhost
10.0.0.14          myhost
192.168.84.3       campus01
```

`/etc/hosts` 파일에서 `campus01`이 이미 정의된 경우 다음 주소를 지정할 때 호스트 이름을 사용할 수 있습니다.

```
# ipadm create-addr -T static -a campus01 net3/v4static
```

### 예 3-2 IP 주소로 네트워크 인터페이스 자동 구성

이 예에서는 이전 예와 동일한 네트워크 장치를 사용하되 DHCP 서버에서 주소를 수신하도록 인터페이스의 IP를 구성합니다.

```
# dladm show-phys
LINK      MEDIA      STATE      SPEED      DUPLEX      DEVICE
net3      Ethernet   up         100Mb      full        bge3

# dladm show-link
LINK      CLASS      MTU        STATE      BRIDGE      OVER
net3      phys       1500       up         --          --

# ipadm create-ip net3

# ipadm create-addr -T dhcp net3/dhcp

# ipadm show-if
IFNAME    CLASS      STATE      ACTIVE      OVER
lo0       loopback   ok         yes         --
net3      ip         ok         yes         --

# ipadm show-addr net3/dhcp
ADDROBJ   TYPE       STATE      ADDR
net3/dhcp dhcp       ok         10.8.48.242/24

# ipadm show-addr
ADDROBJ   TYPE       STATE      ADDR
lo0/?     static     ok         127.0.0.1/8
net3/dhcp dhcp       ok         10.8.48.242/24
```

## 시스템 구성 모드 설정

이 절에서는 로컬 파일 모드 또는 네트워크 클라이언트 모드에서 실행할 시스템을 설정하는 절차에 대해 설명합니다. 로컬 파일 모드에서 실행하는 경우 시스템은 로컬 디렉토리에 있는 파일에서 모든 TCP/IP 구성 정보를 가져옵니다. 네트워크 클라이언트 모드에서는 원격 네트워크 구성 서버가 네트워크의 모든 시스템에 구성 정보를 제공합니다.

일반적으로 다음과 같은 네트워크의 서버는 로컬 파일 모드에서 실행됩니다.

- 네트워크 구성 서버
- NFS 서버
- NIS, LDAP 또는 DNS 서비스를 제공하는 이름 서버
- 메일 서버
- 라우터

클라이언트는 두 모드 중 하나에서 실행할 수 있습니다. 따라서 네트워크에서는 다음 그림과 같이 구성된 다양한 시스템에서 이러한 모드의 조합이 사용될 수 있습니다.

그림 3-2 IPv4 네트워크 토폴로지 시나리오의 시스템

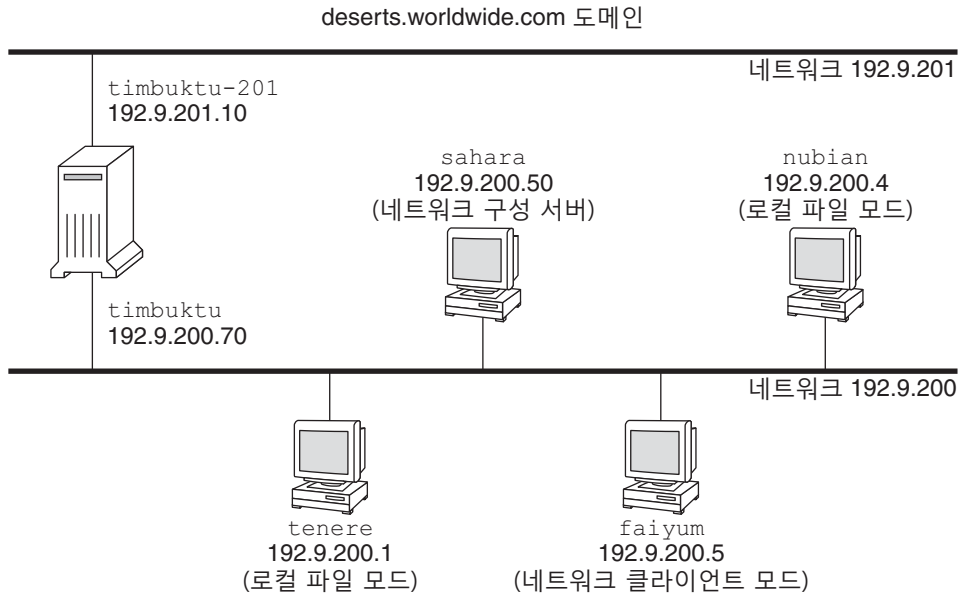


그림 3-2에서는 192.9.200 네트워크의 시스템을 보여 줍니다.

- 모든 시스템이 조직 도메인 `deserts.worldwide.com`에 속합니다.
- `sahara`는 구성 서버입니다. 서버로서, 시스템의 로컬 디스크에서 TCP/IP 구성 정보를 가져오는 로컬 파일 모드로 실행됩니다.

주 - 네트워크 클라이언트 모드에서 실행하도록 클라이언트를 구성할 경우 해당 클라이언트에 구성 정보를 제공할 네트워크 구성 서버를 하나 이상 구성해야 합니다.

- `tenere`, `nubian` 및 `faiyum`은 네트워크의 클라이언트입니다. `tenere` 및 `nubian`은 로컬 파일 모드에서 실행됩니다. `faiyum`의 로컬 디스크에 관계없이 시스템은 네트워크 클라이언트 모드에서 작동하도록 구성됩니다.
- `timbuktu`는 라우터로 구성되므로 로컬 파일 모드에서 작동합니다. 시스템에는 각각 고유하게 구성된 IP 인터페이스가 있는 두 개의 NIC가 포함되어 있습니다. 첫번째 IP 인터페이스는 이름이 `timbuktu`이며 192.9.200 네트워크에 연결됩니다. 두번째 IP 인터페이스는 이름이 `timbuktu-201`이며 192.9.201 네트워크에 연결됩니다.

두 구성 모드에 대한 자세한 개요는 [System Administration Guide: IP Services](#)의 “Determining Host Configuration Modes”을 참조하십시오.

## ▼ 로컬 파일 모드에 대한 시스템 구성 방법

이 절차에 따라 로컬 파일 모드에서 실행되도록 시스템(예: [System Administration Guide: IP Services](#)의 “[Systems That Should Run in Local Files Mode](#)”에 나열된 시스템)을 구성할 수 있습니다.

### 1 지정된 IP 주소로 시스템의 IP 인터페이스를 구성합니다.

절차는 47 페이지 “[IP 인터페이스 구성 방법](#)”을 참조하십시오.

### 2 `/etc/nodename` 파일에서 올바른 호스트 이름이 설정되었는지 확인합니다.

### 3 `/etc/inet/hosts` 파일의 항목이 최신인지 확인합니다.

Oracle Solaris 설치 프로그램이 기본 네트워크 인터페이스, 루프백 주소 및 설치 중 구성된 추가 인터페이스(해당하는 경우)에 대한 항목을 만듭니다.

파일에는 기본 라우터의 이름 및 라우터의 IP 주소도 포함되어야 합니다.

a. (옵션) 설치 후 시스템에 추가된 네트워크 인터페이스에 대한 IP 주소 및 해당 이름을 추가합니다.

b. (옵션) `/usr` 파일 시스템이 NFS 마운트된 시스템인 경우 파일 서버의 IP 주소를 추가합니다.

### 4 `nis/domain` SMF 서비스의 등록 정보로 시스템의 정규화된 도메인을 지정합니다.

예를 들어, `deserts.worldwide.com`을 `nis/domain` SMF 서비스의 `domainname` 등록 정보에 대한 값으로 지정합니다.

### 5 `/etc/defaultrouter` 파일에 라우터의 이름을 입력합니다.

### 6 해당하는 경우 넷마스크 정보를 추가합니다.

---

주 - DHCP 서비스를 사용 중인 경우 이 단계를 건너 뛰십시오.

---

### a. `/etc/inet/netmasks` 파일에 네트워크 번호 및 넷마스크를 입력합니다.

항목을 만들려면 `network-number netmask` 형식을 사용합니다. 예를 들어, 클래스 C 네트워크 번호 192.168.83의 경우 다음과 같이 입력합니다.

**192.168.83.0      255.255.255.0**

CIDR 주소의 경우 네트워크 접두어를 동등한 점으로 구분된 십진수 표현으로 변환합니다. 네트워크 접두어 및 동등한 점으로 구분된 십진수 표현은 [표 1-1](#)에서 확인할 수 있습니다. 예를 들어, CIDR 네트워크 접두어 192.168.3.0/22를 표현하려면 다음을 사용합니다.

```
192.168.3.0      255.255.252.0
```

- b. 로컬 파일이 먼저 검색되도록 스위치의 SMF 등록 정보에서 넷마스크에 대한 조회 순서를 변경한 다음 인스턴스를 새로 고칩니다.

```
# svccfg -s name-service/switch setprop config/host = astring: "files nis"
# svccfg -s name-service/switch:default refresh
```

- 7 시스템을 재부트합니다.

## ▼ 네트워크 클라이언트 모드에 대한 시스템 구성 방법

네트워크 클라이언트 모드에서 구성할 각 호스트에서 다음 절차를 수행합니다.

**시작하기 전에** 네트워크 클라이언트는 네트워크 구성 서버에서 구성 정보를 수신합니다. 따라서 시스템을 네트워크 클라이언트로 구성하기 전에 네트워크에 대해 하나 이상의 네트워크 구성 서버가 설정되었는지 확인해야 합니다.

- 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

- 2 지정된 IP 주소로 시스템의 IP 인터페이스를 구성합니다.

절차는 [47 페이지 “IP 인터페이스 구성 방법”](#)을 참조하십시오.

- 3 `/etc/inet/hosts` 파일에 루프백 네트워크 인터페이스의 `localhost` 이름 및 IP 주소만 포함되어 있는지 확인합니다.

```
# cat /etc/inet/hosts
# Internet host table
#
127.0.0.1      localhost
```

- 4 `nis/domain` SMF 서비스의 `domainname` 등록 정보에 지정된 값을 제거합니다.

- 5 클라이언트 `name-service/switch` 서비스의 검색 경로가 네트워크에 대한 동일한 서비스 요구 사항을 반영하는지 확인합니다.

## ▼ 네트워크 구성 서버 설정 방법

설치 서버 및 부트 서버 설정 정보는 **Oracle Solaris 11 시스템**에서 확인할 수 있습니다.

### 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 “관리 권한을 얻는 방법”을 참조하십시오.

### 2 다음과 같이 `in.tftpd` 데몬을 켭니다.

a. 지정된 네트워크 구성 서버의 루트(/) 디렉토리로 이동합니다.

b. 다음과 같이 `/tftpboot` 디렉토리를 만듭니다.

```
# mkdir /tftpboot
```

이 명령은 시스템을 TFTP, bootparams 및 RARP 서버로 구성합니다.

c. 디렉토리에 대한 심볼릭 링크를 만듭니다.

```
# ln -s /tftpboot/. /tftpboot/tftpboot
```

### 3 `/etc/inetd.conf` 파일에서 `tftp` 행을 추가합니다.

행이 다음과 같이 표시됩니다.

```
tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
```

이 행은 `in.tftpd`가 `/tftpboot`에 있는 파일 이외의 다른 파일을 검색하지 않도록 합니다.

### 4 `/etc/hosts` 데이터베이스에서 네트워크의 모든 클라이언트에 대한 호스트 이름 및 IP 주소를 추가합니다.

### 5 `/etc/ethers` 데이터베이스에서 네트워크 클라이언트 모드로 실행되는 네트워크의 모든 서버에 대한 항목을 만듭니다.

이 데이터베이스의 항목은 다음 형식을 사용합니다.

```
MAC Address      host name      #comment
```

자세한 내용은 **ethers(4)** 매뉴얼 페이지를 참조하십시오.

### 6 `/etc/bootparams` 데이터베이스에서 네트워크 클라이언트 모드로 실행되는 네트워크의 모든 시스템에 대한 항목을 만듭니다.

이 데이터베이스 편집에 대한 자세한 내용은 **bootparams(4)** 매뉴얼 페이지를 참조하십시오.

### 7 `/etc/inetd.conf` 항목을 SMF(서비스 관리 기능) 서비스 매니페스트로 변환하고 결과 서비스를 사용으로 설정합니다.

```
# /usr/sbin/inetconv
```

**8 in.tftpd가 제대로 작동 중인지 확인합니다.**

```
# svcs network/tftp/udp6
```

출력이 다음과 유사하게 표시됩니다.

```
STATE          STIME      FMRI
online         18:22:21  svc:/network/tftp/udp6:default
```

**자세한 정보 in.tftpd 데몬 관리**

in.tftpd 데몬은 서비스 관리 기능을 통해 관리됩니다. in.tftpd에 대한 관리 작업(예: 사용으로 설정, 사용 안함으로 설정 또는 다시 시작)은 `svcadm` 명령을 사용하여 수행할 수 있습니다. 이 서비스에 대한 시작 및 다시 시작 권한은 `inetd`로 위임됩니다. `inetadm` 명령을 사용하여 구성을 변경하고 in.tftpd에 대한 구성 정보를 볼 수 있습니다. `svcs` 명령을 사용하여 서비스 상태를 질의할 수 있습니다. 서비스 관리 기능의 개요는 [Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리\(개요\)”](#)를 참조하십시오.

## IPv4 라우터 구성

라우터는 두 개 이상의 네트워크 간의 인터페이스를 제공합니다. 따라서 라우터의 물리적 네트워크 인터페이스 각각에 고유한 이름과 IP 주소를 지정해야 합니다. 즉, 각 라우터에는 기본 네트워크 인터페이스와 연관된 호스트 이름과 IP 주소를 비롯하여 추가 네트워크 인터페이스 각각에 대한 하나 이상의 고유한 이름과 IP 주소가 있는 것입니다.

다음 절차에 따라 물리적 인터페이스가 하나뿐인 시스템(기본적으로 호스트)을 라우터로 구성할 수도 있습니다. [Oracle Solaris 관리: 네트워크 서비스의 “다이얼 업 PPP 링크 계획”](#)에 설명된 대로 시스템이 PPP 링크의 하나의 끝점으로 사용되는 경우 단일 인터페이스 시스템을 라우터로 구성할 수 있습니다.

## ▼ IPv4 라우터 구성 방법

다음 지침에서는 설치 후 라우터에 대한 인터페이스를 구성 중인 것으로 간주합니다.

**시작하기 전에** 라우터가 네트워크에 물리적으로 설치된 후 [53 페이지 “로컬 파일 모드에 대한 시스템 구성 방법”](#)에 설명된 대로 로컬 파일 모드에서 작동하도록 라우터를 구성합니다. 이 구성은 네트워크 구성 서버의 작동이 중지된 경우 라우터가 부트되도록 합니다.

**1 관리자로 로그인합니다.**

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.



- 2 시스템에 설치된 모든 NIC에 대해 47 페이지 “IP 인터페이스 구성 방법”에 설명된 대로 IP 인터페이스를 구성합니다.

각 IP 인터페이스는 시스템에서 패킷을 경로 지정할 네트워크의 IP 주소로 구성되어 있어야 합니다. 따라서 시스템이 192.168.5.0 및 10.0.5.0 네트워크를 제공하는 경우 각 네트워크에 대해 하나의 NIC를 구성해야 합니다.



주의 - DHCP를 사용하도록 IPv4 라우터를 구성하려면 DHCP 관리를 철저히 파악하고 있어야 합니다.

- 3 /etc/inet/hosts 파일에 각 인터페이스의 호스트 이름 및 IP 주소를 추가합니다.

예를 들어, Router 1의 두 인터페이스에 대해 지정된 이름이 각각 krakatoa와 krakatoa-1이라고 가정합니다. 이 경우 /etc/inet/hosts 파일의 항목은 다음과 같습니다.

```
192.168.5.1      krakatoa      #interface for network 192.168.5.0
10.0.5.1        krakatoa-1    #interface for network 10.0.5.0
```

- 4 나머지 단계를 수행하여 로컬 파일 모드에서 실행되도록 이 라우터를 구성합니다.

53 페이지 “로컬 파일 모드에 대한 시스템 구성 방법”을 참조하십시오.

- 5 라우터가 서브넷 네트워크에 연결된 경우 /etc/inet/netmasks 파일에 네트워크 번호 및 넷마스크를 추가합니다.

예를 들어, 추가 IPv4 주소 표기법(예: 192.168.5.0)의 경우 다음과 같이 입력합니다.

```
192.168.5.0      255.255.255.0
```

- 6 라우터에서 IPv4 패킷 전달을 사용으로 설정합니다.

```
# ipadm set-prop -p forwarding=on ipv4
```

- 7 (옵션) 경로 지정 프로토콜을 시작합니다.

다음 명령 구문 중 하나를 사용합니다.

- # routeadm -e ipv4-routing -u

- # svcadm enable route:default

in.routed 데몬과 연관된 SMF FMRI는 svc:/network/routing/route입니다.

경로 지정 프로토콜을 시작하면 경로 지정 데몬 /usr/sbin/in.routed가 자동으로 경로 지정 테이블을 업데이트합니다. 이 프로세스를 동적 경로 지정이라고 합니다. 경로 지정 유형에 대한 자세한 내용은 59 페이지 “경로 지정 테이블 및 경로 지정 유형”을 참조하십시오. routeadm 명령에 대한 자세한 내용은 routeadm(1M) 매뉴얼 페이지를 참조하십시오.

### 예 3-3 네트워크에 대한 기본 라우터 구성

이 예는 [그림 3-1](#)을 기반으로 합니다. Router 2에는 두 개의 유선 네트워크 연결(172.20.1.0 네트워크에 대한 연결과 10.0.5.0 네트워크에 대한 연결)이 포함되어 있습니다. 예에서는 172.20.1.0 네트워크의 기본 라우터가 되도록 Router 2를 구성하는 방법을 보여 줍니다. 또한 예에서는 [53 페이지 “로컬 파일 모드에 대한 시스템 구성 방법”](#)에 설명된 대로 Router 2가 로컬 파일 모드에서 작동하도록 구성되었다고 간주합니다.

수퍼유저 또는 동등한 역할의 사용자로 로그인한 후 시스템 인터페이스의 상태를 확인합니다.

```
# dladm show-link
LINK      CLASS      MTU      STATE    BRIDGE    OVER
net0      phys       1500     up       --        --
net1      phys       1500     up       --        --
net2      phys       1500     up       --        --
# ipadm show-addr
ADDROBJ   TYPE      STATE    ADDR
lo0/v4    static    ok       127.0.0.1/8
net0/v4    static    ok       172.20.1.10/24
```

net0만 IP 주소로 구성되었습니다. Router 2를 기본 라우터로 설정하려면 물리적으로 net1 인터페이스를 10.0.5.0 네트워크에 연결합니다.

```
# ipadm create-ip net1
# ipadm create-addr -T static -a 10.0.5.10/24 net1/v4
# ipadm show-addr
ADDROBJ   TYPE      STATE    ADDR
lo0/v4    static    ok       127.0.0.1/8
net0/v4    static    ok       172.20.1.10/24
net1/v4    static    ok       10.0.5.10/24
```

그런 다음 새로 구성된 인터페이스 및 연결된 네트워크에 대한 정보로 다음 네트워크 데이터베이스를 업데이트합니다.

```
# vi /etc/inet/hosts
127.0.0.1      localhost
172.20.1.10    router2        #interface for network 172.20.1
10.0.5.10     router2-out    #interface for network 10.0.5
# vi /etc/inet/netmasks
172.20.1.0    255.255.255.0
10.0.5.0     255.255.255.0
```

마지막으로 패킷 전달 및 in.routed 경로 지정 데몬을 사용으로 설정합니다.

```
# ipadm set-prop -p forwarding=on ipv4
# svcadm enable route:default
```

그러면 RIP를 통한 IPv4 패킷 전달 및 동적 경로 지정이 Router 2에서 사용으로 설정되었지만, 172.20.1.0 네트워크에 대한 기본 라우터 구성은 아직 완료되지 않은 것입니다. 다음 작업을 수행해야 합니다.

- 호스트가 새 기본 라우터에서 경로 지정 정보를 가져오도록 172.20.1.0 네트워크에서 각 호스트를 수정합니다. 자세한 내용은 [64 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법”](#)을 참조하십시오.
- Router 2의 경로 지정 테이블에서 경계 라우터에 대한 정적 경로 지정을 정의합니다. 자세한 내용은 [59 페이지 “경로 지정 테이블 및 경로 지정 유형”](#)을 참조하십시오.

## 경로 지정 테이블 및 경로 지정 유형

라우터와 호스트는 모두 **경로 지정 테이블**에서 유지 관리됩니다. 경로 지정 테이블에는 시스템의 로컬 기본 네트워크를 비롯하여 시스템에서 인식한 네트워크의 IP 주소가 나열됩니다. 알려진 각 네트워크에 대한 게이트웨이 시스템의 IP 주소도 나열됩니다. **게이트웨이**는 송신 패킷을 수신하여 로컬 네트워크 외부의 한 홉으로 전달할 수 있습니다.

다음은 IPv4 전용 네트워크의 시스템에 대한 간단한 경로 지정 테이블입니다.

Routing Table: IPv4					
Destination	Gateway	Flags	Ref	Use	Interface
default	172.20.1.10	UG	1	532	net0
224.0.0.0	10.0.5.100	U	1	0	net1
10.0.0.0	10.0.5.100	U	1	0	net1
127.0.0.1	127.0.0.1	UH	1	57	lo0

Oracle Solaris 시스템에서는 두 가지 유형(정적 및 동적)의 경로 지정을 구성할 수 있습니다. 단일 시스템에서 경로 지정 유형 중 하나 또는 두 가지 모두를 구성할 수 있습니다. **동적 경로 지정**을 구현하는 시스템은 경로 지정 프로토콜(IPv4 네트워크의 경우 RIP, IPv6 네트워크의 경우 RIPng)을 사용하여 네트워크 트래픽을 경로 지정하고 테이블의 경로 지정 정보를 업데이트합니다. **정적 경로 지정**을 사용하는 경우 **route** 명령을 사용하여 수동으로 경로 지정 정보를 유지 관리합니다. 자세한 내용은 [route\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

로컬 네트워크 또는 자율 시스템에 대한 경로 지정을 구성할 때는 특정 라우터 및 호스트에서 지원할 경로 지정 유형을 고려하십시오.

다음 표에서는 다양한 경로 지정 유형과 각 경로 지정 유형이 최적으로 적용되는 네트워킹 시나리오를 보여 줍니다.

경로 지정 유형	최적 사용 사례
정적	작은 규모의 네트워크, 기본 라우터에서 경로를 가져오는 호스트, 다음 홉에서 하나 또는 두 개의 라우터에 대해서만 인식해야 할 기본 라우터
동적	보다 큰 규모의 인터넷 네트워크, 호스트가 여러 개인 로컬 네트워크의 라우터, 큰 자율 시스템의 호스트. 거의 모든 네트워크의 시스템에 동적 경로 지정을 선택하는 것이 좋습니다.
정적과 동적 결합	정적으로 경로 지정된 네트워크와 동적으로 경로 지정된 네트워크를 연결하는 라우터, 내부 자율 시스템을 외부 네트워크와 연결하는 경계 라우터. 시스템에서 정적 경로 지정과 동적 경로 지정을 결합하여 사용하는 것이 일반적입니다.

그림 3-1에 표시된 AS는 정적 경로 지정과 동적 경로 지정을 결합한 것입니다.

주 - 시스템에서는 동일한 대상에 대한 두 경로를 통해 자동으로 로드 균형 조정 또는 페일오버를 수행하지 않습니다. 이러한 기능이 필요할 경우 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 14 장, “IPMP 소개”](#)에 설명된 대로 IPMP를 사용하십시오.

## ▼ 경로 지정 테이블에 정적 경로 지정을 추가하는 방법

### 1 경로 지정 테이블의 현재 상태를 확인합니다.

일반 사용자 계정으로 다음 형식의 `netstat` 명령을 실행합니다.

```
% netstat -rn
```

출력이 다음과 유사하게 표시됩니다.

Routing Table: IPv4					
Destination	Gateway	Flags	Ref	Use	Interface
192.168.5.125	192.168.5.10	U	1	5879	net0
224.0.0.0	198.168.5.10	U	1	0	net0
default	192.168.5.10	UG	1	91908	
127.0.0.1	127.0.0.1	UH	1	811302	lo0

### 2 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

### 3 (옵션) 경로 지정 테이블의 기존 항목을 비웁니다.

```
# route flush
```

### 4 시스템 재부트 시 지속되는 경로를 추가합니다.

```
# route -p add -net network-address -gateway gateway-address
```

- p 시스템 재부트 시 지속되어야 할 경로를 만듭니다. 경로를 현재 세션에만 적용하려면 -p 옵션을 사용하지 마십시오.
- net *network-address* 경로가 *network-address*의 주소를 사용하는 네트워크로 이동하도록 지정합니다.
- gateway *gateway-address* 지정된 경로에 대한 게이트웨이 시스템의 IP 주소가 *gateway-address*임을 나타냅니다.

### 예 3-4 경로 지정 테이블에 정적 경로 지정 추가

다음 예에서는 [그림 3-1](#)의 Router 2에 정적 경로 지정을 추가하는 방법을 보여 줍니다. 정적 경로 지정은 AS의 경계 라우터 10.0.5.150에 필요합니다.

Router 2의 경로 지정 테이블을 보려면 다음을 입력합니다.

```
# netstat -rn
Routing Table: IPv4
  Destination          Gateway             Flags Ref  Use  Interface
-----
default                172.20.1.10        UG      1    249  ce0
224.0.0.0              172.20.1.10        U        1     0  ce0
10.0.5.0               10.0.5.20          U        1    78  bge0
127.0.0.1              127.0.0.1          UH       1    57  lo0
```

경로 지정 테이블은 Router 2가 인식하는 두 경로를 나타냅니다. 기본 경로는 Router 2의 172.20.1.10 인터페이스를 게이트웨이로 사용합니다. 두번째 경로 10.0.5.0은 Router 2에서 실행되는 in.routed 데몬을 통해 검색되었습니다. 이 경로에 대한 게이트웨이는 IP 주소가 10.0.5.20인 Router 1입니다.

게이트웨이가 경계 라우터인 10.0.5.0 네트워크에 두번째 경로를 추가하려면 다음을 입력합니다.

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

그러면 경로 지정 테이블에 IP 주소가 10.0.5.150/24인 경계 라우터에 대한 경로가 포함됩니다.

```
# netstat -rn
Routing Table: IPv4
  Destination          Gateway             Flags Ref  Use  Interface
-----
default                172.20.1.10        UG      1    249  ce0
224.0.0.0              172.20.1.10        U        1     0  ce0
10.0.5.0               10.0.5.20          U        1    78  bge0
10.0.5.0               10.0.5.150         U        1   375  bge0
127.0.0.1              127.0.0.1          UH       1    57  lo0
```

## 멀티홈 호스트 구성

Oracle Solaris에서는 인터페이스가 두 개 이상인 시스템을 **멀티홈 호스트**로 간주합니다. 멀티홈 호스트의 인터페이스는 다른 물리적 네트워크 또는 동일한 물리적 네트워크의 서로 다른 서브넷에 연결됩니다.

여러 인터페이스가 동일한 서브넷에 연결되는 시스템에서는 먼저 인터페이스를 하나의 IPMP 그룹으로 구성해야 합니다. 그렇지 않으면 시스템이 멀티홈 호스트가 될 수 없습니다. IPMP에 대한 자세한 내용은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 14 장, “IPMP 소개”](#)를 참조하십시오.

멀티홈 호스트는 IP 패킷을 전달하지 않지만 경로 지정 프로토콜을 실행하도록 구성될 수 있습니다. 일반적으로 다음 유형의 시스템을 멀티홈 호스트로 구성합니다.

- 대규모 사용자 풀에서 파일을 공유하기 위해 NFS 서버, 특히 큰 데이터 센터로 작동하는 서버를 두 개 이상의 네트워크에 연결할 수 있습니다. 이러한 서버는 경로 지정 테이블을 유지 관리할 필요가 없습니다.
- NFS 서버와 마찬가지로 데이터베이스 서버는 대규모 사용자 풀에 리소스를 제공할 네트워크 인터페이스를 여러 개 포함할 수 있습니다.
- 방화벽 게이트웨이는 회사 네트워크와 공용 네트워크(예: 인터넷) 간의 연결을 제공하는 시스템입니다. 관리자는 방화벽을 보안 조치로 설정합니다. 방화벽으로 구성된 호스트는 호스트의 인터페이스에 연결된 네트워크 간에 패킷을 전달하지 않습니다. 단, 이 경우에도 호스트는 권한이 부여된 사용자에게 표준 TCP/IP 서비스(예: ssh)를 제공할 수 있습니다.

---

주 - 멀티홈 호스트의 인터페이스에 여러 유형의 방화벽이 있을 경우 의도치 않게 호스트의 패킷이 중단되지 않도록 주의해야 합니다. 이 문제는 특히 Stateful 방화벽에서 발생할 수 있습니다. 한 가지 해결 방법은 Stateless 방화벽을 구성하는 것입니다. 방화벽에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “방화벽 시스템”](#) 또는 타사 방화벽 설명서를 참조하십시오.

---

### ▼ 멀티홈 호스트를 만드는 방법

#### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

#### 2 Oracle Solaris 설치의 일부로 구성되지 않은 각 추가 네트워크 인터페이스를 구성합니다. 47 페이지 “IP 인터페이스 구성 방법”을 참조하십시오.

### 3 패킷 전달을 사용으로 설정할 경우 이 서비스를 사용 안함으로 설정합니다.

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY   PERM CURRENT   PERSISTENT  DEFAULT  POSSIBLE
ipv4 forwarding  rw   on          --         off      on,off
```

```
ipadm set-prop -p forwarding=off ipv4
```

### 4 (옵션) 멀티홉 호스트에 대한 동적 경로 지정을 켭니다.

다음 명령 구문 중 하나를 사용합니다.

- # routeadm -e ipv4-routing -u

- # svcadm enable route:default

in.routed 데몬과 연관된 SMF FMRI는 svc:/network/routing/route입니다.

## 예 3-5 멀티홉 호스트 구성

다음 예에서는 [그림 3-1](#)에 표시된 멀티홉 호스트를 구성하는 방법을 보여 줍니다. 이 예에서는 시스템의 호스트 이름이 hostc입니다. 이 호스트에는 두 개의 인터페이스가 있으며 모두 192.168.5.0 네트워크에 연결됩니다.

시작하려면 시스템 인터페이스의 상태를 표시합니다.

```
# dladm show-link
LINK      CLASS      MTU      STATE    BRIDGE    OVER
net0      phys       1500     up       --        --
net1      phys       1500     up       --        --

# ipadm show-addr
ADDROBJ   TYPE      STATE      ADDR
lo0/v4    static    ok         127.0.0.1/8
net0/v4    static    ok         192.168.5.82/24
```

dladm show-link 명령이 hostc에 두 개의 데이터 링크가 있는 것으로 보고합니다. 하지만 net0만 IP 주소로 구성되었습니다. hostc를 멀티홉 호스트로 구성하려면 동일한 192.168.5.0 네트워크의 IP 주소로 net1을 구성합니다. net1의 기본적인 물리적 NIC가 네트워크에 물리적으로 연결되었는지 확인합니다.

```
# ipadm create-ip net1
# ipadm create-addr -T static -a 192.168.5.85/24 bge0/v4
# ipadm show-addr
ADDROBJ   TYPE      STATE      ADDR
lo0/v4    static    ok         127.0.0.1/8
net0/v4    static    ok         192.168.5.82/24
net1/v4    static    ok         192.168.5.85/24
```

그런 다음 net1 인터페이스를 /etc/hosts 데이터베이스에 추가합니다.

```
# vi /etc/inet/hosts
127.0.0.1      localhost
192.168.5.82   hostc      #primary network interface for host3
192.168.5.85   hostc-2    #second interface
```

다음으로 이 서비스가 hostc에서 실행 중인 경우 패킷 전달을 끕니다.

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4 forwarding  rw   on         --          off       on,off
```

```
# ipadm set-prop -p forwarding=off ipv4
```

```
# routeadm
```

Configuration Option	Current Configuration	Current System State
IPv4 routing	enabled	enabled
IPv6 routing	disabled	disabled

```
Routing services "route:default ripng:default"
```

routeadm 명령이 in.routed 데몬을 통한 동적 경로 지정이 현재 사용으로 설정된 것으로 보고합니다.

## 단일 인터페이스 시스템에 대한 경로 지정 구성

정적 또는 동적 경로 지정으로 단일 인터페이스 시스템을 구성할 수 있습니다. 정적 경로 지정을 사용하는 경우 호스트는 경로 지정 정보에 기본 라우터의 서비스를 사용해야 합니다. 다음 절차에서는 두 경로 지정 유형을 사용으로 설정하는 지침도 제공합니다.

### ▼ 단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법

다음 절차에 따라 멀티홈 호스트에서 정적 경로 지정을 구성할 수도 있습니다.

#### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

#### 2 시스템이 속한 네트워크에 대한 IP 주소로 시스템의 IP 인터페이스를 구성합니다.

지침은 [47 페이지 “IP 인터페이스 구성 방법”](#)을 참조하십시오.

#### 3 텍스트 편집기에서 시스템에 사용될 라우터의 IP 주소를 추가하여 /etc/defaultrouter 파일을 만들거나 수정합니다.

#### 4 로컬 /etc/inet/hosts 파일에서 기본 라우터에 대한 항목을 추가합니다.



5 경로 지정이 꺼져 있는지 확인합니다.

```
# routeadm
Configuration      Current      Current
                   Option      Configuration  System State
-----
IPv4 routing        enabled      disabled
IPv6 routing        disabled     disabled

Routing services   "route:default ripng:default"

# svcadm disable route:default
```

6 패킷 전달이 꺼져 있는지 확인합니다.

```
# # ipadm show-prop -p forwarding ipv4
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4 forwarding rw on -- off on,off

# ipadm set-prop -p forwarding=off ipv4
```

예 3-6 단일 인터페이스 시스템에서 정적 경로 지정 구성

다음 예에서는 [그림 3-1](#)에 표시된 172.20.1.0 네트워크의 단일 인터페이스 시스템 hostb에 대해 정적 경로 지정을 구성하는 방법을 보여 줍니다. hostb는 Router 2를 기본 라우터로 사용해야 합니다. 이 예에서는 시스템의 IP 인터페이스를 이미 구성한 것으로 간주합니다.

먼저 관리자 권한으로 hostb에 로그인합니다. 그런 다음 /etc/defaultrouter 파일이 시스템에 있는지 여부를 확인합니다.

```
# cd /etc
# ls | grep defaultrouter

# vi /etc/defaultrouter
172.20.1.10

IP 주소 172.20.1.10은 Router 2에 속합니다.

# vi /etc/inet/hosts
127.0.0.1 localhost
172.20.1.18 host2 #primary network interface for host2
172.20.1.10 router2 #default router for host2

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4 forwarding rw on -- off on,off

# ipadm set-prop -p forwarding=off ipv4

# routeadm
Configuration      Current      Current
                   Option      Configuration  System State
-----
```

```
IPv4 routing    enabled          disabled
IPv6 routing    disabled         disabled

Routing services "route:default ripng:default"
```

```
# svcadm disable route:default
```

## ▼ 단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법

경로 지정 프로토콜을 사용하는 동적 경로 지정이 시스템에서 경로 지정을 관리하는 가장 간단한 방법입니다.

### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

### 2 시스템이 속한 네트워크에 대한 IP 주소로 시스템의 IP 인터페이스를 구성합니다.

지침은 [47 페이지 “IP 인터페이스 구성 방법”](#)을 참조하십시오.

### 3 /etc/defaultrouter 파일에서 항목을 삭제합니다.

/etc/defaultrouter 파일이 비어 있으면 시스템이 동적 경로 지정을 사용합니다.

### 4 패킷 전달이 사용 안함으로 설정되었는지 확인합니다.

```
# ipadm set-prop -p forwarding=off ipv4
```

### 5 시스템에서 경로 지정 프로토콜을 사용으로 설정합니다.

다음 명령 중 하나를 사용합니다.

- # routeadm -e ipv4-routing -u
- # svcadm enable route:default

## 예 3-7 단일 인터페이스 시스템에서 동적 경로 지정 실행

다음 예에서는 [그림 3-1](#)에 표시된 192.168.5.0 네트워크에서 단일 인터페이스 시스템 hosta에 대해 동적 경로 지정을 구성하는 방법을 보여 줍니다. 시스템에서는 Router 1을 기본 라우터로 사용합니다. 이 예에서는 시스템의 IP 인터페이스를 이미 구성한 것으로 간주합니다.

먼저 관리자 권한으로 hosta에 로그인합니다. 그런 다음 /etc/defaultrouter 파일이 시스템에 있는지 여부를 확인합니다.

```
# cd /etc
# ls | grep defaultrouter
defaultrouter
```

```
# cat defaultrouter
192.168.5.10
```

파일에 Router 1에 대한 IP 주소인 192.168.5.10 항목이 제대로 포함됩니다.

```
# routecadm Configuration Option Current Configuration Current System State
-----
IPv4 routing disabled disabled
IPv6 routing disabled disabled

Routing services "route:default ripng:default"

# svcadm enable route:default

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4 forwarding rw on -- off on,off

# ipadm set-prop -p forwarding=off ipv4
```

## 네트워크에 서브넷 추가

서브넷을 사용하지 않는 네트워크에서 서브넷을 사용하는 네트워크로 변경 중인 경우 다음 목록의 작업을 수행합니다. 목록에서는 서브넷 스키마를 이미 준비한 것으로 간주합니다. 개요는 [System Administration Guide: IP Services](#)의 “What Is Subnetting?”을 참조하십시오.

- 서브넷에 속한 시스템에 새 서브넷 번호의 IP 주소를 지정합니다.  
자세한 내용은 47 페이지 “IP 인터페이스 구성 방법”을 참조하십시오.
- 각 시스템의 /etc/netmasks 파일에 올바른 IP 주소 및 넷마스크를 추가합니다.
- 각 시스템의 /etc/inet/hosts 파일을 호스트 이름에 해당하는 올바른 IP 주소로 개정합니다.
- 서브넷의 모든 시스템을 재부트합니다.

다음 절차는 서브넷과 밀접한 관련이 있습니다. 서브넷 없이 네트워크를 구성한 후 나중에 서브넷을 구현하는 경우 다음 절차에 따라 변경 사항을 구현합니다.

### ▼ IPv4 주소 및 기타 네트워크 구성 매개변수 변경 방법

이 절차에서는 이전에 설치된 시스템에서 IPv4 주소, 호스트 이름 및 기타 네트워크 매개변수를 수정하는 방법에 대해 설명합니다. 절차에 따라 서버 또는 네트워크로 연결된 독립형 시스템의 IP 주소를 수정할 수 있습니다. 네트워크 클라이언트 또는 어플라이언스에는 이 절차를 사용할 수 없습니다. 단계에서는 재부트 시 지속되는 구성을 만듭니다.

주- 특히 기본 네트워크 인터페이스의 IPv4 주소를 변경하려는 경우 지침을 따르십시오. 시스템에 다른 인터페이스를 추가하려면 47 페이지 “IP 인터페이스 구성 방법”을 참조하십시오.

대부분의 경우 다음 단계에서는 기존 IPv4의 점으로 구분된 십진수 표기법을 사용하여 IPv4 주소 및 서브넷 마스크를 지정합니다. 또는 CIDR 표기법을 사용하여 이 절차의 모든 해당 파일에서 IPv4 주소를 지정할 수도 있습니다. CIDR 표기법 소개는 [System Administration Guide: IP Services](#)의 “IPv4 Addresses in CIDR Format”를 참조하십시오.

#### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

#### 2 ipadm 명령을 사용하여 IP 주소를 수정합니다.

ipadm 명령을 통해서만 IP 주소를 직접 수정할 수 없습니다. 먼저 수정할 IP 주소를 나타내는 주소 객체를 삭제합니다. 그런 다음 동일한 주소 객체 이름을 사용하여 새 주소를 지정합니다.

```
# ipadm delete-addr addrobj
# ipadm create-addr -T static IP-address addrobj
```

#### 3 해당하는 경우 /etc/inet/hosts 파일 또는 동등한 hosts 데이터베이스에서 호스트 이름을 수정합니다.

#### 4 해당하는 경우 system/identity: node SMF 서비스에서 호스트 이름 항목을 수정합니다.

```
# svccfg -s svc:/system/identity:node setprop config/nodename = astring: hostname
```

#### 5 서브넷 마스크가 변경된 경우 /etc/netmasks 파일에서 서브넷 항목을 수정합니다.

#### 6 서브넷 주소가 변경된 경우 /etc/defaultrouter의 기본 라우터 IP 주소를 새 서브넷의 기본 라우터 IP 주소로 변경합니다.

#### 7 시스템을 재부트합니다.

```
# reboot -- -r
```

### 예 3-8 IP 주소 및 호스트 이름 변경

이 예에서는 호스트 이름, 기본 네트워크 인터페이스의 IP 주소 및 서브넷 마스크를 변경하는 방법을 보여 줍니다. 기본 네트워크 인터페이스 bge0에 대한 IP 주소가 10.0.0.14에서 192.168.34.100으로 변경됩니다.

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
```

```

lo0/v4      static  ok      127.0.0.1/8
bge0/v4     static  ok      10.0.0.14/24

# ipadm delete-addr bge0/v4
# ipadm create-addr -T static -a 192.168.34.100/24 bge0/v4
# svccfg -s svc:/system/identity:node setprop config/nodename = astring: mynewhostname

# ipadm show-addr
ADDROBJ      TYPE      STATE     ADDR
lo0/v4       static    ok        127.0.0.1/8
bge0/v4new   static    ok        192.168.34.100/24

# hostname
mynewhostname

```

**참조** 기본 네트워크 인터페이스 이외의 다른 인터페이스에 대한 IP 주소를 변경하려면 [Oracle Solaris 관리: 일반 작업](#) 및 47 페이지 “IP 인터페이스 구성 방법”을 참조하십시오.

## 전송 계층 서비스 모니터 및 수정

전송 계층 프로토콜인 TCP, SCTP 및 UDP는 표준 Oracle Solaris 패키지의 일부입니다. 일반적으로 이러한 프로토콜은 개입 없이도 제대로 실행됩니다. 하지만 사이트의 요구 사항에 따라 전송 계층 프로토콜을 통해 실행되는 서비스를 기록하거나 수정해야 할 수도 있습니다. 그런 다음 [Oracle Solaris 관리: 일반 작업의 6 장](#), “서비스 관리(개요)”에 설명된 대로 SMF(서비스 관리 기능)를 사용하여 해당 서비스에 대한 프로파일을 수정해야 합니다.

inetd 데몬은 시스템 부트 시 표준 인터넷 서비스를 시작합니다. 이러한 서비스에는 TCP, SCTP 또는 UDP를 전송 계층 프로토콜로 사용하는 응용 프로그램이 포함됩니다. SMF 명령을 사용하여 기존 인터넷 서비스를 수정하거나 새 서비스를 추가할 수 있습니다. inetd에 대한 자세한 내용은 136 페이지 “inetd Internet Services Daemon”을 참조하십시오.

전송 계층 프로토콜과 관련된 작업은 다음과 같습니다.

- 모든 수신 TCP 연결 기록
- 전송 계층 프로토콜을 통해 실행되는 서비스 추가, SCTP를 예로 사용
- 액세스 제어를 위해 TCP 래퍼 기능 구성

inetd 데몬에 대한 자세한 내용은 [inetd\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ 모든 수신 TCP 연결의 IP 주소 기록 방법

### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

### 2 inetd로 관리되는 모든 서비스에 대해 TCP 추적을 사용으로 설정합니다.

```
# inetadm -M tcp_trace=TRUE
```

## ▼ SCTP 프로토콜을 사용하는 서비스를 추가하는 방법

SCTP 전송 프로토콜은 TCP와 유사한 방식으로 응용 프로그램 전송 프로토콜에 서비스를 제공합니다. 하지만 SCTP는 둘 중 하나 또는 모두가 멀티홈일 수 있는 두 시스템 간의 통신을 가능하게 합니다. SCTP 연결을 **연관**이라고 합니다. 연관에서 응용 프로그램은 하나 이상의 메시지 스트림으로 전송되거나 **다중 스트림**될 데이터를 구분합니다. SCTP 연결은 IP 주소가 여러 개인 끝점으로 이동할 수 있으므로 전화 기술 응용 프로그램에서 특히 중요합니다. 사이트에서 IP 필터 또는 IPsec를 사용하는 경우 보안상 SCTP의 멀티 홈 기능을 고려해야 합니다. 이러한 고려 사항 중 몇 가지는 [sctp\(7P\)](#) 매뉴얼 페이지에서 설명됩니다.

기본적으로 SCTP는 Oracle Solaris에 포함되어 있으며 추가 구성을 필요로 하지 않습니다. 단, SCTP를 사용하도록 특정 응용 프로그램 계층 서비스를 명시적으로 구성해야 합니다. **echo** 및 **discard**가 이러한 응용 프로그램에 해당합니다. 다음 절차에서는 SCTP 일대일 스타일 소켓을 사용하는 **echo** 서비스를 추가하는 방법을 보여 줍니다.

---

주 - 다음 절차에 따라 TCP 및 UDP 전송 계층 프로토콜에 대한 서비스를 추가할 수도 있습니다.

---

다음 작업에서는 **inetd** 데몬으로 관리되는 SCTP **inet** 서비스를 SMF 저장소에 추가하는 방법을 보여 줍니다. 그런 다음 SMF(서비스 관리 기능) 명령을 사용하여 서비스를 추가하는 방법을 보여 줍니다.

- SMF 명령에 대한 자세한 내용은 [Oracle Solaris 관리: 일반 작업의 “SMF 명령줄 관리 유틸리티”](#)를 참조하십시오.
- 구문 정보는 절차에서 인용된 SMF 명령에 대한 매뉴얼 페이지를 참조하십시오.
- SMF에 대한 자세한 내용은 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

**시작하기 전에** 다음 절차를 수행하기 전에 서비스에 대한 매니페스트 파일을 만드십시오. 절차에서는 **echo** 서비스에 대한 매니페스트(**echo.sctp.xml**)를 예로 사용합니다.

# 1 시스템 파일에 대한 쓰기 권한이 있는 사용자 계정으로 로컬 시스템에 로그인합니다.

## 2 /etc/services 파일을 편집하고 새 서비스에 대한 정의를 추가합니다.

서비스 정의에 대한 다음 구문을 사용합니다.

```
service-name |port/protocol |aliases
```

## 3 새 서비스를 추가합니다.

서비스 매니페스트가 저장된 디렉토리로 이동하여 다음을 입력합니다.

```
# cd dir-name
# svccfg import service-manifest-name
```

svccfg의 전체 구문은 [svccfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

현재 service.dir 디렉토리에 있는 echo.sctp.xml 매니페스트를 사용하여 새 SCTP echo 서비스를 추가하려고 한다고 가정합니다. 다음을 입력합니다.

```
# cd service.dir
# svccfg import echo.sctp.xml
```

## 4 서비스 매니페스트가 추가되었는지 확인합니다.

```
# svcs FMRI
```

FMRI 인수로 서비스 매니페스트의 FMRI(Fault Managed Resource Identifier)를 사용합니다. 예를 들어, SCTP echo 서비스의 경우 다음 명령을 사용합니다.

```
# svcs svc:/network/echo:sctp_stream
```

출력이 다음과 유사하게 표시됩니다.

```
STATE      STIME      FMRI
disabled   16:17:00   svc:/network/echo:sctp_stream
```

svcs 명령에 대한 자세한 내용은 [svcs\(1\)](#) 매뉴얼 페이지를 참조하십시오.

출력은 새 서비스 매니페스트가 현재 사용 안함으로 설정되어 있음을 나타냅니다.

## 5 수정해야 할지 여부를 결정할 서비스의 등록 정보를 나열합니다.

```
# inetadm -l FMRI
```

inetadm 명령에 대한 자세한 내용은 [inetadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예를 들어, SCTP echo 서비스의 경우 다음을 입력합니다.

```
# inetadm -l svc:/network/echo:sctp_stream
SCOPE      NAME=VALUE
           name="echo"
           endpoint_type="stream"
           proto="sctp"
           isrpc=FALSE
           wait=FALSE
```

```

        exec="/usr/lib/inet/in.echod -s"
        .
        .
        default tcp_trace=FALSE
        default tcp_wrappers=FALSE

```

## 6 새 서비스를 사용으로 설정합니다.

```
# inetadm -e FMRI
```

## 7 서비스가 사용으로 설정되었는지 확인합니다.

예를 들어, 새 echo 서비스의 경우 다음을 입력합니다.

```

# inetadm | grep sctp_stream
.
.
    enabled    online          svc:/network/echo:sctp_stream

```

### 예 3-9 SCTP 전송 프로토콜을 사용하는 서비스 추가

다음 예에서는 사용할 명령과 echo 서비스가 SCTP 전송 계층 프로토콜을 사용하도록 하는 데 필요한 파일 항목을 보여 줍니다.

```

$ cat /etc/services
.
.
echo          7/tcp
echo          7/udp
echo          7/sctp

# cd service.dir

# svccfg import echo.sctp.xml

# svcs network/echo*
STATE          STIME      FMRI
disabled       15:46:44  svc:/network/echo:dgram
disabled       15:46:44  svc:/network/echo:stream
disabled       16:17:00  svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
SCOPE          NAME=VALUE
               name="echo"
               endpoint_type="stream"
               proto="sctp"
               isrpc=FALSE
               wait=FALSE
               exec="/usr/lib/inet/in.echod -s"
               user="root"
default        bind_addr=""
default        bind_fail_max=-1
default        bind_fail_interval=-1
default        max_con_rate=-1
default        max_copies=-1
default        con_rate_offline=-1

```



```

default failrate_cnt=40
default failrate_interval=60
default inherit_env=TRUE
default tcp_trace=FALSE
default tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
disabled disabled      svc:/network/echo:stream
disabled disabled      svc:/network/echo:dgram
enabled  online         svc:/network/echo:sctp_stream

```

## ▼ TCP 래퍼를 사용하여 TCP 서비스에 대한 액세스를 제어하는 방법

tcpd 프로그램은 TCP 래퍼를 구현합니다. TCP 래퍼는 데몬과 수신 서비스 요청 사이에서 서비스 데몬(예: ftpd)에 대한 보안 조치를 추가합니다. 또한 연결 시도 성공 및 실패를 기록합니다. TCP 래퍼는 요청 시작 위치에 따라 연결을 허용하거나 거부하여 액세스 제어를 제공할 수도 있습니다. TCP 래퍼를 사용하여 SSH, Telnet, FTP 등의 데몬을 보호할 수 있습니다. sendmail 응용 프로그램은 [Oracle Solaris 관리: 네트워크 서비스의 “sendmail 버전 8.12의 TCP 래퍼에 대한 지원”](#)에 설명된 대로 TCP 래퍼를 사용할 수 있습니다.

### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오.

### 2 TCP 래퍼를 사용으로 설정합니다.

```
# inetadm -M tcp_wrappers=TRUE
```

### 3 hosts\_access(3) 매뉴얼 페이지에 설명된 대로 TCP 래퍼 액세스 제어 정책을 구성합니다.

이 매뉴얼 페이지는 /usr/sfw/man 디렉토리에서 확인할 수 있습니다.



## 네트워크에서 IPv6 사용

---

이 장에서는 네트워크에서 IPv6을 사용하기 위한 작업에 대해 설명합니다. 다음 주요 항목을 다룹니다.

- 75 페이지 “IPv6 인터페이스 구성”
- 76 페이지 “IPv6에 대해 시스템을 구성하는 방법”
- 78 페이지 “IPv6 라우터 구성”
- 80 페이지 “호스트 및 서버에 대해 IPv6 인터페이스 구성 수정”
- 120 페이지 “터널 구성(작업 맵)”
- 86 페이지 “IPv6용 이름 서비스 지원 구성”

IPv6에 대한 다양한 유형의 정보는 다음 리소스를 참조하십시오.

- IPv6 개념에 대한 개요: **System Administration Guide: IP Services**의 3 장, “Introducing IPv6 (Overview)”
- IPv6 계획 작업: 2 장, “IPv6 주소 사용 시 고려 사항”
- IP 터널 사용을 위한 준비: 40 페이지 “네트워크에서 터널 사용 계획”
- 참조 정보: 9 장, “IPv6 참조”

## IPv6 인터페이스 구성

네트워크에서 IPv6을 사용하기 위한 초기 단계로, 시스템의 IP 인터페이스에서 IPv6을 구성하십시오.

Oracle Solaris 설치 프로세스 중 하나 이상의 시스템 인터페이스에서 IPv6을 사용으로 설정할 수 있습니다. 설치 중 IPv6 지원을 사용으로 설정한 경우에는 설치가 완료되면 다음과 같은 IPv6 관련 파일 및 테이블이 생성됩니다.

- `name-service/switch` SMF 서비스는 IPv6 주소를 사용하여 조회가 가능하도록 수정되었습니다.
- IPv6 주소 선택 정책 테이블이 생성됩니다. 이 테이블은 IPv6 지원 인터페이스를 통한 전송에 사용할 IP 주소 형식의 우선 순위를 정합니다.

이 절에서는 Oracle Solaris 설치가 완료된 후 인터페이스에서 IPv6을 사용으로 설정하는 방법에 대해 설명합니다.

## ▼ IPv6에 대해 시스템을 구성하는 방법

IPv6 노드로 사용될 모든 시스템의 인터페이스에서 IPv6을 사용으로 설정하여 IPv6 구성 프로세스를 시작하십시오. 처음에 인터페이스는 [System Administration Guide: IP Services](#)의 “IPv6 Address Autoconfiguration”에 설명된 대로 자동 구성 프로세스를 통해 IPv6 주소를 얻습니다. 그런 다음 IPv6 네트워크의 기능을 기준으로 노드의 구성을 호스트, 서버 또는 라우터로 조정합니다.

---

주 - 인터페이스가 현재 IPv6 접두어를 알리는 라우터와 동일한 링크에 있는 경우, 자동 구성된 주소의 일부로 해당 사이트의 접두어를 얻습니다. 자세한 내용은 [78 페이지 “IPv6 지원 라우터를 구성하는 방법”](#)을 참조하십시오.

---

다음 절차는 Oracle Solaris 설치 이후에 추가된 인터페이스에 대해 IPv6을 사용으로 설정하는 방법에 대해 설명합니다.

- 1 적합한 명령을 사용하여 IP 인터페이스를 구성합니다.  
[47 페이지 “IP 인터페이스 구성 방법”](#)을 참조하십시오.

---

주 - IP 주소를 지정할 경우 올바른 옵션을 사용하여 IPv6 주소를 지정해야 합니다.

```
# ipadm create-addr -T addrconf addrobj
```

주소를 더 추가하려면 다음 구문을 사용합니다.

```
# ipadm create-addr -T static ipv6-address addrobj
```

---

- 2 IPv6 데몬 `in.ndpd`를 시작합니다.  
`# /usr/lib/inet/in.ndpd`
- 3 (옵션) 정적 IPv6 기본 경로를 만듭니다.  
`# /usr/sbin/route -p add -inet6 default ipv6-address`
- 4 (옵션) 노드의 인터페이스 변수에 대한 매개변수를 정의하는 `/etc/inet/ndpd.conf` 파일을 만듭니다.

호스트의 인터페이스에 대해 임시 주소를 만들어야 하는 경우 [80 페이지 “인터페이스에 대해 임시 주소 사용”](#)을 참조하십시오. `/etc/inet/ndpd.conf`에 대한 자세한 내용은 `ndpd.conf(4)` 매뉴얼 페이지 및 [142 페이지 “ndpd.conf 구성 파일”](#)을 참조하십시오.

- 5 (옵션) IPv6 구성을 포함하는 IP 인터페이스의 상태를 표시하려면 다음 명령을 입력합니다.

```
# ipadm show-addr
```

#### 예 4-1 설치 후 IPv6 인터페이스 사용

이 예는 net0 인터페이스에서 IPv6을 사용으로 설정하는 방법을 보여줍니다. 시작하기 전에 시스템에 구성된 모든 인터페이스의 상태를 확인하십시오.

```
# ipadm show-addr
ADDROBJ    TYPE      STATE     ADDR
lo0/v4     static    ok        127.0.0.1/8
net0/v4     static    ok        172.16.27.74/24
```

현재 net0 인터페이스만이 이 시스템에 대해 구성되어 있습니다. 다음과 같이 이 인터페이스에서 IPv6을 사용으로 설정하십시오.

```
# ipadm create-addr -T addrconf net0/v6
# ipadm create-addr -T static -a 2001:db8:3c4d:15:203/64 net0/v6add
# /usr/lib/inet/in.ndpd
```

```
# ipadm show-addr
ADDROBJ    TYPE      STATE     ADDR
lo0/v4     static    ok        127.0.0.1/8
net0/v4     static    ok        172.16.27.74/24
net0/v6     addrconf  ok        fe80::203:baff:fe13:14e1/10
lo0/v6     static    ok        ::1/128
net0/v6add  static    ok        2001:db8:3c4d:15:203/64
```

```
# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

- 다음 순서
- IPv6 노드를 라우터로 구성하려면 78 페이지 “IPv6 라우터 구성”으로 이동합니다.
  - 노드에 대한 주소 자동 구성을 사용 안함으로 설정하려면 77 페이지 “IPv6 주소 자동 구성을 해제하는 방법”을 참조하십시오.
  - 노드를 서버로 조정하려면 85 페이지 “서버에서 IPv6 지원 인터페이스 관리”의 제안 사항을 참조하십시오.

## ▼ IPv6 주소 자동 구성을 해제하는 방법

일반적으로 호스트 및 서버의 인터페이스에 대한 IPv6 주소는 주소 자동 구성을 사용하여 생성해야 합니다. 그러나 83 페이지 “IPv6 토큰 구성”에 설명된 것과 같이, 특히 토큰을 수동으로 구성하려는 경우 주소 자동 구성을 해제할 수 있습니다.

### 1 노드에 대한 /etc/inet/ndpd.conf 파일을 만듭니다.

/etc/inet/ndpd.conf 파일은 특정 노드에 대한 인터페이스 변수를 정의합니다. 모든 서버의 인터페이스에 대한 주소 자동 구성을 해제하려면 이 파일에 다음과 같은 내용이 포함되어야 합니다.

```
if-variable-name StatelessAddrConf false
```

/etc/inet/ndpd.conf에 대한 자세한 내용은 [ndpd.conf\(4\)](#) 매뉴얼 페이지 및 [142 페이지 “ndpd.conf 구성 파일”](#)을 참조하십시오.

- 2 변경 사항으로 IPv6 데몬을 업데이트합니다.

```
# pkill -HUP in.ndpd
```

## IPv6 라우터 구성

이 절에서는 IPv6 라우터 구성 작업에 대해 설명합니다. 사이트 요구 사항에 따라 선택한 작업만 수행해야 할 수 있습니다.

### ▼ IPv6 지원 라우터를 구성하는 방법

다음 절차는 이미 IPv6에 대해 시스템을 구성했다고 가정합니다. 절차는 [75 페이지 “IPv6 인터페이스 구성”](#)을 참조하십시오.

- 1 라우터의 모든 인터페이스에서 IPv6 패킷 전송을 구성합니다.

```
# ipadm set-prop -p forwarding=on ipv6
```

- 2 경로 지정 데몬을 시작합니다.

in.ripngd 데몬은 IPv6 경로 지정을 처리합니다. IPv6 경로 지정은 다음 방법 중 하나로 설정합니다.

- routeadm 명령을 사용합니다.

```
# routeadm -e ipv6-routing -u
```

- 해당 SMF 명령을 사용합니다.

```
# svcadm enable ripng:default
```

routeadm 명령에 대한 구문 정보는 [routeadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오

- 3 /etc/inet/ndpd.conf 파일을 만듭니다.

라우터가 알릴 사이트 접두어 및 기타 구성 정보를 /etc/inet/ndpd.conf에 지정합니다. 이 파일은 IPv6 Neighbor Discovery 프로토콜을 구현하는 in.ndpd 데몬이 읽습니다.

변수 및 허용되는 값 목록은 [142 페이지 “ndpd.conf 구성 파일”](#) 및 [ndpd.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

- 4 /etc/inet/ndpd.conf 파일에 다음 텍스트를 입력합니다.

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on
```

이 텍스트는 IPv6용으로 구성된 라우터의 모든 인터페이스를 통해 라우터 알림을 전송하도록 in.ndpd에 지시합니다.

- 5 /etc/inet/ndpd.conf 파일에 추가 텍스트를 추가하여 라우터의 여러 인터페이스에 사이트 접두어를 구성합니다.

이 텍스트는 다음과 같은 형식이어야 합니다.

```
prefix global-routing-prefix:subnet ID/64 interface
```

다음 샘플 /etc/inet/ndpd.conf 파일은 net0 및 net1 인터페이스를 통해 사이트 접두어 2001:0db8:3c4d::/48을 알리도록 라우터를 구성합니다.

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on
```

```
if net0 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:15::0/64 net0
```

```
if net1 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:16::0/64 net1
```

- 6 시스템을 재부트합니다.

IPv6 라우터가 ndpd.conf 파일에 있는 사이트 접두어를 로컬 사이트에 알립니다.

#### 예 4-2 IPv6 주소를 표시하는 ipadm show-addr 출력

다음 예는 78 페이지 “IPv6 라우터 구성” 절차를 완료하면 표시되는 것과 같은, ipadm show-addr 명령의 출력을 보여줍니다.

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
net0/v4	static	ok	172.16.15.232/24
net1/v4	static	ok	172.16.16.220/24
net0/v6	addrconf	ok	fe80::203:baff:fe11:b115/10
lo0/v6	static	ok	::1/128
net0/v6add	static	ok	2001:db8:3c4d:15:203:baff:fe11:b115/64
net1/v6	addrconf	ok	fe80::203:baff:fe11:b116/10
net1/v6add	static	ok	2001:db8:3c4d:16:203:baff:fe11:b116/64

이 예에서 IPv6용으로 구성된 각 인터페이스가 이제 두 개의 주소를 사용합니다.

*interface/v6*과 같은 주소 객체 이름을 포함하는 항목에는 해당 인터페이스에 대한 링크 로컬 주소가 표시됩니다. *interface/v6add*와 같은 주소 객체 이름을 포함하는 항목에는 전역 IPv6 주소가 표시됩니다. 이 주소에는 인터페이스 ID 이외에도 /etc/ndpd.conf 파일에 구성된 사이트 접두어가 포함됩니다. *v6add* 대상은 무작위로 정의된 문자열입니다. *interface*가 IPv6 주소를 만들려는 인터페이스(예: *net0/mystring*, *net0/ipv6addr* 등)를 나타내는 경우, 다른 문자열을 정의하여 주소 객체 이름의 두번째 부분을 구성할 수 있습니다.

- 참조
- IPv6 네트워크 토폴로지에서 식별한 라우터에서 터널을 구성하려면 119 페이지 “dladm 명령을 통한 터널 구성 및 관리”를 참조하십시오.
  - 네트워크에서 스위치 및 허브 구성에 대한 자세한 내용은 제조업체의 설명서를 참조하십시오.

- IPv6 호스트를 구성하려면 80 페이지 “호스트 및 서버에 대해 IPv6 인터페이스 구성 수정”을 참조하십시오.
- 서버에서 IPv6 지원을 향상시키려면 85 페이지 “서버에서 IPv6 지원 인터페이스 관리”를 참조하십시오.
- IPv6 명령, 파일 및 데몬에 대한 자세한 내용은 141 페이지 “Oracle Solaris IPv6 구현”을 참조하십시오.

## 호스트 및 서버에 대해 IPv6 인터페이스 구성 수정

이 단원에서는 호스트 또는 서버인 노드에서 IPv6 지원 인터페이스의 구성을 수정하는 방법에 대해 설명합니다. 대부분의 경우 [System Administration Guide: IP Services](#)의 “[Stateless Autoconfiguration Overview](#)”에 설명된 대로 IPv6 지원 인터페이스에 대해 주소 자동 구성을 사용해야 합니다. 그러나 이 단원의 작업에 설명된 것과 같이, 필요한 경우 인터페이스의 IPv6 주소를 수정할 수 있습니다.

세 가지 일반 작업을 다음 순서로 수행해야 합니다.

1. IPv6 주소 자동 구성을 해제합니다. 77 페이지 “IPv6 주소 자동 구성을 해제하는 방법”을 참조하십시오.
2. 호스트에 대해 임시 주소를 만듭니다. 81 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.
3. 인터페이스 ID에 대해 IPv6 토큰을 구성합니다. 83 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”을 참조하십시오.

## 인터페이스에 대해 임시 주소 사용

IPv6 임시 주소에는 인터페이스의 MAC 주소 대신 무작위로 생성된 64비트 숫자가 인터페이스 ID로 포함됩니다. 익명으로 유지하려는 IPv6 노드의 인터페이스에 대해 임시 주소를 사용할 수 있습니다. 예를 들어 공개 웹 서버에 액세스해야 하는 호스트의 인터페이스에 대해 임시 주소를 사용할 수 있습니다. 임시 주소는 IPv6 프라이버시의 향상된 기능을 구현합니다. 이러한 향상된 기능은 “[Privacy Extensions for Stateless Address Autoconfiguration in IPv6](#)” (<http://www.ietf.org/rfc/rfc3041.txt?number=3041>)에서 제공하는 RFC 3041에 설명되어 있습니다.

필요한 경우 `/etc/inet/ndpd.conf` 파일에서 하나 이상의 인터페이스에 대해 임시 주소를 사용으로 설정할 수 있습니다. 그러나 자동 구성된 표준 IPv6 주소와 달리, 임시 주소는 64비트 서브넷 접두어와 무작위로 작성된 64비트 숫자로 구성됩니다. 이 무작위 숫자가 IPv6 주소의 인터페이스 ID 세그먼트가 됩니다. 임시 주소를 사용할 경우 링크 로컬 주소가 인터페이스 ID로 생성되지 않습니다.

임시 주소에는 기본 **선호 수명**(1일)이 지정됩니다. 임시 주소 생성을 사용으로 설정한 경우 `/etc/inet/ndpd.conf` 파일에서 다음 변수를 구성할 수도 있습니다.



<i>valid lifetime</i> TmpValidLifetime	호스트에서 주소가 삭제된 후 임시 주소가 존재하는 시간 범위입니다.
<i>preferred lifetime</i> TmpPreferredLifetime	임시 주소가 제거되기 전의 경과 시간입니다. 이 시간 범위는 유효 수명보다 짧아야 합니다.
<i>address regeneration</i>	선호 수명이 만료되기 이전 기간으로, 이 기간 동안 호스트에서 임시 주소를 새로 생성해야 합니다.

임시 주소의 기간은 다음과 같이 표시됩니다.

<i>n</i>	<i>n</i> 은 초 수입니다(기본값).
<i>n h</i>	<i>n</i> 은 시간(h) 수입니다.
<i>n d</i>	<i>n</i> 은 일(d) 수입니다.

## ▼ 임시 주소를 구성하는 방법

- 1 필요한 경우 호스트의 인터페이스에서 IPv6을 사용으로 설정합니다.

76 페이지 “IPv6에 대해 시스템을 구성하는 방법”을 참조하십시오.

- 2 `/etc/inet/ndpd.conf` 파일을 편집하여 임시 주소 생성을 설정합니다.

- 호스트의 모든 인터페이스에서 임시 주소를 구성하려면 `/etc/inet/ndpd.conf`에 다음 행을 추가합니다.

```
ifdefault TmpAddrsEnabled true
```

- 특정 인터페이스에 대해 임시 주소를 구성하려면 `/etc/inet/ndpd.conf`에 다음 행을 추가합니다.

```
if interface TmpAddrsEnabled true
```

- 3 (옵션) 임시 주소의 유효 수명을 지정합니다.

```
ifdefault TmpValidLifetime duration
```

이 구문은 호스트에 있는 모든 인터페이스의 유효 수명을 지정합니다. *duration*의 값은 초, 시간 또는 일 단위여야 합니다. 기본 유효 수명은 7일입니다. `TmpValidLifetime`을 `if interface` 키워드와 함께 사용하여 특정 인터페이스의 임시 주소에 대한 유효 수명을 지정할 수도 있습니다.

- 4 (옵션) 임시 주소의 선호 수명을 지정합니다. 이 기간이 경과하면 주소가 제거됩니다.

```
if interface TmpPreferredLifetime duration
```

이 구문은 특정 인터페이스의 임시 주소에 대한 선호 수명을 지정합니다. 기본 선호 수명은 1일입니다. `TmpPreferredLifetime`을 `ifdefault` 키워드와 함께 사용하여 호스트의 모든 인터페이스에서 임시 주소에 대한 선호 수명을 지정할 수도 있습니다.

주- 기본 주소 선택은 제거된 IPv6 주소에 낮은 우선 순위를 지정합니다. IPv6 임시 주소가 제거된 경우, 기본 주소 선택은 사용 가능한 주소를 패킷의 소스 주소로 선택합니다. 사용 가능한 주소는 자동으로 생성된 IPv6 주소 또는 인터페이스의 IPv4 주소일 수 있습니다. 기본 주소 선택에 대한 자세한 내용은 [108 페이지 “기본 주소 선택 관리”](#)를 참조하십시오.

- 5 (옵션) 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 이 시간 동안 호스트에서 임시 주소를 새로 생성해야 합니다.

**ifdefault TmpRegenAdvance duration**

이 구문은 호스트에 있는 모든 인터페이스의 임시 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 기본값은 5초입니다.

- 6 **in.ndpd** 데몬의 구성을 변경합니다.

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

- 7 **ipadm show-addr** 명령을 실행하여 임시 주소가 만들어졌는지 확인합니다. [예 4-4](#)를 참조하십시오.

명령 출력에서 임시 주소의 CURRENT 필드에 t 플래그가 표시됩니다.

#### 예 4-3 /etc/inet/ndpd.conf 파일의 임시 주소 변수

다음 예는 기본 네트워크 인터페이스에 대해 임시 주소가 사용으로 설정된 /etc/inet/ndpd.conf 파일의 세그먼트를 보여줍니다.

```
ifdefault TmpAddrsEnabled true

ifdefault TmpValidLifetime 14d

ifdefault TmpPreferredLifetime 7d

ifdefault TmpRegenAdvance 6s
```

#### 예 4-4 임시 주소가 사용으로 설정된 ipadm show-addr 명령 출력

이 예는 임시 주소가 생성된 후 netstat 명령의 출력을 보여줍니다. IPv6 관련 정보만 샘플 출력에 포함되어 있습니다.

```
# ipadm show-addr -o all
ADDROBJ  TYPE      STATE  CURRENT  PERSISTENT  ADDR
lo0/v6   static    ok     U----   ---         ::1/128
net0/v6   addrconf  ok     U----   ---         fe80::a00:20ff:feb9:4c54/10
net0/v6a  static    ok     U----   ---         2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?    addrconf  ok     U--t-   ---         2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

주소 객체 `net0/?`의 경우 `t` 플래그가 **CURRENT** 필드 아래에 설정되어 있습니다. 이 플래그는 해당 주소에 임시 인터페이스 ID가 있음을 나타냅니다.

- 참조
- IPv6 주소에 대한 이름 서비스 지원을 설정하려면 [86 페이지 “IPv6용 이름 서비스 지원 구성”](#)을 참조하십시오.
  - 서버에 대해 IPv6 주소를 구성하려면 [83 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”](#)을 참조하십시오.
  - IPv6 노드에 대한 작업을 모니터링하려면 [5 장, “TCP/IP 네트워크 관리”](#)를 참조하십시오.

## IPv6 토큰 구성

IPv6 주소의 64비트 인터페이스 ID를 **토큰**이라고 합니다. [System Administration Guide: IP Services](#)의 [“IPv6 Addressing Overview”](#)를 참조하십시오. 주소 자동 구성 중 토큰은 인터페이스의 MAC 주소와 연관됩니다. 대부분의 경우 비경로 지정 노드인 IPv6 호스트와 서버는 자동 구성된 토큰을 사용해야 합니다.

그러나 시스템 유지 관리의 일부로 인터페이스가 무작위로 교체되는 서버의 경우 자동 구성된 토큰을 사용하면 문제가 발생할 수 있습니다. 인터페이스 카드가 변경되면 MAC 주소도 변경됩니다. 그 결과 정적 IP 주소에 의존하는 서버에서 문제가 발생할 수 있습니다. 네트워크 기반구조의 여러 부분(예: DNS 또는 NIS)에 서버의 인터페이스에 대한 특정 IPv6 주소가 저장되었을 수 있습니다.

주소 변경 문제를 방지하려면 IPv6 주소에서 인터페이스 ID로 사용할 토큰을 수동으로 구성하면 됩니다. 토큰을 만들려면 IPv6 주소의 인터페이스 ID 부분을 차지할 64비트 이하의 16진수를 지정하십시오. 이후 주소 자동 구성 중 Neighbor Discovery는 인터페이스의 MAC 주소를 기반으로 하는 인터페이스 ID를 만들지 않습니다. 대신 수동으로 생성된 토큰이 인터페이스 ID가 됩니다. 이 토큰은 카드가 교체된 후에도 계속 인터페이스에 지정되어 있습니다.

---

주 - 사용자 지정 토큰과 임시 주소의 차이점은 임시 주소는 사용자가 명시적으로 만드는 것이 아니라 무작위로 생성된다는 점입니다.

---

### ▼ 사용자 지정 IPv6 토큰을 구성하는 방법

다음 지침은 인터페이스가 자주 교체되는 서버에 특히 유용합니다. 또한 IPv6 노드에서 사용자 지정 토큰을 구성하는 경우에도 유효합니다.

- 1 토큰을 사용하여 구성할 인터페이스가 존재하며 인터페이스에 IPv6 주소가 구성되지 않았는지 확인합니다.

---

주 - 인터페이스에 구성된 IPv6 주소가 없는지 확인하십시오.

---

```
# ipadm show-if
IFNAME  CLASS      STATE  ACTIVE  OVER
lo0     loopback   ok     yes     ---
net0    ip         ok     yes     ---

# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/v4   static    ok     127.0.0.1/8
```

이 출력은 네트워크 인터페이스 `net0`이 구성된 IPv6 주소 없이 존재함을 보여줍니다.

2. 노드 인터페이스에 대한 토큰으로 사용할 64비트 16진수를 하나 이상 만듭니다. 토큰 예는 [System Administration Guide: IP Services](#)의 “Link-Local Unicast Address”를 참조하십시오.

3. 토큰을 사용하여 각 인터페이스를 구성합니다.

각 인터페이스에 대해 다음 형식의 `ipadm` 명령을 사용하여 사용자 정의 인터페이스 ID(토큰)를 생성합니다.

```
# ipadm create-addr -T addrconf -i interface-ID addrobj
```

예를 들어, 다음 명령으로 토큰을 포함하는 `net0` 인터페이스를 구성할 수 있습니다.

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0/v6add
```

---

주 - 토큰을 사용하여 주소 객체가 생성되면 더 이상 토큰을 수정할 수 없습니다.

---

4. 변경 사항으로 IPv6 데몬을 업데이트합니다.

```
# pkill -HUP in.ndpd
```

#### 예 4-5 IPv6 인터페이스에서 사용자 지정 토큰 구성

다음 예는 IPv6 주소 및 토큰으로 `net0`이 구성됨을 보여줍니다.

```
# ipadm show-if
IFNAME  CLASS      STATE  ACTIVE  OVER
lo0     loopback   ok     yes     ---
net0    ip         ok     yes     ---

# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/v4   static    ok     127.0.0.1/8

# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0/v6
# pkill -HUP in.ndpd
```

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v6       static    ok         ::1/128
net0/v6       addrconf  ok         fe80::1a:2b:3c:4d/10
net0/v6       addrconf  ok         2002:a08:39f0:1:1a:2b:3c:4d/64
```

토큰이 구성되면 주소 객체 `net0/v6`에 링크 로컬 주소와 인터페이스 ID에 대해 구성된 `1a:2b:3c:4d` 주소가 생깁니다. `net0/v6`이 생성된 후에는 더 이상 이 인터페이스에 대해 이 토큰을 수정할 수 없습니다.

- 참조
- 서버의 IPv6 주소로 이름 서비스를 업데이트하려면 [86 페이지 “IPv6용 이름 서비스 지원 구성”](#)을 참조하십시오.
  - 서버 성능을 모니터링하려면 [5 장, “TCP/IP 네트워크 관리”](#)를 참조하십시오.

## 서버에서 IPv6 지원 인터페이스 관리

서버에서 IPv6을 계획한 경우 서버 인터페이스에서 IPv6을 사용으로 설정했으므로 몇 가지 사항을 결정해야 합니다. 이러한 결정 사항은 인터페이스 IPv6 주소의 인터페이스 ID(토큰이라고도 함)를 구성하는 데 사용할 전략에 영향을 미칩니다.

### ▼ 서버 인터페이스에서 IPv6을 사용으로 설정하는 방법

이 절차는 네트워크 서버에서 IPv6을 사용으로 설정하는 일반적인 단계를 제공합니다. IPv6 구현 방식에 따라 몇 가지 단계는 다를 수 있습니다.

- 1 서버의 IP 인터페이스에서 IPv6을 사용으로 설정합니다.  
절차는 [75 페이지 “IPv6 인터페이스 구성”](#)을 참조하십시오.
- 2 서버와 동일한 링크에 있는 라우터에서 IPv6 서브넷 접두어가 구성되었는지 확인합니다.  
자세한 내용은 [78 페이지 “IPv6 라우터 구성”](#)을 참조하십시오.
- 3 서버 IPv6 지원 인터페이스의 인터페이스 ID에 적합한 전략을 사용합니다.  
기본적으로 IPv6 주소 자동 구성은 IPv6 주소의 인터페이스 ID 부분을 만들 때 인터페이스의 MAC 주소를 사용합니다. 인터페이스의 IPv6 주소가 잘 알려진 주소일 경우 한 인터페이스를 다른 인터페이스로 교체하면 문제가 발생할 수 있습니다. 새 인터페이스의 MAC 주소는 다릅니다. 주소 자동 구성 중 토큰은 새 인터페이스 ID가 생성됩니다.
  - 교체하지 않으려는 IPv6 지원 인터페이스의 경우, 자동 구성된 IPv6 주소를 사용합니다. [System Administration Guide: IP Services](#)의 [“IPv6 Address Autoconfiguration”](#)을 참조하십시오.

- 로컬 네트워크 외부에 익명으로 표시되어야 하는 IPv6 지원 인터페이스의 경우, 무작위로 생성된 토큰을 인터페이스 ID로 사용합니다. 지침 및 예제는 [81 페이지 “임시 주소를 구성하는 방법”](#)을 참조하십시오.
- 정기적으로 교체하려는 IPv6 기반 인터페이스의 경우, 인터페이스 ID에 대한 토큰을 만듭니다. 지침 및 예제는 [83 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”](#)을 참조하십시오.

## IPv6용 이름 서비스 지원 구성

이 절에서는 IPv6 서비스를 지원하도록 DNS 및 NIS 이름 서비스를 구성하는 방법에 대해 설명합니다.

---

주 - LDAP은 IPv6 관련 구성 작업 없이 IPv6을 지원합니다.

---

DNS, NIS 및 LDAP 관리에 대한 자세한 내용은 [Oracle Solaris Administration: Naming and Directory Services](#)를 참조하십시오.

### ▼ DNS에 IPv6 주소를 추가하는 방법

- 1 IPv6 지원 노드마다 AAAA 레코드를 추가하여 해당 DNS 영역 파일을 편집합니다.

```
hostname IN AAAA host-address
```

- 2 DNS 역순 영역 파일을 편집하고 PTR 레코드를 추가합니다.

```
hostaddress IN PTR hostname
```

DNS 관리에 대한 자세한 내용은 [Oracle Solaris Administration: Naming and Directory Services](#)를 참조하십시오.

#### 예 4-6 DNS 역순 영역 파일

이 예는 역순 영역 파일의 IPv6 주소를 보여줍니다.

```
$ORIGIN ip6.int.
8.2.5.0.2.1.e.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.2.0.0.0 \
IN PTR vallejo.Eng.apex.COM.
```

## ▼ IPv6 이름 서비스 정보를 표시하는 방법

nslookup 명령을 사용하여 IPv6 이름 서비스 정보를 표시할 수 있습니다.

- 1 사용자 계정으로 nslookup 명령을 실행합니다.

```
% /usr/sbin/nslookup
```

기본 서버 이름과 주소가 표시되고, 이어서 nslookup 명령의 꺾쇠 괄호 프롬프트가 표시됩니다.

- 2 꺾쇠 괄호 프롬프트에 다음 명령을 입력하여 특정 호스트에 대한 정보를 확인합니다.

```
>set q=any
>hostname
```

- 3 AAAA 레코드만 확인하려면 다음 명령을 입력합니다.

```
>set q=AAAA
hostname
```

- 4 exit를 입력하여 nslookup 명령을 종료합니다.

### 예 4-7 nslookup 명령으로 IPv6 정보 표시

이 예는 IPv6 네트워크 환경에서 nslookup의 결과를 보여줍니다.

```
% /usr/sbin/nslookup
Default Server: dnsserve.local.com
Address: 10.10.50.85
> set q=AAAA
> host85
Server: dnsserve.local.com
Address: 10.10.50.85

host85.local.com      IPv6 address = 2::9256:a00:fe12:528
> exit
```

## ▼ DNS IPv6 PTR 레코드가 올바르게 업데이트되었는지 확인하는 방법

이 절차에서는 nslookup 명령을 사용하여 DNS IPv6용 PTR 레코드를 표시합니다.

- 1 사용자 계정으로 nslookup 명령을 실행합니다.

```
% /usr/sbin/nslookup
```

기본 서버 이름과 주소가 표시되고, 이어서 nslookup 명령의 꺾쇠 괄호 프롬프트가 표시됩니다.

- 2 꺾쇠 괄호 프롬프트에 다음을 입력하여 PTR 레코드를 표시합니다.

```
>set q=PTR
```

- 3 **exit**를 입력하여 명령을 종료합니다.

#### 예 4-8 nslookup 명령으로 PTR 레코드 표시

다음 예는 nslookup 명령으로 표시되는 PTR 레코드를 보여줍니다.

```
% /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  192.168.15.78
> set q=PTR
> 8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.2.0.0.0.ip6.int

8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.2.0.0.0.ip6.int name =
vallejo.ipv6.Eng.apex.COM
ip6.int nameserver = space1999.Eng.apex.COM
> exit
```

## ▼ NIS를 통해 IPv6 정보를 표시하는 방법

이 절차에서는 ypmatch 명령을 사용하여 NIS를 통해 IPv6 정보를 표시합니다.

- 사용자 계정으로 다음을 입력하여 NIS에 IPv6 주소를 표시합니다.

```
% ypmatch hostname hosts .byname
```

지정된 *hostname*에 대한 정보가 표시됩니다.



## TCP/IP 네트워크 관리

---

이 장에서는 TCP/IP 네트워크 관리 작업에 대해 설명합니다. 다음 항목을 다룹니다.

- 90 페이지 “주요 TCP/IP 관리 작업(작업 맵)”
- **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 및 주소 모니터링”**
- 91 페이지 “netstat 명령으로 네트워크 상태 모니터링”
- 97 페이지 “ping 명령으로 원격 호스트 확인”
- 99 페이지 “네트워크 상태 화면 관리 및 기록”
- 101 페이지 “traceroute 명령으로 경로 지정 정보 표시”
- 102 페이지 “snoop 명령으로 패킷 전송 모니터링”
- 108 페이지 “기본 주소 선택 관리”

---

주 - 네트워크 인터페이스를 모니터링하려면 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 및 주소 모니터링”**을 참조하십시오.

---

이 작업은 사용자의 사이트에서 TCP/IP 네트워크 즉, IPv4 전용 또는 듀얼 스택 IPv4/IPv6이 작동 가능하다고 가정합니다. 사이트에서 IPv6을 구현하려는 경우 다음 장에서 자세한 내용을 참조하십시오.

- IPv6 구현을 계획하려면 2 장, “IPv6 주소 사용 시 고려 사항”을 참조하십시오.
- IPv6을 구성하고 듀얼 스택 네트워크 환경을 만들려면 4 장, “네트워크에서 IPv6 사용”을 참조하십시오.

## 주요 TCP/IP 관리 작업(작업 맵)

다음 표는 초기 구성 후 네트워크를 관리하기 위한 기타 작업(예: 네트워크 정보 표시)을 보여줍니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	정보
프로토콜별 통계 표시	특정 시스템에서 네트워크 프로토콜의 성능을 모니터링합니다.	91 페이지 “프로토콜별 통계를 표시하는 방법”
네트워크 상태 표시	소켓 및 경로 설정표 항목을 모두 표시하여 시스템을 모니터링합니다. 출력에는 IPv4에 대한 주소 그룹과 IPv6에 대한 inet6 주소 그룹이 포함됩니다.	94 페이지 “소켓 상태를 표시하는 방법”
네트워크 인터페이스의 상태 표시	네트워크 인터페이스의 성능을 모니터링합니다. 이는 전송 문제를 해결하는 데 유용합니다.	93 페이지 “네트워크 인터페이스 상태를 표시하는 방법”
패킷 전송 상태 표시	회선을 통해 전송되는 패킷의 상태를 모니터링합니다.	95 페이지 “특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법”
IPv6 관련 명령의 화면 출력 제어	ping, netstat 및 traceroute 명령의 출력을 제어합니다. inet_type이라는 파일을 만들고, 이 파일에서 DEFAULT_IP 변수를 설정합니다.	99 페이지 “IP 관련 명령의 화면 출력을 제어하는 방법”
네트워크 트래픽 모니터링	snoop 명령을 사용하여 모든 IP 패킷을 표시합니다.	105 페이지 “IPv6 네트워크 트래픽을 모니터링하는 방법”
네트워크 라우터에 알려진 모든 경로 추적	traceroute 명령을 사용하여 모든 경로를 표시합니다.	102 페이지 “모든 경로를 추적하는 방법”

주 - 네트워크 인터페이스를 모니터링하려면 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 및 주소 모니터링”**을 참조하십시오.

## netstat 명령으로 네트워크 상태 모니터링

netstat 명령은 네트워크 상태 및 프로토콜 통계를 표시하는 화면을 생성합니다. TCP, SCTP 및 UDP 끝점을 표 형식으로 표시할 수 있습니다. 경로 설정표 정보 및 인터페이스 정보를 표시할 수도 있습니다.

netstat 명령은 선택한 명령줄 옵션에 따라 다양한 유형의 네트워크 데이터를 표시합니다. 이러한 표시는 시스템 관리에 가장 유용합니다. netstat의 기본 구문은 다음과 같습니다.

```
netstat [-m] [-n] [-s] [-i | -r] [-f address-family]
```

이 절에서는 가장 일반적으로 사용되는 netstat 명령의 옵션에 대해 설명합니다. 모든 netstat 옵션에 대한 자세한 설명은 [netstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### ▼ 프로토콜별 통계를 표시하는 방법

netstat -s 옵션은 UDP, TCP, SCTP, ICMP 및 IP 프로토콜에 대한 프로토콜 통계를 표시합니다.

---

주 - Oracle Solaris 사용자 계정을 사용하여 netstat 명령의 출력을 표시할 수 있습니다.

---

#### ● 프로토콜 상태를 표시합니다.

```
$ netstat -s
```

#### 예 5-1 네트워크 프로토콜 통계

다음 예제는 netstat -s 명령의 출력을 보여줍니다. 출력의 일부는 잘렸습니다. 출력은 프로토콜에 문제가 있는 영역을 나타낼 수 있습니다. 예를 들어 ICMPv4 및 ICMPv6의 통계 정보는 ICMP 프로토콜에서 오류가 발견된 위치를 나타낼 수 있습니다.

```
RAWIP
      rawipInDatagrams    = 4701      rawipInErrors      = 0
      rawipInChecksumErrs = 0        rawipOutDatagrams  = 4
      rawipOutErrors      = 0

UDP
      udpInDatagrams      = 10091     udpInErrors        = 0
      udpOutDatagrams     = 15772     udpOutErrors       = 0

TCP
      tcpRtoAlgorithm     = 4          tcpRtoMin          = 400
      tcpRtoMax           = 60000     tcpMaxConn         = -1
      .
      tcpListenDrop       = 0          tcpListenDropQ0    = 0
      tcpHalfOpenDrop     = 0          tcpOutSackRetrans  = 0
```

```

IPv4  ipForwarding      =    2      ipDefaultTTL      =   255
      ipInReceives   = 300182    ipInHdrErrors      =    0
      ipInAddrErrors =    0      ipInCksumErrs      =    0
      .
      ipsecInFailed   =    0      ipInIPv6           =    0
      ipOutIPv6       =    3      ipOutSwitchIPv6    =    0

IPv6  ipv6Forwarding     =    2      ipv6DefaultHopLimit =   255
      ipv6InReceives  = 13986    ipv6InHdrErrors     =    0
      ipv6InTooBigErrors =    0    ipv6InNoRoutes      =    0
      .
      rawipInOverflows =    0      ipv6InIPv4         =    0
      ipv6OutIPv4      =    0      ipv6OutSwitchIPv4   =    0

ICMPv4 icmpInMsgs         = 43593    icmpInErrors        =    0
      icmpInCksumErrs =    0      icmpInUnknowns      =    0
      .
      icmpInOverflows =    0

ICMPv6 icmp6InMsgs        = 13612    icmp6InErrors        =    0
      icmp6InDestUnreachs =    0    icmp6InAdminProhibs  =    0
      .
      icmp6OutGroupQueries =    0      icmp6OutGroupResps   =    2
      icmp6OutGroupReds   =    0

IGMP:
      12287 messages received
          0 messages received with too few bytes
          0 messages received with bad checksum
      12287 membership queries received

SCTP  sctpRtoAlgorithm    =  vanj
      sctpRtoMin       =  1000
      sctpRtoMax       = 60000
      sctpRtoInitial   =  3000
      sctpTimHearBeatProbe =    2
      sctpTimHearBeatDrop =    0
      sctpListenDrop   =    0
      sctpInClosed     =    0

```

## ▼ 전송 프로토콜의 상태를 표시하는 방법

netstat 명령을 통해 전송 프로토콜의 상태를 표시할 수 있습니다. 자세한 내용은 [netstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

- 1 시스템에서 TCP 및 SCTP 전송 프로토콜의 상태를 표시합니다.

```
$ netstat
```

- 2 시스템에서 특정 전송 프로토콜의 상태를 표시합니다.

```
$ netstat -P transport-protocol
```

*transport-protocol* 변수의 값은 tcp, sctp 또는 udp입니다.

## 예 5-2 TCP 및 SCTP 전송 프로토콜의 상태 표시

이 예는 기본 netstat 명령의 출력을 보여줍니다. IPv4 전용 정보가 표시됩니다.

\$ netstat

TCP: IPv4

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
lhost-1.login	abc.def.local.Sun.COM.980	49640	0	49640	0	ESTABLISHED
lhost-1.login	ghi.jkl.local.Sun.COM.1020	49640	1	49640	0	ESTABLISHED
remhost-1.1014	mno.pqr.remote.Sun.COM.nfsd	49640	0	49640	0	TIME_WAIT

SCTP:

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	StrsI/O	State
*.echo	0.0.0.0	0	0	102400	0	128/1	LISTEN
*.discard	0.0.0.0	0	0	102400	0	128/1	LISTEN
*.9001	0.0.0.0	0	0	102400	0	128/1	LISTEN

## 예 5-3 특정 전송 프로토콜의 상태 표시

이 예는 netstat 명령의 -P 옵션을 지정한 경우에 표시되는 결과를 보여줍니다.

\$ netstat -P tcp

TCP: IPv4

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
lhost-1.login	abc.def.local.Sun.COM.980	49640	0	49640	0	ESTABLISHED
lhost.login	ghi.jkl.local.Sun.COM.1020	49640	1	49640	0	ESTABLISHED
remhost.1014	mno.pqr.remote.Sun.COM.nfsd	49640	0	49640	0	TIME_WAIT

TCP: IPv6

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State	If
localhost.38983	localhost.32777	49152	0	49152	0	ESTABLISHED	
localhost.32777	localhost.38983	49152	0	49152	0	ESTABLISHED	
localhost.38986	localhost.38980	49152	0	49152	0	ESTABLISHED	

## ▼ 네트워크 인터페이스 상태를 표시하는 방법

netstat 명령의 i 옵션은 로컬 시스템에 구성된 네트워크 인터페이스의 상태를 보여줍니다. 이 옵션을 사용하면 시스템이 각 네트워크에서 전송하고 수신하는 패킷 수를 확인할 수 있습니다.

- 네트워크 인터페이스의 상태를 표시합니다.

\$ netstat -i

## 예 5-4 네트워크 인터페이스 상태 표시

다음 예제는 호스트 인터페이스를 통한 IPv4 및 IPv6 패킷 흐름의 상태를 보여줍니다.

예를 들어 서버에 대해 표시되는 입력 패킷 수(Ipkts)는 클라이언트를 부트하려고 할 때마다 늘어나지만, 출력 패킷 수(Opkts)는 그대로 유지됩니다. 이 출력에는 서버가 클라이언트에서 보내는 부트 요청 패킷을 파악하고 있는 것으로 표시됩니다. 그러나 서버가 이에 응답하는 방법을 알지 못합니다. 이러한 혼동은 hosts 또는 ethers 데이터베이스의 주소가 잘못되었기 때문일 수 있습니다.

그러나 시간이 경과해도 입력 패킷 수가 일정할 경우 시스템에서는 패킷을 전혀 알지 못합니다. 이 출력은 다른 유형의 오류(하드웨어 문제)를 보여줍니다.

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
lo0	8232	loopback	localhost	142	0	142	0	0	0
net0	1500	host58	host58	1106302	0	52419	0	0	0

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis
lo0	8252	localhost	localhost	142	0	142	0	0
net0	1500	fe80::a00:20ff:feb9:4c54/10	fe80::a00:20ff:feb9:4c54	1106305	0	52422	0	0

## ▼ 소켓 상태를 표시하는 방법

netstat 명령의 -a 옵션을 사용하여 로컬 호스트에 있는 소켓의 상태를 확인할 수 있습니다.

- 소켓 및 경로 설정표 항목의 상태를 표시하려면 다음을 입력합니다.

사용자 계정을 사용하여 netstat의 이 옵션을 실행할 수 있습니다.

```
% netstat -a
```

## 예 5-5 모든 소켓 및 경로 설정표 항목 표시

netstat -a 명령의 출력은 광범위한 통계를 표시합니다. 다음 예는 일반적인 netstat -a 출력의 일부분을 보여줍니다.

```
UDP: IPv4
  Local Address      Remote Address      State
-----
*.bootpc            Idle
host85.bootpc       Idle
*.                  Unbound
*.                  Unbound
*.sunrpc             Idle
*.                  Unbound
*.32771              Idle
*.sunrpc             Idle
*.                  Unbound
*.32775              Idle
*.time              Idle
```

```

      .
      .
      *.daytime                Idle
      *.echo                   Idle
      *.discard                 Idle
      .

UDP: IPv6
      Local Address              Remote Address          State    If
      -----
      *.*                       Unbound
      *.*                       Unbound
      *.sunrpc                  Idle
      *.*                       Unbound
      *.32771                   Idle
      *.32778                   Idle
      *.syslog                  Idle
      .
      .

TCP: IPv4
      Local Address              Remote Address          Swind Send-Q Rwind Recv-Q  State
      -----
      *.*                       *.*                    0      0 49152    0  IDLE
      localhost.4999            *.*                    0      0 49152    0  LISTEN
      *.sunrpc                  *.*                    0      0 49152    0  LISTEN
      *.*                       *.*                    0      0 49152    0  IDLE
      *.sunrpc                  *.*                    0      0 49152    0  LISTEN
      .
      .
      *.printer                 *.*                    0      0 49152    0  LISTEN
      *.time                    *.*                    0      0 49152    0  LISTEN
      *.daytime                 *.*                    0      0 49152    0  LISTEN
      *.echo                    *.*                    0      0 49152    0  LISTEN
      *.discard                 *.*                    0      0 49152    0  LISTEN
      *.chargen                 *.*                    0      0 49152    0  LISTEN
      *.shell                   *.*                    0      0 49152    0  LISTEN
      *.shell                   *.*                    0      0 49152    0  LISTEN
      *.kshell                  *.*                    0      0 49152    0  LISTEN
      *.login                   *.*                    0      0 49152    0  LISTEN
      .
      .
      *.*                       0      0 49152    0  LISTEN
      .

*TCP: IPv6
      Local Address              Remote Address          Swind Send-Q Rwind Recv-Q  State If
      -----
      *.*                       *.*                    0      0 49152    0  IDLE
      *.sunrpc                  *.*                    0      0 49152    0  LISTEN
      *.*                       *.*                    0      0 49152    0  IDLE
      *.32774                   *.*                    0      0 49152

```

## ▼ 특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법

netstat 명령의 -f 옵션을 사용하면 특정 주소 그룹의 패킷 전송과 관련된 통계를 표시할 수 있습니다.

● IPv4 또는 IPv6 패킷 전송에 대한 통계를 표시합니다.

\$ netstat -f inet | inet6

IPv4 전송 정보를 표시하려면 inet을 netstat -f에 대한 인수로 입력합니다. IPv6 정보를 표시하려면 inet6를 netstat -f에 대한 인수로 사용합니다.

예 5-6 IPv4 패킷 전송 상태

다음 예는 netstat -f inet 명령의 출력을 보여줍니다.

TCP: IPv4

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
host58.734	host19.nfsd	49640	0 49640		0	ESTABLISHED
host58.38063	host19.32782	49640	0 49640		0	CLOSE_WAIT
host58.38146	host41.43601	49640	0 49640		0	ESTABLISHED
host58.996	remote-host.login	49640	0 49206		0	ESTABLISHED

예 5-7 IPv6 패킷 전송 상태

다음 예는 netstat -f inet6 명령의 출력을 보여줍니다.

TCP: IPv6

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State	If
localhost.38065	localhost.32792	49152	0 49152		0	ESTABLISHED	
localhost.32792	localhost.38065	49152	0 49152		0	ESTABLISHED	
localhost.38089	localhost.38057	49152	0 49152		0	ESTABLISHED	

▼ 알려진 경로의 상태를 표시하는 방법

netstat 명령의 -r 옵션은 로컬 호스트의 경로 설정표를 표시합니다. 이 표는 호스트에 알려진 모든 경로의 상태를 보여줍니다. 사용자 계정에서 netstat의 이 옵션을 실행할 수 있습니다.

● IP 경로 설정표를 표시합니다.

\$ netstat -r

예 5-8 netstat 명령에 의한 경로 설정표 출력

다음 예는 netstat -r 명령의 출력을 보여줍니다.

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
host15	myhost	U	1	31059	net0
10.0.0.14	myhost	U	1	0	net0
default	distantrouter	UG	1	2	net0



localhost	localhost	UH	42019361	lo0	
Routing Table: IPv6					
Destination/Mask	Gateway	Flags	Ref	Use	If
2002:0a00:3010:2::/64	2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd	U	1	0	net0:1
fe80::/10	fe80::1a2b:3c4d:5e6f:12a2	U	1	23	net0
ff00::/8	fe80::1a2b:3c4d:5e6f:12a2	U	1	0	net0
default	fe80::1a2b:3c4d:5e6f:12a2	UG	1	0	net0
localhost	localhost	UH	9	21832	lo0

다음 표는 `netstat -r` 명령의 화면 출력에 표시되는 여러 매개변수의 의미에 대해 설명합니다.

매개변수	설명
Destination	경로의 대상 끝점인 호스트를 지정합니다. IPv6 경로 설정표는 6to4 터널 끝점(2002:0a00:3010:2::/64)의 접두어를 경로 대상 끝점으로 표시합니다.
Destination/Mask	
Gateway	패킷 전송에 사용할 게이트웨이를 지정합니다.
Flags	경로의 현재 상태를 나타냅니다. U 플래그는 경로가 작동 중임을 나타냅니다. G 플래그는 경로가 게이트웨이임을 나타냅니다.
Use	전송된 패킷 수를 표시합니다.
Interface	전송의 소스 끝점인 로컬 호스트의 특정 인터페이스를 나타냅니다.

## ping 명령으로 원격 호스트 확인

ping 명령으로 원격 호스트의 상태를 확인할 수 있습니다. ping을 실행하면 ICMP 프로토콜에서 지정된 호스트로 데이터그램을 전송하여 응답을 요청합니다. ICMP는 TCP/IP 네트워크에서 오류 처리를 담당하는 프로토콜입니다. ping을 사용하면 지정된 원격 호스트에 대한 IP 연결이 있는지 확인할 수 있습니다.

다음은 ping의 기본 구문입니다.

```
/usr/sbin/ping host [timeout]
```

이 구문에서 *host*는 원격 호스트의 이름입니다. 선택적 *timeout* 인수는 ping 명령이 계속해서 원격 호스트에 연결하려고 시도하는 시간(초)을 나타냅니다. 기본값은 20초입니다. 추가 구문 및 옵션은 [ping\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### ▼ 원격 호스트가 실행 중인지 확인하는 방법

- ping 명령을 다음과 같은 형식으로 입력합니다.

```
$ ping hostname
```

*hostname* 호스트가 ICMP 전송을 허용하는 경우 다음 메시지가 표시됩니다.

```
hostname is alive
```

이 메시지는 *hostname*이 ICMP 요청에 응답함을 나타냅니다. 그러나 *hostname*이 작동 중지되었거나 ICMP 패킷을 수신할 수 없는 경우, ping 명령으로부터 다음과 같은 응답을 수신합니다.

```
no answer from hostname
```

## ▼ 원격 호스트가 패킷을 삭제하는 중인지 확인하는 방법

ping 명령의 -s 옵션을 사용하여 원격 호스트가 실행 중이지만 패킷이 손실되고 있는지 확인할 수 있습니다.

- ping 명령을 다음과 같은 형식으로 입력합니다.

```
$ ping -s hostname
```

### 예 5-9 패킷 삭제를 발견하기 위한 ping 출력

ping -s *hostname* 명령은 사용자가 인터럽트 문자를 전송하거나 시간 초과가 발생할 때까지 계속해서 패킷을 지정된 호스트로 전송합니다. 다음과 같은 응답이 화면에 표시됩니다.

```
& ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms
```

```
^C
```

```
---host1.domain8 PING Statistics---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

패킷 손실 통계는 호스트에서 패킷이 삭제되었는지 여부를 나타냅니다. ping이 실패할 경우, ipadm 및 netstat 명령으로 보고되는 네트워크 상태를 확인하십시오. [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 및 주소 모니터링” 및 91 페이지 “netstat 명령으로 네트워크 상태 모니터링”](#)을 참조하십시오.

# 네트워크 상태 화면 관리 및 기록

다음 작업은 잘 알려진 네트워킹 명령을 사용하여 네트워크의 상태를 확인하는 방법을 보여줍니다.

## ▼ IP 관련 명령의 화면 출력을 제어하는 방법

IPv4 정보만 표시하거나 IPv4와 IPv6 정보를 모두 표시하도록 `netstat` 명령의 출력을 제어할 수 있습니다.

- 1 `/etc/default/inet_type` 파일을 만듭니다.
- 2 다음 항목 중에서 네트워크에 필요한 항목을 `/etc/default/inet_type`에 추가합니다.

- IPv4 정보만 표시

```
DEFAULT_IP=IP_VERSION4
```

- IPv4 및 IPv6 정보 모두 표시

```
DEFAULT_IP=BOTH
```

또는

```
DEFAULT_IP=IP_VERSION6
```

`inet_type` 파일에 대한 자세한 내용은 [inet\\_type\(4\)](#) 매뉴얼 페이지를 참조하십시오.

---

주 - `netstat` 명령의 `-f` 플래그는 `inet_type` 파일에 설정된 값을 대체합니다.

---

### 예 5-10 IPv4 및 IPv6 정보를 선택하도록 출력 제어

- `inet_type` 파일에 `DEFAULT_IP=BOTH` 또는 `DEFAULT_IP=IP_VERSION6` 변수를 지정할 경우 다음과 같이 출력되어야 합니다.

```
% ipadm show-addr
ADDROBJ    TYPE      STATE    ADDR
lo0/v4      static    ok       127.0.0.1/8
net0/v4      static    ok       10.46.86.54/24
lo0/v6      static    ok       ::1/128
net0/v6      addrconf  ok       fe80::a00:fe73:56a8/10
net0/v6add   static    ok       2001:db8:3c4d:5:a00:fe73:56a8/64
```

- `inet_type` 파일에 `DEFAULT_IP=IP_VERSION4` 변수를 지정할 경우 다음과 같이 출력되어야 합니다.

```
% ipadm show-addr
ADDROBJ    TYPE      STATE    ADDR
lo0/v4      static    ok       127.0.0.1/8
net0/v4      static    ok       10.46.86.54/24
```

## ▼ IPv4 경로 지정 데몬의 작업을 기록하는 방법

IPv4 경로 지정 데몬인 `routed`의 오작동이 의심되는 경우 데몬의 작업을 추적하는 로그를 시작할 수 있습니다. `routed` 데몬이 시작되면 이 로그에는 모든 패킷 전송이 포함됩니다.

- 경로 지정 데몬 작업에 대한 로그 파일을 만듭니다.

```
# /usr/sbin/in.routed /var/log-file-name
```



주의 - 사용량이 많은 네트워크에서는 이 명령이 거의 연속적으로 출력을 생성할 수 있습니다.

### 예 5-11 in.routed 데몬에 대한 네트워크 로그

다음 예는 100 페이지 “IPv4 경로 지정 데몬의 작업을 기록하는 방법” 절차에서 만든 로그의 시작 부분을 보여줍니다.

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0 #1 127.0.0.1 -->127.0.0.1/32
<UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
Add interface net0 #2 10.10.48.112 -->10.10.48.0/25
<UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add 10.0.0.0 -->10.10.48.112 metric=0 net0 <NET_SYN>
Add 10.10.48.85/25 -->10.10.48.112 metric=0 net0 <IF|NOPROP>
```

## ▼ IPv6 Neighbor Discovery 데몬의 작업을 추적하는 방법

IPv6 `in.ndpd` 데몬의 오작동이 의심되는 경우 데몬의 작업을 추적하는 로그를 시작할 수 있습니다. 이 추적은 종료될 때까지 표준 출력에 표시됩니다. `in.ndpd` 데몬이 시작되면 이 추적에는 모든 패킷 전송이 포함됩니다.

- 1 `in.ndpd` 데몬의 추적을 시작합니다.
- 2 필요한 경우 `Ctrl-C`를 입력하여 추적을 종료합니다.

### 예 5-12 in.ndpd 데몬 추적

다음 출력은 `in.ndpd` 추적의 시작 부분을 보여줍니다.

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to ff02::2 (16 bytes) on net0
Nov 18 17:27:28 Source LLA: len 6 <08:00:20:b9:4c:54>
```

```

Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on net0
Nov 18 17:27:28 Max hop limit: 0
Nov 18 17:27:28 Managed address configuration: Not set
Nov 18 17:27:28 Other configuration flag: Not set
Nov 18 17:27:28 Router lifetime: 1800
Nov 18 17:27:28 Reachable timer: 0
Nov 18 17:27:28 Reachable retrans timer: 0
Nov 18 17:27:28 Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28 Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28 On link flag:Set
Nov 18 17:27:28 Auto addrconf flag:Set
Nov 18 17:27:28 Valid time: 2592000
Nov 18 17:27:28 Preferred time: 604800
Nov 18 17:27:28 Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28 On link flag:Set
Nov 18 17:27:28 Auto addrconf flag:Set
Nov 18 17:27:28 Valid time: 2592000
Nov 18 17:27:28 Preferred time: 604800

```

## tracert 명령으로 경로 지정 정보 표시

tracert 명령은 원격 시스템에 대한 IP 패킷의 경로를 추적합니다. tracert에 대한 기술적인 세부 정보는 [tracert\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

tracert 명령을 사용하면 잘못된 경로 지정 구성 및 경로 지정 경로 오류를 찾을 수 있습니다. 특정 호스트에 연결할 수 없는 경우 tracert를 사용하여 원격 호스트에 대한 패킷 경로 및 오류가 발생할 수 있는 위치를 확인할 수 있습니다.

tracert 명령은 대상 호스트에 대한 경로를 따라 전송하는 각 게이트웨이에 대한 라운드 트립 시간도 표시합니다. 이 정보는 두 노드 간의 트래픽이 느려지는 위치를 분석하는 데 유용할 수 있습니다.

### ▼ 원격 호스트에 대한 경로를 찾는 방법

- 원격 시스템에 대한 경로를 찾으려면 다음을 입력합니다.

```
% tracert destination-hostname
```

사용자 계정에서 tracert 명령을 다음 형식으로 실행할 수 있습니다.

#### 예 5-13 tracert 명령으로 원격 호스트에 대한 경로 표시

tracert 명령의 다음 출력은 로컬 호스트 nearhost에서 원격 시스템 farhost로 전송되는 패킷의 7홉 경로를 보여줍니다. 이 출력은 패킷이 각 홉을 순회하는 시간도 표시합니다.

```

istanbul% tracert farhost.faraway.com
tracert to farhost.faraway.com (172.16.64.39), 30 hops max, 40 byte packets
 1 frblbg7c-86 (172.16.86.1) 1.516 ms 1.283 ms 1.362 ms

```

```

2 bldg1a-001 (172.16.1.211) 2.277 ms 1.773 ms 2.186 ms
3 bldg4-bldg1 (172.16.4.42) 1.978 ms 1.986 ms 13.996 ms
4 bldg6-bldg4 (172.16.4.49) 2.655 ms 3.042 ms 2.344 ms
5 ferbldg11a-001 (172.16.1.236) 2.636 ms 3.432 ms 3.830 ms
6 frbldg12b-153 (172.16.153.72) 3.452 ms 3.146 ms 2.962 ms
7 sanfrancisco (172.16.64.39) 3.430 ms 3.312 ms 3.451 ms

```

## ▼ 모든 경로를 추적하는 방법

이 절차는 traceroute 명령의 -a 옵션을 사용하여 모든 경로를 추적합니다.

### ● 로컬 시스템에서 다음 명령을 입력합니다.

```
% traceroute -a host-name
```

사용자 계정에서 traceroute 명령을 다음 형식으로 실행할 수 있습니다.

### 예 5-14 듀얼 스택 호스트에 대한 모든 경로 추적

이 예는 듀얼 스택 호스트에 대해 가능한 모든 경로를 보여줍니다.

```

% traceroute -a v6host.remote.com
traceroute: Warning: Multiple interfaces found; using 2::56:a0:a8 @ eri0:2
traceroute to v6host (2001:db8:4a3b::102:a00:fe79:19b0), 30 hops max, 60 byte packets
 1 v6-rout86 (2001:db8:4a3b:56:a00:fe1f:59a1) 35.534 ms 56.998 ms *
 2 2001:db8::255:0:c0a8:717 32.659 ms 39.444 ms *
 3 farhost.faraway.COM (2001:db8:4a3b::103:a00:fe9a:ce7b) 401.518 ms 7.143 ms *
 4 distant.remote.com (2001:db8:4a3b::100:a00:fe7c:cf35) 113.034 ms 7.949 ms *
 5 v6host (2001:db8:4a3b::102:a00:fe79:19b0) 66.111 ms * 36.965 ms

traceroute to v6host.remote.com (192.168.10.75), 30 hops max, 40 byte packets
 1 v6-rout86 (172.16.86.1) 4.360 ms 3.452 ms 3.479 ms
 2 flrmpj17u.here.COM (172.16.17.131) 4.062 ms 3.848 ms 3.505 ms
 3 farhost.farway.com (10.0.0.23) 4.773 ms * 4.294 ms
 4 distant.remote.com (192.168.10.104) 5.128 ms 5.362 ms *
 5 v6host (192.168.15.85) 7.298 ms 5.444 ms *

```

## snoop 명령으로 패킷 전송 모니터링

snoop 명령을 사용하여 데이터 전송 상태를 모니터링할 수 있습니다. snoop 명령은 네트워크 패킷을 캡처한 다음 사용자가 지정한 형식으로 해당 패킷의 콘텐츠를 표시합니다. 패킷은 수신 즉시 표시하거나 파일에 저장할 수 있습니다. snoop가 중간 파일에 기록할 경우 추적 사용 조건에서 패킷 손실이 발생할 가능성이 거의 없습니다. snoop 자체는 이 파일을 해석하는 데 사용됩니다.

Promiscuous 모드에서 기본 인터페이스에 대한 패킷을 캡처하려면 사용자가 네트워크 관리 역할을 사용하거나 수퍼유저여야 합니다. 요약 양식에서 snoop는 최고 레벨 프로토콜에 해당하는 데이터만 표시합니다. 예를 들어 NFS 패킷은 NFS 정보만 표시합니다. 기본 RPC, UDP, IP 및 이더넷 프레임 정보는 표시되지 않지만, 상세 정보 표시 옵션을 선택하면 표시될 수 있습니다.

snoop 명령을 자주 그리고 일관되게 사용하면 정상적인 시스템 동작에 익숙해질 수 있습니다. 패킷 분석에 대한 지원 정보는 최근 백서 및 RFC에서 특정 영역(예: NFS 또는 NIS)의 전문가 권장 사항을 참조하십시오. snoop 및 옵션 사용에 대한 자세한 내용은 [snoop\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ 모든 인터페이스의 패킷을 확인하는 방법

- 1 시스템에 연결된 인터페이스에 대한 정보를 출력합니다.

```
# ipadm show-if
```

snoop 명령은 일반적으로 첫번째 비루프백 장치(보통 기본 네트워크 인터페이스)를 사용합니다.

- 2 Example 8-19에 표시된 것과 같이, 예 5-15를 인수 없이 입력하여 패킷 캡처를 시작합니다.
- 3 Ctrl-C를 사용하여 프로세스를 정지합니다.

### 예 5-15 snoop 명령의 출력

기본 snoop 명령은 듀얼 스택 호스트에 대해 다음과 비슷한 출력을 반환합니다.

```
% snoop
Using device /dev/net (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
0:10:7b:31:37:80
router5.local.com -> BROADCAST      TFTP Read "network-config" (octet)
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
niserve2.
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

이 출력에 캡처된 패킷은 주소 분석용 NIS 및 DNS 서버에 대한 조치를 비롯하여 원격 로그인 섹션을 보여줍니다. 로컬 라우터에서 보내는 정기 ARP 패킷 및 in.ripngd에 대한 IPv6 링크 로컬 주소 알림도 포함됩니다.

## ▼ snoop 출력을 파일로 캡처하는 방법

- 1 snoop 세션을 파일로 캡처합니다.

```
# snoop -o filename
```

예를 들면 다음과 같습니다.

```
# snoop -o /tmp/cap
Using device /dev/eri (promiscuous mode)
30 snoop: 30 packets captured
```

이 예에서는 패킷 30이 /tmp/cap 파일에 캡처되었습니다. 이 파일은 디스크 공간이 충분한 모든 디렉토리에 있을 수 있습니다. 캡처된 패킷 수는 명령줄에 표시되는데, Ctrl-C를 누르면 언제든지 중단할 수 있습니다.

snoop은 호스트 시스템에 많은 네트워크 로드를 만드는데, 이로 인해 결과가 왜곡될 수 있습니다. 실제 결과를 표시하려면 세번째 시스템에서 snoop을 실행하십시오.

## 2 snoop 출력 캡처 파일을 검사합니다.

```
# snoop -i filename
```

### 예 5-16 snoop 출력 캡처 파일의 내용

다음 출력은 snoop -i 명령의 출력과 같은 다양한 캡처를 보여줍니다.

```
# snoop -i /tmp/cap
1  0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fe8d:4375
    ICMPv6 Neighbor advertisement
...
10 0.91493 10.0.0.40 -> (broadcast) ARP C Who is 10.0.0.40, 10.0.0.40 ?
34 0.43690 nearserver.here.com -> 224.0.1.1 IP D=224.0.1.1 S=10.0.0.40 LEN=28,
    ID=47453, TO =0x0, TTL=1
35 0.00034 10.0.0.40 -> 224.0.1.1 IP D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
    TOS=0x0, TTL=47
```

## ▼ IPv4 서버와 클라이언트 간 패킷을 확인하는 방법

### 1 클라이언트 또는 서버에 연결된 허브와 떨어져 snoop 시스템을 설정합니다.

세번째 시스템(snoop 시스템)은 방해하는 모든 트래픽을 확인하므로 snoop 추적은 회선에서 실제로 발생한 사항을 반영합니다.

### 2 옵션과 함께 snoop를 입력한 다음 출력을 파일에 저장합니다.

### 3 출력 내용을 검사하고 해석합니다.

snoop 캡처 파일에 대한 자세한 내용은 RFC 1761, Snoop Version 2 Packet Capture File Format (<http://www.ietf.org/rfc/rfc1761.txt?number=1761>)을 참조하십시오.



## ▼ IPv6 네트워크 트래픽을 모니터링하는 방법

snoop 명령으로 IPv6 패킷만 표시할 수 있습니다.

- IPv6 패킷을 캡처합니다.

```
# snoop ip6
```

snoop 명령에 대한 자세한 내용은 [snoop\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### 예 5-17 IPv6 네트워크 트래픽만 표시

다음 예는 노드에서 snoop ip6 명령을 실행할 경우 표시되는 출력과 같은 일반 출력을 보여줍니다.

```
# snoop ip6
fe80::a00:20ff:febd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:febd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:febd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

## IP 계층 장치를 사용하여 패킷 모니터링

IP 계층 장치는 IP 관찰을 향상하기 위해 Oracle Solaris에서 도입되었습니다. 이 장치는 시스템의 네트워크 인터페이스와 연관된 주소를 사용하는 모든 패킷에 액세스할 수 있습니다. 이 주소에는 비루프백 인터페이스 또는 논리적 인터페이스에서 호스트된 주소 및 로컬 주소가 포함됩니다. IPv4 주소와 IPv6 주소 둘 다의 트래픽을 관찰할 수 있습니다. 따라서 시스템을 대상으로 하는 모든 트래픽을 모니터링할 수 있습니다. 트래픽은 루프백 IP 트래픽, 원격 시스템에서 보내는 패킷, 시스템에서 전송 중인 패킷 또는 전송된 모든 트래픽일 수 있습니다.

IP 계층 장치를 사용하면 전역 영역 관리자가 영역 간 트래픽과 영역 내 트래픽을 모니터링할 수 있습니다. 비전역 영역의 관리자도 해당 영역에서 전송하고 수신한 트래픽을 관찰할 수 있습니다.

IP 계층에서 트래픽을 모니터링하기 위해 새 옵션인 `-I`가 snoop 명령에 추가되었습니다. 이 옵션은 명령이 기본 링크 계층 장치 대신 새 IP 계층 장치를 사용하여 트래픽 데이터를 표시하도록 지정합니다.

---

주 - 계층 간의 차이를 이해하려면 [System Administration Guide: IP Services](#)의 “Data Encapsulation and the TCP/IP Protocol Stack”을 참조하십시오.

---

## ▼ IP 계층에서 패킷을 확인하는 방법

- 1 필요한 경우 시스템에 연결된 인터페이스에 대한 정보를 출력합니다.

```
# ipadm show-if
```

- 2 특정 인터페이스에서 IP 트래픽을 캡처합니다.

```
# snoop -I interface [-V | -v]
```

### 패킷 확인 예

모든 예는 다음과 같은 시스템 구성을 기반으로 합니다.

```
# ipadm show-addr
ADDROBJ      TYPE      STATE    ADDR
lo0/v4       static    ok       127.0.0.1/8
net0/v4       static    ok       192.68.25.5/24
lo0/?        static    ok       127.0.0.1/8
net0/?        static    ok       172.0.0.3/24
net0/?        static    ok       172.0.0.1/24
lo0/?        static    ok       127.0.0.1/8
```

sandbox 및 toybox라는 두 영역이 다음 IP 주소를 사용한다고 가정합니다.

- sandbox – 172.0.0.3
- toybox – 172.0.0.1

시스템의 서로 다른 인터페이스에서 `snoop -I` 명령을 실행할 수 있습니다. 표시되는 패킷 정보는 사용자가 전역 영역 관리자인지 아니면 비전역 영역의 관리자인지 여부에 따라 달라집니다.

예 5-18 루프백 인터페이스의 트래픽

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

상세 정보 출력을 생성하려면 `-v` 옵션을 사용하십시오.

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET: ----- IPNET Header -----
IPNET:
IPNET: Packet 1 arrived at 10:40:33.68506
IPNET: Packet size = 108 bytes
IPNET: dli_version = 1
IPNET: dli_type = 4
IPNET: dli_srczone = 0
IPNET: dli_dstzone = 0
IPNET:
```

```
IP:  ----- IP Header -----
IP:
IP:  Version = 4
IP:  Header length = 20 bytes
...
```

IP 계층에서는 패킷 관찰이 지원되므로 새 IPNET 헤더가 관찰 중인 패킷의 앞에 표시됩니다. 소스 및 대상 ID가 모두 표시됩니다. ID '0'은 트래픽이 전역 영역에서 생성됨을 나타냅니다.

예 5-19 로컬 영역에 있는 net0 장치의 패킷 흐름

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491
```

이 출력은 시스템 내의 서로 다른 영역에서 발생하는 트래픽을 보여줍니다. 로컬에서 다른 영역으로 전달되는 패킷을 비롯하여 net0 IP 주소와 연관된 모든 패킷을 확인할 수 있습니다. 상세 정보 출력을 생성하면 패킷 흐름과 관련된 영역을 확인할 수 있습니다.

```
# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
IP:  ----- IP Header -----
IP:
IP:  Version = 4
IP:  Header length = 20 bytes
IP:  Type of service = 0x00
IP:      xxx. .... = 0 (precedence)
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = not ECN capable transport
IP:      .... ...0 = no ECN congestion experienced
IP:  Total length = 40 bytes
IP:  Identification = 22629
IP:  Flags = 0x4
IP:      .1.. .... = do not fragment
IP:      ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 64 seconds/hops
IP:  Protocol = 6 (TCP)
IP:  Header checksum = 0000
IP:  Source address = 172.0.0.1, 172.0.0.1
IP:  Destination address = 172.0.0.3, 172.0.0.3
IP:  No options
```

예 5-19 로컬 영역에 있는 net0 장치의 패킷 흐름 (계속)

```

IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:      0... .... = No ECN congestion window reduced
TCP:      .0.. .... = No ECN echo
TCP:      ..0. .... = No urgent pointer
TCP:      ...1 .... = Acknowledgement
TCP:      .... 0... = No push
TCP:      .... .0.. = No reset
TCP:      .... ..0. = No Syn
TCP:      .... ...0 = No Fin
TCP:  Window = 49152
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:

```

IPNET 헤더는 패킷이 전역 영역(ID 0)에서 Sandbox(ID 1)로 제공됨을 나타냅니다.

예 5-20 영역 식별을 통한 트래픽 관찰

```

# snoop -I hme0 sandboxsnop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
#

```

영역을 식별하여 패킷을 관찰하는 기능은 영역이 여러 개 있는 시스템에 유용합니다. 현재는 영역 ID로만 영역을 식별할 수 있습니다. 영역 이름과 함께 snoop를 사용하는 것은 지원되지 않습니다.

## 기본 주소 선택 관리

Oracle Solaris에서는 한 인터페이스에서 여러 개의 IP 주소를 사용할 수 있습니다. 예를 들어 네트워크 다중 경로(IPMP)와 같은 기술이 여러 네트워크 인터페이스 카드(NIC)를 사용하여 동일한 IP 링크 계층에 연결할 수 있도록 해줍니다. 이러한 링크는 여러 개의 IP 주소를 사용할 수 있습니다. 또한 IPv6 지원 시스템의 인터페이스에는 적어도 하나의 인터페이스에 대해 링크 로컬 IPv6 주소 하나, IPv6 경로 지정 주소 하나 이상 및 IPv4 주소 하나가 포함됩니다.

시스템에서 트랜잭션이 시작되면 응용 프로그램은 getaddrinfo 소켓을 호출합니다. getaddrinfo는 대상 시스템에서 사용 중인 가능한 주소를 검색합니다. 그러면 커널에서 이 목록의 우선 순위를 정해 패킷에 사용할 최적의 대상을 찾습니다. 이 프로세스를 대상

**주소 순서 지정**이라고 합니다. 패킷에 대한 최적의 대상 주소가 제공된 경우 Oracle Solaris 커널에서 소스 주소에 적합한 형식을 선택합니다. 이 프로세스를 **주소 선택**이라고 합니다. 대상 주소 순서 지정에 대한 자세한 내용은 [getaddrinfo\(3SOCKET\)](#) 매뉴얼 페이지를 참조하십시오.

IPv4 전용 및 듀얼 스택 IPv4/IPv6 시스템 모두 기본 주소 선택을 수행해야 합니다. 대부분의 경우에는 기본 주소 선택 방식을 변경할 필요가 없습니다. 그러나 IPMP를 지원하거나 6to4 주소 형식을 선호하는 경우 주소 형식의 우선 순위를 변경해야 할 수 있습니다.

## ▼ IPv6 주소 선택 정책 테이블을 관리하는 방법

다음 절차는 주소 선택 정책 테이블을 수정하는 방법에 대해 설명합니다. IPv6 기본 주소 선택에 대한 개념 정보는 [145 페이지 “ipaddrsel 명령”](#)을 참조하십시오.



**주의** - 다음 작업에 표시된 이유가 아니면 IPv6 주소 선택 정책 테이블을 변경하지 마십시오. 정책 테이블이 잘못 구성된 경우 네트워크 문제가 발생할 수 있습니다. 다음 절차에서 수행된 것과 같이, 정책 테이블의 백업 복사본을 반드시 저장하십시오.

### 1 현재 IPv6 주소 선택 정책 테이블을 검토합니다.

```
# ipaddrsel
# Prefix                Precedence Label
::1/128                 50 Loopback
::/0                    40 Default
2002::/16               30 6to4
::/96                   20 IPv4-Compatible
::ffff:0.0.0.0/96      10 IPv4
```

### 2 기본 주소 정책 테이블의 백업 복사본을 만듭니다.

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

### 3 텍스트 편집기를 사용하여 /etc/inet/ipaddrsel.conf에 사용자 정의 내용을 추가합니다.

/etc/inet/ipaddrsel의 항목에 다음 구문을 사용합니다.

```
prefix/prefix-length precedence label [# comment]
```

다음은 정책 테이블에 대해 수행할 수 있는 몇 가지 일반적인 수정 사항입니다.

- 6to4 주소에 가장 높은 우선 순위를 제공합니다.

```
2002::/16                50 6to4
::1/128                  45 Loopback
```

이제 6to4 주소에 가장 높은 우선 순위인 50이 지정됩니다. 루프백의 경우 우선 순위가 이제 50에서 45로 변경됩니다. 기타 주소 지정 형식은 그대로 유지합니다.

- 특정 대상 주소와의 통신에 사용할 특정 소스 주소를 지정합니다.

```

::1/128                    50 Loopback
2001:1111:1111::1/128     40 ClientNet
2001:2222:2222::/48       40 ClientNet
::/0                       40 Default

```

이 특정 항목은 물리적 인터페이스가 한 개뿐인 호스트에 유용합니다.

2001:1111:1111::1/128은 2001:2222:2222::/48 네트워크 내에서 대상에 대해 바운드되는 모든 패킷에 대한 소스 주소로 선호됩니다. 우선 순위 40은 소스 주소 2001:1111:1111::1/128에 대한 우선 순위로, 해당 인터페이스에 대해 구성된 다른 주소 형식보다 높습니다.

- IPv6 주소보다 IPv4 주소를 선호합니다.

```

::ffff:0.0.0.0/96         60 IPv4
::1/128                   50 Loopback
.
.

```

IPv4 형식 ::ffff:0.0.0.0/96의 우선 순위가 10(기본값)에서 60(테이블의 가장 높은 우선 순위)으로 변경되었습니다.

- 4 수정된 정책 테이블을 커널로 로드합니다.

```
ipaddrsel -f /etc/inet/ipaddrsel.conf
```

- 5 수정된 정책 테이블에 문제가 있는 경우 기본 IPv6 주소 선택 정책 테이블을 복원합니다.

```
# ipaddrsel -d
```

## ▼ 현재 세션에 대해서만 IPv6 주소 선택 정책 테이블을 수정하는 방법

/etc/inet/ipaddrsel.conf, 파일을 편집하면 수정 사항이 재부트 후에도 지속됩니다. 수정된 정책 테이블이 현재 세션에서만 사용되도록 하려면 다음 절차를 수행하십시오.

- 1 /etc/inet/ipaddrsel의 내용을 *filename*으로 복사합니다. 여기서 *filename*은 사용자가 선택한 파일의 이름을 나타냅니다.

```
# cp /etc/inet/ipaddrsel filename
```

- 2 *filename*의 정책 테이블을 원하는 지정 사항으로 편집합니다.

- 3 수정된 정책 테이블을 커널로 로드합니다.

```
# ipaddrsel -f filename
```

시스템을 재부트할 때까지 커널에서 새 정책 테이블을 사용합니다.

## IP 터널 구성

---

이 장에서는 IP 터널에 대한 설명 및 Oracle Solaris에서 터널을 구성 및 유지 관리하는 절차에 대해 다룹니다.

### IP 터널 개요

IP 터널은 중간 네트워크에서 도메인의 프로토콜을 지원하지 않을 경우 도메인 간에 데이터 패킷을 전송할 수 있도록 해줍니다. 예를 들면 IPv6 프로토콜이 도입됨으로써 IPv6 네트워크에는 대부분의 네트워크가 IPv4 프로토콜을 사용하는 환경의 경계를 벗어나 통신할 수 있는 방법이 필요합니다. 통신은 터널을 통해 가능해집니다. IP 터널은 IP를 사용하여 연결 가능한 두 노드 간에 가상 링크를 제공합니다. 따라서 이 링크를 사용하면 IPv4 네트워크를 통해 IPv6 패킷을 전송할 수 있으므로 두 IPv6 사이트 간 IPv6 통신이 가능해집니다.

### 이 Oracle Solaris 릴리스에서 IP 터널 관리

이 Oracle Solaris 릴리스에서는 네트워크 데이터 링크 관리를 위한 새 모델과 일치하도록 터널 관리가 수정되었습니다. 이제 터널은 새 `dladm` 하위 명령을 사용하여 생성되고 구성됩니다. 또한 터널은 새 관리 모델의 다른 데이터 링크 기능을 사용할 수도 있습니다. 예를 들어 관리상 선택한 이름이 지원되므로 터널에 의미 있는 이름을 지정할 수 있습니다. `dladm` 하위 명령에 대한 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### 터널의 유형

터널링은 다른 패킷 내에서 IP 패킷을 캡슐화하는 것입니다. 캡슐화는 패킷의 프로토콜을 지원하지 않는 중간 네트워크를 통해 패킷이 대상에 도달할 수 있도록 해줍니다.

터널은 패킷 캡슐화의 유형에 따라 달라집니다. Oracle Solaris에서 지원되는 터널의 유형은 다음과 같습니다.

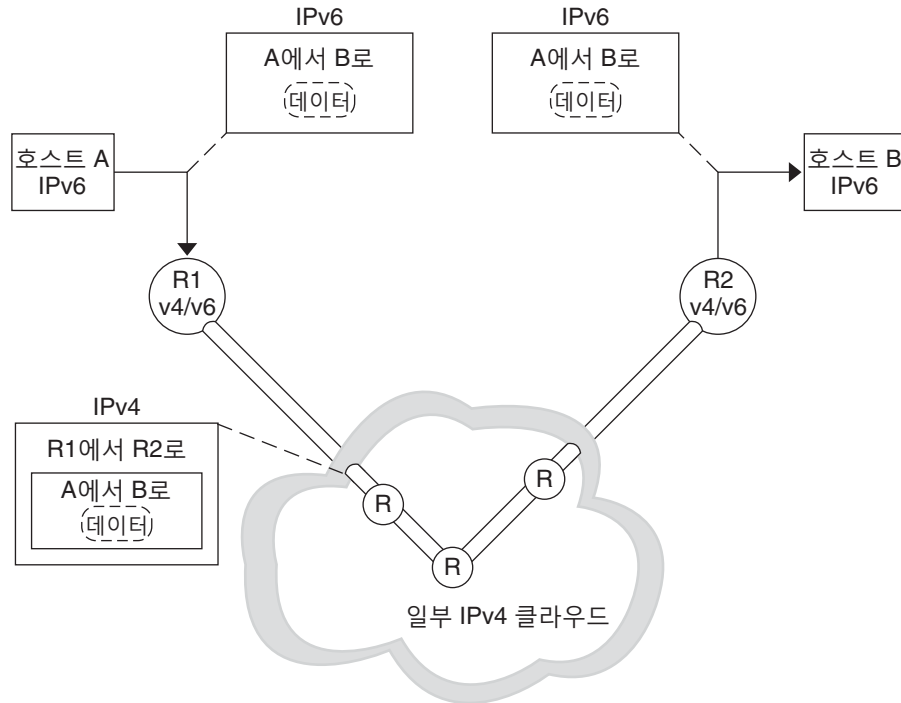
- **IPv4 터널** – IPv4 또는 IPv6 패킷은 IPv4 헤더에서 캡슐화되어 미리 구성된 유니캐스트 IPv4 대상으로 전송됩니다. 터널을 경유하는 패킷을 보다 명확하게 하기 위해 IPv4 터널을 *IPv4 over IPv4 터널* 또는 *IPv6 over IPv4 터널*이라고도 합니다.
- **IPv6 터널** – IPv4 또는 IPv6 패킷은 IPv6 헤더에서 캡슐화되어 미리 구성된 유니캐스트 IPv6 대상으로 전송됩니다. 터널을 경유하는 패킷을 보다 명확하게 하기 위해 IPv6 터널을 *IPv4 over IPv6 터널* 또는 *IPv6 over IPv6 터널*이라고도 합니다.
- **6to4 터널** – IPv6 패킷은 IPv4 헤더에서 캡슐화되어 패킷별로 자동 결정되는 IPv4 대상으로 전송됩니다. 이때 6to4 프로토콜에 정의된 알고리즘을 기준으로 결정됩니다.

## 결합된 IPv6 및 IPv4 네트워크 환경에서의 터널

IPv6 도메인이 있는 대부분의 사이트에서는 IPv4 네트워크를 순회하여 다른 IPv6 도메인과 통신하는데, 이는 IPv6 전용 네트워크보다 IPv4 네트워크에서 더 일반적입니다. 다음 그림은 IPv4 라우터를 경유하는 두 IPv6 호스트 간의 터널링 방식을 보여줍니다. IPv4 라우터는 그림에서 “R”로 표시되어 있습니다.



그림 6-1 IPv6 터널링 방식



이 그림에서 터널은 두 개의 라우터로 구성되는데, 이 라우터는 IPv4 네트워크를 경유하여 두 라우터 간에 가상 지점 간 링크를 갖도록 구성되어 있습니다.

IPv6 패킷은 IPv4 패킷 내에서 캡슐화됩니다. IPv6 네트워크의 경계 라우터는 다양한 IPv4 네트워크를 경유하여 대상 IPv6 네트워크의 경계 라우터에 도달하는 지점 간 터널을 설정합니다. 패킷은 터널을 경유하여 대상 경계 라우터로 전송되며, 여기서 패킷이 캡슐화 해제됩니다. 그러면 라우터가 개별 IPv6 패킷을 대상 노드로 전달합니다.

## 6to4 터널

Oracle Solaris는 주소 지정을 IPv4에서 IPv6으로 전환하는 데 선호하는 중간 방식으로 6to4 터널을 제공합니다. 6to4 터널은 분리된 IPv6 사이트가 IPv6을 지원하지 않는 IPv4 네트워크를 경유하여 자동 터널을 넘어 통신할 수 있도록 해줍니다. 6to4 터널을 사용하려면 IPv6 네트워크의 경계 라우터를 6to4 자동 터널의 한 끝점으로 구성해야 합니다. 그러면 6to4 라우터가 다른 6to4 사이트 또는 필요한 경우 원시 IPv6, 비6to4 사이트에 대한 터널에 참여할 수 있습니다.

이 절에서는 다음 6to4 항목에 대한 참조 자료를 제공합니다.

- 6to4 터널 토폴로지
- 6to4 터널을 경유하는 패킷에 대한 설명
- 6to4 라우터와 6to4 릴레이 라우터 간 터널의 토폴로지
- 6to4 릴레이 라우터 지원을 구성하기 전에 고려할 사항

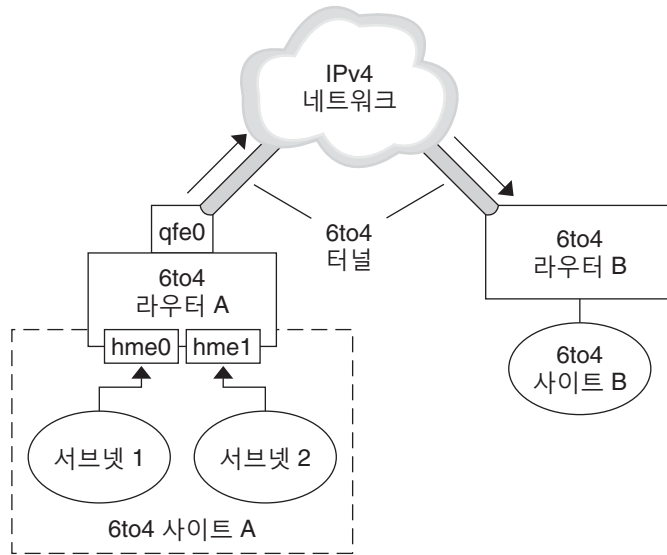
다음 표는 유용한 정보를 추가로 얻기 위해 6to4 터널 및 리소스를 구성하는 추가 작업에 대해 설명합니다.

작업 또는 세부 정보	정보
6to4 터널 구성 작업	124 페이지 “6to4 터널을 구성하는 방법”
6to4 관련 RFC	RFC 3056, "Connection of IPv6 Domains via IPv4 Clouds" ( <a href="http://www.ietf.org/rfc/rfc3056.txt">http://www.ietf.org/rfc/rfc3056.txt</a> )
6to4 릴레이 라우터에 대한 터널을 지원하는 6to4relay 명령에 대한 세부 정보	6to4relay(1M)
6to4 보안 문제	Security Considerations for 6to4 ( <a href="http://www.ietf.org/rfc/rfc3964.txt">http://www.ietf.org/rfc/rfc3964.txt</a> )

## 6to4 터널 토폴로지

6to4 터널은 모든 위치에서 모든 6to4 사이트에 대한 IPv6 연결을 제공합니다. 마찬가지로, 터널이 릴레이 라우터로 전달되도록 구성된 경우 터널은 원시 IPv6 인터넷을 비롯한 모든 IPv6 사이트에 대한 링크 역할도 합니다. 다음 그림은 6to4 터널이 6to4 사이트 간에 이러한 연결을 제공하는 방식을 보여줍니다.

그림 6-2 6to4 사이트 간 터널



이 그림은 분리된 두 6to4 네트워크인 사이트 A 및 사이트 B를 보여줍니다. 각 사이트는 IPv4 네트워크에 대한 외부 연결을 포함하는 라우터를 구성했습니다. IPv4 네트워크를 경유하는 6to4 터널은 6to4 사이트를 연결합니다.

IPv6 사이트가 6to4 사이트가 되려면 먼저 6to4 지원을 위해 적어도 하나의 라우터 인터페이스를 구성해야 합니다. 이 인터페이스는 IPv4 네트워크에 대한 외부 연결을 제공해야 합니다. qfe0에 구성된 주소는 전역적으로 고유해야 합니다. 이 그림에서 라우터 A의 인터페이스인 qfe0은 사이트 A를 IPv4 네트워크에 연결해 줍니다. qfe0을 6to4의 사 인터페이스로 구성하기 전에 이미 qfe0 인터페이스가 IPv4 주소를 사용하도록 구성되어 있어야 합니다.

그림에서 6to4 사이트 A는 두 개의 서브넷으로 구성되며, 두 서브넷은 라우터 A의 hme0 및 hme1 인터페이스에 연결됩니다. 사이트 A의 서브넷에 있는 모든 IPv6 호스트는 라우터 A로부터 알림을 수신하면 6to4 파생 주소를 사용하도록 재구성됩니다.

사이트 B는 또 다른 분리된 6to4 사이트입니다. 사이트 A에서 보내는 트래픽을 올바르게 수신하려면 사이트 B의 경계 라우터가 6to4를 지원하도록 구성되어야 합니다. 그렇지 않으면 라우터가 사이트 A로부터 수신하는 패킷이 인식되지 않고 삭제됩니다.

## 6to4 터널을 경유하는 패킷 흐름

이 절에서는 6to4 사이트의 호스트에서 원격 6to4 사이트의 호스트로의 패킷 흐름에 대해 설명합니다. 이 시나리오는 [그림 6-2](#)에 표시된 토폴로지를 사용합니다. 또한 이 시나리오는 6to4 라우터와 6to4 호스트가 이미 구성되어 있다고 가정합니다.

1. 6to4 사이트 A의 서브넷 1에 있는 호스트가 6to4 사이트 B에 있는 호스트를 대상으로 지정하는 전송을 보냅니다. 각 패킷 헤더에는 6to4 파생 소스 주소와 6to4 파생 대상 주소가 있습니다.
2. 사이트 A의 라우터가 IPv4 헤더 내에서 각 6to4 패킷을 캡슐화합니다. 이 프로세스에서 라우터는 캡슐화 헤더의 IPv4 대상 주소를 사이트 B의 라우터 주소로 설정합니다. 터널 인터페이스를 경유하는 각 IPv6 패킷의 IPv6 대상 주소에는 IPv4 대상 주소도 포함되어 있습니다. 따라서 라우터가 캡슐화 헤더에 설정된 IPv4 대상 주소를 확인할 수 있습니다. 그런 다음 라우터는 표준 IPv4 경로 지정 프로시저를 사용하여 IPv4 네트워크를 통해 패킷을 전달합니다.
3. 패킷이 거쳐 가는 IPv4 라우터는 전달 시 패킷의 IPv4 대상 주소를 사용합니다. 이 주소는 라우터 B에 있는 인터페이스의 전역적으로 고유한 IPv4 주소이며, 6to4 의사 인터페이스로도 사용됩니다.
4. 사이트 A의 패킷이 라우터 B에 도달하여 IPv4 헤더에서 IPv6 패킷이 캡슐화 해제됩니다.
5. 그런 다음 라우터 B가 IPv6 패킷의 대상 주소를 사용하여 패킷을 사이트 B의 수신자 호스트로 전달합니다.

## 6to4 릴레이 라우터에 대한 터널 고려 사항

6to4 릴레이 라우터는 원시 IPv6, 비6to4 네트워크와 통신해야 하는 6to4 라우터에서 터널 끝점으로 사용됩니다. 릴레이 라우터는 기본적으로 6to4 사이트와 원시 IPv6 사이트를 연결해 줍니다. 이 솔루션은 안전하지 않으므로 기본적으로 Oracle Solaris에서는 6to4 릴레이 라우터 지원이 사용으로 설정되어 있지 않습니다. 그러나 사이트에 이러한 터널이 필요할 경우 6to4relay 명령을 사용하여 다음과 같은 터널링 시나리오를 사용으로 설정할 수 있습니다.

그림 6-3 6to4 사이트와 6to4 릴레이 라우터 간 터널

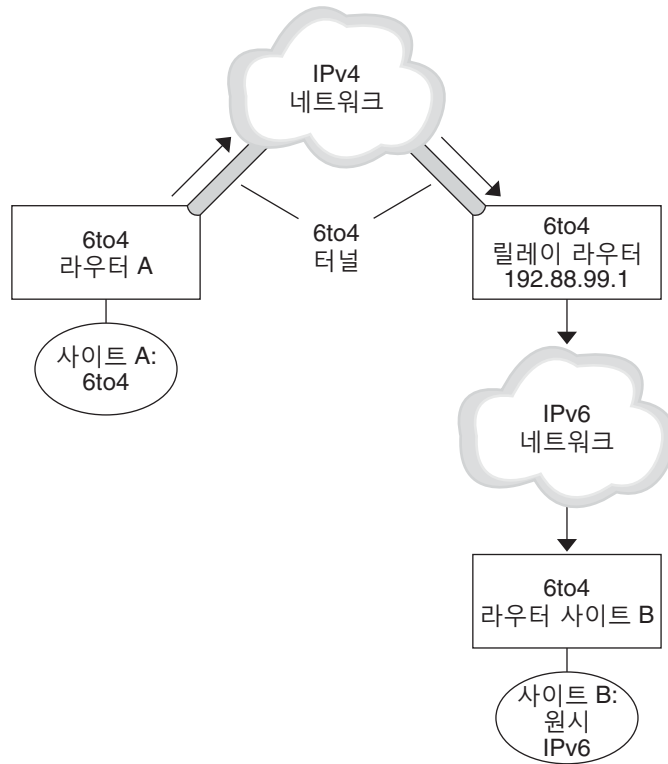


그림 6-3에서 6to4 사이트 A는 원시 IPv6 사이트 B에 있는 노드와 통신해야 합니다. 이 그림은 사이트 A에서 IPv4 네트워크를 경유하여 6to4 터널에 도달하는 트래픽 경로를 보여줍니다. 터널의 끝점은 6to4 라우터 A와 6to4 릴레이 라우터입니다. 6to4 릴레이 라우터를 넘어가면 IPv6 사이트 B가 연결되어 있는 IPv6 네트워크입니다.

### 6to4 사이트와 원시 IPv6 사이트 간 패킷 흐름

이 절에서는 6to4 사이트에서 원시 IPv6 사이트로의 패킷 흐름에 대해 설명합니다. 이 시나리오는 그림 6-3에 표시된 토폴로지를 사용합니다.

1. 6to4 사이트 A에 있는 호스트가 원시 IPv6 사이트 B에 있는 호스트를 대상으로 지정하는 전송을 보냅니다. 각 패킷 헤더에는 6to4 파생 주소가 소스 주소로 포함되어 있습니다. 대상 주소는 표준 IPv6 주소입니다.
2. 사이트 A의 6to4 라우터가 IPv4 헤더 내에서 각 패킷을 캡슐화합니다. 이 헤더에는 6to4 릴레이 라우터의 IPv4 주소가 대상으로 포함되어 있습니다. 6to4 라우터는 표준 IPv4 경로 지정 프로시저를 사용하여 IPv4 네트워크를 통해 패킷을 전달합니다. 패킷이 거쳐 가는 IPv4 라우터는 패킷을 6to4 릴레이 라우터로 전달합니다.

3. 사이트 A와 물리적으로 가장 가까운 애니캐스트 6to4 릴레이 라우터가 192.88.99.1 애니캐스트 그룹에 전송되는 패킷을 검색합니다.

---

주 - 6to4 릴레이 라우터 애니캐스트 그룹의 일부인 6to4 릴레이 라우터의 IP 주소는 192.88.99.1입니다. 이 애니캐스트 주소는 6to4 릴레이 라우터의 기본 주소입니다. 특정 6to4 릴레이 라우터를 사용해야 하는 경우 기본 주소를 대체하고 해당 라우터의 IPv4 주소를 지정할 수 있습니다.

---

4. 릴레이 라우터가 6to4 패킷에서 IPv4 헤더를 캡슐화 해제하여 원시 IPv6 대상 주소를 표시합니다.
5. 이제 패킷 라우터가 IPv6 전용 패킷을 IPv6 네트워크로 전송합니다. 이 네트워크에서 패킷이 사이트 B의 라우터에 의해 검색됩니다. 라우터가 패킷을 대상 IPv6 노드로 전달합니다.

## 터널 배치

IP 터널을 제대로 배치하려면 두 가지 기본 작업을 수행해야 합니다. 먼저 터널 링크를 만드십시오. 그런 다음 터널을 경유하는 IP 인터페이스를 구성하십시오. 이 절에서는 터널 및 해당 IP 인터페이스를 만들기 위한 요구 사항에 대해 설명합니다.

### 터널 만들기 요구 사항

터널을 성공적으로 만들려면 다음 요구 사항을 알고 있어야 합니다.

- 리터럴 IP 주소 대신 호스트 이름을 사용하는 경우 이 이름은 터널 유형과 호환되는 유효한 IP 주소로 분석되어야 합니다.
- 만드는 IPv4 또는 IPv6 터널이 구성된 다른 터널과 동일한 터널 소스 주소 및 터널 대상 주소를 공유해서는 안 됩니다.
- 만드는 IPv4 또는 IPv6 터널이 기존 6to4 터널과 동일한 터널 소스 주소를 공유해서는 안 됩니다.
- 6to4 터널을 만드는 경우, 터널이 구성된 다른 터널과 동일한 터널 소스 주소를 공유해서는 안 됩니다.

네트워크에서 터널 설정에 대한 자세한 내용은 [40 페이지 “네트워크에서 터널 사용 계획”](#)을 참조하십시오.

### 터널 및 IP 인터페이스 요구 사항

각 터널 유형에는 터널을 경유하도록 구성한 IP 인터페이스에 대한 특정 IP 주소 요구 사항이 있습니다. 요구 사항은 다음 표에 요약되어 있습니다.

표 6-1 터널 및 IP 인터페이스 요구 사항

터널 유형	터널을 통해 허용되는 IP 인터페이스	IP 인터페이스 요구 사항
IPv4 터널	IPv4 인터페이스	로컬 및 원격 주소를 수동으로 지정해야 합니다.
	IPv6 인터페이스	<code>ipadm create-addr -T addrconf</code> 명령을 실행하면 로컬 및 원격 링크 로컬 주소가 자동으로 설정됩니다. 자세한 내용은 <a href="#">ipadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
IPv6 터널	IPv4 인터페이스	로컬 및 원격 주소를 수동으로 지정해야 합니다.
	IPv6 인터페이스	<code>ipadm create-addr -T addrconf</code> 명령을 실행하면 로컬 및 원격 링크 로컬 주소가 자동으로 설정됩니다. 자세한 내용은 <a href="#">ipadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
6to4 터널	IPv6 인터페이스만	<code>ipadm create-if</code> 명령을 실행하면 기본 IPv6 주소가 자동으로 설정됩니다. 자세한 내용은 <a href="#">ipadm(1M)</a> 매뉴얼 페이지를 참조하십시오.

6to4 터널의 기본 IPv6 인터페이스 주소는 `ipadm` 명령으로 다른 IPv6 주소를 지정하여 대체할 수 있습니다.

마찬가지로, IPv4 또는 IPv6 터널을 경유하는 IPv6 인터페이스에 대해 자동으로 설정되는 링크 로컬 주소를 대체하려면 터널의 호스트 파일에 다른 소스 및 대상 주소를 지정하면 됩니다.

## dladm 명령을 통한 터널 구성 및 관리

이 절에서는 `dladm` 명령을 사용하여 터널을 구성하는 절차에 대해 설명합니다.

### dladm 하위 명령

이 Oracle Solaris 릴리스부터 터널 관리가 IP 인터페이스 구성과 분리되었습니다. IP 터널의 데이터 링크 측면은 `dladm` 명령으로 관리됩니다. 또한 IP 터널 인터페이스를 비롯한 IP 인터페이스 구성은 `ipadm` 명령으로 수행됩니다.

IP 터널을 구성하는 데 사용되는 `dladm`의 하위 명령은 다음과 같습니다.

- create-iptun
- modify-iptun
- show-iptun
- delete-iptun
- set-linkprop

dladm 명령에 대한 자세한 내용은 **dladm(1M)** 매뉴얼 페이지를 참조하십시오.

주 - IP 터널 관리는 IPsec 구성과 밀접한 관련이 있습니다. 예를 들어 IPsec VPN(Virtual Private Network)은 IP 터널링의 주요 용도 중 하나입니다. Oracle Solaris의 보안에 대한 자세한 내용은 제3부를 참조하십시오. IPsec를 구성하려면 15 장, “IPsec 구성(작업)”을 참조하십시오.

## 터널 구성(작업 맵)

작업	설명	수행 방법
IP 터널 만들기	네트워크를 통한 통신에 사용할 터널을 구성합니다.	120 페이지 “IP 터널을 만들고 구성하는 방법”
터널 구성 수정	터널의 소스 또는 대상 주소 등 터널의 원래 매개변수를 변경합니다.	128 페이지 “IP 터널 구성을 수정하는 방법”
터널 구성 표시	특정 터널 또는 시스템의 모든 IP 터널에 대한 구성 정보를 표시합니다.	129 페이지 “IP 터널 구성을 표시하는 방법”
터널 삭제	터널 구성을 삭제합니다.	130 페이지 “IP 터널을 삭제하는 방법”

## ▼ IP 터널을 만들고 구성하는 방법

## 1 터널을 만듭니다.

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

이 명령에 사용할 수 있는 옵션 및 인수는 다음과 같습니다.

-t 임시 터널을 만듭니다. 기본적으로 이 명령은 영구 터널을 만듭니다.



주 - 터널을 경유하는 영구 IP 인터페이스를 구성하려는 경우 `-t` 옵션을 사용하지 말고 영구 터널을 만들어야 합니다.

`-T type`

만들려는 터널의 유형을 지정합니다. 이 인수는 모든 터널 유형을 만드는 데 필수입니다.

`-a [local|remote]=address,...`

로컬 주소 및 원격 터널 주소에 해당하는 리터럴 IP 주소 또는 호스트 이름을 지정합니다. 주소는 유효해야 하며 이미 시스템에 생성되어 있어야 합니다. 터널의 유형에 따라 주소를 한 개만 지정하거나, 로컬 및 원격 주소를 모두 지정합니다. 로컬 및 원격 주소를 모두 지정하는 경우 주소를 쉼표로 구분해야 합니다.

- IPv4 터널이 작동하려면 로컬 및 원격 IPv4 주소가 필요합니다.
- IPv6 터널이 작동하려면 로컬 및 원격 IPv6 주소가 필요합니다.
- 6to4 터널이 작동하려면 로컬 IPv4 주소가 필요합니다.

주 - 영구 IP 터널 데이터 링크 구성에 호스트 이름을 주소로 사용하는 경우 호스트 이름은 구성 저장소에 저장됩니다. 이후에 시스템을 부트할 때 이름이 터널을 만든 당시에 사용했던 IP 주소와 다른 IP 주소로 분석되는 경우 터널이 새 구성을 사용하게 됩니다.

`tunnel-link`

IP 터널 링크를 지정합니다. 네트워크 링크 관리에서 의미 있는 이름이 지원되는 경우, 터널 이름이 더 이상 만들려는 터널의 유형으로 제한되지 않습니다. 대신 관리상 선택한 이름을 터널에 지정할 수 있습니다. 터널 이름은 문자열과 PPA(Physical Point of Attachment) 번호로 구성됩니다(예: `mytunnel0`). 의미 있는 이름 지정을 제어하는 규칙은 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “유효한 링크 이름 규칙”**을 참조하십시오.

터널 링크를 지정하지 않으면 다음과 같은 이름 지정 규칙에 따라 자동으로 이름이 제공됩니다.

- IPv4 터널: `ip.tun#`
- IPv6 터널: `ip6.tun#`

■ 6to4 터널: ip.6to4tun#

#은 만드는 터널 유형에 사용 가능한 가장 낮은 PPA 번호입니다.

## 2 (옵션) 홉 한계 또는 캡슐화 한계에 대한 값을 설정합니다.

# **dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link**

**hoplimit** IPv6 경유 터널링에 대한 터널 인터페이스의 홉 한계를 지정합니다.  
**hoplimit**는 IPv4 경유 터널의 IPv4 TTL(time to live) 필드에 해당합니다.

**encaplimit** 패킷에 허용되는 중첩 터널링의 레벨 수를 지정합니다. 이 옵션은 IPv6 터널에만 적용됩니다.

패킷에 허용되는 중첩 터널링의 레벨 수를 지정합니다. 이 옵션은 IPv6 터널에만 적용됩니다.

---

주 - **hoplimit** 및 **encaplimit**에 대해 설정한 값은 허용되는 범위 내에 있어야 합니다. **hoplimit** 및 **encaplimit**는 터널 링크 등록 정보입니다. 따라서 이러한 등록 정보는 다른 링크 등록 정보의 경우와 동일한 **dladm** 하위 명령으로 관리됩니다. 해당 하위 명령은 **dladm set-linkprop**, **dladm reset-linkprop** 및 **dladm show-linkprop**입니다. 링크 관리를 위해 **dladm** 명령과 함께 사용되는 여러 하위 명령은 **dladm(1M)** 매뉴얼 페이지를 참조하십시오.

---

## 3 터널을 경유하는 IP 인터페이스를 만듭니다.

# **ipadm create-ip tunnel-interface**

여기서 **tunnel-interface**는 터널 링크와 동일한 이름을 사용합니다.

## 4 터널 인터페이스에 로컬 및 원격 IP 주소를 지정합니다.

# **ipadm create-addr [-t] -T static -a local=address,remote=address addrobj**

**-t** 터널을 경유하는 영구 IP 구성이 아닌 임시 IP 구성을 나타냅니다. 이 옵션을 사용하지 않으면 IP 인터페이스 구성은 영구 구성이 됩니다.

**-T static** 동적 IP 프로시저 대신 정적 IP 주소가 사용됨을 나타냅니다.

**-a local=address,remote=address** 터널 인터페이스의 IP 주소를 지정합니다. 소스 및 대상 IP 주소가 모두 필수이며, **local** 및 **remote**로 표시됩니다. 로컬 및 원격 주소는 IPv4 또는 IPv6 주소일 수 있습니다.

**addrobj** 로컬 및 원격 주소를 소유하는 주소 객체를 지정합니다. **addrobj**는 **interface/user-specified-string** 형식을 사용해야 합니다. **user-specified-string**은

영문자로 시작하며 최대 32자인 영숫자 문자열을 나타냅니다.

ipadm 명령 및 터널 인터페이스를 비롯한 IP 주소를 구성하는 여러 옵션에 대한 자세한 내용은 [ipadm\(1M\) 매뉴얼 페이지](#) 및 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 제II부](#), “데이터 링크 및 인터페이스 구성”을 참조하십시오.

- 5 터널 구성 정보를 `/etc/hosts` 파일에 추가합니다.
- 6 (옵션) 터널 IP 인터페이스 구성의 상태를 확인합니다.

```
# ipadm show-addr interface
```

### 예 6-1 IPv4 터널을 경유하는 IPv6 인터페이스 만들기

이 예는 영구 IIPv6 over IPv4 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
# ipadm create-addr -T addrconf private0/v6
# ipadm show-addr private0/
ADDROBJ      TYPE      STATE      ADDR
private0/v6  static    ok         fe80::a08:392e/10 --> fe80::8191:9a56
```

대체 주소를 추가하려면 동일한 구문을 사용하되, `addrobj`에 다른 *user-specified-string*을 사용합니다. 예를 들어 다음과 같이 전역 주소를 추가할 수 있습니다.

```
# ipadm create-addr -T static -a local=2001:db8:4728::1, \
remote=2001:db8:4728::2 private0/global
# ipadm show-addr private0/
ADDROBJ      TYPE      STATE      ADDR
private0/v6  addrconf  ok         fe80::a08:392e/10 --> fe80::8191:9a56
private0/global static    ok         2001:db8:4728::1 --> 2001:db8:4728::2
```

IPv6 주소의 `2001:db8` 접두어는 설명서 예제에 특별히 사용되는 특수 IPv6 접두어입니다. IPv6 주소 및 형식에 대한 설명은 [System Administration Guide: IP Services](#)의 “IPv6 Addressing Overview”를 참조하십시오.

### 예 6-2 IPv4 터널을 경유하는 IPv4 인터페이스 만들기

이 예는 영구 IPv4 over IPv4 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -T static -a local=10.0.0.1,remote=10.0.0.2 vpn0/v4
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1
vpn0/v4       static    ok         10.0.0.1-->10.0.0.2
```

이 터널을 경유하는 패킷에 보안 연결을 제공하도록 IPsec 정책을 추가로 구성할 수 있습니다. IPsec 구성에 대한 자세한 내용은 15 장, “IPsec 구성(작업)”을 참조하십시오.

### 예 6-3 IPv6 터널을 경유하는 IPv6 인터페이스 만들기

이 예는 영구 IIPv6 over IPv6 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 \
tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0/v6
# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v6	static	ok	::1/128
tun0/v6	addrconf	ok	2001:db8:feed::1234 --> 2001:db8:beef::4321

전역 주소 또는 대체 로컬 및 원격 주소 등의 주소를 추가하려면 다음과 같이 ipadm 명령을 사용하십시오.

```
# ipadm create-addr -T static \
-a local=2001:db8::4728:56bc,remote=2001:db8::1428:57ab tun0/alt
# ipadm show-addr tun0/
```

ADDROBJ	TYPE	STATE	ADDR
tun0/v6	addrconf	ok	2001:db8:feed::1234 --> 2001:db8:beef::4321
tun0/alt	static	ok	2001:db8::4728:56bc --> 2001:db8::1428:57ab

## ▼ 6to4 터널을 구성하는 방법

6to4 터널에서 6to4 라우터는 네트워크의 6to4 사이트에 있는 노드에 대한 IPv6 라우터로 사용되어야 합니다. 따라서 6to4 라우터를 구성할 때 물리적 인터페이스에서 해당 라우터가 IPv6 라우터로도 구성되어야 합니다. IPv6 경로 지정에 대한 자세한 내용은 158 페이지 “IPv6 경로 지정”을 참조하십시오.

### 1 6to4 터널을 만듭니다.

```
# dladm create-iptun -T 6to4 -a local=address tunnel-link
```

이 명령에 사용할 수 있는 옵션 및 인수는 다음과 같습니다.

**-a local=address** 터널 로컬 주소를 지정합니다. 이 주소가 시스템에 이미 존재해야 유효한 주소입니다.

**tunnel-link** IP 터널 링크를 지정합니다. 네트워크 링크 관리에서 의미 있는 이름이 지원되는 경우, 터널 이름이 더 이상 만들려는 터널의 유형으로 제한되지 않습니다. 대신 관리상 선택한 이름을 터널에 지정할 수 있습니다. 터널 이름은 문자열과 PPA 번호로 구성됩니다(예: mytunnel0). 의미 있는 이름 지정을 제어하는 규칙은 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의**

“유효한 링크 이름 규칙”을 참조하십시오.

## 2 터널 IP 인터페이스를 만듭니다.

```
# ipadm create-ip tunnel-interface
```

여기서 *tunnel-interface*는 터널 링크와 동일한 이름을 사용합니다.

## 3 (옵션) 터널용 대체 IPv6 주소를 추가합니다.

## 4 다음 두 행을 추가하여 6to4 경로 지정을 알리도록 */etc/inet/ndpd.conf* 파일을 편집합니다.

```
if subnet-interface AdvSendAdvertisements 1
IPv6-address subnet-interface
```

첫번째 행은 알림을 수신하는 서브넷을 지정합니다. *subnet-interface*는 서브넷이 연결되어 있는 링크를 나타냅니다. 두번째 행의 IPv6 주소에는 6to4 터널의 IPv6 주소에 사용되는 6to4 접두어 2000이 지정됩니다.

ndpd.conf 파일에 대한 자세한 내용은 [ndpd.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## 5 IPv6 전달을 사용으로 설정합니다.

```
# ipadm set-prop -p forwarding=on ipv6
```

## 6 라우터를 재부트합니다.

또는 */etc/inet/in.ndpd* 때문에 대해 sighup을 실행하여 라우터 알림 전송을 시작할 수 있습니다. 6to4 접두어를 수신하기 위해 각 서브넷의 IPv6 노드가 이제 새 6to4 파생 주소로 자동 구성됩니다.

## 7 6to4 사이트에서 사용되는 이름 서비스에 노드의 새 6to4 파생 주소를 추가합니다.

지침은 [86 페이지 “IPv6용 이름 서비스 지원 구성”](#)을 참조하십시오.

### 예 6-4 6to4 터널 만들기

이 예에서 서브넷 인터페이스는 *bge0*이며, */etc/inet/ndpd.conf*가 적합한 단계에서 이 인터페이스를 참조하게 됩니다.

이 예는 6to4 터널을 만드는 방법을 보여줍니다. IPv6 인터페이스만 6to4 터널을 경유하도록 구성할 수 있습니다.

```
# dladm create-iptun -T 6to4 -a local=192.168.35.10 tun0
# ipadm create-ip tun0
# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
bge0/static	static	ok	192.168.35.10/24
lo0/v6	static	ok	::1/128
tun0/_a	static	ok	2002:c0a8:57bc::1/64

```
# ipadm create-addr -T static -a 2002:c0a8:230a::2/16 tun0/a2
# ipadm create-addr -T static -a 2002:c0a8:230a::3/16 tun0/a3
# ipadm show-addr tun0/
ADDROBJ      TYPE      STATE     ADDR
lo0/v4       static    ok        127.0.0.1/8
bge0/static   static    ok        192.168.35.10/24
lo0/v6       static    ok        ::1/128
tun0/_a      static    ok        2002:c0a8:57bc::1/64
tun0/a2      static    ok        2002:c0a8:230a::2/16
tun0/a3      static    ok        2002:c0a8:230a::3/16

# vi /etc/inet/ndpd.conf
if bge0 AdvSendAdvertisements 1
2002:c0a8:57bc::1/64 bge0

# ipadm set-prop -p forwarding=on ipv6
```

6to4 터널에 대한 IPv6 주소 접두어는 2002입니다. 추가 설명은 [System Administration Guide: IP Services](#)의 “Prefixes in IPv6”를 참조하십시오.

## ▼ 6to4 릴레이 라우터에 대한 6to4 터널을 구성하는 방법



주의 - 주요 보안 문제로 인해 6to4 릴레이 라우터 지원은 기본적으로 Oracle Solaris에서 사용 안함으로 설정되어 있습니다. [133 페이지 “6to4 릴레이 라우터로 터널링 시 발생하는 보안 문제”](#)를 참조하십시오.

시작하기 전에 6to4 릴레이 라우터에 대한 터널을 사용으로 설정하기 전에 다음 작업을 수행해야 합니다.

- 사이트에서 6to4 라우터 구성( [120 페이지 “IP 터널을 만들고 구성하는 방법”](#)에 설명됨)
- 6to4 릴레이 라우터에 대한 터널링과 관련된 보안 문제 검토

### 1 다음 형식 중 하나를 사용하여 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

- 애니캐스트 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

```
# /usr/sbin/6to4relay -e
```

-e 옵션은 6to4 라우터와 애니캐스트 6to4 릴레이 라우터 간에 터널을 설정합니다. 애니캐스트 6to4 릴레이 라우터는 잘 알려진 IPv4 주소 192.88.99.1을 사용합니다. 사용자의 사이트와 물리적으로 가장 가까운 애니캐스트 릴레이 라우터가 6to4 터널의 끝점이 됩니다. 이 릴레이 라우터는 6to4 사이트와 원시 IPv6 사이트 간 패킷 전달을 처리합니다.

애니캐스트 6to4 릴레이 라우터에 대한 자세한 내용은 RFC 3068, "An Anycast Prefix for 6to4 Relay Routers" (<ftp://ftp.rfc-editor.org/in-notes/rfc3068.txt>)를 참조하십시오.

- 특정 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

```
# /usr/sbin/6to4relay -e -a relay-router-address
```

-a 옵션은 특정 라우터 주소가 뒤에 이어짐을 나타냅니다. *relay-router-address*는 터널을 사용으로 설정할 특정 6to4 릴레이 라우터의 IPv4 주소로 바꿉니다.

6to4 릴레이 라우터에 대한 터널은 6to4 터널 의사 인터페이스를 제거할 때까지 활성 상태로 유지됩니다.

- 2 터널이 더 이상 필요하지 않을 경우 6to4 릴레이 라우터에 대한 터널을 삭제합니다.

```
# /usr/sbin/6to4relay -d
```

- 3 (옵션) 6to4 릴레이 라우터에 대한 터널이 재부트 후에도 보존되도록 합니다.

6to4 라우터가 재부트될 때마다 사이트에서 6to4 릴레이 라우터에 대한 터널을 원래 상태로 복원해야 하는 이유가 있을 수 있습니다. 이 시나리오를 지원하려면 다음을 수행해야 합니다.

- a. `/etc/default/inetinit` 파일을 편집합니다.

파일의 맨 끝 행을 수정해야 합니다.

- b. `ACCEPT6TO4RELAY=NO` 행의 "NO" 값을 "YES"로 변경합니다.

- c. (옵션) 재부트 후에도 보존되는 특정 6to4 릴레이 라우터에 대한 터널을 만듭니다.

RELAY6TO4ADDR 매개변수에 대해 192.88.99.1 주소를 사용하려는 6to4 릴레이 라우터의 IPv4 주소로 변경합니다.

## 예 6-5 6to4 릴레이 라우터 지원에 대한 상태 정보 가져오기

`/usr/bin/6to4relay` 명령을 사용하여 6to4 릴레이 라우터에 대한 지원을 사용으로 설정할지 여부를 확인할 수 있습니다. 다음 예는 6to4 릴레이 라우터에 대한 지원이 사용 안함으로 설정된 경우(Oracle Solaris의 기본값)의 출력을 보여줍니다.

```
# /usr/sbin/6to4relay
```

```
6to4relay: 6to4 Relay Router communication support is disabled.
```

6to4 릴레이 라우터에 대한 지원이 사용으로 설정되면 다음과 같은 출력이 표시됩니다.

```
# /usr/sbin/6to4relay
```

```
6to4relay: 6to4 Relay Router communication support is enabled.
```

```
IPv4 remote address of Relay Router=192.88.99.1
```

## ▼ IP 터널 구성을 수정하는 방법

- 터널 구성을 변경합니다.

```
# dladm modify-iptun -a [local|remote]=addr,... tunnel-link
```

기존 터널의 유형은 수정할 수 없습니다. 따라서 `-T type` 옵션은 이 명령에 사용할 수 없습니다. 수정 가능한 터널 매개변수는 다음과 같습니다.

```
-a [local|remote]=address,...
```

로컬 주소 및 원격 터널 주소에 해당하는 리터럴 IP 주소 또는 호스트 이름을 지정합니다. 터널의 유형에 따라 주소를 한 개만 지정하거나, 로컬 및 원격 주소를 모두 지정합니다. 로컬 및 원격 주소를 모두 지정하는 경우 주소를 점표로 구분해야 합니다.

- IPv4 터널이 작동하려면 로컬 및 원격 IPv4 주소가 필요합니다.
- IPv6 터널이 작동하려면 로컬 및 원격 IPv6 주소가 필요합니다.
- 6to4 터널이 작동하려면 로컬 IPv4 주소가 필요합니다.

영구 IP 터널 데이터 링크 구성에 호스트 이름을 주소로 사용하는 경우 호스트 이름은 구성 저장소에 저장됩니다. 이후에 시스템을 부트할 때 이름이 터널을 만든 당시에 사용했던 IP 주소와 다른 IP 주소로 분석되는 경우 터널이 새 구성을 사용하게 됩니다.

터널의 로컬 및 원격 주소를 변경하는 경우 해당 주소가 수정하려는 터널의 유형과 일치하는지 확인합니다.

---

주 - 터널 링크의 이름을 변경하려면 `modify-iptun` 하위 명령을 사용하지 마십시오. 대신 `dladm rename-link`를 사용하십시오.

```
# dladm rename-link old-tunnel-link new-tunnel-link
```

마찬가지로, `hoplimit` 또는 `encaplimit`와 같은 터널 등록 정보를 변경하려면 `modify-iptun` 명령을 사용하지 마십시오. 대신 `dladm set-linkprop` 명령을 사용하여 해당 등록 정보의 값을 설정하십시오.

---

### 예 6-6 터널의 주소 및 등록 정보 수정

이 예는 두 개의 절차로 구성됩니다. 먼저 IPv4 터널 `vpn0`의 로컬 및 원격 주소가 일시적으로 변경됩니다. 나중에 시스템을 재부트하면 터널이 원래 주소를 사용하도록 복원됩니다. 두 번째 절차는 `vpn0`의 `hoplimit`를 60으로 변경합니다.



```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.1.2.3 vpn0
# dladm set-linkprop -p hoplimit=60 vpn0
```

## ▼ IP 터널 구성을 표시하는 방법

- IP 터널 구성을 표시합니다.

```
# dladm show-iptun [-p] -o fields [tunnel-link]
```

이 명령과 함께 사용할 수 있는 옵션은 다음과 같습니다.

-p	시스템에서 분석 가능한 형식으로 정보를 표시합니다. 이 인수는 선택적입니다.
-o fields	특정 터널 정보를 표시하는 선택한 필드를 표시합니다.
tunnel-link	표시할 구성 정보를 포함하는 터널을 지정합니다. 이 인수는 선택적입니다. 터널 이름을 생략하면 시스템에 있는 모든 터널에 대한 정보가 표시됩니다.

### 예 6-7 모든 터널에 대한 정보 표시

이 예에서는 한 개의 터널만 시스템에 존재합니다.

```
# dladm show-iptun
LINK    TYPE    FLAGS    LOCAL          REMOTE
tun0    6to4    --       192.168.35.10  --
vpn0    ipv4    --       10.8.48.149    192.1.2.3
```

### 예 6-8 시스템에서 분석 가능한 형식으로 선택한 필드 표시

이 예에서는 터널 정보를 포함하는 특정 필드만 표시됩니다.

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

## ▼ IP 터널 등록 정보를 표시하는 방법

- 터널 링크의 등록 정보를 표시합니다.

```
# dladm show-linkprop [-c] [-o fields] [tunnel-link]
```

이 명령과 함께 사용할 수 있는 옵션은 다음과 같습니다.

- c            시스템에서 분석 가능한 형식으로 정보를 표시합니다. 이 인수는 선택적입니다.
- o fields    링크 등록 정보에 대한 특정 정보를 제공하는 선택한 필드를 표시합니다.
- tunnel-link   표시할 등록 정보에 대한 정보를 포함하는 터널을 지정합니다. 이 인수는 선택적입니다. 터널 이름을 생략하면 시스템에 있는 모든 터널에 대한 정보가 표시됩니다.

예 6-9    터널 등록 정보 표시

이 예는 터널의 링크 등록 정보를 모두 표시하는 방법을 보여줍니다.

```
# dladm show-linkprop tun0
```

LINK	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
tun0	autopush	--	--	--	--
tun0	zone	rw	--	--	--
tun0	state	r-	up	up	up,down
tun0	mtu	r-	65515	--	576-65495
tun0	maxbw	rw	--	--	--
tun0	cpus	rw	--	--	--
tun0	priority	rw	high	high	low,medium,high
tun0	hoplimit	rw	64	64	1-255

▼ IP 터널을 삭제하는 방법

- 1    인터페이스의 유형에 따라 적절한 구문을 사용하여 터널을 경유하도록 구성된 IP 인터페이스를 제거합니다.

```
# ipadm delete-ip tunnel-link
```

주 - 터널을 성공적으로 삭제하기 위해서는 터널에 설정된 기존 IP 인터페이스가 없어야 합니다.

- 2    IP 터널을 삭제합니다.

```
# dladm delete-iptun tunnel-link
```

이 명령의 유일한 옵션은 터널을 일시적으로 삭제하는 -t입니다. 시스템을 재부트하면 터널이 복원됩니다.

예 6-10    IPv6 인터페이스로 구성된 IPv6 터널 삭제

이 예에서는 영구 터널이 영구적으로 삭제됩니다.

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

## 네트워크 문제 해결

---

이 장에서는 네트워크에서 발생할 수 있는 일반적인 문제에 대한 해결 방법에 대해 설명합니다. 다음 항목을 다룹니다.

- 131 페이지 “일반 네트워크 문제 해결 팁”
- 132 페이지 “IPv6 배치 시 발생하는 일반적인 문제”

### 일반 네트워크 문제 해결 팁

네트워크 문제를 나타내는 첫번째 신호 중 하나는 하나 이상의 호스트에서 통신이 끊기는 것입니다. 호스트가 처음에 네트워크에 추가될 때부터 호스트가 응답이 없다면 구성 파일 중 하나에 문제가 있는 것일 수 있습니다. 잘못된 네트워크 인터페이스 카드도 문제일 수 있습니다. 한 호스트에서 갑자기 문제가 발생한다면 네트워크 인터페이스가 원인일 수 있습니다. 네트워크의 호스트가 서로 통신할 수는 있지만 다른 네트워크와 통신할 수 없는 경우 라우터 문제일 수 있습니다. 또는 다른 네트워크에 문제가 있는 것일 수 있습니다.

`ipadm` 명령을 사용하면 네트워크 인터페이스에 대한 정보를 얻을 수 있습니다. `netstat` 명령을 사용하면 경로 설정표 및 프로토콜 통계를 표시할 수 있습니다. 타사 네트워크 진단 프로그램은 여러 가지 문제 해결 도구를 제공합니다. 자세한 내용은 타사 설명서를 참조하십시오.

네트워크 성능을 저하시키는 문제의 원인은 명확하지 않습니다. 예를 들어 `ping`과 같은 도구를 사용하여 호스트에 의한 패킷 손실과 같은 문제를 수량화할 수 있습니다.

### 기본 진단 검사 실행

네트워크 문제가 발생하면 일련의 소프트웨어 검사를 실행하여 기본적인 소프트웨어 관련 문제를 진단하고 수정할 수 있습니다.

## ▼ 기본 네트워크 소프트웨어 검사를 수행하는 방법

- 1 **netstat** 명령을 사용하여 네트워크 정보를 표시합니다.

**netstat** 명령에 대한 구문 및 정보는 [91 페이지 “netstat 명령으로 네트워크 상태 모니터링”](#) 및 [netstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

- 2 **hosts** 데이터베이스를 검사하여 항목이 올바르며 최신 상태인지 확인합니다.

`/etc/inet/hosts` 데이터베이스에 대한 자세한 내용은 [135 페이지 “네트워크 구성 파일”](#) 및 [hosts\(4\)](#) 매뉴얼 페이지를 참조하십시오.

- 3 **RARP**(Reverse Address Resolution Protocol)를 실행 중인 경우 **ethers** 데이터베이스에서 이더넷 주소를 검사하여 항목이 올바르며 최신 상태인지 확인합니다.

- 4 **telnet** 명령을 사용하여 로컬 호스트에 연결을 시도합니다.

**telnet**에 대한 구문 및 정보는 [telnet\(1\)](#) 매뉴얼 페이지를 참조하십시오.

- 5 네트워크 데몬 **inetd**가 실행 중인지 확인합니다.

```
# ps -ef | grep inetd
```

다음 출력은 **inetd** 데몬이 실행 중인지 확인합니다.

```
root 57 1 0 Apr 04 ? 3:19 /usr/sbin/inetd -s
```

- 6 네트워크에서 **IPv6**이 사용 가능한 경우 **IPv6** 데몬 **in.ndpd**가 실행 중인지 확인합니다.

```
# ps -ef | grep in.ndpd
```

다음 출력은 **in.ndpd** 데몬이 실행 중인지 확인합니다.

```
root 123 1 0 Oct 27 ? 0:03 /usr/lib/inet/in.ndpd
```

## IPv6 배치 시 발생하는 일반적인 문제

이 절에서는 사이트에서 **IPv6**을 계획하고 배치할 때 발생할 수 있는 문제에 대해 설명합니다. 실제 계획 작업은 [2 장, “IPv6 주소 사용 시 고려 사항”](#)을 참조하십시오.

### IPv4 라우터를 IPv6으로 업그레이드할 수 없음

기존 장비를 업그레이드할 수 없는 경우 **IPv6** 지원 장비를 구매해야 할 수 있습니다. **IPv6** 지원을 위해 수행해야 하는 장비별 절차는 제조업체의 설명서를 확인하십시오.

특정 **IPv4** 라우터는 **IPv6** 지원을 위해 업그레이드할 수 없습니다. 이 상황이 사용자의 토폴로지에 해당하는 경우 물리적으로 **IPv6** 라우터를 **IPv4** 라우터 옆에 연결하십시오. 그런 다음 **IPv4** 라우터를 통해 **IPv6** 라우터에서 터널링할 수 있습니다. 터널 구성 작업은 [119 페이지 “dladm 명령을 통한 터널 구성 및 관리”](#)를 참조하십시오.

## IPv6으로 서비스 업그레이드 후 발생하는 문제

IPv6을 지원하도록 서비스를 준비할 때 다음과 같은 상황이 발생할 수 있습니다.

- 특정 응용 프로그램의 경우 IPv6으로 이식한 후에도 기본적으로 IPv6 지원이 설정되지 않습니다. IPv6을 설정하도록 이 응용 프로그램을 구성해야 할 수 있습니다.
- 서버에서 여러 서비스를 실행하고 이 중 일부는 IPv4 전용 서비스, 일부는 IPv4 및 IPv6 서비스인 경우 문제가 발생할 수 있습니다. 일부 클라이언트에서 두 유형의 서비스를 사용해야 하는데 이로 인해 서버 측에서 혼동을 일으킬 수 있습니다.

## 현재 ISP가 IPv6을 지원하지 않음

IPv6을 배치하려는데 현재 ISP가 IPv6 주소 지정을 제공하지 않을 경우 ISP를 변경하지 않는 다음 대안을 고려하십시오.

- 사이트에서 IPv6 통신용 다른 회선을 제공하는 ISP를 이용합니다. 이 솔루션은 비용이 많이 듭니다.
- 가상 ISP를 사용합니다. 가상 ISP는 사이트에 링크가 아닌 IPv6 연결을 제공합니다. 대신 사용자의 사이트에서 IPv4 ISP를 경유하여 가상 ISP에 연결되는 터널을 만듭니다.
- 사용자의 ISP를 경유하여 다른 IPv6 사이트에 연결되는 6to4 터널을 사용합니다. 주소의 경우, 6to4 라우터의 등록된 IPv4 주소를 IPv6 주소의 공용 토폴로지 부분으로 사용합니다.

## 6to4 릴레이 라우터로 터널링 시 발생하는 보안 문제

원래 6to4 라우터와 6to4 릴레이 라우터 간 터널은 비보안 상태입니다. 따라서 터널에는 다음과 같은 보안 문제가 내재되어 있습니다.

- 6to4 릴레이 라우터는 패킷 캡슐화 및 캡슐화 해제를 수행하지만 패킷 내에 포함된 데이터는 검사하지 않습니다.
- 6to4 릴레이 라우터에서 주로 발생하는 문제는 주소 스푸핑입니다. 수신 트래픽의 경우 6to4 라우터가 릴레이 라우터의 IPv4 주소를 소스의 IPv6 주소에 대응시킬 수 없습니다. 따라서 IPv6 호스트의 주소가 쉽게 스푸핑될 수 있습니다. 6to4 릴레이 라우터의 주소도 스푸핑될 수 있습니다.
- 기본적으로 6to4 라우터와 6to4 릴레이 라우터 간에는 신뢰 방식이 존재하지 않습니다. 따라서 6to4 라우터는 6to4 릴레이 라우터를 신뢰할 수 있는지 또는 적합한 6to4 릴레이 라우터인지 식별할 수 없습니다. 6to4 사이트와 IPv6 대상 간에는 신뢰 관계가 존재해야 합니다. 그렇지 않으면 두 사이트가 공격 받을 가능성이 있습니다.

6to4 릴레이 라우터에 내재되어 있는 이러한 문제 및 기타 문제는 **Security Considerations for 6to4**의 Internet Draft에 설명되어 있습니다. 일반적으로 다음과 같은 경우에만 6to4 릴레이 라우터에 대한 지원을 사용으로 설정해야 합니다.

- 6to4 사이트가 신뢰할 수 있는 개인 IPv6 네트워크와 통신하려는 경우. 예를 들어 분리된 6to4 사이트와 원시 IPv6 사이트로 구성된 캠퍼스 네트워크에서 6to4 릴레이 라우터 지원을 사용으로 설정할 수 있습니다.
- 6to4 사이트가 특정 원시 IPv6 호스트와 통신할 수 밖에 없는 비즈니스 이유가 있는 경우
- **Security Considerations for 6to4**, Internet Draft에서 권장하는 검사 및 신뢰 모델을 구현한 경우

## IPv4 참조

---

이 장에서는 파일 항목의 유형, 용도 및 형식을 포함하여 네트워크 구성 파일에 대한 TCP/IP 네트워크 참조 정보를 제공합니다.

이 장은 다음 정보를 포함합니다.

- 135 페이지 “네트워크 구성 파일”
- 136 페이지 “inetd Internet Services Daemon”
- 137 페이지 “name-service/switch SMF 서비스”
- 139 페이지 “Oracle Solaris의 경로 지정 프로토콜”

## 네트워크 구성 파일

네트워크에서 구성 정보는 네트워크의 작동 방식을 규제하는 여러 파일과 데이터베이스에 저장됩니다. 이 절에서는 이러한 파일에 대한 간략한 설명을 제공합니다. 네트워크에 대한 변경 사항을 구현할 때 일부 파일은 업데이트 및 유지 관리가 필요합니다. 거의 또는 전혀 관리가 필요하지 않은 파일도 있습니다.

<code>/etc/defaultrouter</code>	이 파일에는 네트워크에 직접 연결된 라우터의 IP 인터페이스 이름이 포함됩니다. 시스템에서 이 파일은 선택 사항입니다. 파일이 존재할 경우 시스템은 정적 경로 지정을 지원하도록 구성됩니다.
<code>/etc/inet/hosts</code>	이 파일에는 네트워크의 IPv4 주소와 이 주소를 구성하는 해당 인터페이스 이름이 포함됩니다. NIS 또는 DNS 이름 서비스나 LDAP 디렉토리 서비스를 사용하는 경우 호스트 정보는 서버에 존재하는 다른 데이터베이스(예: <code>hosts.byname</code> )에 저장됩니다. 자세한 내용은 <a href="#">Oracle Solaris Administration: Naming and Directory Services</a> 를 참조하십시오.
<code>/etc/inet/netmasks</code>	이 파일에는 네트워크 번호(예: <code>192.168.0.0</code> ) 및 해당 네트워크 번호의 네트마스크 정보(예: <code>255.255.255.0</code> )가 포함됩니다. NIS 또는 LDAP를 사용하는 네트워크에서 이 정보는 서버의

	네트마스크 데이터베이스에 저장됩니다. 자세한 내용은 <a href="#">netmasks(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>/etc/bootparams</code>	이 파일에는 네트워크 클라이언트 모드로 부트하도록 구성된 시스템에 대한 부트 프로세스를 결정하는 매개변수가 포함됩니다. 자세한 내용은 <a href="#">51 페이지 “시스템 구성 모드 설정”</a> 을 참조하십시오. 이 파일은 로컬 파일 모드를 사용하지 않는 경우 이름 서비스에서 사용하는 <code>bootparams</code> 데이터베이스를 만들기 위한 기준입니다. 이 파일의 내용 및 형식에 대한 자세한 내용은 <a href="#">bootparams(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>/etc/ethers</code>	이 파일은 호스트 이름과 해당 MAC 주소를 연결합니다. 이 파일은 시스템이 네트워크 클라이언트로 구성된 네트워크에서 사용할 <code>ethers</code> 데이터베이스를 만들기 위한 기준입니다. 자세한 내용은 <a href="#">ethers(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>/etc/inet/networks</code>	이 파일은 네트워크 이름과 네트워크 번호를 연관시켜 놓았습니다. 주석으로 데이터베이스의 각 항목에 대한 부연 설명을 추가할 수도 있습니다. 이 파일이 있기 때문에 응용 프로그램에서 네트워크 번호 대신 네트워크 이름을 사용하고 표시할 수 있습니다. 예를 들어, <code>netstat</code> 프로그램은 이 데이터베이스의 정보를 사용하여 상태 테이블을 생성합니다. 라우터를 통해 로컬 네트워크에 연결하는 모든 부속 네트워크는 이 파일에 포함되어야 합니다. 자세한 내용은 <a href="#">networks(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>/etc/inet/protocols</code>	이 파일은 시스템에 설치된 TCP/IP 프로토콜 및 해당 프로토콜 번호를 나열합니다. 이 파일은 관리가 거의 필요하지 않습니다. 자세한 내용은 <a href="#">protocols(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>/etc/inet/services</code>	이 파일은 TCP와 UDP 서비스의 이름 및 잘 알려진 해당 포트 번호를 나열합니다. 이 파일은 네트워크 서비스를 호출하는 프로그램에서 사용됩니다. 일반적으로 이 파일은 관리가 필요하지 않습니다. 자세한 내용은 <a href="#">services(4)</a> 매뉴얼 페이지를 참조하십시오.

## inetd Internet Services Daemon

`inetd` 데몬은 시스템이 부트할 때 인터넷 표준 서비스를 시작하고 시스템이 실행 중일 때 서비스를 다시 시작할 수 있습니다. `inetd` 데몬에서 시작되는 표준 인터넷 서비스를 수정하거나 서비스를 추가하려면 SMF(서비스 관리 기능)를 사용합니다.

`inetd`에서 시작되는 서비스를 관리하려면 다음 SMF 명령을 사용합니다.



svcadm	서비스에 대한 관리 작업(사용으로 설정, 사용 안함으로 설정 또는 다시 시작 등)에 사용됩니다. 자세한 내용은 <a href="#">svcadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
svcs	서비스 상태 쿼리에 사용됩니다. 자세한 내용은 <a href="#">svcs(1)</a> 매뉴얼 페이지를 참조하십시오.
inetadm	서비스의 등록 정보 표시 및 수정에 사용됩니다. 자세한 내용은 <a href="#">inetadm(1M)</a> 매뉴얼 페이지를 참조하십시오.

특정 서비스에 대한 `inetadm` 프로파일의 `proto` 필드는 서비스가 실행되는 전송 계층 프로토콜을 나타냅니다. 서비스가 IPv4 전용인 경우 `proto` 필드가 `tcp`, `udp` 또는 `sctp`로 지정되어야 합니다.

- SMF 명령 사용 지침은 [Oracle Solaris 관리: 일반 작업의 “SMF 명령줄 관리 유틸리티”](#)를 참조하십시오.
- SMF 명령을 사용하여 SCTP를 통해 실행되는 서비스를 추가하는 작업은 [70 페이지 “SCTP 프로토콜을 사용하는 서비스를 추가하는 방법”](#)을 참조하십시오.
- IPv4 요청과 IPv6 요청을 모두 처리하는 서비스 추가에 대한 자세한 내용은 [136 페이지 “inetd Internet Services Daemon”](#)을 참조하십시오.

## name-service/switch SMF 서비스

`name-service/switch` SMF 서비스는 구성 정보에 대한 네트워크 데이터베이스의 검색 순서를 정의합니다. 이전에 구성 파일에 저장되었던 네트워크 구성 정보 중 일부(예: 기본 도메인)는 이 SMF 서비스의 등록 정보가 되도록 변환되었습니다. 이 SMF 서비스의 등록 정보는 시스템에서 이름 서비스의 구현을 결정합니다. 등록 정보는 다음과 같습니다.

```
% svccfg -s name-service/switch listprop config
config                                application
config/value_authorization           astring      solaris.smf.value.name-service.switch
config/default                       astring      files
config/password                      astring      "files nis"
config/group                         astring      "files nis"
config/host                          astring      "files dns nis"
config/network                       astring      "nis [NOTFOUND=return] files"
config/protocol                      astring      "nis [NOTFOUND=return] files"
config/rpc                           astring      "nis [NOTFOUND=return] files"
config/ether                          astring      "nis [NOTFOUND=return] files"
config/netmask                       astring      "files nis"
config/bootparam                     astring      "nis [NOTFOUND=return] files"
config/publickey                     astring      "nis [NOTFOUND=return] files"
config/netgroup                      astring      nis
config/automount                     astring      "files nis"
config/alias                         astring      "files nis"
config/service                       astring      "files nis"
config/printer                       astring      "user nis"
config/auth_attr                     astring      "files nis"
```

config/prof_attr	astring	"files nis"
config/project	astring	"files nis"

각 등록 정보에 대해 설정된 값은 네트워크 사용자에게 영향을 주는 정보(예: 암호, 별칭 또는 네트워크 마스크)를 검색하는 이름 서비스를 결정합니다. 예를 들어, 자동 마운트 및 암호 등록 정보는 files 및 nis로 설정됩니다. 따라서 자동 마운트 정보 및 암호 정보는 파일과 NIS 서비스에서 가져옵니다.

한 이름 서비스에서 다른 이름 서비스로 변경하려는 경우 선택한 이름 서비스를 사용으로 설정하도록 name-service/switch SMF 서비스의 해당 등록 정보를 설정해야 합니다.

예를 들어, 네트워크에서 LDAP 이름 지정 서비스를 사용하려는 경우를 가정해 보겠습니다. SMF 서비스의 다음 등록 정보를 구성해야 합니다.

- config/default가 파일 및 LDAP를 사용하도록 설정되어야 합니다.
- config/host가 파일 및 DNS를 사용하도록 설정되어야 합니다.
- config/netgroup이 LDAP를 사용하도록 설정되어야 합니다.
- config/printer가 사용자, 파일 및 LDAP를 사용하도록 설정되어야 합니다.

그러므로 이러한 등록 정보를 올바르게 설정하려면 다음 명령을 입력해야 합니다.

```
# svccfg -s name-service/switch setprop config/default = astring: "files ldap"
# svccfg -s name-service/switch setprop config/host = astring: "files dns"
# svccfg -s name-service/switch setprop config/netgroup = astring: "ldap"
# svccfg -s name-service/switch setprop config/printer = astring: "user files ldap"
# svccfg -s name-service/switch:default refresh
```

이름 서비스 스위치에 대한 자세한 내용은 [Oracle Solaris Administration: Naming and Directory Services](#)를 참조하십시오.

## 네트워크 데이터베이스에 대한 이름 서비스의 영향

네트워크 데이터베이스의 형식은 해당 네트워크에 대해 선택하는 이름 서비스의 유형에 따라 달라집니다. 예를 들어, hosts 데이터베이스에는 적어도 로컬 시스템의 호스트 이름과 IPv4 주소 및 로컬 시스템에 직접 연결된 네트워크 인터페이스가 포함됩니다. 하지만 hosts 데이터베이스에는 네트워크의 서비스 이름 유형에 따라 다른 IPv4 주소와 호스트 이름이 포함될 수 있습니다.

네트워크 데이터베이스는 다음과 같이 사용됩니다.

- 이름 서비스에 대해 로컬 파일을 사용하는 네트워크는 /etc/inet 및 /etc 디렉토리의 파일에 의존합니다.
- NIS는 NIS 맵이라는 데이터베이스를 사용합니다.
- DNS는 호스트 정보가 있는 레코드를 사용합니다.

---

주-DNS 부트 및 데이터 파일은 네트워크 데이터베이스에 직접 연결되지 않습니다.

---

NIS, DNS 및 LDAP에서 네트워크 데이터베이스 연결에 대한 자세한 내용은 [Oracle Solaris Administration: Naming and Directory Services](#)를 참조하십시오.

## Oracle Solaris의 경로 지정 프로토콜

이 절에서는 Oracle Solaris에서 지원되는 두 가지 경로 지정 프로토콜인 RIP(Routing Information Protocol) 및 RDISC(ICMP Router Discovery)에 대해 설명합니다. RIP 및 RDISC는 모두 표준 TCP/IP 프로토콜입니다. Oracle Solaris에서 사용 가능한 전체 경로 지정 프로토콜 목록은 [표 8-1](#) 및 [표 8-2](#)를 참조하십시오.

### RIP(Routing Information Protocol)

RIP은 시스템이 부트할 때 자동으로 시작되는 경로 지정 데몬인 `in.routed`로 구현됩니다. 라우터에서 `s` 옵션을 지정하여 실행하면 `in.routed`는 커널 경로 지정 테이블을 모든 접근 가능한 네트워크에 대한 경로로 채우고 모든 네트워크 인터페이스를 통해 “접근 가능성”을 알립니다.

호스트에서 `q` 옵션을 지정하여 실행하면 `in.routed`는 경로 지정 정보를 추출하지만 접근 가능성을 알리지는 않습니다. 호스트에서 경로 지정 정보는 두 가지 방법으로 추출할 수 있습니다.

- `s` 플래그(대문자 “S”: “공간 절약 모드”)를 지정하지 **않습니다**. `in.routed`는 라우터에서 만드는 것과 동일하게 전체 경로 지정 테이블을 만듭니다.
- `s` 플래그를 지정합니다. `in.routed`는 각 사용 가능한 라우터에 대해 단일 기본 경로가 포함된 최소 커널 테이블을 만듭니다.

### RDISC(ICMP Router Discovery) 프로토콜

호스트는 RDISC를 사용하여 라우터에서 경로 지정 정보를 가져옵니다. 따라서 호스트에서 RDISC를 실행하는 경우 라우터 정보를 교환하려면 라우터도 다른 프로토콜(예: RIP)을 실행해야 합니다.

RDISC는 라우터와 호스트에서 모두 실행되어야 하는 `in.routed`로 구현됩니다. 호스트에서 `in.routed`는 RDISC를 사용하여 RDISC를 통해 자신을 알리는 라우터에서 기본 경로를 찾습니다. 라우터에서 `in.routed`는 RDISC를 사용하여 직접 연결된 네트워크의 호스트에 기본 경로를 알립니다. [in.routed\(1M\)](#) 매뉴얼 페이지 및 [gateways\(4\)](#) 매뉴얼 페이지를 참조하십시오.

# Oracle Solaris의 경로 지정 프로토콜 표

다음 표는 Oracle Solaris에서 지원되는 모든 경로 지정 프로토콜을 나열합니다.

표 8-1 Oracle Solaris 경로 지정 프로토콜

프로토콜	연결된 데몬	설명	수행 방법
RIP(Routing Information Protocol)	in.routed	IPv4 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리하는 IGP입니다.	56 페이지 “IPv4 라우터 구성 방법”
ICMP(Internet Control Message Protocol) 라우터 검색	in.routed	호스트에서 네트워크의 라우터를 검색하는 데 사용됩니다.	64 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법” 및 66 페이지 “단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법”
RIPng(Routing Information Protocol, next generation) 프로토콜	in.ripngd	IPv6 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리하는 IGP입니다.	78 페이지 “IPv6 지원 라우터를 구성하는 방법”
ND(Neighbor Discovery) 프로토콜	in.ndpd	IPv6 라우터의 존재를 알리고 네트워크의 IPv6 호스트를 검색합니다.	75 페이지 “IPv6 인터페이스 구성”

다음 표는 Oracle Solaris에서도 지원되는 Quagga 프로토콜을 나열합니다.

표 8-2 OpenSolaris Quagga 프로토콜

프로토콜	데몬	설명
RIP 프로토콜	ripd	IPv4 패킷을 경로 지정하고 주변에 경로 지정 테이블을 알리는 IPv4 거리 벡터링 IGP입니다.
RIPng	ripngd	IPv6 거리 벡터링 IGP입니다. IPv6 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리합니다.
OSPF(Open Shortest Path First) 프로토콜	ospfd	패킷 경로 지정 및고가용성 네트워킹을 위한 IPv4 링크 상태 IGP입니다.
BGP(Border Gateway Protocol)	bgpd	관리 도메인 간에 경로 지정을 위한 IPv4 및 IPv6 EGP입니다.

## IPv6 참조

---

이 장에서는 Oracle Solaris IPv6 구현에 대한 다음 참조 정보에 대해 설명합니다.

- 141 페이지 “Oracle Solaris IPv6 구현”
- 152 페이지 “IPv6 Neighbor Discovery 프로토콜”
- 158 페이지 “IPv6 경로 지정”
- 159 페이지 “Oracle Solaris 이름 서비스에 대한 IPv6 확장”
- 160 페이지 “NFS 및 RPC IPv6 지원”
- 160 페이지 “IPv6 Over ATM 지원”

IPv6 개요는 **System Administration Guide: IP Services**의 3 장, “Introducing IPv6 (Overview)”를 참조하십시오. IPv6 지원 네트워크 구성에 대한 작업은 4 장, “네트워크에서 IPv6 사용”을 참조하십시오. IP 터널에 대한 모든 정보는 6 장, “IP 터널 구성”을 참조하십시오.

## Oracle Solaris IPv6 구현

이 절에서는 Oracle Solaris에서 IPv6을 사용하는 파일, 명령 및 데몬에 대해 설명합니다. IPv6 주소 지정 및 IPv6 헤더 형식에 대한 보다 자세한 개요는 **System Administration Guide: IP Services**의 “IPv6 Addressing Formats Beyond the Basics”를 참조하십시오.

## IPv6 구성 파일

이 절에서는 IPv6 구현에 포함된 구성 파일에 대해 설명합니다.

- 142 페이지 “ndpd.conf 구성 파일”
- 145 페이지 “/etc/inet/ipaddrsel.conf 구성 파일”

## ndpd.conf 구성 파일

/etc/inet/ndpd.conf 파일은 in.ndpd Neighbor Discovery 데몬에서 사용하는 옵션을 구성하는 데 사용됩니다. 라우터의 경우 주로 ndpd.conf를 사용하여 링크에 알릴 사이트 접두어를 구성하십시오. 호스트의 경우 ndpd.conf를 사용하여 주소 자동 구성을 해제하거나 임시 주소를 구성하십시오.

다음 표는 ndpd.conf 파일에 사용되는 키워드를 보여줍니다.

표 9-1 /etc/inet/ndpd.conf 키워드

변수	설명
ifdefault	모든 인터페이스에 대한 라우터 동작을 지정합니다. 라우터 매개변수 및 해당 값을 설정하려면 다음 구문을 사용하십시오.  ifdefault [variable-value]
prefixdefault	접두어 알림에 대한 기본 동작을 지정합니다. 라우터 매개변수 및 해당 값을 설정하려면 다음 구문을 사용하십시오.  prefixdefault [variable-value]
if	인터페이스별 매개변수를 설정합니다. 다음 구문을 사용하십시오.  if interface [variable-value]
prefix	인터페이스별 접두어 정보를 알립니다. 다음 구문을 사용하십시오.  prefix prefix/length interface [variable-value]

ndpd.conf 파일에서 이 표의 키워드를 라우터 구성 변수 세트와 함께 사용하십시오. 이러한 변수는 [RFC 2461, Neighbor Discovery for IP Version 6 \(IPv6\)](http://www.ietf.org/rfc/rfc2461.txt?number=2461) (<http://www.ietf.org/rfc/rfc2461.txt?number=2461>)에 자세히 정의되어 있습니다.

다음 표는 인터페이스 구성에 사용되는 변수를 간략한 설명과 함께 보여줍니다.

표 9-2 /etc/inet/ndpd.conf 인터페이스 구성 변수

변수	기본값	정의
AdvRetransTimer	0	라우터에서 보내는 알림 메시지의 Retrans Timer(재전송 타이머) 필드 값을 지정합니다.
AdvCurHopLimit	인터넷의 현재 반경	라우터에서 보내는 알림 메시지의 현재 홉 한계로 지정할 값을 지정합니다.
AdvDefaultLifetime	3 + MaxRtrAdvInterval	라우터 알림의 기본 수명을 지정합니다.
AdvLinkMTU	0	라우터에서 전송할 MTU(최대 전송 단위) 값을 지정합니다. 0은 라우터에 MTU 옵션이 지정되지 않았음을 나타냅니다.

표 9-2 /etc/inet/ndpd.conf 인터페이스 구성 변수 (계속)

변수	기본값	정의
AdvManaged Flag	False	라우터 알림의 Manage Address Configuration(주소 구성 관리) 플래그에 지정할 값을 나타냅니다.
AdvOtherConfigFlag	False	라우터 알림의 Other Stateful Configuration(기타 Stateful 구성) 플래그에 지정할 값을 나타냅니다.
AdvReachableTime	0	라우터에서 보내는 알림 메시지의 Reachable Time(연결 가능 시간) 필드 값을 지정합니다.
AdvSendAdvertisements	False	노드가 알림을 전송하고 라우터 요청에 응답할지 여부를 나타냅니다. 라우터 알림 기능을 설정하려면 ndpd.conf 파일에서 이 변수를 명시적으로 “TRUE”로 설정해야 합니다. 자세한 내용은 78 페이지 “IPv6 지원 라우터를 구성하는 방법”을 참조하십시오.
DupAddrDetect Transmits	1	로컬 노드 주소의 중복 주소 감지 중 Neighbor Discovery 프로토콜이 보내야 하는 연속 이웃 요청 메시지 수를 정의합니다.
MaxRtrAdvInterval	600초	요청되지 않은 멀티캐스트 알림을 보내는 최대 간격을 지정합니다.
MinRtrAdvInterval	200초	요청되지 않은 멀티캐스트 알림을 보내는 최소 간격을 지정합니다.
StatelessAddrConf	True	Stateless 주소 자동 구성을 통해 노드에서 IPv6 주소가 구성되는지 여부를 제어합니다. ndpd.conf에서 False가 선언된 경우 주소를 수동으로 구성해야 합니다. 자세한 내용은 83 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”을 참조하십시오.
TmpAddrsEnabled	False	모든 인터페이스에 대해 또는 노드의 특정 인터페이스에 대해 임시 주소를 생성할지 여부를 나타냅니다. 자세한 내용은 81 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.
TmpMaxDesyncFactor	600초	in.ndpd가 시작되면 선호 수명 변수 TmpPreferredLifetime에서 차감될 임의 값을 지정합니다. TmpMaxDesyncFactor 변수의 목적은 네트워크에 있는 모든 시스템이 임시 주소를 동시에 재생성하지 않도록 하는 것입니다. TmpMaxDesyncFactor를 사용하여 해당 임의 값에 대한 상한을 변경할 수 있습니다.
TmpPreferredLifetime	False	임시 주소의 선호 수명을 설정합니다. 자세한 내용은 81 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.
TmpRegenAdvance	False	임시 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 자세한 내용은 81 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.
TmpValidLifetime	False	임시 주소의 유효 수명을 설정합니다. 자세한 내용은 81 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.

다음 표는 IPv6 접두어 구성에 사용되는 변수를 보여줍니다.

표 9-3 /etc/inet/ndpd.conf 접두어 구성 변수

변수	기본값	정의
AdvAutonomousFlag	True	Prefix Information(접두어 정보) 옵션의 Autonomous Flag(자동 플래그) 필드에 지정할 값을 지정합니다.
AdvOnLinkFlag	True	Prefix Information(접두어 정보) 옵션의 온-링크 플래그("L-bit")에 지정할 값을 지정합니다.
AdvPreferredExpiration	Not set	접두어의 선호 만료 날짜를 지정합니다.
AdvPreferredLifetime	604800초	Prefix Information(접두어 정보) 옵션의 선호 수명에 지정할 값을 지정합니다.
AdvValidExpiration	Not set	접두어의 유효 만료 날짜를 지정합니다.
AdvValidLifetime	2592000초	구성할 접두어의 유효 수명을 지정합니다.

예 9-1 /etc/inet/ndpd.conf 파일

다음 예는 ndpd.conf 파일에서 키워드 및 구성 변수가 사용되는 방식을 보여줍니다. 변수를 활성화하려면 주석(#)을 제거하십시오.

```
# ifdefault      [variable-value ]*
# prefixdefault [variable-value ]*
# if ifname      [variable-value ]*
# prefix prefix/length ifname
#
# Per interface configuration variables
#
#DupAddrDetectTransmits
#AdvSendAdvertisements
#MaxRtrAdvInterval
#MinRtrAdvInterval
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix: AdvPrefixList configuration variables
#
#
#AdvValidLifetime
#AdvOnLinkFlag
#AdvPreferredLifetime
#AdvAutonomousFlag
#AdvValidExpiration
#AdvPreferredExpiration
```



**예 9-1 /etc/inet/ndpd.conf 파일 (계속)**

```

ifdefault AdvReachableTime 30000 AdvRetransTimer 2000
prefixdefault AdvValidLifetime 240m AdvPreferredLifetime 120m

if qe0 AdvSendAdvertisements 1
prefix 2:0:0:56::/64 qe0
prefix fec0:0:0:56::/64 qe0

if qe1 AdvSendAdvertisements 1
prefix 2:0:0:55::/64 qe1
prefix fec0:0:0:56::/64 qe1

if hme1 AdvSendAdvertisements 1
prefix 2002:8192:56bb:1::/64 qfe0

if hme1 AdvSendAdvertisements 1
prefix 2002:8192:56bb:2::/64 hme1

```

**/etc/inet/ipaddrsel.conf 구성 파일**

/etc/inet/ipaddrsel.conf 파일에는 IPv6 기본 주소 선택 정책 테이블이 포함되어 있습니다. IPv6이 사용 가능한 상태로 Oracle Solaris를 설치하면 이 파일에는 [표 9-4](#)에 표시된 내용이 포함됩니다.

/etc/inet/ipaddrsel.conf의 내용은 편집할 수 있습니다. 그러나 대부분의 경우 이 파일을 수정하지 않는 것이 좋습니다. 수정이 필요할 경우 [109 페이지 “IPv6 주소 선택 정책 테이블을 관리하는 방법”](#) 절차를 참조하십시오. ipaddrsel.conf에 대한 자세한 내용은 [146 페이지 “IPv6 주소 선택 정책 테이블을 수정하는 이유”](#) 및 ipaddrsel.conf(4) 매뉴얼 페이지를 참조하십시오.

**IPv6 관련 명령**

이 절에서는 Oracle Solaris IPv6 구현으로 추가된 명령에 대해 설명합니다. 또한 IPv6을 지원하도록 기존 명령을 수정하는 방법에 대해서도 설명합니다.

**ipaddrsel 명령**

ipaddrsel 명령을 사용하여 IPv6 기본 주소 선택 정책 테이블을 수정할 수 있습니다.

Oracle Solaris 커널은 IPv6 기본 주소 선택 정책 테이블을 사용하여 IPv6 패킷 헤더에 대한 대상 주소 순서 지정 및 소스 주소 선택을 수행합니다. /etc/inet/ipaddrsel.conf 파일에는 정책 테이블이 포함되어 있습니다.

다음 표는 정책 테이블의 기본 주소 형식 및 우선 순위를 보여줍니다. IPv6 주소 선택에 대한 기술적인 세부 정보는 [inet6\(7P\)](#) 매뉴얼 페이지를 참조하십시오.

표 9-4 IPv6 주소 선택 정책 테이블

접두어	우선 순위	정의
::1/128	50	루프백
::/0	40	기본값
2002::/16	30	6to4
::/96	20	IPv4 호환 가능
::ffff:0:0/96	10	IPv4

이 표에서 IPv6 접두어(::1/128 및 ::/0)가 6to4 주소(2002::/16) 및 IPv4 주소(::/96 및 ::ffff:0:0/96)보다 우선적으로 사용됩니다. 따라서 기본적으로 커널은 다른 IPv6 대상으로 이동하는 패킷에 대해 전역 IPv6 주소의 인터페이스를 선택합니다. IPv4 주소의 인터페이스는 특히 IPv6 대상으로 이동하는 패킷에 대해 낮은 우선 순위를 갖습니다. 선택한 IPv6 소스 주소가 제공될 경우, 커널에서는 대상 주소에 대해 IPv6 형식도 사용됩니다.

## IPv6 주소 선택 정책 테이블을 수정하는 이유

대부분의 경우에는 IPv6 기본 주소 선택 정책 테이블을 변경할 필요가 없습니다. 정책 테이블을 관리해야 하는 경우 `ipaddrsel` 명령을 사용하십시오.

다음과 같은 경우에 정책 테이블을 수정할 수 있습니다.

- 시스템 인터페이스가 6to4 터널에 사용되는 경우, 6to4 주소에 더 높은 우선 순위를 제공할 수 있습니다.
- 특정 소스 주소를 특정 대상 주소와의 통신에만 사용하려는 경우, 이 주소를 정책 테이블에 추가하면 됩니다. 그런 다음 `ipadm`을 사용하여 이 주소를 선호 주소로 플래그 지정할 수 있습니다. `ipadm` 명령에 대한 자세한 내용은 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- IPv4 주소가 IPv6 주소보다 우선적으로 사용되게 하려는 경우, ::ffff:0:0/96의 우선 순위를 더 높은 숫자로 변경할 수 있습니다.
- 제거된 주소에 더 높은 우선 순위를 지정해야 하는 경우, 제거된 주소를 정책 테이블에 추가하면 됩니다. 예를 들어 사이트 로컬 주소는 이제 IPv6에서 제거되었습니다. 이러한 주소의 앞에는 `fec0::/10`이 붙습니다. 사이트 로컬 주소에 더 높은 우선 순위를 제공하도록 정책 테이블을 변경할 수 있습니다.

`ipaddrsel` 명령에 대한 자세한 내용은 [ipaddrsel\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 6to4relay 명령

6to4 터널링을 사용하면 분리된 6to4 사이트 간에 통신할 수 있습니다. 그러나 원시 비6to4 IPv6 사이트를 포함하는 패킷을 전송하려면 6to4 라우터가 6to4 릴레이 라우터를

사용하여 터널을 설정해야 합니다. 그러면 **6to4 릴레이 라우터**가 6to4 패킷을 IPv6 네트워크 및 원시 IPv6 사이트로 전송합니다. 6to4 지원 사이트가 원시 IPv6 사이트와 데이터를 교환해야 하는 경우 **6to4relay** 명령을 사용하여 해당 터널을 사용으로 설정하십시오.

릴레이 라우터 사용은 보안되지 않으므로 Oracle Solaris에서는 기본적으로 릴레이 라우터가 사용 안함으로 설정되어 있습니다. 이 시나리오를 배치하기 전에 6to4 릴레이 라우터에 대한 터널 생성과 관련된 문제를 주의 깊게 고려하십시오. 6to4 릴레이 라우터에 대한 자세한 내용은 [116 페이지 “6to4 릴레이 라우터에 대한 터널 고려 사항”](#)을 참조하십시오. 6to4 릴레이 라우터 지원을 사용하려는 경우 [120 페이지 “IP 터널을 만들고 구성하는 방법”](#)에서 관련 절차를 참조하십시오.

## 6to4relay 구문

6to4relay 명령의 구문은 다음과 같습니다.

```
6to4relay -e [-a IPv4-address] -d -h
```

- e                    6to4 라우터와 애니캐스트 6to4 릴레이 라우터 간 터널에 대한 지원을 사용으로 설정합니다. 그러면 터널 끝점 주소가 192.88.99.1(6to4 릴레이 라우터의 애니캐스트 그룹에 대한 기본 주소)로 설정됩니다.
- a IPv4-address    지정된 IPv4-address를 사용하여 6to4 라우터와 6to4 릴레이 라우터 간 터널에 대한 지원을 사용으로 설정합니다.
- d                    6to4 릴레이 라우터에 대한 터널링 지원을 사용 안함으로 설정합니다. 이는 Oracle Solaris의 기본값입니다.
- h                    6to4relay에 대한 도움말을 표시합니다.

자세한 내용은 6to4relay(1M) 매뉴얼 페이지를 참조하십시오.

### 예 9-2 6to4y 릴레이 라우터 지원의 기본 상태 표시

인수가 없는 6to4relay 명령은 6to4 릴레이 라우터 지원의 현재 상태를 표시합니다. 이 예는 IPv6의 Oracle Solaris 구현에 대한 기본값을 보여줍니다.

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is disabled
```

### 예 9-3 6to4 릴레이 라우터 지원을 사용으로 설정하여 상태 표시

릴레이 라우터 지원이 사용으로 설정된 경우, 6to4relay는 다음과 같은 출력을 표시합니다.

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is enabled
IPv4 destination address of Relay Router=192.88.99.1
```

예 9-4 6to4 릴레이 라우터를 지정하여 상태 표시

6to4relay 명령에 -a 옵션과 IPv4 주소를 지정한 경우, -a와 함께 제공한 IPv4 주소가 192.88.99.1 대신 표시됩니다.

6to4relay는 -d, -e 및 -a *IPv4 address* 옵션이 성공적으로 실행되면 이를 보고하지 않습니다. 그러나 이러한 옵션을 실행할 때 생성될 수 있는 오류 메시지는 6to4relay가 표시합니다.

## IPv6 지원을 위한 netstat 명령 수정 사항

netstat 명령이 IPv4 및 IPv6 네트워크 상태를 모두 표시합니다. /etc/default/inet\_type 파일에서 DEFAULT\_IP 값을 설정하거나 -f 명령줄 옵션을 사용하여 표시할 프로토콜 정보를 선택할 수 있습니다. DEFAULT\_IP를 영구적으로 설정하면 netstat가 IPv4 정보만 표시합니다. -f 옵션을 사용하여 이 설정을 대체할 수 있습니다. inet\_type 파일에 대한 자세한 내용은 [inet\\_type\(4\)](#) 매뉴얼 페이지를 참조하십시오.

netstat 명령의 -p 옵션은 net-to-media 테이블(IPv4의 경우 ARP 테이블이고, IPv6의 경우 이웃 캐시임)을 표시합니다. 자세한 내용은 [netstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 이 명령의 사용 절차에 대한 설명은 [94 페이지 “소켓 상태를 표시하는 방법”](#)을 참조하십시오.

## IPv6 지원을 위한 snoop 명령 수정 사항

snoop 명령이 IPv4 및 IPv6 패킷을 모두 캡처할 수 있습니다. 이 명령은 IPv6 헤더, IPv6 확장 헤더, ICMPv6 헤더 및 Neighbor Discovery 프로토콜 데이터를 표시할 수 있습니다. 기본적으로 snoop 명령은 IPv4 및 IPv6 패킷을 모두 표시합니다. ip 또는 ip6 프로토콜 키워드를 지정하면 snoop 명령은 IPv4 또는 IPv6 패킷만 표시합니다. IPv6 필터 옵션을 사용하여 IPv6 패킷만 표시하도록 모든 패킷(IPv4 및 IPv6)을 필터링할 수 있습니다. 자세한 내용은 [snoop\(1M\)](#) 매뉴얼 페이지를 참조하십시오. snoop 명령 사용 절차는 [105 페이지 “IPv6 네트워크 트래픽을 모니터링하는 방법”](#)을 참조하십시오.

## IPv6 지원을 위한 route 명령 수정 사항

route 명령이 IPv4 및 IPv6 경로 모두에서 작동합니다. 이때 기본값은 IPv4 경로입니다. 명령줄에서 route 명령 바로 뒤에 -inet6 옵션을 사용하면 작업이 IPv6 경로에서 수행됩니다. 자세한 내용은 [route\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## IPv6 지원을 위한 ping 명령 수정 사항

ping 명령이 IPv4 및 IPv6 프로토콜을 모두 사용하여 대상 호스트를 검사합니다. 이름 서버가 지정된 대상 호스트에 대해 반환하는 주소에 따라 프로토콜 선택이 달라집니다. 기본적으로 이름 서버가 대상 호스트에 대해 IPv6 주소를 반환하는 경우 ping 명령은

IPv6 프로토콜을 사용합니다. 이름 서버가 IPv4 주소만 반환하는 경우 ping 명령은 IPv4 프로토콜을 사용합니다. -A 명령줄 옵션을 사용하여 사용할 프로토콜을 지정하면 이 작업이 대체됩니다.

자세한 내용은 ping(1M) 매뉴얼 페이지를 참조하십시오. ping 사용 절차는 97 페이지 “ping 명령으로 원격 호스트 확인”을 참조하십시오.

## IPv6 지원을 위한 traceroute 명령 수정 사항

traceroute 명령을 사용하여 특정 호스트에 대해 IPv4 및 IPv6 경로를 추적할 수 있습니다. 프로토콜 관점에서 traceroute는 ping과 동일한 알고리즘을 사용합니다. 이 선택을 대체하려면 -A 명령줄 옵션을 사용하십시오. -a 명령줄 옵션을 사용하면 멀티홉 호스트의 각 주소에 대해 개별 경로를 추적할 수 있습니다.

자세한 내용은 traceroute(1M) 매뉴얼 페이지를 참조하십시오. traceroute 사용 절차는 101 페이지 “traceroute 명령으로 경로 지정 정보 표시”를 참조하십시오.

## IPv6 관련 데몬

이 절에서는 IPv6 관련 데몬에 대해 설명합니다.

### in.ndpd 데몬(Neighbor Discovery용)

in.ndpd 데몬은 IPv6 Neighbor Discovery 프로토콜 및 라우터 검색을 구현합니다. 이 데몬은 IPv6에 대한 주소 자동 구성도 구현합니다. 다음은 in.ndpd의 지원되는 옵션을 보여줍니다.

- d 디버깅을 설정합니다.
- D 특정 이벤트에 대한 디버깅을 설정합니다.
- f 기본 /etc/inet/ndpd.conf 파일 대신 구성 데이터를 읽도록 파일을 지정합니다.
- I 각 인터페이스에 대한 관련 정보를 출력합니다.
- n 라우터 알림을 루프백하지 않습니다.
- r 수신된 패킷을 무시합니다.
- v 다양한 유형의 진단 메시지를 보고하도록 상세 정보 표시 모드를 지정합니다.
- t 패킷 추적을 설정합니다.

in.ndpd 데몬은 /etc/inet/ndpd.conf 구성 파일에 설정된 매개변수와 /var/inet/ndpd\_state.interface 시작 파일의 매개변수로 제어됩니다.

/etc/inet/ndpd.conf 파일이 있으면 이 파일이 구문 분석되어 노드를 라우터로 구성하는 데 사용됩니다. 표 9-1은 이 파일에 나타날 수 있는 키워드를 보여줍니다. 호스트가

부트되는 즉시 라우터가 사용 가능하지 않을 수 있습니다. 라우터에 의해 알려진 패킷은 삭제될 수 있습니다. 또한 호스트에 연결되지 않을 수도 있습니다.

`/var/inet/ndpd_state.interface` 파일은 상태 파일입니다. 이 파일은 각 노드에서 정기적으로 업데이트됩니다. 노드가 실패하여 다시 시작되었을 때 라우터가 없는 경우 노드가 인터페이스를 구성할 수 있습니다. 이 파일에는 파일이 마지막으로 업데이트된 당시의 인터페이스 주소 및 파일 유효 기간이 포함되어 있습니다. 또한 이전 라우터 알림에서 “학습한” 기타 매개변수도 포함되어 있습니다.

---

**주** - 상태 파일의 내용은 변경할 필요가 없습니다. `in.ndpd` 데몬이 자동으로 상태 파일을 유지 관리합니다.

---

구성 변수 및 허용되는 값 목록은 `in.ndpd(1M)` 매뉴얼 페이지 및 `ndpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

## in.ripngd 데몬(IPv6 경로 지정용)

`in.ripngd` 데몬은 IPv6 라우터에 대한 차세대 경로 지정 정보 프로토콜(RIPng)을 구현합니다. RIPng는 IPv6에 해당하는 RIP입니다. `routeadm` 명령으로 IPv6 라우터를 구성하고 IPv6 경로 지정을 설정하면 `in.ripngd` 데몬이 라우터에서 RIPng를 구현합니다.

다음은 RIPng의 지원되는 옵션을 보여줍니다.

- p *n*    *n*은 RIPng 패킷을 전송 또는 수신하는 데 사용되는 대체 포트 번호를 지정합니다.
- q        경로 지정 정보를 표시하지 않습니다.
- s        데몬이 라우터로 사용되지 않는 경우에도 경로 지정 정보를 표시합니다.
- P        Poison Reverse를 사용하지 못하도록 합니다.
- S        `in.ripngd`가 라우터로 사용되지 않는 경우 데몬은 각 라우터의 기본 경로만 통과합니다.

## inetd 데몬 및 IPv6 서비스

IPv6 지원 서버 응용 프로그램은 IPv4 요청과 IPv6 요청을 모두 처리하거나, IPv6 요청만 처리할 수 있습니다. 서버는 항상 IPv6 소켓을 통해 요청을 처리합니다. 또한 해당 클라이언트가 사용하는 것과 동일한 프로토콜을 사용합니다.

IPv6용 서비스를 추가하거나 수정하려면 SMF(서비스 관리 기능)에서 제공하는 명령을 사용하십시오.

- SMF 명령에 대한 자세한 내용은 **Oracle Solaris 관리: 일반 작업의 “SMF 명령줄 관리 유틸리티”**를 참조하십시오.
- SMF를 사용하여 SCTP를 통해 실행되는 IPv4 서비스 매니페스트를 구성하는 예제 작업은 **70 페이지 “SCTP 프로토콜을 사용하는 서비스를 추가하는 방법”**을 참조하십시오.

IPv6 서비스를 구성하려면 해당 서비스에 대한 `inetadm` 프로파일의 `proto` 필드 값에 적합한 값이 나열되어야 합니다.

- IPv4 및 IPv6 요청을 모두 처리하는 서비스의 경우 `tcp6`, `udp6` 또는 `sctp`를 선택합니다. `tcp6`, `udp6` 또는 `sctp6`의 값이 `proto`일 경우 `inetd`는 서버에 IPv6 소켓을 전달합니다. IPv4 클라이언트에 요청이 있는 경우 서버에는 IPv4 매핑 주소가 포함됩니다.
- IPv6 요청만 처리하는 서비스의 경우 `tcp6only` 또는 `udp6only`를 선택합니다. `proto`에 대해 이러한 값 중 하나를 사용할 경우, `inetd`는 서버에 IPv6 소켓을 전달합니다.

Oracle Solaris 명령을 다른 구현으로 바꿀 경우 해당 서비스 구현이 IPv6을 지원하는지 확인해야 합니다. 구현이 IPv6을 지원하지 않는 경우 `proto` 값을 `tcp`, `udp` 또는 `sctp`로 지정해야 합니다.

다음은 IPv4 및 IPv6을 둘 다 지원하고 SCTP를 통해 실행되는 `echo` 서비스 매니페스트에 대해 `inetadm`이 실행되도록 하는 프로파일입니다.

```
# inetadm -l svc:/network/echo:sctp_stream
SCOPE      NAME=VALUE      name="echo"
            endpoint_type="stream"
            proto="sctp6"
            isrpc=FALSE
            wait=FALSE
            exec="/usr/lib/inet/in.echod -s"
            user="root"
default    bind_addr=""
default    bind_fail_max=-1
default    bind_fail_interval=-1
default    max_con_rate=-1
default    max_copies=-1
default    con_rate_offline=-1
default    failrate_cnt=40
default    failrate_interval=60
default    inherit_env=TRUE
default    tcp_trace=FALSE
default    tcp_wrappers=FALSE
```

`proto` 필드의 값을 변경하려면 다음 구문을 사용하십시오.

```
# inetadm -m FMRI proto="transport-protocols"
```

Oracle Solaris 소프트웨어에 제공되는 모든 서버에는 `proto`를 `tcp6`, `udp6` 또는 `sctp6`로 지정하는 프로파일 항목이 하나만 있으면 됩니다. 그러나 원격 셸 서버(`shell`) 및 원격 실행 서버(`exec`)는 이제 단일 서비스 인스턴스로 구성됩니다. 이 경우 `proto` 값에는 `tcp` 및 `tcp6only` 값이 포함됩니다. 예를 들어 `shell`의 `proto` 값을 설정하려면 다음 명령을 실행하십시오.

```
# inetadm -m network/shell:default proto="tcp,tcp6only"
```

소켓을 사용하는 IPv6 지원 서버를 작성하는 방법에 대한 자세한 내용은 **Programming Interfaces Guide**의 IPv6 extensions to the Socket API를 참조하십시오.

## IPv6용 서비스 구성 시 고려 사항

IPv6용 서비스를 추가하거나 수정할 경우 다음 사항에 유의하십시오.

- `proto` 값을 `tcp6`, `sctp6` 또는 `udp6`로 지정해야 IPv4 및 IPv6 연결이 모두 가능합니다. `proto` 값을 `tcp`, `sctp` 또는 `udp`로 지정한 경우 서비스는 IPv4만 사용합니다.
- `inetd`에 대해 일대다 스타일 SCTP 소켓을 사용하는 서비스 인스턴스를 추가할 수는 있지만, 이는 권장되지 않습니다. 일대다 스타일 SCTP 소켓에서는 `inetd`가 작동하지 않습니다.
- `wait-status` 또는 `exec` 등록 정보가 다르기 때문에 서비스에 두 개의 항목이 필요할 경우, 원래 서비스에서 두 개의 인스턴스/서비스를 만들어야 합니다.

# IPv6 Neighbor Discovery 프로토콜

IPv6은 RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (<http://www.ietf.org/rfc/rfc2461.txt?number=2461>)에 설명된 Neighbor Discovery 프로토콜을 사용합니다. 주요 Neighbor Discovery 기능에 대한 개요는 **System Administration Guide: IP Services**의 “IPv6 Neighbor Discovery Protocol Overview”를 참조하십시오.

이 절에서는 Neighbor Discovery 프로토콜의 다음 기능에 대해 설명합니다.

- 153 페이지 “Neighbor Discovery에서 제공하는 ICMP 메시지”
- 153 페이지 “자동 구성 프로세스”
- 155 페이지 “이웃 요청 및 연결 불가”
- 155 페이지 “중복 주소 감지 알고리즘”
- 156 페이지 “ARP 및 관련 IPv4 프로토콜과 Neighbor Discovery 비교”



## Neighbor Discovery에서 제공하는 ICMP 메시지

Neighbor Discovery에서는 5개의 새 ICMP(Internet Control Message Protocol) 메시지를 정의합니다. 이 메시지의 목적은 다음과 같습니다.

- **라우터 요청** - 인터페이스가 사용으로 설정되면 호스트에서 라우터 요청 메시지를 보낼 수 있습니다. 유도는 라우터 알림을 다음 예정 시간에 생성하는 대신 즉시 생성하도록 라우터에 요청합니다.
- **라우터 알림** - 라우터는 자신의 존재, 다양한 링크 매개변수 및 다양한 인터넷 매개변수를 알립니다. 라우터 알림은 정기적으로 수행되거나 라우터 요청 메시지에 대한 응답으로 수행됩니다. 라우터 알림에는 온-링크 결정 또는 주소 구성에 사용되는 접두어, 제안되는 홉 한계 값 등이 포함됩니다.
- **이웃 요청** - 노드가 이웃의 링크 계층 주소를 확인하기 위해 이웃 요청 메시지를 전송합니다. 이웃 요청 메시지는 캐시된 링크 계층 주소를 통해 여전히 이웃에 연결할 수 있는지 확인할 목적으로도 전송됩니다. 이웃 요청은 중복 주소 감지에도 사용됩니다.
- **이웃 알림** - 노드가 이웃 요청 메시지에 대한 응답으로 이웃 알림 메시지를 전송합니다. 또한 링크 계층 주소 변경을 알리기 위해 요청되지 않은 이웃 알림도 전송합니다.
- **재지정** - 라우터는 재지정 메시지를 사용하여 보다 나은 대상의 첫번째 홉을 호스트에 알리거나 대상이 동일한 링크에 있음을 알립니다.

## 자동 구성 프로세스

이 절에서는 자동 구성 중 인터페이스에서 수행하는 일반적인 단계에 대해 간략히 설명합니다. 자동 구성은 멀티캐스트 가능 링크에서만 수행됩니다.

1. 예를 들어 멀티캐스트 가능 인터페이스는 노드의 시스템 시작 중에 사용으로 설정됩니다.
2. 노드는 인터페이스에 대한 링크 로컬 주소를 생성하여 자동 구성 프로세스를 시작합니다.  
링크 로컬 주소는 인터페이스의 MAC(매체 액세스 제어) 주소에서 생성됩니다.
3. 노드가 임시 링크 로컬 주소를 대상으로 포함하는 이웃 요청 메시지를 전송합니다.  
이 메시지의 목적은 예상 주소를 링크의 다른 노드에서 아직 사용하고 있지 않음을 확인하는 것입니다. 확인 후 링크 로컬 주소를 인터페이스에 지정할 수 있습니다.
  - a. 다른 노드에서 이미 제안된 주소를 사용하고 있는 경우 주소가 이미 사용 중임을 나타내는 이웃 알림을 노드에서 반환합니다.
  - b. 다른 노드에서도 동일한 주소를 사용하려고 하는 경우 해당 노드에서도 대상에 대한 이웃 요청을 전송합니다.

이웃 요청 전송/재전송 횟수 및 연속 요청 간격은 링크별로 다릅니다. 필요한 경우 이러한 매개변수를 설정할 수 있습니다.

4. 노드에서 예상 링크 로컬 주소가 고유하지 않다고 판단될 경우 자동 구성이 중지됩니다. 이 경우 인터페이스의 링크 로컬 주소를 수동으로 구성해야 합니다.

간단하게 복구하려면 기본 식별자를 대체하는 대체 인터페이스 ID를 제공하면 됩니다. 그러면 고유한 새 인터페이스 ID를 사용하여 자동 구성 방식이 다시 시작될 수 있습니다.

5. 노드에서 예상 링크 로컬 주소가 고유하다고 판단될 경우 노드가 주소를 인터페이스에 지정합니다.

이 경우 노드가 이웃 노드와 IP 레벨로 연결됩니다. 나머지 자동 구성 단계는 호스트에 의해서만 수행됩니다.

## 라우터 알림 획득

자동 구성의 다음 단계는 라우터 알림을 확보하거나 라우터가 없음을 확인하는 것입니다. 라우터가 있을 경우 호스트에서 수행해야 하는 자동 구성의 유형을 지정하는 라우터 알림이 전송됩니다.

라우터는 라우터 알림을 정기적으로 전송합니다. 그러나 연속 알림 간격은 일반적으로 자동 구성을 수행하는 호스트의 대기 시간보다 깁니다. 알림을 신속하게 확보하기 위해 호스트는 하나 이상의 라우터 요청을 모든 라우터 멀티캐스트 그룹에 전송합니다.

## 접두어 구성 변수

라우터 알림에는 또한 Stateless 주소 자동 구성이 접두어를 생성하는 데 사용되는 정보를 포함하는 접두어 변수도 있습니다. 라우터 알림의 Stateless Address Autoconfiguration(Stateless 주소 자동 구성) 필드는 개별적으로 처리됩니다. 접두어 정보를 포함하는 한 옵션 필드 즉, Address Autoconfiguration(주소 자동 구성) 플래그는 옵션이 Stateless 자동 구성에도 적용되는지 여부를 나타냅니다. 이 옵션 필드가 적용되는 경우 추가 옵션 필드에 서브넷 접두어가 수명 값과 함께 포함됩니다. 이 값은 접두어로부터 생성된 주소가 선호 및 유효 주소로 유지되는 시간을 나타냅니다.

라우터에서는 라우터 알림을 정기적으로 생성하기 때문에 호스트는 계속 새로운 알림을 수신합니다. IPv6 지원 호스트는 각 알림에 포함된 정보를 처리합니다. 그런 다음 정보를 추가합니다. 호스트는 또한 이전 알림에서 수신된 정보를 새로 고칩니다.

## 주소 고유성

보안을 위해 모든 주소는 인터페이스에 지정되기 전에 고유한지 테스트해야 합니다. Stateless 자동 구성을 통해 생성되는 주소마다 상황이 다릅니다. 주소의 고유성은 주로 인터페이스 ID에서 구성되는 주소 부분에 의해 결정됩니다. 따라서 노드에서 이미 링크 로컬 주소의 고유성이 확인된 경우 추가 주소를 개별적으로 테스트할 필요가 없습니다. 주소는 동일한 인터페이스 ID에서 생성되어야 합니다. 반대로, 수동으로 확보된 모든

주소는 개별적으로 고유한지 테스트해야 합니다. 어떤 사이트의 시스템 관리자는 중복 주소 감지를 수행할 때 발생하는 오버헤드가 이점을 능가한다고 생각합니다. 이 사이트의 경우 인터페이스별 구성 플래그를 설정하여 중복 주소 감지 사용을 사용 안함으로 설정할 수 있습니다.

호스트가 라우터 알림을 기다리는 동안 링크 로컬 주소를 생성하고 고유성을 확인하면 자동 구성 프로세스를 신속하게 수행할 수 있습니다. 라우터는 라우터 요청에 대한 응답을 몇 초 동안 지연시킬 수 있습니다. 따라서 두 단계를 연속해서 수행할 경우 자동 구성을 완료하는 데 필요한 총 시간이 상당히 길어질 수 있습니다.

## 이웃 요청 및 연결 불가

Neighbor Discovery는 **이웃 요청** 메시지를 사용하여 둘 이상의 노드에 동일한 유니캐스트 주소가 지정되었는지 확인합니다. **이웃 연결 불가 감지**는 이웃 오류 또는 이웃에 대한 정방향 경로 오류를 찾아냅니다. 이 감지의 경우 이웃으로 전송된 패킷이 실제로 해당 이웃에 도달했다는 긍정적인 확인이 필요합니다. 이웃 연결 불가 감지는 또한 노드의 IP 계층에서 패킷이 올바르게 처리되고 있는지도 확인합니다.

이웃 연결 불가 감지는 상위 계층 프로토콜 및 이웃 요청 메시지라는 두 소스에서 보내는 확인을 사용합니다. 가능한 경우 상위 계층 프로토콜은 연결이 **진행 중**이라는 긍정적인 확인을 제공합니다. 예를 들어 새 TCP 긍정 응답이 수신될 경우 이전에 전송된 데이터가 올바르게 전달되었음이 확인됩니다.

노드가 상위 계층 프로토콜로부터 긍정적인 확인을 받지 못할 경우 유니캐스트 이웃 요청 메시지를 전송합니다. 이 메시지는 다음 홉에서 연결 가능성을 확인해 주는 이웃 알림을 요청합니다. 불필요한 네트워크 트래픽을 줄이려면 노드가 활발하게 패킷을 전송하는 이웃에게만 검사 메시지를 전송해야 합니다.

## 중복 주소 감지 알고리즘

구성된 모든 주소가 특정 링크에서 고유한지 확인하기 위해 노드는 주소에 대해 **중복 주소 감지** 알고리즘을 실행합니다. 주소를 인터페이스에 지정하기 전에 노드에서 이 알고리즘을 실행해야 합니다. 중복 주소 감지 알고리즘은 모든 주소에 대해 수행됩니다.

이 절에 설명된 자동 구성 프로세스는 라우터가 아닌 호스트에만 적용됩니다. 호스트 자동 구성에는 라우터가 알리는 정보가 사용되므로 라우터를 다른 방식으로 구성해야 합니다. 그러나 라우터는 이 장에 설명된 방식을 사용하여 링크 로컬 주소를 생성합니다. 또한 라우터는 주소를 인터페이스에 지정하기 전에 모든 주소에 대한 중복 주소 감지 알고리즘을 성공적으로 전달합니다.

## 프록시 알림

대상 주소 대신 패킷을 수락하는 라우터는 비대체 이웃 알림을 발행할 수 있습니다. 라우터는 이웃 요청에 응답할 수 없는 대상 주소에 대한 패킷을 수락할 수 있습니다. 현재는 프록시 사용이 지정되어 있지 않습니다. 그러나 프록시 알림을 사용하면 오프 링크가 이동된 모바일 노드와 같은 경우를 잠재적으로 처리할 수 있습니다. 프록시 사용은 이 프로토콜을 구현하는 노드를 처리하는 일반적인 방식은 아닙니다.

## 인바운드 로드 균형 조정

복제된 인터페이스를 포함하는 노드의 경우 동일한 링크의 여러 네트워크 인터페이스에서 패킷 수신 로드 균형에 대한 균형을 조정해야 합니다. 이러한 노드에서는 여러 개의 링크 로컬 주소가 동일한 인터페이스에 지정되어 있습니다. 예를 들어 한 개의 네트워크 드라이버가 여러 네트워크 인터페이스를 링크 로컬 주소가 여러 개인 하나의 논리적 인터페이스로 표시할 수 있습니다.

로드 균형 조정은 라우터가 소스 링크 로컬 주소를 라우터 알림 패킷에서 생략하는 방식으로 처리됩니다. 따라서 이웃은 이웃 요청 메시지를 사용하여 라우터의 링크 로컬 주소를 알아내야 합니다. 그러면 요청을 발행한 주체에 따라 달라지는 링크 로컬 주소가 반환된 이웃 알림 메시지에 포함될 수 있습니다.

## 링크 로컬 주소 변경

링크 로컬 주소가 변경되었음을 알고 있는 노드는 요청되지 않은 멀티캐스트 이웃 알림 패킷을 전송할 수 있습니다. 이 노드의 경우 멀티캐스트 패킷을 모든 노드에 전송하여 잘못된 캐시된 링크 로컬 주소를 업데이트할 수 있습니다. 요청되지 않은 알림은 성능 향상을 위한 목적으로만 전송됩니다. 이웃 연결 불가 감지 알고리즘은 지연이 다소 길어지더라도 모든 노드가 새로운 주소를 안정적으로 검색할 수 있도록 해줍니다.

## ARP 및 관련 IPv4 프로토콜과 Neighbor Discovery 비교

IPv6 Neighbor Discovery 프로토콜의 기능은 IPv4 프로토콜의 ARP(Address Resolution Protocol), ICMP(Internet Control Message Protocol) 라우터 검색 및 ICMP 재지정을 결합한 것입니다. IPv4에는 이웃 연결 불가 감지에 대해 일반적으로 합의된 프로토콜이나 방식이 없습니다. 그러나 호스트 요구 사항에 사용 불능 게이트웨이 감지에 대한 알고리즘이 지정되어 있습니다. 사용 불능 게이트웨이 감지는 이웃 연결 불가 감지를 통해 해결되는 문제의 일부입니다.

다음은 Neighbor Discovery 프로토콜을 관련 IPv4 프로토콜 세트와 비교한 목록입니다.

- 라우터 검색은 기본 IPv6 프로토콜 세트의 일부입니다. IPv6 호스트의 경우 라우터를 찾기 위해 경로 지정 프로토콜을 snoop할 필요가 없습니다. IPv4의 경우 라우터를 찾기 위해 ARP, ICMP 라우터 검색 및 ICMP 재지정을 사용합니다.
- IPv6 라우터 알림은 링크 로컬 주소를 전달합니다. 라우터의 링크 로컬 주소를 분석하기 위해 추가 패킷 교환이 필요하지 않습니다.
- 라우터 알림은 링크에 대한 사이트 접두어를 전달합니다. IPv4의 경우와 마찬가지로, 넷마스크를 구성하기 위해 별도의 방식이 필요하지 않습니다.
- 라우터 알림을 통해 주소 자동 구성이 가능해집니다. IPv4에서는 자동 구성이 구현되지 않았습니다.
- Neighbor Discovery를 사용하면 IPv6 라우터가 링크에 사용할 호스트의 MTU를 알릴 수 있습니다. 따라서 잘 알려진 MTU가 없는 링크에 대해 동일한 MTU 값이 모든 노드에서 사용됩니다. 동일한 네트워크에 있는 IPv4 호스트는 다른 MTU를 사용할 수 있습니다.
- IPv4 브로드캐스트 주소와 달리, IPv6 주소 결정 멀티캐스트는 40억( $2^{32}$ )개 이상의 멀티캐스트 주소에 분산되어 있으므로 대상이 아닌 노드에서 주소 결정 관련 인터럽트가 상당히 줄어듭니다. 또한 비IPv6 시스템의 경우 전혀 인터럽트가 발생하지 않습니다.
- IPv6 재지정에는 첫번째 새 홉의 링크 로컬 주소가 포함되어 있습니다. 재지정 수신 시 별도의 주소 결정이 필요하지 않습니다.
- 여러 사이트 접두어가 동일한 IPv6 네트워크와 연관될 수 있습니다. 기본적으로 호스트는 라우터 알림을 통해 모든 로컬 사이트 접두어를 알게 됩니다. 그러나 라우터 알림에서 일부 또는 전체 접두어를 생략하도록 라우터를 구성할 수 있습니다. 이 경우 호스트는 대상이 원격 네트워크에 있다고 가정합니다. 따라서 호스트는 트래픽을 라우터로 전송합니다. 그러면 라우터가 재지정을 적절하게 발행할 수 있습니다.
- IPv4와 달리, IPv6 재지정 메시지의 수신자는 새로운 다음 홉이 로컬 네트워크에 있다고 가정합니다. IPv4에서는 네트워크 마스크에 따라 로컬 네트워크에 있지 않은 다음 홉을 지정하는 재지정 메시지가 호스트에서 무시됩니다. IPv6 재지정 방식은 IPv4의 XRedirect 기능과 비슷합니다. 재지정 방식은 비브로드캐스트 및 공유 매체 링크에 유용합니다. 이러한 네트워크에서 노드는 로컬 링크 대상에 대한 모든 접두어를 검사하면 안 됩니다.
- IPv6 이웃 연결 불가 감지는 라우터에서 오류가 발생할 경우 패킷 전달을 향상시켜 줍니다. 이 기능은 부분적으로 오류가 발생한 링크나 분할된 링크를 통한 패킷 전달을 향상시켜 줍니다. 또한 링크 로컬 주소가 변경된 노드를 통한 패킷 전달도 향상시켜 줍니다. 예를 들어 모바일 노드는 사용되지 않는 ARP 캐시 덕분에 연결을 유지한 상태로 로컬 네트워크에서 이동할 수 있습니다. IPv4에는 이웃 연결 불가 감지에 해당하는 방식이 없습니다.
- ARP와 달리, Neighbor Discovery는 이웃 연결 불가 감지를 통해 반 링크 오류를 감지합니다. Neighbor Discovery는 양방향 연결이 없을 경우 트래픽이 이웃에게 전송되지 못하도록 합니다.

- IPv6 호스트는 라우터를 고유하게 식별하는 링크 로컬 주소를 사용하여 라우터 연관을 유지할 수 있습니다. 라우터를 식별하는 기능은 라우터 알림 및 재지정 메시지에 필요합니다. 사이트에 새 전역 접두어가 사용될 경우 호스트에서 라우터 연관이 유지되어야 합니다. IPv4에는 라우터를 식별하는 해당 방식이 없습니다.
- 수신 시 Neighbor Discovery 메시지의 홑 한계는 255이기 때문에 프로토콜은 오프 링크 노드에서 발생하는 스푸핑 공격의 영향을 받지 않습니다. 반대로, IPv4 오프 링크 노드의 경우 ICMP 재지정 메시지를 전송할 수 있습니다. IPv4 오프 링크 노드의 경우 또한 라우터 알림 메시지도 전송할 수 있습니다.
- ICMP 계층에 주소 결정을 배치하면 Neighbor Discovery는 ARP보다 더 매체 독립적입니다. 따라서 표준 IP 인증 및 보안 방식을 사용할 수 있습니다.

## IPv6 경로 지정

ICIDR(Classless Inter-Domain Routing)에 의거하여 Pv6 경로 지정은 IPv4 경로 지정과 거의 동일합니다. 주소가 32비트 IPv4 주소 대신 128비트 IPv6 주소라는 점만 다릅니다. 매우 간단한 확장을 통해 IPv4의 모든 경로 지정 알고리즘(예: OSPF, RIP, IDRP, IS-IS)을 IPv6의 경로를 지정하는 데 사용할 수 있습니다.

IPv6에는 또한 강력한 새로운 경로 지정 기능을 지원하는 단순 경로 지정 확장도 포함되어 있습니다. 새로운 경로 지정 기능은 다음과 같습니다.

- 정책, 성능 및 비용 등을 기준으로 하는 공급자 선택
- 호스트 이동성, 현재 위치로 경로 지정
- 자동 주소 재지정, 새 주소로 경로 지정

새로운 경로 지정 기능은 IPv6 경로 지정 옵션을 사용하는 IPv6 주소의 순서를 만들어 이용할 수 있습니다. IPv6 소스는 경로 지정 옵션을 사용하여 패킷 대상으로 이동하는 중에 방문할 하나 이상의 중간 노드 또는 토폴로지 그룹을 나열할 수 있습니다. 이 기능은 IPv4의 느슨한 소스 및 레코드 경로 옵션과 매우 비슷합니다.

주소 순서를 일반 기능으로 만들려면 대부분의 경우 IPv6 호스트에서 호스트가 수신하는 패킷의 경로를 역순으로 설정해야 합니다. IPv6 인증 헤더를 사용하여 패킷이 성공적으로 인증되어야 합니다. 패킷에 주소 순서가 포함되어 있어야 패킷이 원래 전송자에게 반환됩니다. 이 기술은 IPv6 호스트 구현에서 소스 경로의 처리 및 전환이 강제로 지원되도록 합니다. 소스 경로의 처리 및 전환은 공급자가 새로운 IPv6 기능(예: 공급자 선택 및 확장 주소)을 구현하는 호스트와 작업할 수 있도록 하는 데 중요합니다.

## 라우터 알림

멀티캐스트 가능 링크 및 지점 간 링크에서 각 라우터는 라우터의 사용 가능성을 알리는 라우터 알림 패킷을 정기적으로 멀티캐스트 그룹에 전송합니다. 호스트는 모든 라우터로부터 라우터 알림을 수신하여 기본 라우터 목록을 작성합니다. 라우터는 몇 초



내에 호스트가 라우터의 존재를 알 수 있도록 자주 라우터 알림을 생성합니다. 그러나 라우터는 알림 부재를 통해 라우터 오류를 감지할 만큼 자주 알림을 전송하지 않습니다. 이웃 연결 불가를 확인하는 별도의 감지 알고리즘을 통해 오류를 감지할 수 있습니다.

## 라우터 알림 접두어

라우터 알림에는 호스트가 라우터와 동일한 링크(온 링크)에 있는지 확인하는 데 사용되는 서브넷 접두어 목록이 포함되어 있습니다. 접두어 목록은 자동 주소 구성에도 사용됩니다. 접두어와 연관된 플래그는 특정 접두어의 의도된 사용을 지정합니다. 호스트는 알림의 온 링크 접두어를 사용하여 패킷 대상이 온 링크인 시점 또는 라우터 외부에 있는 시점을 확인하는 데 사용되는 목록을 작성하고 유지 관리합니다. 대상이 알림의 온 링크 접두어에 의해 처리되지 않더라도 대상은 온 링크 상태일 수 있습니다. 이 경우 라우터가 재지정을 전송할 수 있습니다. 재지정은 대상이 이웃임을 발신자에게 알립니다.

라우터는 라우터 알림 및 접두어별 플래그를 사용하여 Stateless 주소 자동 구성을 수행하는 방법을 호스트에 알릴 수 있습니다.

## 라우터 알림 메시지

라우터 알림 메시지에는 호스트가 송신 패킷에 사용해야 하는 인터넷 매개변수(예: 홑한계)가 포함되어 있습니다. 선택적으로 링크 매개변수(예: 링크 MTU)도 포함될 수 있습니다. 이 기능으로 중요한 매개변수를 중앙에서 관리할 수 있습니다. 매개변수는 라우터에 대해 설정될 수 있으며 연결된 모든 호스트에 자동으로 전파됩니다.

노드는 대상 노드에 해당 링크 계층 주소를 반환하도록 요청하는 이웃 요청을 멀티캐스트 그룹에 전송하는 방식으로 주소 결정을 수행합니다. 멀티캐스트 이웃 요청 메시지는 대상 주소의 요청된 노드 멀티캐스트 주소로 전송됩니다. 대상은 유니캐스트 이웃 알림 메시지에 링크 계층 주소를 반환합니다. 패킷의 단일 요청-응답 쌍만으로 개시자와 대상이 서로의 링크 계층 주소를 결정할 수 있습니다. 이웃 요청에는 개시자의 링크 계층 주소가 포함되어 있습니다.

# Oracle Solaris 이름 서비스에 대한 IPv6 확장

이 절에서는 IPv6 구현으로 도입된 이름 지정 변경 사항에 대해 설명합니다. IPv6 주소는 Oracle Solaris 이름 지정 서비스, NIS, LDAP, DNS 및 파일에 저장할 수 있습니다. IPv6 RPC 전송을 통해 NIS를 사용하여 원하는 NIS 데이터를 검색할 수도 있습니다.

## IPv6에 대한 DNS 확장

IPv6 관련 리소스 레코드인 AAAA 리소스 레코드는 RFC 1886 IP 버전 6 지원을 위한 DNS 확장에 지정되었습니다. 이 AAAA 레코드는 호스트 이름을 128비트 IPv6 주소에

매핑합니다. PTR 레코드는 여전히 IPv6에서 IP 주소를 호스트 이름에 매핑하는 데 사용됩니다. 128비트 주소의 32 x 4 비트 니블은 IPv6 주소에 대해 역순 처리됩니다. 각 니블은 해당 16진 ASCII 값으로 변환됩니다. 그런 다음 ip6.int가 추가됩니다.

## 이름 서비스 명령에 대한 변경 사항

IPv6을 지원하기 위해 기존 이름 서비스 명령을 사용하여 IPv6 주소를 조회할 수 있습니다. 예를 들어 `ypmatch` 명령은 새 NIS 맵에서 작동합니다. `nslookup` 명령은 DNS에서 새 AAAA 레코드를 조회할 수 있습니다.

## NFS 및 RPC IPv6 지원

NFS 소프트웨어 및 원격 프로시저 호출(RPC) 소프트웨어는 일관된 방식으로 IPv6을 지원합니다. NFS 서비스와 관련된 기존 명령은 변경되지 않았습니다. 대부분의 RPC 응용 프로그램도 별다른 변경 없이 IPv6에서 실행될 수 있습니다. 전송 정보를 포함하는 일부 고급 RPC 응용 프로그램의 경우 업데이트가 필요할 수 있습니다.

## IPv6 Over ATM 지원

Oracle Solaris는 IPv6 over ATM, 영구 가상 회선(PVC) 및 정적 전환 가상 회선(SVC)을 지원합니다.



## 제 2 부

# DHCP

이 부분은 DHCP(Dynamic Host Configuration Protocol)에 대한 개념적 정보를 포함하고 DHCP 서비스를 계획, 구성, 관리하고 문제를 해결하기 위한 작업을 설명합니다.



## DHCP 정보(개요)

---

이 장에서는 DHCP(Dynamic Host Configuration Protocol)를 소개하고 프로토콜의 근간을 이루는 개념을 설명합니다. 또한 네트워크에서 DHCP 사용 시의 이점을 설명합니다.

이 장은 다음 정보를 포함합니다.

- 163 페이지 “DHCP 프로토콜 정보”
- 164 페이지 “DHCP 사용 시의 이점”
- 165 페이지 “DHCP의 작동 방식”
- 168 페이지 “ISC DHCP 서버”
- 169 페이지 “DHCP 클라이언트”

## DHCP 프로토콜 정보

DHCP 프로토콜을 사용하여 TCP/IP 네트워크의 호스트 시스템을 부트할 때 네트워크에 대해 자동으로 구성할 수 있습니다. DHCP는 클라이언트-서버 방식을 사용합니다. 서버는 클라이언트에 대한 구성 정보를 저장 및 관리하고, 클라이언트 요청 시 해당 정보를 제공합니다. 이 정보에는 클라이언트의 IP 주소와 클라이언트에 사용 가능한 네트워크 서비스 정보가 포함됩니다.

DHCP는 이전 프로토콜인 BOOTP(TCP/IP 네트워크를 통해 부트하도록 설계)에서 발전한 것입니다. 클라이언트와 서버 간의 메시지에 대해 DHCP는 BOOTP와 동일한 형식을 사용합니다. 그러나 BOOTP 메시지와 달리, DHCP 메시지는 클라이언트에 대한 네트워크 구성 데이터를 포함할 수 있습니다.

DHCP의 주요 장점은 임대를 통해 IP 주소 지정을 관리할 수 있다는 것입니다. **임대**를 사용하면 IP 주소가 사용 중이 아닐 때 재생 이용할 수 있습니다. 재생 이용된 IP 주소는 다른 클라이언트에 재지정할 수 있습니다. DHCP를 사용하는 사이트는 모든 클라이언트에 영구 IP 주소를 지정했을 때 필요한 것보다 작은 IP 주소 풀을 사용할 수 있습니다.

## DHCP 사용 시의 이점

DHCP는 시간이 오래 걸리는 TCP/IP 네트워크 설정 작업이나 일상적인 네트워크 관리 작업을 줄일 수 있습니다. Oracle Solaris 구현에서 DHCP는 IPv4에만 작동합니다.

DHCP는 다음과 같은 이점을 제공합니다.

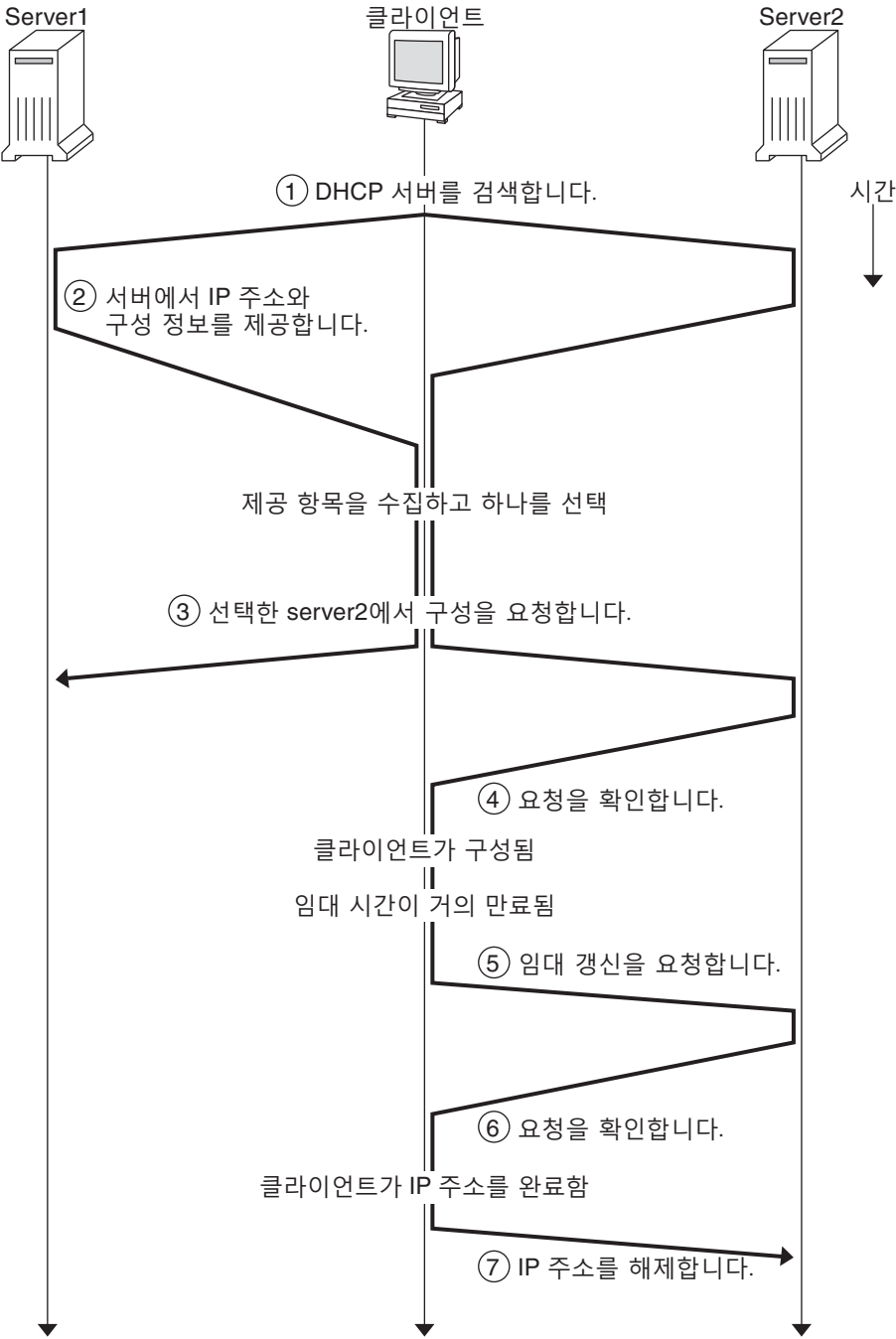
- **IP 주소 관리** - DHCP의 주요 이점은 간편한 IP 주소 관리입니다. DHCP가 없는 네트워크에서는 IP 주소를 수동으로 지정해야 합니다. 매우 신중하게 각 클라이언트에 고유한 IP 주소를 지정하고 각 클라이언트를 개별적으로 구성해야 합니다. 클라이언트가 다른 네트워크로 이동하면 해당 클라이언트를 수동으로 수정해야 합니다. DHCP가 사용으로 설정된 경우 관리자 개입 없이 DHCP 서버가 IP 주소를 관리하고 지정합니다. DHCP 서버로부터 새 네트워크에 적절한 새 클라이언트 정보를 얻으므로 수동 재구성 없이 다른 서브넷으로 클라이언트를 이동할 수 있습니다.
- **중앙화된 네트워크 클라이언트 구성** - 특정 클라이언트 또는 특정 클라이언트 유형에 대해 맞춤형 구성을 만들 수 있습니다. 구성 정보는 DHCP 데이터 저장소의 한 곳에 저장됩니다. 구성을 변경하기 위해 클라이언트에 로그인할 필요가 없습니다. 간단히 데이터 저장소의 정보를 변경하면 여러 클라이언트를 변경할 수 있습니다.
- **BOOTP 클라이언트 지원** - BOOTP 서버와 DHCP 서버는 모두 클라이언트에서 브로드캐스트를 수신하고 응답합니다. DHCP 서버는 DHCP 클라이언트는 물론 BOOTP 클라이언트의 요청에 응답할 수 있습니다. BOOTP 클라이언트는 IP 주소 및 서버에서 부트하는 데 필요한 정보를 수신합니다.
- **로컬 클라이언트 및 원격 클라이언트 지원** - BOOTP는 한 네트워크에서 다른 네트워크로 메시지 중계를 제공합니다. DHCP는 여러 가지 방법으로 BOOTP 중계 기능을 활용합니다. 대부분의 네트워크 라우터는 BOOTP 중계 에이전트로 작동하여 클라이언트 네트워크에 없는 서버로 BOOTP 요청을 전달하도록 구성할 수 있습니다. DHCP 요청은 BOOTP 요청과 구별하기 어렵기 때문에 DHCP 요청을 동일한 방법으로 라우터에 중계할 수 있습니다. 또한 BOOTP 중계를 지원하는 라우터를 사용할 수 없는 경우 DHCP 서버가 BOOTP 중계 에이전트로 작동하도록 구성할 수 있습니다.
- **네트워크 부트** - 클라이언트는 RARP(Reverse Address Resolution Protocol) 및 bootparams 파일을 사용하는 대신, DHCP를 사용하여 네트워크의 서버에서 부트하는 데 필요한 정보를 얻을 수 있습니다. DHCP 서버는 IP 주소, 부트 서버, 네트워크 구성 정보 등 클라이언트가 작동하는 데 필요한 모든 정보를 제공할 수 있습니다. DHCP 요청을 서브넷에서 중계할 수 있으므로 DHCP 네트워크 부트를 사용할 때 네트워크에서 훨씬 적은 부트 서버를 배치할 수 있습니다. RARP로 부트하려면 각 서브넷에 부트 서버가 필요합니다.
- **대형 네트워크 지원** - 수백만 개의 DHCP 클라이언트가 포함된 네트워크에서 DHCP를 사용할 수 있습니다. DHCP 서버는 멀티스레딩을 사용하여 많은 클라이언트 요청을 동시에 처리합니다. 또한 대량의 데이터 처리를 위해 최적화된 데이터 저장소를 지원합니다. 데이터 저장소 액세스는 별도의 프로세싱 모듈로 처리됩니다. 이 데이터 저장소 접근법을 통해 필요한 데이터베이스에 대한 지원을 추가할 수 있습니다.

## DHCP의 작동 방식

먼저 DHCP 서버를 설치하고 구성해야 합니다. 구성 중 클라이언트가 네트워크에서 작동하는 데 필요한 네트워크 정보를 지정합니다. 이 정보가 갖춰진 후에 클라이언트가 네트워크 정보를 요청 및 수신할 수 있습니다.

다음 다이어그램에 DHCP 서비스의 이벤트 순서가 표시됩니다. 원 안의 숫자는 다이어그램에 이어진 설명에서 번호 매기기 항목에 해당합니다.

그림 10-1 DHCP 서비스의 이벤트 순서



앞의 다이어그램은 다음 단계를 보여줍니다.

1. 클라이언트가 로컬 서브넷의 제한된 브로드캐스트 주소(255.255.255.255)로 **Discover 메시지**를 브로드캐스트하여 DHCP 서버를 검색합니다. 라우터가 존재하고 BOOTP 중계 에이전트로 작동하도록 구성된 경우 여러 서브넷의 다른 DHCP 서버로 요청이 전달됩니다. 클라이언트의 **브로드캐스트**에는 Oracle Solaris의 DHCP 구현에서 클라이언트의 MAC(Media Access Control) 주소로부터 파생된 고유한 ID가 포함됩니다. 이더넷 네트워크에서 MAC 주소는 이더넷 주소와 동일합니다.

Discover 메시지를 받은 DHCP 서버는 다음 정보를 확인하여 클라이언트의 네트워크를 결정할 수 있습니다.

- 어떤 네트워크 인터페이스에서 요청이 들어왔습니까? 서버는 클라이언트가 인터페이스로 연결된 네트워크에 있는지, 또는 클라이언트가 해당 네트워크에 연결된 BOOTP 중계 에이전트를 사용 중인지 확인합니다.
  - 요청에 BOOTP 중계 에이전트의 IP 주소가 들어 있습니까? 요청이 중계 에이전트를 통해 전달된 경우 요청 헤더에 중계 에이전트의 주소가 삽입됩니다. 서버가 **중계 에이전트 주소**를 감지한 경우 중계 에이전트가 클라이언트의 네트워크에 연결되어야 하므로 주소의 네트워크 부분이 클라이언트의 네트워크 주소를 나타냅니다.
  - 클라이언트의 네트워크가 서브넷으로 나뉘습니까? 서버가 **netmasks** 테이블을 참조하여 중계 에이전트의 주소 또는 요청을 받은 네트워크 인터페이스의 주소가 가리키는 네트워크에서 사용된 서브넷 마스크를 찾습니다. 일단 서버가 사용된 서브넷 마스크를 알고 나면 네트워크 주소의 어떤 부분이 호스트 부분인지 결정하고, 클라이언트에 적절한 IP 주소를 선택할 수 있습니다. **netmasks**에 대한 자세한 내용은 **netmasks(4)** 매뉴얼 페이지를 참조하십시오.
2. DHCP 서버가 클라이언트의 네트워크를 결정한 후에 적절한 IP 주소를 선택하고 주소가 아직 사용 중이 아닌지 확인합니다. 그런 다음 DHCP 서버가 **Offer 메시지**를 브로드캐스트하여 클라이언트에 응답합니다. Offer 메시지에는 선택된 IP 주소와 클라이언트에 구성할 수 있는 서비스 정보가 포함됩니다. 각 서버는 클라이언트가 IP 주소의 사용 여부를 결정할 때까지 제공된 IP 주소를 임시로 예약합니다.
  3. 클라이언트가 제공된 서비스 개수와 유형을 기반으로 최상의 제안을 선택합니다. 클라이언트가 최상의 제안을 제출한 서버의 IP 주소를 가리키는 요청을 브로드캐스트합니다. 브로드캐스트는 모든 응답 DHCP 서버가 클라이언트가 서버를 선택했음을 알고 있다고 보장합니다. 선택되지 않은 서버는 제공받은 IP 주소의 예약을 취소할 수 있습니다.
  4. 선택된 서버가 클라이언트에 대한 IP 주소를 할당하고 DHCP 데이터 저장소에 정보를 저장합니다. 또한 클라이언트에 확인 메시지(ACK)를 보냅니다. **확인 메시지**는 클라이언트에 대한 네트워크 구성 매개변수를 포함합니다. 클라이언트가 ping 유틸리티를 사용하여 다른 시스템에서 IP 주소를 사용 중이 아닌지 테스트합니다. 그런 다음 클라이언트가 부트를 계속하여 네트워크에 참여합니다.
  5. 클라이언트가 임대 시간을 모니터합니다. 정해진 기간이 경과된 경우 클라이언트가 선택한 서버에 임대 시간을 늘리라는 새 메시지를 보냅니다.

6. 요청을 받은 DHCP 서버는 관리자가 설정한 로컬 임대 정책을 고수하는 경우 임대 시간을 연장합니다. 서버가 20초 안에 응답하지 않으면 클라이언트가 요청을 브로드캐스트하여 다른 DHCP 서버 중 하나가 임대를 연장할 수 있도록 합니다.
7. 클라이언트에 더 이상 IP 주소가 필요하지 않으면 IP 주소가 해제되었음을 서버에 알립니다. 이 통지는 정상적인 종료 중에 발생할 수 있으며 수동으로 실행할 수도 있습니다.

## ISC DHCP 서버

ISC(Internet Systems Consortium) DHCP 서버의 구현이 Oracle Solaris에 추가되었습니다. 이 소프트웨어는 자동으로 설치되지 않으므로 다음 명령을 입력하여 이 서버를 시스템에 추가할 수 있습니다.

```
# pkg install pkg:/service/network/dhcp/isc-dhcp
```

ISC DHCP 서버 dhcpd는 DHCP(Dynamic Host Configuration Protocol) 및 BOOTP(Internet Bootstrap Protocol)를 구현합니다. DHCP를 사용하여 TCP/IP 네트워크의 호스트에서 IP 주소를 요청 및 지정할 수 있고, 연결된 네트워크에 대한 정보를 검색할 수도 있습니다. BOOTP는 비슷한 기능을 제공합니다.

다음은 DHCP의 임대에 관한 중요한 추가 정보를 나열한 것입니다.

- 여러 서비스가 ISC DHCP 및 레거시 Sun DHCP 서비스를 지원하도록 추가되었습니다. DHCP에서 사용된 모든 서비스의 목록은 [196 페이지 “DHCP 서비스에서 사용된 SMF 서비스”](#)를 참조하십시오.
- 세 가지 명령 dhcpd, dhcprelay, omshell이 추가되었습니다. DHCP와 연관된 모든 명령의 목록은 [194 페이지 “DHCP 서비스에서 사용된 파일”](#)을 참조하십시오.
- ISC DHCP의 서버 구성 파일은 /etc/inet/dhcpd4.conf(DHCPv4의 경우) 및 /etc/inet/dhcpd6.conf(DHCPv6의 경우)입니다.
- dhcpserv라는 사용자가 ISC DHCP 서비스를 위해 추가되었습니다.
- 세 가지 신규 명령에 대한 액세스는 solaris.smf.manage.dhcp 및 solaris.smf.value.dhcp 권한 부여를 사용하여 관리할 수 있습니다.

ISC DHCP에 대한 자세한 내용은 [ISC DHCP Documentation](#) 웹 페이지를 참조하십시오.

## 레거시 Sun DHCP 서버

레거시 Sun DHCP 서버 소프트웨어가 Oracle Solaris 11 릴리스에 계속 포함되지만, 더 이상 사용되지 않는 것으로 표시되었고 추후 릴리스에서 제거될 예정입니다. 레거시 DHCP 서비스에 대한 자세한 내용은 [Chapter 11, ISC DHCP 서비스 관리](#)를 참조하십시오.



## DHCP 클라이언트

“클라이언트”라는 용어는 때때로 네트워크에서 클라이언트 역할을 수행하는 물리적 시스템을 지칭합니다. 그러나 이 문서에 설명된 DHCP 클라이언트는 소프트웨어 엔티티입니다. DHCP 클라이언트는 시스템의 Oracle Solaris에서 실행되는 데몬(dhcpagent)으로, DHCP 서버에서 네트워크 구성을 수신하도록 구성됩니다. DHCP 클라이언트는 레거시 Sun DHCP 서버 및 ISC DHCP 서버 모두와 상호 운영할 수 있습니다.

DHCP 클라이언트에 대한 자세한 내용은 [12 장, “DHCP 클라이언트 구성 및 관리”](#)를 참조하십시오.



## ISC DHCP 서비스 관리

---

이 장에서는 ISC DHCP 서비스를 관리할 때 유용한 작업을 설명합니다. 다음 항목을 다룹니다.

- 171 페이지 “DHCP 명령에 사용자 액세스 설정”
- 172 페이지 “DHCP 서버 작업”

### DHCP 명령에 사용자 액세스 설정

기본적으로 root 사용자만 svcadm 및 기타 DHCP 서비스 구성에 필요한 명령을 실행할 수 있습니다. root가 아닌 사용자가 명령을 사용하려면 이러한 명령에 대해 RBAC(역할 기반 액세스 제어)를 설정할 수 있습니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)를 참조하십시오.

`rbac(5)`, `exec_attr(4)`, `user_attr(4)` 등의 매뉴얼 페이지도 유용합니다.

다음 절차는 사용자가 DHCP 명령을 실행할 수 있도록 DHCP Management 프로파일을 지정하는 방법을 설명합니다.

#### ▼ DHCP 명령에 사용자 액세스를 부여하는 방법

- 1 슈퍼유저가 되거나 DHCP Management 프로파일에 할당된 역할이나 사용자 이름을 말합니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)를 참조하십시오.

## 2 /etc/user\_attr 파일에 사용자나 역할을 추가합니다.

/etc/user\_attr 파일을 편집하여 다음 형태의 항목을 추가합니다. DHCP 서비스를 관리할 사용자나 역할마다 하나씩 항목을 추가합니다.

```
username::::type=normal;profiles=DHCP Management
```

예를 들어, 사용자 ram에 대해 다음 항목을 추가합니다.

```
ram::::type=normal;profiles=DHCP Management
```

# DHCP 서버 작업

## ▼ ISC DHCP 서버를 구성하는 방법

이러한 단계를 사용하여 초기에 ISC DHCP 서버를 구성할 수 있습니다.

### 1 슈퍼유저가 되거나 DHCP Management 프로파일에 할당된 역할이나 사용자 이름을 말합니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)를 참조하십시오.

### 2 DHCP 구성 파일을 편집합니다.

/etc/dhcp/dhcpd4.conf 또는 /etc/dhcp/dhcpd6.conf 파일을 만듭니다. 자세한 내용은 dhcpd.conf(5) 매뉴얼 페이지를 참조하십시오.

### 3 필요한 서비스를 사용으로 설정합니다.

```
# svcadm enable service
```

service는 다음 값 중 하나일 수 있습니다.

svc:/network/dhcp/server:ipv4      IPv4 클라이언트에서 DHCP 및 BOOTP 요청을 제공합니다.

svc:/network/dhcp/server:ipv6      IPv6 클라이언트에서 DHCP 및 BOOTP 요청을 제공합니다.

svc:/network/dhcp/relay:ipv4      IPv4 클라이언트에서 DHCP 서버의 네트워크로 DHCP 및 BOOTP 요청을 중계합니다.

svc:/network/dhcp/relay:ipv6      IPv6 클라이언트에서 DHCP 서버의 네트워크로 DHCP 및 BOOTP 요청을 중계합니다.

## ▼ DHCP 서비스의 구성을 수정하는 방법

- 1 수퍼유저가 되거나 DHCP Management 프로파일에 할당된 역할이나 사용자 이름을 말합니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)를 참조하십시오.

- 2 DHCP 구성 파일을 편집합니다.

/etc/dhcp/dhcpd4.conf 또는 /etc/dhcp/dhcpd6.conf 파일을 편집합니다. 자세한 내용은 dhcpd.conf(5) 매뉴얼 페이지를 참조하십시오.

- 3 SMF 데이터를 새로 고칩니다.

```
# svcadm refresh service
```



## DHCP 클라이언트 구성 및 관리

---

이 장에서는 Oracle Solaris에 속하는 DHCP(Dynamic Host Configuration Protocol) 클라이언트에 대해 설명합니다. 클라이언트의 DHCPv4 및 DHCPv6 프로토콜이 작동하는 방법과 클라이언트의 동작에 영향을 주는 방법을 설명합니다.

한 프로토콜인 DHCPv4는 Oracle Solaris에 오랫동안 속해 왔으며, 이를 사용하여 DHCP 서버는 IPv4 네트워크 주소와 같은 구성 매개변수를 IPv4 노드로 전달할 수 있습니다.

다른 프로토콜인 DHCPv6을 사용하여 DHCP 서버는 IPv6 네트워크 주소와 같은 구성 매개변수를 IPv6 노드로 전달할 수 있습니다. DHCPv6은 "IPv6 Stateless 주소 자동 구성"(RFC 2462)에 대응하는 Stateful 항목으로, 구성 매개변수를 얻기 위해 Stateless와 별도로 또는 동시에 사용할 수 있습니다.

이 장은 다음 정보를 포함합니다.

- 175 페이지 “DHCP 클라이언트 정보”
- 182 페이지 “DHCP 클라이언트 사용 및 사용 안함”
- 184 페이지 “DHCP 클라이언트 관리”
- 186 페이지 “다중 네트워크 인터페이스의 DHCP 클라이언트 시스템”
- 186 페이지 “DHCPv4 클라이언트 호스트 이름”
- 188 페이지 “DHCP 클라이언트 시스템 및 이름 서비스”
- 189 페이지 “DHCP 클라이언트 이벤트 스크립트”

## DHCP 클라이언트 정보

DHCP 클라이언트는 dhcpagent 데몬입니다. LiveCD GUI 설치 프로그램을 사용하여 Oracle Solaris를 설치하는 경우 설치된 시스템에 DHCPv4 및 DHCPv6 프로토콜이 사용으로 설정됩니다. 텍스트 설치 프로그램을 사용하여 Oracle Solaris를 설치하는 경우 설치된 시스템에 네트워크를 구성하는 방법을 선택하라는 메시지가 나타납니다. 자동 네트워크 구성을 지정하는 경우 설치된 시스템에 DHCPv4 및 DHCPv6 프로토콜이 사용으로 설정됩니다.

Oracle Solaris 클라이언트가 DHCP를 사용하기 위해 다른 필요한 일은 없습니다. DHCP 서버의 구성에 따라 DHCP 서비스를 사용하는 DHCP 클라이언트 시스템에 어떤 정보가 제공될지 결정됩니다.

클라이언트 시스템이 Oracle Solaris를 이미 실행 중이지만 DHCP를 사용 중이 아닌 경우 DHCP를 사용하도록 클라이언트 시스템을 재구성할 수 있습니다. 또한 DHCP 사용을 중지하고 정적 네트워크 정보를 사용하도록 DHCP 클라이언트 시스템을 재구성할 수도 있습니다. 자세한 내용은 [182 페이지 “DHCP 클라이언트 사용 및 사용 안함”](#)을 참조하십시오.

## DHCPv6 서버

Oracle Solaris에 대해 Sun Microsystems를 통해 사용 가능한 DHCPv6 서버는 없습니다. 타사에서 제공된 서버는 Sun의 DHCPv6과 호환될 수 있고, 네트워크에 DHCPv6 서버가 있는 경우 Sun의 DHCPv6 클라이언트가 이를 사용합니다.

## DHCPv4와 DHCPv6의 차이점

DHCPv4와 DHCPv6의 두 가지 주요 차이점은 다음과 같습니다.

- **관리 모델**
  - DHCPv4 - 관리자가 각 인터페이스마다 DHCP를 사용으로 설정합니다. 논리적 인터페이스 단위로 관리가 이루어집니다.
  - DHCPv6 - 명시적 구성이 필요하지 않습니다. 이 프로토콜은 주어진 물리적 인터페이스에 사용으로 설정됩니다.
- **프로토콜 세부 정보**
  - DHCPv4 - DHCP 서버가 각 주소에 대한 서브넷 마스크를 제공합니다. 호스트 이름 옵션이 시스템 차원의 노드 이름을 설정합니다.
  - DHCPv6 - DHCPv6 서버가 아닌, Router Advertisements에서 서브넷 마스크를 제공합니다. DHCPv6 호스트 이름 옵션이 없습니다.

## DHCP 관리 모델

**DHCPv4**는 명시적 클라이언트 구성이 필요합니다. 필요할 때 주소 지정을 위해 DHCPv4 시스템을 설정해야 하고, 이는 일반적으로 초기 시스템 설치 중에 수행되거나 `ipadm` 명령 사용을 통해 동적으로 실행됩니다. [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

**DHCPv6**은 명시적 클라이언트 구성이 필요하지 않습니다. 대신, DHCP 사용이 네트워크의 등록 정보이고 이를 사용하는 신호가 로컬 라우터에서 Router Advertisement 메시지에 전달됩니다. DHCP 클라이언트는 필요에 따라 자동으로 논리적 인터페이스를 만들고 제거합니다.



DHCPv6 방식은 기존의 IPv6 Stateless (자동) 주소 구성과 관리상 매우 비슷합니다. Stateless 주소 구성의 경우 로컬 라우터에 플래그를 설정하여 주어진 접두어 세트에 대해 각 클라이언트가 보급된 접두어에 로컬 인터페이스 토큰이나 난수를 더해서 자체에 주소를 자동으로 구성해야 합니다. DHCPv6의 경우 동일한 접두어가 필요하지만 주소가 "무작위로" 지정되는 대신 DHCPv6 서버를 통해 획득, 관리됩니다.

## MAC 주소 및 클라이언트 ID

DHCPv4는 MAC 주소 및 주소 지정 목적으로 클라이언트를 식별하는 선택적 클라이언트 ID를 사용합니다. 동일한 클라이언트가 네트워크에 도착할 때마다 가능하면 동일한 주소를 얻습니다.

DHCPv6은 기본적으로 동일한 체계를 사용하지만 클라이언트 ID가 필수이고 거기에 구조를 강제 적용합니다. DHCPv6의 클라이언트 ID는 DUID(DHCP Unique Identifier) 및 IAID(Identity Association Identifier)의 두 부분으로 구성됩니다. DUID는 (DHCPv4에서처럼 단지 인터페이스가 아닌) 클라이언트 시스템을 식별하고 IAID는 해당 시스템의 인터페이스를 식별합니다.

RFC 3315에 기술된 대로, ID 연관은 서버 및 클라이언트에서 관련된 IPv6 주소 세트를 식별, 그룹화, 관리하기 위해 사용되는 수단입니다. 클라이언트는 적어도 하나의 별개의 IA를 각 네트워크 인터페이스와 연관시키고, 지정된 IA를 사용하여 해당 인터페이스의 서버에서 구성 정보를 얻어야 합니다. IA에 대한 추가 정보는 다음 절인 "프로토콜 세부 정보"를 참조하십시오.

DUID+IAID를 DHCPv4와 함께 사용할 수도 있습니다. 이들은 클라이언트 ID로 작동할 수 있도록 분명하게 서로 연결할 수 있습니다. 호환성 이유로 일반 IPv4 인터페이스에는 수행되지 않습니다. 그러나 논리적 인터페이스의 경우(bge0:1) 구성된 클라이언트 ID가 없으면 DUID+IAID가 사용됩니다.

IPv4 DHCP와 달리, DHCPv6은 "클라이언트 이름" 옵션을 제공하지 않으므로 DHCPv6 혼자만 기반으로 시스템에 이름을 지정할 방법이 없습니다. 대신, DHCPv6에서 제공된 주소와 어울리는 DNS 이름을 알아야 하는 경우 해당하는 이름 정보를 찾으려면 DNS 역분석(getaddrinfo(3SOCKET) 함수를 통해 주소-이름 질의)을 사용하십시오. 이에 따라 DHCPv6만 사용 중이고 노드에 특정 이름을 부여하려면 다음과 같이 `svccfg` 명령을 사용하여 노드 이름을 지정해야 합니다.

```
# svccfg -s svc:/system/identity:node setprop config/nodename = astring: hostname
```

## 프로토콜 세부 정보

DHCPv4에서는 DHCP 서버가 지정된 주소에 사용할 서브넷 마스크를 제공합니다. DHCPv6에서는 서브넷 마스크("접두어 길이"라고도 함)가 Router Advertisements로 지정되고 DHCP 서버에서 제어하지 않습니다.

DHCPv4는 시스템 차원의 노드 이름을 설정하는 데 사용되는 호스트 이름 옵션을 전달합니다. DHCPv6에는 해당 옵션이 없습니다.

DHCPv6용 클라이언트 ID를 구성하려면 시스템에서 자동 선택을 허용하기보다는 DUID를 지정해야 합니다. 이는 때문에 대해 전역적으로 또는 인터페이스 단위로 수행할 수 있습니다. 다음 형식을 사용하여 전역 DUID를 설정합니다(처음의 점 주의).

#### **.v6.CLIENT\_ID=DUID**

특정 인터페이스에서 주어진 DUID를 사용하도록 설정하려면(시스템에서 다중 독립 클라이언트가 DHCPv6 서버로 보임) 다음을 사용합니다.

#### **bge0.v6 CLIENT ID=DUID**

각 ID 연관(IA)은 한가지 유형의 주소를 보유합니다. 예를 들어, 임시 주소용 ID 연관(IA\_TA)은 임시 주소를 보유하고 비임시 주소용 ID 연관(IA\_NA)은 영구적인 지정된 주소를 전달합니다. 이 설명서에 기술된 DHCPv6 버전은 IA\_NA 연관만 제공합니다.

Oracle Solaris는 요청 시 정확히 하나의 IAID를 각 인터페이스에 지정하고 IAID는 루트 파일 시스템의 파일에 저장되므로 시스템 전체 수명 동안 일정하게 유지됩니다.

## 논리적 인터페이스

DHCPv4 클라이언트에서 각 논리적 인터페이스는 독립적이며 관리 단위입니다. 0번째 논리적 인터페이스에 더해서(식별자로 인터페이스 MAC 주소가 기본 설정) 사용자는 dhcpagent 구성 파일에서 CLIENT\_ID를 지정하여 특정 논리적 인터페이스에서 DHCP가 실행되도록 구성할 수 있습니다. 예를 들면 다음과 같습니다.

#### **hme0:1.CLIENT\_ID=orangutan**

DHCPv6은 다르게 작동합니다. IPv4와 달리, IPv6 인터페이스의 0번째 논리적 인터페이스는 항상 link-local입니다. link-local을 사용하면 DHCP 서버와 같은 사용 가능한 지정 방법이 없을 때 IP 네트워크의 장치에 IP 주소를 자동으로 지정할 수 있습니다. 0번째 논리적 인터페이스를 DHCP 통제하에 놓을 수 없으므로 DHCPv6이 0번째 논리적 인터페이스("물리적" 인터페이스라고도 함)에서 실행되더라도 0이 아닌 논리적 인터페이스에만 주소가 할당됩니다.

DHCPv6 클라이언트 요청에 대한 응답으로 DHCPv6 서버는 구성할 클라이언트에 대한 주소 목록을 반환합니다.

## 옵션 협상

DHCPv6에는 클라이언트가 선호하는 내용을 서버에 힌트로 알려주는 Option Request Option이 있습니다. 모든 가능한 옵션을 서버에서 클라이언트로 보낸 경우 그 중 일부가 클라이언트로 가능 도중에 삭제될 것이라는 정보를 보낼 수 있습니다. 서버는 힌트를 사용하여 회신에 포함할 옵션을 고를 수 있습니다. 다른 방법으로, 서버가 힌트를 무시하고 다른 항목을 고를 수 있습니다. 예를 들어, Oracle Solaris에서 선호 옵션이 Oracle Solaris DNS 주소 도메인 또는 NIS 주소 도메인을 포함할 수 있지만, net BIOS 서버를 포함하지는 않습니다.

DHCPv4에도 동일한 유형의 힌트가 제공되지만 특수한 Option Request Option이 없습니다. 대신, DHCPv4는 /etc/default/dhcpagent의 PARAM\_REQUEST\_LIST를 사용합니다.

## 구성 구문

/etc/default/dhcpagent를 사용하여 기존 DHCPv4 클라이언트와 동일한 방법으로 DHCPv6 클라이언트를 구성합니다.

구문의 인수는 인터페이스 이름(있는 경우)과 구성될 매개변수 사이에 ".v6" 표시자로 지정됩니다. 예를 들어, 전역 IPv4 옵션 요청 목록은 다음과 같이 설정됩니다.

```
PARAM_REQUEST_LIST=1,3,6,12,15,28,43
```

다음과 같이 개별 인터페이스에서 호스트 이름 옵션을 생략하도록 구성할 수 있습니다.

```
bge0.PARAM_REQUEST_LIST=1,3,6,15,28,43
```

DHCPv6의 전역 요청 목록을 설정하려면 선행 점에 주의하십시오.

```
.v6.PARAM_REQUEST_LIST=23,24
```

또는, 개별 인터페이스를 설정하려면 다음 예제를 따르십시오.

```
bge0.v6.PARAM_REQUEST_LIST=21,22,23,24
```

참조용으로 여기에 DHCPv6 구성의 실제 /etc/default/dhcpagent 파일이 있습니다.

```
# The default DHCPv6 parameter request list has preference (7), unicast (12),
# DNS addresses (23), DNS search list (24), NIS addresses (27), and
# NIS domain (29). This may be changed by altering the following parameter-
# value pair. The numbers correspond to the values defined in RFC 3315 and
# the IANA dhcpv6-parameters registry.
.v6.PARAM_REQUEST_LIST=7,12,23,24,27,29
```

## DHCP 클라이언트 시작

대부분의 경우 DHCPv6 클라이언트 시작을 위해 아무것도 필요하지 않습니다. in.ndpd 데몬이 필요할 때 자동으로 DHCPv6을 시작합니다.

그러나 DHCPv4의 경우 Oracle Solaris 설치 중에 시작되지 않았으면 클라이언트 시작을 요청해야 합니다. [183 페이지](#) “DHCP 클라이언트를 사용으로 설정하는 방법”을 참조하십시오.

dhcpagent 데몬은 시스템 부트와 관련한 다른 프로세스에서 필요한 구성 정보를 얻습니다. 이러한 이유로 시스템 시작 스크립트가 부트 프로세스에서 조기에 dhcpagent를 시작하고 DHCP 서버에서 네트워크 구성 정보가 도착할 때까지 기다립니다.

기본값은 DHCPv6을 실행하는 것이지만 DHCPv6이 실행되지 않도록 선택할 수 있습니다. DHCPv6이 실행을 시작한 후에 `ipadm delete-addr` 명령을 사용하여 중지할 수 있습니다. `/etc/inet/ndpd.conf` 파일을 수정하여 DHCPv6이 재부트 시 시작되지 않도록 사용 안함으로 설정할 수도 있습니다.

다음 예는 DHCPv6을 즉시 종료하는 방법을 보여줍니다.

```
ex# echo ifdefault StatefulAddrConf false >> /etc/inet/ndpd.conf
ex# pkill -HUP -x in.ndpd
ex# ipadm delete-addr -r dhcp-addrobj
```

시작 시, 영구 DHCP 구성이 시스템에 존재하면 `dhcpageant`가 시작 스크립트 프로세스의 일부로 시작됩니다. 그런 다음 `dhcpageant`가 [165 페이지 “DHCP의 작동 방식”](#)에 설명된 대로 네트워크 인터페이스를 구성합니다.

## DHCPv6 통신

수동 구성으로 호출된 DHCPv4와 달리, DHCPv6은 RA(Router Advertisements)로 호출됩니다. 라우터 구성 방법에 따라 시스템이 Router Advertisement 메시지가 수신된 인터페이스에서 DHCPv6을 자동으로 호출하고 DHCP를 사용하여 주소나 기타 매개변수를 얻거나, 또는 시스템이 DHCPv6을 사용하여 주소 이외의 데이터(예: DNS 서버)만 요청합니다.

`in.ndpd` 데몬이 Router Advertisement 메시지를 수신합니다. 이는 시스템에서 IPv6용으로 배관된 모든 인터페이스에서 자동으로 수행됩니다. `in.ndpd`가 DHCPv6이 실행되도록 지정하는 RA를 발견하면 이를 호출합니다.

`in.ndpd`에서 DHCPv6이 시작하지 못하도록 하려면 `/etc/inet/ndpd.conf` 파일을 변경할 수 있습니다.

다음 `ipadm` 버전 중 하나를 사용하여 DHCPv6을 시작한 후에 중지할 수도 있습니다.

```
ipadm delete-addr dhcp-addrobj
```

또는

```
ipadm delete-addr -r dhcp-addrobj
```

## DHCP 클라이언트 프로토콜이 네트워크 구성 정보를 관리하는 방법

DHCPv4 및 DHCPv6 클라이언트 프로토콜은 여러 가지 방법으로 네트워크 구성 정보를 관리합니다. 주요 차이점은, DHCPv4에서는 단일 주소의 임대 및 이와 어울리는 옵션을 협상하는 것입니다. DHCPv6에서는 일괄 주소 및 일괄 옵션에 걸쳐 협상이 이루어집니다.

DHCPv4 클라이언트와 서버 간의 상호 작용에 대한 배경 정보는 10 장, “DHCP 정보(개요)”를 참조하십시오.

## DHCPv4 클라이언트가 네트워크 구성 정보를 관리하는 방법

DHCP 서버에서 정보 패킷을 얻은 후에 `dhcpcagent`가 네트워크 인터페이스를 구성하고 인터페이스를 가져옵니다. 때문에 IP 주소에 대한 임대 기간 동안 인터페이스를 제어하고 내부 테이블에서 구성 데이터를 유지 관리합니다. 시스템 시작 스크립트가 `dhcpcinfo` 명령을 사용하여 내부 테이블에서 구성 옵션 값을 추출합니다. 값을 사용하여 시스템을 구성하고 네트워크에서 통신이 가능합니다.

`dhcpcagent` 데몬은 시간이 경과할 때까지(대개 임대 시간의 절반) 수동적으로 기다립니다. 그런 다음 데몬이 DHCP 서버에서 임대 연장을 요청합니다. 인터페이스가 작동 중지되거나 IP 주소가 변경되었다고 `dhcpcagent`에 알리면 `ipadm` 명령에서 별도로 지시할 때까지 데몬이 인터페이스를 제어하지 않습니다. 인터페이스가 작동 중이고 IP 주소가 변경되지 않았음을 `dhcpcagent`가 알게 되면 데몬이 서버에 임대 갱신 요청을 보냅니다. 임대를 갱신할 수 없으면 `dhcpcagent`가 임대 시간 끝에 인터페이스를 끌어내립니다.

`dhcpcagent`가 임대에 관련된 조치를 실행할 때마다 데몬이 `/etc/dhcp/eventhook`라는 실행 파일을 찾습니다. 이 이름을 가진 실행 파일을 찾으면 `dhcpcagent`가 실행 파일을 호출합니다. 이벤트 실행 파일 사용에 대한 자세한 내용은 189 페이지 “DHCP 클라이언트 이벤트 스크립트”를 참조하십시오.

## DHCPv6 클라이언트가 네트워크 구성 정보를 관리하는 방법

클라이언트와 서버 간의 DHCPv6 통신은 클라이언트가 서버를 찾기 위해 Solicit 메시지를 발송하는 것으로 시작합니다. 응답에서 DHCP 서비스에 사용 가능한 모든 서버가 Advertise 메시지를 보냅니다. 서버 메시지는 여러 IA\_NA(Identity Association Non-Temporary Address) 레코드와 기타 서버가 제공할 수 있는 옵션(예: DNS 서버 주소)을 포함합니다.

클라이언트가 Request 메시지에 고유의 IA\_NA/IAADDR 레코드를 설정하여 특정 주소(및 이것의 배수)를 요청할 수 있습니다. 클라이언트는 일반적으로 이전 주소가 기록된 경우 특정 주소를 요청하고, 서버는 가능하면 똑같은 것을 제공합니다. 클라이언트가 무엇이든 관계없이(주소를 전혀 요청하지 않더라도) 단일 DHCPv6 트랜잭션에 대해 서버가 원하는 수의 주소를 클라이언트에 제공할 수 있습니다.

이것은 클라이언트와 서버 간에 발생하는 메시지 대화입니다.

- 클라이언트가 서버를 찾기 위해 Solicit 메시지를 보냅니다.
- 서버가 Advertise 메시지를 보내어 DHCP 서비스에 사용 가능성을 나타냅니다.
- 클라이언트가 Request 메시지를 보내어 가장 큰 선호 값으로 서버로부터 IP 주소를 포함한 구성 매개변수를 요청합니다. 서버 선호 값이 관리자에 의해 설정되고 하한값 0부터 상한값 255까지 확장됩니다.
- 서버가 주소 임대 및 구성 데이터를 포함하는 Reply 메시지를 보냅니다.

Advertise 메시지의 선호 값이 255이면 DHCPv6 클라이언트가 해당 서버를 즉시 선택합니다. 가장 선호되는 서버가 응답하지 않거나 Request 메시지에 성공적인 Reply를 실패하면 더 이상 Advertise 메시지를 구할 수 없을 때까지 (순서대로) 덜 선호되는 서버를 계속 찾습니다. 이 시점에서 클라이언트가 Solicit 메시지를 다시 보내어 시작합니다.

선택한 서버가 Solicit 또는 Request 메시지에 대한 응답으로 지정된 주소 및 구성 매개변수를 포함하는 Reply 메시지를 보냅니다.

## DHCP 클라이언트 종료

종료 시, 클라이언트가 Release 메시지를 클라이언트에 주소를 지정한 서버에 보내어 클라이언트가 더 이상 하나 이상의 지정된 주소를 사용하지 않음을 나타냅니다. DHCPv4 클라이언트 시스템이 정상적으로 종료할 때 dhcpagent가 현재 구성 정보를 파일(있는 경우)에 작성합니다. DHCPv4의 파일 이름은 /etc/dhcp/interface.dhc이고 DHCPv6의 파일 이름은 /etc/dhcp/interface.dh6입니다. 기본적으로 임대는 해제가 아니라 저장되므로 DHCP 서버에서 IP 주소가 활성 사용 중이 아님을 감지할 수 없습니다. 따라서 클라이언트가 다음 부트 시 주소를 쉽게 되찾을 수 있습니다. 이 기본 동작은 `ipadm delete-addr dhcp-addrobj` 명령과 동일합니다.

시스템을 재부트할 때 해당 파일의 임대가 여전히 유효하면 dhcpagent가 동일한 IP 주소 및 네트워크 구성 정보를 사용하도록 약속 요청을 보냅니다. DHCPv4의 경우 이것은 Request 메시지입니다. DHCPv6의 경우 Confirm 메시지입니다.

DHCP 서버가 이 요청을 허가하면 dhcpagent가 시스템을 종료할 때 디스크에 작성된 정보를 사용할 수 있습니다. 서버가 클라이언트의 정보 사용을 허가하지 않으면 dhcpagent가 165 페이지 “DHCP의 작동 방식”에 설명된 DHCP 프로토콜 시퀀스를 시작합니다. 그 결과, 클라이언트가 새 네트워크 구성 정보를 얻습니다.

## DHCP 클라이언트 사용 및 사용 안함

Oracle Solaris를 이미 실행 중이고 DHCP를 사용 중이 아닌 시스템에서 DHCP 클라이언트를 사용으로 설정하려면 먼저 시스템 구성을 해제해야 합니다. 시스템을 부트할 때 시스템을 설정하고 DHCP 클라이언트를 사용으로 설정하려면 몇 가지 명령을 실행해야 합니다.

---

주 - 대부분의 배치에서 혼란 방법은 DHCP를 사용하기보다, 기반구조의 중요한 부분을 정적 IP 주소로 설정하는 것입니다. 네트워크의 어떤 장치(예: 라우터 및 특정 서버)가 클라이언트여야 하고 어떤 것이 안되는지 결정하는 것은, 이 설명서의 범위를 벗어납니다.

---



## ▼ DHCP 클라이언트를 사용으로 설정하는 방법

이 절차는 DHCPv4가 Oracle Solaris 설치 중 사용으로 설정되지 않은 경우에만 필요합니다. DHCPv6에는 필요 없습니다.

- 1 슈퍼유저가 되거나 DHCP Management 프로파일에 할당된 역할이나 사용자 이름을 말합니다.

DHCP 관리 프로파일에 대한 자세한 내용은 171 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 시스템을 재구성합니다.

다음 구성 방법 중 하나를 선택합니다.

- 대화식으로 시스템을 재구성합니다.

```
# sysconfig configure
```

시스템 구성 대화식 도구를 시작할 때 Network(네트워크) 화면에서 Automatic(자동) 네트워크 구성을 선택합니다.

- 비대화식으로 시스템을 재구성합니다.

```
# sysconfig configure -c sc_profile
```

sc\_profile 구성 파일 사용에 대한 자세한 내용은 [sysconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ DHCP 클라이언트를 사용 안함으로 설정하는 방법

- 1 슈퍼유저가 되거나 DHCP Management 프로파일에 할당된 역할이나 사용자 이름을 말합니다.

DHCP 관리 프로파일에 대한 자세한 내용은 171 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 시스템을 재구성합니다.

다음 구성 방법 중 하나를 선택합니다.

- 대화식으로 시스템을 재구성합니다.

```
# sysconfig configure
```

시스템 구성 대화식 도구를 시작할 때 Network(네트워크) 화면에서 네트워크 구성으로 Manual(수동) 또는 None(없음)을 선택합니다.

- 비대화식으로 시스템을 재구성합니다.

```
# sysconfig configure -c sc_profile
```

sc\_profile 구성 파일 사용에 대한 자세한 내용은 [sysconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## DHCP 클라이언트 관리

DHCP 클라이언트 소프트웨어는 정상적인 시스템 운영하에서 관리가 필요하지 않습니다. dhcpagent 데몬은 시스템을 부트할 때 자동으로 시작하고, 임대를 재협상하고, 시스템을 종료할 때 중지합니다. 직접 dhcpagent 데몬을 수동으로 시작 및 중지하면 안됩니다. 대신, 클라이언트 시스템에서 슈퍼유저로 ipadm 명령을 사용하여 필요한 경우 dhcpagent의 네트워크 인터페이스 관리에 영향을 미칠 수 있습니다.

## DHCP 클라이언트와 함께 사용된 ipadm 명령 옵션

이 절은 [ipadm\(1M\)](#) 매뉴얼 페이지에 문서화된 명령 옵션을 요약합니다.

ipadm 명령은 다음을 수행할 수 있습니다.

- **IP 인터페이스 만들기** - ipadm create-ip 명령이 IP 인터페이스를 만들면 사용자가 IP 주소로 구성합니다. 주소는 정적 또는 동적일 수 있습니다. IP 인터페이스를 만드는 것은 주소를 지정하기 전에 꼭 필요한 명령입니다.
- **DHCP 클라이언트 시작** - ipadm create-addr -T dhcp dhcp-addrobj 명령이 dhcpagent와 DHCP 서버 간에 상호 작용을 시작하여 IP 주소 및 새로운 구성 옵션을 얻습니다. 이 명령은 IP 주소를 추가하거나 서브넷 마스크를 변경할 때와 같이 클라이언트가 즉시 사용할 정보를 변경할 때 유용합니다.
- **네트워크 구성 정보만 요청** - ipadm refresh-addr -i dhcp-addrobj 명령은 dhcpagent가 IP 주소를 제외한 네트워크 구성 매개변수에 대한 요청을 실행하도록 합니다. 이 명령은 네트워크 인터페이스에 정적 IP 주소가 있지만 클라이언트 시스템에 업데이트된 네트워크 옵션이 필요할 때 유용합니다. 예를 들어, 이 명령은 DHCP를 IP 주소 관리에 사용하지 않지만 네트워크의 호스트 구성에 사용하려는 경우 유용합니다.
- **임대 연장 요청** - ipadm refresh-addr dhcp-addrobj 명령은 dhcpagent가 임대 갱신 요청을 실행하도록 합니다. 클라이언트가 자동으로 임대를 갱신하도록 요청합니다. 그러나 임대 시간을 변경한 후에 다음 임대 갱신 시도를 기다리지 않고 새 임대 시간을 즉시 사용하도록 하려면 이 명령을 사용할 수 있습니다.
- **IP 주소 해제** - ipadm delete-addr -r dhcp-addrobj 명령은 dhcpagent가 네트워크 인터페이스에서 사용된 IP 주소를 양도하도록 합니다. IP 주소 해제는 임대가 만료될 때 자동으로 발생합니다. 램탑에서 네트워크를 남겨 두고 새 네트워크에서 시스템을



시작하려고 할 때 이 명령을 실행할 수 있습니다. `/etc/default/dhcpagent` 구성 파일 `RELEASE_ON_SIGTERM` 등록 정보를 참조하십시오.

- **IP 주소 삭제** - `ipadm delete-addr dhcp-addrobj` 명령은 `dhcpagent`가 DHCP 서버에 알리지 않고 네트워크 인터페이스를 끌어내리고 파일 시스템에 임대를 캐싱하도록 합니다. 이 명령으로 클라이언트는 재부트할 때 동일한 IP 주소를 사용할 수 있습니다.

주 - 현재 `ipadm` 명령에는 `ifconfig [inet6] interface status` 명령에 상응하는 기능이 없습니다.

## DHCP 클라이언트 구성 매개변수 설정

클라이언트 시스템의 `/etc/default/dhcpagent` 파일은 `dhcpagent`의 조정 가능한 매개변수를 포함합니다. 텍스트 편집기를 사용하여 클라이언트 운영에 영향을 주는 여러 매개변수를 변경할 수 있습니다. `/etc/default/dhcpagent` 파일은 잘 문서화되어 있으므로 자세한 내용은 이 파일과 [dhcpagent\(1M\)](#) 매뉴얼 페이지를 참조해야 합니다.

기본적으로 DHCP 클라이언트는 다음과 같이 구성됩니다.

### DHCPv4

- 클라이언트 시스템에 특정 호스트 이름이 필요하지 않습니다.  
클라이언트가 특정 호스트 이름을 요청하도록 하려면 [186 페이지 “DHCPv4 클라이언트 호스트 이름”](#)을 참조하십시오.
- 클라이언트의 기본 요청은 `/etc/default/dhcpagent`에 제공되고 DNS 서버, DNS 도메인 및 브로드캐스트 주소를 포함합니다.  
`/etc/default/dhcpagent` 파일의 `PARAM_REQUEST_LIST` 키워드에서 DHCP 클라이언트의 매개변수 파일이 더 많은 옵션을 요청하도록 설정할 수 있습니다. DHCP 서버가 특별히 요청되지 않은 옵션을 제공하도록 구성할 수 있습니다. DHCP 서버 매크로를 사용하여 클라이언트에 정보를 보내는 방법은 `dhcpcd(8)` 매뉴얼 페이지 및 [System Administration Guide: IP Services](#)의 “Working With DHCP Macros (Task Map)”을 참조하십시오.

### DHCPv4 및 DHCPv6

- 클라이언트 시스템이 하나의 물리적 네트워크 인터페이스에서 DHCP를 사용합니다.  
여러 개의 물리적 네트워크 인터페이스에서 DHCP를 사용하려면 [186 페이지 “다중 네트워크 인터페이스의 DHCP 클라이언트 시스템”](#)을 참조하십시오.
- DHCP 클라이언트가 Oracle Solaris 설치 후에 구성된 경우 이름 서비스 클라이언트로 자동으로 구성되지 않습니다.  
DHCP 클라이언트에서 이름 서비스 사용에 대한 자세한 내용은 [188 페이지 “DHCP 클라이언트 시스템 및 이름 서비스”](#)를 참조하십시오.

## 다중 네트워크 인터페이스의 DHCP 클라이언트 시스템

DHCP 클라이언트는 한 시스템에서 여러 다른 인터페이스를 동시에 관리할 수 있습니다. 인터페이스는 물리적 인터페이스 또는 논리적 인터페이스일 수 있습니다. 각 인터페이스에는 고유의 IP 주소 및 임대 시간이 있습니다. 여러 개의 네트워크 인터페이스가 DHCP용으로 구성된 경우 클라이언트가 이들을 구성하기 위해 별도의 요청을 실행합니다. 클라이언트는 각 인터페이스마다 별도의 네트워크 구성 매개변수를 유지 관리합니다. 매개변수가 별도로 저장되더라도 일부는 사실상 전역 매개변수입니다. 전역 매개변수는 특정 네트워크 인터페이스가 아닌 시스템에 전체적으로 적용됩니다.

전역 매개변수의 예로 호스트 이름, NIS 도메인 이름, 시간대 등이 있습니다. 전역 매개변수는 대개 각 인터페이스마다 다른 값을 가집니다. 그러나 각 시스템과 연관된 각 전역 매개변수에 대해 하나의 값만 사용할 수 있습니다. 전역 매개변수에 대한 질의 응답이 하나만 있도록 하려면 기본 네트워크 인터페이스의 매개변수만 사용됩니다.

DHCP 클라이언트는 논리적 인터페이스 및 물리적 인터페이스에 대한 임대를 동일하게 관리합니다. 단, 논리적 인터페이스에는 다음 제한 사항이 있습니다.

- DHCP 클라이언트가 논리적 인터페이스와 연관된 기본 경로를 관리하지 않습니다.

Oracle Solaris 커널이 경로를 논리적 인터페이스가 아닌 물리적 인터페이스와 연관시킵니다. 물리적 인터페이스의 IP 주소가 설정된 경우 필요한 기본 경로가 경로 지정 테이블에 배치되어야 합니다. DHCP가 나중에 물리적 인터페이스와 연관된 논리적 인터페이스를 구성하는 경우 필요한 경로가 이미 제자리에 있어야 합니다. 논리적 인터페이스가 동일한 경로를 사용합니다.

물리적 인터페이스에서 임대가 만료되면 DHCP 클라이언트가 인터페이스와 연관된 기본 경로를 제거합니다. 논리적 인터페이스에서 임대가 만료되면 DHCP 클라이언트가 논리적 인터페이스와 연관된 기본 경로를 제거하지 않습니다. 연관된 물리적 인터페이스 및 다른 가능한 논리적 인터페이스가 동일한 경로를 사용해야 할 수 있습니다.

DHCP 제어 인터페이스와 연관된 기본 경로를 추가/제거해야 하는 경우 DHCP 클라이언트 이벤트 스크립트 방식을 사용할 수 있습니다. [189 페이지 “DHCP 클라이언트 이벤트 스크립트”](#)를 참조하십시오.

## DHCPv4 클라이언트 호스트 이름

기본적으로 DHCPv4 클라이언트는 DHCP 서버에서 호스트 이름을 제공할 것으로 기대하기 때문에 고유의 호스트 이름을 제공하지 않습니다. DHCPv4 서버는 기본적으로 DHCPv4 클라이언트에 호스트 이름을 제공하도록 구성됩니다. DHCPv4 클라이언트와 서버를 함께 사용할 때 이러한 기본값이 잘 작동합니다. 그러나 DHCPv4 클라이언트를 타사 DHCP 서버와 사용할 때 클라이언트가 서버에서 호스트 이름을 받지 못할 수 있습니다. DHCP 클라이언트가 DHCP를 통해 호스트 이름을 받지 못하면 클라이언트

시스템이 호스트 이름으로 사용할 이름에 대해 `svc:/system/identity:node` 서비스의 `config/nodename` 등록 정보에 설정된 값을 검사합니다. 파일이 비어 있으면 호스트 이름이 `unknown`으로 설정됩니다.

DHCP 서버가 DHCP Hostname 옵션에 이름을 제공하면 `svc:/system/identity:node` 서비스의 `config/nodename` 등록 정보에 설정된 값과 다르더라도 클라이언트가 호스트 이름을 사용합니다. 클라이언트가 특정 호스트 이름을 사용하도록 하려면 클라이언트가 해당 이름을 요청하도록 설정할 수 있습니다. 다음 절차를 참조하십시오.

주 - 다음 절차는 모든 DHCP 서버와 작동하지 않습니다. 이 절차를 통해 클라이언트가 DHCP 서버에 특정 호스트 이름을 보내고 교대로 동일한 이름을 기대하도록 요구하게 됩니다.

그러나 DHCP 서버는 이 요청을 존중할 필요가 없으며 대부분 무시합니다. 간단히 다른 이름을 반환합니다.

## ▼ DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법

수행 단계는 IP 인터페이스가 DHCP 주소로 존재하는지 여부에 따라 다릅니다.

### 1 IP 인터페이스가 DHCP 주소로 존재하는 경우 다음을 수행합니다.

#### a. 기존 DHCP 주소를 삭제합니다.

```
# ipadm delete-addr -r dhcp-addrobj
```

#### b. 사용할 특정 호스트 이름으로 새 DHCP 주소를 등록합니다.

```
# ipadm create-addr -T dhcp -h hostname dhcp-addrobj
```

### 2 IP 인터페이스가 아직 존재하지 않는 경우 다음을 수행합니다.

#### a. IP 인터페이스를 만듭니다.

```
# ipadm create-ip interface
```

#### b. 사용할 특정 호스트 이름으로 DHCP 주소를 등록합니다.

```
# ipadm create-addr -T dhcp -h hostname dhcp-addrobj
```

## DHCP 클라이언트 시스템 및 이름 서비스

Oracle Solaris 시스템은 DNS, NIS 및 로컬 파일 저장소(/etc/inet/hosts)와 같은 이름 서비스를 지원합니다. 각 이름 서비스는 사용하기 전에 일부 구성이 필요합니다.

name-service/switch SMF 서비스가 적절히 구성되어야 합니다. 자세한 내용은 [nsswitch.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

DHCP 클라이언트 시스템이 이름 서비스를 사용하기 전에 시스템을 이름 서비스의 클라이언트로 구성해야 합니다. 기본적으로, 그리고 시스템 설치 중에 구성되지 않는 한 로컬 파일만 사용됩니다.

다음 표는 각 이름 서비스 및 DHCP에 관련된 문제를 요약한 것입니다. 이 표는 각 이름 서비스에 대한 클라이언트를 설정하는 데 도움이 되는 문서에 대한 상호 참조를 포함합니다.

표 12-1 DHCP 클라이언트 시스템에 대한 이름 서비스 클라이언트 설정 정보

이름 서비스	클라이언트 설정 정보
NIS	<p>DHCP를 사용하여 Oracle Solaris 네트워크 설치 정보를 클라이언트 시스템으로 보내는 경우 NISservs 및 NISdmain 옵션을 포함하는 구성 매크로를 사용할 수 있습니다. 이러한 옵션은 NIS 서버의 IP 주소와 NIS 도메인 이름을 클라이언트로 전달합니다. 그러면 클라이언트가 자동으로 NIS 클라이언트가 됩니다.</p> <p>DHCP 클라이언트 시스템이 Oracle Solaris를 이미 실행 중인 경우 DHCP 서버가 NIS 정보를 클라이언트로 보낼 때 NIS 클라이언트가 해당 시스템에 자동으로 구성되지 않습니다.</p> <p>DHCP 서버가 NIS 정보를 DHCP 클라이언트 시스템으로 보내도록 구성된 경우 다음과 같이 클라이언트에서 dhcpinfo 명령을 사용하면 클라이언트에 제공된 값을 볼 수 있습니다.</p> <pre># /usr/sbin/dhcpinfo NISdmain</pre> <pre># /usr/sbin/dhcpinfo NISServs</pre> <p>주 - DHCPv6의 경우 다음과 같이 -v6 및 다른 프로토콜 키워드를 명령에 포함합니다.</p> <pre># /usr/sbin/dhcpinfo -v6 NISDomain</pre> <pre># /usr/sbin/dhcpinfo -v6 NISServers</pre> <p>시스템을 NIS 클라이언트로 설정할 때 NIS 도메인 이름 및 NIS 서버에 대한 반환된 값을 사용합니다.</p> <p>표준 방법으로 DHCP 클라이언트 시스템에 대해 NIS 클라이언트를 설정합니다. <a href="#">Oracle Solaris Administration: Naming and Directory Services</a>의 6 장, “Setting Up and Configuring NIS (Tasks)”을 참조하십시오.</p> <p>참고 - dhcpinfo 및 ypinit를 사용하는 스크립트를 작성하여 DHCP 클라이언트 시스템에서 NIS 클라이언트 구성을 자동화할 수 있습니다.</p>

표 12-1 DHCP 클라이언트 시스템에 대한 이름 서비스 클라이언트 설정 정보 (계속)

이름 서비스	클라이언트 설정 정보
/etc/inet/hosts	이름 서비스로 /etc/inet/hosts를 사용할 DHCP 클라이언트 시스템에 대해 /etc/inet/hosts 파일을 설정해야 합니다.  DHCP 도구에 의해 DHCP 클라이언트 시스템의 호스트 이름이 고유의 /etc/inet/hosts 파일에 추가됩니다. 그러나 네트워크의 다른 시스템의 /etc/inet/hosts 파일에 호스트 이름을 수동으로 추가해야 합니다. DHCP 서버 시스템이 이름 분석에 /etc/inet/hosts를 사용하는 경우 시스템에서 클라이언트의 호스트 이름을 수동으로 추가해야 합니다.
DNS	DHCP 클라이언트 시스템이 DHCP를 통해 DNS 도메인 이름을 수신하는 경우 dns/client SMF 서비스의 등록 정보도 자동으로 구성됩니다. DNS에 대한 자세한 내용은 <a href="#">Oracle Solaris Administration: Naming and Directory Services</a> 를 참조하십시오.

## DHCP 클라이언트 이벤트 스크립트

DHCP 클라이언트를 설정하여 클라이언트 시스템에 적절한 동작을 수행할 수 있는 실행 파일 프로그램 또는 스크립트를 실행할 수 있습니다. 프로그램 또는 스크립트는 **이벤트 스크립트**라고 하며, 특정 DHCP 임대 이벤트가 발생한 후 자동으로 실행됩니다. 이벤트 스크립트를 사용하여 특정 임대 이벤트에 대한 응답으로 다른 명령, 프로그램 또는 스크립트를 실행할 수 있습니다. 이 기능을 사용하려면 고유의 이벤트 스크립트를 제공해야 합니다.

다음 이벤트 키워드가 dhcpgent에서 DHCP 임대 이벤트를 구별하는 데 사용됩니다.

이벤트 키워드	설명
BOUND 및 BOUND6	DHCP용으로 인터페이스가 구성됩니다. 클라이언트가 DHCP 서버에서 확인 메시지(DHCPv4 ACK 또는 DHCPv6 Reply)를 수신하여 IP 주소에 대한 임대 요청을 부여합니다. 인터페이스를 성공적으로 구성한 후에 즉시 이벤트 스크립트가 호출됩니다.
EXTEND 및 EXTEND6	클라이언트가 리스를 성공적으로 연장합니다. 클라이언트가 DHCP 서버에서 갱신 요청에 대한 확인 메시지를 수신한 후에 즉시 이벤트 스크립트가 호출됩니다.
EXPIRE 및 EXPIRE6	임대 시간이 다 되었을 때 임대가 만료됩니다. DHCPv4의 경우, 임대된 주소가 인터페이스에서 제거되고 인터페이스가 작동 중지로 표시되기 전에 즉시 이벤트 스크립트가 호출됩니다. DHCPv6의 경우, 마지막 남은 임대된 주소가 인터페이스에서 제거되기 전에 바로 이벤트 스크립트가 호출됩니다.
DROP 및 DROP6	클라이언트가 임대를 취소하여 DHCP 컨트롤에서 인터페이스를 제거합니다. 인터페이스를 DHCP 제어에서 제거한 후에 즉시 이벤트 스크립트가 호출됩니다.

RELEASE 및 RELEASE6	클라이언트가 IP 주소를 양도합니다. 클라이언트가 인터페이스에서 주소를 해제하고 DHCPv4 RELEASE 또는 DHCPv6 Release 패킷을 DHCP 서버로 보내기 전에 즉시 이벤트 스크립트가 호출됩니다.
INFORM 및 INFORM6	인터페이스가 DHCPv4 INFORM 또는 DHCPv6 Information-Request 메시지를 통해 DHCP 서버에서 신규 또는 업데이트된 구성 정보를 획득합니다. 이러한 이벤트는 DHCP 클라이언트가 서버에서 구성 매개변수만 얻고 IP 주소 임대를 얻지 않을 때 발생합니다.
LOSS6	임대 만료 중 하나 이상의 유효한 임대가 계속 남아 있으면 만료된 주소가 제거되기 전에 바로 이벤트 스크립트가 호출됩니다. 이러한 제거 예정 항목은 IFF_DEPRECATED 플래그로 표시됩니다.

이러한 이벤트를 사용하여 `dhcpgagent`는 다음 명령을 호출합니다.

```
/etc/dhcp/eventhook interface event
```

여기서 *interface*는 DHCP를 사용 중인 인터페이스이고 *event*는 이전에 설명된 이벤트 키워드 중 하나입니다. 예를 들어, 인터페이스가 DHCP용으로 처음 구성될 때 다음과 같이 `dhcpgagent`가 이벤트 스크립트를 호출합니다.

```
/etc/dhcp/eventhook net0 BOUND
```

이벤트 스크립트 기능을 사용하려면 다음을 수행해야 합니다.

- 실행 파일 이름을 `/etc/dhcp/eventhook`로 지정합니다.
- `root`가 될 파일의 소유자를 설정합니다.
- 사용 권한을 755(`rwxr-xr-x`)로 설정합니다.
- 스크립트 또는 프로그램을 작성하여 문서화된 이벤트의 응답으로 동작 순서를 수행합니다. Sun이 새 이벤트를 추가할 수 있으므로 인식할 수 없거나 조치가 필요하지 않은 이벤트를 프로그램이 자동으로 무시해야 합니다. 예를 들어, 프로그램 또는 스크립트는 이벤트가 `RELEASE`일 때 로그 파일에 작성하고 다른 모든 이벤트를 무시할 수 있습니다.
- 스크립트 또는 프로그램을 비대화식으로 만듭니다. 이벤트 스크립트를 호출하기 전에 `stdin`, `stdout`, `stderr`이 `/dev/null`에 연결됩니다. 출력 또는 오류를 보려면 파일로 재지정해야 합니다.

이벤트 스크립트가 `dhcpgagent`로부터 프로그램 환경을 상속받고 `root` 권한으로 실행합니다. 스크립트가 `dhcpinfo` 유틸리티를 사용하여 필요한 경우 인터페이스에 대한 추가 정보를 얻을 수 있습니다. 자세한 내용은 [dhcpinfo\(1\)](#) 매뉴얼 페이지를 참조하십시오.

dhcagent 데몬이 이벤트 스크립트가 모든 이벤트에서 종료되기를 기다립니다. 이벤트 스크립트가 55초 후에 종료되지 않으면 dhcagent가 SIGTERM 신호를 스크립트 프로세스로 보냅니다. 추가 3초 후에도 여전히 프로세스가 종료되지 않으면 데몬이 프로세스를 종료하기 위해 SIGKILL 신호를 보냅니다.

[dhcagent\(1M\)](#) 매뉴얼 페이지에 이벤트 스크립트의 예가 포함됩니다.





## DHCP 명령 및 파일(참조)

이 장에서는 DHCP 명령과 DHCP 파일 사이의 관계를 설명합니다. 그러나 명령 사용 방법은 설명하지 않습니다.

이 장은 다음 정보를 포함합니다.

- 193 페이지 “DHCP 명령”
- 194 페이지 “DHCP 서비스에서 사용된 파일”
- 196 페이지 “DHCP 서비스에서 사용된 SMF 서비스”

## DHCP 명령

다음 표는 네트워크에서 DHCP를 관리하는 데 사용할 수 있는 명령을 나열합니다.

표 13-1 DHCP에 사용된 명령

명령	설명
<code>/usr/lib/inet/dhcd</code>	ISC DHCP 전용: ISC DHCP 서버 데몬입니다. 자세한 내용은 <code>dhcd(8)</code> 매뉴얼 페이지를 참조하십시오.
<code>/usr/lib/inet/dhcrelay</code>	ISC DHCP 전용: DHCP 서버가 없는 네트워크의 클라이언트에서 다른 네트워크의 서버로 DHCP 및 BOOTP 요청을 중계하기 위한 수단입니다. 자세한 내용은 <code>dhcrelay(8)</code> 매뉴얼 페이지를 참조하십시오.
<code>/usr/lib/inet/in.dhcd</code>	레거시 Sun DHCP 전용: 레거시 Sun DHCP 서버 데몬입니다. 시스템을 시작할 때 데몬이 시작됩니다. 서버 데몬을 직접 시작하면 안 됩니다. DHCP 관리자, <code>svcadm</code> 명령 또는 <code>dhcpconfig</code> 를 사용하여 데몬을 시작 및 중지합니다. 문제 해결을 위해 디버그 모드로 서버를 실행하는 경우에만 데몬을 직접 호출해야 합니다. 자세한 내용은 <code>in.dhcd(1M)</code> 매뉴얼 페이지를 참조하십시오.
<code>/usr/sadm/admin/bin/dhcpmgr</code>	레거시 Sun DHCP 전용: DHCP 서비스 구성 및 관리에 사용되는 그래픽 사용자 인터페이스(GUI) 도구인 DHCP 관리자입니다. DHCP 관리자는 권장되는 DHCP 관리 도구입니다. 자세한 내용은 <code>dhcpmgr(1M)</code> 매뉴얼 페이지를 참조하십시오.

표 13-1 DHCP에 사용된 명령 (계속)

명령	설명
/usr/sbin/dhcpagent	DHCP 프로토콜의 클라이언트측을 구현하는 DHCP 클라이언트 데몬입니다. 자세한 내용은 <a href="#">dhcpagent(1M)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhcpconfig	레거시 Sun DHCP 전용: DHCP 서버 및 BOOTP 중계 에이전트를 구성/구성 해제하는 데 사용됩니다. 또한 다른 데이터 저장소 형식으로 변환하고 DHCP 구성 데이터를 가져오고 내보내는 데 사용됩니다. 자세한 내용은 <a href="#">dhcpconfig(1M)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhcpinfo	레거시 Sun DHCP 전용: Oracle Solaris 클라이언트 시스템의 시스템 시작 스크립트가 DHCP 클라이언트 데몬 <a href="#">dhcpagent</a> 에서 호스트 이름 등의 정보를 얻는 데 사용됩니다. 스크립트 또는 명령줄에서 <a href="#">dhcpinfo</a> 를 사용하여 지정된 매개변수 값을 얻을 수도 있습니다. 자세한 내용은 <a href="#">dhcpinfo(1)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhtadm	레거시 Sun DHCP 전용: <a href="#">dhcptab</a> 테이블의 옵션 및 매크로를 변경하는 데 사용됩니다. 이 명령은 DHCP 정보 변경을 자동화하기 위해 만드는 스크립트에 가장 유용합니다. <a href="#">dhcptab</a> 테이블에서 특정 옵션 값을 검색하는 가장 빠른 방법은 <a href="#">dhtadm</a> 을 -p 옵션과 함께 사용하고 <a href="#">grep</a> 명령을 통해 출력 결과를 파이프로 연결하는 것입니다. 자세한 내용은 <a href="#">dhtadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/ipadm	시스템 부트 시 IP 주소를 네트워크 인터페이스에 지정하거나 네트워크 인터페이스 매개변수를 구성하기 위해(또는 둘 다) 사용됩니다. DHCP 클라이언트에서 <a href="#">ipadm</a> 이 DHCP를 시작하여 네트워크 인터페이스를 구성하는 데 필요한 매개변수(IP 주소 포함)를 얻습니다. 자세한 내용은 <a href="#">ipadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/omshell	ISC DHCP 전용: OMAPI(Object Management API)를 사용하여 ISC DHCP 서버의 상태를 질의하고 변경하는 방법을 제공합니다. 자세한 내용은 <a href="#">omshell(1)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/pntadm	레거시 Sun DHCP 전용: 클라이언트 ID를 IP 주소에 매핑하는 DHCP 네트워크 테이블을 변경하고, 선택적으로 구성 정보를 IP 주소와 연관시키는 데 사용됩니다. 자세한 내용은 <a href="#">pntadm(1M)</a> 매뉴얼 페이지를 참조하십시오.
/usr/sbin/snoop	네트워크에서 전달되는 패킷의 내용을 캡처하고 표시하는 데 사용됩니다. <a href="#">snoop</a> 은 DHCP 서비스 관련 문제를 해결하는 데 유용합니다. 자세한 내용은 <a href="#">snoop(1M)</a> 매뉴얼 페이지를 참조하십시오.

## DHCP 서비스에서 사용된 파일

다음 표는 DHCP와 연관된 파일을 나열합니다.

표 13-2 DHCP 데몬 및 명령에서 사용된 파일 및 테이블

파일 또는 테이블 이름	설명
dhcptab	레거시 Sun DHCP 전용: 옵션과 함께 지정된 값으로 기록된(이후 매크로로 그룹화됨) DHCP 구성 정보의 테이블을 지칭하는 일반 용어입니다. dhcptab 테이블의 이름과 위치는 DHCP 정보에 사용할 데이터 저장소에 의해 결정됩니다. 자세한 내용은 <a href="#">dhcptab(4)</a> 매뉴얼 페이지를 참조하십시오.
DHCP 네트워크 테이블	레거시 Sun DHCP 전용: IP 주소를 클라이언트 ID 및 구성 옵션에 매핑합니다. 네트워크의 IP 주소(예: 10.21.32.0)에 따라 DHCP 네트워크 테이블 이름이 지정됩니다. dhcp_network라는 파일이 없습니다. DHCP 네트워크 테이블의 이름과 위치는 DHCP 정보에 사용할 데이터 저장소에 의해 결정됩니다. 자세한 내용은 <a href="#">dhcp_network(4)</a> 매뉴얼 페이지를 참조하십시오.
/etc/dhcp/eventhook	레거시 Sun DHCP 전용: dhcpagent 데몬이 자동으로 실행할 수 있는 스크립트 또는 실행 파일입니다. 자세한 내용은 <a href="#">dhcpagent(1M)</a> 매뉴얼 페이지를 참조하십시오.
/etc/inet/dhcpd4.conf /etc/inet/dhcpd6.conf	ISC DHCP 전용: ISC DHCP 서버 dhcpd에 대한 구성 정보를 포함합니다. 자세한 내용은 dhcpd.conf(5) 매뉴얼 페이지를 참조하십시오.
/etc/inet/dhcpsvc.conf	레거시 Sun DHCP 전용: DHCP 데몬의 시작 옵션 및 데이터 저장소 정보를 저장합니다. 이 파일은 수동으로 편집하면 안 됩니다. dhcpconfig 명령을 사용하여 시작 옵션을 변경합니다. 자세한 내용은 <a href="#">dhcpsvc.conf(4)</a> 매뉴얼 페이지를 참조하십시오.
/etc/dhcp/interface.dhc /etc/dhcp/interface.dh6	제공된 네트워크 인터페이스에 대해 DHCP에서 얻은 구성 매개변수를 포함합니다. DHCPv4의 경우 파일 이름이 dhc로 끝납니다. DHCPv6의 경우 파일 이름이 dh6으로 끝납니다. 인터페이스의 IP 주소 임대를 삭제할 때 /etc/dhcp/interface.dhc에 현재 구성 정보를 캐싱합니다. 예를 들어, DHCP가 <code>qe0</code> 인터페이스에 사용된 경우 dhcpagent가 /etc/dhcp/qe0.dhc에 구성 정보를 캐싱합니다. 다음에 인터페이스에서 DHCP를 시작할 때 임대가 만료되지 않았을 경우 클라이언트가 캐시된 구성을 사용하도록 요청합니다. DHCP 서버가 요청을 거부하면 클라이언트가 DHCP 임대 협상의 표준 프로세스를 시작합니다.
/etc/default/dhcpagent	dhcpagent 클라이언트 데몬에 대한 매개변수 값을 설정합니다. 매개변수에 대한 자세한 내용은 /etc/default/dhcpagent 파일 또는 <a href="#">dhcpagent(1M)</a> 매뉴얼 페이지를 참조하십시오.

표 13-2 DHCP 데몬 및 명령에서 사용된 파일 및 테이블 (계속)

파일 또는 테이블 이름	설명
/etc/dhcp/inittab /etc/dhcp/inittab6	<p>레거시 Sun DHCP 전용: 데이터 유형과 같은 DHCP 옵션 코드의 여러 측면을 정의하고 니모닉 레이블을 지정합니다. 파일 구문에 대한 자세한 내용은 <a href="#">dhcp_inittab(4)</a> 매뉴얼 페이지를 참조하십시오. /etc/dhcp/inittab6은 DHCPv6 클라이언트에서 사용됩니다.</p> <p>클라이언트에서 /etc/dhcp/inittab 파일의 정보를 dhcpinfo 명령에서 사용하여 정보 구독자에게 보다 의미있는 정보를 제공합니다. DHCP 서버 시스템에서 이 파일을 DHCP 데몬 및 관리 도구에서 사용하여 DHCP 옵션 정보를 얻습니다.</p> <p>/etc/dhcp/inittab 파일은 이전 릴리스에서 사용된 /etc/dhcp/dhcptags 파일을 대체합니다.</p>
/var/db/isc-dhcp/dhcp4.leases /var/db/isc-dhcp/dhcp4.leases- /var/db/isc-dhcp/dhcp6.leases /var/db/isc-dhcp/dhcp6.leases-	ISC DHCP 전용: DHCPv4 및 DHCPv6 서버의 임대를 나열합니다. 파일 이름 끝에 "-"가 붙은 파일은 이전 복사본입니다.

## DHCP 서비스에서 사용된 SMF 서비스

다음 표는 DHCP와 연관된 SMF 서비스를 나열합니다.

표 13-3 DHCP 데몬 및 명령에서 사용된 SMF 서비스

SMF 서비스 이름	설명
svc:/network/dhcp-server:default	레거시 Sun DHCP 서비스에 대한 정보를 포함합니다.
svc:/network/dhcp/server:ipv4 svc:/network/dhcp/server:ipv6	ISC DHCP 서비스에 대한 정보를 포함합니다.
svc:/network/dhcp/relay:ipv4 svc:/network/dhcp/relay:ipv6	DHCP 또는 BOOTP 요청을 원격 ISC DHCP 서버로 중계할 수 있는 서비스의 정보를 포함합니다.
svc:/network/dns/client	DNS 질의를 분석하는 데 사용된 정보를 포함합니다. DHCP 서버 구성 중, 이 SMF 서비스를 참조하여 DNS 도메인 및 DNS 서버에 대한 정보를 찾을 수 있습니다.
svc:/system/name-service/switch	이름 서비스 데이터베이스의 위치와 이름 서비스가 다양한 종류의 정보를 검색하는 순서를 지정합니다. 이 서비스는 DHCP 서비스를 구성할 때 정확한 구성 정보를 제공합니다.

## 제 3 부

# IP 보안

이 절에서는 네트워크 보안을 중점적으로 다룹니다. IPsec(IP security architecture)는 패킷 레벨에서 네트워크를 보호합니다. IKE(Internet key management)는 IPsec에 대한 키를 관리합니다. Oracle Solaris의 IP 필터 기능은 방화벽을 제공합니다.



## IP 보안 아키텍처(개요)

---

IPsec(IP Security Architecture)는 IPv4 및 IPv6 네트워크 패킷에서 IP 데이터그램에 대한 암호화 보호를 제공합니다.

이 장은 다음 정보를 포함합니다.

- 199 페이지 “IPsec 소개”
- 202 페이지 “IPsec 패킷 흐름”
- 205 페이지 “IPsec 보안 연결”
- 206 페이지 “IPsec 보호 방식”
- 209 페이지 “IPsec 보호 정책”
- 209 페이지 “IPsec의 전송 및 터널 모드”
- 211 페이지 “VPN(Virtual Private Networks) 및 IPsec”
- 212 페이지 “IPsec 및 NAT 순회”
- 213 페이지 “IPsec 및 SCTP”
- 213 페이지 “IPsec 및 Oracle Solaris 영역”
- 213 페이지 “IPsec 및 논리적 도메인”
- 214 페이지 “IPsec 유틸리티 및 파일”

네트워크에서 IPsec를 구현하려면 15 장, “IPsec 구성(작업)”을 참조하십시오. 참조 정보는 16 장, “IP 보안 아키텍처(참조)”를 참조하십시오.

## IPsec 소개

IPsec는 패킷을 인증하거나 패킷을 암호화하거나 둘 다 수행하여 IP 패킷을 보호합니다. IPsec는 IP 모듈 내에서 수행됩니다. 따라서 인터넷 응용 프로그램에서는 IPsec를 사용하도록 구성할 필요 없이 IPsec를 활용할 수 있습니다. 제대로 사용되면 IPsec는 네트워크 트래픽을 보호하는 효과적인 도구가 될 수 있습니다.

IPsec 보호에는 다음 주요 구성 요소가 관련됩니다.

- **보안 프로토콜** - IP 데이터그램 보호 방식입니다. AH(인증 헤더)는 IP 패킷의 해시를 포함하고 무결성을 보장합니다. 데이터그램의 콘텐츠는 암호화되지 않지만, 수신자에게 패킷 콘텐츠가 변경되지 않았음을 보장합니다. 또한 패킷이 발신자에 의해 보내졌음을 수신자에게 보장합니다. ESP(보안 페이로드 캡슐화)는 IP 데이터를 암호화하므로 패킷 전송 중 콘텐츠를 숨깁니다. 또한 ESP는 인증 알고리즘 옵션을 통해 데이터 무결성을 보장할 수 있습니다.
- **SA(보안 연결)** - 네트워크 트래픽의 특정 흐름에 적용되는 암호화 매개변수 및 IP 보안 프로토콜입니다. 각 SA는 SPI(Security Parameters Index)라는 고유한 참조를 가집니다.
- **SADB(보안 연결 데이터베이스)** - 보안 프로토콜과 IP 대상 주소 및 색인화 번호를 연결하는 데이터베이스입니다. 색인화 번호는 SPI(보안 매개변수 색인)라고 합니다. 이러한 세 가지 요소(보안 프로토콜, 대상 주소 및 SPI)는 적절한 IPsec 패킷을 고유하게 식별합니다. 데이터베이스는 패킷 대상에 도달하는 보호된 패킷을 수신자가 인식할 수 있도록 합니다. 또한 수신자는 데이터베이스의 정보를 사용하여 통신을 해독하고, 패킷이 변경되지 않았음을 확인하며, 패킷을 재어셈블하고, 패킷을 최종 대상에 전달합니다.
- **키 관리** - 암호화 알고리즘 및 SPI에 대한 키 생성 및 배포입니다.
- **보안 방식** - IP 데이터그램에서 데이터를 보호하는 인증 및 암호화 알고리즘입니다.
- **SPD(보안 정책 데이터베이스)** - 패킷에 적용되는 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷이 어떻게 처리되어야 하는지 결정합니다. 패킷은 폐기할 수 있습니다. 패킷은 투명하게 전달할 수 있습니다. 또는 패킷은 IPsec로 보호할 수 있습니다. 아웃바운드 패킷에 대해 SPD 및 SADB는 적용할 보호 레벨을 결정합니다. 인바운드 패킷에 대해 SPD는 패킷에 대한 보호 레벨이 합당한지 여부를 결정하는 데 도움을 줍니다. 패킷이 IPsec로 보호되는 경우 패킷을 해독하고 확인한 후 SPD를 참조합니다.

IPsec는 IP 대상 주소로 이동하는 IP 데이터그램에 보안 방식을 적용합니다. 수신자는 SADB의 정보를 사용하여 도달한 패킷이 적절한지 확인하고 해독합니다. 응용 프로그램에서는 IPsec를 호출하여 소켓별 레벨에서도 IP 데이터그램에 보안 방식을 적용할 수 있습니다.

포트의 소켓이 연결되고 나중에 해당 포트에 IPsec 정책이 적용될 경우 해당 소켓을 사용하는 트래픽은 IPsec로 보호되지 않습니다. 물론, IPsec 정책이 포트에 적용된 이후 포트에서 열린 소켓은 IPsec 정책으로 보호됩니다.



## IPsec RFC

IETF(Internet Engineering Task Force)는 IP 계층에 대한 보안 아키텍처를 설명하는 여러 RFC(Requests for Comment)를 게시했습니다. 모든 RFC는 Internet Society에 의해 암호화됩니다. RFC에 대한 링크는 <http://www.ietf.org/>를 참조하십시오. 다음 RFC 목록은 일반적인 IP 보안 참조를 다룹니다.

- RFC 2411, “IP Security Document Roadmap,” 1998년 11월
- RFC 2401, “Security Architecture for the Internet Protocol,” 1998년 11월
- RFC 2402, “IP Authentication Header,” 1998년 11월
- RFC 2406, “IP Encapsulating Security Payload (ESP),” 1998년 11월
- RFC 2408, “Internet Security Association and Key Management Protocol (ISAKMP),” 1998년 11월
- RFC 2407, “The Internet IP Security Domain of Interpretation for ISAKMP,” 1998년 11월
- RFC 2409, “The Internet Key Exchange (IKE),” 1998년 11월
- RFC 3554, “On the Use of Stream Control Transmission Protocol (SCTP) with IPsec,” 2003년 7월

## IPsec 용어

IPsec RFC는 시스템에서 IPsec를 구현할 때 알아두면 유용한 많은 용어를 정의합니다. 다음 표에서는 IPsec 용어 및 일반적으로 사용되는 약어를 나열하고 각 용어를 정의합니다. 키 협상에서 사용되는 용어 목록은 표 17-1을 참조하십시오.

표 14-1 IPsec 용어, 약어 및 사용

IPsec 용어	머리글자어	정의
보안 연결	SA	네트워크 트래픽의 특정 흐름에 적용되는 암호화 매개변수 및 IP 보안 프로토콜입니다. SA는 보안 프로토콜, 고유한 SPI(보안 매개변수 색인), IP 대상, 이렇게 3중으로 정의됩니다.
보안 연결 데이터베이스	SADB	모든 활성 보안 연결을 포함하는 데이터베이스입니다.
보안 매개변수 색인	SPI	보안 연결에 대한 색인화 값입니다. SPI는 동일한 IP 대상 및 보안 프로토콜을 가지는 SA 사이에서 구분되는 32비트 값입니다.
보안 정책 데이터베이스	SPD	아웃바운드 패킷 및 인바운드 패킷이 지정된 보호 레벨을 가지는지 여부를 결정하는 데이터베이스입니다.
키 교환		비대칭 암호화 알고리즘을 사용하여 키를 생성하는 프로세스입니다. 두 가지 주요 방식은 RSA 및 Diffie-Hellman입니다.

표 14-1 IPsec 용어, 약어 및 사용 (계속)

IPsec 용어	머리글자어	정의
Diffie-Hellman	DH	키 생성 및 키 인증을 허용하는 키 교환 알고리즘입니다. <b>인증된 키 교환</b> 이라고도 합니다.
RSA	RSA	키 생성 및 키 배포를 허용하는 키 교환 알고리즘입니다. 프로토콜 이름은 Rivest, Shamir, Adleman 등 3인의 저작자 이름에서 따왔습니다.
인터넷 보안 연결 및 키 관리 프로토콜	ISAKMP	SA 속성 형식 설정과 SA 협상, 수정 및 삭제를 위한 공통 프레임워크입니다. ISAKMP는 IKE 교환 처리를 위한 IETF 표준입니다.

## IPsec 패킷 흐름

그림 14-1은 IPsec가 아웃바운드 패킷에서 호출될 때 IP 주소 지정된 패킷이 **IP 데이터그램**의 일부로 진행되는지 보여줍니다. 흐름 다이어그램은 AH(authentication header) 및 ESP(encapsulating security payload) 엔티티를 어디에서 패킷에 적용할 수 있는지 보여줍니다. 이러한 엔티티를 적용하는 방법 및 알고리즘을 선택하는 방법은 다음 절에서 설명합니다.

그림 14-2는 IPsec 인바운드 프로세스를 보여 줍니다.

그림 14-1 아웃바운드 패킷 프로세스에 적용된 IPsec

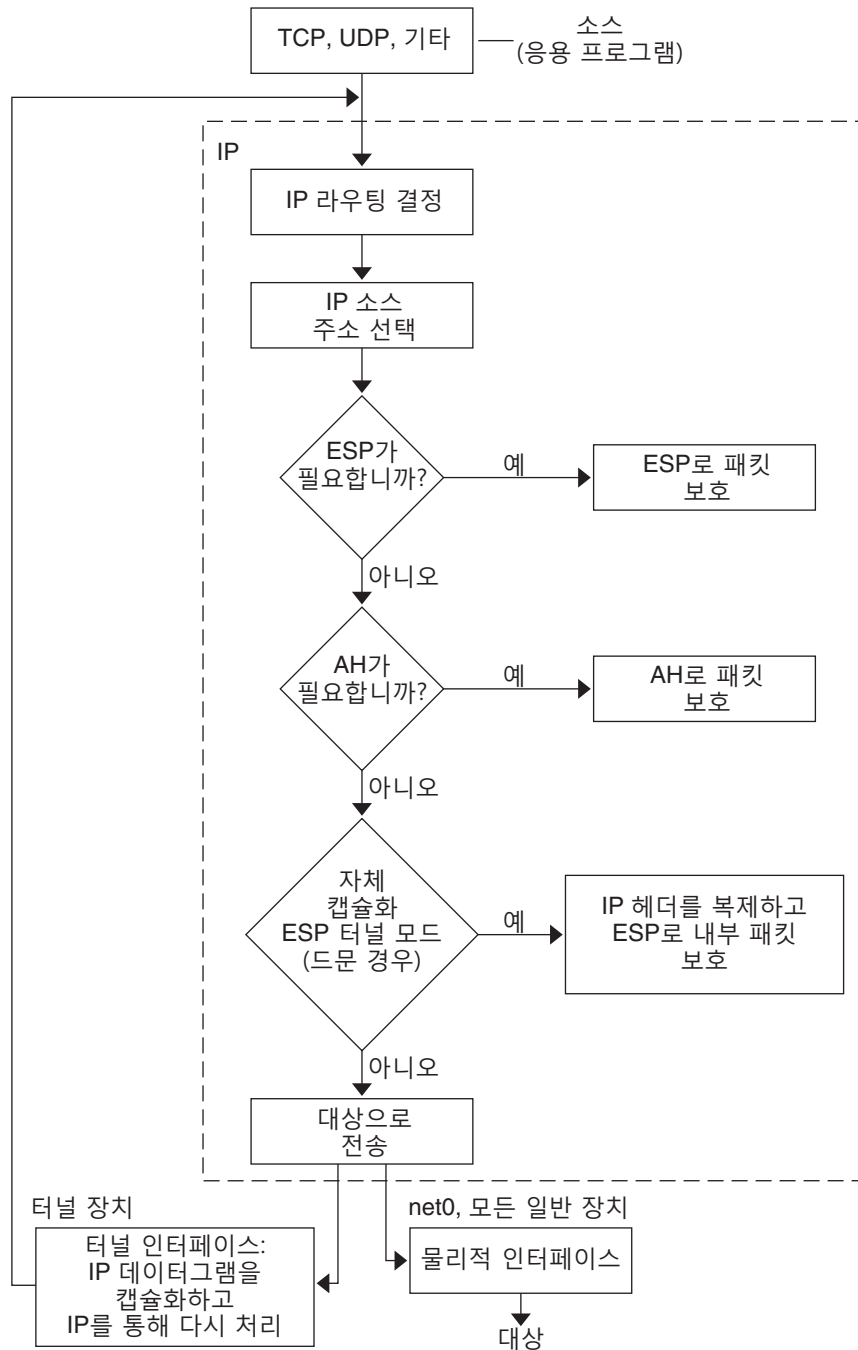
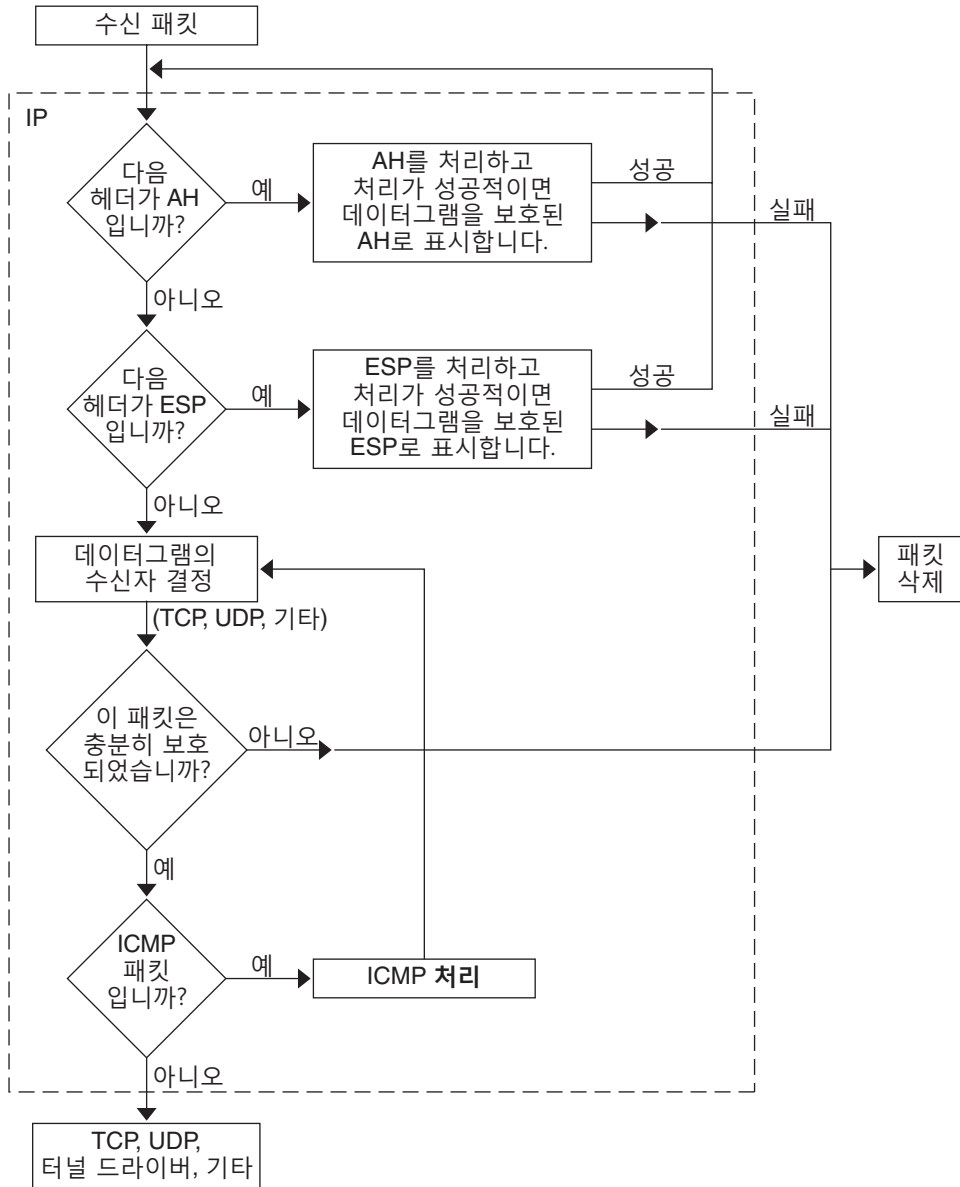


그림 14-2 인바운드 패킷 프로세스에 적용된 IPsec



## IPsec 보안 연결

IPsec SA(보안 연결)는 통신 호스트에서 인식할 수 있는 보안 등록 정보를 지정합니다. 단일 SA는 한 방향의 데이터를 보호합니다. 단일 호스트 또는 그룹(멀티캐스트) 주소에 대한 보호입니다. 대부분의 통신은 피어 투 피어 또는 클라이언트-서버이므로 양방향에서 트래픽을 보호하려면 두 SA가 존재해야 합니다.

다음 세 가지 요소는 IPsec SA를 고유하게 식별합니다.

- 보안 프로토콜(AH 또는 ESP)
- 대상 IP 주소
- SPI(보안 매개변수 색인)

임의의 32비트 값인 SPI는 AH 또는 ESP 패킷으로 전송됩니다. [ipsecah\(7P\)](#) 및 [ipsecesp\(7P\)](#) 매뉴얼 페이지에서 AH 및 ESP가 제공하는 보호의 범위를 설명합니다. 무결성 체크섬 값은 패킷을 인증하는 데 사용됩니다. 인증을 실패할 경우 패킷은 삭제됩니다.

보안 연결은 SADB(보안 연결 데이터베이스)에 저장됩니다. 소켓 기반 관리 인터페이스인 PF\_KEY는 권한이 부여된 응용 프로그램이 데이터베이스를 관리하도록 합니다. 예를 들어, IKE 응용 프로그램 및 `ipseckey` 명령은 PF\_KEY 소켓 인터페이스를 사용합니다.

- IPsec SADB에 대한 자세한 설명은 [240 페이지 “IPsec에 대한 보안 연결 데이터베이스”](#)를 참조하십시오.
- SADB 관리 방법에 대한 자세한 내용은 [pf\\_key\(7P\)](#) 매뉴얼 페이지를 참조하십시오.

## IPsec에서 키 관리

SA(보안 연결)에는 인증 및 암호화를 위한 키 입력 자료가 필요합니다. 이 키 입력 자료 관리를 키 관리라고 합니다. IKE(Internet Key Exchange) 프로토콜은 키 관리를 자동으로 처리합니다. 또한 `ipseckey` 명령을 사용하여 수동으로 키를 관리할 수 있습니다.

IPv4 및 IPv6 소켓에 대한 SA에서는 이러한 두 가지 키 관리 방식을 사용할 수 있습니다. 수동 키 관리를 사용해야 하는 분명한 이유가 없다면 IKE를 사용하는 것이 좋습니다.

Oracle Solaris의 SMF(서비스 관리 기능) 기능은 IPsec에 대한 다음 키 관리 서비스를 제공합니다.

- `svc:/network/ipsec/ike:default` 서비스 - 자동 키 관리를 위한 SMF 서비스입니다. `ike` 서비스는 `in.iked` 데몬을 실행하여 자동 키 관리를 제공합니다. IKE에 대한 설명은 17 장, “[Internet Key Exchange\(개요\)](#)”을 참조하십시오. `in.iked` 데몬에 대한 자세한 내용은 `in.iked(1M)` 매뉴얼 페이지를 참조하십시오. `ike` 서비스에 대한 자세한 내용은 281 페이지 “[IKE 서비스](#)”를 참조하십시오.
- `svc:/network/ipsec/manual-key:default` 서비스 - 수동 키 관리를 위한 SMF 서비스입니다. `manual-key` 서비스는 `ipseckey` 명령을 다양한 옵션과 함께 실행하여 키를 수동으로 관리합니다. `ipseckey` 명령에 대한 설명은 241 페이지 “[IPsec에서 SA 생성을 위한 유틸리티](#)”를 참조하십시오. `ipseckey` 명령 옵션에 대한 자세한 설명은 `ipseckey(1M)` 매뉴얼 페이지를 참조하십시오.

## IPsec 보호 방식

IPsec는 데이터 보호를 위한 두 가지 보안 프로토콜을 제공합니다.

- AH(Authentication Header)
- ESP(Encapsulating Security Payload)

AH는 인증 알고리즘으로 데이터를 보호합니다. ESP는 암호화 알고리즘으로 데이터를 보호합니다. ESP는 인증 방식과 함께 사용할 수 있으며 그렇게 사용해야 합니다. NAT를 통과하지 않는 경우 ESP와 AH를 결합할 수 있습니다. 그렇지 않은 경우 인증 알고리즘 및 암호화 방식을 ESP와 함께 사용할 수 있습니다. 결합된 모드 알고리즘(예: AES-GCM)은 단일 알고리즘 내에서 암호화와 인증을 제공합니다.

### AH(Authentication Header)

**인증 헤더**는 IP 데이터그램에 데이터 인증, 강력한 무결성 및 재생 보호 기능을 제공합니다. AH는 IP 데이터그램의 많은 부분을 보호합니다. 다음 그림에 나온 대로 AH는 IP 헤더와 전송 헤더 사이에 삽입됩니다.

IP 헤더	AH	TCP 헤더	
-------	----	--------	--

전송 헤더는 TCP, UDP, SCTP 또는 ICMP가 될 수 있습니다. **터널**이 사용되는 경우 전송 헤더는 다른 IP 헤더가 될 수 있습니다.

## ESP(Encapsulating Security Payload)

ESP(보안 페이로드 캡슐화) 모듈은 ESP가 캡슐화하는 콘텐츠에 대한 기밀성을 제공합니다. 또한 ESP는 AH가 제공하는 서비스도 제공합니다. 하지만 ESP는 ESP가 캡슐화하는 데이터그램의 부분에 대해서만 보호 기능을 제공합니다. ESP는 보호된 패킷의 무결성을 위해 선택적 인증 서비스를 제공합니다. ESP는 암호화 지원 기술을 사용하므로 ESP를 제공하는 시스템은 가져오기 및 내보내기 제어 규칙에 종속될 수 있습니다.

ESP는 데이터를 캡슐화하므로 ESP는 다음 그림에 나온 대로 데이터그램에서 시작 이후의 데이터만 보호합니다.



### ■ 암호화됨

TCP 패킷에서 ESP는 TCP 헤더 및 해당 데이터만 캡슐화합니다. 패킷이 IP-in-IP 데이터그램인 경우 ESP는 내부 IP 데이터그램을 보호합니다. 소켓별 정책에서는 **자체 캡슐화**를 허용하므로 ESP에서 필요할 때 ESP가 IP 옵션을 캡슐화할 수 있습니다.

자체 캡슐화가 설정되면 IP 헤더의 복사본이 IP-in-IP 데이터그램을 생성하게 됩니다. 예를 들어, 자체 캡슐화가 TCP 소켓에서 설정되지 않은 경우 데이터그램은 다음 형식으로 보내집니다.

[ IP(a -> b) options + TCP + data ]

자체 캡슐화가 TCP 소켓에서 설정된 경우 데이터그램은 다음 형식으로 보내집니다.

[ IP(a -> b) + ESP [ IP(a -> b) options + TCP + data ] ]

자세한 내용은 [209 페이지 “IPsec의 전송 및 터널 모드”](#)를 참조하십시오.

## AH 및 ESP를 사용할 때 보안 고려 사항

다음 표는 AH 및 ESP에서 제공하는 보호 기능을 비교한 것입니다.

표 14-2 IPsec에서 AH 및 ESP로 제공되는 보호 기능

프로토콜	패킷 범위	보호	공격 방어
AH	IP 헤더에서 전송 헤더까지 패킷을 보호합니다.	강력한 무결성, 데이터 인증을 제공합니다. <ul style="list-style-type: none"> <li>■ 발신자가 보낸 콘텐츠를 그대로 수신자가 수신할 수 있도록 합니다.</li> <li>■ AH에서 재생 보호를 사용으로 설정하지 않을 경우 재생 공격에 취약합니다.</li> </ul>	재생, 잘라내기 및 붙여넣기
ESP	데이터그램에서 ESP 시작 이후의 패킷을 보호합니다.	암호화 옵션을 사용하여 IP 페이로드를 암호화합니다. 기밀성을 유지합니다.  인증 옵션을 사용하여 AH와 동일한 페이로드 보호 기능을 제공합니다.  두 옵션을 모두 사용하면 강력한 무결성, 데이터 인증 및 기밀성을 제공할 수 있습니다.	도청  재생, 잘라내기 및 붙여넣기  재생, 잘라내기 및 붙여넣기, 도청

## IPsec의 인증 및 암호화 알고리즘

IPsec 보안 프로토콜에서는 인증 및 암호화의 두 가지 알고리즘 유형을 사용합니다. AH 모듈은 인증 알고리즘을 사용합니다. ESP 모듈은 인증 알고리즘과 함께 암호화를 사용할 수 있습니다. 시스템의 알고리즘 및 해당 등록 정보 목록은 `ipsecalg` 명령을 사용하여 얻을 수 있습니다. 자세한 내용은 [ipsecalg\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 또한 [getipseccalgbynname\(3NSL\)](#) 매뉴얼 페이지에 설명된 기능을 사용하여 알고리즘의 등록 정보를 검색할 수 있습니다.

IPsec는 Oracle Solaris의 암호화 프레임워크 기능을 사용하여 알고리즘에 액세스합니다. 암호화 프레임워크는 다른 서비스와 함께 알고리즘에 대한 중앙 저장소를 제공합니다. 프레임워크를 통해 IPsec는 높은 성능의 암호화 하드웨어 가속기를 활용할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- **Oracle Solaris 관리: 보안 서비스의 11 장, “암호화 프레임워크(개요)”**
- **Developer’s Guide to Oracle Solaris 11 Security의 8 장, “Introduction to the Oracle Solaris Cryptographic Framework”**

## IPsec의 인증 알고리즘

인증 알고리즘은 데이터 및 키를 기반으로 하는 무결성 체크섬 값 또는 **다이제스트**를 생성합니다. AH 모듈은 인증 알고리즘을 사용합니다. ESP 모듈은 인증 알고리즘도 사용할 수 있습니다.

## IPsec의 암호화 알고리즘

암호화 알고리즘은 키로 데이터를 암호화합니다. IPsec의 ESP 모듈은 암호화 알고리즘을 사용합니다. 알고리즘은 **블록 크기** 단위로 데이터에 작동합니다.



## IPsec 보호 정책

IPsec 보호 정책에서는 모든 보안 방식을 사용할 수 있습니다. IPsec 정책은 다음 레벨에서 적용할 수 있습니다.

- 시스템 전역 레벨
- 소켓별 레벨

IPsec는 아웃바운드 데이터그램 및 인바운드 데이터그램에 시스템 전역 정책을 적용합니다. 아웃바운드 데이터그램은 보호 기능과 함께 또는 보호 기능 없이 보낼 수 있습니다. 보호 기능이 적용된 경우 알고리즘은 특정 또는 비특정입니다. 시스템에서 알고 있는 추가 데이터로 인해 아웃바운드 데이터그램에 추가 규칙을 적용할 수 있습니다. 인바운드 데이터그램은 수용하거나 삭제할 수 있습니다. 인바운드 데이터그램의 삭제 또는 수용 결정은 때때로 겹치거나 충돌하는 여러 조건을 기준으로 합니다. 충돌은 먼저 구문 분석된 규칙을 결정하여 해결됩니다. 트래픽이 모든 기타 정책을 우회해야 하는 정책 항목 상태일 때를 제외하고 트래픽은 자동으로 수용됩니다.

일반적으로 데이터그램을 보호하는 정책은 우회할 수 있습니다. 시스템 전역 정책에서 예외 사항을 지정하거나 소켓별 정책에서 우회를 요청할 수 있습니다. 시스템 내부 트래픽의 경우 정책이 적용되지만 실제 보안 방식은 적용되지 않습니다. 대신 시스템 간 패킷에 대한 아웃바운드 정책은 해당 방식이 적용된 인바운드 패킷으로 변환됩니다.

`ipsecinit.conf` 파일 및 `ipsecconf` 명령을 사용하여 IPsec 정책을 구성합니다. 자세한 내용 및 예는 [ipsecconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## IPsec의 전송 및 터널 모드

IPsec 표준에서는 **전송 모드** 및 **터널 모드**의 두 가지 고유 IPsec 작업 모드를 정의합니다. 모드는 패킷의 인코딩에 영향을 주지 않습니다. 패킷은 각 모드에서 AH, ESP 또는 둘 다로 보호됩니다. 모드는 내부 패킷이 IP 패킷일 때 정책 적용 면에서 다음과 같이 다릅니다.

- 전송 모드에서 외부 헤더는 내부 IP 패킷을 보호하는 IPsec 정책을 결정합니다.
- 터널 모드에서 내부 IP 패킷은 해당 콘텐츠를 보호하는 IPsec 정책을 결정합니다.

전송 모드에서 외부 헤더, 다음 헤더 및 다음 헤더가 지원하는 모든 포트는 IPsec 정책을 결정하는 데 사용될 수 있습니다. 실제로 IPsec는 두 IP 주소 사이에 서로 다른 전송 모드 정책을 적용하여 단일 포트를 세분화할 수 있습니다. 예를 들어, 다음 헤더가 포트를 지원하는 TCP인 경우 IPsec 정책을 외부 IP 주소의 TCP 정책에 대해 설정할 수 있습니다. 마찬가지로 다음 헤더가 IP 헤더인 경우 외부 헤더 및 내부 IP 헤더를 사용하여 IPsec 정책을 결정할 수 있습니다.

터널 모드는 IP-in-IP 데이터그램에 대해서만 작동합니다. 터널 모드의 터널링은 집에 있는 컴퓨터 작업자가 중앙 컴퓨터 위치에 연결할 때 유용할 수 있습니다. 터널 모드에서 IPsec 정책은 내부 IP 데이터그램의 콘텐츠에 적용됩니다. 서로 다른 내부 IP 주소에 대해

서로 다른 IPsec 정책을 적용할 수 있습니다. 즉, 내부 IP 헤더, 다음 헤더 및 다음 헤더가 지원하는 포트가 정책을 적용할 수 있습니다. 전송 모드와 달리 터널 모드에서는 외부 IP 헤더가 내부 IP 데이터그램의 정책을 결정하지 않습니다.

따라서 터널 모드에서 IPsec 정책은 라우터 뒤의 LAN 서브넷 및 이러한 서브넷의 포트에 대해 지정할 수 있습니다. 또한 IPsec 정책은 이러한 서브넷에 있는 특정 IP 주소(즉, 호스트)에 대해 지정할 수도 있습니다. 이러한 호스트의 포트도 특정 IPsec 정책을 가질 수 있습니다. 하지만 동적 경로 지정 프로토콜이 터널을 통해 실행되는 경우 피어 네트워크의 네트워크 토폴로지에 대한 뷰가 변경될 수 있으므로 서브넷 선택이나 주소 선택을 사용하지 마십시오. 변경되면 정적 IPsec 정책이 무효화됩니다. 정적 경로 구성을 포함하는 터널링 절차의 예는 [223 페이지 “IPsec를 사용하여 VPN 보호”](#)를 참조하십시오.

Oracle Solaris에서 터널 모드는 IP 터널링 네트워크 인터페이스에만 적용할 수 있습니다. 터널링 인터페이스에 대한 자세한 내용은 [6 장, “IP 터널 구성”](#)을 참조하십시오. `ipsecconf` 명령은 IP 터널링 네트워크 인터페이스를 선택하기 위한 `tunnel` 키워드를 제공합니다. `tunnel` 키워드가 규칙에 존재하는 경우 해당 규칙에서 지정된 모든 선택기가 내부 패킷에 적용됩니다.

전송 모드에서는 ESP, AH 또는 둘 다 데이터그램을 보호할 수 있습니다.

다음 그림은 보호되지 않는 TCP 패킷의 IP 헤더를 보여줍니다.

그림 14-3 TCP 정보를 전달하는 보호되지 않는 IP 패킷



전송 모드에서 ESP가 다음 그림에 나온 대로 데이터를 보호합니다. 음영 영역은 패킷의 암호화된 부분을 나타냅니다.

그림 14-4 TCP 정보를 전달하는 보호된 IP 패킷



■ 암호화됨

전송 모드에서 AH가 다음 그림에 나온 대로 데이터를 보호합니다.

그림 14-5 인증 헤더로 보호된 패킷

IP 헤더	AH	TCP 헤더	
-------	----	--------	--

AH 보호는 전송 모드라도 IP 헤더의 대부분을 포함합니다.

터널 모드에서 전체 데이터그램은 IPsec 헤더의 보호 **내부**에 있습니다. [그림 14-3](#)의 데이터그램은 다음 그림에 나온 대로 외부 IPsec 헤더(이 경우 ESP)로 터널 모드에서 보호됩니다.

그림 14-6 터널 모드에서 보호된 IPsec 패킷

IP 헤더	ESP	IP 헤더	TCP 헤더
-------	-----	-------	--------

#### ■ 암호화됨

ipsecconf 명령에는 터널을 터널 모드 또는 전송 모드로 설정하는 키워드가 포함되어 있습니다.

- 소켓별 정책에 대한 자세한 내용은 [ipsec\(7P\)](#) 매뉴얼 페이지를 참조하십시오.
- 소켓별 정책의 예는 [221 페이지](#) “IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법”을 참조하십시오.
- 터널에 대한 자세한 내용은 [ipsecconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- 터널 구성의 예는 [227 페이지](#) “터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법”을 참조하십시오.

## VPN(Virtual Private Networks) 및 IPsec

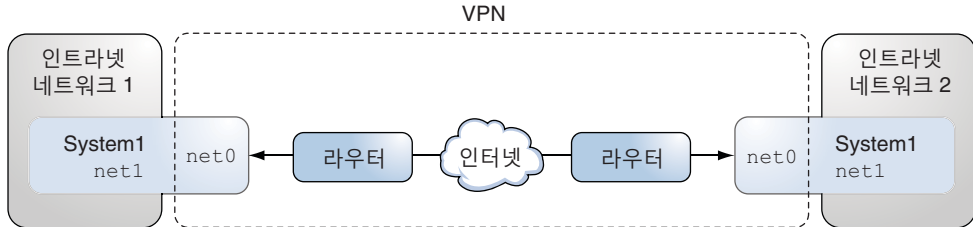
구성된 터널은 지점간 인터페이스입니다. 터널을 통해 한 IP 패킷을 다른 IP 패킷 내부에 캡슐화할 수 있습니다. 올바르게 구성된 터널에는 터널 소스와 터널 대상이 필요합니다. 자세한 내용은 [120 페이지](#) “IP 터널을 만들고 구성하는 방법”을 참조하십시오.

터널은 IP에 대한 분명한 **물리적 인터페이스**를 만듭니다. 물리적 링크의 무결성은 기본 보안 프로토콜에 의존합니다. SA(보안 연결)를 안전하게 설정할 경우 터널을 신뢰할 수 있습니다. 터널에서 나온 패킷은 터널 대상에 지정된 피어로부터 나왔어야 합니다. 이 신뢰가 존재할 경우 인터페이스별 IP 전달을 사용하여 **VPN(가상 사설망)**을 만들 수 있습니다.

IPsec 보호를 VPN에 추가할 수 있습니다. IPsec는 연결을 보호합니다. 예를 들어, VPN 기술을 사용하여 별도 네트워크의 사무실을 연결하는 조직에서는 IPsec를 추가하여 두 사무실 사이의 트래픽을 보호할 수 있습니다.

다음 그림은 IPsec를 사용하는 VPN의 두 사무실이 해당 네트워크 시스템에서 어떻게 배치되었는지 보여줍니다.

그림 14-7 VPN(Virtual Private Networks)



설정 절차의 자세한 예는 227 페이지 “터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법”을 참조하십시오.

## IPsec 및 NAT 순회

IKE는 NAT 장치에 걸쳐 IPsec SA를 협상할 수 있습니다. 시스템이 NAT 장치 뒤에 있더라도 이 기능을 통해 시스템은 원격 네트워크에서 안전하게 연결할 수 있습니다. 예를 들어, 집에서 작업하거나 회의실에서 로그인하는 직원은 IPsec를 사용하여 트래픽을 보호할 수 있습니다.

NAT는 네트워크 주소 변환(network address translation)을 나타냅니다. NAT 장치를 사용하여 개인 내부 주소를 고유한 인터넷 주소로 변환할 수 있습니다. NAT는 호텔과 같은 인터넷 공용 액세스 지점에서 매우 일반적입니다. 자세한 내용은 295 페이지 “IP 필터의 NAT 기능 사용”을 참조하십시오.

NAT 장치가 통신 시스템 사이에 있을 때 IKE를 사용하는 기능을 NAT 순회 또는 NAT-T라고 합니다. NAT-T에는 다음 제한 사항이 있습니다.

- AH 프로토콜은 변경되지 않는 IP 헤더에 의존하므로 AH는 NAT-T와 함께 작동할 수 없습니다. ESP 프로토콜은 NAT-T와 함께 사용됩니다.
- NAT 장치는 특수 처리 규칙을 사용하지 않습니다. 특수한 IPsec 처리 규칙을 사용하는 NAT 장치는 NAT-T의 구현에 방해가 될 수 있습니다.
- NAT-T는 IKE 개시자가 NAT 장치의 뒤에 있는 시스템일 때만 작동합니다. 장치가 IKE 패킷을 장치 뒤의 해당 개별 시스템에 전달하도록 프로그래밍되지 않은 경우 IKE 응답자는 NAT 장치 뒤에 있을 수 없습니다.

다음 RFC는 NAT 기능 및 NAT-T의 제한 사항을 설명합니다. RFC의 사본은 <http://www.rfc-editor.org>에서 검색할 수 있습니다.

- RFC 3022, “Traditional IP Network Address Translator (Traditional NAT),” 2001년 1월

- RFC 3715, “IPsec-Network Address Translation (NAT) Compatibility Requirements,” 2004년 3월
- RFC 3947, “Negotiation of NAT-Traversal in the IKE,” 2005년 1월
- RFC 3948, “UDP Encapsulation of IPsec Packets,” 2005년 1월

NAT에 걸쳐 IPsec를 사용하려면 272 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”을 참조하십시오.

## IPsec 및 SCTP

Oracle Solaris는 SCTP(Streams Control Transmission Protocol)를 지원합니다. IPsec 정책 지정에 위한 SCTP 프로토콜 및 SCTP 포트 번호 사용은 지원되지만 안정적이지는 않습니다. RFC 3554에 지정된 SCTP에 대한 IPsec 확장 기능은 아직 구현되지 않았습니다. 이러한 제한 사항으로 인해 SCTP에 대한 IPsec 정책을 만들 때 복잡해질 수 있습니다.

SCTP는 단일 SCTP 연결 컨텍스트에서 여러 소스 및 대상 주소를 활용할 수 있습니다. IPsec 정책이 단일 소스나 단일 대상 주소에 적용된 경우 SCTP가 해당 연결의 소스나 대상 주소를 바꾸면 통신이 실패합니다. IPsec 정책은 원래 주소만 인식할 수 있습니다. SCTP에 대한 자세한 내용은 RFC 및 [System Administration Guide: IP Services](#)의 “SCTP Protocol”을 참조하십시오.

## IPsec 및 Oracle Solaris 영역

공유 IP 영역의 경우, IPsec는 전역 영역에서 구성됩니다. IPsec 정책 구성 파일 `ipsecinit.conf`는 전역 영역에만 존재합니다. 파일에는 비전역 영역에 적용되는 항목과 전역 영역에 적용되는 항목이 있을 수 있습니다.

배타적 IP 영역의 경우, IPsec는 비전역 영역별로 구성됩니다.

영역에서 IPsec를 사용하는 방법에 대한 자세한 내용은 217 페이지 “IPsec를 사용하여 트래픽 보호”를 참조하십시오. 영역에 대한 자세한 내용은 [Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리](#)의 15 장, “Oracle Solaris Zones 소개”를 참조하십시오.

## IPsec 및 논리적 도메인

IPsec는 논리적 도메인에서 작동합니다. 논리적 도메인은 IPsec가 포함된 Oracle Solaris 버전(예: Oracle Solaris 10 릴리스)을 실행하고 있어야 합니다.

논리적 도메인을 만들려면 Oracle VM Server for SPARC(이전의 논리적 도메인)를 사용해야 합니다. 논리적 도메인 구성 방법에 대한 자세한 내용은 [Oracle VM Server for SPARC 2.1 Administration Guide](#) 또는 [Oracle VM Server for SPARC 2.0 Administration Guide](#)를 참조하십시오.

## IPsec 유틸리티 및 파일

표 14-3에서는 IPsec를 구성하고 관리하는 데 사용되는 파일, 명령 및 서비스 식별자를 설명합니다. 전체성을 위해 표에는 키 관리 파일, 소켓 인터페이스 및 명령도 포함되어 있습니다.

서비스 식별자에 대한 자세한 내용은 **Oracle Solaris 관리: 일반 작업의 6 장**, “서비스 관리(개요)”를 참조하십시오.

- 네트워크에서 IPsec 구현에 대한 지침은 217 페이지 “IPsec를 사용하여 트래픽 보호”를 참조하십시오.
- IPsec 유틸리티 및 파일에 대한 자세한 내용은 16 장, “IP 보안 아키텍처(참조)”를 참조하십시오.

표 14-3 일부 IPsec 유틸리티 및 파일 목록

IPsec 유틸리티, 파일 또는 서비스	설명	매뉴얼 페이지
svc:/network/ipsec/ipsecalg	IPsec 알고리즘을 관리하는 SMF 서비스입니다.	<a href="#">ipsecalgs(1M)</a>
svc:/network/ipsec/manual-key	키 입력 IPsec SA를 수동으로 관리하는 SMF 서비스입니다.	<a href="#">ipseckey(1M)</a>
svc:/network/ipsec/policy	IPsec 정책을 관리하는 SMF 서비스입니다.	<a href="#">smf(5)</a> , <a href="#">ipseconf(1M)</a>
svc:/network/ipsec/ike	IKE를 사용하여 IPsec SA의 자동 관리를 위한 SMF 서비스입니다.	<a href="#">smf(5)</a> , <a href="#">in.iked(1M)</a>
/etc/inet/ipsecinit.conf 파일	IPsec 정책 파일입니다.  SMF <b>policy</b> 서비스에서는 이 파일을 사용하여 시스템 부트 시 IPsec 정책을 구성합니다.	<a href="#">ipseconf(1M)</a>
ipseconf 명령	IPsec 정책 명령입니다. 현재 IPsec 정책을 보고 수정하며 테스트하는 데 유용합니다.  SMF <b>policy</b> 서비스에서 시스템 부트 시 IPsec 정책을 구성하는 데 사용됩니다.	<a href="#">ipseconf(1M)</a>
PF_KEY 소켓 인터페이스	SADB(보안 연결 데이터베이스)에 대한 인터페이스입니다. 수동 키 관리 및 자동 키 관리를 처리합니다.	<a href="#">pf_key(7P)</a>
ipseckey 명령	IPsec SA 키 입력 명령. <b>ipseckey</b> 는 PF_KEY 인터페이스에 대한 명령줄 프론트 엔드입니다. <b>ipseckey</b> 는 SA를 만들거나 삭제하거나 수정할 수 있습니다.	<a href="#">ipseckey(1M)</a>
/etc/inet/secret/ipseckeys 파일	수동으로 키를 입력한 SA가 포함됩니다.  SMF <b>manual-key</b> 서비스에서 시스템 부트 시 SA를 수동으로 구성하는 데 사용됩니다.	

표 14-3 일부 IPsec 유틸리티 및 파일 목록 (계속)

IPsec 유틸리티, 파일 또는 서비스	설명	매뉴얼 페이지
ipsecalgs 명령	IPsec 알고리즘 명령입니다. IPsec 알고리즘 및 해당 등록 정보 목록을 보고 수정하는 데 유용합니다.  SMF ipsecalgls 서비스에서 시스템 부트 시 알려진 IPsec 알고리즘을 커널과 동기화하는 데 사용됩니다.	<a href="#">ipsecalgs(1M)</a>
/etc/inet/ipsecalgls 파일	구성된 IPsec 프로토콜 및 알고리즘 정의를 포함합니다. 이 파일은 ipsecalgls 명령으로 관리되며 수동으로 편집하면 안 됩니다.	
/etc/inet/ike/config 파일	IKE 구성 및 정책 파일입니다. 기본적으로 이 파일은 존재하지 않습니다. 관리는 /etc/inet/ike/config 파일의 규칙 및 전역 매개변수를 기준으로 합니다. <a href="#">246 페이지 “IKE 유틸리티 및 파일”</a> 을 참조하십시오.  이 파일이 존재하는 경우 svc:/network/ipsec/ike 서비스가 IKE 데몬 in.iked를 시작하여 자동 키 관리를 제공합니다.	<a href="#">ike.config(4)</a>





## IPsec 구성(작업)

이 장에서는 네트워크에서 IPsec를 구현하기 위한 절차를 설명합니다. 관련 절차는 다음 절에서 설명합니다.

- 217 페이지 “IPsec를 사용하여 트래픽 보호”
- 223 페이지 “IPsec를 사용하여 VPN 보호”
- 230 페이지 “IPsec 및 IKE 관리”

IPsec에 대한 개요 정보는 14 장, “IP 보안 아키텍처(개요)”를 참조하십시오. IPsec에 대한 참조 정보는 16 장, “IP 보안 아키텍처(참조)”를 참조하십시오.

### IPsec를 사용하여 트래픽 보호

이 절에서는 두 시스템 간의 트래픽을 보호하고 웹 서버의 보안을 유지할 수 있는 절차를 제공합니다. VPN을 보호하려면 223 페이지 “IPsec를 사용하여 VPN 보호”를 참조하십시오. IPsec를 관리하고 IPsec 및 IKE에서 SMF 명령을 사용하는 추가 절차는 230 페이지 “IPsec 및 IKE 관리”를 참조하십시오.

다음 정보는 모든 IPsec 구성 작업에 적용됩니다.

- **IPsec 및 영역** – 공유 IP 비전역 영역에 대한 IPsec 정책 및 키를 관리하려면 전역 영역에서 IPsec 정책 파일을 만들고 전역 영역에서 IPsec 구성 명령을 실행합니다. 구성 중인 비보안 영역에 해당하는 소스 주소를 사용합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.
- **IPsec 및 RBAC** – 역할을 사용하여 IPsec를 관리하려면 **Oracle Solaris 관리: 보안 서비스의 9 장, “역할 기반 액세스 제어 사용(작업)”**을 참조하십시오. 예는 232 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”을 참조하십시오.
- **IPsec 및 SCTP** – IPsec는 SCTP(Streams Control Transmission Protocol) 연결을 보호하는데 사용할 수 있지만 주의해야 합니다. 자세한 내용은 213 페이지 “IPsec 및 SCTP”를 참조하십시오.

- **IPsec 및 Trusted Extensions 레이블** – Oracle Solaris의 Trusted Extensions 기능으로 구성된 시스템에서는 레이블을 IPsec 패킷에 추가할 수 있습니다. 자세한 내용은 [Trusted Extensions 구성 및 관리](#)의 “레이블이 있는 IPsec 관리”를 참조하십시오.
- **IPv4 및 IPv6 주소** – 이 설명서의 IPsec 예에서는 IPv4 주소를 사용합니다. Oracle Solaris에서는 IPv6 주소도 지원합니다. IPv6 네트워크에 대해 IPsec를 구성하려면 예에서 IPv6 주소를 대체하십시오. IPsec를 사용하여 터널을 보호하는 경우 내부 및 외부 주소에 대해 IPv4 및 IPv6 주소를 혼합할 수 있습니다. 예를 들어, 이러한 구성을 사용하면 IPv4 네트워크를 통해 IPv6을 터널링할 수 있습니다.

다음 작업 맵에서는 하나 이상의 시스템 사이에 IPsec를 설정하는 절차를 안내합니다. [ipsecconf\(1M\)](#), [ipseckey\(1M\)](#) 및 [ipadm\(1M\)](#) 매뉴얼 페이지에서도 각 예제 절에서 유용한 절차를 설명합니다.

작업	설명	수행 방법
두 시스템 사이의 트래픽을 보호합니다.	한 시스템에서 다른 시스템으로의 패킷을 보호합니다.	<a href="#">218 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”</a>
IPsec 정책을 사용하여 웹 서버를 보호합니다.	비웹 트래픽에서 IPsec를 사용하도록 합니다. 웹 클라이언트는 IPsec 검사를 우회하는 특정 포트로 식별됩니다.	<a href="#">221 페이지 “IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법”</a>
IPsec 정책을 표시합니다.	현재 적용 중인 IPsec 정책을 적용 순서대로 표시합니다.	<a href="#">222 페이지 “IPsec 정책을 표시하는 방법”</a>
IKE를 사용하여 IPsec SA에 대한 키 입력 자료를 자동으로 만듭니다.	보안 연결을 위한 원시 데이터를 제공합니다.	<a href="#">251 페이지 “IKE 구성(작업 맵)”</a>
보안 VPN(virtual private network)을 설정합니다.	인터넷을 거치는 두 시스템 사이에 IPsec를 설정합니다.	<a href="#">223 페이지 “IPsec를 사용하여 VPN 보호”</a>

## ▼ IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법

이 절차에서는 다음 설정을 가정합니다.

- 두 시스템의 이름은 `enigma` 및 `partym`입니다.
- 각 시스템에는 IP 주소가 있습니다. 이 주소는 IPv4 주소 또는 IPv6 주소 또는 둘 다 될 수 있습니다.
- 각 시스템에는 AES 알고리즘을 사용한 ESP 암호화(128비트의 키 필요) 및 SHA-2 메시지 다이제스트를 사용한 ESP 인증(512비트의 키 필요)이 필요합니다.
- 각 시스템은 공유 보안 연결을 사용합니다.  
공유 SA를 사용하여 두 시스템을 보호하는 데 한 쌍의 SA만 필요합니다.

주 - Trusted Extensions 시스템에서 레이블이 있는 IPsec를 사용하려면 **Trusted Extensions 구성 및 관리**의 “다중 레벨 Trusted Extensions 네트워크에서 IPsec 보호를 적용하는 방법”을 참조하십시오.

**시작하기 전에** IPsec 정책은 전역 영역 또는 배타적 IP 스택 영역에서 구성할 수 있습니다. 공유 IP 스택에 대한 정책은 전역 영역에서 구성해야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.

#### 1 관리자가 됩니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 “관리 권한을 얻는 방법”을 참조하십시오. 원격으로 로그인할 경우 보안 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 예 15-1을 참조하십시오.

#### 2 각 시스템에서 호스트 항목을 /etc/inet/hosts 파일에 추가합니다.

이 단계를 통해 SMF(서비스 관리 기능)에서 존재하지 않는 이름 지정 서비스에 의존하지 않고 시스템 이름을 사용할 수 있습니다. 자세한 내용은 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

##### a. 이름이 partym인 시스템에서 hosts 파일에 다음을 입력합니다.

```
# Secure communication with enigma
192.168.116.16 enigma
```

##### b. 이름이 enigma인 시스템에서 hosts 파일에 다음을 입력합니다.

```
# Secure communication with partym
192.168.13.213 partym
```

#### 3 각 시스템에서 IPsec 정책 파일을 만듭니다.

파일 이름은 /etc/inet/ipsecinit.conf입니다. 예는 /etc/inet/ipsecinit.sample 파일을 참조하십시오.

#### 4 IPsec 정책 항목을 ipsecinit.conf 파일에 추가합니다.

##### a. enigma 시스템에서 다음 정책을 추가합니다.

```
{laddr enigma raddr partym} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

##### b. partym 시스템에서 동일한 정책을 추가합니다.

```
{laddr partym raddr enigma} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

IPsec 정책 항목의 구문은 [ipsecconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

#### 5 각 시스템에서 IKE를 구성하여 두 시스템 사이에 IPsec SA 쌍을 추가합니다.

[251 페이지 “IKE 구성\(작업 맵\)”](#)의 구성 절차 중 하나에 따라 IKE를 구성합니다. IKE 구성 파일의 구문은 [ike.config\(4\)](#) 매뉴얼 페이지를 참조하십시오.

주 - 키를 수동으로 생성하고 유지 관리해야 하는 경우 [231 페이지](#) “IPsec 키를 수동으로 만드는 방법”을 참조하십시오.

## 6 IPsec 정책 파일의 구문을 확인합니다.

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

## 7 IPsec 정책을 새로 고칩니다.

```
# svcadm refresh svc:/network/ipsec/policy:default
```

IPsec 정책은 기본적으로 사용으로 설정되므로 **새로 고칩니다**. IPsec 정책을 사용 안함으로 설정한 경우 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/policy:default
```

## 8 IPsec에 대한 키를 활성화합니다.

- **ike** 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/ike:default
```

- **ike** 서비스가 사용으로 설정된 경우 다시 시작합니다.

```
# svcadm restart svc:/network/ipsec/ike:default
```

[단계 5](#)에서 키를 수동으로 구성한 경우 [231 페이지](#) “IPsec 키를 수동으로 만드는 방법”을 완료하여 키를 활성화합니다.

## 9 패킷이 보호되고 있는지 확인합니다.

절차는 [235 페이지](#) “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

## 예 15-1 ssh 연결을 사용할 때 IPsec 정책 추가

이 예에서는 root 역할의 관리자가 ssh 명령을 사용하여 두번째 시스템에 접근한 다음 두 시스템에서 IPsec 정책 및 키를 구성합니다. 자세한 내용은 [ssh\(1\)](#) 매뉴얼 페이지를 참조하십시오.

- 먼저 관리자는 위 절차의 [단계 2 ~ 단계 6](#)를 수행하여 첫번째 시스템을 구성합니다.
- 그런 다음 다른 터미널 창에서 관리자는 ssh 명령을 사용하여 두번째 시스템에 로그인합니다.

```
local-system # ssh other-system
other-system #
```

- ssh 세션의 터미널 창에서 관리자는 [단계 2 ~ 단계 8](#)를 완료하여 두번째 시스템의 IPsec 정책 및 키를 구성합니다.
- 그런 다음 관리자는 ssh 세션을 종료합니다.

```
other-system # exit
local-system #
```

- 마지막으로 관리자는 **단계 7** 및 **단계 8**를 완료하여 첫번째 시스템에서 IPsec 정책을 사용으로 설정합니다.

ssh 연결 사용을 포함하여 다음에 두 시스템이 통신할 때 통신이 IPsec로 보호됩니다.

## ▼ IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법

보안 웹 서버를 통해 웹 클라이언트가 웹 서비스와 통신할 수 있습니다. 보안 웹 서버에서 웹 트래픽이 아닌 트래픽은 보안 검사를 **통과해야** 합니다. 다음 절차에는 웹 트래픽에 대한 우회가 포함됩니다. 또한 이 웹 서버는 비보안 DNS 클라이언트 요청을 할 수 있습니다. 기타 모든 트래픽에는 AES 및 SHA-2 알고리즘을 사용하는 ESP가 필요합니다.

**시작하기 전에** IPsec 정책을 구성하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다. **218 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”**을 완료했으므로 다음 조건이 적용됩니다.

- 두 시스템 사이의 통신이 IPsec로 보호됩니다.
- 키 입력 자료가 IKE에 의해 생성됩니다.
- 패킷이 보호되고 있는지 확인했습니다.

### 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”**을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 **ssh** 명령을 사용합니다. 예는 **예 15-1**을 참조하십시오.

### 2 보안 정책 검사를 우회해야 하는 서비스를 결정합니다.

웹 서버의 경우 이러한 서비스에는 TCP 포트 80(HTTP) 및 443(보안 HTTP)이 포함됩니다. 웹 서버에서 DNS 이름 조회를 제공하는 경우 TCP 및 UDP 모두에 대해 포트 53가 서버에 포함되어야 할 수도 있습니다.

### 3 웹 서버 정책을 IPsec 정책 파일에 추가합니다.

다음 행을 `/etc/inet/ipsecinit.conf` 파일에 추가합니다.

```
# Web traffic that web server should bypass.
{lport 80 ulp tcp dir both} bypass {}
{lport 443 ulp tcp dir both} bypass {}

# Outbound DNS lookups should also be bypassed.
{rport 53 dir both} bypass {}

# Require all other traffic to use ESP with AES and SHA-2.
# Use a unique SA for outbound traffic from the port
{} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

이 구성은 [단계 2](#)에서 설명한 우회 예외 사항과 함께 보안 트래픽만 시스템에 액세스할 수 있도록 허용합니다.

**4 IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

**5 IPsec 정책을 새로 고칩니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

**6 IPsec에 대한 키를 새로 고칩니다.**

ike 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/ipsec/ike
```

키를 수동으로 구성한 경우 [231 페이지](#) “IPsec 키를 수동으로 만드는 방법”의 지침을 따릅니다.

설정이 완료되었습니다. 선택적으로 [단계 7](#)를 수행할 수 있습니다.

**7 (옵션) 원격 시스템이 비웹 트래픽에 대해 웹 서버와 통신할 수 있도록 설정합니다.**

다음 행을 원격 시스템의 /etc/inet/ipsecinit.conf 파일에 추가합니다.

```
# Communicate with web server about nonweb stuff
#
{laddr webserver} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

구문을 확인한 다음 IPsec 정책을 새로 고쳐 활성화합니다.

```
remote-system # ipsecconf -c -f /etc/inet/ipsecinit.conf
remote-system # svcadm refresh svc:/network/ipsec/policy:default
```

원격 시스템은 시스템의 IPsec 정책이 일치할 경우에만 비웹 트래픽에 대해 웹 서버와 안전하게 통신할 수 있습니다.

## ▼ IPsec 정책을 표시하는 방법

ipsecconf 명령을 인수 없이 실행하면 시스템에서 구성된 정책을 볼 수 있습니다.

**시작하기 전에** ipsecconf 명령은 전역 영역에서 실행해야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 ipsecconf 명령을 실행합니다.

**1 Network IPsec Management 프로파일이 포함된 역할을 말합니다.**

네트워크 보안에 대한 고유의 역할을 만들고 사용자에게 지정하려면 [232 페이지](#) “네트워크 보안에 대한 역할을 구성하는 방법”을 참조하십시오.

## 2 IPsec 정책을 표시합니다.

- 항목이 추가된 순서대로 전역 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf
```

명령은 색인 다음에 번호와 함께 각 항목을 표시합니다.

- 일치하는 순서대로 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf -l -n
```

- 터널별 항목을 포함하여 일치하는 순서대로 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf -L -n
```

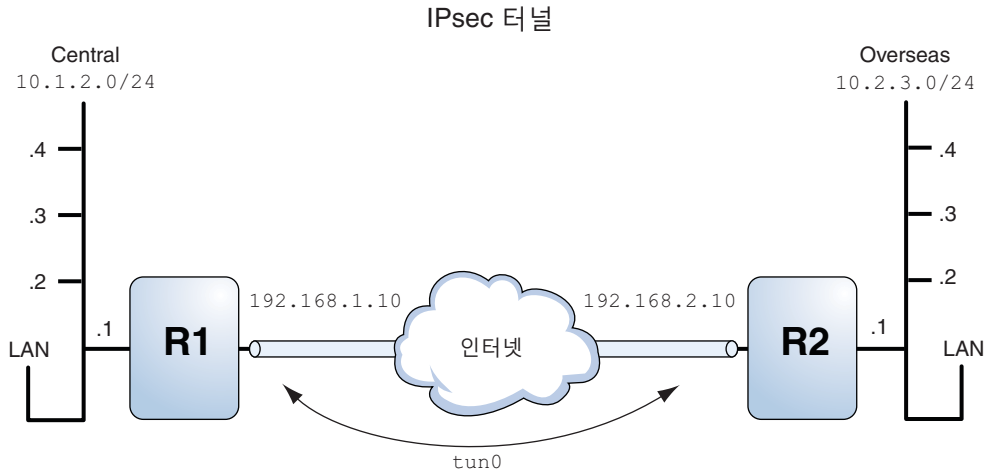
## IPsec를 사용하여 VPN 보호

Oracle Solaris는 IPsec로 보호되는 VPN을 구성할 수 있습니다. 터널은 **터널 모드** 또는 **전송 모드**에서 만들 수 있습니다. 자세한 내용은 [209 페이지 “IPsec의 전송 및 터널 모드”](#)를 참조하십시오. 이 절의 예와 절차에서는 IPv4 주소를 사용하지만, 예와 절차는 IPv6 VPN에도 적용됩니다. 추가 정보는 [217 페이지 “IPsec를 사용하여 트래픽 보호”](#)를 참조하십시오.

터널 모드에서 터널에 대한 IPsec 정책의 예는 [223 페이지 “터널 모드를 사용하여 IPsec로 VPN을 보호하는 예”](#)를 참조하십시오.

## 터널 모드를 사용하여 IPsec로 VPN을 보호하는 예

그림 15-1 IPsec로 보호되는 터널



다음 예에서는 터널이 LAN의 모든 서브넷에 대해 구성되어 있다고 가정합니다.

```
## Tunnel configuration ##
# Tunnel name is tun0
# Intranet point for the source is 10.1.2.1
# Intranet point for the destination is 10.2.3.1
# Tunnel source is 192.168.1.10
# Tunnel destination is 192.168.2.10

# Tunnel name address object is tun0/to-central
# Tunnel name address object is tun0/to-overseas
```

#### 예 15-2 모든 서브넷에서 사용할 수 있는 터널 만들기

이 예에서는 [그림 15-1](#)에 나온 Central LAN 로컬 LAN의 모든 트래픽이 Router 1을 거쳐 Router 2로 터널링된 다음 Overseas LAN의 모든 로컬 LAN에 전달될 수 있습니다. 이 트래픽은 AES로 암호화됩니다.

```
## IPsec policy ##
{tunnel tun0 negotiate tunnel}
  ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

#### 예 15-3 두 서브넷만 연결하는 터널 만들기

이 예에서는 Central LAN의 서브넷 10.1.2.0/24와 Overseas LAN의 서브넷 10.2.3.0/24 사이의 트래픽만 터널링되고 암호화됩니다. Central에 대한 다른 IPsec 정책이 없을 때 Central LAN에서 이 터널을 통해 다른 LAN에 대한 트래픽을 경로 지정하려고 시도하면 트래픽이 Router 1에서 삭제됩니다.



## 예 15-3 두 서브넷만 연결하는 터널 만들기 (계속)

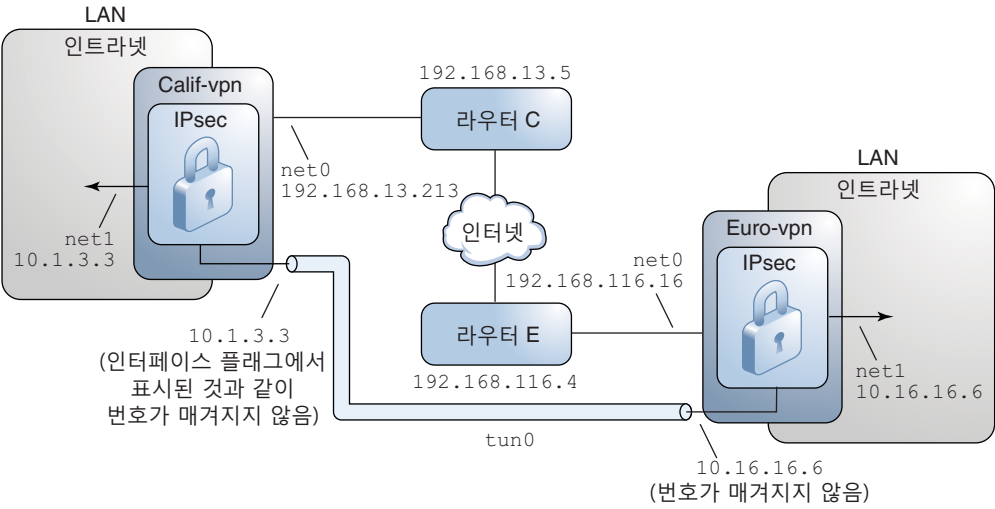
```
## IPsec policy ##
{tunnel tun0 negotiate tunnel laddr 10.1.2.0/24 raddr 10.2.3.0/24}
ipsec {encr_algs aes encr_auth_algs sha512 shared}
```

## VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명

이 절에 나오는 절차에서는 다음 설정을 가정합니다. 네트워크 그림은 [그림 15-2](#)를 참조하십시오.

- 각 시스템은 IPv4 주소 공간을 사용합니다.
- 각 시스템에는 두 개의 인터페이스가 있습니다. `net0` 인터페이스는 인터넷에 연결됩니다. 이 예에서 인터넷 IP 주소는 `192.168`로 시작됩니다. `net1` 인터페이스는 회사의 LAN(인트라넷)에 연결됩니다. 이 예에서는 인트라넷 IP 주소가 숫자 `10`으로 시작됩니다.
- 각 시스템에는 SHA-2 알고리즘을 사용하는 ESP 인증이 필요합니다. 이 예에서 SHA-2 알고리즘에는 512비트 키가 필요합니다.
- 각 시스템에는 AES 알고리즘을 사용하는 ESP 암호화가 필요합니다. AES 알고리즘은 128비트 또는 256비트 키를 사용합니다.
- 각 시스템은 인터넷에 직접 액세스되는 라우터에 연결할 수 있습니다.
- 각 시스템은 공유 보안 연결을 사용합니다.

그림 15-2 인터넷으로 연결된 사무실 사이의 샘플 VPN



위의 그림에 나온 대로 절차에서는 다음 구성 매개변수를 사용합니다.

매개변수	유럽	캘리포니아
시스템 이름	euro-vpn	calif-vpn
시스템 인트라넷 인터페이스	net1	net1
시스템 인트라넷 주소(또한 단계 7의 <i>-point</i> 주소)	10.16.16.6	10.1.3.3
시스템 인트라넷 주소 객체	net1/inside	net1/inside
시스템 인터넷 인터페이스	net0	net0
시스템 인터넷 주소(또한 단계 7의 <i>tsrc</i> 주소)	192.168.116.16	192.168.13.213
인터넷 라우터의 이름	router-E	router-C
인터넷 라우터의 주소	192.168.116.4	192.168.13.5
터널 이름	tun0	tun0
터널 이름 주소 객체	tun0/v4tunaddr	tun0/v4tunaddr

터널 이름에 대한 정보는 119 페이지 “[dladm 명령을 통한 터널 구성 및 관리](#)”를 참조하십시오. 주소 객체에 대한 정보는 47 페이지 “[IP 인터페이스 구성 방법](#)” 및 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ 터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법

터널 모드에서 내부 IP 패킷은 해당 콘텐츠를 보호하는 IPsec 정책을 결정합니다.

이 절차는 절차 218 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”을 확장합니다. 설정은 225 페이지 “VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명”에 설명되어 있습니다.

특정 명령을 실행하는 이유에 대한 자세한 설명은 218 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”에서 해당하는 단계를 참조하십시오.

---

주 - 두 시스템에서 이 절차의 단계를 수행하십시오.

---

두 시스템 연결과 함께 이러한 두 시스템에 연결되는 두 인트라넷을 연결하게 됩니다. 이 절차에서 시스템은 게이트웨이로 작동합니다.

---

주 - Trusted Extensions 시스템에서 레이블이 있는 터널 모드로 IPsec를 사용하려면 **Trusted Extensions 구성 및 관리**의 “신뢰할 수 없는 네트워크에서 터널을 구성하는 방법”에서 이 절차의 확장을 참조하십시오.

---

**시작하기 전에** 시스템 또는 공유 IP 영역에 대한 IPsec 정책을 구성하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.

### 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 “관리 권한을 얻는 방법”을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 예 15-1을 참조하십시오.

### 2 IPsec를 구성하기 전에 패킷의 흐름을 제어합니다.

#### a. IP 전달 및 IP 동적 경로 지정을 사용 안함으로 설정합니다.

```
# routeadm -d ipv4-routing
# ipadm set-prop -p forwarding=off ipv4
# routeadm -u
```

IP 전달을 해제하면 패킷이 이 시스템을 통해 한 네트워크에서 다른 네트워크로 전달되지 않습니다. routeadm 명령에 대한 설명은 routeadm(1M) 매뉴얼 페이지를 참조하십시오.

#### b. IP 엄격한 다중 홈 지정을 설정합니다.

```
# ipadm set-prop -p hostmodel=strong ipv4
```

IP 엄격한 다중 홉 지정을 설정하면 시스템의 대상 주소 중 하나에 대한 패킷이 올바른 대상 주소에 도달해야 합니다.

hostmodel 매개변수가 **strong**으로 설정되면 특정 인터페이스에 도달하는 패킷이 해당 인터페이스의 로컬 IP 주소 중 하나로 지정되어야 합니다. 기타 모든 패킷은 시스템의 다른 로컬 주소로 지정된 패킷이라도 삭제됩니다.

**c. 대부분의 네트워크 서비스가 사용 안함으로 설정되었는지 확인합니다.**

루프백 마운트 및 ssh 서비스가 실행 중인지 확인합니다.

```
# svcs | grep network
online          Aug_02   svc:/network/loopback:default
...
online          Aug_09   svc:/network/ssh:default
```

**3 IPsec 정책을 추가합니다.**

/etc/inet/ipsecinit.conf 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다. 추가 예는 [223 페이지 “터널 모드를 사용하여 IPsec로 VPN을 보호하는 예”](#)를 참조하십시오.

이 정책에서 로컬 LAN의 시스템과 게이트웨이의 내부 IP 주소 사이에는 IPsec 보호가 필요하지 않으므로 **bypass** 명령문이 추가됩니다.

**a. euro-vpn 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.**

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.16.16.6 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-2.
{tunnel tun0 negotiate tunnel}
ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

**b. calif-vpn 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.**

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-2.
{tunnel tun0 negotiate tunnel}
ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

**4 각 시스템에서 IKE를 구성하여 두 시스템 사이에 IPsec SA 쌍을 추가합니다.**

[251 페이지 “IKE 구성\(작업 맵\)”](#)의 구성 절차 중 하나에 따라 IKE를 구성합니다. IKE 구성 파일의 구문은 [ike.config\(4\)](#) 매뉴얼 페이지를 참조하십시오.

---

주- 키를 수동으로 생성하고 유지 관리해야 하는 경우 [231 페이지 “IPsec 키를 수동으로 만드는 방법”](#)을 참조하십시오.

---

**5 IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

## 6 IPsec 정책을 새로 고칩니다.

```
# svcadm refresh svc:/network/ipsec/policy:default
```

IPsec 정책은 기본적으로 사용으로 설정되므로 **새로 고칩니다**. IPsec 정책을 사용 안함으로 설정한 경우 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/policy:default
```

## 7 tunnel-name 터널을 만들고 구성합니다.

다음 명령은 내부 및 외부 인터페이스를 구성하고, tun0 터널을 만들며, IP 주소를 터널에 지정합니다.

### a. calif-vpn 시스템에서 터널을 만들고 구성합니다.

net1 인터페이스가 존재하지 않을 경우 첫번째 명령이 만듭니다.

```
# ipadm create-addr -T static -a local=10.1.3.3 net1/inside
# dladm create-iptun -T ipv4 -a local=10.1.3.3,remote=10.16.16.6 tun0
# ipadm create-addr -T static \
-a local=192.168.13.213,remote=192.168.116.16 tun0/v4tunaddr
```

### b. euro-vpn 시스템에서 터널을 만들고 구성합니다.

```
# ipadm create-addr -T static -a local=10.16.16.6 net1/inside
# dladm create-iptun -T ipv4 -a local=10.16.16.6,remote=10.1.3.3 tun0
# ipadm create-addr -T static \
-a local=192.168.116.16,remote=192.168.13.213 tun0/v4tunaddr
```

---

주 - ipadm 명령에 대한 -T 옵션은 만들 주소의 유형을 지정합니다. dladm 명령에 대한 -T 옵션은 터널을 지정합니다.

---

이러한 명령에 대한 자세한 내용은 [dladm\(1M\)](#) 및 [ipadm\(1M\)](#) 매뉴얼 페이지와 [47 페이지 “IP 인터페이스 구성 방법”](#)을 참조하십시오. 사용자 정의된 이름에 대한 자세한 내용은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “네트워크 장치 및 데이터 링크 이름”](#)을 참조하십시오.

## 8 각 시스템에서 다음을 구성합니다.

```
# ipadm set-ifprop -m ipv4 -p forwarding=on net1
# ipadm set-ifprop -m ipv4 -p forwarding=off net0
```

IP 전달은 다른 곳에서 도달한 패킷을 전달할 수 있음을 의미합니다. 또한 IP 전달은 이 인터페이스에서 떠난 패킷이 다른 곳에서 왔을 수 있음을 의미합니다. 패킷을 성공적으로 전달하려면 수신 인터페이스와 전송 인터페이스에 모두 IP 전달이 설정되어 있어야 합니다.

net1 인터페이스는 인트라넷 **내부**에 있으므로 net1에 대해 IP 전달이 설정되어 있어야 합니다. tun0은 인터넷을 통해 두 시스템을 연결하므로 tun0에 대해 IP 전달이 설정되어 있어야 합니다. net0 인터페이스의 경우 **외부** 공격자가 보호된 인트라넷에 패킷을 주입하지 못하도록 IP 전달이 해제되어 있습니다. **외부**는 인터넷을 의미합니다.

9 각 시스템에서 개인 인터페이스의 알림을 막습니다.

```
# ipadm set-addrprop -p private=on net0
```

net0에 IP 전달이 해제되어 있더라도 경로 지정 프로토콜 구현은 여전히 인터페이스를 알릴 수 있습니다. 예를 들어, in.routed 프로토콜은 net0이 인트라넷 내부의 피어에 패킷을 전달할 수 있음을 알릴 수 있습니다. 인터페이스의 개인 플래그를 설정하여 알림을 막을 수 있습니다.

10 네트워크 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/initial:default
```

11 net0 인터페이스를 통한 기본 경로를 수동으로 추가합니다.

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

a. calif-vpn 시스템에서 다음 경로를 추가합니다.

```
# route -p add net default 192.168.13.5
```

b. euro-vpn 시스템에서 다음 경로를 추가합니다.

```
# route -p add net default 192.168.116.4
```

net0 인터페이스는 인트라넷의 일부가 아니지만 net0은 인터넷을 거쳐 피어 시스템에 도달할 필요가 없습니다. 피어를 찾으려면 net0은 인터넷 경로 지정에 대한 정보가 필요합니다. VPN 시스템은 나머지 인터넷에 라우터가 아닌 호스트로 나타납니다. 따라서 기본 라우터를 사용하거나 라우터 검색 프로토콜을 실행하여 피어 시스템을 찾을 수 있습니다. 자세한 내용은 [route\(1M\)](#) 및 [in.routed\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

# IPsec 및 IKE 관리

다음 작업 맵에서는 IPsec를 관리할 때 사용할 수 있는 작업을 안내합니다.

작업	설명	수행 방법
보안 연결을 수동으로 만들거나 바꿉니다.	보안 연결을 위한 원시 데이터를 제공합니다. <ul style="list-style-type: none"><li>■ IPsec 알고리즘 이름 및 키 입력 자료</li><li>■ SPI(보안 매개변수 색인)</li><li>■ IP 소스 및 대상 주소와 기타 매개변수</li></ul>	231 페이지 “IPsec 키를 수동으로 만드는 방법”
네트워크 보안 역할을 만듭니다.	보안 네트워크를 설정할 수 있지만 root 역할보다 권한이 적은 역할을 만듭니다.	232 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”
IPsec 및 키 입력 자료를 SMF 서비스의 일부로 관리합니다.	서비스를 사용으로 설정, 사용 안함으로 설정, 새로 고침 및 다시 시작하는 명령을 언제, 어떻게 사용하는지 설명합니다. 또한 서비스의 등록 정보 값을 변경하는 명령을 설명합니다.	234 페이지 “IPsec 및 IKE 서비스를 관리하는 방법”

작업	설명	수행 방법
IPsec가 패킷을 보호하고 있는지 확인합니다.	IP 데이터그램이 어떻게 보호되는지 나타내는 특정 헤더에 대한 snoop 출력을 검사합니다.	235 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”

## ▼ IPsec 키를 수동으로 만드는 방법

다음 절차에서는 [단계 5 in 218 페이지](#) “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”에 대한 키 입력 자료를 제공합니다. `partym` 및 `enigma`의 두 시스템에 대한 키를 생성합니다. 한 시스템에서 키를 생성한 다음 첫번째 시스템의 키를 두 시스템에서 모두 사용합니다.

시작하기 전에 비전역 영역에 대한 키 입력 자료를 수동으로 관리하려면 전역 영역에 있어야 합니다.

### 1 SA에 대한 키 입력 자료를 생성합니다.

#### a. 필요한 키를 결정합니다.

아우바운드 트래픽에 대한 3개의 16진수 임의 숫자 및 인바운드 트래픽에 대한 3개의 16진수 임의 숫자가 필요합니다. 따라서 한 시스템에서 다음 숫자를 생성해야 합니다.

- spi 키워드에 대한 값으로 2개의 16진수 임의 숫자. 하나는 아웃바운드 트래픽용입니다. 다른 하나는 인바운드 트래픽용입니다. 각 숫자 모두 최대 8자까지만 허용됩니다.
- AH의 SHA-2 알고리즘에 대한 2개의 16진수 임의 숫자. 각 숫자 모두 512자까지만 허용됩니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.
- ESP의 3DES 알고리즘에 대한 2개의 16진수 임의 숫자. 각 숫자의 길이는 168자여야 합니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.

#### b. 필요한 키를 생성합니다.

- 사이트에 임의 숫자 생성기가 있을 경우 생성기를 사용하십시오.
- [Oracle Solaris 관리: 보안 서비스](#)의 “`pktool` 명령을 사용하여 대칭 키를 생성하는 방법” 및 해당 절의 IPsec 예에 나온 대로 `pktool` 명령을 사용합니다.

### 2 각 시스템에서 root 역할을 사용하여 IPsec에 대한 수동 키 파일에 키를 추가합니다.

#### a. `enigma` 시스템에서 `/etc/inet/secret/ipseckeys` 파일을 다음과 유사하게 편집합니다.

```
# ipseckeys - This file takes the file format documented in
# ipseckey(1m).
# Note that naming services might not be available when this file
# loads, just like ipsecinit.conf.
#
# Backslashes indicate command continuation.
```

```
#
# for outbound packets on enigma
add esp spi 0x8bcd1407 \
  src 192.168.116.16 dst 192.168.13.213 \
  encr_alg 3des \
  auth_alg sha512 \
  encrkey d41fb74470271826a8e7a80d343cc5aa... \
  authkey e896f8df7f78d6cab36c94ccf293f031...
#
# for inbound packets
add esp spi 0x122a43e4 \
  src 192.168.13.213 dst 192.168.116.16 \
  encr_alg 3des \
  auth_alg sha512 \
  encrkey dd325c5c137fb4739a55c9b3a1747baa... \
  authkey ad9ced7ad5f255c9a8605fba5eb4d2fd...
```

b. 읽기 전용 권한으로 파일을 보호합니다.

```
# chmod 400 /etc/inet/secret/ipseckey
```

c. 파일의 구문을 확인합니다.

```
# ipseckey -c -f /etc/inet/secret/ipseckey
```

---

주 - 두 시스템의 키 입력 자료는 동일해야 합니다.

---

### 3 IPsec에 대한 키를 활성화합니다.

- manual-key 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/manual-key:default
```

- manual-key 서비스가 사용으로 설정된 경우 새로 고칩니다.

```
# svcadm refresh ipsec/manual-key
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## ▼ 네트워크 보안에 대한 역할을 구성하는 방법

Oracle Solaris의 RBAC(role-based access control) 기능을 사용하여 시스템을 관리하는 경우 이 절차에 따라 네트워크 관리 역할 또는 네트워크 보안 역할을 제공합니다.

### 1 사용 가능한 네트워크 관련 권한 프로파일을 나열합니다.

```
% getent prof_attr | grep Network | more
Console User:RO::Manage System as the Console User...
Network Management:RO::Manage the host and network configuration...
Network Autoconf Admin:RO::Manage Network Auto-Magic configuration via nwamd...
Network Autoconf User:RO::Network Auto-Magic User...
```



```

Network ILB:RO::Manage ILB configuration via ilbadm...
Network LLDP:RO::Manage LLDP agents via lldpadm...
Network VRRP:RO::Manage VRRP instances...
Network Observability:RO::Allow access to observability devices...
Network Security:RO::Manage network and host security...:profiles=Network Wifi
Security,Network Link Security,Network IPsec Management...
Network Wifi Management:RO::Manage wifi network configuration...
Network Wifi Security:RO::Manage wifi network security...
Network Link Security:RO::Manage network link security...
Network IPsec Management:RO::Manage IPsec and IKE...
System Administrator:RO::Can perform most non-security administrative tasks:profiles=...Network Management...
Information Security:RO::Maintains MAC and DAC security policies:profiles=...Network Security...

```

Network Management 프로파일은 System Administrator 프로파일의 보조 프로파일입니다. 역할에 System Administrator 권한 프로파일을 포함시킨 경우 해당 역할은 Network Management 프로파일의 명령을 실행할 수 있습니다.

## 2 Network Management 권한 프로파일의 명령을 나열합니다.

```

% getent exec_attr | grep "Network Management"
...
Network Management:solaris:cmd:::/sbin/dlstat:euid=dladm;egid=sys
...
Network Management:solaris:cmd:::/usr/sbin/snoop:privs=net_observability
Network Management:solaris:cmd:::/usr/sbin/spray:euid=0 ...

```

## 3 사이트에서 네트워크 보안 역할의 범위를 결정합니다.

단계 1의 권한 프로파일 정의를 사용하여 결정합니다.

- 모든 네트워크 보안을 처리하는 역할을 만들려면 Network Security 권한 프로파일을 사용합니다.
- IPsec 및 IKE만 처리하는 역할을 만들려면 Network IPsec Management 권한 프로파일을 사용합니다.

## 4 Network Management 권한 프로파일을 포함하는 네트워크 보안 역할을 만듭니다.

Network Management 권한 프로파일과 함께 Network Security 또는 Network IPsec Management 권한 프로파일을 가진 역할은 대표적으로 해당 권한으로 ipadm, ipseckey 및 snoop 명령을 실행할 수 있습니다.

역할을 만들고, 사용자에게 역할을 지정하고, 이름 지정 서비스에 변경 사항을 등록하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.

## 예 15-4 역할 간 네트워크 보안 책임 구분

이 예에서는 관리자가 두 역할 간에 네트워크 보안 책임을 구분합니다. 한 역할은 wifi 링크 보안을 관리하고, 다른 역할은 IPsec 및 IKE를 관리합니다. 각 역할은 교대당 한 사람씩 세 명의 사용자에게 지정됩니다.

역할은 관리자가 다음과 같이 만듭니다.

- 관리자는 첫번째 역할 이름을 LinkWifi로 지정합니다.
  - 관리자는 Network Wifi, Network Link Security 및 Network Management 권한 프로파일을 역할에 지정합니다.
  - 그런 다음 관리자는 LinkWifi 역할을 해당 사용자에게 지정합니다.
- 관리자는 두번째 역할 이름을 IPsec Administrator로 지정합니다.
  - 관리자는 Network IPsec Management 및 Network Management 권한 프로파일을 역할에 지정합니다.
  - 그런 다음 관리자는 IPsec Administrator 역할을 해당 사용자에게 지정합니다.

## ▼ IPsec 및 IKE 서비스를 관리하는 방법

다음 단계에서는 IPsec, IKE 및 수동 키 관리에 대한 SMF 서비스의 가장 일반적인 사용을 제공합니다. 기본적으로 policy 및 ipsecalgss 서비스는 사용으로 설정됩니다. 또한 기본적으로 ike 및 manual-key 서비스는 사용 안함으로 설정됩니다.

### 1 IPsec 정책을 관리하려면 다음 중 하나를 수행합니다.

- ipsecinit.conf 파일에 새 정책을 추가한 후 policy 서비스를 새로 고칩니다.
 

```
# svcadm refresh svc:/network/ipsec/policy
```
- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 policy 서비스를 새로 고치고 다시 시작합니다.
 

```
# svccfg -s policy setprop config/config_file=/etc/inet/MyIpsecinit.conf
# svccfg -s policy listprop config/config_file
config/config_file astring /etc/inet/MyIpsecinit.conf
# svcadm refresh svc:/network/ipsec/policy
# svcadm restart svc:/network/ipsec/policy
```

### 2 키를 자동으로 관리하려면 다음 중 하나를 수행합니다.

- /etc/inet/ike/config 파일에 항목을 추가한 후 ike 서비스를 사용으로 설정합니다.
 

```
# svcadm enable svc:/network/ipsec/ike
```
- /etc/inet/ike/config 파일에서 항목을 변경한 후 ike 서비스를 다시 시작합니다.
 

```
# svcadm restart svc:/network/ipsec/ike:default
```
- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 서비스를 새로 고치고 다시 시작합니다.
 

```
# svccfg -s ike setprop config/admin_privilege = astring: "modkeys"
# svccfg -s ike listprop config/admin_privilege
config/admin_privilege astring modkeys
```

```
# svcadm refresh svc:/network/ipsec/ike
# svcadm restart svc:/network/ipsec/ike
```

- **ike** 서비스를 중지하려면 사용 안함으로 설정합니다.

```
# svcadm disable svc:/network/ipsec/ike
```

### 3 키를 수동으로 관리하려면 다음 중 하나를 수행합니다.

- **/etc/inet/secret/ipseckey** 파일에 항목을 추가한 후 **manual-key** 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/manual-key:default
```

- **ipseckey** 파일을 변경한 수 서비스를 새로 고칩니다.

```
# svcadm refresh manual-key
```

- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 서비스를 새로 고치고 다시 시작합니다.

```
# svccfg -s manual-key setprop config/config_file=/etc/inet/secret/MyIpseckeyfile
# svccfg -s manual-key listprop config/config_file
config/config_file astring /etc/inet/secret/MyIpseckeyfile
# svcadm refresh svc:/network/ipsec/manual-key
# svcadm restart svc:/network/ipsec/manual-key
```

- 수동 키 관리를 막으려면 **manual-key** 서비스를 사용 안함으로 설정합니다.

```
# svcadm disable svc:/network/ipsec/manual-key
```

### 4 IPsec 프로토콜 및 알고리즘 테이블을 수정할 경우 **ipsecalgs** 서비스를 새로 고칩니다.

```
# svcadm refresh svc:/network/ipsec/ipsecalgs
```

**일반 오류** `svcs service` 명령을 사용하여 서비스의 상태를 찾습니다. 서비스가 maintenance 모드인 경우 `svcs -x service` 명령 출력의 디버깅 제안을 따릅니다.

## ▼ IPsec로 패킷이 보호되는지 확인하는 방법

패킷이 보호되는지 확인하려면 **snoop** 명령을 사용하여 연결을 테스트합니다. 다음 접두어가 **snoop** 출력에 나타날 수 있습니다.

- **AH**: 접두어는 AH가 헤더를 보호하고 있음을 나타냅니다. **auth\_alg**를 사용하여 트래픽을 보호하는 경우 **AH**:를 보게 됩니다.
- **ESP**: 접두어는 암호화된 데이터가 보내지고 있음을 나타냅니다. **encr\_auth\_alg** 또는 **encr\_alg**를 사용하여 트래픽을 보호하는 경우 **ESP**:를 보게 됩니다.

**시작하기 전에** **snoop** 출력을 만들려면 **root** 역할을 가진 사용자여야 합니다. 연결을 테스트하려면 두 시스템에 대한 액세스 권한이 있어야 합니다.

**1 한 시스템(예:partym)에서 root 역할을 맡습니다.**

```
% su -
Password:      Type root password
#
```

**2 partym 시스템에서 원격 시스템으로부터 패킷 스누핑을 준비합니다.**

partym의 터미널 창에서 enigma 시스템으로부터 패킷을 스누핑합니다.

```
# snoop -d net0 -v enigma
Using device /dev/bge (promiscuous mode)
```

**3 원격 시스템에서 패킷을 보냅니다.**

다른 터미널 창에서 enigma 시스템에 원격으로 로그인합니다. 암호를 제공합니다. 그런 다음 root 역할을 맡고 enigma 시스템에서 partym 시스템으로 패킷을 보냅니다. 패킷은 snoop -v enigma 명령으로 캡처해야 합니다.

```
% ssh enigma
Password:      Type your password
% su -
Password:      Type root password
# ping partym
```

**4 snoop 출력을 검사합니다.**

partym 시스템에서 초기 IP 헤더 정보 이후 AH 및 ESP 정보가 포함된 출력을 볼 수 있어야 합니다. 다음과 유사한 AH 및 ESP 정보는 패킷이 보호되고 있음을 나타냅니다.

```
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 51 (AH)
IP:   Header checksum = 4e0e
IP:   Source address = 192.168.116.16, enigma
IP:   Destination address = 192.168.13.213, partym
IP:   No options
IP:
AH:   ----- Authentication Header -----
AH:
AH:   Next header = 50 (ESP)
AH:   AH length = 4 (24 bytes)
AH:   <Reserved field = 0x0>
AH:   SPI = 0xb3a8d714
AH:   Replay = 52
AH:   ICV = c653901433ef5a7d77c76eaa
AH:
ESP:   ----- Encapsulating Security Payload -----
ESP:
ESP:   SPI = 0xd4f40a61
ESP:   Replay = 52
ESP:   ....ENCRYPTED DATA....

ETHER:   ----- Ether Header -----
...
```

## IP 보안 아키텍처(참조)

---

이 장에는 다음 참조 정보가 포함되어 있습니다.

- 237 페이지 “IPsec 서비스”
- 238 페이지 “ipseccnf 명령”
- 238 페이지 “ipsecinit.conf 파일”
- 240 페이지 “ipsecalgs 명령”
- 240 페이지 “IPsec에 대한 보안 연결 데이터베이스”
- 241 페이지 “IPsec에서 SA 생성을 위한 유틸리티”
- 242 페이지 “snoop 명령 및 IPsec”

네트워크에서 IPsec를 구현하는 방법에 대한 지침은 15 장, “IPsec 구성(작업)”을 참조하십시오. IPsec의 개요는 14 장, “IP 보안 아키텍처(개요)”를 참조하십시오.

## IPsec 서비스

SMF(서비스 관리 기능)는 IPsec에 대한 다음 서비스를 제공합니다.

- `svc:/network/ipsec/policy` 서비스 - IPsec 정책을 관리합니다. 기본적으로 이 서비스는 사용으로 설정됩니다. `config_file` 등록 정보의 값은 `ipsecinit.conf` 파일의 위치를 결정합니다. 초기 값은 `/etc/inet/ipsecinit.conf`입니다.
- `svc:/network/ipsec/ipsecalgs` 서비스 - IPsec에 사용 가능한 알고리즘을 관리합니다. 기본적으로 이 서비스는 사용으로 설정됩니다.
- `svc:/network/ipsec/manual-key` 서비스 - 수동 키 관리를 활성화합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. `config_file` 등록 정보의 값은 `ipseckeys` 구성 파일의 위치를 결정합니다. 초기 값은 `/etc/inet/secret/ipseckeys`입니다.
- `svc:/network/ipsec/ike` 서비스 - IKE를 관리합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 구성 가능한 등록 정보는 281 페이지 “IKE 서비스”를 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 관리: 일반 작업의 6 장**, “서비스 관리(개요)”를 참조하십시오. 또한 `smf(5)`, `svcadm(1M)`, `svccfg(1M)` 매뉴얼 페이지를 참조하십시오.

## ipsecconf 명령

ipsecconf 명령을 사용하여 호스트에 대한 IPsec 정책을 구성합니다. 명령을 실행하여 정책을 구성할 때 시스템은 커널에 IPsec 정책 항목을 만듭니다. 시스템은 이러한 항목을 사용하여 모든 인바운드 및 아웃바운드 IP 데이터그램에 대한 정책을 확인합니다. 전달된 데이터그램은 이 명령을 사용하여 추가된 정책 확인에 종속되지 않습니다. ipsecconf 명령은 SPD(보안 정책 데이터베이스)도 구성합니다. IPsec 정책 옵션은 [ipsecconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

ipsecconf 명령을 호출하려면 root 역할을 가진 사용자여야 합니다. 명령은 양방향에서 트래픽을 보호하는 항목을 허용합니다. 또한 명령은 한 방향에서만 트래픽을 보호하는 항목도 허용합니다.

로컬 주소 및 원격 주소 형식의 정책 항목은 단일 정책 항목으로 양방향에서 트래픽을 보호할 수 있습니다. 예를 들어, laddr host1 및 raddr host2 패턴을 포함하는 항목은 이름 지정된 호스트에 대해 지정된 방향이 없더라도 양방향에서 트래픽을 보호합니다. 따라서 각 호스트에 대해 하나의 정책 항목만 필요합니다.

ipsecconf 명령으로 추가된 정책 항목은 시스템을 재부트하면 없어집니다. 시스템이 부트할 때 IPsec 정책이 활성화되도록 하려면 정책 항목을 /etc/inet/ipsecinit.conf 파일에 추가한 다음 policy 서비스를 새로 고치거나 사용으로 설정합니다. 예는 [217 페이지 “IPsec를 사용하여 트래픽 보호”](#)를 참조하십시오.

## ipsecinit.conf 파일

Oracle Solaris를 시작할 때 IPsec 보안 정책을 사용으로 설정하려면 구성 파일을 만들어 특정 IPsec 정책 항목으로 IPsec를 초기화합니다. 이 파일에 대한 기본 이름은 /etc/inet/ipsecinit.conf입니다. 정책 항목 및 해당 형식에 대한 자세한 내용은 [ipsecconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 정책이 구성된 후 svcadm refresh ipsec/policy 명령으로 정책을 새로 고칠 수 있습니다.

## 샘플 ipsecinit.conf 파일

Oracle Solaris 소프트웨어에는 샘플 IPsec 정책 파일 ipsecinit.sample이 포함되어 있습니다. 이 파일을 템플릿으로 사용하여 자신의 ipsecinit.conf 파일을 만들 수 있습니다. ipsecinit.sample 파일에는 다음 예가 포함되어 있습니다.

```
...
# In the following simple example, outbound network traffic between the local
# host and a remote host will be encrypted. Inbound network traffic between
# these addresses is required to be encrypted as well.
#
# This example assumes that 10.0.0.1 is the IPv4 address of this host (laddr)
# and 10.0.0.2 is the IPv4 address of the remote host (raddr).
```

```
#

{laddr 10.0.0.1 raddr 10.0.0.2} ipsec
{encr_algs aes encr_auth_algs sha256 sa shared}

# The policy syntax supports IPv4 and IPv6 addresses as well as symbolic names.
# Refer to the ipseconf(1M) man page for warnings on using symbolic names and
# many more examples, configuration options and supported algorithms.
#
# This example assumes that 10.0.0.1 is the IPv4 address of this host (laddr)
# and 10.0.0.2 is the IPv4 address of the remote host (raddr).
#
# The remote host will also need an IPsec (and IKE) configuration that mirrors
# this one.
#
# The following line will allow ssh(1) traffic to pass without IPsec protection:

{lport 22 dir both} bypass {}

#
# {laddr 10.0.0.1 dir in} drop {}
#
# Uncommenting the above line will drop all network traffic to this host unless
# it matches the rules above. Leaving this rule commented out will allow
# network packets that does not match the above rules to pass up the IP
# network stack. ,,,
```

## ipsecinit.conf 및 ipseconf에 대한 보안 고려 사항

IPsec 정책은 설정된 연결에 대해 변경할 수 없습니다. 정책을 변경할 수 없는 소켓을 **잠긴 소켓**이라고 합니다. 새 정책 항목은 이미 잠긴 소켓을 보호하지 않습니다. 자세한 내용은 [connect\(3SOCKET\)](#) 및 [accept\(3SOCKET\)](#) 매뉴얼 페이지를 참조하십시오. 의심스러운 경우 연결을 다시 시작하십시오.

이름 지정 시스템을 보호합니다. 다음 두 조건이 충족될 경우 호스트 이름을 더 이상 신뢰할 수 없습니다.

- 소스 주소가 네트워크를 통해 조회할 수 있는 호스트입니다.
- 이름 지정 시스템이 침해되었습니다.

보안 취약성은 실제 도구가 도구의 오용으로 인해 발생하기도 합니다. ipseconf 명령을 사용할 때는 주의해야 합니다. 가장 안전한 작업 모드를 위해서는 ssh 또는 콘솔 또는 기타 하드 연결된 TTY를 사용합니다.

## ipsecalgس 명령

Oracle Solaris의 암호화 프레임워크 기능은 IPsec에 인증 및 암호화 알고리즘을 제공합니다. `ipsecalgس` 명령은 각 IPsec 프로토콜이 지원하는 알고리즘을 나열할 수 있습니다. `ipsecalgس` 구성은 `/etc/inet/ipsecalgس` 파일에 저장됩니다. 일반적으로 이 파일은 수정할 필요가 없습니다. 하지만 파일을 수정해야 하는 경우 `ipsecalgس` 명령을 사용합니다. 파일을 직접 편집하면 안됩니다. 지원되는 알고리즘은 시스템 부트 시 `svc:/network/ipsec/ipsecalgس:default` 서비스로 커널과 동기화됩니다.

유효한 IPsec 프로토콜 및 알고리즘은 RFC 2407에 포함된 ISAKMP DOI(Domain of Interpretation)에 설명되어 있습니다. 일반적으로 DOI는 데이터 형식, 네트워크 트래픽 교환 유형 및 보안 관련 정보의 이름 지정 규칙을 정의합니다. 보안 관련 정보의 예로 보안 정책, 암호화 알고리즘, 암호화 모드 등이 있습니다.

구체적으로 ISAKMP DOI는 유효한 IPsec 알고리즘 및 해당 프로토콜(`PROTO_IPSEC_AH` 및 `PROTO_IPSEC_ESP`)에 대한 이름 지정 및 번호 지정 규칙을 정의합니다. 각 알고리즘은 정확히 하나의 프로토콜과 연결됩니다. 이러한 ISAKMP DOI 정의는 `/etc/inet/ipsecalgس` 파일에 있습니다. 알고리즘 및 프로토콜 번호는 IANA(Internet Assigned Numbers Authority)에 의해 정의됩니다. `ipsecalgس` 명령은 IPsec에 대한 알고리즘 목록을 확장할 수 있도록 합니다.

알고리즘에 대한 자세한 내용은 [ipsecalgس\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 암호화 프레임워크에 대한 자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 11 장, “암호화 프레임워크\(개요\)”](#)를 참조하십시오.

## IPsec에 대한 보안 연결 데이터베이스

IPsec 보안 서비스에 대한 키 자료 정보는 보안 연결 데이터베이스(SADB)에서 유지 관리됩니다. SA(보안 연결)는 인바운드 패킷 및 아웃바운드 패킷을 보호합니다. SADB는 특수한 종류의 소켓을 통해 메시지를 보내는 사용자 프로세스 또는 여러 동시 작업 프로세스로 유지 관리됩니다. 이 SADB 유지 관리 방식은 [route\(7P\)](#) 매뉴얼 페이지에 설명된 방식과 유사합니다. `root` 역할만 데이터베이스에 액세스할 수 있습니다.

`in.iked` 데몬 및 `ipseckey` 명령은 `PF_KEY` 소켓 인터페이스를 사용하여 SADB를 유지 관리합니다. SADB가 요청 및 메시지를 처리하는 방법에 대한 자세한 내용은 [pf\\_key\(7P\)](#) 매뉴얼 페이지를 참조하십시오.



## IPsec에서 SA 생성을 위한 유틸리티

IKE 프로토콜은 IPv4 및 IPv6 데이터베이스에 대한 자동 키 관리를 제공합니다. IKE를 설정하는 방법에 대한 지침은 18 장, “IKE 구성(작업)”을 참조하십시오. 수동 키 입력 유틸리티는 `ipseckey` 명령이며, 이 명령은 [ipseckey\(1M\)](#) 매뉴얼 페이지에 설명되어 있습니다.

`ipseckey` 명령을 사용하여 SADB(보안 연결 데이터베이스)를 수동으로 채웁니다. 일반적으로 수동 SA 생성은 사정상 IKE를 사용할 수 없을 때 사용됩니다. 하지만 SPI 값이 고유한 경우 수동 SA 생성과 IKE를 동시에 사용할 수 있습니다.

`ipseckey` 명령은 키가 수동으로 또는 IKE로 추가되었는지 여부에 상관 없이 시스템에 알려진 모든 SA를 보는 데 사용할 수 있습니다. `ipseckey` 명령은 `-c` 옵션과 함께 인수로 제공하는 키 파일의 구문을 검사합니다.

`ipseckey` 명령으로 추가된 IPsec SA는 시스템을 재부트하면 없어집니다. 시스템 부트 시 수동으로 추가한 SA를 사용으로 설정하려면 항목을 `/etc/inet/secret/ipseckey` 파일에 추가한 다음 `svc:/network/ipsec/manual-key:default` 서비스를 사용으로 설정합니다. 절차는 231 페이지 “IPsec 키를 수동으로 만드는 방법”을 참조하십시오.

`ipseckey` 명령에는 제한된 수의 일반 옵션만 있지만 명령은 풍부한 명령 언어를 지원합니다. 수동 키 입력에 대한 프로그래밍 인터페이스로 해당 요청이 전달되도록 지정할 수 있습니다. 추가 정보는 [pf\\_key\(7P\)](#) 매뉴얼 페이지를 참조하십시오.

## ipseckey에 대한 보안 고려 사항

`ipseckey` 명령은 Network Security 또는 Network IPsec Management 권한 프로파일을 가진 역할이 민감한 암호화 키 입력 정보를 입력할 수 있도록 합니다. 공격자가 이 정보에 대한 액세스 권한을 획득할 경우 IPsec 트래픽의 보안을 침해할 수 있습니다.

---

주 - 가능한 경우 `ipseckey`로 수동 키 입력이 아닌 IKE를 사용합니다.

---

키 입력 자료를 처리하고 `ipseckey` 명령을 사용할 때 다음 사항을 고려해야 합니다.

- 키 입력 자료를 새로 고쳤습니까? 정기적인 키 새로 고침은 기본적인 보안 방식입니다. 키 새로 고침은 잠재적인 알고리즘 및 키 취약성으로부터 보호하고 노출된 키의 손상을 제한합니다.
- TTY가 네트워크를 통해 이동합니까? `ipseckey` 명령이 대화식 모드입니까?
  - 대화식 모드에서는 키 입력 자료의 보안이 이 TTY의 트래픽에 대한 네트워크 경로의 보안입니다. 일반 텍스트 텔넷 또는 `rlogin` 세션을 통해 `ipseckey` 명령을 사용하는 것을 피해야 합니다.
  - 로컬 창이라도 창 이벤트를 읽는 숨겨진 프로그램의 공격 대상이 될 수 있습니다.

- -f 옵션을 사용했습니까? 파일이 네트워크를 통해 액세스합니까? 파일을 누구나 읽을 수 있습니까?
- 공격자는 파일이 읽혀질 때 네트워크 마운트 파일을 읽을 수 있습니다. 키 입력 자료가 포함된 누구나 읽을 있는 파일 사용을 피해야 합니다.
- 이름 지정 시스템을 보호합니다. 다음 두 조건이 충족될 경우 호스트 이름을 더 이상 신뢰할 수 없습니다.
  - 소스 주소가 네트워크를 통해 조회할 수 있는 호스트입니다.
  - 이름 지정 시스템이 침해되었습니다.

보안 취약성은 실제 도구가 도구의 오용으로 인해 발생하기도 합니다. `ipseckey` 명령을 사용할 때는 주의해야 합니다. 가장 안전한 작업 모드를 위해서는 `ssh` 또는 콘솔 또는 기타 하드 연결된 TTY를 사용합니다.

## snoop 명령 및 IPsec

`snoop` 명령은 AH 및 ESP 헤더를 구문 분석할 수 있습니다. ESP는 데이터를 암호화하므로 `snoop` 명령은 ESP로 보호된 암호화된 헤더를 볼 수 없습니다. AH는 데이터를 암호화하지 않습니다. 따라서 AH로 보호된 트래픽은 `snoop` 명령으로 검사할 수 있습니다. 명령에 대한 -v 옵션은 AH가 패킷에서 언제 사용되었는지 표시합니다. 자세한 내용은 [snoop\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

보호된 패킷에 대한 상세 정보 `snoop` 출력의 예는 [235 페이지](#) “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

이 릴리스에 포함된 무료 오픈 소스 소프트웨어인 [Wireshark](http://www.wireshark.org/about.html) (<http://www.wireshark.org/about.html>) 등의 타사 네트워크 분석기도 사용할 수 있습니다.

## Internet Key Exchange(개요)

---

IKE(Internet Key Exchange)는 IPsec의 키 관리를 자동화합니다. Oracle Solaris는 IKEv1을 구현합니다. 이 장은 IKE에 대한 다음 정보를 포함합니다.

- 243 페이지 “IKE로 키 관리”
- 244 페이지 “IKE 키 협상”
- 245 페이지 “IKE 구성 선택”
- 246 페이지 “IKE 유틸리티 및 파일”

IKE 구현 지침은 18 장, “IKE 구성(작업)”을 참조하십시오. 참고 사항은 19 장, “Internet Key Exchange(참조)”를 참조하십시오. IPsec에 대한 내용은 14 장, “IP 보안 아키텍처(개요)”를 참조하십시오.

### IKE로 키 관리

IPsec 보안 연관(SA)에 대한 키 관련 자료를 관리하는 것을 **키 관리**라고 합니다. 자동 키 관리를 위해서는 키 생성, 인증, 교환을 위한 통신 보안 채널이 필요합니다. Oracle Solaris는 IKE(Internet Key Exchange) 버전 1을 사용하여 키 관리를 자동화합니다. IKE는 대용량 트래픽에 보안 채널을 제공하도록 쉽게 확장됩니다. IPv4 및 IPv6 패킷의 IPsec SA는 IKE를 활용할 수 있습니다.

IKE는 사용 가능한 하드웨어 가속 및 하드웨어 저장소를 활용할 수 있습니다. 하드웨어 가속기를 사용하여 집중적인 키 작업을 시스템에서 처리할 수 있습니다. 하드웨어의 키 저장소는 추가적 보호 계층을 제공합니다.

## IKE 키 협상

IKE 데몬 `in.iked`는 IPsec SA에 대한 키 관련 자료를 안전한 방식으로 협상하고 인증합니다. 데몬은 OS에서 제공된 내부 함수에서 키의 무작위 시드를 사용합니다. IKE는 PFS(완전 순방향 비밀성)를 제공합니다. PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송 키를 만드는 데 사용된 시드는 재사용되지 않습니다. [in.iked\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## IKE 키 용어

다음 표는 키 협상에 사용되는 용어를 나열하고 흔히 사용되는 머리글자어를 제공하며 각 용어에 대한 정의 및 사용을 제시합니다.

표 17-1 키 협상 용어, 머리글자어 및 사용

키 협상 용어	머리글자어	정의 및 사용
키 교환		비대칭 암호화 알고리즘에 대한 키를 생성하는 프로세스입니다. 두 가지 주요 방법은 RSA 및 Diffie-Hellman 프로토콜입니다.
Diffie-Hellman 알고리즘	DH	키 생성 및 키 인증을 제공하는 키 교환 알고리즘입니다. <b>인증된 키 교환</b> 이라고도 합니다.
RSA 알고리즘	RSA	키 생성 및 키 전송을 제공하는 키 교환 알고리즘입니다. 프로토콜 이름은 Rivest, Shamir, Adleman 등 3인의 저작자 이름에서 따왔습니다.
완전 순방향 비밀성	PFS	인증된 키 교환에만 적용됩니다. PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송을 보호하는 키의 소스도 추가 키를 파생하는 데 사용되지 않습니다.
Oakley 그룹		안전한 방식으로 Phase 2의 키를 설정하는 방법입니다. Oakley 방법은 PFS를 협상하는 데 사용됩니다.

## IKE Phase 1 교환

Phase 1 교환을 **기본 모드**라고 합니다. Phase 1 교환에서 IKE는 공개 키 암호화 방법을 사용하여 피어 IKE 엔티티로 자체 인증합니다. 그 결과는 ISAKMP(Internet Security Association and Key Management Protocol) 보안 연관(SA)입니다. ISAKMP SA는 IP 데이터그램에 대한 키 관련 자료를 협상하기 위한 IKE의 보안 채널입니다. IPsec SA와 달리, ISAKMP SA는 양방향이므로 하나의 보안 연관만 필요합니다.

IKE가 Phase 1 교환에서 키 관련 자료를 협상하는 방법을 구성할 수 있습니다. IKE는 `/etc/inet/ike/config` 파일에서 구성 정보를 읽습니다. 구성 정보는 다음과 같습니다.

- 공개 키 인증서 이름과 같은 전역 매개변수
- PFS(완전 순방향 비밀성)의 사용 여부

- 영향을 받는 인터페이스
- 보안 프로토콜 및 해당 알고리즘
- 인증 방법

두 가지 인증 방법은 미리 공유한 키와 공개 키 인증서입니다. 공개 키 인증서는 자체 서명할 수 있습니다. 또는 공개 키 기반구조(PKI) 조직에서 **인증 기관(CA)**에 의해 인증서를 발행할 수 있습니다.

## IKE Phase 2 교환

Phase 2 교환을 **빠른 모드**라고 합니다. Phase 2 교환에서 IKE는 IKE 데몬을 실행 중인 시스템 간에 IPsec SA를 만들고 관리합니다. IKE는 Phase 1 교환에서 만든 보안 채널을 사용하여 키 관련 자료의 전송을 보호합니다. IKE 데몬은 `/dev/random` 장치를 사용하여 난수 생성기로부터 키를 만듭니다. 데몬이 구성 가능한 비율로 키를 새로 고칩니다. IPsec 정책용 구성 파일인 `ipsecinit.conf`에 지정된 알고리즘에서 키 관련 자료를 사용할 수 있습니다.

## IKE 구성 선택

`/etc/inet/ike/config` 구성 파일은 IKE 정책 항목을 포함합니다. 두 IKE 데몬이 서로 인증하려면 항목이 유효해야 합니다. 또한 키 관련 자료를 사용할 수 있어야 합니다. 구성 파일의 항목에 따라 키 관련 자료를 사용하여 Phase 1 교환을 인증하는 방법이 결정됩니다. 미리 공유한 키 또는 공개 키 인증서를 선택할 수 있습니다.

`auth_method preshared` 항목은 미리 공유한 키가 사용됨을 나타냅니다. `preshared`가 아닌 `auth_method`의 값은 공개 키 인증서가 사용될지 나타냅니다. 공개 키 인증서를 자체 서명할 수도 있고, PKI 조직에서 인증서를 설치할 수도 있습니다. 자세한 내용은 [ike.config\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## IKE와 미리 공유한 키 인증

미리 공유한 키는 두 피어 시스템을 인증하는 데 사용됩니다. 미리 공유한 키는 한 시스템에서 관리자가 만든 16진수 또는 ASCII 문자열입니다. 그런 다음 안전한 방식으로 피어 시스템의 관리자와 키를 공유합니다. 악의적 사용자가 미리 공유한 키를 가로채면 피어 시스템 중 하나로 가장할 수 있습니다.

이 인증 방법을 사용하는 피어에서 미리 공유한 키는 동일해야 합니다. 키는 특정 IP 주소 또는 주소 범위에 묶입니다. 각 시스템의 `/etc/inet/secret/ike.preshared` 파일에 키가 놓입니다. 자세한 내용은 [ike.preshared\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## IKE와 공개 키 인증서

공개 키 인증서를 사용하면 통신 시스템이 대역 외에서 보안 키 관련 자료를 공유할 필요가 없습니다. 공개 키는 키 인증 및 협상을 위해 [Diffie-Hellman 알고리즘\(DH\)](#)을 사용합니다. 공개 키 인증서는 두 종류로 나뉩니다. 인증서를 자체 서명할 수도 있고, [인증 기관\(CA\)](#)에서 인증서를 공인할 수도 있습니다.

자체 서명된 공개 키 인증서는 관리자 스스로 만듭니다. `ikecert certlocal -ks` 명령은 시스템의 공개-개인 키 쌍 중 개인 부분을 만듭니다. 그런 다음 원격 시스템에서 X.509 형식의 자체 서명된 인증서 출력을 가져옵니다. 키 쌍의 공개 부분을 위해 원격 시스템의 인증서가 `ikecert certdb` 명령에 입력됩니다. 자체 서명된 인증서는 통신 시스템의 `/etc/inet/ike/publickeys` 디렉토리에 상주합니다. `-T` 옵션을 사용하면 인증서가 연결된 하드웨어에 상주합니다.

자체 서명된 인증서는 미리 공유한 키와 CA 사이의 중간 지점입니다. 미리 공유한 키와 달리, 자체 서명된 인증서는 모바일 시스템이나 번호를 다시 매길 수 있는 시스템에서 사용할 수 있습니다. 고정 번호 없이 시스템에 인증서를 자체 서명하려면 `DNS(www.example.org)` 또는 `email(root@domain.org)` 대체 이름을 사용하십시오.

PKI 또는 CA 조직에서 공개 키를 전달할 수 있습니다. `/etc/inet/ike/publickeys` 디렉토리에 공개 키와 동반 CA를 설치합니다. `-T` 옵션을 사용하면 인증서가 연결된 하드웨어에 상주합니다. 또한 공급업체가 CRL(인증서 해지 목록)을 발행합니다. 관리자는 키 및 CA 설치와 함께 `/etc/inet/ike/crls` 디렉토리에 CRL을 설치할 책임이 있습니다.

CA는 사이트 관리자가 아닌 외부 조직에서 공인된다는 장점이 있습니다. 어떤 의미에서 CA는 공증된 인증서입니다. 자체 서명된 인증서와 마찬가지로, CA는 모바일 시스템이나 번호를 다시 매길 수 있는 시스템에서 사용할 수 있습니다. 자체 서명된 인증서와 달리, CA는 많은 수의 통신 시스템을 보호하도록 매우 쉽게 확장할 수 있습니다.

## IKE 유틸리티 및 파일

다음 표는 IKE 정책의 구성 파일, IKE 키의 저장소 위치 및 IKE를 구현하는 다양한 명령과 서비스를 요약합니다. 서비스에 대한 자세한 내용은 [Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리\(개요\)”](#)를 참조하십시오.

표 17-2 IKE 구성 파일, 키 저장소 위치, 명령 및 서비스

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
<code>svc:/network/ipsec/ike</code>	IKE를 관리하는 SMF 서비스입니다.	<a href="#">smf(5)</a>
<code>/usr/lib/inet/in.iked</code>	IKE(Internet Key Exchange) 데몬입니다. <code>ike</code> 서비스를 사용으로 설정할 때 자동화된 키 관리를 활성화합니다.	<a href="#">in.iked(1M)</a>

표 17-2 IKE 구성 파일, 키 저장소 위치, 명령 및 서비스 (계속)

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
/usr/sbin/ikeadm	IKE 정책을 확인하고 일시적으로 수정하기 위한 IKE 관리 명령입니다. Phase 1 알고리즘과 같은 IKE 관리 객체와 사용 가능한 Diffie-Hellman 그룹을 볼 수 있습니다.	<a href="#">ikeadm(1M)</a>
/usr/sbin/ikecert	공개 키 인증서를 보유하는 로컬 데이터베이스를 조작하기 위한 인증서 데이터베이스 관리 명령입니다. 데이터베이스를 연결된 하드웨어에 저장할 수도 있습니다.	<a href="#">ikecert(1M)</a>
/etc/inet/ike/config	IKE 정책의 기본 구성 파일입니다. 인바운드 IKE 요청을 일치시키고 아웃바운드 IKE 요청을 준비하기 위한 사이트 규칙을 포함합니다.  이 파일이 존재하면 <code>ike</code> 서비스를 사용으로 설정할 때 <code>in.iked</code> 데몬을 시작합니다. 이 파일의 위치는 <code>svccfg</code> 명령으로 변경할 수 있습니다.	<a href="#">ike.config(4)</a>
ike.preshared	/etc/inet/secret 디렉토리의 미리 공유한 키 파일입니다. Phase 1 교환에서 인증을 위한 보안 키 관련 자료를 포함합니다. 미리 공유한 키로 IKE를 구성할 때 사용됩니다.	<a href="#">ike.preshared(4)</a>
ike.privatekeys	/etc/inet/secret 디렉토리의 개인 키 디렉토리입니다. 공개-개인 키 쌍의 일부인 개인 키를 포함합니다.	<a href="#">ikecert(1M)</a>
publickeys 디렉토리	공개 키 및 인증서 파일을 보유하는 /etc/inet/ike 디렉토리 안의 디렉토리입니다. 공개-개인 키 쌍 중 공개 키 부분을 포함합니다.	<a href="#">ikecert(1M)</a>
crls 디렉토리	공개 키 및 인증서 파일에 대한 해지 목록을 보유하는 /etc/inet/ike 디렉토리 안의 디렉토리입니다.	<a href="#">ikecert(1M)</a>
Sun Crypto Accelerator 6000 보드	운영 체제에서 작업 부담을 덜어서 공개 키 작업을 가속화하는 하드웨어입니다. 또한 공개 키, 개인 키 및 공개 키 인증서를 저장합니다. Sun Crypto Accelerator 6000 보드는 레벨 3의 FIPS 140-2 공인 장치입니다.	<a href="#">ikecert(1M)</a>





## IKE 구성(작업)

---

이 장에서는 시스템의 인터넷 키 교환(IKE) 구성 방법에 대해 설명합니다. IKE가 구성되면 네트워크의 IPsec에 대한 키 입력 도구가 자동으로 생성됩니다. 이 장은 다음 정보를 포함합니다.

- 249 페이지 “IKE 정보 표시”
- 251 페이지 “IKE 구성(작업 맵)”
- 251 페이지 “미리 공유한 키로 IKE 구성(작업 맵)”
- 256 페이지 “공개 키 인증서로 IKE 구성(작업 맵)”
- 272 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”
- 279 페이지 “연결된 하드웨어를 찾도록 IKE 구성”

IKE에 대한 개요 정보는 17 장, “Internet Key Exchange(개요)”을 참조하십시오. IKE에 대한 참조 정보는 19 장, “Internet Key Exchange(참조)”을 참조하십시오. 자세한 절차는 [ikeadm\(1M\)](#), [ikecert\(1M\)](#) 및 [ike.config\(4\)](#) 매뉴얼 페이지의 Examples 절을 참조하십시오.

### IKE 정보 표시

1단계 IKE 협상에서 사용할 수 있는 알고리즘 및 그룹을 확인할 수 있습니다.

#### ▼ 1단계 IKE 교환에 사용 가능한 그룹 및 알고리즘 표시 방법

이 절차에서는 1단계 IKE 교환에 사용 가능한 Diffie-Hellman 그룹을 결정합니다. 또한 IKE 1단계 교환에 사용 가능한 암호화 및 인증 알고리즘을 확인할 수 있습니다. 숫자 값은 [IANA](#)(Internet Assigned Numbers Authority)에서 해당 알고리즘에 대해 지정한 값과 일치합니다.

### 1 IKE가 1단계에서 사용할 수 있는 Diffie-Hellman 그룹 목록을 표시합니다.

Diffie-Hellman 그룹이 IKE SA를 설정합니다.

```
# ikeadm dump groups
Value Strength Description
1      66      ietf-ike-grp-modp-768
2      77      ietf-ike-grp-modp-1024
5      91      ietf-ike-grp-modp-1536
14     110     ietf-ike-grp-modp-2048
15     130     ietf-ike-grp-modp-3072
16     150     ietf-ike-grp-modp-4096
17     170     ietf-ike-grp-modp-6144
18     190     ietf-ike-grp-modp-8192
```

Completed dump of groups

다음과 같이 IKE 1단계 변환에서 이러한 값 중 하나를 oakley\_group 매개변수에 대한 인수로 사용합니다.

```
pl_xform
{ auth_method preshared oakley_group 15 auth_alg sha encr_alg aes }
```

### 2 IKE가 1단계에서 사용할 수 있는 인증 알고리즘 목록을 표시합니다.

```
# ikeadm dump authalgs
Value Name
1      md5
2      sha1
4      sha256
5      sha384
6      sha512
```

Completed dump of authalgs

다음과 같이 IKE 1단계 변환에서 이러한 이름 중 하나를 auth\_alg 매개변수에 대한 인수로 사용합니다.

```
pl_xform
{ auth_method preshared oakley_group 15 auth_alg sha256 encr_alg 3des }
```

### 3 IKE가 1단계에서 사용할 수 있는 암호화 알고리즘 목록을 표시합니다.

```
# ikeadm dump encralgs
Value Name
3      blowfish-cbc
5      3des-cbc
1      des-cbc
7      aes-cbc
```

Completed dump of encralgs

다음과 같이 IKE 1단계 변환에서 이러한 이름 중 하나를 encr\_alg 매개변수에 대한 인수로 사용합니다.

```
pl_xform
{ auth_method preshared oakley_group 15 auth_alg sha256 encr_alg aes }
```

**참조** 이러한 값이 필요한 IKE 규칙을 구성하는 작업은 251 페이지 “IKE 구성(작업 맵)”을 참조하십시오.

## IKE 구성(작업 맵)

미리 공유한 키, 자체 서명된 인증서 및 인증 기관(CA)의 인증서를 사용하여 IKE를 인증할 수 있습니다. 규칙은 보호되고 있는 끝점에 특정 IKE 인증 방법을 연결합니다. 따라서 시스템에서 IKE 인증 방법 중 하나 또는 전체를 사용할 수 있습니다. PKCS #11 라이브러리에 대한 포인터를 통해 IKE는 연결된 하드웨어 가속기를 사용할 수 있습니다.

IKE를 구성한 후에는 IKE 구성을 사용하는 IPsec 작업을 완료합니다. 다음 표에서는 특정 IKE 구성을 중점적으로 다루는 작업 맵에 대해 설명합니다.

작업	설명	수행 방법
미리 공유한 키로 IKE를 구성합니다.	시스템이 보안 키를 공유함으로써 두 시스템 간의 통신을 보호합니다.	251 페이지 “미리 공유한 키로 IKE 구성(작업 맵)”
공개 키 인증서로 IKE를 구성합니다.	공개 키 인증서로 통신을 보호합니다. 인증서는 자체 서명될 수도 있고, PKI 조직에 의해 보장될 수도 있습니다.	256 페이지 “공개 키 인증서로 IKE 구성(작업 맵)”
NAT 경계를 벗어납니다.	모바일 시스템과 통신하도록 IPsec 및 IKE를 구성합니다.	272 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”
하드웨어 키 저장소를 사용하여 인증서 쌍을 생성하도록 IKE를 구성합니다.	Sun Crypto Accelerator 6000 보드가 IKE 작업 속도를 향상시키고 공개 키 인증서를 저장할 수 있도록 합니다.	279 페이지 “연결된 하드웨어를 찾도록 IKE 구성”

## 미리 공유한 키로 IKE 구성(작업 맵)

다음 표에서는 미리 공유한 키로 IKE를 구성 및 유지 관리하는 절차에 대해 설명합니다.

작업	설명	수행 방법
미리 공유한 키로 IKE를 구성합니다.	IKE 구성 파일과 공유할 키 하나를 만듭니다.	252 페이지 “미리 공유한 키로 IKE를 구성하는 방법”
실행 중인 IKE 시스템에 미리 공유한 키를 추가합니다.	현재 IKE 정책을 적용 중인 시스템에 새 IKE 정책 항목 및 새 키 입력 도구를 추가합니다.	254 페이지 “새 피어 시스템에 대한 IKE 업데이트 방법”

## 미리 공유한 키로 IKE 구성

미리 공유한 키는 가장 간단한 IKE 인증 방법입니다. IKE를 사용하도록 피어 시스템을 구성 중이며, 해당 시스템의 관리자라면 미리 공유한 키를 사용하는 것이 좋습니다. 단, 공개 키 인증서와 달리 미리 공유한 키는 IP 주소와 연관되어 있습니다. 미리 공유한 키를 특정 IP 주소 또는 IP 주소 범위와 연관시킬 수 있습니다. 번호 재지정이 지정된 IP 주소 범위에 속하지 않을 경우, 번호가 재지정될 수 있는 모바일 시스템이나 시스템에서는 미리 공유한 키를 사용할 수 없습니다.

### ▼ 미리 공유한 키로 IKE를 구성하는 방법

IKE 구현은 키 길이가 다양한 알고리즘을 제공합니다. 키 길이는 사이트 보안에 따라 선택할 수 있습니다. 일반적으로 길이가 긴 키는 길이가 짧은 키에 비해 더 강력한 보안을 제공합니다.

이 절차에서는 ASCII 형식으로 키를 생성합니다.

이 절차에서는 `enigma` 및 `partym` 시스템 이름을 사용합니다. `enigma` 및 `partym` 이름을 사용자의 현재 시스템 이름으로 대체하십시오.

---

주 - Trusted Extensions 시스템에서 레이블이 있는 IPsec를 사용하려면 [Trusted Extensions 구성 및 관리의 “다중 레벨 Trusted Extensions 네트워크에서 IPsec 보호를 적용하는 방법”](#)을 참조하십시오.

---

#### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

#### 2 각 시스템에서 `/etc/inet/ike/config` 파일을 만듭니다.

`/etc/inet/ike/config.sample`을 템플릿으로 사용할 수 있습니다.

#### 3 각 시스템의 `ike/config` 파일에 규칙 및 전역 매개변수를 입력합니다.

이 파일의 규칙 및 전역 매개변수는 시스템의 `ipsecinit.conf` 파일에 설정되어 있는 IPsec 정책이 성공하도록 허용해야 합니다. 다음 IKE 구성 예는 [218 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”](#)의 `ipsecinit.conf` 예와 함께 작동합니다.

##### a. 예를 들어, `enigma` 시스템에서 `/etc/inet/ike/config` 파일을 수정합니다.

```
### ike/config file on enigma, 192.168.116.16

## Global parameters
```

```
#
## Defaults that individual rules can override.
p1_xform
{ auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2
#
## The rule to communicate with partym
# Label must be unique
{ label "enigma-partym"
  local_addr 192.168.116.16
  remote_addr 192.168.13.213
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha256 encr_alg aes }
  p2_pfs 5
}
```

**b. partym 시스템에서 /etc/inet/ike/config 파일을 수정합니다.**

```
### ike/config file on partym, 192.168.13.213
## Global Parameters
#
p1_xform
{ auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2

## The rule to communicate with enigma
# Label must be unique
{ label "partym-enigma"
  local_addr 192.168.13.213
  remote_addr 192.168.116.16
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha256 encr_alg aes }
  p2_pfs 5
}
```

**4 각 시스템에서 파일의 구문을 확인합니다.**

```
# /usr/lib/inet/in.iked -c -f /etc/inet/ike/config
```

**5 각 시스템에서 /etc/inet/secret/ike.preshared 파일을 만듭니다.**

각 파일에 미리 공유한 키를 삽입합니다.

**a. 예를 들어, enigma 시스템에서 ike.preshared 파일이 다음과 유사하게 표시됩니다.**

```
# ike.preshared on enigma, 192.168.116.16
#...
{ localidtype IP
  localid 192.168.116.16
  remoteidtype IP
  remoteid 192.168.13.213
  # The preshared key can also be represented in hex
  # as in 0xf47cb0f432e14480951095f82b
  # key "This is an ASCII Cqret phrAz, use str0ng p@ssword tekniqes"
}
```

**b. partym 시스템에서 ike.preshared 파일이 다음과 유사하게 표시됩니다.**

```
# ike.preshared on partym, 192.168.13.213
#...
{ localidtype IP
  localid 192.168.13.213
  remoteidtype IP
  remoteid 192.168.116.16
  # The preshared key can also be represented in hex
# as in 0xf47cb0f432e14480951095f82b
  key "This is an ASCII Cqret phrAz, use str0ng p@ssword tekniques"
}
```

**6 IKE 서비스를 사용으로 설정합니다.**

```
# svcadm enable ipsec/ike
```

**예 18-1 IKE 미리 공유한 키 새로 고침**

IKE 관리자가 미리 공유한 키를 새로 고치려고 할 경우, 피어 시스템에서 이 파일을 편집하고 in.iked 데몬을 다시 시작합니다.

먼저 관리자는 192.168.13.0/24 서브넷의 호스트에 유효한 미리 공유한 키 항목을 추가합니다.

```
#...
{ localidtype IP
  localid 192.168.116.0/24
  remoteidtype IP
  remoteid 192.168.13.0/24
  # Enigma and partym's shared passphrase for keying material
key "LOooong key Th@t m^st Be Ch*angEd \"reguLarLy)"
}
```

그런 다음 관리자는 모든 시스템에서 IKE 서비스를 다시 시작합니다.

```
# svcadm enable ipsec/ike
```

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

**▼ 새 피어 시스템에 대한 IKE 업데이트 방법**

같은 피어 간의 작업 구성에 IPsec 정책 항목을 추가할 경우에는 IPsec 정책 서비스를 새로 고쳐야 합니다. IKE는 재구성하거나 다시 시작하지 않아도 됩니다.

IPsec 정책에 새 피어를 추가할 경우 IPsec 변경 외에 IKE 구성도 수정해야 합니다.

**시작하기 전에** ipsecinit.conf 파일을 업데이트했으며 피어 시스템에 대한 IPsec 정책을 새로 고쳤습니다.

## 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”**을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

## 2 IPsec를 사용 중인 새 시스템에 대한 키를 관리할 IKE 규칙을 만듭니다.

### a. 예를 들어, enigma 시스템에서 /etc/inet/ike/config 파일에 다음 규칙을 추가합니다.

```
### ike/config file on enigma, 192.168.116.16

## The rule to communicate with ada

{label "enigma-to-ada"
 local_addr 192.168.116.16
 remote_addr 192.168.15.7
 pl_xform
 {auth_method preshared oakley_group 5 auth_alg sha256 encr_alg aes}
 p2_pfs 5
}
```

### b. ada 시스템에서 다음 규칙을 추가합니다.

```
### ike/config file on ada, 192.168.15.7

## The rule to communicate with enigma

{label "ada-to-enigma"
 local_addr 192.168.15.7
 remote_addr 192.168.116.16
 pl_xform
 {auth_method preshared oakley_group 5 auth_alg sha256 encr_alg aes}
 p2_pfs 5
}
```

## 3 피어 시스템에 대해 IKE 미리 공유한 키를 만듭니다.

### a. enigma 시스템에서 /etc/inet/secret/ike.preshared 파일에 다음 정보를 추가합니다.

```
# ike.preshared on enigma for the ada interface
#
{ localidtype IP
  localid 192.168.116.16
  remoteidtype IP
  remoteid 192.168.15.7
  # enigma and ada's shared key
  key "Twas brillig and the slivey toves did *s0mEtHiNg* be CareFULL hEEEr"
}
```

### b. ada 시스템에서 ike.preshared 파일에 다음 정보를 추가합니다.

```
# ike.preshared on ada for the enigma interface
#
{ localidtype IP
```

```
localid 192.168.15.7
remoteidtype IP
remoteid 192.168.116.16
# ada and enigma's shared key
key "Twas brillig and the slivey toves did *s0mEthiNg* be CareFULL hEEEr"
}
```

**4 각 시스템에서 ike 서비스를 새로 고칩니다.**

```
# svcadm refresh ike
```

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## 공개 키 인증서로 IKE 구성(작업 맵)

다음 표에서는 IKE에 대한 공개 키 인증서를 만드는 절차에 대해 설명합니다. 이 절차에서는 인증서를 빠르게 만들고 연결된 하드웨어에 저장하는 방법을 설명합니다.

공개 인증서는 고유해야 하므로 공개 키 인증서 작성자는 인증서의 이름을 임의적으로 고유한 이름으로 생성합니다. 일반적으로 X.509 식별 이름이 사용됩니다. 식별을 위해 대체 이름을 사용할 수도 있습니다. 이러한 이름의 형식은 *tag=value*입니다. 이 값은 임의적이지만 값의 형식은 태그 유형에 적합해야 합니다. 예를 들어, email 태그의 형식은 *name@domain.suffix*입니다.

작업	설명	수행 방법
자체 서명된 공개 키 인증서로 IKE를 구성합니다.	다음 두 개의 인증서를 만들어 각 시스템에 배치합니다. <ul style="list-style-type: none"><li>■ 자체 서명된 인증서</li><li>■ 피어 시스템의 공개 키 인증서</li></ul>	257 페이지 “자체 서명된 공개 키 인증서로 IKE를 구성하는 방법”
PKI 인증 기관으로 IKE를 구성합니다.	인증서 요청을 만들고 각 시스템에 다음 세 개의 인증서를 배치합니다. <ul style="list-style-type: none"><li>■ 인증 기관(CA)이 요청에 따라 만든 인증서</li><li>■ CA의 공개 키 인증서</li><li>■ CA의 CRL</li></ul>	262 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”
로컬 하드웨어에서 공개 키 인증서를 구성합니다.	다음 작업 중 하나를 수행합니다. <ul style="list-style-type: none"><li>■ 로컬 하드웨어에서 자체 서명된 인증서를 생성한 다음 원격 시스템의 공개 키를 하드웨어에 추가합니다.</li><li>■ 로컬 하드웨어에서 인증서 요청을 생성한 다음 CA의 공개 키 인증서를 하드웨어에 추가합니다.</li></ul>	266 페이지 “공개 키 인증서를 생성하여 하드웨어에 저장하는 방법”



작업	설명	수행 방법
PKI에서 인증서 해지 목록(CRL)을 업데이트합니다.	중앙 배포 지점에서 CRL에 액세스합니다.	270 페이지 “인증서 해지 목록 처리 방법”

주 - Trusted Extensions 시스템에서 패킷 및 IKE 협상에 레이블을 지정하려면 **Trusted Extensions 구성 및 관리**의 “레이블이 있는 IPsec 구성(작업 맵)”의 절차를 따르십시오.

공개 키 인증서는 Trusted Extensions 시스템의 전역 영역에서 관리됩니다. Trusted Extensions는 인증서 관리 및 저장 방법을 변경하지 않습니다.

## 공개 키 인증서로 IKE 구성

공개 키 인증서를 사용하면 통신하는 시스템이 대역 외 연결에서 보안 키 입력 도구를 공유할 필요가 없습니다. 미리 공유한 키와 달리 공개 키 인증서는 모바일 시스템 또는 번호가 재지정될 수 있는 시스템에서 사용할 수 있습니다.

또한 공개 키 인증서를 생성하여 연결된 하드웨어에 저장할 수 있습니다. 절차는 279 페이지 “연결된 하드웨어를 찾도록 IKE 구성”을 참조하십시오.

### ▼ 자체 서명된 공개 키 인증서로 IKE를 구성하는 방법

이 절차에서는 인증서 쌍을 만듭니다. 개인 키는 로컬 인증서 데이터베이스의 디스크에 저장되며 `certlocal` 하위 명령을 사용하여 참조할 수 있습니다. 인증서 쌍의 공개 부분은 공개 인증서 데이터베이스에 저장됩니다. 이는 `certdb` 하위 명령을 사용하여 참조할 수 있습니다. 피어 시스템과 공개 부분을 교환합니다. 두 인증서의 조합은 IKE 전송 인증에 사용됩니다.

자체 서명된 인증서는 CA의 공개 인증서보다 오버헤드가 적지만 확장이 어렵습니다. CA에서 발급한 인증서와 달리 자체 서명된 인증서는 대역 외 연결에서 확인해야 합니다.

#### 1 관리자 로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 “관리 권한을 얻는 방법”을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

#### 2 `ike.privatekeys` 데이터베이스에 자체 서명된 인증서를 만듭니다.

```
# ikecert certlocal -ks -m keysize -t keytype \
-D dname -A altname \
[-S validity-start-time] [-F validity-end-time] [-T token-ID]
-ks
```

자체 서명된 인증서를 만듭니다.

-m <i>keysize</i>	키의 크기입니다. <i>keysize</i> 는 512, 1024, 2048, 3072 또는 4096일 수 있습니다.
-t <i>keytype</i>	사용할 알고리즘의 유형을 지정합니다. <i>keytype</i> 은 <i>rsa-sha1</i> , <i>rsa-md5</i> 또는 <i>dsa-sha1</i> 일 수 있습니다.
-D <i>dname</i>	인증서 주체에 대한 X.509 식별 이름입니다. 일반적으로 <i>dname</i> 의 형식은 <i>C=country</i> , <i>O=organization</i> , <i>OU=organizational unit</i> , <i>CN=common name</i> 입니다. 유효한 태그는 C, O, OU 및 CN입니다.
-A <i>altname</i>	인증서의 대체 이름입니다. <i>altname</i> 의 형식은 <i>tag=value</i> 입니다. 유효한 태그는 IP, DNS, email 및 DN입니다.
-S <i>validity-start-time</i>	인증서 시작 시간을 유효한 절대 또는 상대 시작 시간으로 지정합니다.
-F <i>validity-end-time</i>	인증서 종료 시간을 유효한 절대 또는 상대 종료 시간으로 지정합니다.
-T <i>token-ID</i>	PKCS #11 하드웨어 토큰이 키를 생성할 수 있도록 합니다. 그러면 인증서가 하드웨어에 저장됩니다.

**a. 예를 들어, *partym* 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha1 \
-D "O=exampleco, OU=IT, C=US, CN=partym" \
-A IP=192.168.13.213
Creating private key.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEfdZgKjANBgkqhkiG9w0BAQUFADAAMRgwFgYDVQQDEw9T
a...+
zBGi4QkNdI3f
-----END X509 CERTIFICATE-----
```

---

주 - -D 및 -A 옵션의 값은 임의의 값입니다. 이 값은 인증서를 식별하는 데만 사용됩니다. 192.168.13.213 등의 시스템을 식별하는 데는 사용되지 않습니다. 실제로 이러한 값은 고유하므로 피어 시스템에 올바른 인증서가 설치되어 있는지 대역 외 연결에서 확인해야 합니다.

---

**b. *enigma* 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha1 \
-D "O=exampleco, OU=IT, C=US, CN=enigma" \
-A IP=192.168.116.16
Creating private key.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEB15JnjANBgkqhkiG9w0BAQUFADAAMRgwFgYDVQQDEw9T
...
y85m6LHJYtC6
-----END X509 CERTIFICATE-----
```

### 3 인증서를 저장하여 원격 시스템으로 보냅니다.

출력은 인증서 공개 부분의 인코딩된 버전입니다. 이 인증서는 전자 메일에 안전하게 첨부할 수 있습니다. 수신자는 **단계 b**와 같이 올바른 인증서를 설치했는지 대역 외 연결에서 확인해야 합니다.

#### a. 예를 들어, **partym** 인증서의 공개 부분을 **enigma** 관리자에게 보냅니다.

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEfdZgKjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
a...+
zBGi4QkNdI3f
-----END X509 CERTIFICATE-----
```

#### b. **enigma** 관리자로부터 **enigma** 인증서의 공개 부분을 받습니다.

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEBl5JnjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
...
y85m6LHJYtC6
-----END X509 CERTIFICATE-----
```

### 4 각 시스템에서 공개 키 데이터베이스에 수신한 인증서를 추가합니다.

#### a. **root**가 읽을 수 있는 파일에 관리자의 전자 메일을 저장합니다.

#### b. **ikecert** 명령에 파일을 재지정합니다.

```
# ikcert certdb -a < /tmp/certificate.eml
```

이 명령은 BEGIN 태그와 END 태그 사이의 텍스트를 가져옵니다.

### 5 다른 관리자가 이 인증서를 보낸 것인지 해당 관리자에게 확인합니다.

예를 들어, 다른 관리자와 전화 통화를 통해 수신한 공개 인증서의 해시가 해당 관리자만 가진 개인 인증서의 해시와 일치하는지 확인할 수 있습니다.

#### a. **partym**에 저장된 인증서를 나열합니다.

다음 예에서 Note 1은 슬롯 0에 있는 인증서의 식별 이름(DN)을 나타냅니다. 슬롯 0에 있는 개인 인증서가 동일한 해시를 가지므로 이러한 인증서는 동일한 인증서 쌍입니다. 공개 인증서가 작동하려면 일치 쌍이 있어야 합니다. **certdb** 하위 명령은 공개 부분을 나열하며 **certlocal** 하위 명령은 개인 부분을 나열합니다.

```
partym # ikcert certdb -l
```

```
Certificate Slot Name: 0    Key Type: rsa
(Private key in certlocal slot 0)
Subject Name: <0=exampleco, OU=IT, C=US, CN=partym>    Note 1
Key Size: 2048
```

Public key hash: **80829EC52FC5BA910F4764076C20FDCF**

Certificate Slot Name: 1 Key Type: rsa  
(Private key in certlocal slot 1)  
Subject Name: <O=exampleco, OU=IT, C=US, CN=Ada>  
Key Size: 2048  
Public key hash: FEA65C5387BBF3B2C8F16C019FEB388

partym # **ikecert certlocal -l**

Local ID Slot Name: 0 Key Type: rsa  
Key Size: 2048  
Public key hash: **80829EC52FC5BA910F4764076C20FDCF** *Note 3*

Local ID Slot Name: 1 Key Type: rsa-sha1  
Key Size: 2048  
Public key hash: FEA65C5387BBF3B2C8F16C019FEB388

Local ID Slot Name: 2 Key Type: rsa  
Key Size: 2048  
Public key hash: 2239A6A127F88EE0CB40F7C24A65B818

이 검사에서 partym 시스템에 유효한 인증서 쌍이 있는 것이 확인되었습니다.

**b. enigma 시스템에 partym의 공개 인증서가 있는지 확인합니다.**

전화를 통해 공개 키 해시를 확인할 수 있습니다.

이전 단계에서 확인된 partym의 Note 3 해시를 enigma의 Note 4와 비교합니다.

enigma # **ikecert certdb -l**

Certificate Slot Name: 0 Key Type: rsa  
(Private key in certlocal slot 0)  
Subject Name: <O=exampleco, OU=IT, C=US, CN=Ada>  
Key Size: 2048  
Public key hash: 2239A6A127F88EE0CB40F7C24A65B818

Certificate Slot Name: 1 Key Type: rsa  
(Private key in certlocal slot 1)  
Subject Name: <O=exampleco, OU=IT, C=US, CN=enigma>  
Key Size: 2048  
Public key hash: FEA65C5387BBF3B2C8F16C019FEB388

Certificate Slot Name: 2 Key Type: rsa  
(Private key in certlocal slot 2)  
Subject Name: <O=exampleco, OU=IT, C=US, CN=partym>  
Key Size: 2048  
Public key hash: **80829EC52FC5BA910F4764076C20FDCF** *Note 4*

enigma의 공개 인증서 데이터베이스에 저장된 마지막 인증서의 공개 키 해시 및 주체 이름이 이전 단계의 partym에 대한 개인 인증서의 해시와 일치합니다.

## 6 각 시스템에서 두 인증서를 인증합니다.

인증서가 인식되도록 `/etc/inet/ike/config` 파일을 편집합니다.

원격 시스템의 관리자가 `cert_trust`, `remote_addr` 및 `remote_id` 매개변수에 대한 값을 제공합니다.

### a. 예를 들어, `partym` 시스템에서 `ike/config` 파일은 다음과 유사하게 표시됩니다.

```
# Explicitly trust the self-signed certs
# that we verified out of band. The local certificate
# is implicitly trusted because we have access to the private key.

cert_trust "O=exampleco, OU=IT, C=US, CN=enigma"

# We could also use the Alternate name of the certificate,
# if it was created with one. In this example, the Alternate Name
# is in the format of an IP address:
# cert_trust "192.168.116.16"

## Parameters that may also show up in rules.

p1_xform
{ auth_method preshared oakley_group 5 auth_alg sha256 encr_alg 3des }
p2_pfs 5

{
  label "US-partym to JA-enigmax"
  local_id_type dn
  local_id "O=exampleco, OU=IT, C=US, CN=partym"
  remote_id "O=exampleco, OU=IT, C=US, CN=enigma"

  local_addr 192.168.13.213
  # We could explicitly enter the peer's IP address here, but we don't need
  # to do this with certificates, so use a wildcard address. The wildcard
  # allows the remote device to be mobile or behind a NAT box
  remote_addr 0.0.0.0/0

  p1_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha256 encr_alg aes}
}
```

### b. `enigma` 시스템의 `ike/config` 파일에서 로컬 매개변수에 대한 `enigma` 값을 추가합니다.

원격 매개변수의 경우 `partym` 값을 사용합니다. `label` 키워드가 로컬 시스템에서 고유한지 확인합니다.

```
...
{
  label "JA-enigmax to US-partym"
  local_id_type dn
  local_id "O=exampleco, OU=IT, C=US, CN=enigma"
  remote_id "O=exampleco, OU=IT, C=US, CN=partym"

  local_addr 192.168.116.16
  remote_addr 0.0.0.0/0
  ...
}
```

## 7 피어 시스템에서 IKE를 사용으로 설정합니다.

```
partym # svcadm enable ipsec/ike
enigma # svcadm enable ipsec/ike
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## ▼ CA가 서명한 인증서로 IKE를 구성하는 방법

인증 기관(CA)의 공개 인증서를 사용하려면 외부 조직과의 협상이 필요합니다. 간편한 인증서 확장을 통해 통신하는 여러 시스템을 보호할 수 있습니다.

### 1 관리자 로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

### 2 `ikecert certlocal -kc` 명령을 사용하여 인증서 요청을 만듭니다.

명령 인수에 대한 설명은 [단계 b in 257 페이지 “자체 서명된 공개 키 인증서로 IKE를 구성하는 방법”](#)을 참조하십시오.

```
# ikecert certlocal -kc -m keysize -t keytype \
-D dname -A altname
```

#### a. 예를 들어, 다음 명령은 partym 시스템에서 인증서 요청을 만듭니다.

```
# ikecert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany\, Inc., OU=US-Partym, CN=Partym" \
> -A "DN=C=US, O=PartyCompany\, Inc., OU=US-Partym"
Creating software private keys.
Writing private key to file /etc/inet/secret/ike.privatekeys/2.
Enabling external key providers - done.
Certificate Request:
  Proceeding with the signing operation.
  Certificate request generated successfully (.../publickeys/0)
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIBByJCCATMCAQAwUzELMAKGA1UEBhMCVVMxHTABBgNVBAoTFTEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zck080m09X
-----END CERTIFICATE REQUEST-----
```

#### b. 다음 명령은 enigma 시스템에서 인증서 요청을 만듭니다.

```
# ikecert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax, CN=Enigmax" \
> -A "DN=C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax"
Creating software private keys.
```

```
...
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
8qlqdjaStLGfhD00
-----END CERTIFICATE REQUEST-----
```

### 3 PKI 조직에 인증서 요청을 제출합니다.

PKI 조직에서 인증서 요청 제출 방법을 제공할 수 있습니다. 대부분 조직에는 제출 양식을 제공하는 웹 사이트가 있습니다. 양식을 사용하려면 제출이 적합한지 증명해야 합니다. 일반적으로 양식에 인증서 요청을 붙여 넣습니다. 요청을 확인한 조직에서는 다음 두 개의 인증서 객체와 해지된 인증서 목록을 발급합니다.

- 공개 키 인증서 - 이 인증서는 사용자가 해당 조직에 제출한 요청을 기반으로 합니다. 제출한 요청은 이 공개 키 인증서의 일부입니다. 인증서는 사용자를 고유하게 식별합니다.
- 인증 기관 - 조직의 서명입니다. CA는 공개 키 인증서가 적합한지 확인합니다.
- 인증서 해지 목록(CRL) - 조직에서 해지한 최신 인증서 목록입니다. CRL에 대한 액세스 권한이 공개 키 인증서에 포함된 경우 CRL이 인증서 객체로 별도로 전송되지 않습니다.

CRL에 대한 URI가 공개 키 인증서에 포함된 경우 IKE가 자동으로 CRL을 검색할 수 있습니다. 마찬가지로 DN(LDAP 서버의 디렉토리 이름) 항목이 공개 키 인증서에 포함된 경우 IKE가 지정된 LDAP 서버에서 CRL을 검색하여 캐시할 수 있습니다.

공개 키 인증서에 포함된 URI 및 포함된 DN 항목의 예는 [270 페이지 “인증서 해지 목록 처리 방법”](#)을 참조하십시오.

### 4 시스템에 각 인증서를 추가합니다.

`ikecert certdb -a`에 대한 `-a` 옵션은 붙여 넣은 객체를 시스템의 적합한 인증서 데이터베이스에 추가합니다. 자세한 내용은 [246 페이지 “IKE와 공개 키 인증서”](#)를 참조하십시오.

#### a. 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

#### b. PKI 조직에서 수신한 공개 키 인증서를 추가합니다.

```
# ikecert certdb -a < /tmp/PKIcert.eml
```

#### c. PKI 조직의 CA를 추가합니다.

```
# ikecert certdb -a < /tmp/PKIca.eml
```

- d. PKI 조직에서 해지된 인증서 목록을 보낸 경우 `certtrldb` 데이터베이스에 CRL을 추가합니다.

```
# ikecert certtrldb -a
    Press the Return key
    Paste the CRL:
-----BEGIN CRL-----
...
-----END CRL-----
    Press the Return key
<Control>-D
```

- 5 `cert_root` 키워드를 사용하여 `/etc/inet/ike/config` 파일에서 PKI 조직을 식별합니다. PKI 조직에서 제공한 이름을 사용합니다.

- a. 예를 들어, `partym` 시스템의 `ike/config` 파일은 다음과 유사하게 표시될 수 있습니다.

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform
{ auth_method rsa_sig oakley_group 1 auth_alg sha384 encr_alg aes}
p2_pfs 2

{
    label "US-partym to JA-enigmax - Example PKI"
    local_id_type dn
    local_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
    remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

    local_addr 192.168.13.213
    remote_addr 192.168.116.16

    p1_xform
    {auth_method rsa_sig oakley_group 2 auth_alg sha256 encr_alg aes}
}
```

---

주 - `auth_method` 매개변수에 대한 모든 인수는 동일한 행에 있어야 합니다.

---

- b. `enigma` 시스템에서 유사한 파일을 만듭니다.

특히 `enigma ike/config` 파일은 다음을 따라야 합니다.

- 동일한 `cert_root` 값을 포함합니다.
- 로컬 매개변수에 `enigma` 값을 사용합니다.
- 원격 매개변수에 `partym` 값을 사용합니다.



- label 키워드에 고유한 값을 만듭니다. 이 값은 원격 시스템의 label 값과 달라야 합니다.

```
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"
...
{
  label "JA-enigmax to US-party - Example PKI"
  local_id_type dn
  local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
  remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

  local_addr 192.168.116.16
  remote_addr 192.168.13.213
...

```

## 6 IKE에 CRL 처리 방법을 알립니다.

적합한 옵션을 선택합니다.

### ■ 사용 가능한 CRL 없음

PKI 조직에서 CRL을 제공하지 않을 경우 `ignore_crls` 키워드를 `ike/config` 파일에 추가합니다.

```
# Trusted root cert
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example,..."
ignore_crls
...
```

`ignore_crls` 키워드는 IKE에 CRL을 검색하지 않도록 알립니다.

### ■ 사용 가능한 CRL 있음

PKI 조직에서 CRL에 대한 중앙 배포 지점을 제공할 경우 해당 위치를 가리키도록 `ike/config` 파일을 수정할 수 있습니다.

예는 [270 페이지](#) “인증서 해지 목록 처리 방법”을 참조하십시오.

## 예 18-2 IKE 구성 시 `rsa_encrypt` 사용

`ike/config` 파일의 `auth_method rsa_encrypt`를 사용할 경우 `publickeys` 데이터베이스에 피어의 인증서를 추가해야 합니다.

### 1. 원격 시스템의 관리자에게 인증서를 보냅니다.

이 인증서는 전자 메일에 첨부할 수 있습니다.

예를 들어, `partym` 관리자가 다음 전자 메일을 보냅니다.

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
```

```
MII...
-----END X509 CERTIFICATE-----
```

enigma 관리자가 다음 전자 메일을 보냅니다.

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MI
...
-----END X509 CERTIFICATE-----
```

2. 각 시스템에서 로컬 `publickeys` 데이터베이스에 전자 메일을 통해 전송된 인증서를 추가합니다.

```
# ikecert certdb -a < /tmp/saved.cert.eml
```

RSA 암호화에 대한 인증 방법은 IKE에서 도청자에게 ID를 숨깁니다. `rsa_encrypt` 메소드는 피어의 ID를 숨기므로 IKE는 피어의 인증서를 검색할 수 없습니다. 즉, `rsa_encrypt` 메소드를 사용하려면 IKE 피어가 상대의 공개 키를 알고 있어야 합니다.

따라서 `/etc/inet/ike/config` 파일에 있는 `rsa_encrypt`의 `auth_method`를 사용할 경우 `publickeys` 데이터베이스에 피어의 인증서를 추가해야 합니다. 그러면 `publickeys` 데이터베이스가 통신하는 시스템 쌍의 각각에 대해 다음 세 개의 인증서를 보유합니다.

- 공개 키 인증서
- CA 인증서
- 피어의 공개 키 인증서

**문제 해결** - 세 개의 인증서를 포함하는 IKE 페이로드는 너무 커서 `rsa_encrypt`를 통해 암호화하지 못할 수 있습니다. “authorization failed”, “malformed payload” 등의 오류는 `rsa_encrypt` 메소드가 전체 페이로드를 암호화할 수 없음을 나타내는 것일 수 있습니다. 두 개의 인증서만 필요로 하는 `rsa_sig` 등의 메소드를 사용하여 페이로드 크기를 줄이십시오.

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## ▼ 공개 키 인증서를 생성하여 하드웨어에 저장하는 방법

공개 키 인증서를 생성하여 하드웨어에 저장하는 작업은 시스템에서 공개 키 인증서를 생성하여 저장하는 작업과 유사합니다. 하드웨어에서 `ikecert certlocal` 및 `ikecert certdb` 명령이 하드웨어를 식별해야 합니다. 토큰 ID를 사용하는 `-T` 옵션은 명령에 대한 하드웨어를 식별합니다.

**시작하기 전에** ■ 하드웨어가 구성되어 있어야 합니다.

- /etc/inet/ike/config 파일의 pkcs11\_path 키워드가 다른 라이브러리를 가리키지 않을 경우 하드웨어는 /usr/lib/libpkcs11.so 라이브러리를 사용합니다. RSA Security Inc. PKCS #11 암호화 토큰 인터페이스(Cryptoki), 즉 PKCS #11 라이브러리 표준에 따라 라이브러리가 구성되어 있어야 합니다.

설정 지침은 279 페이지 “Sun Crypto Accelerator 6000 보드를 찾도록 IKE를 구성하는 방법”을 참조하십시오.

### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스](#)의 “관리 권한을 얻는 방법”을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

### 2 자체 서명된 인증서 또는 인증서 요청을 생성하고 토큰 ID를 지정합니다.

다음 옵션 중 하나를 선택합니다.

---

주 - Sun Crypto Accelerator 6000 보드는 RSA에 대해 최대 2048비트의 키를 지원합니다. DSA의 경우 이 보드는 최대 1024비트의 키를 지원합니다.

---

- 자체 서명된 인증서의 경우 다음 구문을 사용합니다.

```
# ikcert certlocal -ks -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

-T 옵션에 대한 인수는 연결된 Sun Crypto Accelerator 6000 보드의 토큰 ID입니다.

- 인증서 요청의 경우 다음 구문을 사용합니다.

```
# ikcert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

ikcert 명령 인수에 대한 설명은 [ikcert\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### 3 PIN에 대한 프롬프트에서 Sun Crypto Accelerator 6000 사용자, 콜론 및 사용자 암호를 입력합니다.

Sun Crypto Accelerator 6000 보드에 암호가 rgm4tigt인 사용자 ikemgr이 있을 경우 다음을 입력합니다.

Enter PIN for PKCS#11 token: **ikemgr:rgm4tigt**

---

주 - PIN 응답은 디스크에 일반 텍스트로 저장됩니다.

---

암호를 입력하면 인증서가 다음과 같이 출력됩니다.

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
-----BEGIN X509 CERTIFICATE-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
oKUDBbZ90/pLWYGr
-----END X509 CERTIFICATE-----
```

#### 4 상대방이 사용할 인증서를 보냅니다.

다음 옵션 중 하나를 선택합니다.

- 원격 시스템에 자체 서명된 인증서를 보냅니다.

이 인증서는 전자 메일에 첨부할 수 있습니다.

- PKI를 처리하는 조직에 인증서 요청을 보냅니다.

PKI 조직의 지침에 따라 인증서 요청을 제출합니다. 자세한 설명은 [단계 3 of 262 페이지](#) “CA가 서명한 인증서로 IKE를 구성하는 방법”을 참조하십시오.

#### 5 시스템에서 인증서가 인식되도록 /etc/inet/ike/config 파일을 편집합니다.

다음 옵션 중 하나를 선택합니다.

- 자체 서명된 인증서

원격 시스템의 관리자가 cert\_trust, remote\_id 및 remote\_addr 매개변수에 대해 제공하는 값을 사용합니다. 예를 들어, enigma 시스템에서 ike/config 파일은 다음과 유사하게 표시됩니다.

```
# Explicitly trust the following self-signed certs
# Use the Subject Alternate Name to identify the cert

cert_trust "192.168.116.16"      Local system's certificate Subject Alt Name
cert_trust "192.168.13.213"     Remote system's certificate Subject Alt name

...
{
    label "JA-enigmax to US-party"
    local_id_type dn
    local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
    remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

    local_addr 192.168.116.16
    remote_addr 192.168.13.213

    pl_xform
    {auth_method rsa_sig oakley_group 2 auth_alg sha256 encr_alg aes}
}
```

## ■ 인증서 요청

PKI 조직에서 `cert_root` 키워드에 대한 값으로 제공하는 이름을 입력합니다. 예를 들어, `enigma` 시스템의 `ike/config` 파일은 다음과 유사하게 표시될 수 있습니다.

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

...
{
  label "JA-enigmax to US-party - Example PKI"
  local_id_type dn
  local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
  remote_id "C=US, O=PartyCompany, OU=US-Party, CN=Party"

  local_addr 192.168.116.16
  remote_addr 192.168.13.213

  pl_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha256 encr_alg aes}
}
```

## 6 하드웨어에서 상대방의 인증서를 배치합니다.

단계 3에서 응답한 대로 PIN 요청에 응답합니다.

---

주 - 반드시 개인 키를 생성한 것과 동일한 연결된 하드웨어에 공개 키 인증서를 추가해야 합니다.

---

## ■ 자체 서명된 인증서

원격 시스템의 자체 서명된 인증서를 추가합니다. 이 예에서는 인증서가 `DCA.ACCEL.STOR.CERT` 파일에 저장됩니다.

```
# ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
Enter PIN for PKCS#11 token:      Type user:password
```

자체 서명된 인증서가 `rsa_encrypt`를 `auth_method` 매개변수에 대한 값으로 사용한 경우 하드웨어 저장소에 피어의 인증서를 추가합니다.

## ■ PKI 조직의 인증서

인증서 요청에 따라 조직에서 생성한 인증서를 추가하고 인증 기관(CA)을 추가합니다.

```
# ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
Enter PIN for PKCS#11 token:      Type user:password
```

```
# ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CA.CERT
Enter PIN for PKCS#11 token:      Type user:password
```

PKI 조직의 인증서 해지 목록(CRL)을 추가하려면 270 페이지 “인증서 해지 목록 처리 방법”을 참조하십시오.

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## ▼ 인증서 해지 목록 처리 방법

인증서 해지 목록(CRL)에는 인증 기관의 오래되거나 손상된 인증서가 포함됩니다. 네 가지 방법으로 CRL을 처리할 수 있습니다.

- CA 조직에서 CRL을 발급하지 않은 경우 CRL을 무시하도록 IKE에 알려야 합니다. 이 옵션은 **단계 6 in 262 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”**에서 설명됩니다.
- CA의 공개 키 인증서에 주소가 포함된 URI(Uniform Resource Indicator)의 CRL에 액세스하도록 IKE에 알릴 수 있습니다.
- CA의 공개 키 인증서에 디렉토리 이름(DN) 항목이 포함된 LDAP 서버의 CRL에 액세스하도록 IKE에 알릴 수 있습니다.
- `ikecert certrl db` 명령에 대한 인수로 CRL을 제공할 수 있습니다. 예는 **예 18-3**을 참조하십시오.

다음 절차에서는 중앙 배포 지점의 CRL을 사용하도록 IKE에 알리는 방법에 대해 설명합니다.

### 1 CA에서 수신한 인증서를 표시합니다.

```
# ikecert certdb -lv certspec
```

-l IKE 인증서 데이터베이스의 인증서를 나열합니다.

-v 상세 정보 표시 모드로 인증서를 나열합니다. 이 옵션은 주의해서 사용하십시오.

*certspec* IKE 인증서 데이터베이스의 인증서와 일치하는 패턴입니다.

예를 들어, Oracle에서 발급한 인증서는 다음과 같습니다. 세부 정보는 변경되었습니다.

```
# ikecert certdb -lv example-protect.oracle.com
Certificate Slot Name: 0   Type: dsa-sha1
  (Private key in certlocal slot 0)
Subject Name: <O=Oracle, CN=example-protect.oracle.com>
Issuer Name: <CN=Oracle CA (Cl B), O=Oracle>
SerialNumber: 14000D93
Validity:
  Not Valid Before: 2011 Sep 19th, 21:11:11 GMT
  Not Valid After:  2015 Sep 18th, 21:11:11 GMT
Public Key Info:
```

```

Public Modulus (n) (2048 bits): C575A...A5
Public Exponent (e) ( 24 bits): 010001
Extensions:
  Subject Alternative Names:
    DNS = example-protect.oracle.com
  Key Usage: DigitalSignature KeyEncipherment
  [CRITICAL]
CRL Distribution Points:
  Full Name:
    URI = #Ihttp://www.oracle.com/pki/pkismica.crl#i
    DN = <CN=Oracle CA (Cl B), O=Oracle>
  CRL Issuer:
  Authority Key ID:
  Key ID:          4F ... 6B
  SubjectKeyID:    A5 ... FD
  Certificate Policies
  Authority Information Access

```

CRL Distribution Points 항목을 확인합니다. URI 항목은 이 조직의 CRL을 웹에서 사용할 수 있음을 나타냅니다. DN 항목은 CRL을 LDAP 서버에서 사용할 수 있음을 나타냅니다. IKE가 액세스한 CRL은 나중에 사용할 수 있도록 캐시됩니다.

CRL에 액세스하려면 배포 지점에 연결해야 합니다.

## 2 중앙 배포 지점에서 CRL에 액세스하는 데 사용할 다음 방법 중 하나를 선택합니다.

### ■ URI 사용

use\_http 키워드를 호스트의 /etc/inet/ike/config 파일에 추가합니다. 예를 들어, ike/config 파일은 다음과 유사하게 표시됩니다.

```

# Use CRL from organization's URI
use_http
...

```

### ■ 웹 프록시 사용

proxy 키워드를 ike/config 파일에 추가합니다. proxy 키워드는 다음에서와 같이 URL을 인수로 사용합니다.

```

# Use own web proxy
proxy "http://proxy1:8080"

```

### ■ LDAP 서버 사용

호스트의 /etc/inet/ike/config 파일에서 LDAP 서버를 ldap-list 키워드에 대한 인수로 지정합니다. 조직에서 LDAP 서버의 이름을 제공합니다. ike/config 파일의 항목은 다음과 유사하게 표시됩니다.

```

# Use CRL from organization's LDAP
ldap-list "ldap1.oracle.com:389,ldap2.oracle.com"
...

```

IKE가 CRL을 검색하고 인증서가 만료될 때까지 CRL을 캐시합니다.

### 예 18-3 로컬 certltdb 데이터베이스에 CRL 붙여넣기

중앙 배포 지점에서 PKI 조직의 CRL을 사용할 수 없을 경우 수동으로 로컬 certltdb 데이터베이스에 CRL을 추가할 수 있습니다. PKI 조직의 지침에 따라 CRL을 파일에 추출한 다음 `ikecert certltdb -a` 명령을 사용하여 데이터베이스에 CRL을 추가합니다.

```
# ikcert certltdb -a < Oracle.Cert.CRL
```

## 모바일 시스템에 대한 IKE 구성(작업 맵)

다음 표에서는 원격으로 중앙 사이트에 로그인한 시스템을 처리하도록 IKE를 구성하는 절차에 대해 설명합니다.

작업	설명	수행 방법
오프사이트의 중앙 사이트와 통신합니다.	오프사이트 시스템이 중앙 사이트와 통신할 수 있도록 합니다. 오프사이트 시스템은 모바일일 수 있습니다.	272 페이지 “오프사이트 시스템에 대한 IKE 구성 방법”
모바일 시스템의 트래픽을 승인하는 중앙 시스템에서 CA의 공개 인증서 및 IKE를 사용합니다.	고정 IP 주소가 없는 시스템의 IPsec 트래픽을 승인하도록 게이트웨이 시스템을 구성합니다.	예 18-4
고정 IP 주소가 없는 시스템에서 CA의 공개 인증서 및 IKE를 사용합니다.	회사 본사 등의 중앙 사이트에 대한 트래픽을 보호하도록 모바일 시스템을 구성합니다.	예 18-5
모바일 시스템의 트래픽을 승인하는 중앙 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	모바일 시스템의 IPsec 트래픽을 승인하도록 자체 서명된 인증서로 게이트웨이 시스템을 구성합니다.	예 18-6
고정 IP 주소가 없는 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	중앙 사이트에 대한 트래픽을 보호하도록 자체 서명된 인증서로 모바일 시스템을 구성합니다.	예 18-7

## 모바일 시스템에 대한 IKE 구성

제대로 구성된 경우 자택 근무 시, 그리고 모바일 랩탑에서 IPsec 및 IKE를 사용하여 회사의 중앙 컴퓨터와 통신할 수 있습니다. 공개 키 인증 방법과 결합된 총괄 IPsec 정책을 통해 오프사이트 시스템은 중앙 시스템에 대한 트래픽을 보호할 수 있습니다.

### ▼ 오프사이트 시스템에 대한 IKE 구성 방법

IPsec 및 IKE에는 소스 및 대상을 식별할 고유한 ID가 필요합니다. 고유한 IP 주소가 없는 오프사이트 또는 모바일 시스템의 경우 다른 ID 유형을 사용해야 합니다. DNS, DN, email 등의 ID 유형을 사용하여 시스템을 고유하게 식별할 수 있습니다.



고유한 IP 주소가 있는 오프사이트 또는 모바일 시스템은 다른 ID 유형으로 구성하는 것이 좋습니다. 예를 들어, 시스템이 NAT 박스 뒤에 있는 중앙 사이트에 연결하려고 시도할 경우 고유한 주소가 사용되지 않습니다. NAT 박스는 중앙 시스템에서 인식할 수 없는 임의적인 IP 주소를 지정합니다.

미리 공유한 키도 모바일 시스템에 대한 인증 방식으로 작동하지 않습니다. 미리 공유한 키에는 고정 IP 주소가 필요하기 때문입니다. 모바일 시스템은 자체 서명된 인증서 또는 PKI의 인증서를 통해 중앙 사이트와 통신할 수 있습니다.

## 1 관리자 로깅인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”**을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

## 2 모바일 시스템을 인식하도록 중앙 시스템을 구성합니다.

### a. `ipsecinit.conf` 파일을 구성합니다.

중앙 시스템에는 광범위한 IP 주소를 허용하는 정책이 필요합니다. 나중에 IKE 정책의 인증서를 사용하면 연결하는 시스템이 적합한 것으로 보장됩니다.

```
# /etc/inet/ipsecinit.conf on central
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}
```

### b. IKE 구성 파일을 구성합니다.

DNS가 중앙 시스템을 식별합니다. 인증서는 시스템을 인증하는 데 사용됩니다.

```
## /etc/inet/ike/ike.config on central
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates
cert_root "C=US, O=Domain Org, CN=Domain STATE"
#
# List self-signed certificates - trust server and enumerated others
#cert_trust "DNS=central.domain.org"
#cert_trust "DNS=mobile.domain.org"
#cert_trust "DN=CN=Domain Org STATE (CLASS), O=Domain Org"
#cert_trust "email=root@central.domain.org"
#cert_trust "email=user1@mobile.domain.org"
#

# Rule for mobile systems with certificate
```

```

{
    label "Mobile systems with certificate"
    local_id type DNS
    # CA's public certificate ensures trust,
    # so allow any remote_id and any remote IP address.
    remote_id ""
    remote_addr 0.0.0.0/0

    p2_pfs 5

    p1_xform
    {auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}

```

### 3 각 모바일 시스템에 로그인하고 중앙 시스템을 찾으러 시스템을 구성합니다.

#### a. /etc/hosts 파일을 구성합니다.

/etc/hosts 파일은 모바일 시스템의 주소를 필요로 하지 않지만 제공할 수 있습니다. 파일에는 중앙 시스템에 대한 공용 IP 주소가 포함되어야 합니다.

```

# /etc/hosts on mobile
central 192.xxx.xxx.x

```

#### b. ipsecinit.conf 파일을 구성합니다.

모바일 시스템이 공용 IP 주소로 중앙 시스템을 찾아야 합니다. 시스템은 동일한 IPsec 정책을 구성해야 합니다.

```

# /etc/inet/ipsecinit.conf on mobile
# Find central
{raddr 192.xxx.xxx.x} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

```

#### c. IKE 구성 파일을 구성합니다.

IP 주소는 식별자일 수 없습니다. 모바일 시스템에 유효한 식별자는 다음과 같습니다.

- DN=ldap-directory-name
- DNS=domain-name-server-address
- email=email-address

인증서는 모바일 시스템을 인증하는 데 사용됩니다.

```

## /etc/inet/ike/ike.config on mobile
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates

```

```

cert_root "C=US, O=Domain Org, CN=Domain STATE"
#
# Self-signed certificates - trust me and enumerated others
#cert_trust "DNS=mobile.domain.org"
#cert_trust "DNS=central.domain.org"
#cert_trust "DN=CN=Domain Org STATE (CLASS), O=Domain Org"
#cert_trust "email=user1@domain.org"
#cert_trust "email=root@central.domain.org"
#
# Rule for off-site systems with root certificate
{
    label "Off-site mobile with certificate"
    local_id_type DNS

    # NAT-T can translate local_addr into any public IP address
    # central knows me by my DNS

    local_id "mobile.domain.org"
    local_addr 0.0.0.0/0

    # Find central and trust the root certificate
    remote_id "central.domain.org"
    remote_addr 192.xxx.xxx.x

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}

```

#### 4. ike 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/ike
```

### 예 18-4 모바일 시스템의 IPsec 트래픽을 승인하도록 중앙 컴퓨터 구성

IKE는 NAT 박스 뒤에서 협상을 시작할 수 있습니다. 하지만 적합한 IKE 설정은 개입하는 NAT 박스가 없는 것입니다. 다음 예에서는 CA의 공개 인증서가 모바일 시스템 및 중앙 시스템에 배치되었습니다. 중앙 시스템이 NAT 박스 뒤에 있는 시스템의 IPsec 협상을 승인합니다. main1은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 예 18-5를 참조하십시오.

```

## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http

```

```
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
    label "Off-site system with root certificate"
    local_id_type DNS
    local_id "main1.domain.org"
    local_addr 192.168.0.100

    # CA's public certificate ensures trust,
    # so allow any remote_id and any remote IP address.
    remote_id ""
    remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
}
```

### 예 18-5 IPsec로 NAT 뒤에 있는 시스템 구성

다음 예에서는 CA의 공개 인증서가 모바일 시스템 및 중앙 시스템에 배치됩니다. **mobile1**은 자택에서 회사 본사에 연결하고 있습니다. 인터넷 서비스 제공업체(ISP) 네트워크는 NAT 박스를 사용하여 ISP가 **mobile1**에 개인 주소를 지정할 수 있도록 합니다. 그러면 NAT 박스는 다른 ISP 네트워크 노드와 공유되는 공용 IP 주소로 개인 주소를 변환합니다. 회사 본사는 NAT 뒤에 없습니다. 회사 본사에서 컴퓨터를 설정하려면 [예 18-4](#)를 참조하십시오.

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
```

```
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
    label "Off-site mobile1 with root certificate"
    local_id_type DNS
    local_id "mobile1.domain.org"
    local_addr 0.0.0.0/0

# Find main1 and trust the root certificate
remote_id "main1.domain.org"
remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}
```

## 예 18-6 모바일 시스템의 자체 서명된 인증서 승인

다음 예에서는 자체 서명된 인증서가 발급되었으며 모바일 및 중앙 시스템에 배치됩니다. main1은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 [예 18-7](#)을 참조하십시오.

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Self-signed certificates - trust me and enumerated others
cert_trust "DNS=main1.domain.org"
cert_trust "jdoe@domain.org"
cert_trust "user2@domain.org"
cert_trust "user3@domain.org"
#
# Rule for off-site systems with trusted certificate
{
    label "Off-site systems with trusted certificates"
    local_id_type DNS
    local_id "main1.domain.org"
```

```

local_addr 192.168.0.100

# Trust the self-signed certificates
# so allow any remote_id and any remote IP address.
remote_id ""
remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}

```

### 예 18-7 자체 서명된 인증서를 사용하여 중앙 시스템에 연결

다음 예에서는 mobile1이 자택에서 회사 본사에 연결하고 있습니다. 인증서가 발급되었으며 모바일 및 중앙 시스템에 배치됩니다. ISP 네트워크는 NAT 박스를 사용하여 ISP가 mobile1에 개인 주소를 지정할 수 있도록 합니다. 그러면 NAT 박스는 다른 ISP 네트워크 노드와 공유되는 공용 IP 주소로 개인 주소를 변환합니다. 회사 본사는 NAT 뒤에 없습니다. 회사 본사에서 컴퓨터를 설정하려면 예 18-6을 참조하십시오.

```

## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters

# Self-signed certificates - trust me and the central system
cert_trust "jdoe@domain.org"
cert_trust "DNS=main1.domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site mobile1 with trusted certificate"
  local_id_type email
  local_id "jdoe@domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}

```

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

## 연결된 하드웨어를 찾도록 IKE 구성

연결된 하드웨어에도 공개 키 인증서를 저장할 수 있습니다. Sun Crypto Accelerator 6000 보드는 저장소를 제공하고 공개 키 작업이 시스템에서 보드로 오프로드될 수 있도록 합니다.

### ▼ Sun Crypto Accelerator 6000 보드를 찾도록 IKE를 구성하는 방법

**시작하기 전에** 다음 절차에서는 Sun Crypto Accelerator 6000 보드가 시스템에 연결된 것으로 간주합니다. 또한 절차에서는 보드용 소프트웨어가 설치되었으며 소프트웨어가 구성된 것으로 간주합니다. 지침은 **Sun Crypto Accelerator 6000 Board Version 1.1 User's Guide**를 참조하십시오.

#### 1 관리자로 로그인합니다.

자세한 내용은 [Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”](#)을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 [예 15-1](#)을 참조하십시오.

#### 2 PKCS#11 라이브러리가 연결되어 있는지 확인합니다.

IKE는 라이브러리의 루틴을 사용하여 Sun Crypto Accelerator 6000 보드에서의 키 생성 및 키 저장을 처리합니다. 다음 명령을 입력하여 PKCS#11 라이브러리가 연결되었는지 여부를 확인합니다.

```
$ ikeadm get stats
...
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

#### 3 연결된 Sun Crypto Accelerator 6000 보드에 대한 토큰 ID를 찾습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":
```

```
"Sun Metaslot"
```

라이브러리가 32자의 토큰 ID([키 저장소 이름](#)이라고도 함)을 반환합니다. 이 예에서는 `ikecert` 명령에 Sun Metaslot 토큰을 사용하여 IKE 키를 저장하고 속도를 향상시킬 수 있습니다.

토큰 사용 방법에 대한 지침은 [266 페이지 “공개 키 인증서를 생성하여 하드웨어에 저장하는 방법”](#)을 참조하십시오.

`ikecert` 명령을 통해 자동으로 후행 공백이 채워집니다.

**예 18-8 Metaslot 토큰 찾기 및 사용**

토큰은 디스크, 연결된 보드 또는 암호화 프레임워크가 제공하는 소프트웨어 토큰 키 저장소에 저장할 수 있습니다. 소프트웨어 토큰 키 저장소 토큰 ID는 다음과 유사할 수 있습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":
```

```
"Sun Metaslot"
```

소프트 토큰 키 저장소에 대한 암호문을 만들려면 [pktool\(1\)](#) 매뉴얼 페이지를 참조하십시오.

다음과 유사한 명령이 소프트웨어 토큰 키 저장소에 인증서를 추가합니다. Sun.Metaslot.cert는 CA 인증서가 포함된 파일입니다.

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:      Type user:passphrase
```

**다음 순서** IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.



## Internet Key Exchange(참조)

---

이 장은 IKE에 대한 다음 참조 정보를 포함합니다.

- 281 페이지 “IKE 서비스”
- 282 페이지 “IKE 데몬”
- 282 페이지 “IKE 구성 파일”
- 283 페이지 “ikeadm 명령”
- 284 페이지 “IKE 미리 공유한 키 파일”
- 284 페이지 “IKE 공개 키 데이터베이스 및 명령”

IKE 구현 지침은 18 장, “IKE 구성(작업)”을 참조하십시오. 개요 정보는 17 장, “Internet Key Exchange(개요)”를 참조하십시오.

## IKE 서비스

`svc:/network/ipsec/ike:default` 서비스 - SMF(서비스 관리 기능)는 IKE를 관리하기 위해 `ike` 서비스를 제공합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 이 서비스를 사용으로 설정하기 전에 IKE 구성 파일 `/etc/inet/ike/config`를 만들어야 합니다.

다음 `ike` 서비스 등록 정보를 구성할 수 있습니다.

- `config_file` 등록 정보 - IKE 구성 파일의 위치입니다. 초기 값은 `/etc/inet/ike/config`입니다.
- `debug level` 등록 정보 - `in.iked` 데몬의 디버깅 레벨입니다. 초기 값은 `op` 또는 `operational`입니다. 가능한 값은 `ikeadm(1M)` 매뉴얼 페이지에서 **객체 유형** 아래의 디버그 레벨 테이블을 참조하십시오.
- `admin_privilege` 등록 정보 - `in.iked` 데몬의 권한 레벨입니다. 초기 값은 `base`입니다. 다른 값으로 `modkeys` 및 `keymat`가 있습니다. 세부 정보는 283 페이지 “ikeadm 명령”을 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리(개요)”**를 참조하십시오. 또한 [smf\(5\)](#), [svcadm\(1M\)](#), [svccfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## IKE 데몬

`in.iked` 데몬은 Oracle Solaris 시스템에서 IPsec에 대한 암호화 키 관리를 자동화합니다. 데몬은 동일한 프로토콜을 실행 중인 원격 시스템과 협상하여 보안 연관(SA)에 대한 인증된 키 관련 자료를 안전한 방식으로 제공합니다. 안전하게 통신하려는 모든 시스템에서 데몬을 실행 중이어야 합니다.

기본적으로 `svc:/network/ipsec/ike:default` 서비스는 사용으로 설정되지 않습니다. `/etc/inet/ike/config` 파일을 구성하고 `ike` 서비스를 사용으로 설정한 후에 시스템 부트 시 `in.iked` 데몬이 실행됩니다.

IKE 데몬을 실행할 때 시스템이 Phase 1 교환에서 피어 IKE 엔티티로 자체 인증합니다. 피어는 인증 방법과 마찬가지로 IKE 정책 파일에 정의됩니다. 그런 다음 데몬이 Phase 2 교환에 대한 키를 설정합니다. 정책 파일에 지정된 간격으로 IKE 키를 자동으로 새로 고칩니다. `in.iked` 데몬이 네트워크에서 들어오는 IKE 요청과 `PF_KEY` 소켓을 통과하는 아웃바운드 트래픽 요청을 수신합니다. 자세한 내용은 [pf\\_key\(7P\)](#) 매뉴얼 페이지를 참조하십시오.

두 가지 명령이 IKE 데몬을 지원합니다. `ikeadm` 명령을 사용하여 IKE 정책을 확인하고 일시적으로 수정할 수 있습니다. IKE 정책을 영구적으로 수정하려면 `ike` 서비스의 등록 정보를 수정합니다. IKE 서비스의 등록 정보를 수정하려면 [234 페이지 “IPsec 및 IKE 서비스를 관리하는 방법”](#)을 참조하십시오. 또한 `ikeadm` 명령을 사용하여 Phase 1 SA, 정책 규칙, 미리 공유한 키, 사용 가능한 Diffie-Hellman 그룹, Phase 1 암호화 및 인증 알고리즘, 인증서 캐시 등을 볼 수 있습니다.

`ikecert` 명령을 사용하여 공개 키 데이터베이스를 보고 관리할 수 있습니다. 이 명령은 로컬 데이터베이스인 `ike.privatekeys` 및 `publickeys`를 관리합니다. 또한 이 명령은 공개 키 작업 및 하드웨어의 공개 키 저장소를 관리합니다.

## IKE 구성 파일

IKE 구성 파일 `/etc/inet/ike/config`는 IPsec 정책 파일 `/etc/inet/ipsecinit.conf`에서 보호되는 인터페이스의 키를 관리합니다.

IKE의 키 관리에는 규칙 및 전역 매개변수가 관여합니다. IKE 규칙은 키 관련 자료를 보안하는 시스템 또는 네트워크를 식별합니다. 또한 규칙은 인증 방법을 지정합니다. 전역 매개변수에는 연결된 하드웨어 가속기의 경로와 같은 항목이 포함됩니다. IKE 정책 파일의 예는 [251 페이지 “미리 공유한 키로 IKE 구성\(작업 맵\)”](#)을 참조하십시오. IKE 정책 항목의 예제 및 설명은 [ike.config\(4\)](#) 매뉴얼 페이지를 참조하십시오.

IKE가 지원하는 IPsec SA는 IPsec 구성 파일 `/etc/inet/ipsecinit.conf`의 정책에 따라 IP 데이터그램을 보호합니다. IKE 정책 파일은 IPsec SA를 만들 때 PFS(완전 순방향 비밀성)의 사용 여부를 결정합니다.

`/etc/inet/ike/config` 파일은 RSA Security Inc.의 PKCS #11 암호화 토큰 인터페이스(Cryptoki) 표준에 따라 구현되는 라이브러리의 경로를 포함할 수 있습니다. IKE는 이 PKCS #11 라이브러리를 사용하여 키 가속 및 키 저장을 위한 하드웨어에 액세스합니다.

`ike/config` 파일에 대한 보안 고려 사항은 `ipsecinit.conf` 파일의 고려 사항과 비슷합니다. 세부 정보는 [239 페이지 “ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항”](#)을 참조하십시오.

## ikeadm 명령

ikeadm 명령을 사용하여 다음을 수행할 수 있습니다.

- IKE 상태의 여러 측면을 봅니다.
- IKE 데몬의 등록 정보를 변경합니다.
- Phase 1 교환 중 SA 생성에 대한 통계를 표시합니다.
- IKE 프로토콜 교환을 디버그합니다.
- 모든 Phase 1 SA, 정책 규칙, 미리 공유한 키, 사용 가능한 Diffie-Hellman 그룹, Phase 1 암호화 및 인증 알고리즘, 인증서 캐시 등의 IKE 데몬 객체를 표시합니다.

이 명령의 옵션에 대한 예제 및 전체 설명은 [ikeadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

실행 중인 IKE 데몬의 권한 레벨에 따라 IKE 데몬의 어떤 측면을 보고 수정할 수 있는지 결정됩니다. 3단계 권한 레벨이 가능합니다.

**base 레벨**            키 관련 자료를 보거나 수정할 수 없습니다. base 레벨이 기본 권한 레벨입니다.

**modkeys 레벨**        미리 공유한 키를 제거, 변경, 추가할 수 있습니다.

**keymat 레벨**        ikeadm 명령을 사용하여 실제 키 관련 자료를 볼 수 있습니다.

일시적 권한 변경은 ikeadm 명령을 사용할 수 있습니다. 영구적 변경은 `ike` 서비스의 `admin_privilege` 등록 정보를 변경합니다. 절차는 [234 페이지 “IPsec 및 IKE 서비스를 관리하는 방법”](#)을 참조하십시오.

ikeadm 명령에 대한 보안 고려 사항은 `ipseckey` 명령의 고려 사항과 비슷합니다. 세부 정보는 [241 페이지 “ipseckey에 대한 보안 고려 사항”](#)을 참조하십시오.

## IKE 미리 공유한 키 파일

미리 공유한 키를 수동으로 만들 때 `/etc/inet/secret` 디렉토리의 파일에 키가 저장됩니다. `ike.preshared` 파일은 ISAKMP(Internet Security Association and Key Management Protocol) SA에 대한 미리 공유한 키를 포함합니다. `ipseckey` 파일은 IPsec SA에 대한 미리 공유한 키를 포함합니다. 파일은 `0600`에서 보호됩니다. `secret` 디렉토리는 `0700`에서 보호됩니다.

- `ike/config` 파일에서 미리 공유한 키를 요구하도록 구성할 때 `ike.preshared` 파일을 만듭니다. `ike.preshared` 파일에 IKE 인증인 ISAKMP SA에 대한 키 관련 자료를 입력합니다. 미리 공유한 키를 사용하여 Phase 1 교환을 인증하므로 `in.iked` 데몬을 시작하기 전에 파일이 유효해야 합니다.
- `ipseckey` 파일은 IPsec SA에 대한 키 관련 자료를 포함합니다. 파일 수동 관리의 예는 [231 페이지 “IPsec 키를 수동으로 만드는 방법”](#)을 참조하십시오. IKE 데몬은 이 파일을 사용하지 않습니다. IPsec SA에 대해 IKE가 생성하는 키 관련 자료는 커널에 저장됩니다.

## IKE 공개 키 데이터베이스 및 명령

`ikecert` 명령은 로컬 시스템의 공개 키 데이터베이스를 조작합니다. `ike/config` 파일에 공개 키 인증서가 필요할 때 이 명령을 사용합니다. IKE는 이러한 데이터베이스를 사용하여 Phase 1 교환을 인증하므로 `in.iked` 데몬을 활성화하기 전에 데이터베이스를 채워야 합니다. 세 가지 하위 명령 `certlocal`, `certdb`, `certldb`가 각각 세 데이터베이스를 처리합니다.

`ikecert` 명령은 키 저장소도 처리합니다. 디스크, 연결된 Sun Crypto Accelerator 6000 보드 또는 `softtoken` 키 저장소에 키를 저장할 수 있습니다. 암호화 프레임워크의 `metaslot`를 사용하여 하드웨어 장치와 통신할 때 `softtoken` 키 저장소를 사용할 수 있습니다. `ikecert` 명령은 PKCS #11 라이브러리를 사용하여 키 저장소를 찾습니다.

자세한 내용은 [ikecert\(1M\)](#) 매뉴얼 페이지를 참조하십시오. `metaslot` 및 `softtoken` 키 저장소에 대한 내용은 [cryptoadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ikecert tokens 명령

`tokens` 인수는 사용 가능한 토큰 ID를 나열합니다. 토큰 ID를 통해 `ikecert certlocal` 및 `ikecert certdb` 명령에서 공개 키 인증서 및 인증서 요청을 생성할 수 있습니다. 또한 암호화 프레임워크에서 `softtoken` 키 저장소 또는 연결된 Sun Crypto Accelerator 6000 보드에 인증서 및 인증서 요청을 저장할 수 있습니다. `ikecert` 명령은 PKCS #11 라이브러리를 사용하여 인증서 저장소를 찾습니다.

## ikecert certlocal 명령

certlocal 하위 명령은 개인 키 데이터베이스를 관리합니다. 이 하위 명령의 옵션을 사용하여 개인 키를 추가, 보기, 제거할 수 있습니다. 또한 이 하위 명령은 자체 서명된 인증서 또는 인증서 요청을 만듭니다. -ks 옵션은 자체 서명된 인증서를 만듭니다. -kc 옵션은 인증서 요청을 만듭니다. 키는 /etc/inet/secret/ike.privatekeys 디렉토리에서 시스템에 저장되거나, -T 옵션을 사용하여 연결된 하드웨어에 저장됩니다.

개인 키를 만들 때 ikecert certlocal 명령의 옵션이 ike/config 파일의 항목과 관련을 맺어야 합니다. ikecert 옵션과 ike/config 항목 사이의 관련성이 다음 표에 표시됩니다.

표 19-1 ikecert 옵션과 ike/config 항목 사이의 관련성

ikecert 옵션	ike/config 항목	설명
-A subject-alternate-name	cert_trust subject-alternate-name	인증서를 고유하게 식별하는 별명입니다. 가능한 값은 IP 주소, 전자 메일 주소 또는 도메인 이름입니다.
-D X.509-distinguished-name	X.509-distinguished-name	국가(C), 조직 이름(ON), 조직 단위(OU), 공통 이름(CN)을 포함하는 인증 기관의 전체 이름입니다.
-t dsa-sha1	auth_method dsa_sig	RSA보다 약간 느린 인증 방법입니다.
-t rsa-md5 및	auth_method rsa_sig	DSA보다 약간 빠른 인증 방법입니다.
-t rsa-sha1		RSA 공개 키는 가장 큰 페이로드를 암호화할 만큼 충분히 커야 합니다. 일반적으로 X.509 식별 이름과 같은 신원 페이로드가 가장 큰 페이로드입니다.
-t rsa-md5 및	auth_method rsa_encrypt	RSA 암호화는 도청자로부터 IKE의 신원을 숨기지만 IKE 피어가 서로의 공개 키를 알아야 합니다.
-t rsa-sha1		

ikecert certlocal -kc 명령으로 인증서 요청을 발행하면 명령의 출력을 PKI 조직이나 인증 기관(CA)으로 보냅니다. 회사에서 고유의 PKI를 실행하는 경우 PKI 관리자에게 출력을 보냅니다. 그런 다음 PKI 조직, CA 또는 PKI 관리자가 인증서를 만듭니다. PKI 또는 CA가 반환하는 인증서는 certdb 하위 명령으로 입력됩니다. PKI가 반환하는 CRL(인증서 해지 목록)은 certrdb 하위 명령으로 입력됩니다.

## ikecert certdb 명령

certdb 하위 명령은 공개 키 데이터베이스를 관리합니다. 이 하위 명령의 옵션을 사용하여 인증서 및 공개 키를 추가, 보기, 제거할 수 있습니다. 이 명령은 원격 시스템에서 ikecert certlocal -ks 명령으로 생성된 인증서를 입력으로 받아들입니다. 절차는 257 페이지 “자체 서명된 공개 키 인증서로 IKE를 구성하는 방법”을

참조하십시오. 또한 이 명령은 PKI 또는 CA로부터 받은 인증서를 입력으로 받아들입니다. 절차는 [262 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”](#)을 참조하십시오.

인증서 및 공개 키는 `/etc/inet/ike/publickeys` 디렉토리에서 시스템에 저장됩니다. -T 옵션은 연결된 하드웨어에 인증서, 개인 키, 공개 키를 저장합니다.

## ikecert certrl db 명령

`certrl db` 하위 명령은 CRL(인증서 해지 목록) 데이터베이스인 `/etc/inet/ike/crls`를 관리합니다. CRL 데이터베이스는 공개 키에 대한 해지 목록을 유지 관리합니다. 이 목록에는 더 이상 유효하지 않은 인증서가 있습니다. PKI에서 CRL을 제공할 때 `ikecert certrl db` 명령을 사용하여 CRL 데이터베이스에 CRL을 설치할 수 있습니다. 절차는 [270 페이지 “인증서 해지 목록 처리 방법”](#)을 참조하십시오.

## /etc/inet/ike/publickeys 디렉토리

`/etc/inet/ike/publickeys` 디렉토리는 공개-개인 키 쌍의 공개 부분과 해당 인증서를 파일이나 슬롯에 넣습니다. 디렉토리는 0755에서 보호됩니다. `ikecert certdb` 명령은 디렉토리를 채웁니다. -T 옵션은 `publickeys` 디렉토리가 아닌 Sun Crypto Accelerator 6000 보드에 키를 저장합니다.

슬롯은 다른 시스템에서 생성된 인증서의 X.509 식별 이름을 인코딩된 형태로 포함합니다. 자체 서명한 인증서를 사용하는 경우 원격 시스템의 관리자로부터 받은 인증서를 명령의 입력으로 사용합니다. CA의 인증서를 사용하는 경우 CA에서 서명한 두 인증서를 이 데이터베이스로 설치합니다. CA로 보낸 인증서 서명 요청에 준하는 인증서를 설치합니다. 또한 CA의 인증서를 설치합니다.

## /etc/inet/secret/ike.privatekeys 디렉토리

`/etc/inet/secret/ike.privatekeys` 디렉토리는 공개-개인 키 쌍의 일부인 개인 키 파일을 보유합니다. 디렉토리는 0700에서 보호됩니다. `ikecert certlocal` 명령은 `ike.privatekeys` 디렉토리를 채웁니다. 대응하는 공개 키, 자체 서명한 인증서 또는 CA를 설치할 때까지 개인 키는 효과가 없습니다. 대응하는 공개 키는 `/etc/inet/ike/publickeys` 디렉토리 또는 지원되는 하드웨어에 저장됩니다.

## /etc/inet/ike/crls 디렉토리

`/etc/inet/ike/crls` 디렉토리는 CRL(인증서 해지 목록) 파일을 포함합니다. 각 파일은 `/etc/inet/ike/publickeys` 디렉토리의 공개 인증서 파일에 해당합니다. PKI 조직은 해당 인증서에 대한 CRL을 제공합니다. `ikecert certrl db` 명령을 사용하여 데이터베이스를 채울 수 있습니다.

## Oracle Solaris의 IP 필터(개요)

이 장에서는 Oracle Solaris 기능인 IP 필터의 개요를 제공합니다. IP 필터 작업은 21 장, “IP 필터(작업)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

- 287 페이지 “IP 필터 소개”
- 288 페이지 “IP 필터 패킷 처리”
- 290 페이지 “IP 필터 사용 지침”
- 291 페이지 “IP 필터 구성 파일 사용”
- 292 페이지 “IP 필터 규칙 세트 사용”
- 297 페이지 “패킷 필터 후크”
- 297 페이지 “IP 필터용 IPv6”
- 298 페이지 “IP 필터 매뉴얼 페이지”

## IP 필터 소개

Oracle Solaris의 IP 필터 기능은 OS의 SunScreen 방화벽을 대체합니다. SunScreen 방화벽과 마찬가지로 IP 필터는 Stateful 패킷 필터링 및 NAT(Network Address Translation)를 제공합니다. IP 필터에는 Stateless 패킷 필터링을 비롯하여 주소 풀 생성 및 관리 기능도 포함되어 있습니다.

패킷 필터링은 네트워크 기반 공격에 대비한 기본적인 보호를 제공합니다. IP 필터는 IP 주소, 포트, 프로토콜, 네트워크 인터페이스 및 트래픽 방향을 기준으로 필터링을 수행할 수 있습니다. 개별 소스 IP 주소, 대상 IP 주소, IP 주소 범위 또는 주소 풀을 기준으로도 필터링을 수행할 수 있습니다.

IP 필터는 오픈 소스 IP 필터 소프트웨어에서 파생되었습니다. 오픈 소스 IP 필터에 대한 라이선스 약관, 저작권 및 저작권 설명을 볼 수 있는 기본 경로는 `/usr/lib/ipf/IPFILTER.LICENCE`입니다. Oracle Solaris가 기본 경로 이외의 다른 경로에 설치된 경우 설치된 위치의 파일에 액세스할 수 있도록 지정된 경로를 수정하십시오.

## 오픈 소스 IP 필터에 대한 정보 소스

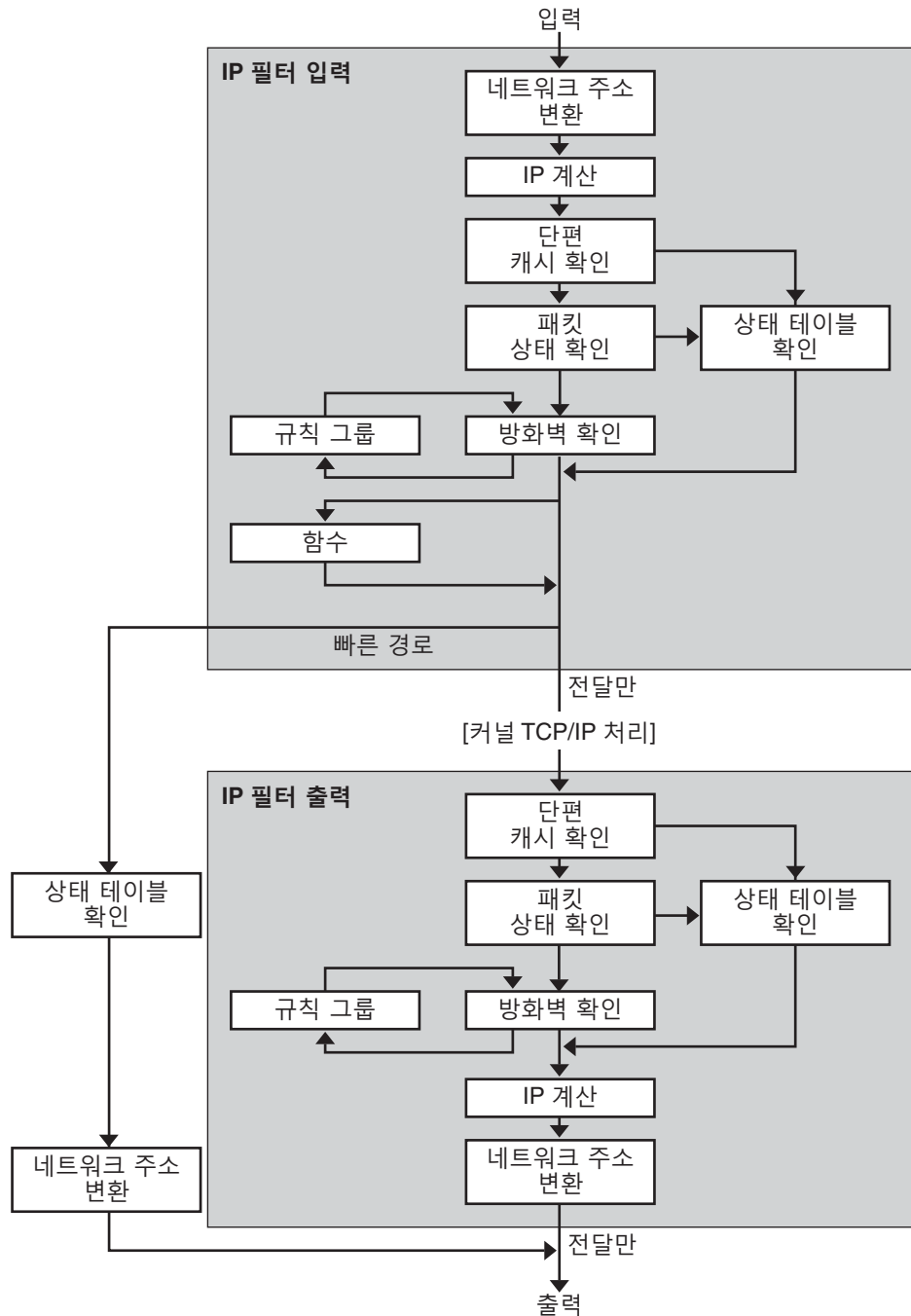
Darren Reed의 오픈 소스 IP 필터 소프트웨어 홈 페이지는 <http://coombs.anu.edu.au/~avalon/ip-filter.html>에서 확인할 수 있습니다. 이 사이트에서는 “IP Filter Based Firewalls HOWTO”(Brendan Conoboy and Erik Fichtner, 2002) 자습서에 대한 링크를 비롯하여 오픈 소스 IP 필터에 대한 정보를 제공합니다. 이 자습서는 BSD UNIX 환경에서 방화벽을 구축하는 단계별 지침을 제공합니다. 자습서는 BSD UNIX 환경에 대해 작성된 것이기는 하지만 IP 필터 기능 구성과도 관련이 있습니다.

## IP 필터 패킷 처리

IP 필터는 패킷이 처리되는 일련의 단계를 실행합니다. 다음 다이어그램에서는 패킷 처리 단계 및 필터링과 TCP/IP 프로토콜 스택의 통합 방법을 보여 줍니다.



그림 20-1 패킷 처리 순서



패킷 처리 순서는 다음과 같습니다.

- **NAT(Network Address Translation)**

개인 IP 주소를 다른 공용 주소로 변환하거나 다중 개인 주소의 별칭을 단일 공용 주소로 변환합니다. 기존 네트워크가 있으며 인터넷에 액세스해야 하는 조직에서는 NAT를 통해 IP 주소 소모 문제를 해결할 수 있습니다.

- **IP 계산**

통과하는 바이트 수를 기록하여 입력 및 출력 규칙을 별도로 설정할 수 있습니다. 규칙 일치가 발생할 때마다 패킷 바이트 수가 규칙에 추가되므로 연속 통계를 수집할 수 있습니다.

- **단편 캐시 확인**

현재 트래픽의 다음 패킷이 단편이고 이전 패킷이 허용된 경우 상태 테이블 및 규칙 확인이 무시되어 패킷 단편도 허용됩니다.

- **패킷 상태 확인**

keep state가 규칙에 포함된 경우 규칙이 pass를 의미하는지 아니면 block을 의미하는지에 따라 지정된 세션의 모든 패킷이 자동으로 전달 또는 차단됩니다.

- **방화벽 확인**

IP 필터를 통해 패킷이 허용될지 여부에 따라 커널 TCP/IP 루틴으로 들어오거나 네트워크를 통해 나가는 입력 및 출력 규칙을 별도로 설정할 수 있습니다.

- **그룹**

그룹을 통해 트리 형식으로 규칙 세트를 작성할 수 있습니다.

- **함수**

함수는 수행할 작업입니다. 가능한 함수로는 block, pass, literal 및 send ICMP response가 있습니다.

- **빠른 경로**

빠른 경로는 경로 지정을 위해 패킷이 UNIX IP 스택으로 전달되지 않도록 IP 필터에 신호를 보냅니다. 해당 스택으로 전달될 경우 TTL이 줄어듭니다.

- **IP 인증**

인증 처리를 방지하기 위해 인증된 패킷은 방화벽 루프를 통해 한 번만 전달됩니다.

## IP 필터 사용 지침

- IP 필터는 SMF 서비스 svc:/network/pfil 및 svc:/network/ipfilter를 통해 관리됩니다. SMF에 대한 전체 개요는 **Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리(개요)”**를 참조하십시오. SMF와 관련된 단계별 절차에 대한 자세한 내용은 **Oracle Solaris 관리: 일반 작업의 7 장, “서비스 관리(작업)”**를 참조하십시오.
- IP 필터를 사용하려면 구성 파일을 직접 편집해야 합니다.

- IP 필터는 Oracle Solaris의 일부로 설치됩니다. 기본적으로 새 설치 후 IP 필터가 활성화되지 않습니다. 필터링을 구성하려면 구성 파일을 편집하고 수동으로 IP 필터를 활성화해야 합니다. 시스템을 재부트하거나 `ipadm` 명령으로 인터페이스를 연결하여 필터링을 활성화할 수 있습니다. 자세한 내용은 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오. IP 필터를 사용으로 설정하는 것과 관련된 작업은 301 페이지 “IP 필터 구성”을 참조하십시오.
- IP 필터를 관리하려면 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인할 수 있어야 합니다. 만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.
- IPMP(IP Network Multipathing)는 Stateless 필터링만 지원합니다.  
IP 필터가 IPMP 그룹과 주고 받는 트래픽에 대해 Stateless 필터링을 수행하도록하려면 `ipmp_hook_emulation` 매개변수를 설정해야 합니다. 기본적으로 매개변수는 0으로 설정되어 있습니다. 기본값을 사용할 경우 IP 필터가 IPMP 그룹에 속하는 물리적 인터페이스에서 트래픽에 대해 Stateful 패킷 검사를 수행할 수 없습니다. IPMP 패킷 필터링을 사용으로 설정하려면 다음 명령을 실행하십시오.  

```
ndd -set /dev/ip ipmp_hook_emulation 1
```
- Oracle Solaris Cluster 소프트웨어의 경우 확장 가능한 서비스에 대해서는 IP 필터를 통한 필터링을 지원하지 않지만 패일오버 서비스에 대해서는 IP 필터를 지원합니다. 클러스터에서 IP 필터를 구성하는 경우 지침 및 제한 사항은 **Oracle Solaris Cluster 소프트웨어 설치 설명서**의 “Oracle Solaris OS 기능 제한 사항”을 참조하십시오.
- 시스템의 다른 영역에 대한 가상 라우터로 작동하는 영역에서 IP 필터 규칙이 구현된 경우 영역 간의 필터링이 지원됩니다.

## IP 필터 구성 파일 사용

IP 필터를 사용하여 방화벽 서비스 또는 NAT(Network Address Translation)를 제공할 수 있습니다. 로드 가능한 구성 파일을 통해 IP 필터를 구현할 수 있습니다. IP 필터에는 `/etc/ipf`라는 디렉토리가 있습니다. `ipf.conf`, `ipnat.conf` 및 `ippool.conf`라는 구성 파일을 만들어 `/etc/ipf` 디렉토리에 저장할 수 있습니다. 이러한 파일은 `/etc/ipf` 디렉토리에 상주한 경우 부트 프로세스 중 자동으로 로드됩니다. 구성 파일을 다른 위치에 저장하고 수동으로 파일을 로드할 수도 있습니다. 구성 파일 예는 326 페이지 “IP 필터 구성 파일 만들기 및 편집”을 참조하십시오.

## IP 필터 규칙 세트 사용

방화벽을 관리하려면 IP 필터를 사용하여 네트워크 트래픽 필터링에 사용할 규칙 세트를 지정하십시오. 다음 유형의 규칙 세트를 만들 수 있습니다.

- 패킷 필터링 규칙 세트
- NAT(Network Address Translation) 규칙 세트

또한 IP 주소 그룹을 참조할 주소 풀을 만들 수 있습니다. 그런 다음 나중에 규칙 세트에서 이러한 풀을 사용할 수 있습니다. 주소 풀을 사용하면 규칙 처리 속도가 빨라집니다. 또한 주소 풀을 사용하면 큰 주소 그룹을 간편하게 관리할 수 있습니다.

## IP 필터의 패킷 필터링 기능 사용

패킷 필터링 규칙 세트를 사용하여 패킷 필터링을 설정합니다. `ipf` 명령을 사용하여 패킷 필터링 규칙 세트와 관련된 작업을 수행할 수 있습니다. `ipf` 명령에 대한 자세한 내용은 `ipf(1M)` 명령을 참조하십시오.

명령줄에서 `ipf` 명령을 사용하거나 패킷 필터링 구성 파일에서 패킷 필터링 규칙을 만들 수 있습니다. 부트 시 패킷 필터링 규칙이 로드되도록 하려면 패킷 필터링 규칙을 배치할 `/etc/ipf/ipf.conf`라는 구성 파일을 만듭니다. 부트 시 패킷 필터링 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipf.conf` 파일을 배치하고 `ipf` 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

IP 필터를 사용하여 두 개의 패킷 필터링 규칙 세트(활성 규칙 세트 및 비활성 규칙 세트)를 유지 관리할 수 있습니다. 대부분의 경우 활성 규칙 세트와 관련된 작업을 수행합니다. 하지만 `ipf -I` 명령을 사용하여 비활성 규칙 목록에 명령 작업을 적용할 수 있습니다. 비활성 규칙 목록을 선택하지 않을 경우 해당 목록은 IP 필터에 사용되지 않습니다. 비활성 규칙 목록은 활성 패킷 필터링에 영향을 끼치지 않고 규칙을 저장할 수 있는 위치를 제공합니다.

IP 필터는 패킷을 전달하거나 차단하기 전에 구성된 규칙 목록의 처음부터 규칙 목록의 끝까지 규칙 목록에 있는 규칙을 처리합니다. IP 필터는 패킷 전달 여부를 결정하는 플래그를 유지 관리합니다. 전체 규칙 세트를 확인하고 마지막 일치 규칙을 기반으로 패킷을 전달할지 아니면 차단할지 결정합니다.

이 프로세스에는 두 가지 예외가 있습니다. 첫번째 예외는 패킷이 `quick` 키워드를 포함하는 규칙과 일치하는 경우입니다. 규칙에 `quick` 키워드가 포함되면 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다. 두번째 예외는 패킷이 `group` 키워드를 포함하는 규칙과 일치하는 경우입니다. 패킷이 그룹과 일치되면 그룹 태그가 지정된 규칙만 확인됩니다.

## 패킷 필터링 규칙 구성

다음 구문을 사용하여 패킷 필터링 규칙을 만들 수 있습니다.

*action [in|out] option keyword, keyword...*

1. 각 규칙은 작업으로 시작합니다. IP 필터는 패킷이 규칙과 일치하는 경우 패킷에 작업을 적용합니다. 다음은 패킷에 적용되는 가장 일반적으로 사용되는 작업을 나열한 것입니다.

<b>block</b>	패킷이 필터를 통과하지 못하도록 합니다.
<b>pass</b>	패킷이 필터를 통과할 수 있도록 합니다.
<b>log</b>	패킷을 기록하되 패킷 차단 또는 통과를 결정하지 않습니다. <b>ipmon</b> 명령을 사용하여 로그를 확인할 수 있습니다.
<b>count</b>	필터 통계에 패킷을 포함합니다. <b>ipfstat</b> 명령을 사용하여 통계를 확인할 수 있습니다.
<b>skip number</b>	필터가 <i>number</i> 개의 필터링 규칙을 건너 뛸 수 있도록 합니다.
<b>auth</b>	패킷 정보를 검증하는 사용자 프로그램이 패킷 인증을 수행하도록 요청합니다. 프로그램에서 패킷 전달 또는 차단을 결정합니다.

2. 작업 뒤에 오는 단어는 **in** 또는 **out**이어야 합니다. 선택한 단어에 따라 패킷 필터링 규칙이 수신 패킷에 적용될지 아니면 송신 패킷에 적용될지 결정됩니다.
3. 그런 다음 옵션 목록에서 옵션을 선택할 수 있습니다. 옵션을 두 개 이상 사용할 경우 여기에 표시되는 순서를 따라야 합니다.

<b>log</b>	규칙이 마지막 일치 규칙인 경우 패킷을 기록합니다. <b>ipmon</b> 명령을 사용하여 로그를 확인할 수 있습니다.
<b>quick</b>	패킷 일치가 있을 경우 <b>quick</b> 옵션이 포함된 규칙을 실행합니다. 모든 후속 규칙 확인이 중지됩니다.
<b>on interface-name</b>	패킷이 지정된 인터페이스 내부 또는 외부로 이동되고 있는 경우에만 규칙을 적용합니다.
<b>dup-to interface-name</b>	패킷을 복사하고 <i>interface-name</i> 의 중복 출력을 선택적으로 지정된 IP 주소로 보냅니다.
<b>to interface-name</b>	패킷을 <i>interface-name</i> 의 아웃바운드 대기열로 이동합니다.

4. 옵션을 지정한 후 패킷이 규칙과 일치하는지 여부를 확인하는 다양한 키워드를 선택할 수 있습니다. 다음 키워드는 여기에 표시된 순서대로 사용해야 합니다.

---

주 - 기본적으로 구성 파일의 규칙과 일치하지 않는 패킷은 필터를 통해 전달됩니다.

---

<b>tos</b>	16진수 또는 십진수 정수로 표시되는 <b>type-of-service</b> 값을 기준으로 패킷을 필터링합니다.
<b>tll</b>	<b>time-to-live</b> 값을 기준으로 패킷을 일치시킵니다. 패킷에 저장된 <b>time-to-live</b> 값은 패킷을 폐기하기 전에 네트워크에 보관할 수 있는 기간을 나타냅니다.

<code>proto</code>	특정 프로토콜을 일치시킵니다. <code>/etc/protocols</code> 파일에 지정된 프로토콜 이름을 사용할 수도 있고, 십진수를 사용하여 프로토콜을 나타낼 수도 있습니다. <code>tcp/udp</code> 키워드를 사용하여 TCP 또는 UDP 패킷을 일치시킬 수 있습니다.
<code>from/to/all/ any</code>	소스 IP 주소, 대상 IP 주소, 포트 번호 중 일부 또는 전체와 일치시킵니다. <code>all</code> 키워드는 모든 소스에서 수신되고 모든 대상으로 송신되는 패킷을 승인할 수 있습니다.
<code>with</code>	패킷과 연관되어 있는 지정된 속성을 일치시킵니다. 옵션이 없는 경우에만 패킷을 일치시키려면 키워드 앞에 <code>not</code> 또는 <code>no</code> 단어를 삽입하십시오.
<code>flags</code>	설정된 TCP 플래그를 기준으로 필터링할 TCP에 사용됩니다. TCP 플래그에 대한 자세한 내용은 <a href="#">ipf(4)</a> 매뉴얼 페이지를 참조하십시오.
<code>icmp-type</code>	ICMP 유형에 따라 필터링합니다. 이 키워드는 <code>proto</code> 옵션이 <code>icmp</code> 로 설정된 경우에만 사용되며 <code>flags</code> 옵션이 설정된 경우 사용되지 않습니다.
<code>keep keep-options</code>	패킷에 대해 보관되는 정보를 결정합니다. 사용 가능한 <code>keep-options</code> 로는 <code>state</code> 옵션과 <code>frags</code> 옵션이 있습니다. <code>state</code> 옵션은 세션에 대한 정보를 보관하며 TCP, UDP 및 ICMP 패킷에 보관될 수 있습니다. <code>frags</code> 옵션은 패킷 단편에 정보를 보관하며 후속 단편에 정보를 적용합니다. <code>keep-options</code> 를 사용하면 액세스 제어 목록을 확인하지 않고서도 일치 패킷을 전달할 수 있습니다.
<code>head number</code>	<code>number</code> 번호로 표시되는 필터링 규칙에 대한 새 그룹을 만듭니다.
<code>group number</code>	기본 그룹 대신 그룹 번호 <code>number</code> 에 규칙을 추가합니다. 지정된 다른 그룹이 없을 경우 모든 필터링 규칙이 그룹 0에 배치됩니다.

다음 예에서는 규칙을 만드는 패킷 필터링 규칙 구문을 배치하는 방법을 보여 줍니다. IP 주소 `192.168.0.0/16`의 수신 트래픽을 차단하려면 규칙 목록에 다음 규칙을 포함시킵니다.

```
block in quick from 192.168.0.0/16 to any
```

패킷 필터링 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 [ipf\(4\)](#) 매뉴얼 페이지를 참조하십시오. 패킷 필터링과 관련된 작업은 [308 페이지](#) “IP 필터에 대한 패킷 필터링 규칙 세트 관리”를 참조하십시오. 예에 표시된 IP 주소 체계(`192.168.0.0/16`)에 대한 설명은 [1 장](#), “네트워크 배치 계획”을 참조하십시오.

## IP 필터의 NAT 기능 사용

NAT는 소스 및 대상 IP 주소를 다른 인터넷 또는 인트라넷 주소로 변환하는 매핑 규칙을 설정합니다. 이러한 규칙은 수신 또는 송신 IP 패킷의 소스 및 대상 주소를 수정하고 패킷을 보냅니다. NAT를 사용하여 포트 간에 트래픽을 재지정할 수도 있습니다. NAT는 패킷이 수정되거나 재지정되는 동안 패킷의 무결성을 유지합니다.

`ipnat` 명령을 사용하여 NAT 규칙 목록과 관련된 작업을 수행할 수 있습니다. `ipnat` 명령에 대한 자세한 내용은 `ipnat(1M)` 명령을 참조하십시오.

명령줄에서 `ipnat` 명령을 사용하거나 NAT 구성 파일에서 NAT 규칙을 만들 수 있습니다. NAT 구성 규칙은 `ipnat.conf` 파일에 상주합니다. 부트 시 NAT 규칙이 로드되도록 하려면 NAT 규칙을 배치할 `/etc/ipf/ipnat.conf`라는 파일을 만듭니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치하고 `ipnat` 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

### NAT 규칙 구성

다음 구문을 사용하여 NAT 규칙을 만들 수 있습니다.

*command interface-name parameters*

1. 각 규칙은 다음 명령 중 하나로 시작합니다.

<code>map</code>	제한되지 않은 라운드 로빈 프로세스에서 특정 IP 주소 또는 네트워크를 다른 IP 주소 또는 네트워크에 매핑합니다.
<code>rdr</code>	특정 IP 주소와 포트 쌍의 패킷을 다른 IP 주소와 포트 쌍으로 재지정합니다.
<code>bimap</code>	외부 IP 주소와 내부 IP 주소 간에 양방향 NAT를 설정합니다.
<code>map-block</code>	정적 IP 주소 기반 변환을 설정합니다. 이 명령은 주소를 강제로 대상 범위로 변환하는 알고리즘을 기반으로 합니다.

2. 명령 뒤에 오는 단어는 인터페이스 이름(예: `bge0`)입니다.
3. 그런 다음 NAT 구성을 결정하는 다양한 매개변수를 선택할 수 있습니다. 몇 가지 매개변수는 다음과 같습니다.

<code>ipmask</code>	네트워크 마스크를 지정합니다.
<code>dstipmask</code>	<code>ipmask</code> 가 변환되는 주소를 지정합니다.
<code>mapport</code>	포트 번호 범위와 함께 <code>tcp</code> , <code>udp</code> 또는 <code>tcp/udp</code> 프로토콜을 지정합니다.

다음 예에서는 NAT 규칙을 만드는 NAT 규칙 구문을 배치하는 방법을 보여 줍니다. 소스 주소가 `192.168.1.0/24`인 `de0` 장치에서 송신되는 패킷을 재작성하고 외부적으로 소스 주소를 `10.1.0.0/16`으로 표시하려면 NAT 규칙 세트에 다음 규칙을 포함시킵니다.

```
map de0 192.168.1.0/24 -> 10.1.0.0/16
```

NAT 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 [ipnat\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## IP 필터의 주소 풀 기능 사용

주소 풀은 주소/넷마스크 쌍 그룹의 이름을 지정하는 데 사용되는 단일 참조를 설정합니다. 주소 풀은 IP 주소를 규칙과 일치시키는 데 필요한 시간을 단축시킬 프로세스를 제공합니다. 또한 주소 풀을 사용하면 큰 주소 그룹을 간편하게 관리할 수 있습니다.

주소 풀 구성 규칙은 `ippool.conf` 파일에 상주합니다. 부트 시 주소 풀 규칙이 로드되도록 하려면 주소 풀 규칙을 배치할 `/etc/ipf/ippool.conf`라는 파일을 만듭니다. 부트 시 주소 풀 규칙이 로드되지 않도록 하려면 선택한 위치에 `ippool.conf` 파일을 배치하고 `ippool` 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

### 주소 풀 구성

다음 구문을 사용하여 주소 풀을 만들 수 있습니다.

```
table role = role-name type = storage-format number = reference-number
```

**table**      여러 주소에 대한 참조를 정의합니다.

**role**        IP 필터의 풀 역할을 지정합니다. 지금은 `ipf` 역할만 참조할 수 있습니다.

**type**        풀에 대한 저장 형식을 지정합니다.

**number**      필터링 규칙에 사용되는 참조 번호를 지정합니다.

예를 들어, `10.1.1.1` 및 `10.1.1.2` 주소 그룹과 `192.16.1.0` 네트워크를 풀 번호 13으로 참조하려면 주소 풀 구성 파일에 다음 규칙을 포함시킵니다.

```
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24 };
```

그런 다음 필터링 규칙의 풀 번호 13을 참조하려면 다음 예와 유사한 규칙을 생성합니다.

```
pass in from pool/13 to any
```

풀에 대한 참조를 포함하는 규칙 파일을 로드하기 전에 풀 파일을 로드해야 합니다. 그렇지 않을 경우 다음 출력과 같이 풀이 정의되지 않습니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in from pool/13(!) to any
```



나중에 풀을 추가하는 경우에도 풀 추가로 인해 커널 규칙 세트가 업데이트되지 않습니다. 또한 풀을 참조하는 규칙 파일을 재로드해야 합니다.

패킷 필터링 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 [ippool\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## 패킷 필터 후크

현재 릴리스에서는 패킷 필터 후크가 `pfil` 모듈을 대체하여 IP 필터를 사용으로 설정하는 데 사용됩니다. 이전 Solaris 릴리스에서는 추가 IP 필터 설정 단계로 `pfil` 모듈을 구성해야 했습니다. 이 추가 구성 요구 사항으로 인해 IP 필터가 제대로 작동하지 않을 수 있는 오류 발생 위험이 많았습니다. 또한 IP와 장치 드라이버 사이에 `pfil` STREAMS 모듈을 삽입하는 것도 성능 저하의 원인이었습니다. 마지막으로 `pfil` 모듈은 영역 간에 패킷 가로채기를 수행할 수 없습니다.

패킷 필터 후크를 사용하면 IP 필터를 사용으로 설정하는 절차가 간소화됩니다. 이러한 후크를 통해 IP 필터는 사전 경로 지정(입력) 및 사후 경로 지정(출력) 필터 탭을 사용하여 Oracle Solaris 시스템에서의 패킷 흐름 입력 및 출력을 제어합니다.

패킷 필터 후크를 사용하면 `pfil` 모듈이 필요하지 않습니다. 따라서 모듈과 연관된 다음 구성 요소도 제거되었습니다.

- `pfil` 드라이버
- `pfil` 데몬
- `svc:/network/pfil` SMF 서비스

IP 필터를 사용으로 설정하는 것과 관련된 작업은 [21 장](#), “[IP 필터\(작업\)](#)”를 참조하십시오.

## IP 필터용 IPv6

Solaris 6/06 릴리스부터 IP 필터와 함께 IPv6에 대한 지원이 제공됩니다. IPv6 패킷 필터링은 소스/대상 IPv6 주소, IPv6 주소를 포함하는 풀 및 IPv6 확장 헤더를 기준으로 필터링을 수행할 수 있습니다.

IPv6은 여러 측면에서 IPv4와 유사합니다. 단, IP의 두 버전 간에 헤더 및 패킷 크기가 다르므로 IP 필터를 사용할 때 반드시 고려해야 합니다. IPv6 패킷([점보그램](#)이라고도 함)에는 65,535바이트 이상의 데이터그램이 포함되어 있습니다. IP 필터는 IPv6 점보그램을 지원하지 않습니다. 기타 IPv6 기능에 대해 자세히 알아보려면 [System Administration Guide: IP Services](#)의 “[Major Features of IPv6](#)”을 참조하십시오.

주 - 점보그램에 대한 자세한 내용은 IETF(Internet Engineering Task Force)[<http://www.ietf.org/rfc/rfc2675.txt>]의 IPv6 Jumbograms, RFC 2675 문서를 참조하십시오.

IPv6과 관련된 IP 필터 작업은 IPv4와 유사합니다. 가장 큰 차이는 특정 명령에 -6 옵션을 사용한다는 점입니다. `ipf` 명령과 `ipfstat` 명령에는 IPv6 패킷 필터링에 사용할 -6 옵션이 포함됩니다. `ipf` 명령에 -6 옵션을 사용하여 IPv6 패킷 필터링 규칙을 로드하고 비울 수 있습니다. IPv6 통계를 표시하려면 `ipfstat` 명령에 -6 옵션을 사용하십시오. `ipmon` 및 `ippool` 명령도 IPv6을 지원하지만 IPv6 지원과 관련된 옵션이 없습니다. `ipmon` 명령이 IPv6 패킷 로깅을 수행하도록 개선되었습니다. `ippool` 명령은 IPv6 주소와 함께 풀을 지원합니다. IPv4 주소와 IPv6 주소 중 하나에 대해서만 풀을 만들 수도 있고, 동일한 풀 내에 IPv4 주소와 IPv6 주소를 모두 포함하는 풀을 만들 수도 있습니다.

`ipf6.conf` 파일을 사용하여 IPv6에 대한 패킷 필터링 규칙 세트를 만들 수 있습니다. 기본적으로 `ipf6.conf` 구성 파일은 `/etc/ipf` 디렉토리에 포함됩니다. 다른 필터링 구성 파일에서와 마찬가지로 `ipf6.conf` 파일은 부트 프로세스 동안 자동으로 로드됩니다(이 파일이 `/etc/ipf` 디렉토리에 저장되어 있는 경우). 다른 위치에 IPv6 구성 파일을 만들어 저장하고 수동으로 파일을 로드할 수도 있습니다.

IPv6에 대한 패킷 필터링 규칙이 설정되면 인터페이스를 만들어 IPv6 패킷 필터링 기능을 활성화하십시오.

IPv6에 대한 자세한 내용은 **System Administration Guide: IP Services**의 3 장, “Introducing IPv6 (Overview)”를 참조하십시오. IP 필터와 관련된 작업은 21 장, “IP 필터(작업)”를 참조하십시오.

## IP 필터 매뉴얼 페이지

다음 표에서는 IP 필터와 관련된 매뉴얼 페이지에 대해 설명합니다.

매뉴얼 페이지	설명
<a href="#">ipf(1M)</a>	다음 작업을 완료하려면 <code>ipf</code> 명령을 사용합니다. <ul style="list-style-type: none"> <li>■ 패킷 필터링 규칙 세트와 관련된 작업을 수행합니다.</li> <li>■ 필터링을 사용 안함/사용으로 설정합니다.</li> <li>■ 통계를 재설정하고 커널 내 인터페이스 목록을 현재 인터페이스 상태 목록과 다시 동기화합니다.</li> </ul>
<a href="#">ipf(4)</a>	IP 필터 패킷 필터링 규칙 생성 문법 및 구문을 포함합니다.
<a href="#">ipfilter(5)</a>	오픈 소스 IP 필터 라이선스 정보를 제공합니다.

매뉴얼 페이지	설명
<a href="#">ipfs(1M)</a>	재부트 시 NAT 정보 및 상태 테이블 정보를 저장하고 복원하려면 <code>ipfs</code> 명령을 사용합니다.
<a href="#">ipfstat(1M)</a>	패킷 처리에 대한 통계를 검색하고 표시하려면 <code>ipfstat</code> 명령을 사용합니다.
<a href="#">ipmon(1M)</a>	로그 장치를 열고 패킷 필터링과 NAT에 대해 기록된 패킷을 보려면 <code>ipmon</code> 명령을 사용합니다.
<a href="#">ipnat(1M)</a>	다음 작업을 완료하려면 <code>ipnat</code> 명령을 사용합니다. <ul style="list-style-type: none"> <li>■ NAT 규칙과 관련된 작업을 수행합니다.</li> <li>■ NAT 통계를 검색하고 표시합니다.</li> </ul>
<a href="#">ipnat(4)</a>	NAT 규칙 생성 문법 및 구문을 포함합니다.
<a href="#">ippool(1M)</a>	주소 풀을 만들고 관리하려면 <code>ippool</code> 명령을 사용합니다.
<a href="#">ippool(4)</a>	IP 필터 주소 풀 생성 문법 및 구문을 포함합니다.
<a href="#">ndd(1M)</a>	<code>pfil</code> STREAMS 모듈의 현재 필터링 매개변수 및 조정 가능한 매개변수의 현재 값을 표시합니다.



## IP 필터(작업)

이 장에서는 단계별 작업 지침을 제공합니다. IP 필터에 대한 개요 정보는 20 장, “Oracle Solaris의 IP 필터(개요)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

- 301 페이지 “IP 필터 구성”
- 305 페이지 “IP 필터 비활성화 및 사용 안함으로 설정”
- 307 페이지 “IP 필터 규칙 세트 작업”
- 318 페이지 “IP 필터에 대한 통계 및 정보 표시”
- 322 페이지 “IP 필터 로그 파일 작업”
- 326 페이지 “IP 필터 구성 파일 만들기 및 편집”

## IP 필터 구성

다음 작업 맵에서는 IP 필터 구성과 관련된 절차를 식별합니다.

표 21-1 IP 필터 구성(작업 맵)

작업	설명	수행 방법
초기에 IP 필터를 사용으로 설정합니다.	기본적으로 IP 필터는 사용으로 설정되어 있지 않습니다. 수동으로 사용으로 설정하거나 <code>/etc/ipf/</code> 디렉토리의 구성 파일을 사용한 후 시스템을 재부트해야 합니다. 패킷 필터 후크가 <code>pfil</code> 모듈을 대체하여 IP 필터를 사용으로 설정하는 데 사용됩니다.	302 페이지 “IP 필터를 사용으로 설정하는 방법”

표 21-1 IP 필터 구성(작업 맵) (계속)

작업	설명	수행 방법
IP 필터를 다시 사용으로 설정합니다.	IP 필터가 비활성화되거나 사용 안함으로 설정된 경우 시스템을 재부트하거나 <code>ipf</code> 명령을 사용하여 IP 필터를 다시 사용으로 설정할 수 있습니다.	303 페이지 “IP 필터를 다시 사용으로 설정하는 방법”
루프백 필터링을 사용으로 설정합니다.	선택적으로 영역 간의 트래픽을 필터링하는 등의 용도로 루프백 필터링을 사용으로 설정할 수 있습니다.	304 페이지 “루프백 필터링을 사용으로 설정하는 방법”

## ▼ IP 필터를 사용으로 설정하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 패킷 필터링 규칙 세트를 만듭니다.

패킷 필터링 규칙 세트에는 IP 필터에 사용되는 패킷 필터링 규칙이 포함되어 있습니다. 부트 시 패킷 필터링 규칙이 로드되도록 하려면 IPv4 패킷 필터링이 구현되도록 `/etc/ipf/ipf.conf` 파일을 편집합니다. IPv6 패킷 필터링 규칙에 `/etc/ipf/ipf6.conf` 파일을 사용합니다. 부트 시 패킷 필터링 규칙이 로드되지 않도록 하려면 선택한 파일에 규칙을 배치하고 수동으로 패킷 필터링을 활성화합니다. 패킷 필터링에 대한 자세한 내용은 [292 페이지 “IP 필터의 패킷 필터링 기능 사용”](#)을 참조하십시오. 구성 파일 사용에 대한 자세한 내용은 [326 페이지 “IP 필터 구성 파일 만들기 및 편집”](#)를 참조하십시오.

- 3 (옵션) NAT(Network Address Translation) 구성 파일을 만듭니다.

주 - NAT(Network Address Translation)는 IPv6을 지원하지 않습니다.

NAT를 사용하려면 `ipnat.conf` 파일을 만듭니다. 부트 시 NAT 규칙이 로드되도록 하려면 NAT 규칙을 배치할 `/etc/ipf/ipnat.conf`라는 파일을 만듭니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치하고 수동으로 NAT 규칙을 활성화합니다.

NAT에 대한 자세한 내용은 [295 페이지 “IP 필터의 NAT 기능 사용”](#)을 참조하십시오.

#### 4 (옵션) 주소 풀 구성 파일을 만듭니다.

단일 주소 풀로 사용할 주소 그룹을 나타내려면 `ipool.conf` 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되도록 하려면 주소 풀을 배치할 `/etc/ipf/ippool.conf` 라는 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되지 않도록 하려면 선택한 위치에 `ippool.conf` 파일을 배치하고 수동으로 규칙을 활성화합니다.

주소 풀에는 IPv4 주소와 IPv6 주소 중 하나만 포함될 수도 있고, IPv4 주소와 IPv6 주소가 모두 포함될 수도 있습니다.

주소 풀에 대한 자세한 내용은 296 페이지 “IP 필터의 주소 풀 기능 사용”을 참조하십시오.

#### 5 (옵션) 루프백 트래픽의 필터링을 사용으로 설정합니다.

시스템에서 구성된 영역 간의 트래픽을 필터링하려면 루프백 필터링을 사용으로 설정해야 합니다. 304 페이지 “루프백 필터링을 사용으로 설정하는 방법”을 참조하십시오. 영역에 적용할 적합한 규칙 세트도 정의해야 합니다.

#### 6 IP 필터를 활성화합니다.

```
# svcadm enable network/ipfilter
```

## ▼ IP 필터를 다시 사용으로 설정하는 방법

패킷 필터링을 일시적으로 사용 안함으로 설정한 후 다시 사용으로 설정할 수 있습니다.

#### 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

#### 2 다음 방법 중 하나로 IP 필터를 사용으로 설정하고 필터링을 활성화합니다.

- 시스템을 재부트합니다.

```
# reboot
```

---

주 - IP 필터가 사용으로 설정되면 재부트 후 `/etc/ipf/ipf.conf` 파일, `/etc/ipf/ipf6.conf` 파일(IPv6을 사용하는 경우) 또는 `/etc/ipf/ipnat.conf` 파일이 있을 경우 로드됩니다.

---

- 다음과 같은 일련의 명령을 실행하여 IP 필터를 사용으로 설정하고 필터링을 활성화합니다.

- a. IP 필터를 사용으로 설정합니다.

```
# ipf -E
```

- b. 패킷 필터링을 활성화합니다.

```
# ipf -f filename
```

- c. (옵션) NAT를 활성화합니다.

```
# ipnat -f filename
```

---

주 - NAT(Network Address Translation)는 IPv6을 지원하지 않습니다.

---

## ▼ 루프백 필터링을 사용으로 설정하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 IP 필터가 실행 중인 경우 중지합니다.

```
# svcadm disable network/ipfilter
```

- 3 파일 시작 부분에 다음 행을 추가하여 `/etc/ipf.conf` 또는 `/etc/ipf6.conf` 파일을 편집합니다.

```
set intercept_loopback true;
```

파일에서 정의된 모든 IP 필터 규칙 앞에 이 행이 와야 합니다. 단, 다음 예와 유사하게 행 앞에 주석을 삽입할 수 있습니다.

```
#
# Enable loopback filtering to filter between zones
#
set intercept_loopback true;
#
# Define policy
#
block in all
block out all
<other rules>
...
```

- 4 IP 필터를 시작합니다.

```
# svcadm enable network/ipfilter
```

- 5 루프백 필터링 상태를 확인하려면 다음 명령을 사용합니다.

```
# ipf -T ipf_loopback
ipf_loopback    min 0    max 0x1 current 1
#
```



루프백 필터링이 사용 안함으로 설정된 경우 명령으로 다음 출력이 생성됩니다.

```
ipf_loopback    min 0    max 0x1 current 0
```

## IP 필터 비활성화 및 사용 안함으로 설정

다음과 같은 경우 패킷 필터링 및 NAT를 비활성화하거나 사용 안함으로 설정할 수 있습니다.

- 테스트 용도로 사용하려는 경우
- 문제의 원인이 IP 필터인 것으로 간주되어 시스템 문제를 해결하려는 경우

다음 작업 맵에서는 IP 필터 기능을 비활성화하거나 사용 안함으로 설정하는 것과 관련된 절차를 식별합니다.

표 21-2 IP 필터 비활성화 및 사용 안함으로 설정(작업 맵)

작업	설명	수행 방법
패킷 필터링을 비활성화합니다.	ipf 명령을 사용하여 패킷 필터링을 비활성화합니다.	305 페이지 “패킷 필터링 비활성화 방법”
NAT를 비활성화합니다.	ipnat 명령을 사용하여 NAT를 비활성화합니다.	306 페이지 “NAT 비활성화 방법”
패킷 필터링 및 NAT를 사용 안함으로 설정합니다.	ipf 명령을 사용하여 패킷 필터링 및 NAT를 사용 안함으로 설정합니다.	306 페이지 “패킷 필터링을 사용 안함으로 설정하는 방법”

### ▼ 패킷 필터링 비활성화 방법

다음 절차에서는 활성 필터링 규칙 세트에서 패킷 필터링 규칙을 비워 IP 필터 패킷 필터링을 비활성화합니다. 이 절차에서는 IP 필터를 사용 안함으로 설정하지 않습니다. 규칙 세트에 규칙을 추가하여 IP 필터를 재활성화할 수 있습니다.

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.

- 2 다음 방법 중 하나로 IP 필터 규칙을 비활성화합니다.

- 커널에서 활성 규칙 세트를 제거합니다.

```
# ipf -Fa
```

이 명령은 모든 패킷 필터링 규칙을 비활성화합니다.

- 수신 패킷 필터링 규칙을 제거합니다.

```
# ipf -Fi
```

이 명령은 수신 패킷에 대한 패킷 필터링 규칙을 비활성화합니다.

- 송신 패킷 필터링 규칙을 제거합니다.

```
# ipf -Fo
```

이 명령은 송신 패킷에 대한 패킷 필터링 규칙을 비활성화합니다.

## ▼ NAT 비활성화 방법

다음 절차에서는 활성 NAT 규칙 세트에서 NAT 규칙을 비워 IP 필터 NAT 규칙을 비활성화합니다. 이 절차에서는 IP 필터를 사용 안함으로 설정하지 않습니다. 규칙 세트에 규칙을 추가하여 IP 필터를 재활성화할 수 있습니다.

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 커널에서 NAT를 제거합니다.

```
# ipnat -FC
```

-C 옵션은 현재 NAT 규칙 목록의 모든 항목을 제거합니다. -F 옵션은 현재 활성 NAT 매핑을 보여 주는 현재 NAT 변환 테이블의 모든 활성 항목을 제거합니다.

## ▼ 패킷 필터링을 사용 안함으로 설정하는 방법

이 절차를 실행하면 커널에서 패킷 필터링과 NAT가 모두 제거됩니다. 이 절차를 사용할 경우 패킷 필터링 및 NAT를 재활성화하려면 IP 필터를 다시 사용으로 설정해야 합니다. 자세한 내용은 [303 페이지 “IP 필터를 다시 사용으로 설정하는 방법”](#)을 참조하십시오.

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 패킷 필터링을 사용 안함으로 설정하고 모든 패킷을 네트워크로 전달할 수 있도록 허용합니다.

# ipf -D

주 - ipf -D 명령은 규칙 세트에서 규칙을 비웁니다. 필터링을 다시 사용으로 설정하는 경우 규칙 세트에 규칙을 추가해야 합니다.

## IP 필터 규칙 세트 작업

다음 작업 맵에서는 IP 필터 규칙 세트와 관련된 절차를 식별합니다.

표 21-3 IP 필터 규칙 세트 작업(작업 맵)

작업	설명	수행 방법
IP 필터 패킷 필터링 규칙 세트를 관리, 확인 및 수정합니다.		308 페이지 “IP 필터에 대한 패킷 필터링 규칙 세트 관리”
	활성 패킷 필터링 규칙 세트를 확인합니다.	308 페이지 “활성 패킷 필터링 규칙 세트 확인 방법”
	비활성 패킷 필터링 규칙 세트를 확인합니다.	309 페이지 “비활성 패킷 필터링 규칙 세트 확인 방법”
	다른 활성 규칙 세트를 활성화합니다.	309 페이지 “다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법”
	규칙 세트를 제거합니다.	311 페이지 “패킷 필터링 규칙 세트 제거 방법”
	규칙 세트에 규칙을 추가합니다.	311 페이지 “활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법” 312 페이지 “비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법”
	활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.	313 페이지 “활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법”
	커널에서 비활성 규칙 세트를 삭제합니다.	314 페이지 “커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법”
IP 필터 NAT 규칙을 관리, 확인 및 수정합니다.		314 페이지 “IP 필터에 대한 NAT 규칙 관리”

표 21-3 IP 필터 규칙 세트 작업(작업 맵) (계속)

작업	설명	수행 방법
IP 필터 주소 풀을 관리, 확인 및 수정합니다.	활성 NAT 규칙을 확인합니다.	315 페이지 “활성 NAT 규칙 확인 방법”
	NAT 규칙을 제거합니다.	315 페이지 “NAT 규칙 제거 방법”
	NAT 규칙에 다른 규칙을 추가합니다.	316 페이지 “NAT 규칙에 규칙을 추가하는 방법”
		316 페이지 “IP 필터에 대한 주소 풀 관리”
	활성 주소 풀을 확인합니다.	317 페이지 “활성 주소 풀 확인 방법”
	주소 풀을 제거합니다.	317 페이지 “주소 풀 제거 방법”
	주소 풀에 다른 규칙을 추가합니다.	318 페이지 “주소 풀에 규칙을 추가하는 방법”

## IP 필터에 대한 패킷 필터링 규칙 세트 관리

사용으로 설정된 경우 활성 및 비활성 패킷 필터링 규칙 세트가 모두 커널에 상주할 수 있습니다. 활성 규칙 세트에 따라 수신 패킷 및 송신 패킷에 대해 수행하려는 필터링이 결정됩니다. 비활성 규칙 세트도 규칙을 저장합니다. 비활성 규칙 세트를 활성 규칙 세트로 설정하지 않은 경우 해당 규칙이 사용되지 않습니다. 활성 및 비활성 패킷 필터링 규칙 세트를 모두 관리, 확인 및 수정할 수 있습니다.

### ▼ 활성 패킷 필터링 규칙 세트 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스](#)의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

- 2 커널에서 로드된 활성 패킷 필터링 규칙 세트를 확인합니다.

```
# ipfstat -io
```

#### 예 21-1 활성 패킷 필터링 규칙 세트 보기

다음 예에서는 커널에서 로드된 활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe1 from 192.168.1.0/24 to any
```

```
pass in all
block in on dmfe1 from 192.168.1.10/32 to any
```

## ▼ 비활성 패킷 필터링 규칙 세트 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 비활성 패킷 필터링 규칙 세트를 확인합니다.

```
# ipfstat -I -io
```

### 예 21-2 비활성 패킷 필터링 규칙 세트 보기

다음 예에서는 비활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
```

## ▼ 다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법

다음 작업 중 하나를 수행하려면 이 절차를 사용하십시오.

- 현재 IP 필터에 사용되고 있는 규칙 세트가 아닌 다른 패킷 필터링 규칙 세트를 활성화합니다.
- 새로 업데이트된 동일한 필터링 규칙 세트를 재로드합니다.

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 다음 단계 중 하나를 선택합니다.

- 완전히 다른 규칙 세트를 활성화하려면 선택한 별도의 파일에 새 규칙 세트를 만듭니다.
- 해당 규칙 세트를 포함하는 구성 파일을 편집하여 현재 규칙 세트를 업데이트합니다.

- 3 현재 규칙 세트를 제거하고 새 규칙 세트를 로드합니다.

```
# ipf -Fa -f filename
```

*filename*은 새 규칙 세트를 포함하는 새 파일 또는 활성 규칙 세트를 포함하는 업데이트된 파일일 수 있습니다.

커널에서 활성 규칙 세트가 제거되고, *filename* 파일의 규칙이 활성 규칙 세트가 됩니다.

---

**주** - 현재 구성 파일을 재로드하는 중인 경우에도 명령을 실행해야 합니다. 그렇지 않으면 기존 규칙 세트가 계속 작동하고 업데이트된 구성 파일의 수정된 규칙 세트가 적용되지 않습니다.

업데이트된 규칙 세트를 로드하려면 `ipf -D, svcadm restart` 등의 명령을 사용하지 마십시오. 새 규칙 세트를 로드하기 전에 먼저 방화벽을 사용 안함으로 설정하면 해당 명령으로 인해 네트워크가 노출됩니다.

---

### 예 21-3 다른 패킷 필터링 규칙 세트 활성화

다음 예에서는 특정 패킷 필터링 규칙 세트를 별도의 구성 파일 `/etc/ipf/ipf.conf`에 있는 다른 패킷 필터링 규칙 세트로 바꾸는 방법을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe all
# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
```

### 예 21-4 업데이트된 패킷 필터링 규칙 세트 재로드

다음 예에서는 현재 활성 상태이며 업데이트된 패킷 필터링 규칙 세트를 재로드하는 방법을 보여 줍니다. 이 예에서 사용하는 파일은 `/etc/ipf/ipf.conf`입니다.

```
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any

(Edit the /etc/ipf/ipf.conf configuration file.)

# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any
block in quick on elx10 from 192.168.0.0/12 to any
```

## ▼ 패킷 필터링 규칙 세트 제거 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스**의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

- 2 규칙 세트를 제거합니다.

```
# ipf -F [a|i|o]
-a   규칙 세트에서 모든 필터링 규칙을 제거합니다.
-i   수신 패킷에 대한 필터링 규칙을 제거합니다.
-o   송신 패킷에 대한 필터링 규칙을 제거합니다.
```

### 예 21-5 패킷 필터링 규칙 세트 제거

다음 예에서는 활성 필터링 규칙 세트에서 모든 필터링 규칙을 제거하는 방법을 보여 줍니다.

```
# ipfstat -io
block out log on dmfo all
block in log quick from 10.0.0.0/8 to any
# ipf -Fa
# ipfstat -io
empty list for ipfilter(out)
empty list for ipfilter(in)
```

## ▼ 활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스**의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

- 2 다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.

- ipf -f - 명령을 사용하여 명령줄에서 규칙 세트에 규칙을 추가합니다.
 

```
# echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
```
- 다음 명령을 실행합니다.
  - a. 선택한 파일에 규칙 세트를 만듭니다.
  - b. 만든 규칙을 활성 규칙 세트에 추가합니다.

```
# ipf -f filename
```

활성 규칙 세트의 끝에 *filename*의 규칙이 추가됩니다. IP 필터는 “마지막 일치 규칙” 알고리즘을 사용하므로 **quick** 키워드를 사용하지 않는 경우 추가되는 규칙에 따라 필터링 우선 순위가 결정됩니다. 패킷이 **quick** 키워드를 포함하는 규칙과 일치하는 경우 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다.

## 예 21-6 활성 패킷 필터링 규칙 세트에 규칙 추가

다음 예에서는 명령줄에서 활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
# echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

## ▼ 비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스**의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

- 2 선택한 파일에 규칙 세트를 만듭니다.
- 3 만든 규칙을 비활성 규칙 세트에 추가합니다.

```
# ipf -I -f filename
```

비활성 규칙 세트의 끝에 *filename*의 규칙이 추가됩니다. IP 필터는 “마지막 일치 규칙” 알고리즘을 사용하므로 **quick** 키워드를 사용하지 않는 경우 추가되는 규칙에 따라 필터링 우선 순위가 결정됩니다. 패킷이 **quick** 키워드를 포함하는 규칙과 일치하는 경우 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다.

## 예 21-7 비활성 규칙 세트에 규칙 추가

다음 예에서는 파일에서 비활성 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipfstat -I -io
pass out quick on dmfe1 all
```



```
pass in quick on dmfe1 all
# ipf -I -f /etc/ipf/ipf.conf
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
```

## ▼ 활성화 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 활성화 규칙 세트와 비활성 규칙 세트 간에 전환합니다.

```
# ipf -s
```

이 명령을 사용하면 커널에서 활성화 규칙 세트와 비활성 규칙 세트 간에 전환할 수 있습니다. 비활성 규칙 세트가 비어 있을 경우 패킷 필터링이 없는 것입니다.

### 예 21-8 활성화 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환

다음 예에서는 ipf -s 명령을 사용하여 비활성 규칙 세트를 활성화 규칙 세트로 전환하고 활성화 규칙 세트를 비활성 규칙 세트로 전환하는 방법을 보여 줍니다.

- ipf -s 명령을 실행하기 전에 ipfstat -I -io 명령의 출력은 비활성 규칙 세트의 규칙을 보여 줍니다. ipfstat -io 명령의 출력은 활성화 규칙 세트의 규칙을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
```

- ipf -s 명령을 실행한 후 ipfstat -I -io 및 ipfstat -io 명령의 출력은 두 개 규칙 세트의 내용이 전환되었음을 보여 줍니다.

```
# ipf -s
Set 1 now inactive
# ipfstat -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
# ipfstat -I -io
```

```
empty list for inactive ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

## ▼ 커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 "모두 비우기" 명령에 비활성 규칙 세트를 지정합니다.

```
# ipf -I -Fa
```

이 명령은 커널에서 비활성 규칙 세트를 비웁니다.

---

주 - 나중에 ipf -s를 실행할 경우 비어 있는 비활성 규칙 세트가 활성 규칙 세트로 전환됩니다. 활성 규칙 세트가 비어 있을 경우 필터링이 수행되지 **않습니다**.

---

### 예 21-9 커널에서 비활성 패킷 필터링 규칙 세트 제거

다음 예에서는 모든 규칙이 제거되도록 비활성 패킷 필터링 규칙 세트를 비우는 방법을 보여 줍니다.

```
# ipfstat -I -io
empty list for inactive ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
# ipf -I -Fa
# ipfstat -I -io
empty list for inactive ipfilter(out)
empty list for inactive ipfilter(in)
```

## IP 필터에 대한 NAT 규칙 관리

다음 절차에 따라 NAT 규칙을 관리, 확인 및 수정할 수 있습니다.

## ▼ 활성 NAT 규칙 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 활성 NAT 규칙을 확인합니다.

```
# ipnat -l
```

### 예 21-10 활성 NAT 규칙 보기

다음 예에서는 활성 NAT 규칙 세트의 출력을 보여 줍니다.

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

## ▼ NAT 규칙 제거 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 현재 NAT 규칙을 제거합니다.

```
# ipnat -C
```

### 예 21-11 NAT 규칙 제거

다음 예에서는 현재 NAT 규칙의 항목을 제거하는 방법을 보여 줍니다.

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
# ipnat -C
1 entries flushed from NAT list
# ipnat -l
List of active MAP/Redirect filters:
```

List of active sessions:

## ▼ NAT 규칙에 규칙을 추가하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스**의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

- 2 다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.

- `ipnat -f` - 명령을 사용하여 명령줄에서 NAT 규칙 세트에 규칙을 추가합니다.

```
# echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
```

- 다음 명령을 실행합니다.

- a. 선택한 파일에 추가 NAT 규칙을 만듭니다.

- b. 만든 규칙을 활성 NAT 규칙에 추가합니다.

```
# ipnat -f filename
```

NAT 규칙의 끝에 *filename*의 규칙이 추가됩니다.

### 예 21-12 NAT 규칙 세트에 규칙 추가

다음 예에서는 명령줄에서 NAT 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipnat -l
```

List of active MAP/Redirect filters:

List of active sessions:

```
# echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
```

```
# ipnat -l
```

List of active MAP/Redirect filters:

```
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32
```

List of active sessions:

## IP 필터에 대한 주소 풀 관리

다음 절차에 따라 주소 풀을 관리, 확인 및 수정할 수 있습니다.

## ▼ 활성 주소 풀 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 활성 주소 풀을 확인합니다.

```
# ippool -l
```

### 예 21-13 활성 주소 풀 보기

다음 예에서는 활성 주소 풀의 콘텐츠를 확인하는 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
    { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

## ▼ 주소 풀 제거 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 현재 주소 풀의 항목을 제거합니다.

```
# ippool -F
```

### 예 21-14 주소 풀 제거

다음 예에서는 주소 풀 제거 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
    { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# ippool -F
1 object flushed
# ippool -l
```

▼ 주소 풀에 규칙을 추가하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.  
만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스**의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.
- 2 다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.
  - `ippool -f` - 명령을 사용하여 명령줄에서 규칙 세트에 규칙을 추가합니다.

```
# echo "table role = ipf type = tree number = 13
{10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24};" | ippool -f -
```
  - 다음 명령을 실행합니다.
    - a. 선택한 파일에 추가 주소 풀을 만듭니다.
    - b. 만든 규칙을 활성 주소 풀에 추가합니다.

```
# ippool -f filename
```

활성 주소 풀의 끝에 *filename*의 규칙이 추가됩니다.

예 21-15 주소 풀에 규칙 추가

다음 예에서는 명령줄에서 주소 풀 규칙 세트에 주소 풀을 추가하는 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# echo "table role = ipf type = tree number = 100
{10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24};" | ippool -f -
# ippool -l
table role = ipf type = tree number = 100
{ 10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24; };
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

IP 필터에 대한 통계 및 정보 표시

표 21-4 IP 필터 통계 및 정보 표시(작업 맵)

작업	설명	수행 방법
상태 테이블을 확인합니다.	<code>ipfstat</code> 명령을 사용하여 패킷 필터링에 대한 정보를 얻을 수 있는 상태 테이블을 확인합니다.	319 페이지 “IP 필터에 대한 상태 테이블 확인 방법”

표 21-4 IP 필터 통계 및 정보 표시(작업 맵) (계속)

작업	설명	수행 방법
상태 통계를 확인합니다.	<code>ipfstat -s</code> 명령을 사용하여 패킷 상태 정보에 대한 통계를 확인합니다.	320 페이지 “IP 필터에 대한 상태 통계 확인 방법”
NAT 통계를 확인합니다.	<code>ipnat -s</code> 명령을 사용하여 NAT 통계를 확인합니다.	321 페이지 “IP 필터에 대한 NAT 통계 확인 방법”
주소 풀 통계를 확인합니다.	<code>ippool -s</code> 명령을 사용하여 주소 풀 통계를 확인합니다.	321 페이지 “IP 필터에 대한 주소 풀 통계 확인 방법”

## ▼ IP 필터에 대한 상태 테이블 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 상태 테이블을 확인합니다.

```
# ipfstat
```

주 -t 옵션을 사용하여 최상위 유틸리티 형식으로 상태 테이블을 확인할 수 있습니다.

### 예 21-16 IP 필터에 대한 상태 테이블 보기

다음 예에서는 상태 테이블 확인 방법을 보여 줍니다.

```
# ipfstat
bad packets:           in 0    out 0
  input packets:       blocked 160 passed 11 nomatch 1 counted 0 short 0
output packets:       blocked 0 passed 13681 nomatch 6844 counted 0 short 0
  input packets logged: blocked 0 passed 0
output packets logged: blocked 0 passed 0
  packets logged:      input 0 output 0
log failures:          input 0 output 0
fragment state(in):    kept 0  lost 0
fragment state(out):    kept 0  lost 0
packet state(in):      kept 0  lost 0
packet state(out):      kept 0  lost 0
ICMP replies: 0        TCP RSTs sent: 0
Invalid source(in):    0
Result cache hits(in): 152      (out): 6837
IN Pullups succeeded: 0         failed: 0
OUT Pullups succeeded: 0         failed: 0
Fastroute successes: 0         failures: 0
```

```
TCP cksum fails(in): 0      (out): 0
IPF Ticks: 14341469
Packet log flags set: (0)
none
```

## ▼ IP 필터에 대한 상태 통계 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 상태 통계를 확인합니다.

```
# ipfstat -s
```

### 예 21-17 IP 필터에 대한 상태 통계 보기

다음 예에서는 상태 통계 확인 방법을 보여 줍니다.

```
# ipfstat -s
IP states added:
    0 TCP
    0 UDP
    0 ICMP
    0 hits
    0 misses
    0 maximum
    0 no memory
    0 max bucket
    0 active
    0 expired
    0 closed
State logging enabled

State table bucket statistics:
    0 in use
    0.00% bucket usage
    0 minimal length
    0 maximal length
    0.000 average length
```



## ▼ IP 필터에 대한 NAT 통계 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 NAT 통계를 확인합니다.

```
# ipnat -s
```

### 예 21-18 IP 필터에 대한 NAT 통계 보기

다음 예에서는 NAT 통계 확인 방법을 보여 줍니다.

```
# ipnat -s
mapped in      0      out      0
added  0      expired 0
no memory      0      bad nat 0
inuse  0
rules  1
wilds  0
```

## ▼ IP 필터에 대한 주소 풀 통계 확인 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 주소 풀 통계를 확인합니다.

```
# ippool -s
```

### 예 21-19 IP 필터에 대한 주소 풀 통계 보기

다음 예에서는 주소 풀 통계 확인 방법을 보여 줍니다.

```
# ippool -s
Pools:  3
Hash Tables:  0
Nodes:  0
```

# IP 필터 로그 파일 작업

표 21-5 IP 필터 로그 파일 작업(작업 맵)

작업	설명	수행 방법
로그 파일을 만듭니다.	별도의 IP 필터 로그 파일을 만듭니다.	322 페이지 “IP 필터 로그 파일 설정 방법”
로그 파일을 확인합니다.	ipmon 명령을 사용하여 상태, NAT 및 일반 로그 파일을 확인합니다.	323 페이지 “IP 필터 로그 파일 확인 방법”
패킷 로그 버퍼를 비웁니다.	ipmon -F 명령을 사용하여 패킷 로그 버퍼의 콘텐츠를 제거합니다.	324 페이지 “패킷 로그 파일을 비우는 방법”
기록된 패킷을 파일에 저장합니다.	나중에 참조할 수 있도록 기록된 패킷을 파일에 저장합니다.	325 페이지 “기록된 패킷을 파일에 저장하는 방법”

## ▼ IP 필터 로그 파일 설정 방법

기본적으로 IP 필터에 대한 모든 로그 정보는 syslogd 파일에 기록됩니다. 기본 로그 파일에 기록될 수 있는 다른 데이터와 별도로 IP 필터 트래픽 정보가 기록되도록 로그 파일을 설정해야 합니다. 다음 단계를 수행하십시오.

### 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스](#)의 “RBAC 초기 구성(작업 맵)”을 참조하십시오.

### 2 다음 두 행을 추가하여 /etc/syslog.conf 파일을 편집합니다.

```
# Save IP Filter log output to its own file
local0.debug          /var/log/log-name
```

주 - 두번째 행에서 스페이스바가 아닌 Tab 키를 사용하여 local0.debug와 /var/log/log-name을 구분해야 합니다.

### 3 새 로그 파일을 만듭니다.

```
# touch /var/log/log-name
```

### 4 system-log 서비스를 다시 시작합니다.

```
# svcadm restart system-log
```

## 예 21-20 IP 필터 로그 만들기

다음 예에서는 IP 필터 정보를 아카이브할 `ipmon.log`를 만드는 방법을 보여 줍니다.

`/etc/syslog.conf`에서 다음을 입력합니다.

```
# Save IP Filter log output to its own file
local0.debug          /var/log/ipmon.log
```

명령줄에서 다음을 입력합니다.

```
# touch /var/log/ipmon.log
# svcadm restart system-log
```

## ▼ IP 필터 로그 파일 확인 방법

시작하기 전에 IP 필터 데이터를 기록할 별도의 로그 파일을 만들어야 합니다. 322 페이지 “IP 필터 로그 파일 설정 방법”을 참조하십시오.

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

- 2 상태, NAT 또는 일반 로그 파일을 확인합니다. 로그 파일을 보려면 적합한 옵션을 사용하여 다음 명령을 입력합니다.

```
# ipmon -o [S|N|I] filename
```

S     상태 로그 파일을 표시합니다.

N     NAT 로그 파일을 표시합니다.

I     일반 IP 로그 파일을 표시합니다.

모든 상태, NAT 및 일반 로그 파일을 보려면 옵션을 모두 사용합니다.

```
# ipmon -o SNI filename
```

- 먼저 수동으로 `ipmon` 데몬을 중지한 경우 다음 명령을 사용하여 상태, NAT 및 IP 필터 로그 파일을 표시할 수도 있습니다.

```
# ipmon -a filename
```

주 - ipmon 데몬이 아직 실행 중인 경우 `ipmon -a` 구문을 사용하지 마십시오.  
일반적으로 데몬은 시스템 부트 시 자동으로 시작됩니다. `ipmon -a` 명령을 실행하면  
`ipmon`의 다른 복사본이 열립니다. 이 경우 두 복사본은 동일한 로그 정보를 읽고  
하나의 복사본만 특정 로그 메시지를 가져옵니다.

로그 파일 확인에 대한 자세한 내용은 `ipmon(1M)` 매뉴얼 페이지를 참조하십시오.

## 예 21-21 IP 필터 로그 파일 보기

다음 예에서는 `/var/ipmon.log`의 출력을 보여 줍니다.

```
# ipmon -o SNI /var/ipmon.log
02/09/2004 15:27:20.606626 bge0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

또는

```
# pkill ipmon
# ipmon -aD /var/ipmon.log
02/09/2004 15:27:20.606626 bge0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

## ▼ 패킷 로그 파일을 비우는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어  
사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스](#)의 “RBAC 초기 구성(작업 맵)”을  
참조하십시오.

- 2 패킷 로그 버퍼를 비웁니다.

```
# ipmon -F
```

## 예 21-22 패킷 로그 파일 비우기

다음 예에서는 로그 파일 제거 시 출력을 보여 줍니다. 시스템에서는 이 예에서와 같이  
로그 파일에 저장된 항목이 없는 경우에도 보고서를 제공합니다.

```
# ipmon -F
0 bytes flushed from log buffer
0 bytes flushed from log buffer
0 bytes flushed from log buffer
```

## ▼ 기록된 패킷을 파일에 저장하는 방법

- 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.

- 2 기록된 패킷을 파일에 저장합니다.

```
# cat /dev/ipl > filename
```

명령줄 프롬프트를 다시 가져올 Ctrl-C를 입력하여 프로시저를 중단할 때까지 *filename* 파일에 패킷이 계속 기록됩니다.

### 예 21-23 기록된 패킷을 파일에 저장

다음 예에서는 기록된 패킷을 파일에 저장한 후의 결과를 보여 줍니다.

```
# cat /dev/ipl > /tmp/logfile
^C#
```

```
# ipmon -f /tmp/logfile
02/09/2004 15:30:28.708294 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 52 -S IN
02/09/2004 15:30:28.708708 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.792611 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 70 -AP IN
02/09/2004 15:30:28.872000 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872142 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 43 -AP IN
02/09/2004 15:30:28.872808 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872951 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 47 -AP IN
02/09/2004 15:30:28.926792 bge0 @0:1 p 129.146.157.149,33923 ->
    129.146.157.145,23 PR tcp len 20 40 -A IN
.
.
(output truncated)
```

## IP 필터 구성 파일 만들기 및 편집

규칙 세트 및 주소 풀을 만들고 수정하려면 구성 파일을 직접 편집해야 합니다. 구성 파일은 표준 UNIX 구문 규칙을 따릅니다.

- 파운드 기호(#)는 행에 주석이 포함되어 있음을 나타냅니다.
- 규칙과 주석은 동일한 행에 함께 사용될 수 있습니다.
- 규칙을 쉽게 읽을 수 있도록 임의로 공백을 사용할 수 있습니다.
- 규칙의 길이는 두 행 이상일 수 있습니다. 행 끝에 백슬래시(\)를 사용하여 규칙이 다음 행에서 계속됨을 나타낼 수 있습니다.

### ▼ IP 필터에 대한 구성 파일을 만드는 방법

이 절차에서는 다음을 설정하는 방법에 대해 설명합니다.

- 패킷 필터링 구성 파일
- NAT 규칙 구성 파일
- 주소 풀 구성 파일

#### 1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.

#### 2 선택한 파일 편집기를 시작합니다. 구성할 기능에 대한 구성 파일을 만들거나 편집합니다.

- 패킷 필터링 규칙에 대한 구성 파일을 만들려면 ipf.conf 파일을 편집합니다.

IP 필터는 ipf.conf 파일에 배치된 패킷 필터링 규칙을 사용합니다. 패킷 필터링에 대한 규칙 파일을 /etc/ipf/ipf.conf 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 필터링 규칙이 로드되지 않도록 하려면 선택한 파일에 배치합니다. 그런 다음 309 페이지 “다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법”에 설명된 대로 ipf 명령을 사용하여 규칙을 활성화할 수 있습니다.

패킷 필터링 규칙을 만드는 방법은 292 페이지 “IP 필터의 패킷 필터링 기능 사용”을 참조하십시오.

주 - ipf.conf 파일이 비어 있을 경우 필터링이 없는 것입니다. ipf.conf 파일이 비어 있을 경우 다음을 읽는 규칙 세트가 있는 것과 동일합니다.

```
pass in all
pass out all
```

- NAT 규칙에 대한 구성 파일을 만들려면 ipnat.conf 파일을 편집합니다.  
IP 필터는 ipnat.conf 파일에 배치된 NAT 규칙을 사용합니다. NAT에 대한 규칙 파일을 /etc/ipf/ipnat.conf 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 ipnat.conf 파일을 배치합니다. 그런 다음 ipnat 명령을 사용하여 NAT 규칙을 활성화할 수 있습니다.  
NAT에 대한 규칙을 만드는 방법은 295 페이지 “IP 필터의 NAT 기능 사용”을 참조하십시오.
- 주소 풀에 대한 구성 파일을 만들려면 ippool.conf 파일을 편집합니다.  
IP 필터는 ippool.conf 파일에 배치된 주소 풀을 사용합니다. 주소 풀에 대한 규칙 파일을 /etc/ipf/ippool.conf 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 주소 풀이 로드되지 않도록 하려면 선택한 위치에 ippool.conf 파일을 배치합니다. 그런 다음 ippool 명령을 사용하여 주소 풀을 활성화할 수 있습니다.  
주소 풀을 만드는 방법은 296 페이지 “IP 필터의 주소 풀 기능 사용”을 참조하십시오.

## IP 필터 구성 파일 예

다음 예에서는 필터링 구성에 사용되는 패킷 필터링 규칙의 실례를 제공합니다.

### 예 21-24 IP 필터 호스트 구성

이 예에서는 bge 네트워크 인터페이스가 있는 호스트 시스템에 대한 구성을 보여 줍니다.

```
# pass and log everything by default
pass in log on bge0 all
pass out log on bge0 all

# block, but don't log, incoming packets from other reserved addresses
block in quick on bge0 from 10.0.0.0/8 to any
block in quick on bge0 from 172.16.0.0/12 to any

# block and log untrusted internal IPs. 0/32 is notation that replaces
# address of the machine running Solaris IP Filter.
block in log quick from 192.168.1.15 to <thishost>
block in log quick from 192.168.1.43 to <thishost>
```

## 예 21-24 IP 필터 호스트 구성 (계속)

```
# block and log X11 (port 6000) and remote procedure call
# and portmapper (port 111) attempts
block in log quick on bge0 proto tcp from any to bge0/32 port = 6000 keep state
block in log quick on bge0 proto tcp/udp from any to bge0/32 port = 111 keep state
```

이 규칙 세트는 bge 인터페이스에서 모든 항목을 주고 받을 수 있도록 허용하는 제한되지 않은 두 개의 규칙으로 시작합니다. 두번째 규칙 세트는 개인 주소 공간 10.0.0.0 및 172.16.0.0의 수신 패킷이 방화벽에 들어오지 못하도록 차단합니다. 다음 규칙 세트는 호스트 시스템의 특정 내부 주소를 차단합니다. 마지막 규칙 세트는 포트 6000 및 포트 111에서 수신되는 패킷을 차단합니다.

## 예 21-25 IP 필터 서버 구성

이 예에서는 웹 서버로 사용되는 호스트 시스템에 대한 구성을 보여 줍니다. 이 시스템에는 e1000g 네트워크 인터페이스가 있습니다.

```
# web server with an e1000g interface
# block and log everything by default;
# then allow specific services
# group 100 - inbound rules
# group 200 - outbound rules
# (0/32) resolves to our IP address)
*** FTP proxy ***
```

```
# block short packets which are packets
# fragmented too short to be real.
block in log quick all with short
```

```
# block and log inbound and outbound by default,
# group by destination
block in log on e1000g0 from any to any head 100
block out log on e1000g0 from any to any head 200
```

```
# web rules that get hit most often
pass in quick on e1000g0 proto tcp from any \
to e1000g0/32 port = http flags S keep state group 100
pass in quick on e1000g0 proto tcp from any \
to e1000g0/32 port = https flags S keep state group 100
```

```
# inbound traffic - ssh, auth
pass in quick on e1000g0 proto tcp from any \
to e1000g0/32 port = 22 flags S keep state group 100
pass in log quick on e1000g0 proto tcp from any \
to e1000g0/32 port = 113 flags S keep state group 100
pass in log quick on e1000g0 proto tcp from any port = 113 \
to e1000g0/32 flags S keep state group 100
```



## 예 21-25 IP 필터 서버 구성 (계속)

```
# outbound traffic - DNS, auth, NTP, ssh, www, smtp
pass out quick on e1000g0 proto tcp/udp from e1000g0/32 \
to any port = domain flags S keep state group 200
pass in quick on e1000g0 proto udp from any \
port = domain to e1000g0/32 group 100

pass out quick on e1000g0 proto tcp from e1000g0/32 \
to any port = 113 flags S keep state group 200
pass out quick on e1000g0 proto tcp from e1000g0/32 port = 113 \
to any flags S keep state group 200

pass out quick on e1000g0 proto udp from e1000g0/32 to any \
port = ntp group 200
pass in quick on e1000g0 proto udp from any \
port = ntp to e1000g0/32 port = ntp group 100

pass out quick on e1000g0 proto tcp from e1000g0/32 \
to any port = ssh flags S keep state group 200

pass out quick on e1000g0 proto tcp from e1000g0/32 \
to any port = http flags S keep state group 200
pass out quick on e1000g0 proto tcp from e1000g0/32 \
to any port = https flags S keep state group 200

pass out quick on e1000g0 proto tcp from e1000g0/32 \
to any port = smtp flags S keep state group 200

# pass icmp packets in and out
pass in quick on e1000g0 proto icmp from any to e1000g0/32 keep state group 100
pass out quick on e1000g0 proto icmp from e1000g0/32 to any keep state group 200

# block and ignore NETBIOS packets
block in quick on e1000g0 proto tcp from any \
to any port = 135 flags S keep state group 100

block in quick on e1000g0 proto tcp from any port = 137 \
to any flags S keep state group 100
block in quick on e1000g0 proto udp from any to any port = 137 group 100
block in quick on e1000g0 proto udp from any port = 137 to any group 100

block in quick on e1000g0 proto tcp from any port = 138 \
to any flags S keep state group 100
block in quick on e1000g0 proto udp from any port = 138 to any group 100

block in quick on e1000g0 proto tcp from any port = 139 to any flags S keep state
group 100
block in quick on e1000g0 proto udp from any port = 139 to any group 100
```

## 예 21-26 IP 필터 라우터 구성

이 예에서는 내부 인터페이스 nge 및 외부 인터페이스 ce1이 있는 라우터에 대한 구성을 보여 줍니다.

## 예 21-26 IP 필터 라우터 구성 (계속)

```
# internal interface is nge0 at 192.168.1.1
# external interface is nge1 IP obtained via DHCP
# block all packets and allow specific services
*** NAT ***
*** POOLS ***

# Short packets which are fragmented too short to be real.
block in log quick all with short

# By default, block and log everything.
block in log on nge0 all
block in log on nge1 all
block out log on nge0 all
block out log on nge1 all

# Packets going in/out of network interfaces that aren't on the loopback
# interface should not exist.
block in log quick on nge0 from 127.0.0.0/8 to any
block in log quick on nge0 from any to 127.0.0.0/8
block in log quick on nge1 from 127.0.0.0/8 to any
block in log quick on nge1 from any to 127.0.0.0/8

# Deny reserved addresses.
block in quick on nge1 from 10.0.0.0/8 to any
block in quick on nge1 from 172.16.0.0/12 to any
block in log quick on nge1 from 192.168.1.0/24 to any
block in quick on nge1 from 192.168.0.0/16 to any

# Allow internal traffic
pass in quick on nge0 from 192.168.1.0/24 to 192.168.1.0/24
pass out quick on nge0 from 192.168.1.0/24 to 192.168.1.0/24

# Allow outgoing DNS requests from our servers on .1, .2, and .3
pass out quick on nge1 proto tcp/udp from nge1/32 to any port = domain keep state
pass in quick on nge0 proto tcp/udp from 192.168.1.2 to any port = domain keep state
pass in quick on nge0 proto tcp/udp from 192.168.1.3 to any port = domain keep state

# Allow NTP from any internal hosts to any external NTP server.
pass in quick on nge0 proto udp from 192.168.1.0/24 to any port = 123 keep state
pass out quick on nge1 proto udp from any to any port = 123 keep state

# Allow incoming mail
pass in quick on nge1 proto tcp from any to nge1/32 port = smtp keep state
pass in quick on nge1 proto tcp from any to nge1/32 port = smtp keep state
pass out quick on nge1 proto tcp from 192.168.1.0/24 to any port = smtp keep state
```

## 예 21-26 IP 필터 라우터 구성 (계속)

```
# Allow outgoing connections: SSH, WWW, NNTP, mail, whois
pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = 22 keep state
pass out quick on nge1 proto tcp from 192.168.1.0/24 to any port = 22 keep state

pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass out quick on nge1 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = 443 keep state
pass out quick on nge1 proto tcp from 192.168.1.0/24 to any port = 443 keep state

pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = nntp keep state
block in quick on nge1 proto tcp from any to any port = nntp keep state
pass out quick on nge1 proto tcp from 192.168.1.0/24 to any port = nntp keep state

pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = smtp keep state

pass in quick on nge0 proto tcp from 192.168.1.0/24 to any port = whois keep state
pass out quick on nge1 proto tcp from any to any port = whois keep state

# Allow ssh from offsite
pass in quick on nge1 proto tcp from any to nge1/32 port = 22 keep state

# Allow ping out
pass in quick on nge0 proto icmp all keep state
pass out quick on nge1 proto icmp all keep state

# allow auth out
pass out quick on nge1 proto tcp from nge1/32 to any port = 113 keep state
pass out quick on nge1 proto tcp from nge1/32 port = 113 to any keep state

# return rst for incoming auth
block return-rst in quick on nge1 proto tcp from any to any port = 113 flags S/SA

# log and return reset for any TCP packets with S/SA
block return-rst in log on nge1 proto tcp from any to any flags S/SA

# return ICMP error packets for invalid UDP packets
block return-icmp(net-unr) in proto udp all
```



## 제 4 부

# 네트워크 성능

이 부분에서는 통합 로드 균형 조정 및 가상 라우터 중복성 프로토콜 등 네트워크 성능 기능에 대해 설명합니다.



## 통합된 로드 밸런서 개요

---

통합된 로드 밸런서(ILB)는 Oracle Solaris의 기능으로, SPARC 및 x86 기반 시스템에 설치된 Oracle Solaris에 대한 계층 3 및 계층 4 로드 균형 조정 기능을 제공합니다. ILB는 클라이언트의 수신 요청을 가로채서 로드 균형 조정 규칙을 기반으로 요청을 처리할 백엔드 서버를 결정한 다음 선택한 서버로 요청을 전달합니다. ILB는 선택적 건전성 검사를 수행하고 선택한 서버가 수신 요청을 처리할 수 있는지 여부를 확인할 수 있도록 로드 균형 조정 알고리즘에 대한 데이터를 제공합니다.

이 장에서는 다음 절을 다룹니다.

- 336 페이지 “ILB 용어”
- 338 페이지 “ILB의 기능”
- 342 페이지 “ILB 프로세스”
- 343 페이지 “ILB 사용 지침”
- 343 페이지 “ILB 및 서비스 관리 기능”
- 344 페이지 “ILB 명령 및 하위 명령”

ILB의 주요 기능은 다음과 같습니다.

- IPv4와 IPv6에 대한 Stateless DSR(Direct Server Return) 및 NAT(Network Address Translation) 작동 모드를 지원합니다.
- 명령줄 인터페이스(CLI)를 통한 ILB 관리를 허용합니다.
- 건전성 검사를 통한 서버 모니터링 기능을 제공합니다.

ILB의 세 가지 주요 구성 요소는 다음과 같습니다.

- `ilbadm` CLI - 이 인터페이스를 사용하여 로드 균형 조정 규칙을 구성하고, 선택적 건전성 검사를 수행하고, 통계를 볼 수 있습니다.
- `libilb` 구성 라이브러리 - `ilbadm` 및 기타 타사 응용 프로그램이 ILB 관리를 위해 `libilb`에서 구현된 기능을 사용할 수 있습니다.
- `ilbd` 데몬 - 이 데몬은 다음 작업을 수행합니다.
  - 지속 구성을 관리합니다.

- 구성 정보를 처리하여 실행을 위해 ILB 커널 모듈로 보내 ILB 커널 모듈에 대한 순차적 액세스를 제공합니다.
- 로드 분배가 제대로 조정되도록 건전성 검사를 수행하고 ILB 커널에 결과를 통지합니다.

## ILB 용어

이 절에서는 시스템에서 ILB를 구현할 때 알아두어야 할 몇 가지 유용한 용어에 대해 설명합니다.

### 연결 드레이닝

관리 용도로 사용 안함으로 설정된 서버에 새 연결이 설정되지 않도록 하는 기능을 제공하는 방식입니다. 이 기능은 활성 연결 또는 세션에 장애를 일으키지 않고 서버를 종료하는 데 유용합니다. 서버에 대한 기존 연결은 정상적으로 작동합니다. 서버가 요청을 처리할 준비가 되면 관리 용도로 다시 서버를 사용으로 설정할 수 있으며 로드 밸런서가 서버로 새 연결을 전달합니다. ILB는 NAT 기반 가상 서비스를 사용하는 서버에만 이 기능을 제공합니다.

### DSR (Direct Server Return) 모드

백엔드 서버에 대한 수신 요청의 로드 균형을 조정하고 클라이언트로 직접 요청을 보내 서버의 반환 트래픽이 로드 밸런서를 무시할 수 있도록 합니다. DSR의 현재 ILB 구현에서는 TCP 연결 추적을 제공하지 않습니다(Stateless임을 의미함).

이점:

- 패킷의 대상 MAC 주소가 변경되고 서버가 직접 클라이언트에 응답하므로 NAT보다 성능이 뛰어납니다.
- 완전한 투명 효과: 서버가 클라이언트 IP 주소에서 직접 연결을 확인하고 기본 게이트웨이를 통해 클라이언트에 응답합니다.

단점:

- 백엔드 서버가 고유의 IP 주소(건전성 검사용)와 가상 IP 주소(로드 균형이 조정된 트래픽용)에 모두 응답해야 합니다.
- 로드 밸런서가 연결 상태를 유지 관리하지 않으므로(Stateless임을 의미함) 서버를 추가하거나 제거하면 연결 장애가 발생합니다.

**로드 균형 조정 알고리즘** ILB가 수신 요청을 위해 서버 그룹에서 백엔드 서버를 선택할 때 사용하는 알고리즘입니다.



로드 균형 조정 규칙	ILB에서 가상 서버는 로드 균형 조정 규칙으로 표시되며 다음 매개변수로 정의됩니다. <ul style="list-style-type: none"> <li>■ 가상 IP 주소</li> <li>■ 전송 프로토콜: TCP 또는 UDP</li> <li>■ 포트 번호(또는 포트 범위)</li> <li>■ 로드 균형 조정 알고리즘</li> <li>■ 로드 균형 조정 모드의 유형(DSR, Full-NAT 또는 Half-NAT)</li> <li>■ 일련의 백엔드 서버로 구성된 서버 그룹</li> <li>■ 서버 그룹의 각 서버에 대해 실행할 수 있는 선택적 서버 건전성 검사</li> <li>■ 건전성 검사에 사용할 선택적 포트</li> </ul>
-------------	--

주 - 특정 포트 또는 `ilbd` 데몬이 서버 포트 범위에서 임의적으로 선택하는 포트에 대해 지정할 수 있습니다.

NAT 기반 로드 균형 조정	<ul style="list-style-type: none"> <li>■ 가상 서비스를 표시할 규칙 이름</li> </ul> <p>IP 헤더 정보 재작성을 포함하며 요청 및 응답 트래픽을 모두 처리합니다. NAT의 두 가지 유형은 Half-NAT와 Full-NAT입니다. 두 유형 모두 대상 IP 주소를 재작성합니다. 단, Full-NAT는 소스 IP 주소도 재작성하여 서버에 모든 연결이 로드 밸런서에서 시작되는 것으로 표시되도록 합니다. NAT는 TCP 연결 추적을 제공합니다(Stateful임을 의미함).</p> <p>이점:</p> <ul style="list-style-type: none"> <li>■ 기본 게이트웨이가 로드 밸런서를 가리키도록 변경하여 모든 백엔드 서버와 작동합니다.</li> <li>■ 로드 밸런서가 연결 상태를 유지 관리하므로 연결 장애 없이 서버를 추가하거나 제거할 수 있습니다.</li> </ul> <p>단점:</p> <ul style="list-style-type: none"> <li>■ 처리 시 IP 헤더가 조작되고 서버가 로드 밸런서로 응답을 보내므로 DSR에 비해 성능이 떨어집니다.</li> <li>■ 모든 백엔드 서버가 로드 밸런서를 기본 게이트웨이로 사용해야 합니다.</li> </ul>
-----------------	---

지속 구성	ILB 컨텍스트에서 지속 구성은 재부트 및 패키지 업데이트 시 지속되는 구성(일련의 로드 균형 조정 규칙)입니다.
프록시 소스	프록시로 사용될 수 있는 IP 주소 범위입니다. 범위는 열 개의 IP 주소로 제한됩니다. 전체 NAT 구현이 있는 경우에만 프록시 소스가 필요합니다.
세션	특정 기간 동안 동일한 클라이언트에서 오며 전체적으로 특정 의미를 가질 수 있는 여러 패킷으로 구성됩니다.

## 세션 지속성

클라이언트의 모든 패킷이 동일한 백엔드 서버로 전송될 수 있도록 합니다. 고착성이라고도 합니다. `pmask=prefix length` 및 `persist-timeout=value in seconds` 옵션을 지정하여 가상 서비스에 대해 간단한 세션 지속성(즉, 소스 주소 지속성)을 설정할 수 있습니다. 클라이언트와 서버 간에 세션 지속성이 설정되면 지속성이 존재하는 한 클라이언트에서 가상 서비스로의 모든 패킷이 동일한 백엔드 서버로 전달됩니다. CIDR 표기법의 접두어 길이는 0-32의 값(IPv4의 경우) 및 0-128의 값(IPv6의 경우)입니다.

## 서버 그룹

0개 이상의 백엔드 서버로 구성되며, 가상 서비스에 사용되는 경우 하나 이상의 서버를 포함해야 합니다. 예를 들어, HTTP 요청의 로드 균형을 조정하려면 하나 이상의 백엔드 서버로 구성된 서버 그룹으로 ILB를 구성해야 합니다. ILB는 구성된 일련의 서버 간에 HTTP 트래픽의 로드 균형을 조정합니다.

## 서버 ID

서버가 서버 그룹에 추가될 때 시스템에서 지정한 IP 주소의 고유한 이름입니다.

## 가상 IP 주소 (VIP)

가상 서비스의 IP 주소입니다.

## 가상 서비스

클라이언트가 `VIP:port`로 확인하는 서비스입니다. 예를 들어, `www.foo.com:80`입니다. 잠재적으로 두 개 이상의 서버로 구성된 서버 그룹이 서비스를 처리하고 있더라도 서버 그룹은 가상 서비스의 클라이언트에 단일 `IP address:port`로 표시됩니다. 단일 서버는 두 개 이상의 서버 그룹에 포함될 수 있으므로 다중 가상 서비스를 제공할 수 있습니다. 또한 단일 서버 그룹은 다중 가상 서비스를 제공할 수 있습니다.

## ILB의 기능

이 절에서는 ILB의 주요 기능에 대해 설명합니다.

### ILB 작동 모드

ILB는 단일 각 및 이중 각 토폴로지에서 IPv4와 IPv6에 대해 Stateless DSR 및 NAT 작동 모드를 지원합니다.

- **Stateless DSR 모드** - DSR 모드에서 ILB는 백엔드 서버에 대한 수신 요청의 로드 균형을 조정하지만 서버에서 클라이언트로의 반환 트래픽이 로드 균형 조정을 무시할 수 있도록 합니다. 단, 백엔드 서버에 대한 라우터로 사용되도록 ILB를 설정할 수도 있습니다. 이 경우 백엔드 서버에서 클라이언트로의 응답은 ILB를 실행 중인 시스템을 통해 경로가 지정됩니다. Stateless DSR을 사용할 경우 ILB는 기본 통계를 제외하고 처리된 패킷에 대한 상태 정보를 저장하지 않습니다. ILB가 이 모드에서 상태를 저장하지 않으므로 성능은 일반적인 IP 전달 성능과 거의 유사합니다. 이 모드는 연결 없는 프로토콜에 가장 적합합니다.
- **NAT 모드(Full-NAT 및 Half-NAT)** - ILB가 로드 균형 조정 기능에 독립형 모드로 NAT를 엄격히 사용합니다. 이 모드에서 ILB는 헤더 정보를 재작성하고 수신 및 송신 트래픽을 처리합니다. NAT 모드는 추가 보안을 제공하므로 HTTP(또는 SSL) 트래픽에 가장 적합합니다.

---

주 - ILB에서 구현된 NAT 코드 경로는 Oracle Solaris의 IP 필터 기능에서 구현된 코드 경로와 다릅니다. 이러한 코드 경로를 동시에 사용하지 **마십시오**.

---

## ILB 알고리즘

ILB 알고리즘은 트래픽 분배를 제어하고 다양한 로드 분배 및 서버 선택 특성을 제공합니다. ILB는 두 가지 작동 모드에 대해 다음 알고리즘을 제공합니다.

- 라운드 로빈 - 라운드 로빈 알고리즘에서 로드 밸런서는 회전을 기준으로 서버 목록에 요청을 지정합니다. 서버에 요청이 지정되면 서버는 목록 끝으로 이동됩니다.
- *src IP* 해시 - 소스 IP 해시 메소드에서 로드 밸런서는 수신 요청의 소스 IP 주소에 대한 해시 값을 기반으로 서버를 선택합니다.
- *src-IP, port* 해시 - 소스 IP, 포트 해시 메소드에서 로드 밸런서는 수신 요청의 소스 IP 주소 및 소스 포트에 대한 해시 값을 기반으로 서버를 선택합니다.
- *src-IP, VIP* 해시 - 소스 IP, VIP 해시 메소드에서 로드 밸런서는 수신 요청의 소스 IP 주소 및 대상 IP 주소에 대한 해시 값을 기반으로 서버를 선택합니다.

## ILB 명령줄 인터페이스

CLI는 `/usr/sbin/ilbadm` 디렉토리에 있습니다. 로드 균형 조정 규칙, 서버 그룹 및 건전성 검사를 구성할 하위 명령이 포함되어 있습니다. 또한 통계를 표시하고 구성 세부 정보를 확인할 하위 명령이 포함되어 있습니다. 하위 명령은 다음과 같은 두 가지 범주로 구분될 수 있습니다.

- 구성 하위 명령 - 이러한 하위 명령을 통해 다음 작업을 수행할 수 있습니다.
  - 로드 균형 조정 규칙 만들기 및 삭제
  - 로드 균형 조정 규칙 사용 및 사용 안함으로 설정
  - 서버 그룹 만들기 및 삭제
  - 서버를 서버 그룹에 추가 및 서버 그룹에서 제거
  - 백엔드 서버를 사용 및 사용 안함으로 설정
  - 로드 균형 조정 규칙 내 서버 그룹에 대한 서버 건전성 검사 만들기 및 삭제

---

주 - 구성 하위 명령을 관리하려면 권한이 필요합니다. 이러한 권한은 역할 기반 액세스 제어(RBAC)를 통해 얻을 수 있습니다. 적합한 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성(작업 맵)”**을 참조하십시오.

---

- 보기 하위 명령 - 이러한 하위 명령을 통해 다음 작업을 수행할 수 있습니다.
  - 구성된 로드 균형 조정 규칙, 서버 그룹 및 건전성 검사 보기

- 패킷 전달 통계 보기
- NAT 연결 테이블 보기
- 건전성 검사 결과 보기
- 세션 지속성 매핑 테이블 보기

---

주 - 보기 하위 명령을 관리하는 데는 권한이 필요하지 않습니다.

---

ilbadm 하위 명령 목록은 344 페이지 “ILB 명령 및 하위 명령”을 참조하십시오. ilbadm 하위 명령에 대한 자세한 내용은 [ilbadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## ILB 서버 모니터링 기능

ILB는 다음 기능과 함께 서버 건전성 검사를 제공할 수 있는 선택적 서버 모니터링 기능을 제공합니다.

- 내장 ping 검사
- 내장 TCP 검사
- 내장 UDP 검사
- 서버 건전성 검사로 실행될 수 있는 사용자 제공 테스트

기본적으로 ILB는 건전성 검사를 수행하지 않습니다. 로드 균형 조정 규칙을 만들 때 각 서버 그룹에 대한 건전성 검사를 지정할 수 있습니다. 로드 균형 조정 규칙당 하나의 건전성 검사만 구성할 수 있습니다. 가상 서비스가 사용으로 설정된 동안, 사용으로 설정된 가상 서비스와 연관된 서버 그룹에 건전성 검사가 자동으로 시작되고 정기적으로 반복합니다. 가상 서비스를 사용 안함으로 설정하면 즉시 건전성 검사가 중지됩니다. 가상 서비스를 다시 사용으로 설정할 때 이전 건전성 검사 상태는 보존되지 않습니다.

건전성 검사를 실행하기 위해 TCP, UDP 또는 사용자 정의 테스트 검사를 지정할 때 ILB는 기본적으로 ping 검사를 보내어 지정된 TCP, UDP 또는 사용자 정의 테스트 검사를 서버로 보내기 전에 서버에 연결할 수 있는지 확인합니다. ping 검사는 서버 건전성을 모니터링하는 방법입니다. ping 검사를 실패하면 해당 서버가 건전성 검사 상태 **unreachable**와 함께 사용 안함으로 설정됩니다. ping 검사를 성공하지만 TCP, UDP 또는 사용자 정의 테스트 검사를 실패하면 건전성 검사 상태 **dead**와 함께 서버가 사용 안함으로 설정됩니다.

---

주 -

- 기본 ping 검사를 사용 안함으로 설정할 수 있습니다.
  - UDP 검사에 대해서는 기본 ping 검사를 사용 안함으로 설정할 수 없습니다. 따라서 UDP 건전성 검사의 경우 ping 검사는 항상 기본 검사입니다.
- 

다음 표의 매개변수로 건전성 검사를 구성할 수 있습니다.

표 22-1 건전성 검사 매개변수 구성

건전성 검사 매개변수	설명
hc-test	수행할 건전성 검사의 유형을 지정합니다.
hc-timeout	건전성 검사가 완료되지 않은 경우 시간 초과를 시작합니다.
hc-interval	연속되는 건전성 검사의 간격을 지정합니다.  주 - 간격은 $0.5 * hc\_interval$ 값과 $1.5 * hc\_interval$ 값 사이에서 임의로 지정됩니다.
hc-count	서버를 결함이 있는 것으로 간주하기 전까지 몇 번의 연속된 검사 실패를 허용할지 해당 횟수를 지정합니다.

## 추가 ILB 기능

이 절에서는 ILB의 추가 기능에 대해 설명합니다.

- **클라이언트가 가상 IP(VIP) 주소를 ping할 수 있도록 함** - ILB는 클라이언트가 보낸 VIP에 대한 ICMP(Internet Control Message Protocol) 에코 요청에 응답할 수 있습니다. ILB는 DSR 및 NAT 작동 모드에 이 기능을 제공합니다.
- **서비스 인터럽트 없이 서버 그룹에서 서버를 추가 및 제거할 수 있도록 함** - 기존에 설정된 백엔드 서버와의 연결에 대한 인터럽트 없이 동적으로 서버 그룹에서 서버를 추가 및 제거할 수 있습니다. ILB는 NAT 작동 모드에 이 기능을 제공합니다.
- **세션 지속성(고착성)을 구성할 수 있도록 함** - 동일한 클라이언트로부터 온 일련의 연결과 패킷 중 하나 또는 두 가지가 모두 동일한 백엔드 서버로 전송되어야 하는 응용 프로그램이 많습니다. `create-rule[{-m persist=<netmask>}]` 하위 명령에서 넷마스크를 지정하여 가상 서비스에 대한 세션 지속성을 구성할 수 있습니다. 지속 매핑이 만들어지면 클라이언트의 소스 IP 주소와 일치하는 가상 서비스로의 연결과 패킷 중 하나에 대한 후속 요청이 동일한 백엔드 서버로 전달됩니다. 세션 지속성 방식에 대한 지원은 DSR 및 NAT 작동 모드에서 모두 제공됩니다.
- **연결 드레이닝을 수행할 수 있도록 함** - ILB는 NAT 기반 가상 서비스의 서버에 대해서만 이 기능을 지원합니다. 이 기능은 새 연결이 사용 안함으로 설정된 서버로 전송되지 않도록 합니다. 서버에 대한 기존 연결은 계속 작동합니다. 해당 서버에 대한 모든 연결이 종료되면 유지 관리 용도로 서버를 종료할 수 있습니다. 서버가 요청을 처리할 준비가 되면 로드 밸런서가 새 연결을 전달할 수 있도록 서버를 사용으로 설정하십시오. 이 기능을 사용하면 활성 연결 또는 세션에 장애를 일으키지 않고 유지 관리 용도로 서버를 종료할 수 있습니다.
- **TCP 및 UDP 포트의 로드 균형을 조정할 수 있도록 함** - ILB는 각 포트에 대한 명시적 규칙이 설정되지 않은 경우에도 다양한 일련의 서버 간에 지정된 IP 주소의 모든 포트에 대한 로드 균형을 조정할 수 있습니다. ILB는 DSR 및 NAT 작동 모드에 이 기능을 제공합니다.

- **동일한 서버 그룹 내 가상 서비스에 대해 별도의 포트를 지정할 수 있도록 함** - 이 기능을 사용하면 ILB에서 NAT 작동 모드에 대해 동일한 서버 그룹 내 다양한 서버의 다른 대상 포트를 지정할 수 있습니다.
- **간단한 포트 범위의 로드 균형을 조정할 수 있도록 함** - ILB는 지정된 서버 그룹 간에 VIP의 포트 범위에 대한 로드 균형을 조정할 수 있습니다. 다양한 일련의 백엔드 서버 간에 동일한 VIP의 다른 포트 범위에 대한 로드 균형을 조정하여 편리하게 IP 주소를 절약할 수 있습니다. 또한 NAT 모드에 대한 세션 지속성이 설정된 경우 ILB는 범위 내 여러 포트에 대한 동일한 클라이언트 IP 주소에서 온 요청을 동일한 백엔드 서버로 보냅니다.
- **포트 범위를 이동 및 축소할 수 있도록 함** - 포트 범위 이동 및 축소는 로드 균형 조정 규칙의 서버 포트 범위에 따라 결정됩니다. 따라서 서버 포트 범위가 VIP 포트 범위와 다른 경우 자동으로 포트 이동이 구현됩니다. 포트 축소는 서버 포트 범위가 단일 포트인 경우 구현됩니다. 이러한 기능은 NAT 작동 모드에 제공됩니다.

## ILB 프로세스

이 절에서는 클라이언트-서버 패킷 처리, 서버-클라이언트 패킷 처리 등의 ILB 프로세스의 작동에 대해 설명합니다.

### 클라이언트-서버 패킷 처리:

1. ILB에서 클라이언트가 VIP 주소로 보낸 수신 요청을 받아 로드 균형 조정 규칙과 일치시킵니다.
2. ILB에서 일치하는 로드 균형 조정 규칙을 찾을 경우 로드 균형 조정 알고리즘을 사용하여 작동 모드에 따라 요청을 백엔드 서버로 전달합니다.
  - DSR 모드에서는 ILB가 수신 요청의 MAC 헤더를 선택된 백엔드 서버의 MAC 헤더로 바꿉니다.
  - Half-NAT 모드에서는 ILB가 수신 요청의 대상 IP 주소 및 전송 프로토콜 포트를 선택된 백엔드 서버의 대상 IP 주소 및 전송 프로토콜 포트로 바꿉니다.
  - Full-NAT 모드에서는 ILB가 수신 요청의 소스 IP 주소 및 전송 프로토콜 포트 번호를 로드 균형 조정 규칙의 NAT 소스 주소로 바꿉니다. 또한 ILB가 수신 요청의 대상 IP 주소 및 전송 프로토콜 포트를 선택된 백엔드 서버의 대상 IP 주소 및 전송 프로토콜 포트로 바꿉니다.
3. ILB가 수정된 수신 요청을 선택된 백엔드 서버로 전달합니다.

### 서버-클라이언트 패킷 처리:

1. 백엔드 서버가 클라이언트에서 온 수신 요청에 대한 응답으로 ILB로 응답을 보냅니다.
2. 백엔드 서버로부터 응답을 받은 후 ILB의 작업은 다음과 같이 작동 모드를 기반으로 합니다.

- 일반 DSR 모드에서 백엔드 서버로부터의 응답은 ILB를 무시하고 클라이언트로 직접 전송됩니다. 단, ILB가 백엔드 서버의 라우터로도 사용되는 경우 백엔드 서버에서 클라이언트로 보낸 응답은 ILB를 실행하는 시스템을 통해 경로가 지정됩니다.
- Half-NAT 모드 및 Full-NAT 모드에서 ILB는 백엔드 서버의 응답을 수신 요청과 일치시키고 변경된 IP 주소 및 전송 프로토콜 포트 번호를 원래 수신 요청의 IP 주소 및 전송 프로토콜 포트 번호로 바꿉니다. 그런 다음 ILB는 응답을 클라이언트로 전달합니다.

## ILB 사용 지침

다음 지침에서는 ILB 사용 방법에 대해 설명합니다.

- ILB를 관리하려면 ILB Management 권한 프로파일을 포함하는 역할을 맡거나 슈퍼유저가 될 수 있어야 합니다. 사용자가 만든 역할에 ILB Management 권한 프로파일을 할당할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.
- ILB 구성 명령 감사를 사용으로 설정하려면 시스템 차원의 관리 감사 클래스를 미리 선택해야 합니다. 이 작업을 수행하려면 [Oracle Solaris 관리: 보안 서비스의 “감사 서비스 구성\(작업 맵\)”](#)을 참조하십시오.
- ILB userland 구성 요소는 Oracle Solaris 저장소에서 패키지 이름이 SUNwlb로 시작하는 별도의 IPS 패키지로 전달됩니다. pkg install 명령을 사용하여 Oracle Solaris 저장소에서 해당 패키지를 다운로드해야 합니다. ILB 설치 지침은 [347 페이지 “통합 로드 밸런서 설치”](#)를 참조하십시오.
- 독립형 모드의 ILB NAT 구현은 로드 균형 조정 기능으로만 제한됩니다.
- ILB는 시스템 실패 시에만 중복을 제공하며 스위치 실패를 처리하지 않습니다. 현재 ILB는 ILB를 실행하는 여러 시스템 간에 동기화를 제공하지 않습니다.

## ILB 및 서비스 관리 기능

ILB는 SMF(서비스 관리 기능) 서비스 svc:/network/loadbalancer/ilb:default를 통해 관리됩니다. SMF 개요는 [Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리\(개요\)”](#)를 참조하십시오. SMF와 관련된 단계별 절차는 [Oracle Solaris 관리: 일반 작업의 7 장, “서비스 관리\(작업\)”](#)를 참조하십시오.



## ILB 명령 및 하위 명령

ilbadm 및 하위 명령을 사용하여 로드 균형 조정 규칙을 조작할 수 있습니다. ilbadm 하위 명령에 대한 자세한 내용은 [ilbadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

표 22-2 로드 균형 조정 규칙 조작에 사용되는 ILB 명령 및 하위 명령

ILB 명령	설명
ilbadm create-rule	지정된 특성으로 rule name을 만듭니다.
ilbadm show-rule	지정된 규칙의 특성 또는 모든 규칙(지정된 규칙이 없을 경우)을 표시합니다.
ilbadm delete-rule	rule name과 관련된 모든 정보를 제거합니다. name이 존재하지 않을 경우 이 하위 명령이 실패합니다.
ilbadm enable-rule	명명된 규칙 또는 모든 규칙(지정된 이름이 없을 경우)을 사용으로 설정합니다.
ilbadm disable-rule	명명된 규칙 또는 모든 규칙(지정된 이름이 없을 경우)을 사용 안함으로 설정합니다.
ilbadm show-statistics	통계를 표시합니다. 예를 들어, 이 하위 명령에 -t를 사용할 경우 모든 헤더와 함께 시간 기록이 포함됩니다.
ilbadm show-hc-result	지정된 규칙 이름 rule-name과 연관된 서버에 대한 건전성 검사 결과를 표시합니다. rule-name이 지정되지 않은 경우 모든 규칙에 대한 서버의 건전성 검사 결과가 표시됩니다.
ilbadm show-nat	NAT 테이블 정보를 표시합니다.
ilbadm create-servergroup	서버 그룹을 만듭니다. ilbadm add-server를 사용하여 서버를 더 추가할 수 있습니다.
ilbadm delete-servergroup	서버 그룹을 삭제합니다.
ilbadm show-servergroup	특정 서버 그룹 또는 모든 서버 그룹(지정된 서버 그룹이 없을 경우)을 나열합니다.
ilbadm enable-server	사용 안함으로 설정된 서버를 사용으로 설정합니다.
ilbadm disable-server	지정된 서버를 사용 안함으로 설정합니다.
ilbadm add-server	지정된 서버를 서버 그룹에 추가합니다.
ilbadm show-server	명명된 규칙과 연관된 서버 또는 모든 서버(규칙 이름이 지정되지 않은 경우)를 표시합니다.
ilbadm remove-server	서버 그룹에서 서버를 제거합니다.
ilbadm create-healthcheck	규칙 설정에 사용할 수 있는 건전성 검사 정보를 설정합니다.
ilbadm show-persist	세션 지속성 매핑 테이블을 표시합니다.



표 22-2 로드 균형 조정 규칙 조작에 사용되는 ILB 명령 및 하위 명령 (계속)

ILB 명령	설명
<code>ilbadm export-config filename</code>	<code>ilbadm import</code> 를 사용하여 필요에 따라 가져오기 작업에 적합한 형식으로 기존 구성 파일을 내보냅니다. <i>filename</i> 이 지정되지 않은 경우 <code>stdout</code> 에 <code>ilbadm export</code> 를 씁니다.
<code>ilbadm import-config -p filename</code>	파일을 가져오고 기존 구성을 가져온 파일의 내용으로 바꿉니다. <i>filename</i> 이 지정되지 않은 경우 <code>stdin</code> 에서 <code>ilbadm import</code> 를 읽습니다.



## 통합 로드 밸런서 구성(작업)

---

이 장은 통합 로드 밸런서(ILB)의 설치 및 구성에 대해 설명하며 다음 절로 구성되어 있습니다.

- 347 페이지 “통합 로드 밸런서 설치”
- 348 페이지 “ILB 사용 및 사용 안함”
- 349 페이지 “ILB 구성”
- 353 페이지 “ILB 고가용성 구성(능동-수동 모드 전용)”
- 358 페이지 “ILB 구성 하위 명령에 대한 사용자 권한 부여 설정”
- 359 페이지 “ILB 서버 그룹 관리”
- 360 페이지 “ILB에서 백엔드 서버 관리”
- 362 페이지 “ILB에서 건전성 검사 관리”
- 365 페이지 “ILB 규칙 관리”
- 367 페이지 “ILB 통계 표시”
- 368 페이지 “Import 및 Export 하위 명령 사용”

### 통합 로드 밸런서 설치

이 절에서는 ILB 설치에 대해 설명합니다.

ILB는 커널과 userland의 두 부분으로 구성됩니다. 커널 부분은 Oracle Solaris 11 설치의 일부로 자동으로 설치됩니다. 그러나 ILB의 userland 부분을 얻으려면 사용자가 service/network/load-balancer/ilb 패키지에 있는 ilb를 수동으로 설치해야 합니다.

# ILB 사용 및 사용 안함

이 절에서는 ILB를 사용 및 사용 안함으로 설정하는 절차를 설명합니다.

## ▼ ILB를 사용으로 설정하는 방법

**시작하기 전에** 시스템의 RBAC(역할 기반 액세스 제어) 속성 파일에 다음 항목이 있는지 확인합니다. 항목이 없는 경우 수동으로 추가하십시오.

- 파일 이름: /etc/security/auth\_attr
  - `solaris.network.ilb.config::Network ILB Configuration::help=NetworkILBconf.html`
  - `solaris.network.ilb.enable::Network ILB Enable Configuration::help=NetworkILBenable.html`
  - `solaris.smf.manage.ilb::Manage Integrated Load Balancer Service States::help=SmfILBStates.html`
- 파일 이름: /etc/security/prof\_attr
  - `Network ILB::Manage ILB configuration via ilbadm:auths=solaris.network.ilb.config,solaris.network.ilb.enable;help=RtNetILB.htm`
  - 파일의 Network Management 항목에 `solaris.smf.manage.ilb`가 있어야 합니다.
- 파일 이름: /etc/user\_attr
  - `daemon:::auths=solaris.smf.manage.ilb,solaris.smf.modify.application`

### 1 ILB Management 권한 프로파일이 포함된 역할의 사용자 또는 슈퍼유저로 로그인합니다.

사용자가 만든 역할에 ILB Management 권한 프로파일을 할당할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.

### 2 IPv4 또는 IPv6 중에서(또는 둘 다) 적절한 전달 서비스를 사용으로 설정합니다.

```
#svcadm enable svc:/network/ipv4-forwarding
# svcadm enable svc:/network/ipv6-forwarding
```

### 3 ILB 서비스를 사용으로 설정합니다.

```
# svcadm enable ilb
```

### 4 ILB 서비스가 사용으로 설정되었는지 확인합니다.

```
# svcs ilb
```

## ▼ ILB를 사용 안함으로 설정하는 방법

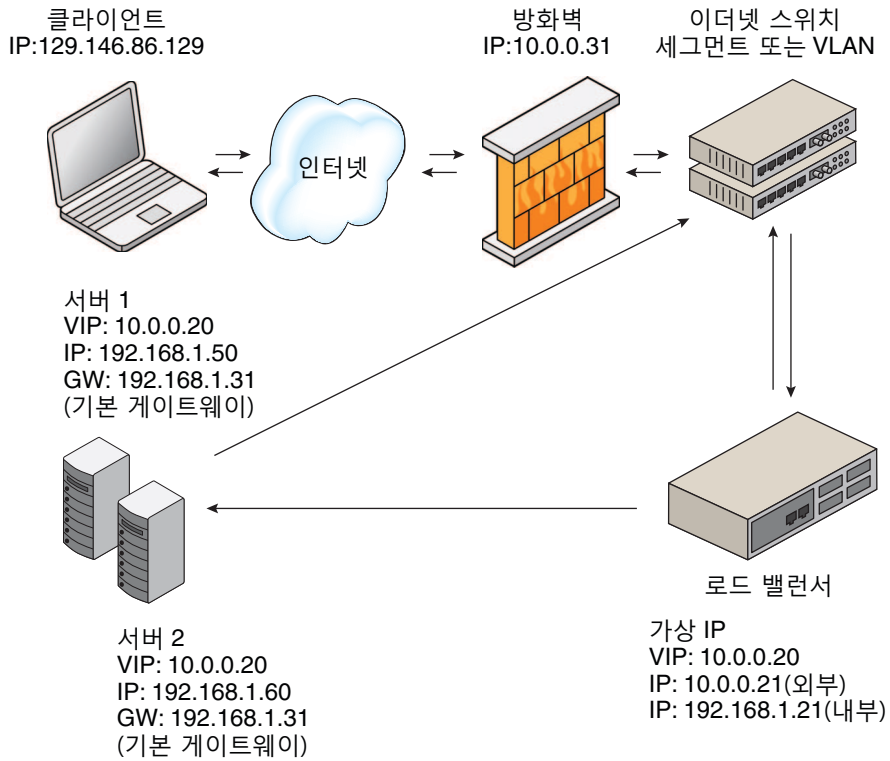
- 1 ILB Management 권한 프로파일을 포함하는 역할을 맡거나, 수퍼유저가 됩니다.  
사용자가 만든 역할에 ILB Management 권한 프로파일을 할당할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 [Oracle Solaris 관리: 보안 서비스의 “RBAC 초기 구성\(작업 맵\)”](#)을 참조하십시오.
- 2 ILB 서비스를 사용 안함으로 설정합니다.  
`# svcadm disable ilb`
- 3 ILB 서비스가 사용 안함으로 설정되었는지 확인합니다.  
`# svcs ilb`

## ILB 구성

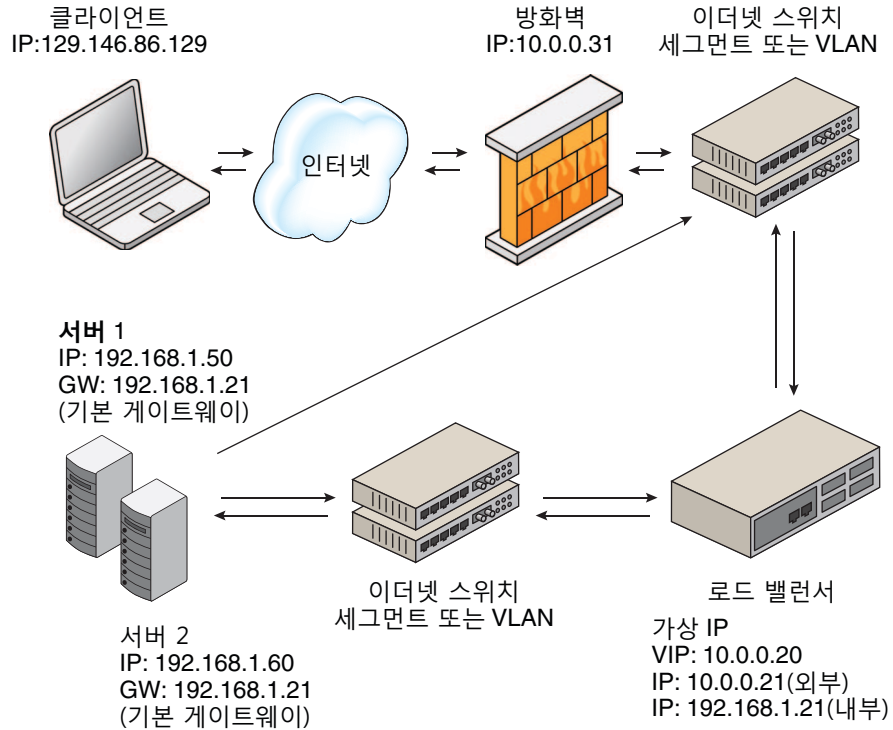
이 절에서는 DSR, Half-NAT, Full-NAT 토폴로지를 사용한 ILB 구현에 대해 설명합니다.

### DSR, Full-NAT, Half-NAT 토폴로지

다음 그림은 DSR 토폴로지를 사용한 ILB 구현을 보여줍니다.



ILB는 Half-NAT 및 Full-NAT 모드에서 모두 작동합니다. 일반적인 NAT 토폴로지 구현은 다음 그림에 나타난 것과 같습니다.



## Half-NAT 로드 균형 조정 토폴로지

Half-NAT 모드의 ILB 작동에서는 ILB가 패킷의 헤더에 대상 IP 주소만 다시 씁니다. Half-NAT 구현을 사용하는 경우 서버가 상주하는 동일한 서브넷에서 서비스의 가상 IP(VIP) 주소에 연결할 수 없습니다.

표 23-1 Half-NAT 구현의 요청 흐름과 응답 흐름

요청 흐름	소스 IP 주소	대상 IP 주소
1. 클라이언트 -> 로드 밸런서	클라이언트	로드 밸런서의 VIP
2. 로드 밸런서 -> 서버	클라이언트	서버
응답 흐름		
3. 서버 -> 로드 밸런서	서버	클라이언트
4. 로드 밸런서 -> 클라이언트	로드 밸런서의 VIP	클라이언트

클라이언트 PC를 서버와 동일한 네트워크에 연결하면 의도한 서버가 클라이언트에 직접 응답합니다. 4번째 단계는 발생하지 않으므로 클라이언트에 대한 서버 응답의 소스 IP 주소가 잘못되었습니다. 클라이언트가 로드 밸런서로 연결 요청을 보내면 의도한 서버에서 응답이 발생합니다. 그 이후로 클라이언트의 IP 스택이 정확하게 모든 응답을 삭제합니다.

이 경우 요청 흐름과 응답 흐름은 다음 표에 나타난 대로 진행합니다.

표 23-2 Half-NAT 구현의 요청 흐름과 응답 흐름

요청 흐름	소스 IP 주소	대상 IP 주소
1. 클라이언트 -> 로드 밸런서	클라이언트	로드 밸런서의 VIP
2. 로드 밸런서 -> 서버	클라이언트	서버
응답 흐름		
3. 서버 -> 클라이언트	서버	클라이언트

## Full-NAT 로드 균형 조정 토폴로지

Full-NAT 구현에서는 트래픽이 로드 밸런서를 양방향으로 통과하도록 소스 및 대상 IP 주소를 다시 씁니다. Full-NAT 토폴로지에서는 서버가 상주하는 동일한 서브넷에서 VIP에 연결할 수 있습니다. 다음 표는 ILB의 Full-NAT 토폴로지를 설명합니다. 서버를 통과하는 데 필요한 기본 경로는 없습니다. 로드 밸런서를 통과하는 기본 경로는 서브넷 C의 라우터 주소입니다. 이 시나리오에서 로드 밸런서는 프록시로 작동합니다.

표 23-3 Full-NAT 구현의 요청 흐름과 응답 흐름

요청 흐름	소스 IP 주소	대상 IP 주소
1. 클라이언트 -> 로드 밸런서	클라이언트	로드 밸런서의 VIP
2. 로드 밸런서 -> 서버	로드 밸런서의 인터페이스 주소(서브넷 C)	서버
응답 흐름		
3. 서버 -> 로드 밸런서	서버	로드 밸런서의 인터페이스 주소(서브넷 C)
4. 로드 밸런서 -> 클라이언트	로드 밸런서의 VIP	클라이언트



## ILB 고가용성 구성(능동-수동 모드 전용)

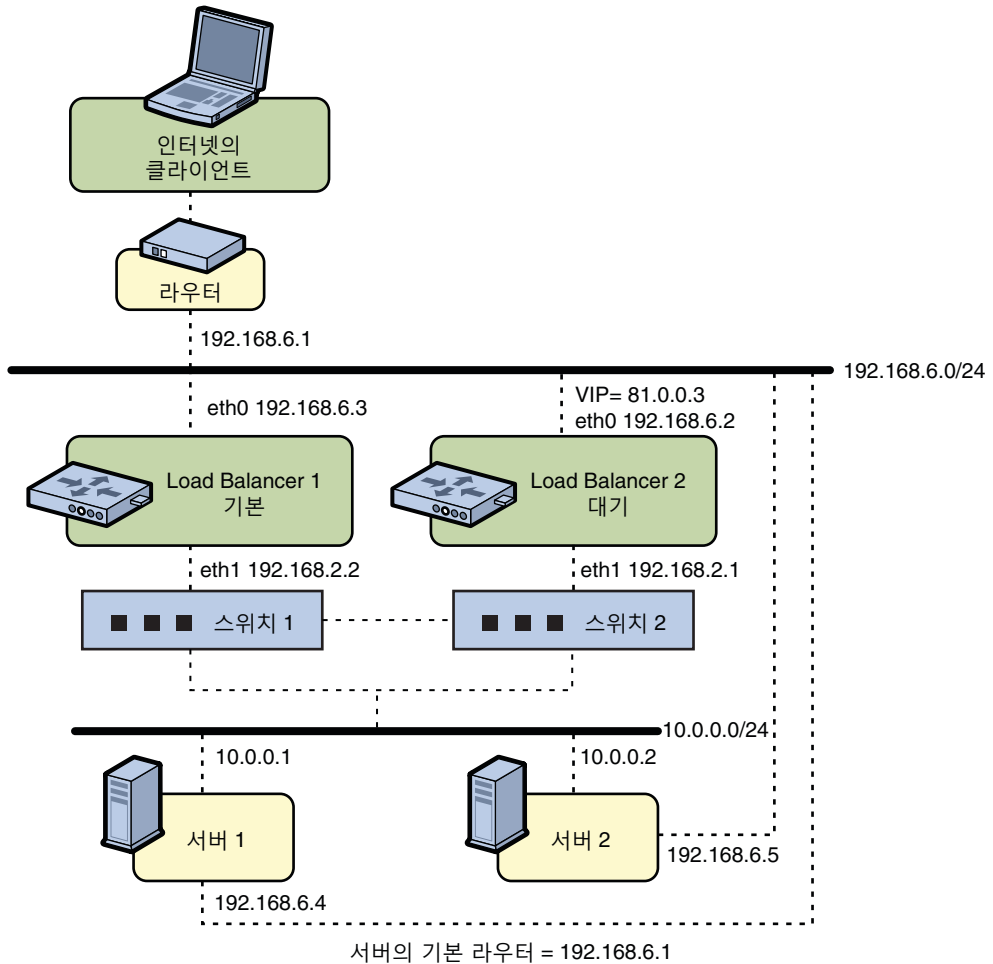
이 절에서는 DSR, Half-NAT, Full-NAT 토폴로지를 사용한 ILB의 고가용성 구성에 대해 설명합니다.

### DSR 토폴로지를 사용하여 ILB HA 구성

이 절에서는 DSR 토폴로지를 사용하여 고가용성(HA)을 이루도록 ILB 연결을 설정하는 방법을 설명합니다. 기본 로드 밸런서와 대기 로드 밸런서의 두 가지 로드 밸런서를 설정해야 합니다. 기본 로드 밸런서를 실패하면 대기 로드 밸런서가 기본 로드 밸런서의 역할을 맡습니다.

다음 그림은 HA를 이루도록 ILB 연결을 구성하기 위한 DSR 토폴로지를 보여줍니다.

## DSR 토폴로지



Load Balancer의 모든 VIP는 서브넷 192.168.6.0/24를 대상으로 하는 인터페이스에 구성됩니다.

## ▼ DSR 토폴로지를 사용하여 고가용성을 이루도록 ILB를 구성하는 방법

- 1 다음 로드 밸런서 명령을 사용하여 기본 및 대기 로드 밸런서를 모두 구성합니다.

```
# ilbadm create-servergroup -s server=10.0.0.1,10.0.0.2 sg1
# ilbadm create-rule -i vip=81.0.0.3,port=9001 \
-m lbalg=hash-ip-port,type=DSR -o servergroup=sg1 rule1
```

## 2 모든 서버의 lo0 인터페이스에 VIP가 구성되었는지 확인합니다.

```
Server1# ipadm create-addr -T static -d -a 81.0.0.3/24 lo0/server1
Server2# ipadm create-addr -T static -d -a 81.0.0.3/24 lo0/server2
```

## 3 Load Balancer 1이 기본 로드 밸런서로 작동하도록 구성합니다.

```
LB1# dladm create-vnic -m vrrp -V 1 -A inet -l eth0 vnic1
LB1# vrrpadm create-router -V 1 -A inet -l eth0 -p 255 vrrp1
LB1# ipadm create-addr -T static -d -a 81.0.0.3/24 vnic1/lb1
```

## 4 Load Balancer 2가 대기 로드 밸런서로 작동하도록 구성합니다.

```
LB2# dladm create-vnic -m vrrp -V 1 -A inet -l eth0 vnic1
LB2# vrrpadm create-router -V 1 -A inet -l eth0 -p 100 vrrp1
LB2# ipadm create-addr -T static -d -a 81.0.0.3/24 vnic1/lb2
```

위의 구성은 다음 실패 시나리오에 대해 보호를 제공합니다.

- Load Balancer 1을 실패하면 Load Balancer 2가 기본 역할을 맡고 VIP 81.0.0.3에 대한 주소 결정을 인수하여 클라이언트의 모든 패킷을 대상 IP 주소 81.0.0.3으로 처리합니다.

Load Balancer 1을 복구하면 Load Balancer 2가 대기 모드로 돌아갑니다.

- Load Balancer 1의 인터페이스 한쪽 또는 양쪽을 실패하면 Load Balancer 2가 기본 역할을 인수합니다. 따라서 Load Balancer 2가 VIP 81.0.0.3에 대한 주소 결정을 인수하여 클라이언트의 모든 패킷을 대상 IP 주소 81.0.0.3으로 처리합니다.

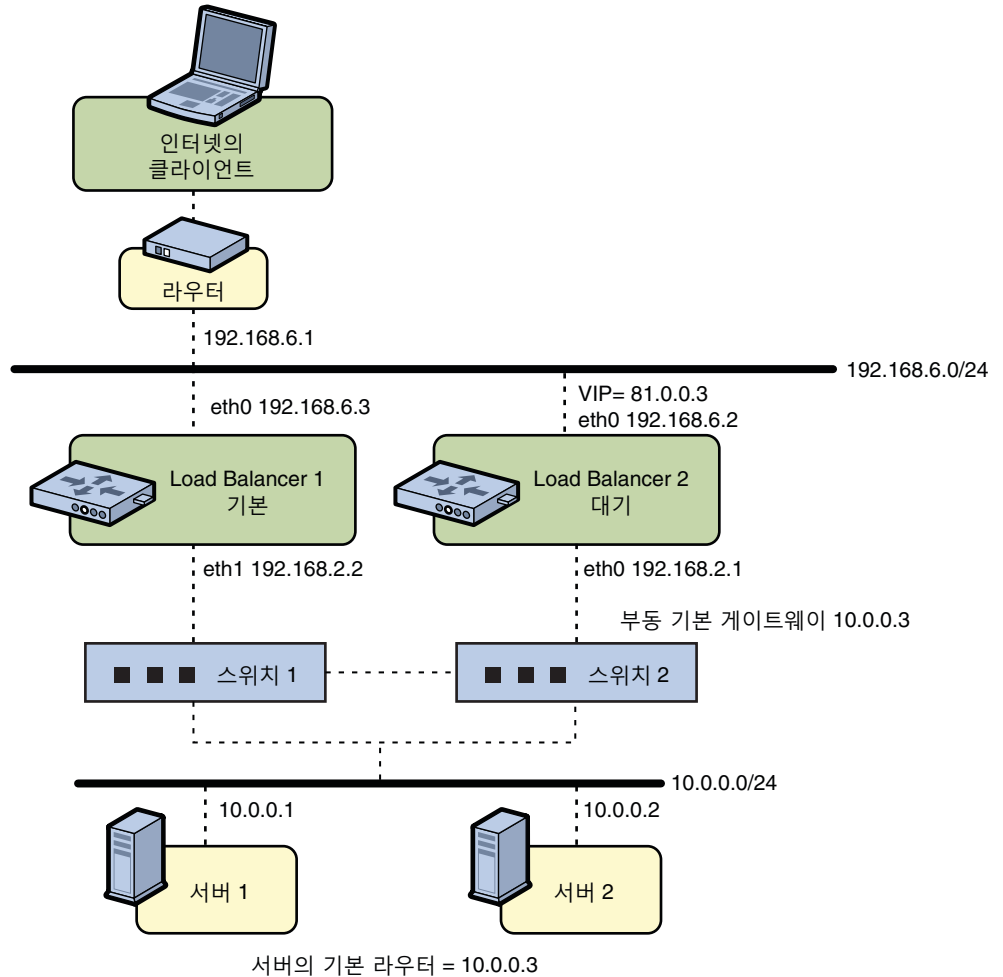
Load Balancer 1의 인터페이스 양쪽을 복구하면 Load Balancer 2가 대기 모드로 돌아갑니다.

# Half-NAT 토폴로지를 사용하여 ILB 고가용성 구성

이 절에서는 Half-NAT 토폴로지를 사용하여 HA를 이루도록 ILB 연결을 설정하는 방법을 설명합니다. 기본 로드 밸런서와 대기 로드 밸런서의 두 가지 로드 밸런서를 설정해야 합니다. 기본 로드 밸런서를 실패하면 대기 로드 밸런서가 기본 로드 밸런서의 역할을 맡습니다.

다음 그림은 HA를 이루도록 ILB 연결을 구성하기 위한 Half-NAT 토폴로지를 보여줍니다.

## Half-NAT 토폴로지



Load Balancer의 모든 VIP는 서브넷 192.168.6.0/24를 대상으로 하는 인터페이스에 구성됩니다.

### ▼ Half-NAT 토폴로지를 사용하여 고가용성을 이루도록 ILB를 구성하는 방법

- 1 기본 및 대기 로드 밸런서를 모두 구성합니다.

```
# ilbadm create servergroup -s server=10.0.0.1,10.0.0.2 sg1
# ilbadm create-rule -ep -i vip=81.0.0.3,port=9001-9006,protocol=udp \
-m lbalg=roundrobin,type=HALF-NAT,pmask=24 \
-h hc-name=hc1,hc-port=9006 \
```

```
-t conn-drain=70,nat-timeout=70,persist-timeout=70 -o servergroup=sg1 rule1
```

## 2 Load Balancer 1이 기본 로드 밸런서로 작동하도록 구성합니다.

```
LB1# dladm create-vnic -m vrrp -V 1 -A inet -l eth0 vnic1
LB1# ipadm create-addr -T static -d -a 81.0.0.3/24 vnic1/lb1
LB1# vrrpadm create-router -V 1 -A inet -l eth0 -p 255 vrrp1
LB1# dladm create-vnic -m vrrp -V 2 -A inet -l eth1 vnic2
LB1# ipadm create-addr -T static -d -a 10.0.0.3/24 vnic2/lb1
LB1# vrrpadm create-router -V 2 -A inet -l eth1 -p 255 vrrp2
```

## 3 Load Balancer 2가 대기 로드 밸런서로 작동하도록 구성합니다.

```
LB2# dladm create-vnic -m vrrp -V 1 -A inet -l eth0 vnic1
LB2# ipadm create-addr -T static -d -a 81.0.0.3/24 vnic1/lb2
LB2# vrrpadm create-router -V 1 -A inet -l eth0 -p 100 vrrp1
LB2# dladm create-vnic -m vrrp -V 2 -A inet -l eth1 vnic2
LB2# ipadm create-addr -T static -d -a 10.0.0.3/24 vnic2/lb2
LB2# vrrpadm create-router -V 2 -A inet -l eth1 -p 100 vrrp2
```

## 4 부동 기본 게이트웨이의 IP 주소를 양쪽 서버에 추가합니다.

```
# route add net 192.168.6.0/24 10.0.0.3
```

위의 구성은 다음 실패 시나리오에 대해 보호를 제공합니다.

- Load Balancer 1을 실패하면 Load Balancer 2가 기본 역할을 맡고 VIP 81.0.0.3에 대한 주소 결정을 인수하여 클라이언트의 모든 패킷을 대상 IP 주소 81.0.0.3으로 처리합니다. 또한 부동 게이트웨이 주소 10.0.0.3으로 보내진 모든 패킷을 처리해야 합니다.

Load Balancer 1을 복구하면 Load Balancer 2가 대기 모드로 돌아갑니다.

- Load Balancer 1의 인터페이스 한쪽 또는 양쪽을 실패하면 Load Balancer 2가 기본 역할을 인수합니다. 따라서 Load Balancer 2가 VIP 81.0.0.3에 대한 주소 결정을 인수하여 클라이언트의 모든 패킷을 대상 IP 주소 81.0.0.3으로 처리합니다. 또한 부동 게이트웨이 주소 10.0.0.3으로 보내진 모든 패킷을 처리해야 합니다.

Load Balancer 1의 인터페이스 양쪽을 복구하면 Load Balancer 2가 대기 모드로 돌아갑니다.

---

주 - 현재 ILB 구현은 기본 및 대기 로드 밸런서를 동기화하지 않습니다. 기본 로드 밸런서를 실패하고 대기 로드 밸런서가 역할을 인수할 때 기존 연결을 실패합니다. 그러나 동기화 없는 HA는 기본 로드 밸런서를 실패한 상황에서 여전히 유용합니다.

---

## ILB 구성 하위 명령에 대한 사용자 권한 부여 설정

다음 ILB 구성 하위 명령을 실행하려면 `solaris.network.ilb.config` RBAC 권한 부여가 있어야 합니다.

```
create-servergroup
delete-servergroup groupname
show-servergroup
add-server
remove-server
enable-server
disable-server
show-server
create-healthcheck
show-healthcheck
delete-healthcheck
show-rule
delete-rule
enable-rule
disable-rule
show-statistics
show-hc-result
show-nat
show-persist
export-config
import-config
```

기존 사용자에게 권한 부여를 지정하려면 [Oracle Solaris 관리: 보안 서비스의 9 장, “역할 기반 액세스 제어 사용\(작업\)”](#)을 참조하십시오.

또한 시스템에 새 사용자 계정을 만들 때 권한 부여를 제공할 수 있습니다. 예를 들면 다음과 같습니다.

```
useradd -g 10 -u 1210 -A solaris.network.ilb.config ilbadmin
```

`useradd` 명령은 `/etc/passwd`, `/etc/shadow`, `/etc/user_attr` 파일에 새 사용자를 추가합니다. `-A` 옵션은 사용자에게 권한 부여를 지정합니다.

## ILB 서버 그룹 관리

ilbadm 명령을 사용하여 ILB 서버 그룹을 만들고 삭제 및 나열할 수 있습니다. 서버 그룹의 정의는 [336 페이지 “ILB 용어”](#)를 참조하십시오.

### ▼ 서버 그룹을 만드는 방법

- 1 만들려는 서버 그룹의 이름을 선택합니다.
- 2 서버 그룹에 포함될 서버를 선택합니다.  
서버는 호스트 이름/IP 주소 및 선택적 포트로 지정할 수 있습니다.
- 3 서버 그룹을 만듭니다.

```
# ilbadm create-servergroup -s servers=webserv1,webserv2,webserv3 webgroup
```

#### 예 23-1 서버 그룹 만들기

다음 예는 세 개의 서버로 구성된 webgroup이라는 서버 그룹을 만듭니다.

```
# ilbadm create-servergroup -s servers=webserv1,webserv2,webserv3 webgroup
```

### ▼ 서버 그룹을 삭제하는 방법

- 1 제거할 서버 그룹을 선택합니다.  
서버 그룹이 활성 규칙에서 사용 중이면 안됩니다. 그렇지 않으면 삭제를 실패합니다.
- 2 터미널 창에서 서버 그룹을 삭제합니다.

```
# ilbadm delete-servergroup webgroup
```

#### 예 23-2 서버 그룹 삭제

다음 예는 webgroup이라는 서버 그룹을 제거합니다.

```
# ilbadm delete-servergroup webgroup
```

## 서버 그룹 표시

터미널 창에서 show-servergroup 하위 명령을 입력하여 특정 서버 그룹 또는 모든 서버 그룹에 대한 정보를 얻습니다.

다음 예는 모든 서버 그룹에 대한 세부 정보를 나열합니다.

```
# ilbadm show-servergroup -o all
```

sgname	serverID	minport	maxport	IP_address
specgroup	_specgroup.0	7001	7001	199.199.67.18
specgroup	_specgroup.1	7001	7001	199.199.67.19
test123	_test123.0	7001	7001	199.199.67.18
test123	_test123.1	7001	7001	199.199.67.19

## ILB에서 백엔드 서버 관리

ilbadm을 사용하여 서버 그룹 내에 하나 이상의 백엔드 서버를 추가, 제거하고 사용 및 사용 안함으로 설정할 수 있습니다. 정의 목록은 [336 페이지 “ILB 용어”](#)를 참조하십시오.

### ▼ 서버 그룹에 백엔드 서버를 추가하는 방법

- 서버 그룹에 백엔드 서버를 추가합니다.

서버 사양은 호스트 이름 또는 IP 주소를 포함해야 하고, 선택적 포트 또는 포트 범위를 포함할 수도 있습니다. 동일한 IP 주소의 서버 항목은 서버 그룹 내에서 허용되지 않습니다.

```
# ilbadm add-server -e -s server=192.168.89.1,192.168.89.2 ftpgroup
# ilbadm add-server -e -s server=[2001:7::feed:6]:8080 sgrp
```

-e 옵션은 서버를 그룹에 추가하는 동시에 사용으로 설정합니다.

---

주 - IPv6 주소는 대괄호로 둘러싸야 합니다.

---

#### 예 23-3 서버 그룹에 백엔드 서버 추가

다음 예는 서버 그룹 ftpgroup 및 sgrp에 서버를 추가하고 사용으로 설정합니다.

```
# ilbadm add-server -e -s \
server=192.168.89.1,192.168.89.2 ftpgroup
# ilbadm add-server -e -s server=[2001:7::feed:6]:8080 sgrp
```



## ▼ 서버 그룹에서 백엔드 서버를 제거하는 방법

- 1 특정 서버 그룹에서 서버를 제거하려면 다음 단계를 따릅니다.
  - a. 서버 그룹에서 제거할 서버의 서버 ID를 식별합니다. 서버 ID는 `show-servergroup -o all` 하위 명령의 출력에서 얻을 수 있습니다.
  - b. 서버를 제거합니다.
 

```
# ilbadm remove-server -s server=_specgroup.0 specgroup
```
- 2 모든 서버 그룹에서 서버를 제거하려면 아래 제공된 단계를 따릅니다.
  - a. 제거할 서버의 IP 주소 및 호스트 이름을 식별합니다.
  - b. `ilbadm show-servergroup -o all` 명령의 출력을 사용하여 서버를 포함하는 서버 그룹을 식별합니다.
  - c. 각 서버 그룹에 대해 다음 하위 명령을 실행하여 서버 그룹에서 서버를 제거합니다.

### 예 23-4 서버 그룹에서 백엔드 서버 제거

다음 예는 서버 그룹 `websg`에서 서버 ID `10.1.1.2`를 가진 서버를 제거합니다.

```
# ilbadm remove-server -s server=_specgroup.0 specgroup
```

다음 사항에 유의하십시오.

- NAT 또는 Half-NAT 규칙에서 서버를 사용 중인 경우 제거 전에 `disable-server` 하위 명령을 사용하여 서버를 사용 안함으로 설정합니다. 서버가 사용 안함으로 설정되면 연결 드레인 상태로 진입합니다. 모든 연결이 드레인된 후에 `remove-server` 하위 명령을 사용하여 서버를 제거할 수 있습니다. `disable-server` 명령을 실행한 후에 정기적으로 NAT 테이블을 검사하여(`show-nat` 명령 사용) 문제의 서버에 여전히 연결이 있는지 확인합니다. 모든 연결이 드레인된 후에(`show-nat` 명령 출력에 서버가 표시되지 않음) `remove-server` 명령을 사용하여 서버를 제거할 수 있습니다.
- `conn-drain` 시간 초과 값이 설정된 경우 시간 초과 기간 종결 시 연결 드레인 상태가 완료됩니다. `conn-drain` 시간 초과 값의 기본값은 0이며, 이는 연결이 적절하게 종료될 때까지 계속 기다립니다.

## ▼ 백엔드 서버를 다시 사용 또는 사용 안함으로 설정하는 방법

- 1 다시 사용 또는 사용 안함으로 설정할 서버의 IP 주소, 호스트 이름 또는 서버 ID를 식별합니다. IP 주소나 호스트 이름이 지정된 경우 연관된 모든 규칙에 대해 서버가 다시 사용 또는 사용 안함으로 설정됩니다. 서버 ID가 지정된 경우 서버 ID와 연관된 특정 규칙에 대해 서버가 다시 사용 또는 사용 안함으로 설정됩니다.

---

주 - 서버가 다중 서버 그룹에 속할 경우 여러 서버 ID를 가질 수 있습니다.

---

- 2 서버를 다시 사용 또는 사용 안함으로 설정합니다.

```
# ilbadm enable-server server
# ilbadm disable-server server
```

### 예 23-5 백엔드 서버 다시 사용 및 사용 안함

다음 예에서 서버 ID websg.1을 가진 서버를 사용으로 설정했다가 사용 안함으로 설정합니다.

```
# ilbadm enable-server websg.1
# ilbadm disable-server websg.1
```

## ILB에서 건전성 검사 관리

ILB는 사용자에게 다음과 같은 선택적 유형의 서버 건전성 검사를 제공합니다.

- 내장 ping 검사
- 내장 TCP 검사
- 내장 UDP 검사
- 건전성 검사로 실행할 수 있는 사용자 제공 테스트

기본적으로 ILB는 건전성 검사를 수행하지 않습니다. 로드 균형 조정 규칙을 만들 때 각 서버 그룹에 대한 건전성 검사를 지정할 수 있습니다. 로드 균형 조정 규칙당 하나의 건전성 검사만 구성할 수 있습니다. 가상 서비스가 사용으로 설정된 동안, 사용으로 설정된 가상 서비스와 연관된 서버 그룹에 건전성 검사가 자동으로 시작되고 정기적으로 반복합니다. 가상 서비스를 사용 안함으로 설정하면 즉시 건전성 검사가 중지됩니다. 가상 서비스를 다시 사용으로 설정할 때 이전 건전성 검사 상태는 보존되지 않습니다.

건전성 검사를 실행하기 위해 TCP, UDP 또는 사용자 정의 테스트 검사를 지정할 때 ILB는 기본적으로 ping 검사를 보내 지정된 TCP, UDP 또는 사용자 정의 테스트 검사를 서버로 보내기 전에 서버에 연결할 수 있는지 확인합니다. ping 검사는 서버 건전성을 모니터링하는 방법입니다. ping 검사를 실패하면 해당 서버가 건전성 검사 상태

unreachable과 함께 사용 안함으로 설정됩니다. ping 검사를 성공하지만 TCP, UDP 또는 사용자 정의 테스트 검사를 실패하면 건전성 검사 상태 **dead**와 함께 서버가 사용 안함으로 설정됩니다.

ilbadm 명령을 사용하여 건전성 검사를 만들고 삭제 및 나열할 수 있습니다. 정의 목록은 [336 페이지 “ILB 용어”](#)를 참조하십시오.

## 건전성 검사 만들기

다음 예에서 두 가지 건전성 검사 *objects, hc1* 및 *hc-myscript*가 생성됩니다. 첫번째 건전성 검사는 내장 TCP 검사를 사용합니다. 두번째 건전성 검사는 사용자 정의 테스트인 */var/tmp/my-script*를 사용합니다.

```
# ilbadm create-healthcheck \
-h hc-timeout=3,hc-count=2,hc-interval=8,hc-test=tcp hc1
# ilbadm create-healthcheck \
-h hc-timeout=3,hc-count=2,hc-interval=8,hc-test=/var/tmp/my-script hc-myscript
```

hc-test는 건전성 검사의 유형을 지정합니다.

hc-interval은 연속 건전성 검사 사이의 간격을 지정합니다. 동기화를 피하기 위해 실제 간격은  $0.5 * hc-interval$  및  $1.5 * hc-interval$  사이에 무작위로 설정됩니다.

hc-timeout은 건전성 검사가 완료되지 않은 경우 실패한 것으로 간주할 때 시간 초과를 지정합니다.

hc-count는 hc-test 건전성 검사를 실행할 시도 횟수를 지정합니다.

---

주 -hc-test의 포트 사양은 create-rule 하위 명령의 hc-port 키워드로 지정됩니다. 자세한 내용은 [ilbadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

---

## 사용자 제공 테스트 세부 정보

사용자 제공 테스트는 다음 기준을 충족해야 합니다.

- 테스트는 이진 또는 스크립트일 수 있습니다.
- 테스트는 시스템의 어디에든 상주할 수 있으며 create-healthcheck 하위 명령을 사용할 때 절대 경로를 지정해야 합니다.

create-rule 하위 명령에서 건전성 검사 사양의 일부로 테스트를 지정할 때(예: */var/tmp/my-script*) 다음과 같이 ilbd 데몬이 프로세스를 포크하고 테스트를 실행합니다.

```
/var/tmp/my-script $1 $2 $3 $4 $5
```

인수의 설명은 다음과 같습니다.

- \$1 VIP(리터럴 IPv4 또는 IPv6 주소)
- \$2 서버 IP(리터럴 IPv4 또는 IPv6 주소)
- \$3 프로토콜(UDP, TCP를 문자열로)
- \$4 숫자 포트 범위(hc-port에 대한 사용자 지정 값)
- \$5 실패를 반환하기 전에 테스트가 기다리는 최대 시간(초). 지정된 시간을 넘어 테스트를 실행하면 중지될 수 있고 테스트가 실패로 간주됩니다. 이 값은 사용자 정의 값으로 hc-timeout에 지정됩니다.

사용자 제공 테스트 *my-script*는 모든 인수를 사용할 수도 있고 아닐 수도 있지만, 반드시 다음 중 하나를 반환해야 합니다.

- 마이크로초 단위의 왕복 시간(RTT)
- 테스트가 RTT를 계산하지 않는 경우 0 값
- 실패한 경우 -1 값

기본적으로 건전성 검사 테스트는 PRIV\_PROC\_FORK, RIV\_PROC\_EXEC, RIV\_NET\_ICMPACCESS 권한으로 실행됩니다.

더 광범위한 권한 세트가 필요한 경우 테스트에 `setuid`를 구현해야 합니다. 권한에 대한 자세한 내용은 [privileges\(5\)](#) 매뉴얼 페이지를 참조하십시오.

## 건전성 검사 삭제

다음 예는 *hc1*이라는 건전성 검사를 삭제합니다.

```
# ilbadm destroy-healthcheck hc1
```

## 건전성 검사 나열

`list-healthcheck` 하위 명령을 사용하여 구성된 건전성 검사에 대한 세부 정보를 얻을 수 있습니다. 다음 예는 두 가지 구성된 건전성 검사를 나열합니다.

```
# ilbadm list-healthcheck
```

NAME	TIMEOUT	COUNT	INTERVAL	DEF_PING	TEST
hc1	3	2	8	Y	tcp
hc2	3	2	8	N	/var/usr-script

## 건전성 검사 결과 표시

list-hc-result 하위 명령을 사용하여 건전성 검사 결과를 얻을 수 있습니다. 규칙이나 건전성 검사가 지정되지 않은 경우 모든 건전성 검사가 나열됩니다.

다음 예는 rule1이라는 규칙과 연관된 건전성 검사 결과를 표시합니다.

```
# ilbadm list-hc-result rule1
```

RULE	HC	SERVERID	TEST	STATUS	FAIL	LAST	NEXT
rule1	hc1	sg1:0	tcp	server-alive3		11:23:30	11:23:40
rule1	hc1	sg1:1	tcp	server-dead 4		11:23:30	11:23:40

## ILB 규칙 관리

ilbadm을 사용하여 로드 균형 조정 규칙을 만들고 삭제 및 나열할 수 있습니다. 로드 균형 조정 규칙의 정의 및 규칙을 만드는 데 필요한 매개변수는 [336 페이지 “ILB 용어”](#)를 참조하십시오.

### ▼ 규칙을 만드는 방법

- 1 적절한 백엔드 서버를 포함하는 서버 그룹을 만듭니다.

```
# ilbadm create-servergroup -s server=60.0.0.10:6000-6009,60.0.0.11:7000-7009 sg1
```

- 2 서버 건전성 검사를 규칙과 연관시키려면 건전성 검사 객체를 만듭니다.

```
# ilbadm create-healthcheck -h hc-test=tcp,hc-timeout=2,hc-count=3,hc-interval=10 hc1
```

- 3 규칙과 연관될 VIP, 포트 및 선택적 프로토콜을 식별합니다.

- 4 DSR, Full-NAT, Half-NAT 중에서 사용할 작업을 선택합니다. NAT가 선택된 경우 proxy-src 주소로 사용될 IP 주소 범위를 지정해야 합니다.

- 5 사용될 로드 균형 조정 알고리즘을 선택합니다.

- 6 다른 선택적 기능을 선택합니다. 세부 정보는 [ilbadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

- 7 규칙 이름을 선택합니다.

- 8 규칙을 만들고 사용으로 설정합니다.

```
# ilbadm create-rule -e -i vip=81.0.0.10,port=5000-5009,protocol=tcp\
-m lbalg=rr,type=NAT,proxy-src=60.0.0.101-60.0.0.104,persist=24 -h hc-name=hc1 -o servergroup=sg1 rule1
```

예 23-6 전전성 검사 세션 지속성과 함께 Full-NAT 규칙 만들기

이 예는 hc1이라는 전전성 검사와 sg1이라는 서버 그룹(두 개의 서버로 구성되고 각각 포트 범위로 지정)을 만듭니다. 마지막 명령은 Full-NAT 모드에서 rule1이라는 규칙을 만들어 사용으로 설정하고 서버 그룹 및 전전성 검사에 연관시킵니다. 서버 그룹 및 전전성 검사 만들기는 규칙 만들기보다 선행되어야 합니다.

```
ilbadm create-healthcheck -h hc-test=tcp,hc-timeout=2,hc-count=3,hc-interval=10 hc1
ilbadm create-servergroup -s server=60.0.0.10:6000-6009,60.0.0.11:7000-7009 sg1
ilbadm create-rule -e -i vip=81.0.0.10,port=5000-5009,protocol=tcp \
-m lbalg=rr,type=NAT,proxy-src=60.0.0.101-60.0.0.104,persist=/24
-h hc-name=hc1 -o servergroup=sg1 rule1
```

NAT/Half-NAT 규칙을 만들 때 conn-drain 시간 초과 값을 지정할 것을 권장합니다. conn-drain 시간 초과 값의 기본값은 0이며, 이는 연결이 적절하게 종료될 때까지 계속 기다립니다.

규칙 삭제

규칙을 삭제하려면 delete-rule 하위 명령을 사용합니다. 모든 규칙을 제거하려면 -a 옵션을 사용합니다. 다음 예는 rule1이라는 규칙을 삭제합니다.

```
# ilbadm delete-rule rule1
```

규칙 나열

규칙의 구성 세부 정보를 나열하려면 list-rule 하위 명령을 사용합니다. 규칙 이름이 지정되지 않으면 모든 규칙에 대한 정보가 제공됩니다.

```
# ilbadm list-rule
```

Rulename (+ = enabled)	LB-alg	Type	Proto	VIP/port
rule-http +	HIPP	H-NAT	TCP	10.0.0.1/http
rule-dns	HIP	DSR	UDP	10.0.0.1/53
rule-abc	RR	NAT	TCP	2003::1/1024
rule-xyz +	HIPV	NAT	TCP	2003::1/2048-2050

## ILB 통계 표시

ilbadm 명령을 사용하여 서버나 규칙의 통계를 인쇄하거나 NAT 테이블 정보 및 세션 지속성 매핑 테이블을 표시하는 등의 정보를 얻을 수 있습니다. 정의 목록은 [336 페이지](#) “ILB 용어”를 참조하십시오.

### show-statistics 하위 명령을 사용하여 통계 정보 얻기

show-statistics 하위 명령을 사용하여 로드 분배 세부 정보를 봅니다. 다음 예는 show-statistics 하위 명령의 사용법을 보여줍니다.

```
ilbadm show-statics
PKT_P  BYTES_P  PKT_U  BYTES_U  PKT_D  BYTES_D
9       636      0       0       0       0
```

설명

- PKT\_P: 처리된 패킷
- BYTES\_P: 처리된 바이트
- PKT\_U: 처리되지 않은 패킷
- BYTES\_U: 처리되지 않은 바이트

### NAT 연결 테이블 표시

show-nat 하위 명령을 사용하여 NAT 연결 테이블을 봅니다. 이 명령의 연속적 실행에서 요소의 상대적 위치에 대해 어떤 가정도 없어야 합니다. 예를 들어, {{ ilbadm show-nat 10}}을 두 번 실행하면 (특히 혼잡한 시스템에서) 똑같은 10개 항목이 두 번 표시된다고 보증할 수 없습니다. 개수 값이 지정되지 않은 경우 전체 NAT 연결 테이블이 표시됩니다.

다음 예는 NAT 연결 테이블에서 5개 항목을 표시합니다.

예 23-7 NAT 연결 테이블 항목 ilbadm show-nat 5

```
UDP: 124.106.235.150.53688 > 85.0.0.1.1024 >>> 82.0.0.39.4127 > 82.0.0.56.1024
UDP: 71.159.95.31.61528 > 85.0.0.1.1024 >>> 82.0.0.39.4146 > 82.0.0.55.1024
UDP: 9.213.106.54.19787 > 85.0.0.1.1024 >>> 82.0.0.40.4114 > 82.0.0.55.1024
UDP: 118.148.25.17.26676 > 85.0.0.1.1024 >>> 82.0.0.40.4112 > 82.0.0.56.1024
UDP: 69.219.132.153.56132 > 85.0.0.1.1024 >>> 82.0.0.39.4134 > 82.0.0.55.1024
```

항목의 형식은 다음과 같습니다.

```
T: IP1 > IP2 >>> IP3 > IP4
```

T: The transport protocol used in this entry.

IP1: The client's IP address and port.

IP2: The VIP and port.

IP3: If half-NAT mode, the client's IP address and port.

예 23-7 NAT 연결 테이블 항목 `ilbadm show-nat 5` (계속)

If full-NAT mode, the client's IP address and port.  
IP4: The back-end server's IP address and port.

## 세션 지속성 매핑 테이블 표시

`show-persist` 하위 명령을 사용하여 세션 지속성 매핑 테이블을 봅니다.

예 23-8 `ilbadm show-persist 5`

다음 예는 테이블에서 5개 항목을 표시합니다.

```
rule2: 124.106.235.150 --> 82.0.0.56
rule3: 71.159.95.31 --> 82.0.0.55
rule3: 9.213.106.54 --> 82.0.0.55
rule1: 118.148.25.17 --> 82.0.0.56
rule2: 69.219.132.153 --> 82.0.0.55
```

항목의 형식은 다음과 같습니다.

R: IP1 --> IP2

R: The rule that this persistence entry is tied to.  
IP1: The client's IP address.  
IP2: The back-end server's IP address.

## Import 및 Export 하위 명령 사용

`export` 하위 명령은 현재 구성을 사용자 지정 파일로 내보냅니다. 그런 다음 이 정보를 `import` 하위 명령에 대한 입력으로 사용할 수 있습니다. 별도로 구성을 유지하라고 지시하지 않는 한, `import` 하위 명령은 가져오기 전에 기존 구성을 삭제합니다. 파일 이름을 생략하면 표준 입력에서 읽거나 표준 입력으로 씁니다.

ILB 구성을 내보내려면 `export-config` 명령을 사용합니다. 다음 예는 `import` 하위 명령을 사용하여 가져오기에 적합한 형식으로 현재 구성을 `/var/tmp/ilb_config` 파일로 내보냅니다.

```
# ilbadm export-config /var/tmp/ilb_config
```

ILB 구성을 가져오려면 `import-config` 명령을 사용합니다. 다음 예는 `/var/tmp/ilb_config` 파일의 구성 내용을 읽고 기존 구성을 대체합니다.

```
# ilbadm import-config /var/tmp/ilb_config
```



## Virtual Router Redundancy Protocol(개요)

---

VRRP(Virtual Router Redundancy Protocol)는 [Virtual Router Redundancy Protocol Version 3 for IPv4 and IPv6](#)에 지정된 인터넷 표준 프로토콜로, 고가용성을 제공하기 위해 Oracle Solaris에서 지원됩니다. Oracle Solaris에서는 VRRP 서비스를 구성 관리하는 대체 도구를 제공합니다.

LAN과 같은 네트워크를 설정할 때 고가용성 서비스를 제공하는 것이 매우 중요합니다. 네트워크의 신뢰성을 높이는 한 가지 방법은 네트워크에 중요 구성 요소의 백업을 제공하는 것입니다. 라우터, 스위치, 링크 등의 구성 요소를 네트워크에 추가하면 오류가 발생해도 서비스가 계속 실행될 수 있습니다. 네트워크 끝점에 중복성을 제공하는 것은 중요한 작업으로, VRRP로 손쉽게 수행할 수 있습니다. VRRP를 사용하면 LAN에서 가상 라우터를 사용할 수 있으므로 라우터에 대한 오류 복구가 가능합니다.

VRRP에 사용되는 용어에 대한 자세한 내용은 [370 페이지 “VRRP 용어”](#)를 참조하십시오.

이 장은 다음과 같은 절로 구성됩니다.

- [370 페이지 “VRRP 용어”](#)
- [370 페이지 “VRRP 아키텍처 개요”](#)
- [373 페이지 “VRRP 제한 사항”](#)

VRRP는 가상 라우터의 책임을 LAN 내의 VRRP 라우터 중 하나에 동적으로 지정하는 선택 프로토콜로, LAN에서 동적으로 구성된 라우터에 하나 이상의 백업 라우터를 제공합니다.

마스터 라우터라고 하는 VRRP 라우터는 가상 라우터와 연관된 IPv4 또는 IPv6 주소를 제어합니다. 가상 라우터는 마스터 라우터의 IP 주소로 전송되는 패킷을 전달합니다.

선택 프로세스는 이러한 IP 주소로 전송된 패킷을 전달하는 중 동적 페일오버를 제공합니다. VRRP는 정적 기본 경로 지정 환경에 내재되어 있는 단일 오류 지점을 제거합니다.

Oracle Solaris에서 VRRP 기능을 사용하면 모든 단말 호스트에서 동적 경로 지정 또는 라우터 검색 프로토콜을 구성할 필요 없이 경로 지정 프로세스에 대해 보다 가용성이 높은 기본 경로를 제공할 수 있습니다.

## VRRP 용어

이 절에서는 시스템에서 VRRP를 구현할 때 알아 두면 유용한 몇 가지 용어에 대해 설명합니다.

백업 라우터	활성 상태지만 마스터 상태는 아닌 VRID의 VRRP 인스턴스입니다. VRID에 대해 원하는 수의 백업이 있을 수 있습니다. 현재 마스터 라우터에서 오류가 발생할 경우 백업 라우터가 마스터 라우터 역할을 맡게 됩니다.
마스터 라우터	지정된 시점에 가상 라우터에 대해 경로 지정 기능을 수행하는 VRRP 인스턴스입니다. 지정된 VRID에 대한 마스터 라우터는 한 번에 하나만 활성 상태일 수 있습니다.
가상 IP 주소	다른 호스트가 네트워크 서비스를 확보하는 데 사용할 수 있는 VRID와 연관된 IP 주소입니다. VRIP는 VRID에 속한 VRRP 인스턴스에 의해 관리됩니다.
가상 MAC 주소	매체(예: MAC 주소 지정을 사용하는 이더넷)에서 실행되는 동안 VRRPA 인스턴스에 의해 사용되는 미리 정의된 MAC 주소입니다. 가상 MAC 주소로 경로 지정 기능을 제공하는 실제 라우터와 가상 라우터의 작업을 분리할 수 있으며, 가상 MAC 주소는 실제 MAC 주소 대신 사용됩니다. 가상 MAC 주소는 VRID에서 파생됩니다.
가상 라우터 ID (VRID)	가상 라우터를 식별하는 데 사용되는 고유한 숫자입니다. VRID는 지정된 네트워크 세그먼트에서 고유해야 합니다.
VNIC	시스템의 물리적 네트워크 어댑터를 기반으로 구성되는 의사 네트워크 인터페이스로, NIC(네트워크 인터페이스 카드)라고도 합니다. 물리적 인터페이스에는 VNIC가 여러 개 있을 수 있습니다. VNIC는 네트워크 가상화의 필수 구성 요소입니다. 자세한 내용은 <a href="#">Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 제III부</a> , “네트워크 가상화 및 리소스 관리”를 참조하십시오.
VRRP 인스턴스	VRRP 구현을 사용하여 라우터에서 실행되는 프로그램입니다. 하나의 VRRP 인스턴스가 여러 가상 라우터에 VRRP 기능을 제공할 수 있습니다.
VRRP 라우터	VRRP를 사용하는 하나 이상의 라우터 작업을 통해 생성되는 단일 라우터 이미지입니다.

## VRRP 아키텍처 개요

### VRRP 라우터

VRRP는 각 VRRP 라우터에서 실행되며 라우터의 상태를 관리합니다. 한 호스트에 여러 개의 VRRP 라우터가 구성될 수 있으며, 각 VRRP 라우터는 서로 다른 가상 라우터에 속합니다.

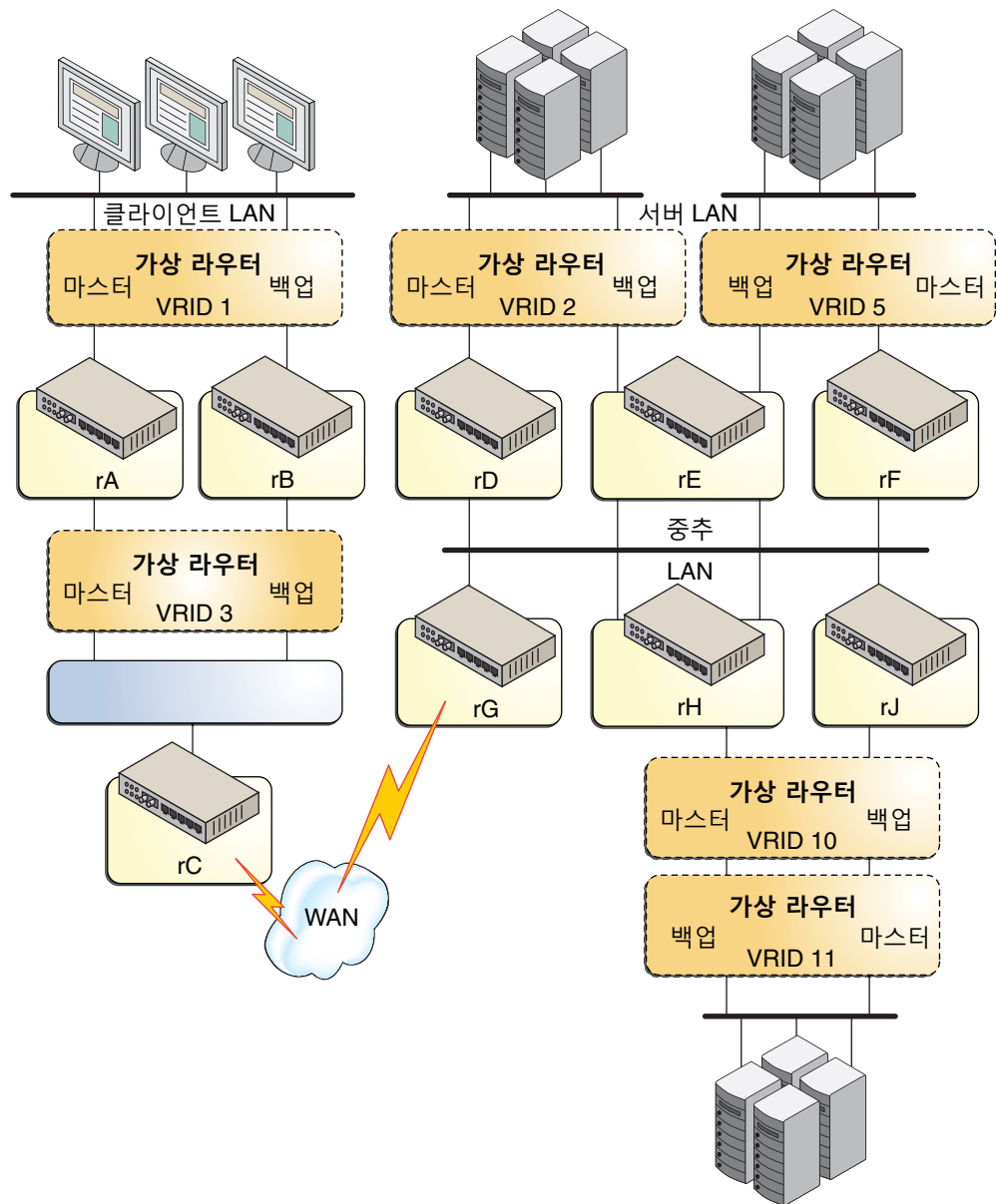
VRRP 라우터의 속성은 다음과 같습니다.

- 라우터 이름 - 시스템 전체 고유 식별자
- VRID - LAN 내에서 가상 라우터를 식별합니다.

- 기본 IP 주소 - VRRP 알림의 소스 IP 주소로 사용됩니다.
- 가상 IP 주소
- VRRP 매개변수 - 우선 순위, 알림 간격, 선취 모드 및 승인 모드를 포함합니다.
- VRRP 상태 정보 및 통계

## VRRP 프로세스

다음 그림은 VRRP의 작동 방식을 보여줍니다.



위 그림에 표시된 것과 같이, VRRP는 다음과 같은 구성 요소를 사용하여 작동합니다.

- 라우터 rA는 가상 라우터 VRID 1의 마스터 라우터이자 VRID 3의 백업 라우터입니다. 라우터 rA는 VRID 1의 VIP에 지정된 패킷의 경로 지정을 처리하며 VRID 3에 대해 경로 지정 역할을 맡습니다.
- 라우터 rB는 가상 라우터 VRID 3의 마스터 라우터이자 VRID 1의 백업 라우터입니다. 라우터 rB는 VRID 3의 VIP에 지정된 패킷의 경로 지정을 처리하며 VRID 1에 대해 경로 지정 역할을 맡습니다.
- 라우터 rC에는 VRRP 기능이 없지만 VRID 3에 VIP를 사용하여 클라이언트 LAN 서브넷에 도달합니다.
- 라우터 rD는 VRID 2의 마스터 라우터입니다. 라우터 rF는 VRID 5의 마스터 라우터입니다. 라우터 rE는 이 두 VRID의 백업 라우터입니다. rD 또는 rF에서 오류가 발생하면 rE는 해당 VRID의 마스터 라우터가 됩니다. rD와 rF에서 동시에 오류가 발생할 수 있습니다. 한 VRID에 대한 마스터 라우터인 VRRP 라우터가 또 다른 VRID에 대한 마스터 라우터가 될 수도 있습니다.
- 라우터 rG는 중추 LAN의 WAN 게이트웨이입니다. 중추에 연결된 모든 라우터는 OSPF(Open Shortest Path First)와 같은 동적 경로 지정 프로토콜을 사용하여 WAN에 있는 라우터와 경로 지정 정보를 공유합니다. 클라이언트 LAN 서브넷에 대한 경로가 VRID 3의 VIP를 거친다는 사실을 라우터 rC에서 알리더라도 VRRP는 이에 관여하지 않습니다.
- 라우터 rH는 VRID 10의 마스터 라우터이자 VRID 11의 백업 라우터입니다. 마찬가지로, 라우터 rJ는 VRID 11의 마스터 라우터이자 VRID 10의 백업 라우터입니다. 이 VRRP 로드 공유 구성은 하나의 라우터 인터페이스에 여러 개의 VRID가 있을 수 있음을 보여줍니다.

VRRP는 네트워크의 모든 시스템에 거의 완전한 경로 지정 중복성을 제공하는 네트워크 설계의 일부로 사용될 수 있습니다.

## VRRP 제한 사항

### 배타적 IP 영역 지원

각 배타적 IP 영역에서는 VRRP 라우터가 특정 영역에서 생성되면 VRRP 서비스 svc:/network/vrrp/default가 자동으로 사용으로 설정됩니다. VRRP 서비스는 이러한 특정 영역에 대한 VRRP 라우터를 관리합니다.

그러나 다음과 같은 이유로 배타적 IP 영역에 대한 지원은 제한적입니다.

- 비전역 영역 내에서는 VNIC를 만들 수 없습니다. 그러므로 먼저 전역 영역에서 VRRP VNIC를 만든 다음 VRRP 라우터가 상주하는 비전역 영역에 VNIC를 지정하십시오. 그러면 VRRP 라우터가 `vrpadm` 명령을 사용하여 비전역 영역에서 생성되어 시작됩니다.
- 단일 Oracle Solaris 시스템에서 동일한 가상 라우터와 함께 관여할 두 개의 VRRP 라우터를 서로 다른 영역에 만들 수 없습니다. 이 때문에 Oracle Solaris에서는 동일한 MAC 주소를 사용하는 두 개의 VNIC를 만들 수 없습니다.

## 다른 네트워크 기능과의 상호 작업

VRRP 서비스는 IPMP(IP Network Multipathing) 인터페이스에서 작동할 수 없습니다. 이유는 VRRP에서 IPMP가 IP 층에서 완전히 작동하는 동안 특정 VRRP MAC 주소를 필요로 하기 때문입니다.

또한 VRRP 가상 IP 주소는 정적으로만 구성될 수 있으며, IP 주소에 대한 두 개의 기존 자동 구성 도구인 `in.ndpd`(IPv6 자동 구성의 경우) 및 `dhcpgent`(DHCP 구성의 경우)로 자동 구성할 수 없습니다. 마스터 및 백업 VRRP 라우터(VNIC)가 동일한 MAC 주소를 공유하기 때문에 `in.ndpd`와 `dhcpgent`가 혼동될 수 있습니다. 결국 예상치 않은 결과가 발생할 수 있습니다. 따라서 IPv6 자동 구성 및 DHCP 구성은 VRRP VNIC를 통해 지원되지 않습니다. VRRP VNIC를 통해 IPv6 자동 구성 또는 DHCP를 구성하면 자동으로 구성된 IP 주소를 가져오려는 시도가 실패하고 자동 구성 작업도 실패합니다.

## VRRP 구성(작업)

---

VRRP 라우터는 VRRP를 실행하며 동일한 가상 라우터와 함께 관여하는 다른 VRRP 라우터에서 작동합니다. VRRP에는 가상 IP 주소 세트가 있습니다.

이 장에서는 다음 절에 대해 설명합니다.

- 376 페이지 “VRRP VNIC 만들기”
- 376 페이지 “vrrpadm 구성”
- 379 페이지 “보안 고려 사항”

LAN 내에서 각 가상 라우터는 VRID, 주소 그룹으로 고유하게 식별되며, 보호된 가상 IP 주소 집합과 연관됩니다.

각 참여 VRRP 라우터에는 우선 순위, 알림 간격 및 승인 모드와 같은 추가 매개변수가 있습니다. 한 번에 하나의 VRRP 라우터(마스터)만 가상 라우터의 책임을 맡게 되며 가상 IP 주소로 전송된 패킷을 전달합니다.

마스터에서 오류가 발생할 때마다 다른 참여 VRRP 라우터가 마스터 부재를 감지하고 다른 VRRP 라우터가 마스터로 선택되어 책임을 맡게 됩니다.

동일한 가상 라우터를 사용하는 모든 VRRP 라우터는 동일한 VRRP 가상 MAC 주소를 공유합니다. 가상 MAC 주소는 가상 라우터의 주소 그룹 또는 VRID를 기준으로 계산됩니다(인터넷 표준 비트 순서의 16진 형식). 예를 들면 다음과 같습니다.

IPv4: 00-00-5E-00-01-{VRID}

IPv6: 00-00-5E-00-02-{VRID}

따라서 VRRP 라우터가 제대로 작동하려면 가상 MAC 주소를 사용하는 특수 VRRP VNIC를 먼저 만들어야 합니다. 이 VNIC에 있는 모든 IP 주소는 VRRP 라우터가 보호하는 가상 IP 주소로 간주됩니다. 이러한 가상 IP 주소는 백업 라우터에 있다가 라우터가 마스터 라우터가 될 때 가져오므로 이러한 가상 IP 주소에 높은 가용성이 제공됩니다.

## VRRP VNIC 만들기

VRRP VNIC를 만들 수 있도록 기존 `dladm create-vnic` 하위 명령이 확장되었습니다. 구문은 다음과 같습니다.

```
# dladm create-vnic [-t] [-R root-dir] [-l link] [-m vrrp -V VRID -A {inet | inet6}] [-v vlan-id] [-p prop=value[,...]] vnic-link
```

새 VNIC 주소 유형인 `vrrp`가 도입되었습니다. 이 새 VNIC 주소 유형을 갖는 VRID 및 주소 그룹을 지정해야 합니다.

그 결과 잘 알려진 가상 라우터 MAC 주소를 사용하는 VNIC가 생성됩니다.

## vrrpadm 구성

다음 절에서는 `vrrpadm` 하위 명령을 요약하여 보여줍니다. 자세한 내용은 [vrrpadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. `vrrpadm show-router` 하위 명령을 제외한 모든 하위 명령은 보존됩니다. 예를 들어 `vrrpadm create-router`로 생성된 VRRP 라우터는 재부트 이후에도 보존됩니다.

### vrrpadm create-router 하위 명령

`vrrpadm create-router` 하위 명령은 제공된 매개변수를 사용하여 지정된 VRID 및 주소 그룹의 VRRP 라우터를 만듭니다. 각 VRRP 라우터의 경우 특수 VRRP VNIC를 만들어야 하는데, VNIC는 `dladm create-vnic` 명령을 사용하여 만들 수 있습니다. 자세한 내용은 [vrrpadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 구문은 다음과 같습니다.

```
# vrrpadm create-router -V vrid -l link -A {inet | inet6} [-p \ priority] [-i adv-interval] [-o flags] router-name
```

`-o` 옵션은 VRRP 라우터의 선취 모드 및 승인 모드를 구성하는 데 사용됩니다. 값은 `preempt`, `un_preempt`, `accept`, `no_accept`일 수 있습니다. 기본적으로 두 모드가 모두 `true`로 설정되어 있습니다.

`router-name`은 이 VRRP 라우터의 고유 식별자로 사용되며, 다른 `vrrpadm` 하위 명령에 사용됩니다. 라우터 이름에 허용되는 문자는 영숫자(a-z, A-Z, 0-9) 및 밑줄('\_')입니다. 라우터 이름의 최대 길이는 31자입니다.

### vrrpadm modify-router 하위 명령

`vrrpadm modify-router` 하위 명령은 지정된 VRRP 라우터의 구성을 변경합니다. 구문은 다음과 같습니다.

```
# vrrpadm modify-router [-p priority] [-i adv-interval] [-o flags] \ router-name
```



## vrrpadm delete-router 하위 명령

vrrpadm delete-router 하위 명령은 지정된 VRRP 라우터를 삭제합니다. 구문은 다음과 같습니다.

```
# vrrpadm delete-router router-name
```

## vrrpadm disable-router 하위 명령

VRRP 라우터가 사용으로 설정할 때까지 작동하지 않습니다. 기본적으로 VRRP 라우터는 처음 만들 때 사용으로 설정됩니다. 그러나 일시적으로 VRRP 라우터를 사용 안함으로 설정하여 구성을 변경한 후 라우터를 다시 사용으로 설정하는 것이 유용할 때도 있습니다. 구문은 다음과 같습니다.

```
# vrrpadm disable-router router-name
```

## vrrpadm enable-router 하위 명령

사용 안함으로 설정된 VRRP 라우터는 enable-router 하위 명령을 사용하여 다시 사용으로 설정할 수 있습니다. 라우터를 사용으로 설정하는 경우, VRRP 라우터를 만들 때 경유하는 기본 데이터 링크(vrrpadm create-router로 라우터를 만들 때 -l 옵션을 사용하여 지정됨) 및 라우터의 VRRP VNIC가 존재해야 합니다. 그렇지 않으면 사용으로 설정 작업이 실패합니다. 구문은 다음과 같습니다.

```
# vrrpadm enable-router router-name
```

## vrrpadm show-router 하위 명령

vrrpadm show-router 하위 명령은 지정된 VRRP 라우터의 구성 및 상태를 표시합니다. 자세한 내용은 [vrrpadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 구문은 다음과 같습니다.

```
# vrrpadm show-router [-P | -x] [-p] [-o field[,...]] [router-name]
```

다음은 vrrpadm show-router 출력 예제입니다.

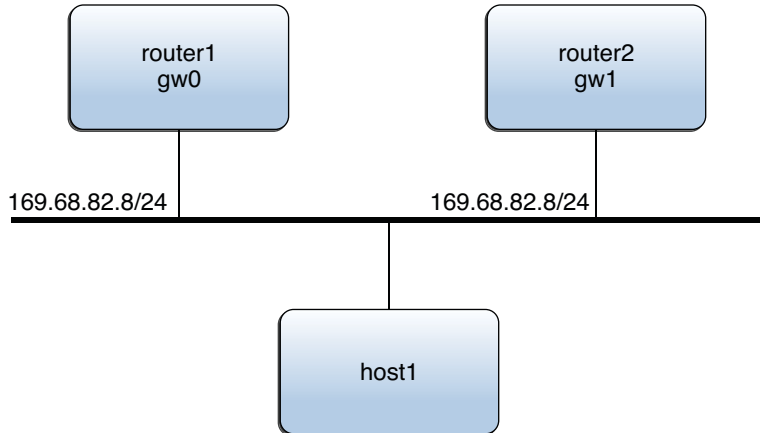
```
# vrrpadm show-router vrrp1
NAME VRID LINK AF PRIO ADV_INTV MODE STATE VNIC
vrrp1 1 bge1 IPv4 100 1000 e-pa- BACK vnic1
```

```
# vrrpadm show-router -x vrrp1
NAME STATE PRV_STAT STAT_LAST VNIC PRIMARY_IP VIRTUAL_IPS
vrrp1 BACK MAST 1m17s vnic1 10.0.0.100 10.0.0.1
```

```
# vrrpadm show-router -P vrrp1
NAME PEER P_PRIO P_INTV P_ADV_LAST M_DOWN_INTV
vrrp1 10.0.0.123 120 1000 0.313s 3609
```

## 예 25-1 VRRP 구성 예제

다음 그림은 일반 VRRP 구성을 보여줍니다.



이 예에서 IP 주소 169.68.82.8은 host1의 기본 게이트웨이로 구성됩니다. 이 IP 주소는 두 개의 VRRP 라우터( router1 및 router2)로 구성된 가상 라우터가 보호하는 가상 IP 주소입니다. 한 번에 두 라우터 중 하나만 마스터 라우터로 사용되어 가상 라우터의 책임을 맡으며 host1에서 보내는 패킷을 전달합니다.

가상 라우터의 VRID는 12라고 가정하며, 다음은 router1 및 router2에서 위의 VRRP 구성을 구성하는 데 사용되는 단계를 보여줍니다. router1은 가상 IP 주소 169.68.82.8의 소유자이며, 우선 순위는 기본값(255)입니다. router2는 우선 순위가 100인 백업입니다.

```

router1:
# dladm create-vnic -m vrrp -V 12 -A inet -l gw0 vnic1
# vrrpadm create-router -V 12 -A inet -l gw0 vrrp1
# ipadm create-addr -T static -d -a 169.68.82.8/24 vnic1/router1
# ipadm create-addr -T static -d -a 169.68.82.100/24 gw0/router1
# vrrpadm show-router -x vrrp1
NAME STATE PRV_STAT STAT_LAST VNIC PRIMARY_IP VIRTUAL_IPS
vrrp1 MAST BACK 1m17s vnic1 169.68.82.100 169.68.82.8
router2:
# dladm create-vnic -m vrrp -V 12 -A inet -l gw1 vnic1
# vrrpadm create-router -V 12 -A inet -l gw1 -p 100 vrrp1
# ipadm create-addr -T static -d -a 169.68.82.8/24 vnic1/router2
# ipadm create-addr -T static -d -a 169.68.82.101/24 gw0/router2
# vrrpadm show-router -x vrrp1
NAME STATE PRV_STAT STAT_LAST VNIC PRIMARY_IP VIRTUAL_IPS
vrrp1 BACK INIT 2m32s vnic1 169.68.82.101 169.68.82.8
  
```

router1의 구성을 예로 사용할 경우 gw0을 통해 적어도 하나의 IP 주소를 구성해야 합니다. 다음 예에서 라우터 1의 이 IP 주소는 기본 IP 주소로, VRRP 알림 패킷을 전송하는 데 사용됩니다.

예 25-1 VRRP 구성 예제 (계속)

```
# vrrpadm show-router -x vrrp1
NAME STATE PRV_STAT STAT_LAST VNIC PRIMARY_IP VIRTUAL_IPS
vrrp1 MAST BACK 1m17s vnic1 169.68.82.100 169.68.82.8
```

## 보안 고려 사항

새로운 `solaris.network.vrrp` 권한 부여가 도입되었으므로 VRRP 서비스를 구성해야 합니다. 읽기 전용 작업인 `vrrpadm showrouter`의 경우에는 이 권한 부여가 필요하지 않습니다.

`solaris.network.vrrp` 권한 부여는 Network Management 프로파일에 추가되었습니다.



## 혼잡 제어 구현

이 장에서는 Oracle Solaris에서 혼잡 제어를 구현하는 방법을 설명합니다. TCP 및 SCTP 트래픽의 혼잡을 막기 위해 제어를 설정합니다.

### 네트워크 혼잡 및 혼잡 제어

네트워크 혼잡은 일반적으로 노드가 네트워크 수용량보다 많은 패킷을 보낼 때 라우터 버퍼 오버플로우의 형태로 발생합니다. 다양한 알고리즘을 통해 전송 시스템에 제어를 설정하여 트래픽 혼잡을 막을 수 있습니다. 이러한 알고리즘은 Oracle Solaris에서 지원되며 운영 체제에 쉽게 추가하거나 직접 플러그인할 수 있습니다.

다음 표는 지원되는 알고리즘을 나열하고 설명합니다.

알고리즘	Oracle Solaris 이름	설명
NewReno	newreno	Oracle Solaris의 기본 알고리즘입니다. 제어 방식에는 발신자의 혼잡 윈도우, 느린 시작, 혼잡 회피가 포함됩니다.
HighSpeed	highspeed	고속 네트워크용으로 설계된, 가장 유명하고 가장 간단한 NewReno의 수정판 중 하나입니다.
CUBIC	cubic	현재 Linux 2.6의 기본 알고리즘입니다. 혼잡 회피 위상을 선형 윈도우 증가에서 3차 함수로 변경합니다.
Vegas	vegas	실제 패킷 손실을 트리거하지 않고 혼잡을 예측하려고 시도하는 고전적인 지연 기반 알고리즘입니다.

Oracle Solaris에서 다음과 같은 제어 관련 TCP 등록 정보를 설정하여 혼잡 제어를 사용으로 설정할 수 있습니다. 이러한 등록 정보는 TCP용으로 나열되지만 해당 등록 정보로 사용으로 설정되는 제어 방식은 SCTP 트래픽에도 적용됩니다.

- **cong\_enabled** - 현재 시스템에서 운영되는 알고리즘 목록을 콤마로 구분하여 나타냅니다. 알고리즘을 추가하거나 제거하여 원하는 알고리즘만 사용으로 설정할 수 있습니다.
- **cong\_default** - 응용 프로그램에서 소켓 옵션에 명시적으로 알고리즘을 지정하지 않을 때 기본적으로 사용되는 알고리즘입니다. 현재 **cong\_default** 등록 정보의 값은 전역 및 비전역 영역에 모두 적용됩니다.

이러한 등록 정보를 설정하려면 **ipadm set-prop** 명령을 사용합니다. 알고리즘을 추가하려면 += 수정자를 사용하고, 알고리즘을 제거하려면 -= 수정자를 사용할 수 있습니다.

## ▼ TCP 및 SCTP 네트워크 혼잡 제어를 구현하는 방법

### 1 관리자로 로그인합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스의 “관리 권한을 얻는 방법”**을 참조하십시오.

### 2 TCP 프로토콜의 혼잡 제어 등록 정보의 현재 값을 표시합니다.

```
# ipadm show-prop -p cong_enabled,cong_default tcp
```

등록 정보를 지정하지 않으면 모든 등록 정보가 표시됩니다.

명령은 현재 값뿐만 아니라 등록 정보에 지정 가능한 알고리즘도 표시합니다.

### 3 TCP 프로토콜의 혼잡 제어 등록 정보를 설정합니다.

```
# ipadm set-prop -p cong-ctrl-property+=algorithm tcp
```

설명

**cong-ctrl-property**      **cong\_enabled** 등록 정보 또는 **cong\_default** 등록 정보를 가리킵니다.

**algorithm**                등록 정보에 대해 설정 중인 알고리즘을 지정합니다. **ipadm show-prop** 명령의 출력에서 POSSIBLE 필드 머리글 아래에 나열된 알고리즘을 지정할 수 있습니다.

### 4 (옵션) 현재 사용으로 설정된 알고리즘을 제거합니다.

```
# ipadm set-prop -p cong-ctrl-property-=algorithm tcp
```

주 - 알고리즘을 추가/제거할 때 따라야 할 시퀀스 규칙은 없습니다. 다른 알고리즘을 등록 정보에 추가하기 전에 알고리즘을 제거할 수 있습니다. 그러나 `cong_default` 등록 정보에는 항상 정의된 알고리즘이 있어야 합니다.

## 5 (옵션) 혼잡 제어 등록 정보의 새 값을 표시합니다.

```
# ipadm show-prop -p cong_enabled,cong_default tcp
```

### 예 26-1 혼잡 제어용 알고리즘 설정

이 예에서는 TCP 프로토콜의 기본 알고리즘을 `newreno`에서 `cubic`으로 변경합니다. 또한 사용으로 설정된 알고리즘 목록에서 `vegas`를 제거합니다.

```
# ipadm show-prop -p cong_default,cong_enabled tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	cong_default	rw	newreno	--	newreno	-
tcp	cong_enabled	rw	newreno,cubic, highspeed, vegas	--	newreno	newreno,cubic, highspeed,vegas

```
# ipadm set-prop -p cong_enabled==vegas tcp
```

```
# ipadm set-prop -p cong_default=cubic tcp
```

```
# ipadm show-prop -p cong_default,cong_enabled tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	cong_default	rw	cubic	--	newreno	-
tcp	cong_enabled	rw	newreno,cubic, highspeed	--	newreno	newreno,cubic, highspeed,vegas





## 제 5 부

# IPQoS(IP Quality of Service)

이 파트에서는 Oracle Solaris의 차별화 서비스 구현인 IPQoS(IP Quality of Service) 관련 작업과 정보를 다룹니다.



## IPQoS 소개(개요)

---

IPQoS(IP Quality of Service)를 통해 계산 통계 우선 순위 지정, 제어 및 수집이 가능합니다. IPQoS를 사용하면 네트워크 사용자에게 일관된 레벨의 서비스를 제공할 수 있습니다. 또한 트래픽 관리를 통해 네트워크 정체를 피할 수 있습니다.

다음은 이 장에 포함된 항목 목록입니다.

- 387 페이지 “IPQoS 기본”
- 389 페이지 “IPQoS에서 QoS 제공”
- 391 페이지 “IPQoS를 사용하여 네트워크 효율성 향상”
- 392 페이지 “차별화 서비스 모델”
- 397 페이지 “IPQoS 사용 네트워크에서 트래픽 전달”

## IPQoS 기본

IPQoS는 IETF(Internet Engineering Task Force)의 Differentiated Services Working Group에서 정의한 Diffserv(차별화 서비스) 아키텍처를 사용으로 설정합니다. Oracle Solaris에서 IPQoS는 TCP/IP 프로토콜 스택의 IP 레벨에서 구현됩니다.

### 차별화 서비스란?

IPQoS를 사용으로 설정하면 선택한 고객 및 선택한 응용 프로그램에 대해 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 서로 다른 레벨의 서비스를 총칭하여 **차별화 서비스**라고 합니다. 고객에게 제공하는 차별화 서비스는 회사에서 고객에게 제공하는 서비스 레벨의 구조를 기준으로 할 수 있습니다. 또한 네트워크의 응용 프로그램이나 사용자에게 대해 설정된 우선 순위를 기준으로 차별화 서비스를 제공할 수 있습니다.

QoS 제공에는 다음 작업이 포함됩니다.

- 서로 다른 그룹(예: 고객 또는 엔터프라이즈의 부서)에 서비스 레벨 지정

- 특정 그룹이나 응용 프로그램에 제공되는 네트워크 서비스 우선 순위 지정
- 네트워크 병목 영역 및 기타 형태의 정체 발견 및 제거
- 네트워크 성능 모니터링 및 성능 통계 제공
- 네트워크 리소스 간의 대역폭 규제

## IPQoS 기능

IPQoS에는 다음과 같은 기능이 있습니다.

- QoS 정책 구성을 위한 `ipqosconf` 명령줄 도구
- 조직의 QoS 정책을 구성하는 필터를 기준으로 작업을 선택하는 분류기
- Diffserv 모델과 호환되어 네트워크 트래픽을 측정하는 측정 모듈
- 패킷의 IP 헤더를 전달 정보로 표시하는 기능을 기반으로 하는 서비스 차별화
- 트래픽 흐름에 대한 통계를 수집하는 흐름 계산 모듈
- UNIX® `kstat` 명령을 통해 트래픽 클래스에 대한 통계 수집
- SPARC® 및 x86 아키텍처 지원
- IPv4 및 IPv6 주소 지원
- IP 보안 아키텍처(IPsec)와 상호 운용성
- VLAN(virtual local area networks)에 대한 802.1D 사용자 우선 순위 표시 지원

## QoS(Quality-of-Service) 이론 및 실체에 대한 추가 정보를 얻을 수 있는 위치

차별화 서비스 및 QoS에 대한 정보는 서적 및 온라인 소스에서 찾을 수 있습니다.

### QoS 관련 서적

QoS 이론 및 실체에 대한 자세한 내용은 다음 서적을 참조하십시오.

- Ferguson, Paul 및 Geoff Huston. *Quality of Service*. John Wiley & Sons, Inc., 1998.
- Kilkki, Kalevi. *Differentiated Services for the Internet*. Macmillan Technical Publishing, 1999.

### QoS에 대한 RFC(Requests for Comments)

IPQoS는 다음 RFC 및 다음 인터넷 초안에 설명된 사양을 따릅니다.

- [RFC 2474, Definition of the Differentiated Services Field \(DS Field\) in the IPv4 and IPv6 Headers](http://www.ietf.org/rfc/rfc2474.txt?number=2474) (<http://www.ietf.org/rfc/rfc2474.txt?number=2474>) - 차별화 서비스 지원을 위한 IPv4 및 IPv6 패킷 헤더의 ToS(type of service) 필드 또는 DS 필드에 대한 향상된 기능을 설명합니다.
- [RFC 2475, An Architecture for Differentiated Services](http://www.ietf.org/rfc/rfc2475.txt?number=2475) (<http://www.ietf.org/rfc/rfc2475.txt?number=2475>) - Diffserv 구조의 구성 및 모듈에 대한 자세한 설명을 제공합니다.

- RFC 2597, Assured Forwarding PHB Group (<http://www.ietf.org/rfc/rfc2597.txt?number=2597>) – 혼별 AF(assured forwarding) 동작이 어떻게 작동하는지 설명합니다.
- RFC 2598, An Expedited Forwarding PHB (<http://www.ietf.org/rfc/rfc2598.txt?number=2598>) – 혼별 EF(expedited forwarding) 동작이 어떻게 작동하는지 설명합니다.
- 인터넷 초안, *An Informal Management Model for Diffserv Routers* – 라우터에서 Diffserv 아키텍처 구현을 위한 모델을 제공합니다.

## QoS 정보 웹 사이트

IETF의 Differentiated Services Working Group은 Diffserv 인터넷 초안에 대한 링크를 제공하는 웹 사이트(<http://www.ietf.org/html.charters/diffserv-charter.html>)를 유지 관리합니다.

라우터 제조업체(예: Cisco Systems 및 Juniper Networks)는 차별화 서비스가 자사 제품에서 어떻게 구현되는지 설명하는 자사 웹 사이트에서 정보를 제공합니다.

## IPQoS 매뉴얼 페이지

IPQoS 설명서에는 다음 매뉴얼 페이지가 포함되어 있습니다.

- [ipqosconf\(1M\)](#) - IPQoS 구성 파일 설정을 위한 명령을 설명합니다.
- [ipqos\(7ipp\)](#) - Diffserv 아키텍처 모델의 IPQoS 구현을 설명합니다.
- [ipgpc\(7ipp\)](#) - Diffserv 분류기의 IPQoS 구현을 설명합니다.
- [tokenmt\(7ipp\)](#) - IPQoS tokenmt 측정기를 설명합니다.
- [tswtclmt\(7ipp\)](#) - IPQoS tswtclmt 측정기를 설명합니다.
- [dscpmk\(7ipp\)](#) - DSCP 표시기 모듈을 설명합니다.
- [dlcosmk\(7ipp\)](#) - IPQoS 802.1D 사용자 우선 순위 표시기 모듈을 설명합니다.
- [flowacct\(7ipp\)](#) - IPQoS 흐름 계산 모듈을 설명합니다.
- [acctadm\(1M\)](#) - Oracle Solaris 확장 계산 기능을 구성하는 명령을 설명합니다. acctadm 명령에는 IPQoS 확장이 포함됩니다.

# IPQoS에서 QoS 제공

IPQoS 기능을 통해 인터넷 서비스 제공자(ISP) 및 응용 프로그램 서비스 제공자(ASP)는 고객에게 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 이러한 기능을 통해 개별 회사 및 교육 기관은 내부 조직이나 주요 응용 프로그램에 대한 서비스 우선 순위를 지정할 수 있습니다.

## 서비스 단계 계약 구현

조직이 ISP 또는 ASP인 경우 IPQoS 구성을 회사에서 고객에게 제공하는 **SLA(서비스 단계 계약)**를 기준으로 할 수 있습니다. SLA에서 서비스 제공자는 고객에게 가격 구조를 기준으로 특정 레벨의 네트워크 서비스를 보장합니다. 예를 들어, 높은 가격의 SLA는 고객이 모든 유형의 네트워크 트래픽에 대해 24시간 내내 가장 높은 우선 순위를 가지도록 보장할 수 있습니다. 반면, 중간 가격의 SLA는 고객이 업무 시간 중 전자 메일에 대해서만 높은 우선 순위를 가지도록 보장할 수 있습니다. 기타 모든 트래픽은 24시간 내내 중간 우선 순위를 가집니다.

## 개별 조직에 대해 QoS 보장

조직이 엔터프라이즈이거나 기관인 경우 해당 네트워크에 대한 QoS를 제공할 수도 있습니다. 특정 그룹이나 특정 응용 프로그램의 트래픽이 더 높거나 낮은 서비스 레벨을 가지도록 보장할 수 있습니다.

## QoS 정책 소개

*QoS(quality-of-service)* 정책을 정의하여 QoS를 구현합니다. QoS 정책은 고객 또는 응용 프로그램의 우선 순위 및 서로 다른 범주의 트래픽 처리를 위한 작업과 같이 다양한 네트워크 속성을 정의합니다. IPQoS 구성 파일에서 조직의 QoS 정책을 구현합니다. 이 파일은 Oracle Solaris 커널에 상주하는 IPQoS 모듈을 구성합니다. IPQoS 정책이 적용된 호스트는 **IPQoS 사용 시스템**으로 간주됩니다.

QoS 정책에서는 일반적으로 다음을 정의합니다.

- **서비스 클래스**라고 하는 고유의 네트워크 트래픽 그룹.
- 각 클래스에 대한 네트워크 트래픽의 양을 규제하기 위한 측정 단위. 이러한 측정 단위는 **측정**이라고 하는 트래픽 측정 프로세스를 제어합니다.
- IPQoS 시스템 및 Diffserv 라우터가 패킷 흐름에 적용해야 하는 작업. 이 유형의 작업을 **흡별 동작(PHB)**이라고 합니다.
- 조직이 서비스 클래스에 대해 필요로 하는 통계 수집. 예를 들면 고객이나 특정 응용 프로그램이 생성하는 트래픽입니다.

패킷이 네트워크에 전달될 때 IPQoS 사용 시스템은 패킷 헤더를 검사합니다. IPQoS 시스템이 수행하는 작업은 QoS 정책으로 결정됩니다.

QoS 정책 설계를 위한 작업은 **405 페이지 “서비스 품질 정책 계획”**에 설명되어 있습니다.

## IPQoS를 사용하여 네트워크 효율성 향상

IPQoS에는 QoS를 구현할 때 네트워크 성능을 더욱 효율화할 수 있는 기능이 포함되어 있습니다. 컴퓨터 네트워크가 확장되면 사용자 수 및 더욱 강력한 프로세서 증가로 생성되는 네트워크 트래픽 관리의 필요성도 높아집니다. 과다 사용되는 네트워크의 증상으로는 데이터 손실 및 트래픽 정체를 들 수 있습니다. 두 증상은 모두 느린 응답 시간이라는 결과를 초래합니다.

과거에는 시스템 관리자가 더 많은 대역폭을 추가하여 네트워크 트래픽 문제를 처리했습니다. 링크에서 트래픽 레벨은 광범위하게 변동하는 경우가 많습니다. IPQoS를 사용하면 기존 네트워크의 트래픽을 관리하고 확장이 필요한지 여부 및 필요한 위치를 평가할 수 있습니다.

예를 들어, 엔터프라이즈나 기관의 경우 트래픽 병목 현상을 피하려면 효율적인 네트워크를 유지 관리해야 합니다. 또한 그룹이나 응용 프로그램에서 할당된 대역폭보다 많이 소비하지 않도록 해야 합니다. ISP 또는 ASP의 경우, 고객이 지불한 레벨의 네트워크 서비스를 받도록 네트워크 성능을 관리해야 합니다.

## 대역폭이 네트워크 트래픽에 미치는 영향

IPQoS를 사용하여 네트워크 **대역폭**(완전히 사용된 네트워크 링크나 장치에서 전송할 수 있는 최대 데이터 양)을 규제할 수 있습니다. QoS 정책에서 대역폭 사용 우선 순위를 지정하여 고객이나 사용자에게 QoS를 제공해야 합니다. IPQoS 측정 모듈을 통해 IPQoS 사용 호스트에서 다양한 트래픽 클래스 간에 대역폭 할당을 측정하고 제어할 수 있습니다.

네트워크의 트래픽을 효과적으로 관리할 수 있으려면 먼저 대역폭 사용에 대한 다음 질문에 답해야 합니다.

- 귀사의 로컬 네트워크에서 트래픽 문제가 있는 영역은 어디입니까?
- 사용 가능한 대역폭을 최대한 사용하기 위해 무엇을 해야 합니까?
- 우선 순위가 가장 높은 사이트의 중요 응용 프로그램은 무엇입니까?
- 정체에 민감한 응용 프로그램은 무엇입니까?
- 낮은 우선 순위로 지정해도 되는 덜 중요한 응용 프로그램은 무엇입니까?

## 서비스 클래스를 사용하여 트래픽 우선 순위 지정

QoS를 구현하기 위해 네트워크 트래픽을 분석하여 트래픽을 나눌 수 있는 대략적인 그룹을 결정합니다. 그런 다음 여러 그룹을 개별 특성 및 개별 우선 순위를 가지는 서비스 클래스로 조직합니다. 이러한 클래스는 조직에 대한 QoS 정책의 기준이 되는 기본 범주를 형성합니다. 서비스 클래스는 제어할 트래픽 그룹을 나타냅니다.

예를 들어, 제공자는 계단식 가격 구조로 프리미엄, 골드, 실버 및 브론즈 레벨의 서비스를 제공할 수 있습니다. 프리미엄 SLA는 ISP가 고객을 위해 호스트하는 웹

사이트를 대상으로 한 수신 트래픽에 대해 가장 높은 우선 순위를 보장합니다. 따라서 고객의 웹 사이트에 대한 수신 트래픽이 하나의 트래픽 클래스가 될 수 있습니다.

엔터프라이즈의 경우, 부서 요구 사항을 기준으로 서비스 클래스를 만들 수 있습니다. 또는 네트워크 트래픽에서 특정 응용 프로그램의 수를 기준으로 클래스를 만들 수 있습니다. 다음은 엔터프라이즈에 대한 트래픽 클래스의 몇 가지 예입니다.

- 특정 서버에 대한 전자 메일 및 나가는 FTP와 같이 자주 사용되는 응용 프로그램(둘 다 하나의 클래스가 될 수 있음). 직원들은 이러한 응용 프로그램을 지속적으로 사용하므로 QoS 정책에서 전자 메일 및 나가는 FTP에 대해 적은 양의 대역폭과 낮은 우선 순위를 보장할 수 있습니다.
- 하루 24시간 실행되어야 하는 주문 입력 데이터베이스. 엔터프라이즈에 대한 데이터베이스 응용 프로그램의 중요도에 따라 데이터베이스에 많은 양의 대역폭과 높은 우선 순위를 지정할 수 있습니다.
- 중요한 업무 또는 민감한 업무를 수행하는 부서(예: 급여 부서). 조직에 대한 부서의 중요도에 따라 해당 부서에 지정하는 우선 순위 및 대역폭의 양이 결정됩니다.
- 회사의 외부 웹 사이트로 들어오는 호출. 이 클래스에는 낮은 우선 순위로 실행되는 적당한 양의 대역폭을 지정할 수 있습니다.

## 차별화 서비스 모델

IPQoS에는 RFC 2475에서 정의된 **차별화 서비스(Diffserv)** 아키텍처의 일부인 다음 모듈이 포함됩니다.

- 분류기
- 측정기
- 표시기

IPQoS는 Diffserv 모델에 다음 향상된 기능을 추가합니다.

- 흐름 계산 모듈
- 802.1D 데이터그램 표시기

이 절에서는 IPQoS에서 사용되는 Diffserv 모듈을 소개합니다. QoS 정책을 설정하려면 이러한 모듈, 이름 및 용도에 대해 알고 있어야 합니다. 각 모듈에 대한 자세한 내용은 [461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”](#)을 참조하십시오.

## 분류기(ipgpc) 개요

Diffserv 모델에서 **분류기**는 네트워크 트래픽 흐름에서 패킷을 선택합니다. **트래픽 흐름**은 다음 IP 헤더 필드에서 동일한 정보를 가지는 패킷 그룹을 구성합니다.

- 소스 주소
- 대상 주소



- 소스 포트
- 대상 포트
- 프로토콜 번호

IPQoS에서 이러한 필드를 5-튜플이라고 합니다.

IPQoS 분류기 모듈의 이름은 `ipgpc`로 지정됩니다. `ipgpc` 분류기는 트래픽 흐름을 IPQoS 구성 파일에서 구성하는 특성을 기준으로 클래스로 분류합니다.

`ipgpc`에 대한 자세한 내용은 [461 페이지 “분류기 모듈”](#)을 참조하십시오.

## IPQoS 클래스

클래스는 유사한 특성을 공유하는 네트워크 흐름의 그룹입니다. 예를 들어, ISP는 고객에게 제공하는 서로 다른 서비스 레벨을 나타내기 위해 클래스를 정의할 수 있습니다. ASP는 다양한 응용 프로그램에 서로 다른 서비스 레벨을 제공하는 SLA를 정의할 수 있습니다. ASP QoS 정책의 경우, 클래스에는 특정 대상 IP 주소로 향하는 FTP 트래픽이 포함될 수 있습니다. 회사의 외부 웹 사이트의 송신 트래픽도 클래스로 정의될 수 있습니다.

트래픽을 클래스로 그룹화하는 것은 QoS 정책 계획에서 큰 부분을 차지합니다. `ipqosconf` 유틸리티를 사용하여 클래스를 만드는 경우 실제로는 `ipgpc` 분류기를 구성하는 것입니다.

클래스를 정의하는 방법에 대한 자세한 내용은 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#)을 참조하십시오.

## IPQoS 필터

필터는 선택기라는 매개변수를 포함하는 규칙 집합입니다. 각 필터는 클래스를 가리켜야 합니다. IPQoS는 패킷을 각 필터의 선택기에 대해 일치시켜 패킷이 해당 필터의 클래스에 속하는지 여부를 결정합니다. IPQoS 5-튜플 및 기타 공통 매개변수 등의 다양한 선택기를 사용하여 패킷을 필터링할 수 있습니다.

- 소스 주소 및 대상 주소
- 소스 포트 및 대상 포트
- 프로토콜 번호
- 사용자 ID
- 프로젝트 ID
- 차별화 서비스 코드 포인트(DSCP)
- 인터페이스 인덱스

예를 들어, 단순 필터에는 값이 80인 대상 포트가 포함될 수 있습니다. 그런 다음 `ipgpc` 분류기는 대상 포트 80(HTTP)으로 향하는 모든 패킷을 선택하고 QoS 정책에서 정의된 대로 패킷을 처리합니다.

필터 만들기에 대한 자세한 내용은 [410 페이지 “QoS 정책에서 필터를 정의하는 방법”](#)을 참조하십시오.

## 측정기(tokenmt 및 tswtclmt) 개요

Diffserv 모델에서 **측정기**는 클래스별 기준으로 트래픽 흐름의 전송 속도를 추적합니다. 측정기는 흐름의 실제 속도가 구성된 속도를 얼마나 준수하는지 평가하여 해당하는 결과를 결정합니다. 트래픽 흐름의 결과를 기준으로 측정기는 후속 작업을 선택합니다. 후속 작업에는 다른 작업으로 패킷 보내기 또는 추가 처리 없이 네트워크로 패킷 돌려보내기가 포함될 수 있습니다.

IPQoS 측정기는 네트워크 흐름이 QoS 정책에서 해당 클래스에 대해 정의된 전송 속도를 준수하는지 여부를 결정합니다. IPQoS에는 두 측정 모듈이 포함됩니다.

- tokenmt – 두 토큰 버킷 측정 체계를 사용합니다.
- tswtclmt – 시간별 창 측정 체계를 사용합니다.

두 측정 모듈은 모두 빨간색, 노란색 및 녹색의 세 가지 결과를 인식합니다.

red\_action\_name, yellow\_action\_name 및 green\_action\_name 매개변수에서 각 결과에 대해 수행할 작업을 정의합니다.

또한 tokenmt가 색상을 인식하도록 구성할 수 있습니다. 색상 인식 측정 인스턴스에서는 패킷의 크기, DSCP, 트래픽 속도 및 구성된 매개변수를 사용하여 결과를 결정합니다. 측정기는 DSCP를 사용하여 패킷의 결과를 녹색, 노란색 또는 빨간색으로 매핑합니다.

IPQoS 측정기의 매개변수 정의에 대한 자세한 내용은 [411 페이지 “흐름 제어 계획 방법”](#)을 참조하십시오.

## 표시기(dscpmk 및 dlcosmk) 개요

Diffserv 모델에서 **표시기**는 패킷을 전달 동작이 반영된 값으로 표시합니다. 표시는 패킷의 헤더에 값을 두어 패킷을 네트워크에 어떻게 전달할지 나타내는 프로세스입니다. IPQoS에는 두 표시기 모듈이 포함됩니다.

- dscpmk – IP 패킷 헤더의 DS 필드를 **차별화 서비스 코드 포인트** 또는 *DSCP*라는 숫자 값으로 표시합니다. 그러면 Diffserv 인식 라우터에서 DS 코드 포인트를 사용하여 알맞은 전달 동작을 패킷에 적용할 수 있습니다.
- dlcosmk – 이더넷 프레임 헤더의 VLAN(virtual local area network) 태그를 **사용자 우선 순위**라는 숫자 값으로 표시합니다. 사용자 우선 순위는 데이터그램에 적용할 알맞은 전달 동작을 정의하는 *CoS(서비스 클래스)*를 나타냅니다.

dlcosmk는 IETF에서 설계한 Diffserv 모델의 일부가 아닌 IPQoS 추가 기능입니다.

QoS 정책의 표시기 전략 구현에 대한 자세한 내용은 [414 페이지 “전달 동작 계획 방법”](#)을 참조하십시오.

## 흐름 계산(flowacct) 개요

IPQoS는 flowacct 계산 모듈을 Diffserv 모델에 추가합니다. flowacct를 사용하여 트래픽 흐름에 대한 통계를 수집하고 해당 SLA에 따라 고객에게 청구할 수 있습니다. 흐름 계산은 용량 계획 및 시스템 모니터링에도 유용합니다.

flowacct 모듈은 acctadm 명령과 함께 작동하여 계산 로그 파일을 만듭니다. 기본 로그에는 다음 목록에 나온 대로 IPQoS 5-튜플 및 두 가지 추가 속성이 포함됩니다.

- 소스 주소
- 소스 포트
- 대상 주소
- 대상 포트
- 프로토콜 번호
- 패킷 수
- 바이트 수

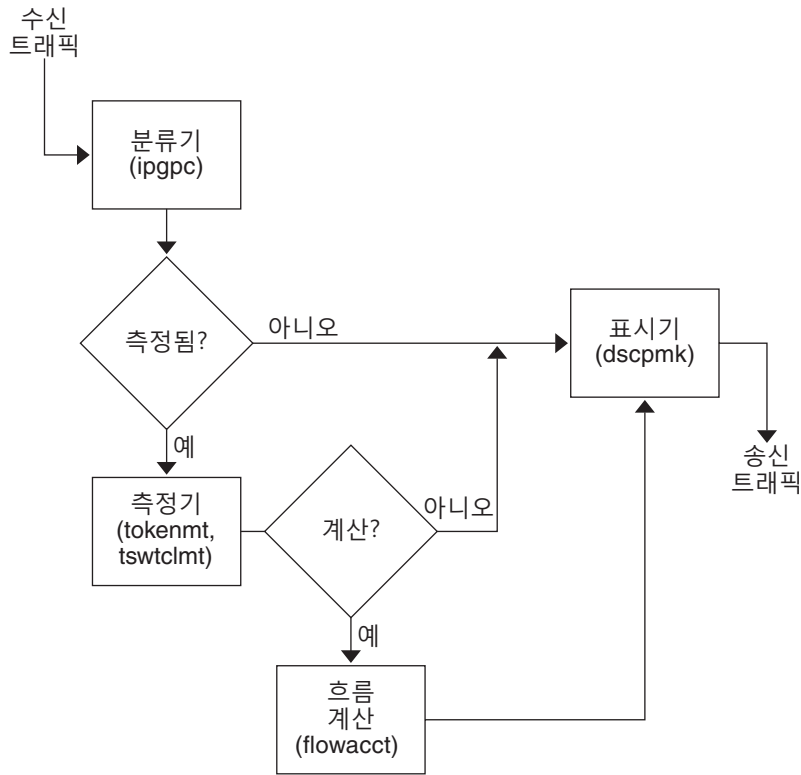
또한 [456 페이지](#) “트래픽 흐름에 대한 정보 기록”과 flowacct(7ipp) 및 acctadm(1M) 매뉴얼 페이지에 설명된 대로 다른 속성에 대한 통계도 수집할 수 있습니다.

흐름 계산 전략 계획에 대한 자세한 내용은 [416 페이지](#) “흐름 계산 계획 방법”을 참조하십시오.

## IPQoS 모듈을 통한 트래픽 흐름 방식

다음 그림은 수신 트래픽이 몇 가지 IPQoS 모듈을 통과할 수 있는 경로를 보여줍니다.

그림 27-1 Diffserv 모델의 IPQoS 구현을 통한 트래픽 흐름



이 그림은 IPQoS 사용 시스템에서 일반적인 트래픽 흐름 시퀀스를 보여줍니다.

1. 분류기는 패킷 스트림에서 시스템 QoS 정책의 필터링 조건과 일치하는 모든 패킷을 선택합니다.
2. 그런 다음 선택된 패킷은 수행할 다음 작업에 대해 평가됩니다.
3. 분류기는 흐름 제어가 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
4. 흐름 제어가 필요한 트래픽은 측정기로 보내집니다.
5. 측정기는 구성된 속도를 적용합니다. 그런 다음 측정기는 트래픽 준수 값을 흐름 제어 패킷에 지정합니다.
6. 그런 다음 흐름 제어 패킷은 평가되어 패킷에 계산이 필요한지 여부를 결정합니다.
7. 측정기는 흐름 계산이 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
8. 흐름 계산 모듈은 수신된 패킷에 대한 통계를 수집합니다. 그런 다음 모듈은 패킷을 표시기로 보냅니다.
9. 표시기는 DS 코드 포인트를 패킷 헤더에 지정합니다. 이 DSCP는 Diffserv 인식 시스템에서 패킷에 적용해야 하는 흐름 동작을 나타냅니다.

# IPQoS 사용 네트워크에서 트래픽 전달

이 절에서는 IPQoS 사용 네트워크에서 패킷 전달과 관련된 요소를 소개합니다. IPQoS 사용 시스템은 네트워크 스트림에서 시스템의 IP 주소를 대상으로 가지는 모든 패킷을 처리합니다. 그런 다음 IPQoS 시스템은 QoS 정책을 패킷에 적용하여 차별화 서비스를 설정합니다.

## DS 코드 포인트

DSCP(DS 코드 포인트)는 패킷 헤더에서 Diffserv 인식 시스템이 표시된 패킷에 대해 수행해야 하는 작업을 정의합니다. Diffserv 아키텍처는 사용할 IPQoS 사용 시스템 및 Diffserv 라우터에 대한 DS 코드 포인트 집합을 정의합니다. 또한 Diffserv 아키텍처는 DSCP와 일치하는 **전달 동작**이라는 작업 집합을 정의합니다. IPQoS 사용 시스템은 패킷 헤더에서 DS 필드의 우선권 비트를 DSCP로 표시합니다. 라우터가 DSCP 값이 있는 패킷을 수신하면 라우터는 해당 DSCP와 연결된 전달 동작을 적용합니다. 그런 다음 패킷은 네트워크로 보내집니다.

---

주 - dlcsmk 표시기는 DSCP를 사용하지 않습니다. 대신 dlcsmk가 이더넷 프레임 헤더를 CoS 값으로 표시합니다. VLAN 장치를 사용하는 네트워크에서 IPQoS를 구성하려는 경우 [466 페이지](#) “**표시기 모듈**”을 참조하십시오.

---

## 홉별 동작

Diffserv 용어에서 DSCP에 지정된 전달 동작을 **PHB(홉별 동작)**이라고 합니다. PHB는 Diffserv 인식 시스템에서 다른 트래픽과 관련하여 표시된 패킷이 수신하는 전달 우선권을 정의합니다. 이 우선권은 최종적으로 IPQoS 사용 시스템이나 Diffserv 라우터가 표시된 패킷을 전달할지 또는 삭제할지 결정합니다. 전달되는 패킷의 경우, 대상으로 향하는 경로에서 패킷이 만나는 각 Diffserv 라우터는 동일한 PHB를 적용합니다. 다른 Diffserv 시스템이 DSCP를 변경할 경우는 예외입니다. PHB에 대한 자세한 내용은 [466 페이지](#) “**패킷 전달을 위해 dscpmk 표시기 사용**”을 참조하십시오.

PHB의 목적은 지정된 양의 네트워크 리소스를 인접 네트워크의 트래픽 클래스에 제공하는 것입니다. QoS 정책에서 이 목적을 달성할 수 있습니다. 트래픽 흐름이 IPQoS 사용 시스템을 떠날 때 트래픽 클래스에 대한 우선권 레벨을 나타내는 DSCP를 정의합니다. 우선권은 높은 우선권/낮은 삭제 가능성에서 낮은 우선권/높은 삭제 가능성의 범위에 있을 수 있습니다.

예를 들어, QoS 정책은 한 트래픽 클래스에 낮은 삭제 가능성의 PHB를 보장하는 DSCP를 지정할 수 있습니다. 그러면 이 트래픽 클래스는 Diffserv 인식 라우터에서 이 클래스의 패킷에 대역폭을 보장하는 낮은 삭제 우선권의 PHB를 수신합니다. 다양한 레벨의 우선권을 다른 트래픽 클래스에 지정하는 다른 DSCP를 QoS 정책에 추가할 수 있습니다. 낮은 우선권의 패킷에는 패킷의 DSCP에 표시된 우선 순위에 따라 Diffserv 시스템에서 대역폭을 제공합니다.

IPQoS는 Diffserv 아키텍처에서 정의된 빠른 전달 및 보장 전달의 두 가지 전달 동작 유형을 지원합니다.

## 빠른 전달

**EF(빠른 전달)** 홉별 동작은 EF 관련 DSCP를 가진 트래픽 클래스에 가장 높은 우선 순위가 부여되도록 합니다. EF DSCP를 가진 트래픽은 대기열에 두지 않습니다. EF는 낮은 손실, 대기 시간 및 지터를 제공합니다. EF에 권장되는 DSCP는 101110입니다. 101110으로 표시된 패킷은 대상으로 향하는 경로에서 Diffserv 인식 네트워크를 통과할 때 보장된 낮은 삭제 우선권을 받습니다. 프리미엄 SLA의 고객이나 응용 프로그램에 우선 순위를 지정할 때 EF DSCP를 사용합니다.

## 보장 전달

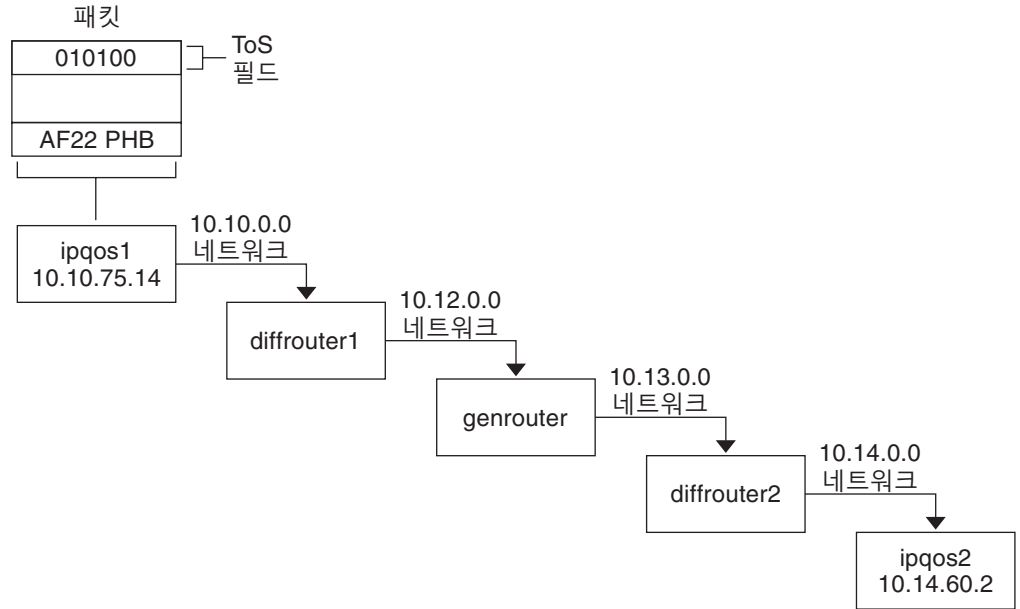
**AF(보장 전달)** 홉별 동작은 패킷에 지정할 수 있는 네 가지 서로 다른 전달 클래스를 제공합니다. 모든 전달 클래스는 [표 32-2](#)에 나온 대로 세 가지 삭제 우선권을 제공합니다.

다양한 AF 코드 포인트는 고객 및 응용 프로그램에 서로 다른 레벨의 서비스를 지정할 수 있는 기능을 제공합니다. QoS 정책에서는 QoS 정책을 계획할 때 네트워크에서 트래픽 및 서비스의 우선 순위를 지정할 수 있습니다. 그런 다음 서로 다른 AF 레벨을 우선 순위가 지정된 트래픽에 지정할 수 있습니다.

## Diffserv 환경에서 패킷 전달

다음 그림은 부분적으로 Diffserv 사용 환경을 갖춘 회사 인트라넷의 일부를 보여줍니다. 이 시나리오에서 10.10.0.0 및 10.14.0.0 네트워크의 모든 호스트는 IPQoS가 사용으로 설정되어 있고, 두 네트워크의 로컬 라우터는 Diffserv를 인식합니다. 하지만 임시 네트워크는 Diffserv에 대해 구성되지 않았습니다.

그림 27-2 Diffserv 인식 네트워크 홉에서 패킷 전달



다음 단계에서는 이 그림에 표시된 패킷의 흐름을 추적합니다. 단계는 ipqos1 호스트에서 발생하는 패킷의 진행부터 시작됩니다. 그런 다음 단계는 여러 홉을 거쳐 ipqos2 호스트로 계속됩니다.

1. ipqos1의 사용자는 ftp 명령을 실행하여 세 홉 떨어진 ipqos2 호스트에 액세스합니다.
2. ipqos1은 QoS 정책을 결과 패킷 흐름에 적용합니다. 그런 다음 ipqos1은 ftp 트래픽을 성공적으로 분류합니다.

시스템 관리자는 로컬 네트워크 10.10.0.0에서 발생하는 모든 나가는 ftp 트래픽에 대한 클래스를 만들었습니다. ftp 클래스에 대한 트래픽에는 클래스 2인 중간 삭제 우선권의 AF22 홉별 동작이 지정되었습니다. ftp 클래스에 대해서는 2Mb/초의 트래픽 흐름 속도가 구성되었습니다.

3. ipqos-1은 ftp 흐름을 측정하여 흐름이 2Mb/초의 약정된 속도를 초과하는지 여부를 결정합니다.
4. ipqos1의 표시기는 나가는 ftp 패킷의 DS 필드를 AF22 PHB와 일치하는 010100 DSCP로 표시합니다.
5. diffrouter1 라우터는 ftp 패킷을 수신합니다. 그런 다음 diffrouter1은 DSCP를 검사합니다. diffrouter1가 정채된 경우 AF22로 표시된 패킷은 삭제됩니다.
6. ftp 트래픽은 diffrouter1의 파일에서 AF22에 대해 구성된 홉별 동작에 따라 다음 홉으로 전달됩니다.
7. ftp 트래픽은 10.12.0.0 네트워크를 통과하여 Diffserv를 인식하지 못하는 genrouter로 이동합니다. 결과적으로 트래픽은 “최선 조건” 전달 동작을 수신합니다.

8. genrouter는 ftp 트래픽을 10.13.0.0 네트워크에 전달합니다. 여기에서 트래픽은 diffrouter2로 수신됩니다.
9. diffrouter2는 Diffserv를 인식합니다. 따라서 라우터는 AF22 패킷에 대한 라우터 정책에서 정의된 PHB에 따라 ftp 패킷을 네트워크에 전달합니다.
10. ipqos2는 ftp 트래픽을 수신합니다. 그런 다음 ipqos2는 ipqos1의 사용자에게 사용자 이름과 암호를 물어봅니다.



## IPQoS 사용 네트워크 계획(작업)

Oracle Solaris를 실행하는 모든 시스템에서 IPQoS를 구성할 수 있습니다. 그러면 IPQoS 시스템은 Diffserv 인식 라우터와 함께 작동하여 인트라넷에서 차별화된 서비스와 트래픽 관리를 제공합니다.

이 장에는 Diffserv 인식 네트워크에 IPQoS 사용 시스템을 추가하는 계획 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

- 401 페이지 “일반 IPQoS 구성 계획(작업 맵)”
- 402 페이지 “Diffserv 네트워크 토폴로지 계획”
- 405 페이지 “서비스 품질 정책 계획”
- 406 페이지 “QoS 정책 계획(작업 맵)”
- 417 페이지 “IPQoS 구성 예 소개”

### 일반 IPQoS 구성 계획(작업 맵)

네트워크에서 IPQoS를 비롯하여 차별화된 서비스를 구현하려면 광범위한 계획이 필요합니다. 각 IPQoS 사용 시스템의 위치 및 기능뿐 아니라 각 시스템과 로컬 네트워크에 있는 라우터의 관계도 고려해야 합니다. 다음 작업 맵에서는 네트워크에서 IPQoS를 구현하는 주요 계획 작업을 나열하고 작업을 완료하는 데 필요한 절차와 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 시스템을 통합하는 Diffserv 네트워크 토폴로지를 계획합니다.	다양한 Diffserv 네트워크 토폴로지를 조사하여 사이트에 가장 적합한 솔루션을 결정합니다.	402 페이지 “Diffserv 네트워크 토폴로지 계획”.
2. IPQoS 시스템이 제공할 다양한 유형의 서비스를 계획합니다.	네트워크가 제공하는 서비스의 유형을 SLA(서비스 단계 계약)별로 구성합니다.	405 페이지 “서비스 품질 정책 계획”.

작업	설명	수행 방법
3. 각 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	각 SLA 구현에 필요한 클래스, 측정 및 계산 기능을 결정합니다.	405 페이지 “서비스 품질 정책 계획”.
4. 해당하는 경우 Diffserv 라우터에 대한 정책을 계획합니다.	IPQoS 시스템에서 사용되는 Diffserv 라우터에 대한 일정 잡기 및 대기열 지정 정책을 결정합니다.	대기열 지정 및 일정 잡기 정책은 라우터 설명서를 참조하십시오.

## Diffserv 네트워크 토폴로지 계획

네트워크에 대해 차별화된 서비스를 제공하려면 하나 이상의 IPQoS 사용 시스템 및 Diffserv 인식 라우터가 필요합니다. 이 절에 설명된 다양한 방법으로 이와 같은 기본 시나리오를 확장할 수 있습니다.

### Diffserv 네트워크에 대한 하드웨어 전략

일반적으로 고객은 서버 및 서버 통합(예: Oracle의 Sun Enterprise™ 서버)에서 IPQoS를 실행합니다. 반대로 네트워크 요구 사항에 따라 데스크탑 시스템(예: UltraSPARC® 시스템)에서도 IPQoS를 실행할 수 있습니다. 다음 목록에서는 IPQoS 구성이 가능한 시스템에 대해 설명합니다.

- 웹 서버, 데이터베이스 서버 등의 다양한 서비스를 제공하는 Oracle Solaris 시스템
- 전자 메일, FTP 또는 기타 많이 사용되는 네트워크 응용 프로그램을 제공하는 애플리케이션 서버
- 웹 캐시 서버 또는 프록시 서버
- Diffserv 인식 로드 밸런서가 관리하는 IPQoS 사용 서버 팜의 네트워크
- 단일 이기종 네트워크에 대한 트래픽을 관리하는 방화벽
- 가상 근거리 통신망(LAN)의 일부인 IPQoS 시스템

Diffserv 인식 라우터가 이미 작동되고 있는 네트워크 토폴로지에 IPQoS 시스템을 도입할 수 있습니다. 라우터가 현재 Diffserv를 제공하지 않을 경우 Cisco Systems, Juniper Networks 및 기타 라우터 제조업체에서 제공하는 Diffserv 솔루션을 고려해 보십시오. 로컬 라우터가 Diffserv를 구현하지 않은 경우 라우터는 표시를 평가하지 않은 상태로 표시된 패킷을 다음 홉으로 전달합니다.

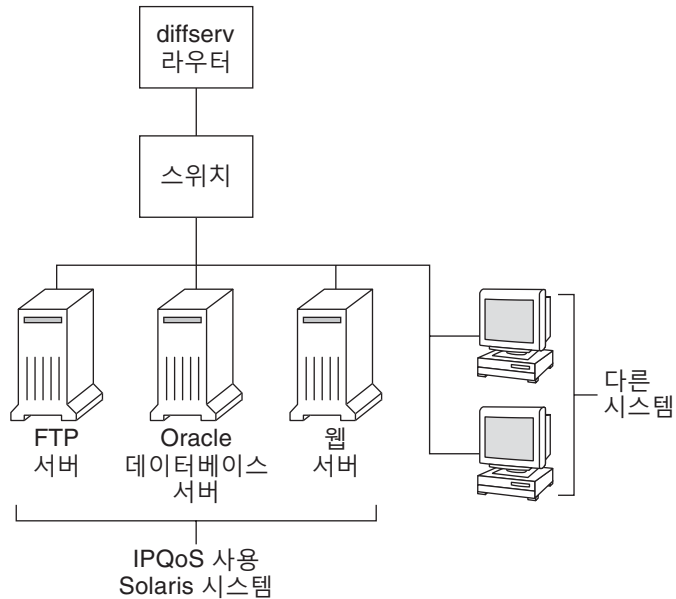
### IPQoS 네트워크 토폴로지

이 절에서는 다양한 네트워크 요구 사항에 대한 IPQoS 전략에 대해 설명합니다.

## 개별 호스트의 IPQoS

다음 그림에서는 IPQoS 사용 시스템의 단일 네트워크를 보여 줍니다.

그림 28-1 네트워크 세그먼트의 IPQoS 시스템



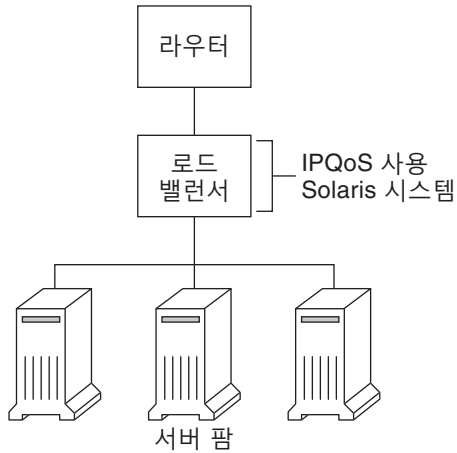
이 네트워크는 회사 인트라넷의 유일한 세그먼트입니다. 애플리케이션 서버 및 웹 서버에서 IPQoS를 사용으로 설정하면 각 IPQoS 시스템이 송신 트래픽을 릴리스하는 속도를 제어할 수 있습니다. 라우터가 Diffserv를 인식하도록 설정할 경우 추가로 수신 및 송신 트래픽을 제어할 수 있습니다.

본 설명서의 예에는 “개별 호스트의 IPQoS” 시나리오가 사용됩니다. 설명서 전체에서 사용되는 토폴로지 예는 [그림 28-4](#)를 참조하십시오.

## 서버 팜 네트워크의 IPQoS

다음 그림에서는 이기종 서버 팜이 여러 개인 네트워크를 보여 줍니다.

그림 28-2 IPQoS 사용 서버 팜의 네트워크



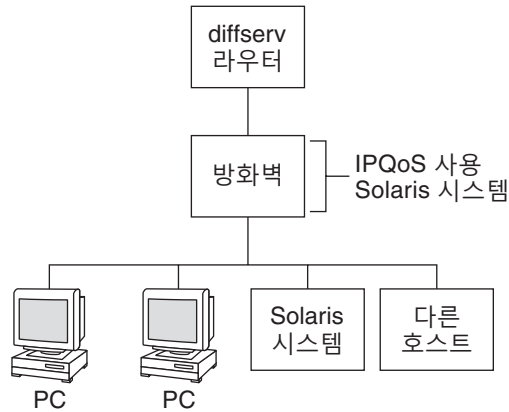
해당 토폴로지에서는 라우터가 Diffserv를 인식하므로 수신 트래픽과 송신 트래픽을 모두 대기열에 지정하고 속도를 제어할 수 있습니다. 로드 밸런서도 Diffserv를 인식하며 서버 팜에서 IPQoS가 사용됩니다. 로드 밸런서가 사용자 ID, 프로젝트 ID 등의 선택기를 사용하여 라우터 이외의 추가 필터링을 제공할 수 있습니다. 이러한 선택기는 응용 프로그램 데이터에 포함되어 있습니다.

이 시나리오는 로컬 네트워크의 혼잡을 관리할 수 있도록 흐름 제어 및 트래픽 전달을 제공합니다. 또한 이 시나리오는 서버 팜의 송신 트래픽으로 인해 인트라넷의 다른 부분이 과부화되지 않도록 합니다.

## 방화벽의 IPQoS

다음 그림에서는 방화벽을 통해 다른 세그먼트로부터 보호되는 회사 네트워크 세그먼트를 보여 줍니다.

그림 28-3 IPQoS 사용 방화벽으로 보호되는 네트워크



이 시나리오에서 트래픽은 패킷이 필터링되고 대기열에 지정되는 Diffserv 인식 라우터로 들어옵니다. 라우터가 전달한 모든 수신 트래픽은 IPQoS 사용 방화벽을 통과합니다. IPQoS를 사용하려면 방화벽이 IP 전달 스택을 무시하지 않아야 합니다.

방화벽의 보안 정책에 따라 수신 트래픽이 내부 네트워크로 들어오거나 나갈 수 있는 여부가 결정됩니다. QoS 정책은 방화벽을 통과한 수신 트래픽의 서비스 레벨을 제어합니다. QoS 정책에 따라 송신 트래픽에 전달 동작을 표시할 수도 있습니다.

## 서비스 품질 정책 계획

서비스 품질(QoS) 정책을 계획할 때는 네트워크가 제공하는 서비스를 검토 및 분류하고 우선 순위를 설정해야 합니다. 또한 사용 가능한 대역폭을 평가하여 각 트래픽 클래스가 네트워크로 릴리스되는 속도를 결정해야 합니다.

## QoS 정책 계획 지원

IPQoS 구성 파일에 필요한 정보를 포함하는 형식으로 QoS 정책을 계획하기 위한 정보를 수집합니다. 예를 들어, 다음 템플리트를 사용하여 IPQoS 구성 파일에서 사용될 주요 정보 범주를 나열할 수 있습니다.

표 28-1 QoS 계획 템플리트

클래스	우선 순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 1 필터 3	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	흐름 계산 통계 필요

표 28-1 QoS 계획 템플리트 (계속)

클래스	우선 순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음
클래스 2	2	필터 1	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	흐름 계산 통계 필요
클래스 2	2	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음

각 주요 범주를 구분하여 추가로 QoS 정책을 정의할 수 있습니다. 후속 절에서는 템플리트에 표시되는 범주에 대한 정보를 얻는 방법을 설명합니다.

## QoS 정책 계획(작업 맵)

이 작업 맵에서는 QoS 정책 계획에 대한 주요 작업을 나열하고 각 작업에 대한 수행 지침과 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS를 지원하도록 네트워크 토폴로지를 설계합니다.	네트워크에서 차별화된 서비스를 제공할 호스트 및 라우터를 식별합니다.	407 페이지 “네트워크에서 IPQoS를 준비하는 방법”
2. 네트워크의 서비스를 구분해야 할 클래스를 정의합니다.	사이트에서 제공하는 서비스 유형 및 SLA를 확인하고 해당 서비스가 속한 고유한 트래픽 클래스를 결정합니다.	407 페이지 “QoS 정책에 대한 클래스 정의 방법”
3. 클래스에 대한 필터를 정의합니다.	특정 클래스의 트래픽과 네트워크 트래픽 흐름을 구분할 가장 적합한 방법을 결정합니다.	410 페이지 “QoS 정책에서 필터를 정의하는 방법”
4. 패킷이 IPQoS 시스템에서 나갈 때 트래픽을 측정할 흐름 제어 속도를 정의합니다.	각 트래픽 클래스에 대해 허용 가능한 흐름 속도를 결정합니다.	411 페이지 “흐름 제어 계획 방법”
5. QoS 정책에서 사용할 DSCP 또는 사용자 우선 순위 값을 정의합니다.	라우터 또는 스위치가 흐름을 처리할 때 트래픽 흐름에 지정되는 전달 동작을 결정할 체계를 계획합니다.	414 페이지 “전달 동작 계획 방법”
6. 해당하는 경우 네트워크의 트래픽 흐름에 대한 통계 모니터링 계획을 설정합니다.	트래픽 클래스를 평가하여 계산 또는 통계 용도로 모니터링해야 할 트래픽 흐름을 결정합니다.	416 페이지 “흐름 계산 계획 방법”

주 - 이 절의 나머지 부분에서는 IPQoS 사용 시스템의 QoS 정책을 계획하는 방법에 대해 설명합니다. Diffserv 라우터에 대한 QoS 정책을 계획하려면 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

## ▼ 네트워크에서 IPQoS를 준비하는 방법

다음 절차에서는 QoS 정책을 만들기 전에 수행할 일반적인 계획 작업을 나열합니다.

- 1 네트워크 토폴로지를 검토합니다. 그런 다음 IPQoS 시스템 및 Diffserv 라우터를 사용하는 전략을 계획합니다.  
토폴로지 예는 [402 페이지 “Diffserv 네트워크 토폴로지 계획”](#)을 참조하십시오.
- 2 토폴로지에서 IPQoS를 필요로 하거나 IPQoS 서비스로 사용 가능한 적합한 후보가 될 수 있는 호스트를 식별합니다.
- 3 동일한 QoS 정책을 사용할 수 있는 IPQoS 사용 시스템을 결정합니다.  
예를 들어, 네트워크의 모든 호스트에서 IPQoS를 사용으로 설정하려면 동일한 QoS 정책을 사용할 수 있는 호스트를 식별합니다. 각 IPQoS 사용 시스템에는 해당 IPQoS 구성 파일에서 구현되는 로컬 QoS 정책이 있어야 합니다. 하지만 특정 범위의 시스템에서 사용할 하나의 IPQoS 구성 파일을 만들 수 있습니다. 그런 다음 QoS 정책 요구 사항이 동일한 모든 시스템에 구성 파일을 복사할 수 있습니다.
- 4 네트워크의 Diffserv 라우터에 필요한 계획 작업을 검토하고 수행합니다.  
자세한 내용은 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

## ▼ QoS 정책에 대한 클래스 정의 방법

첫 번째 QoS 정책 정의 단계는 트래픽 흐름을 클래스로 구성하는 것입니다. Diffserv 네트워크에서 모든 유형의 트래픽에 대해 클래스를 만들 필요는 없습니다. 네트워크 토폴로지에 따라 각 IPQoS 사용 시스템에 대해 다른 QoS 정책을 만들어야 할 수도 있습니다.

주 - 클래스 개요는 [393 페이지 “IPQoS 클래스”](#)를 참조하십시오.

다음 절차에서는 [407 페이지 “네트워크에서 IPQoS를 준비하는 방법”](#)에 설명된 대로 IPQoS를 사용할 네트워크의 시스템을 결정했다고 가정합니다.

- 1 QoS 정책 정보를 구성하는 데 필요한 QoS 계획 테이블을 만듭니다.  
제안 사항은 [표 28-1](#)을 참조하십시오.

## 2 네트워크에 있는 모든 QoS 정책에 대해 나머지 단계를 수행합니다.

### 3 QoS 정책에서 사용할 클래스를 정의합니다.

다음 질문은 가능한 클래스 정의를 위한 네트워크 트래픽 분석 지침입니다.

#### ■ 회사에서 고객에게 서비스 단계 계약을 제공합니까?

그럴 경우 회사에서 고객에게 제공하는 SLA의 상대적인 우선 순위 레벨을 평가합니다. 다른 우선 순위 레벨이 보장된 고객에게 동일한 응용 프로그램을 제공할 수 있습니다.

예를 들어, 회사에서 각 고객에게 웹 사이트 호스팅을 제공할 수 있습니다. 이 경우 각 고객 웹 사이트에 대한 클래스를 정의해야 합니다. 고급 웹 사이트를 하나의 서비스 레벨로 제공하는 SLA도 있을 수 있고, 할인 고객에게 "최상의" 개인 웹 사이트를 제공하는 SLA도 있을 수 있습니다. 이 요소는 다양한 웹 사이트를 클래스뿐만 아니라 웹 사이트 클래스에 지정되는 잠재적으로 다른 흐름 동작도 나타냅니다.

#### ■ IPQoS 시스템이 흐름 제어가 필요할 수 있는 많이 사용되는 응용 프로그램을 제공합니까?

과도한 트래픽을 생성하는 많이 사용되는 응용 프로그램을 제공하는 서버에서 IPQoS를 사용으로 설정하여 네트워크 성능을 향상시킬 수 있습니다. 일반적인 예로 전자 메일, 네트워크 뉴스 및 FTP를 들 수 있습니다. 가능한 경우 서비스 유형별로 수신 및 송신 트래픽에 대해 별도의 클래스를 만드는 것이 좋습니다. 예를 들어, 메일 서버용 QoS 정책에 대해 mail-in 클래스와 mail-out 클래스를 만들 수 있습니다.

#### ■ 네트워크에서 우선 순위가 가장 높은 전달 동작을 필요로 하는 특정 응용 프로그램이 실행됩니까?

우선 순위가 가장 높은 전달 동작을 필요로 하는 중요한 응용 프로그램은 라우터 대기열에서 가장 높은 우선 순위를 받아야 합니다. 일반적인 예로 스트리밍 비디오 및 스트리밍 오디오를 들 수 있습니다.

이와 같이 우선 순위가 높은 응용 프로그램에 대해 수신 클래스와 송신 클래스를 정의합니다. 그런 다음 응용 프로그램을 제공하는 IPQoS 사용 시스템과 Diffserv 라우터의 QoS 정책에 클래스를 추가합니다.

#### ■ 흐름에 대역폭이 많이 사용되어 네트워크에서 트래픽 흐름이 제어되어야 합니까?

netstat, snoop 및 기타 네트워크 모니터링 유틸리티를 사용하여 네트워크에 문제를 일으키고 있는 트래픽의 유형을 검색할 수 있습니다. 지금까지 만든 클래스를 검토한 다음 정의되지 않은 문제 트래픽 범주에 대해 새 클래스를 만듭니다. 문제 트래픽 범주에 대한 클래스를 이미 정의한 경우 문제 트래픽을 제어할 측정기의 속도를 정의합니다.

네트워크에 있는 모든 IPQoS 사용 시스템의 문제 트래픽에 대한 클래스를 만듭니다. 그러면 각 IPQoS 시스템이 트래픽 흐름을 네트워크로 릴리스하는 속도를 제한하여 문제 트래픽을 처리할 수 있습니다. 또한 Diffserv 라우터에서 QoS 정책에 해당 문제 클래스를 정의해야 합니다. 그러면 라우터가 QoS 정책에 구성된 대로 문제 흐름을 대기열에 지정하고 일정을 잡을 수 있습니다.

#### ■ 특정 유형의 트래픽에 대한 통계를 얻어야 합니까?



간단한 SLA 검토를 통해 계산해야 할 고객 트래픽의 유형을 확인할 수 있습니다. 사이트에서 SLA를 제공하는 경우 계산해야 할 트래픽에 대한 클래스가 이미 만들어진 상태일 것입니다. 모니터링하고 있는 트래픽 흐름에 대한 통계 수집을 사용으로 설정할 클래스를 정의할 수도 있습니다. 또한 보안상 액세스를 제한할 트래픽에 대한 클래스를 만들 수 있습니다.

- 4 1단계에서 만든 QoS 계획 테이블에서 정의한 클래스를 나열합니다.
- 5 각 클래스에 우선 순위 레벨을 지정합니다.  
예를 들어, 우선 순위 레벨 1이 가장 높은 우선 순위의 클래스를 나타내도록 지정하고 나머지 클래스에 우선 순위를 내림차순으로 지정합니다. 지정한 우선 순위 레벨은 구조적인 용도로만 사용됩니다. QoS 정책 템플릿에서 설정한 우선 순위 레벨은 IPQoS에 실제로 사용되지 않습니다. QoS 정책에 적합한 경우 두 개 이상의 클래스에 동일한 우선 순위를 지정할 수도 있습니다.
- 6 클래스 정의가 완료되면 [410 페이지 “QoS 정책에서 필터를 정의하는 방법”](#)에 설명된 대로 각 클래스에 대한 필터를 정의합니다.

## 자세한 정보 클래스 우선 순위 설정

클래스를 만들면 우선 순위가 가장 높은 클래스, 우선 순위가 중간인 클래스, 우선 순위가 최상인 클래스를 빠르게 파악할 수 있습니다. 적합한 클래스 우선 순위 설정 체계는 [414 페이지 “전달 동작 계획 방법”](#)에 설명된 대로 송신 트래픽에 흐름별 동작을 지정할 때 특히 중요합니다.

클래스에 PHB를 지정하는 것 외에 클래스에 대한 필터에 우선 순위 선택기를 정의할 수도 있습니다. 우선 순위 선택기는 IPQoS 사용 호스트에서만 활성화됩니다. 속도와 DSCP가 동일한 여러 클래스가 IPQoS 시스템에서 나갈 때 대역폭 경합이 발생하는 경우가 있다고 가정합니다. 이 경우 각 클래스의 우선 순위 선택기가 동일한 값의 클래스에 지정된 서비스 레벨의 순서를 추가로 지정할 수 있습니다.

## 필터 정의

패킷 흐름을 특정 클래스의 구성원으로 식별할 필터를 만듭니다. 각 필터에는 패킷 흐름 평가 기준을 정의하는 선택기가 포함되어 있습니다. 그러면 IPQoS 사용 시스템이 선택기의 기준을 사용하여 트래픽 흐름에서 패킷을 추출합니다. 그런 다음 IPQoS 시스템이 패킷을 클래스와 연관시킵니다. 필터 소개는 [393 페이지 “IPQoS 필터”](#)를 참조하십시오.

다음 표에서는 가장 일반적으로 사용되는 선택기를 나열합니다. 처음 다섯 개의 선택기는 IPQoS 시스템이 패킷을 흐름 구성원으로 식별하는 데 사용하는 IPQoS 5 튜플을 나타냅니다. 전체 선택기 목록은 [표 32-1](#)을 참조하십시오.

표 28-2 일반적인 IPQoS 선택기

이름	정의
saddr	소스 주소입니다.
daddr	대상 주소입니다.
sport	소스 포트 번호입니다. <code>/etc/services</code> 에 정의된 잘 알려진 포트 번호 또는 사용자 정의 포트 번호를 사용할 수 있습니다.
dport	대상 포트 번호입니다.
protocol	<code>/etc/protocols</code> 의 트래픽 흐름 유형에 지정된 IP 프로토콜 번호 또는 프로토콜 이름입니다.
ip_version	사용할 주소 지정 스타일입니다. IPv4 또는 IPv6을 사용하십시오. IPv4가 기본값입니다.
dsfield	DS 필드 내용, 즉 DSCP입니다. 이미 특정 DSCP가 표시된 수신 패킷을 추출하려면 이 선택기를 사용하십시오.
priority	클래스에 지정된 우선 순위 레벨입니다. 자세한 내용은 <a href="#">407 페이지 “QoS 정책에 대한 클래스 정의 방법”</a> 을 참조하십시오.
user	상위 레벨 응용 프로그램이 실행될 때 사용되는 UNIX 사용자 ID 또는 사용자 이름입니다.
projid	상위 레벨 응용 프로그램이 실행될 때 사용되는 프로젝트 ID입니다.
direction	트래픽 흐름 방향입니다. 값은 LOCAL_IN, LOCAL_OUT, FWD_IN 또는 FWD_OUT입니다.

주 - 선택기를 선택할 때는 신중하십시오. 클래스에 대한 패킷을 추출하는 데 필요한 만큼만 선택기를 사용하십시오. 선택기를 많이 정의할수록 IPQoS 성능에 끼치는 영향이 커집니다.

## ▼ QoS 정책에서 필터를 정의하는 방법

시작하기 전에 다음 단계를 수행하려면 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#) 절차를 완료해야 합니다.

- 1 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#)에서 만든 QoS 계획 테이블에 각 클래스에 대한 필터를 하나 이상 만듭니다.

가능한 경우 클래스별로 수신 및 송신 트래픽에 대해 별도의 필터를 만드는 것이 좋습니다. 예를 들어, IPQoS 사용 FTP 서버의 QoS 정책에 ftp-in 필터 및 ftp-out 필터를 추가합니다. 그런 다음 기본 선택기 외에 적합한 direction 선택기도 정의할 수 있습니다.

## 2 클래스의 각 필터에 대한 선택기를 하나 이상 정의합니다.

표 28-1에서 소개된 QoS 계획 테이블을 사용하여 정의한 클래스에 대한 필터를 채웁니다.

### 예 28-1 FTP 트래픽에 대한 필터 정의

다음 표는 송신 FTP 트래픽에 대한 필터를 정의하는 방법을 보여 주는 예입니다.

클래스	우선 순위	필터	선택기
ftp-traffic	4	ftp-out	saddr 10.190.17.44 daddr 10.100.10.53 sport 21 direction LOCAL_OUT

- 참조
- 흐름 제어 체계를 정의하려면 411 페이지 “흐름 제어 계획 방법”을 참조하십시오.
  - 흐름이 네트워크 스트림으로 반환될 때의 흐름에 대한 전달 동작을 정의하려면 414 페이지 “전달 동작 계획 방법”을 참조하십시오.
  - 특정 유형의 트래픽에 대한 흐름 계산을 계획하려면 416 페이지 “흐름 계산 계획 방법”을 참조하십시오.
  - QoS 정책에 다른 클래스를 추가하려면 407 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
  - QoS 정책에 다른 필터를 추가하려면 410 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.

## ▼ 흐름 제어 계획 방법

흐름 제어 과정에서는 클래스에 대한 트래픽 흐름이 측정되고 정의된 속도로 패킷이 네트워크로 릴리스됩니다. 흐름 제어를 계획할 때 IPQoS 측정 모듈에 사용할 매개변수를 정의합니다. 측정기는 트래픽이 네트워크로 릴리스되는 속도를 결정합니다. 측정 모듈 소개는 394 페이지 “측정기(tokenmt 및 tswtclmt) 개요”를 참조하십시오.

다음 절차에서는 410 페이지 “QoS 정책에서 필터를 정의하는 방법”에 설명된 대로 필터 및 선택기를 정의했다고 가정합니다.

## 1 네트워크에 대한 최대 대역폭을 확인합니다.

- 2 네트워크에서 지원되는 SLA를 검토합니다. 고객 및 각 고객에게 보장되는 서비스의 유형을 식별합니다.

특정 레벨의 서비스를 보장하려면 고객이 생성한 특정 트래픽 클래스를 측정해야 할 수도 있습니다.

- 3 407 페이지 “QoS 정책에 대한 클래스 정의 방법”에서 만든 클래스 목록을 검토합니다.

SLA와 연관된 클래스 이외의 다른 클래스를 측정해야 할지 여부를 결정합니다.

IPQoS 시스템이 높은 레벨의 트래픽을 생성하는 응용 프로그램을 실행한다고 가정합니다. 응용 프로그램의 트래픽을 분류한 후 흐름을 측정하여 흐름의 패킷이 네트워크로 반환되는 속도를 제어합니다.

---

주 - 모든 클래스를 측정해야 하는 것은 아닙니다. 클래스 목록을 검토할 때 이 지침을 염두에 두십시오.

---

- 4 흐름 제어가 필요한 트래픽을 선택하는 각 클래스의 필터를 결정합니다. 그런 다음 측정이 필요한 클래스 목록을 세분화합니다.

필터가 두 개 이상인 클래스의 경우 하나의 필터에 대해서만 측정해야 합니다. 특정 클래스의 수신 및 송신 트래픽에 대한 필터를 정의한 것으로 가정합니다. 한 방향의 트래픽만 흐름 제어가 필요한 것으로 결론지을 수 있습니다.

- 5 흐름을 제어할 각 클래스에 대한 측정기 모듈을 선택합니다.

QoS 계획 테이블의 측정기 열에 모듈 이름을 추가합니다.

- 6 구조적 테이블에 측정할 각 클래스에 대한 속도를 추가합니다.

tokenmt 모듈을 사용하는 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

이러한 속도가 특정 클래스를 측정하기에 충분할 경우 tokenmt에 대한 커밋 속도 및 커밋 버스트만 정의할 수 있습니다.

필요한 경우 다음 속도도 정의할 수 있습니다.

- 커밋 버스트
- 최고 버스트

tokenmt 속도에 대한 전체 정의는 465 페이지 “두 속도 측정기로 tokenmt 구성”을 참조하십시오. tokenmt(7ipp) 매뉴얼 페이지에서도 자세한 내용을 확인할 수 있습니다.

tswtclmt 모듈을 사용할 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

창 크기(밀리초)도 정의할 수 있습니다. 이러한 속도는 466 페이지 “tswtclmt 측정 모듈” 및 twstclmt(7ipp) 매뉴얼 페이지에 정의되어 있습니다.

## 7 측정된 트래픽에 대한 트래픽 준수 결과를 추가합니다.

두 측정 모듈의 결과는 녹색, 빨간색 및 노란색입니다. 정의한 속도에 적용되는 트래픽 준수 결과를 QoS 구조적 테이블에 추가합니다. 측정기 결과는 463 페이지 “측정기 모듈”에서 자세히 설명됩니다.

커밋 속도를 준수하는 트래픽 또는 준수하지 않는 트래픽에 대해 수행해야 할 작업을 결정해야 합니다. 항상은 아니지만 이 작업은 패킷 헤더에 해당 동작을 표시하는 경우가 많습니다. 트래픽 흐름이 커밋 속도를 초과하지 않는 상태에서 녹색 레벨의 트래픽에 대해 허용 가능한 작업 중 하나는 처리를 계속하는 것일 수 있습니다. 흐름이 최고 속도를 초과할 경우 클래스의 패킷을 삭제하는 작업을 수행할 수도 있습니다.

### 예 28-2 측정기 정의

다음 표는 전자 메일 트래픽의 클래스에 대한 측정기 항목을 보여 주는 예입니다. IPQoS 시스템이 있는 네트워크의 총 대역폭은 100메가비트/초 또는 100000000비트/초입니다. QoS 정책은 전자 메일 클래스에 낮은 우선 순위를 지정합니다. 또한 이 클래스는 최상의 전달 동작을 수신합니다.

클래스	우선 순위	필터	선택기	속도
email	8	mail_in	daddr10.50.50.5  dport imap  direction LOCAL_IN	
email	8	mail_out	saddr10.50.50.5  sport imap  direction LOCAL_OUT	meter=tokenmt  커밋 속도=5000000  커밋 버스트=5000000  최고 속도=10000000  최고 버스트=1000000  녹색 우선 순위=처리 계속  노란색 우선 순위=노란색 PHB 표시  빨간색 우선 순위=삭제

참조 ■ 패킷이 네트워크 스트림으로 반환될 때의 흐름에 대한 전달 동작을 정의하려면 414 페이지 “전달 동작 계획 방법”을 참조하십시오.

- 특정 유형의 트래픽에 대한 흐름 계산을 계획하려면 [416 페이지 “흐름 계산 계획 방법”](#)을 참조하십시오.
- QoS 정책에 다른 클래스를 추가하려면 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#)을 참조하십시오.
- QoS 정책에 다른 필터를 추가하려면 [410 페이지 “QoS 정책에서 필터를 정의하는 방법”](#)을 참조하십시오.
- 다른 흐름 제어 체계를 정의하려면 [411 페이지 “흐름 제어 계획 방법”](#)을 참조하십시오.
- IPQoS 구성 파일을 만들려면 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

## ▼ 전달 동작 계획 방법

전달 동작에 따라 네트워크로 전달될 트래픽 흐름의 우선 순위 및 삭제 우선 순위가 결정됩니다. 두 가지 주요 전달 동작(다른 트래픽 클래스와 관계가 있는 클래스의 흐름 우선 순위 설정 또는 전체 흐름 삭제)을 선택할 수 있습니다.

Diffserv 모델은 표시자를 사용하여 선택된 전달 동작을 트래픽 흐름에 지정합니다. IPQoS는 다음 표시자 모듈을 제공합니다.

- `dscpmk` - IP 패킷의 DS 필드에 DSCP를 표시하는 데 사용됩니다.
- `dlcosmk` - 데이터그램의 VLAN 태그에 CoS(class-of-service) 값을 표시하는 데 사용됩니다.

---

주 - 이 절의 제안 사항은 IP 패킷에 해당하는 것입니다. IPQoS 시스템에 VLAN 장치가 포함된 경우 `dlcosmk` 표시자를 사용하여 데이터그램에 대한 전달 동작을 표시할 수 있습니다. 자세한 내용은 [469 페이지 “VLAN 장치에서 `dlcosmk` 표시기 사용”](#)을 참조하십시오.

---

IP 트래픽의 우선 순위를 설정하려면 각 패킷에 DSCP를 지정해야 합니다. `dscpmk` 표시자는 패킷의 DS 필드에 DSCP를 표시합니다. 전달 동작 유형과 연관된 잘 알려진 코드점 그룹에서 클래스에 대한 DSCP를 선택합니다. 이러한 잘 알려진 코드점은 EF PHB의 경우 46(101110)이며 AF PHB의 경우 코드점 범위입니다. DSCP 및 전달에 대한 개요 정보는 [397 페이지 “IPQoS 사용 네트워크에서 트래픽 전달”](#)을 참조하십시오.

**시작하기 전에** 다음 단계에서는 QoS 정책에 대한 클래스 및 필터를 정의했다고 가정합니다. 측정기와 표시자를 함께 사용하여 트래픽을 제어하는 경우가 많기는 하지만 표시자만으로도 전달 동작을 정의할 수 있습니다.

- 1 **지금까지 만든 클래스 및 각 클래스에 지정한 우선 순위를 검토합니다.**  
모든 트래픽 클래스를 표시해야 하는 것은 아닙니다.

## 2 우선 순위가 가장 높은 클래스에 EF 흡당 동작을 지정합니다.

EF PHB는 EFDSCP 46(101110)이 지정된 패킷이 AF PHB가 지정된 패킷보다 먼저 네트워크에 릴리스되도록 합니다. 우선 순위가 가장 높은 트래픽에 EF PHB를 사용합니다. EF에 대한 자세한 내용은 [467 페이지 “EF\(빠른 전달\) PHB”](#)를 참조하십시오.

## 3 측정할 트래픽이 있는 클래스에 전달 동작을 지정합니다.

## 4 클래스에 지정한 우선 순위에 따라 나머지 클래스에 DS 코드점을 지정합니다.

### 예 28-3 게임 응용 프로그램에 대한 QoS 정책

일반적으로 트래픽은 다음 이유로 측정됩니다.

- 네트워크 사용량이 많을 때 SLA가 이 클래스의 패킷에 대한 서비스 레벨을 보장합니다.
- 우선 순위가 보다 낮은 클래스가 네트워크의 혼잡을 야기할 수 있습니다.

표시자와 측정기를 함께 사용하여 이러한 클래스에 차별화된 서비스 및 대역폭 관리를 제공합니다. 예를 들어, 다음 표에서는 QoS 정책의 일부를 보여 줍니다. 이 정책은 높은 레벨의 트래픽을 생성하는 많이 사용되는 게임 응용 프로그램에 대한 클래스를 정의합니다.

클래스	우선 순위	필터	선택기	속도	전달 여부
games_app	9	games_in	sport 6080	해당 없음	해당 없음
games_app	9	games_out	dport 6081	meter=tokenmt 커밋 속도=5000000 커밋 버스트=5000000 최고 속도=10000000 최고 버스트=15000000 녹색 우선 순위=처리 계속 노란색 우선 순위=노란색 PHB 표시 빨간색 우선 순위=삭제	녹색=AF31 노란색=AF42 빨간색=drop

전달 동작은 커밋 속도를 준수하거나 최고 속도에 미치지 않는 games\_app 트래픽에 우선 순위가 낮은 DSCP를 지정합니다. games\_app 트래픽이 최고 속도를 초과하면 QoS 정책은 games\_app의 패킷이 삭제되도록 합니다. 모든 AF 코드점은 [표 32-2](#)에서 나열됩니다.

- 참조
- 특정 유형의 트래픽에 대한 흐름 계산을 계획하려면 [416 페이지 “흐름 계산 계획 방법”](#)을 참조하십시오.
  - QoS 정책에 다른 클래스를 추가하려면 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#)을 참조하십시오.
  - QoS 정책에 다른 필터를 추가하려면 [410 페이지 “QoS 정책에서 필터를 정의하는 방법”](#)을 참조하십시오.
  - 흐름 제어 체계를 정의하려면 [411 페이지 “흐름 제어 계획 방법”](#)을 참조하십시오.
  - 패킷이 네트워크 스트림으로 반환될 때의 흐름에 대한 추가 전달 동작을 정의하려면 [414 페이지 “전달 동작 계획 방법”](#)을 참조하십시오.
  - IPQoS 구성 파일을 만들려면 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

## ▼ 흐름 계산 계획 방법

IPQoS flowacct 모듈을 사용하여 청구 또는 네트워크 관리 용도로 트래픽 흐름을 추적할 수 있습니다. 다음 절차에 따라 QoS 정책에 흐름 계산이 포함되어야 할지 여부를 결정하십시오.

### 1 회사에서 고객에게 SLA를 제공합니까?

그럴 경우 흐름 계산을 사용해야 합니다. SLA를 검토하여 회사에서 고객에게 청구할 네트워크 트래픽의 유형을 결정합니다. 그런 다음 QoS 정책을 검토하여 청구할 트래픽을 선택하는 클래스를 결정합니다.

### 2 네트워크 문제가 발생하지 않도록 모니터링하거나 테스트해야 할 응용 프로그램이 있습니까?

있을 경우 흐름 계산을 사용하여 이러한 응용 프로그램의 동작을 관찰하는 것이 좋습니다. QoS 정책을 검토하여 모니터링해야 할 트래픽에 지정한 클래스를 확인합니다.

### 3 QoS 계획 테이블에서 흐름 계산이 필요한 각 클래스에 대해 흐름 계산 열에 Y를 표시합니다.

- 참조
- QoS 정책에 다른 클래스를 추가하려면 [407 페이지 “QoS 정책에 대한 클래스 정의 방법”](#)을 참조하십시오.
  - QoS 정책에 다른 필터를 추가하려면 [410 페이지 “QoS 정책에서 필터를 정의하는 방법”](#)을 참조하십시오.
  - 흐름 제어 체계를 정의하려면 [411 페이지 “흐름 제어 계획 방법”](#)을 참조하십시오.
  - 패킷이 네트워크 스트림으로 반환될 때의 흐름에 대한 전달 동작을 정의하려면 [414 페이지 “전달 동작 계획 방법”](#)을 참조하십시오.



- 특정 유형의 트래픽에 대한 추가 흐름 계산을 계획하려면 416 페이지 “흐름 계산 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

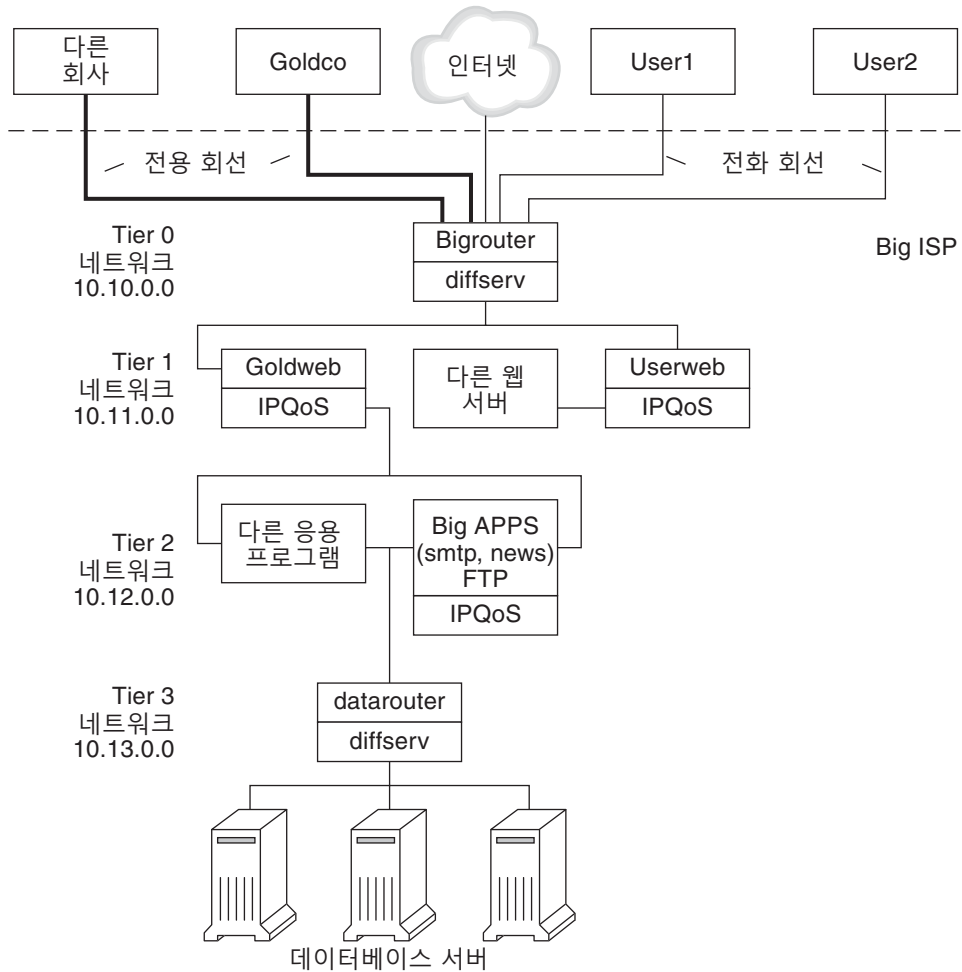
## IPQoS 구성 예 소개

본 설명서의 나머지 장에 나오는 작업에서는 이 절에 소개된 IPQoS 구성 예를 사용합니다. 예에서는 가상 서비스 제공업체인 BigISP의 공용 인트라넷에 있는 차별화된 서비스 솔루션을 보여 줍니다. BigISP는 전용 회선을 통해 BigISP에 연결하는 대규모 회사에 서비스를 제공합니다. 모뎀을 통한 전화 접속을 사용하는 개인도 BigISP에서 서비스를 구입할 수 있습니다.

## IPQoS 토폴로지

다음 그림에서는 BigISP의 공용 인트라넷에 사용되는 네트워크 토폴로지를 보여 줍니다.

그림 28-4 IPQoS 예 토폴로지



BigISP는 공용 인터넷에서 다음 네 계층을 구현했습니다.

- **Tier 0 - 10.10.0.0** 네트워크에는 외부 인터페이스와 내부 인터페이스가 모두 있는 **Bigrouter**라는 큰 Diffserv 라우터가 있습니다. **Goldco**라는 대규모 조직을 비롯하여 여러 회사가 **Bigrouter**에서 종료되는 전용 회선 서비스를 임대했습니다. Tier 0은 전화 회선 또는 ISDN을 통해 연결하는 개인 고객도 처리합니다.
- **Tier 1 - 10.11.0.0** 네트워크는 웹 서비스를 제공합니다. **Goldweb** 서버는 **Goldco**가 BigISP로부터 구매한 고급 서비스에 포함된 웹 사이트를 호스팅합니다. **Userweb** 서버는 개인 고객이 구매한 작은 웹 사이트를 호스팅합니다. **Goldweb**과 **Userweb**에는 모두 IPQoS가 사용됩니다.

- **Tier 2 - 10.12.0.0** 네트워크는 모든 고객에게 사용할 응용 프로그램을 제공합니다. 애플리케이션 서버 중 하나인 BigAPPS에는 IPQoS가 사용됩니다. BigAPPS는 SMTP, 뉴스 및 FTP 서비스를 제공합니다.
- **Tier 3 - 10.13.0.0** 네트워크는 큰 데이터베이스 서버를 다룹니다. Tier 3에 대한 액세스는 Diffserv 라우터인 datarouter를 통해 제어됩니다.



## IPQoS 구성 파일 만들기(작업)

이 장에서는 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이 장에서는 다음 항목을 다룹니다.

- 421 페이지 “IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)”
- 422 페이지 “QoS 정책을 만들기 위한 도구”
- 423 페이지 “웹 서버에 대한 IPQoS 구성 파일 만들기”
- 436 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일 만들기”
- 445 페이지 “라우터에서 차별화 서비스 제공”

이 장에서는 완전한 QoS 정책이 정의되어 있고, 이 정책을 IPQoS 구성 파일에 대한 기준으로 사용할 준비가 되어 있다고 가정합니다. QoS 정책 계획에 대한 자세한 내용은 405 페이지 “서비스 품질 정책 계획”을 참조하십시오.

## IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)

이 작업 맵에서는 IPQoS 구성 파일을 만들기 위한 일반적인 작업을 나열하고 작업 수행 단계를 설명하는 각 절에 대한 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 네트워크 구성을 계획합니다.	로컬 시스템에서 IPQoS 사용 시스템이 되어야 하는 시스템을 결정합니다.	407 페이지 “네트워크에서 IPQoS를 준비하는 방법”
2. 네트워크에서 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	트래픽 흐름을 고유의 서비스 클래스로 식별합니다. 그런 다음 트래픽 관리가 필요한 흐름을 결정합니다.	405 페이지 “서비스 품질 정책 계획”
3. IPQoS 구성 파일을 만들고 첫번째 작업을 정의합니다.	IPQoS 파일을 만들고 IP 분류기를 호출한 다음 처리할 클래스를 정의합니다.	425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”

작업	설명	수행 방법
4. 클래스에 대한 필터를 만듭니다.	어떤 클래스가 선택되고, 클래스로 구성되는지 제어하는 필터를 추가합니다.	<a href="#">427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”</a>
5. 더 많은 클래스와 필터를 IPQoS 구성 파일에 추가합니다.	IP 분류기로 처리할 더 많은 클래스와 필터를 만듭니다.	<a href="#">433 페이지 “최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법”</a>
6. 측정 모듈을 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 흐름 제어를 요구하는 경우 흐름 제어 속도 및 준수 레벨을 측정기에 지정합니다.	<a href="#">442 페이지 “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”</a>
7. 표시기를 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 차별화된 전달 동작을 요구하는 경우 트래픽 클래스가 전달되는 방식을 정의합니다.	<a href="#">429 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”</a>
8. 흐름 계산 모듈을 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 트래픽 흐름에 대한 통계 수집을 요구하는 경우 계산 통계가 수집되는 방식을 정의합니다.	<a href="#">432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”</a>
9. IPQoS 구성 파일을 적용합니다.	지정된 IPQoS 구성 파일의 내용을 해당하는 커널 모듈에 추가합니다.	<a href="#">448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”</a>
10. 라우터 파일에서 전달 동작을 구성합니다.	네트워크의 IPQoS 구성 파일에서 전달 동작을 정의하는 경우 결과 DSCP를 라우터의 해당하는 일정 파일에 추가합니다.	<a href="#">445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”</a>

## QoS 정책을 만들기 위한 도구

네트워크에 대한 QoS 정책은 IPQoS 구성 파일에 상주합니다. 이 구성 파일은 텍스트 편집기를 사용하여 만듭니다. 그런 다음 파일을 `ipqosconf`(IPQoS 구성 유틸리티)에 인수로 제공합니다. `ipqosconf`가 구성 파일에서 정의된 정책을 적용하도록 지시하면 정책이 커널 IPQoS 시스템에 쓰여집니다. `ipqosconf` 명령에 대한 자세한 내용은 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오. `ipqosconf` 사용에 대한 자세한 내용은 [448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”](#)을 참조하십시오.

## 기본 IPQoS 구성 파일

IPQoS 구성 파일은 405 페이지 “서비스 품질 정책 계획”에서 정의한 QoS 정책을 구현하는 action 명령문 트리로 구성됩니다. IPQoS 구성 파일은 IPQoS 모듈을 구성합니다. 각 작업 명령문에는 작업 명령문에서 호출된 모듈로 처리될 클래스, 필터 또는 매개변수 집합이 포함됩니다.

IPQoS 구성 파일의 전체 구문은 예 32-3 및 ipqosconf(1M) 매뉴얼 페이지를 참조하십시오.

### IPQoS 예제 토폴로지 구성

이 장의 작업에서는 세 IPQoS 사용 시스템에 대한 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이러한 시스템은 그림 28-4에 소개된 BigISP 회사의 네트워크 토폴로지에 속합니다.

- Goldweb – 프리미엄 레벨 SLA를 구매한 고객을 위한 웹 사이트를 호스트하는 웹 서버입니다.
- Userweb – “최선 조건” SLA를 구매한 가정 사용자를 위한 개인용 웹 사이트를 호스트하는 덜 강력한 웹 서버입니다.
- BigAPPS – 골드 레벨 및 최선 조건 고객을 위한 메일, 네트워크 뉴스 및 FTP를 서비스하는 애플리케이션 서버입니다.

이러한 세 구성 파일은 가장 일반적인 IPQoS 구성을 보여줍니다. 다음 절에 나오는 샘플 파일을 고유의 IPQoS 구현을 위한 템플릿으로 사용할 수 있습니다.

## 웹 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 프리미엄 웹 서버에 대한 구성을 만드는 방법을 통해 IPQoS 구성 파일을 소개합니다. 그런 다음 개인용 웹 사이트를 호스트하는 서버에 대한 다른 구성 파일에서 완전히 다른 레벨의 서비스를 구성하는 방법을 보여줍니다. 두 서버는 그림 28-4에 나온 네트워크 예의 일부입니다.

다음 구성 파일은 Goldweb 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 프리미엄 SLA를 구매한 회사인 Goldco에 대한 웹 사이트를 호스트합니다.

예 29-1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
}
```

예 29-1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일 (계속)

```

class {
    name goldweb
    next_action markAF11
    enable_stats FALSE
}
class {
    name video
    next_action markEF
    enable_stats FALSE
}
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class goldweb
}
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
}
action {
    module dscpmk
    name markAF11
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
action {
    module dscpmk
    name markEF
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
action {
    module flowacct
    name acct
    params {
        enable_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
    }
}
}

```



다음 구성 파일은 Userweb에 대한 IPQoS 작업을 정의합니다. 이 서버는 낮은 가격 또는 **최선 조건 SLA**의 개인을 위한 웹 사이트를 호스트합니다. 이 레벨의 서비스는 IPQoS 시스템에서 더 높은 가격 SLA의 고객 트래픽을 처리한 후 최선 조건 고객에게 제공할 수 있는 최상의 서비스를 보장합니다.

예 29-2 최선 조건 웹 서버에 대한 샘플 구성

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name Userweb
        next_action markAF12
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class Userweb
    }
}

action {
    module dscpmk
    name markAF12
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

## ▼ IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법

유지 관리하기 가장 쉬운 디렉토리에서 첫번째 IPQoS 구성 파일을 만들 수 있습니다. 이 장의 작업에서는 IPQoS 구성 파일에 대한 위치로 `/var/ipqos` 디렉토리를 사용합니다. 다음 절차에서는 예 29-1에 소개된 IPQoS 구성 파일의 초기 세그먼트를 만듭니다.

---

주 - IPQoS 구성 파일을 만들 때 각 action 명령문과 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 예 29-1을 참조하십시오.

---

### 1 프리미엄 웹 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

모든 IPQoS 구성 파일은 버전 번호 `fmt_version 1.0`이 첫번째 주석 처리되지 않은 행으로 시작되어야 합니다.

## 2 일반 IP 분류기 `ipgpc`를 구성하는 초기 `action` 명령문에서 여는 매개변수를 따릅니다.

이 초기 작업은 IPQoS 구성 파일을 구성하는 `action` 명령문 트리를 시작합니다. 예를 들어, `/var/ipqos/Goldweb.qos` 파일은 `ipgpc` 분류기를 호출하는 초기 `action` 명령문으로 시작됩니다.

```
fmt_version 1.0
```

```
action {
    module ipgpc
    name ipgpc.classify
```

`fmt_version 1.0` IPQoS 구성 파일을 시작합니다.

`action {` 작업 명령문을 시작합니다.

`module ipgpc` `ipgpc` 분류기를 구성 파일의 첫번째 작업으로 구성합니다.

`name ipgpc.classify` 항상 `ipgpc.classify`가 되어야 하는 분류기 `action` 명령문의 이름을 정의합니다.

`action` 명령문에 대한 자세한 구문 정보는 [474 페이지 “action 명령문”](#) 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

## 3 통계 매개변수 `global_stats`와 함께 `params` 절을 추가합니다.

```
params {
    global_stats TRUE
}
```

`ipgpc.classify` 명령문의 `global_stats TRUE` 매개변수는 해당 작업에 대한 통계 수집을 사용으로 설정합니다. 또한 `global_stats TRUE`는 클래스 절 정의에서 `enable_stats TRUE`를 지정할 때마다 클래스별 통계 수집을 사용으로 설정합니다.

통계를 설정하면 성능이 영향을 받습니다. 새 IPQoS 구성 파일에 대한 통계를 수집하여 IPQoS가 제대로 작동하는지 확인할 수 있습니다. 나중에 `global_stats` 인수를 `FALSE`로 변경하여 통계 수집을 해제할 수 있습니다.

전역 통계는 `params` 절에서 정의할 수 있는 유일한 매개변수 유형입니다. `params` 절에 대한 구문 및 기타 자세한 내용은 [476 페이지 “params 절”](#) 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

## 4 프리미엄 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
    name goldweb
    next_action markAF11
    enable_stats FALSE
}
```

이 명령문을 클래스 절이라고 합니다. `class` 절에는 다음과 같은 내용이 있습니다.

`name goldweb` Goldweb 서버로 향하는 트래픽을 식별하는 `goldweb` 클래스를 만듭니다.

`next_action markAF11`      `ipgpc` 모듈이 `goldweb` 클래스의 패킷을 `markAF11` 작업 명령문에 전달하도록 지시합니다. `markAF11` 작업 명령문은 `dscpmk` 표시기를 호출합니다.

`enable_stats FALSE`      `goldweb` 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 이 클래스에 대한 통계는 설정되지 않습니다.

`class` 절의 구문에 대한 자세한 내용은 [475 페이지 “class 절”](#) 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

## 5 가장 높은 우선 순위 전달을 가져야 하는 응용 프로그램을 식별하는 클래스를 정의합니다.

```
class {
    name video
    next_action markEF
    enable_stats FALSE
}
```

`name video`      `Goldweb` 서버에서 나가는 스트리밍 비디오 트래픽을 식별하는 `video` 클래스를 만듭니다.

`next_action markEF`      `ipgpc`가 처리를 완료한 후 `ipgpc` 모듈이 `video` 클래스의 패킷을 `markEF` 명령문에 전달하도록 지시합니다. `markEF` 명령문은 `dscpmk` 표시기를 호출합니다.

`enable_stats FALSE`      `video` 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 이 클래스에 대한 통계 수집은 설정되지 않습니다.

- 참조
- 방금 만든 클래스에 대한 필터를 정의하려면 [427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”](#)을 참조하십시오.
  - 구성 파일에 대한 다른 클래스 절을 만들려면 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

## ▼ IPQoS 구성 파일에서 필터를 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에서 클래스에 대한 필터를 정의하는 방법을 보여줍니다.

**시작하기 전에** 이 절차에서는 이미 파일 만들기를 시작하고 클래스를 정의했다고 가정합니다. 단계는 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)에서 만든 `/var/ipqos/Goldweb.qos` 파일 만들기를 계속합니다.

주-IPQoS 구성 파일을 만들 때 각 **class** 절 및 각 **filter** 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 [예 29-1](#)을 참조하십시오.

### 1 IPQoS 구성 파일을 열고 정의한 마지막 클래스의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 class 절 이후 시작할 수 있습니다.

```
class {
    name video
    next_action markEF
    enable_stats FALSE
}
```

### 2 IPQoS 시스템의 송신 트래픽을 선택하는 filter 절을 정의합니다.

```
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class goldweb
}
```

name webout                      webout 이름을 필터에 제공합니다.

sport 80                          HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.

direction LOCAL\_OUT          로컬 시스템의 송신 트래픽을 추가로 선택합니다.

class goldweb                  필터가 속한 클래스(이 경우 goldweb 클래스)를 식별합니다.

IPQoS 구성 파일의 filter 절에 대한 구문 및 자세한 내용은 [476 페이지](#) “filter 절”을 참조하십시오.

### 3 IPQoS 시스템에서 스트리밍 비디오 트래픽을 선택하는 filter 절을 정의합니다.

```
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
```

name videoout                  videoout 이름을 필터에 제공합니다.

sport videosrv                  이 시스템의 스트리밍 비디오 응용 프로그램에 대해 이전에 정의한 포트인 소스 포트 videosrv의 트래픽을 선택합니다.

direction LOCAL\_OUT          로컬 시스템의 송신 트래픽을 추가로 선택합니다.

class video                      필터가 속한 클래스(이 경우 video 클래스)를 식별합니다.

- 참조
- 표시기 모듈에 대한 전달 동작을 정의하려면 429 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”을 참조하십시오.
  - 측정 모듈에 대한 흐름 제어 매개변수를 정의하려면 442 페이지 “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”을 참조하십시오.
  - IPQoS 구성 파일을 활성화하려면 448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
  - 추가 필터를 정의하려면 427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
  - 응용 프로그램의 트래픽 흐름에 대한 클래스를 만들려면 438 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

## ▼ IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에 클래스에 대한 홈별 동작을 추가하여 트래픽 전달을 정의하는 방법을 보여줍니다.

**시작하기 전에** 이 절차에서는 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 예 29-1에서 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

---

주 - 이 절차에서는 dscpmk 표시기 모듈을 사용하여 트래픽 전달을 구성하는 방법을 보여줍니다. dlclsmk 표시기를 사용하여 VLAN 시스템에서 트래픽 전달에 대한 자세한 내용은 469 페이지 “VLAN 장치에서 dlcosmk 표시기 사용”을 참조하십시오.

---

### 1 IPQoS 구성 파일을 열고 정의한 마지막 필터의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 filter 절 이후 시작할 수 있습니다.

```
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
```

이 filter 절은 ipgpc 분류기 action 명령문의 끝에 있습니다. 그러므로 필터를 종료하는 닫는 중괄호와 action 명령문을 종료하는 두번째 닫는 중괄호가 필요합니다.

### 2 다음 action 명령문으로 표시기를 호출합니다.

```
action {
    module dscpmk
    name markAF11
```

module dscpmk     표시기 모듈 dscpmk를 호출합니다.

`name markAF11` `markAF11` 이름을 `action` 명령문에 제공합니다.

이전에 정의한 클래스 `goldweb`에는 `next_action markAF11` 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 흐름을 `markAF11` 작업 명령문에 보냅니다.

### 3 표시기가 트래픽 흐름에 대해 수행할 작업을 정의합니다.

```
params {
    global_stats FALSE
    dscp_map{0-63:10}
    next_action continue
}
```

`global_stats FALSE` `markAF11` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계는 수집되지 않습니다.

`dscp_map{0-63:10}` 표시기에서 현재 처리 중인 트래픽 클래스 `goldweb`의 패킷 헤더에 DSCP 10을 지정합니다.

`next_action continue` 트래픽 클래스 `goldweb`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP 10은 표시기가 `dscp` 맵의 모든 항목을 십진수 값 10(이진수 001010)으로 설정하도록 지시합니다. 이 코드 포인트는 `goldweb` 트래픽 클래스의 패킷이 AF11 홉별 동작에 종속된다는 것을 나타냅니다. AF11은 DSCP 10의 모든 패킷이 낮은 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다. 따라서 `Goldweb`의 프리미엄 고객에 대한 송신 트래픽에는 AF(보장 전달) PHB에 대해 사용 가능한 가장 높은 우선 순위가 제공됩니다. AF에 대해 가능한 DSCP 표는 [표 32-2](#)를 참조하십시오.

### 4 다른 표시기 action 명령문을 시작합니다.

```
action {
    module dscpmk
    name markEF
```

`module dscpmk` 표시기 모듈 `dscpmk`를 호출합니다.

`name markEF` `markEF` 이름을 `action` 명령문에 제공합니다.

### 5 표시기가 트래픽 흐름에 대해 수행할 작업을 정의합니다.

```
params {
    global_stats TRUE
    dscp_map{0-63:46}
    next_action acct
}
```

<code>global_stats TRUE</code>	스트리밍 비디오 패킷을 선택하는 <code>video</code> 클래스에 대한 통계 수집을 사용으로 설정합니다.
<code>dscp_map{0-63:46}</code>	표시기에서 현재 처리 중인 트래픽 클래스 <code>video</code> 의 패킷 헤더에 DSCP 46을 지정합니다.
<code>next_action acct</code>	<code>dscpmk</code> 가 처리를 완료한 후 <code>dscpmk</code> 모듈이 <code>video</code> 클래스의 패킷을 <code>acct action</code> 명령문에 전달하도록 지시합니다. <code>acct action</code> 명령문은 <code>flowacct</code> 모듈을 호출합니다.

DSCP 46은 `dscpmk` 모듈이 `dscp` 맵의 모든 항목을 DS 필드에서 십진수 값 46(이진수 101110)으로 설정하도록 지시합니다. 이 코드 포인트는 `video` 트래픽 클래스의 패킷이 EF(빠른 전달) 흐름 동작에 종속된다는 것을 나타냅니다.

---

주 - EF에 대해 권장되는 코드 포인트는 46(이진수 101110)입니다. 기타 DSCP는 AF PHB를 패킷에 지정합니다.

---

EF PHB는 DSCP 46의 패킷이 IPQoS 및 Diffserv 인식 시스템에서 가장 높은 우선권을 받도록 보장합니다. 스트리밍 응용 프로그램에는 가장 높은 우선 순위의 서비스가 필요하므로 QoS 정책에서 스트리밍 응용 프로그램에 EF PHB를 지정하게 됩니다. 빠른 전달 PHB에 대한 자세한 내용은 [467 페이지 “EF\(빠른 전달\) PHB”](#)를 참조하십시오.

## 6 방금 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

자세한 내용은 [445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”](#)을 참조하십시오.

- 참조**
- 트래픽 흐름에 대한 흐름 계산 통계 수집을 시작하려면 [432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”](#)을 참조하십시오.
  - 표시기 모듈에 대한 전달 동작을 정의하려면 [429 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”](#)을 참조하십시오.
  - 측정 모듈에 대한 흐름 제어 매개변수를 정의하려면 [442 페이지 “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”](#)을 참조하십시오.
  - IPQoS 구성 파일을 활성화하려면 [448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”](#)을 참조하십시오.
  - 추가 필터를 정의하려면 [427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”](#)을 참조하십시오.
  - 응용 프로그램의 트래픽 흐름에 대한 클래스를 만들려면 [438 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”](#)을 참조하십시오.

## ▼ IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법

다음 절차에서는 IPQoS 구성 파일에서 트래픽 클래스에 대한 계산을 사용으로 설정하는 방법을 보여줍니다. 절차는 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)에 소개된 video 클래스에 대한 흐름 계산을 정의하는 방법을 보여줍니다. 이 클래스는 프리미엄 고객의 SLA의 일부로 청구되어야 하는 스트리밍 비디오 트래픽을 선택합니다.

시작하기 전에 이 절차에서는 이미 정의된 클래스, 필터, 측정 작업(해당하는 경우) 및 표시 작업(해당하는 경우)이 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 [예 29-1](#)에서 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

### 1 IPQoS 구성 파일을 열고 정의한 마지막 action 명령문의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 markEF action 명령문 이후 시작할 수 있습니다.

```
action {
  module dscpmk
  name markEF
  params {
    global_stats TRUE
    dscp_map{0-63:46}
    next_action acct
  }
}
```

### 2 흐름 계산을 호출하는 action 명령문을 시작합니다.

```
action {
  module flowacct
  name acct
```

module flowacct      흐름 계산 모듈 flowacct를 호출합니다.

name acct              acct 이름을 action 명령문에 제공합니다.

### 3 트래픽 클래스에 대한 계산을 제어하는 params 절을 정의합니다.

```
params {
  global_stats TRUE
  timer 10000
  timeout 10000
  max_limit 2048
  next_action continue
}
```

global\_stats TRUE      스트리밍 비디오 패킷을 선택하는 video 클래스에 대한 통계 수집을 사용으로 설정합니다.



<code>timer 10000</code>	시간 초과된 흐름에 대해 흐름 테이블이 검사되는 간격(밀리초)를 지정합니다. 이 매개변수에서 간격은 10000밀리초입니다.
<code>timeout 10000</code>	최소 간격 시간 초과 값을 지정합니다. 흐름 패킷이 시간 초과 간격 동안 보이지 않으면 흐름이 “시간 초과”됩니다. 이 매개변수에서 패킷은 10000밀리초 후 시간 초과됩니다.
<code>max_limit 2048</code>	이 작업 인스턴스에 대한 흐름 테이블에서 최대 활성 흐름 레코드 수를 설정합니다.
<code>next_action continue</code>	트래픽 클래스 <code>video</code> 의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

`flowacct` 모듈은 지정된 `timeout` 값에 도달할 때까지 특정 클래스의 패킷 흐름에 대한 통계 정보를 수집합니다.

- 참조
- 라우터에 대한 흐름별 동작을 구성하려면 445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
  - IPQoS 구성 파일을 활성화하려면 448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
  - 응용 프로그램의 트래픽 흐름에 대한 클래스를 만들려면 438 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

## ▼ 최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법

최선 조건 웹 서버에 대한 IPQoS 구성 파일은 프리미엄 웹 서버에 대한 IPQoS 구성 파일과 약간 다릅니다. 예로 절차에서는 예 29-2의 구성 파일을 사용합니다.

- 1 최선 조건 웹 서버에 로그인합니다.
- 2 `.qos` 확장자로 새 IPQoS 구성 파일을 만듭니다.

```
fmt_vesion 1.0
action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
}
```

/var/ipqos/userweb.qos 파일은 ipgpc 분류기를 호출하는 부분 action 명령문으로 시작되어야 합니다. 또한 action 명령문에는 통계 수집을 설정하는 params 절도 있어야 합니다. 이 action 명령문에 대한 설명은 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

### 3 최선 조건 웹 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
    name userweb
    next_action markAF12
    enable_stats FALSE
}
```

name userweb           사용자의 웹 트래픽 전달을 위한 userweb이라는 클래스를 만듭니다.

next\_action markAF1    ipgpc가 처리를 완료한 후 ipgpc 모듈이 userweb 클래스의 패킷을 markAF12 action 명령문에 전달하도록 지시합니다. markAF12 action 명령문은 dscpmk 표시기를 호출합니다.

enable\_stats FALSE    userweb 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable\_stats의 값이 FALSE이므로 이 클래스에 대한 통계 수집은 발생하지 않습니다.

class 절 작업에 대한 설명은 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

### 4 userweb 클래스에 대한 트래픽 흐름을 선택하는 filter 절을 정의합니다.

```
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class userweb
}
}
```

name webout           webout 이름을 필터에 제공합니다.

sport 80               HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.

direction LOCAL\_OUT   로컬 시스템의 송신 트래픽을 추가로 선택합니다.

class userweb          필터가 속한 클래스(이 경우 userweb 클래스)를 식별합니다.

filter 절 작업에 대한 설명은 [427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”](#)을 참조하십시오.

### 5 dscpmk 표시기를 호출하는 action 명령문을 시작합니다.

```
action {
    module dscpmk
    name markAF12
}
```

`module dscpmk` 표시기 모듈 `dscpmk`를 호출합니다.

`name markAF12` `markAF12` 이름을 `action` 명령문에 제공합니다.

이전에 정의한 클래스 `userweb`에는 `next_action markAF12` 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 흐름을 `markAF12 action` 명령문에 보냅니다.

## 6 표시기가 트래픽 흐름 처리를 위해 사용할 매개변수를 정의합니다.

```
params {
    global_stats FALSE
    dscp_map{0-63:12}
    next_action continue
}
```

`global_stats FALSE` `markAF12` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계 수집은 발생하지 않습니다.

`dscp_map{0-63:12}` 표시기에서 현재 처리 중인 트래픽 클래스 `userweb`의 패킷 헤더에 DSCP 12를 지정합니다.

`next_action continue` 트래픽 클래스 `userweb`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP 12는 표시기가 `dscp` 맵의 모든 항목을 십진수 값 12(이진수 001100)로 설정하도록 지시합니다. 이 코드 포인트는 `userweb` 트래픽 클래스의 패킷이 AF12 홉별 동작에 종속된다는 것을 나타냅니다. AF12는 DS 필드에서 DSCP 12의 모든 패킷이 중간 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다.

## 7 IPQoS 구성 파일을 완료한 경우 구성을 적용합니다.

- 참조
- 응용 프로그램의 트래픽 흐름에 대한 클래스 및 기타 구성을 추가하려면 [438 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”](#)을 참조하십시오.
  - 라우터에 대한 홉별 동작을 구성하려면 [445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”](#)을 참조하십시오.
  - IPQoS 구성 파일을 활성화하려면 [448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”](#)을 참조하십시오.

## 애플리케이션 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 고객에게 주요 응용 프로그램을 제공하는 애플리케이션 서버에 대한 구성 파일을 만드는 방법을 설명합니다. 이 절차에서는 예로 [그림 28-4](#)의 BigAPPS 서버를 사용합니다.

다음 구성 파일은 BigAPPS 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 고객을 위한 FTP, 전자 메일(SMTP) 및 네트워크 뉴스(NNTP)를 호스트합니다.

**예 29-3** 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name smtp
        enable_stats FALSE
        next_action markAF13
    }
    class {
        name news
        next_action markAF21
    }
    class {
        name ftp
        next_action meterftp
    }
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
    filter {
        name ftpout
        sport ftp
        class ftp
    }
    filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
action {
    module dscpmk
```

예 29-3 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일 (계속)

```

        name markAF13
        params {
            global_stats FALSE
            dscp_map{0-63:14}
            next_action continue
        }
    }
    action {
        module dscpmk
        name markAF21
        params {
            global_stats FALSE
            dscp_map{0-63:18}
            next_action continue
        }
    }
    action {
        module tokenmt
        name meterftp
        params {
            committed_rate 50000000
            committed_burst 50000000
            red_action_name AF31
            green_action_name markAF22
            global_stats TRUE
        }
    }
    action {
        module dscpmk
        name markAF31
        params {
            global_stats TRUE
            dscp_map{0-63:26}
            next_action continue
        }
    }
    action {
        module dscpmk
        name markAF22
        params {
            global_stats TRUE
            dscp_map{0-63:20}
            next_action continue
        }
    }
}

```

## ▼ 애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법

- 1 IPQoS 사용 애플리케이션 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

예를 들어, 애플리케이션 서버에 대해 /var/ipqos/BigAPPS.qos 파일을 만듭니다. 다음 필수 문구로 시작하여 ipgpc 분류기를 호출하는 action 명령문을 시작합니다.

```
fmt_version 1.0
```

```
action {
  module ipgpc
  name ipgpc.classify
  params {
    global_stats TRUE
  }
}
```

여는 action 명령문에 대한 설명은 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

- 2 BigAPPS 서버에서 세 응용 프로그램의 트래픽을 선택하는 클래스를 만듭니다.

action 명령문을 연 후 클래스 정의를 추가합니다.

```
class {
  name smtp
  enable_stats FALSE
  next_action markAF13
}
class {
  name news
  next_action markAF21
}
class {
  name ftp
  enable_stats TRUE
  next_action meterftp
}
```

**name smtp** SMTP 응용 프로그램에서 처리할 전자 메일 트래픽 흐름을 포함하는 smtp라는 클래스를 만듭니다.

**enable\_stats FALSE** smtp 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable\_stats의 값이 FALSE이므로 이 클래스에 대한 통계는 수집되지 않습니다.

**next\_action markAF13** ipgpc가 처리를 완료한 후 ipgpc 모듈이 smtp 클래스의 패킷을 markAF13 action 명령문에 전달하도록 지시합니다.

**name news** NNTP 응용 프로그램에서 처리할 네트워크 뉴스 트래픽 흐름을 포함하는 news라는 클래스를 만듭니다.

next_action markAF21	ipgpc가 처리를 완료한 후 ipgpc 모듈이 news 클래스의 패킷을 markAF21 작업 명령문에 전달하도록 지시합니다.
name ftp	FTP 응용 프로그램에서 처리할 송신 트래픽을 처리하는 ftp라는 클래스를 만듭니다.
enable_stats TRUE	ftp 클래스에 대한 통계 수집을 사용으로 설정합니다.
next_action meterftp	ipgpc가 처리를 완료한 후 ipgpc 모듈이 ftp 클래스의 패킷을 meterftp action 명령문에 전달하도록 지시합니다.

클래스 정의에 대한 자세한 내용은 [425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”](#)을 참조하십시오.

### 3 2단계에서 정의한 클래스의 트래픽을 선택하는 filter 절을 정의합니다.

```
filter {
    name smtpout
    sport smtp
    class smtp
}
filter {
    name newsout
    sport nntp
    class news
}
    filter {
        name ftpout
        sport ftp
        class ftp
    }
    filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
```

name smtpout	smtpout 이름을 필터에 제공합니다.
sport smtp	sendmail(SMTP) 응용 프로그램에 대해 잘 알려진 포트인 소스 포트 25의 트래픽을 선택합니다.
class smtp	필터가 속한 클래스(이 경우 smtp 클래스)를 식별합니다.
name newsout	newsout 이름을 필터에 제공합니다.
sport nntp	네트워크 뉴스(NNTP) 응용 프로그램에 대해 잘 알려진 포트 이름인 소스 포트 이름 nntp의 트래픽을 선택합니다.
class news	필터가 속한 클래스(이 경우 news 클래스)를 식별합니다.
name ftpout	ftpout 이름을 필터에 제공합니다.

sport ftp	FTP 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 21의 제어 데이터를 선택합니다.
name ftpdata	ftpdata 이름을 필터에 제공합니다.
sport ftp-data	FTP 데이터 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 20의 트래픽을 선택합니다.
class ftp	ftpout 및 ftpdata 필터가 속한 클래스(이 경우 ftp 클래스)를 식별합니다.

- 참조
- 필터를 정의하려면 427 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
  - 응용 프로그램 트래픽에 대한 전달 동작을 정의하려면 440 페이지 “IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법”을 참조하십시오.
  - 측정 모듈을 사용하여 흐름 제어를 구성하려면 442 페이지 “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”을 참조하십시오.
  - 흐름 계산을 구성하려면 432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.

## ▼ IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법

다음 절차에서는 응용 프로그램 트래픽에 대한 전달을 구성하는 방법을 보여줍니다. 이 절차에서는 네트워크의 다른 트래픽보다 낮은 우선권을 가질 수 있는 응용 프로그램 트래픽 클래스에 대한 휴별 동작을 정의합니다. 단계는 예 29-3의 /var/ipqos/BigAPPS.qos 파일 만들기를 계속합니다.

시작하기 전에 이 절차에서는 표시할 응용 프로그램에 대해 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다.

- 1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 열고 마지막 filter 절의 끝을 찾습니다.

/var/ipqos/BigAPPS.qos 파일에서 마지막 필터는 다음과 같습니다.

```
filter {
    name ftpdata
    sport ftp-data
    class ftp
}
```



## 2 표시기를 다음과 같이 호출합니다.

```
action {
    module dscpmk
    name markAF13
```

module dscpmk     표시기 모듈 dscpmk를 호출합니다.

name markAF13     markAF13 이름을 action 명령문에 제공합니다.

## 3 전자 메일 트래픽 흐름에 대해 표시할 홉별 동작을 정의합니다.

```
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
```

global\_stats FALSE     markAF13 표시기 action 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable\_stats의 값이 FALSE이므로 통계는 수집되지 않습니다.

dscp\_map{0-63:14}     표시기에서 현재 처리 중인 트래픽 클래스 smtp의 패킷 헤더에 DSCP 14를 지정합니다.

next\_action continue     트래픽 클래스 smtp의 패킷에 추가 처리가 필요하지 않음을 나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있습니다.

DSCP 14는 표시기가 dscp 맵의 모든 항목을 십진수 값 14(이진수 001110)로 설정하도록 지시합니다. DSCP 14는 AF13 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 14의 smtp 트래픽 클래스 패킷을 표시합니다.

AF13은 DSCP 14의 모든 패킷을 높은 삭제 우선권으로 지정합니다. 하지만 AF13은 클래스 1 우선 순위도 보장하므로 라우터는 대기열에서 나가는 전자 메일 트래픽을 높은 우선 순위로 보장합니다. 가능한 AF 코드 포인트 표는 [표 32-2](#)를 참조하십시오.

## 4 네트워크 뉴스 트래픽에 대한 홉별 동작을 정의하는 표시기 action 명령문을 추가합니다.

```
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
```

name markAF21     markAF21 이름을 action 명령문에 제공합니다.

dscp\_map{0-63:18}     표시기에서 현재 처리 중인 트래픽 클래스 nntp의 패킷 헤더에 DSCP 18을 지정합니다.

DSCP 18은 표시기가 dscp 맵의 모든 항목을 십진수 값 18(이진수 010010)로 설정하도록 지시합니다. DSCP 18은 AF21 휴별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 18의 news 트래픽 클래스 패킷을 표시합니다.

AF21은 DSCP 18의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 2 우선 순위를 가집니다. 따라서 네트워크 뉴스 트래픽이 삭제될 가능성은 낮습니다.

- 참조
- 웹 서버에 대한 구성 정보를 추가하려면 425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.
  - 측정 모듈을 사용하여 흐름 제어를 구성하려면 442 페이지 “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”을 참조하십시오.
  - 흐름 계산을 구성하려면 432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
  - 라우터에 대한 전달 동작을 구성하려면 445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
  - IPQoS 구성 파일을 활성화하려면 448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

## ▼ IPQoS 구성 파일에서 흐름 제어를 구성하는 방법

특정 트래픽 흐름이 네트워크로 전송되는 속도를 제어하려면 측정기에 대한 매개변수를 정의해야 합니다. IPQoS 구성 파일에서 두 가지 측정기 모듈 tokenmt 또는 tswtclmt를 사용할 수 있습니다.

다음 절차에서는 예 29-3의 애플리케이션 서버에 대한 IPQoS 구성 파일 만들기를 계속합니다. 이 절차에서는 측정기뿐 아니라 측정기 action 명령문 내에서 호출되는 두 표시기 작업도 구성합니다.

시작하기 전에 단계에서는 흐름을 제어할 응용 프로그램에 대한 클래스 및 필터를 이미 정의했다고 가정합니다.

### 1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 엽니다.

/var/ipqos/BigAPPS.qos 파일에서 다음 표시기 작업 이후 시작합니다.

```
action {
  module dscpmk
  name markAF21
  params {
    global_stats FALSE
    dscp_map{0-63:18}
    next_action continue
  }
}
```

2 ftp 클래스의 트래픽을 흐름 제어하는 측정기 action 명령문을 만듭니다.

```
action {
    module tokenmt
    name meterftp
}
```

module tokenmt      tokenmt 측정기를 호출합니다.

name meterftp      meterftp 이름을 action 명령문에 제공합니다.

3 측정기의 속도를 구성하는 매개변수를 추가합니다.

```
params {
    committed_rate 50000000
    committed_burst 50000000
}
```

committed\_rate 50000000      ftp 클래스의 트래픽에 전송 속도 50,000,000bps를 지정합니다.

committed\_burst 50000000      ftp 클래스의 트래픽에 버스트 크기 50,000,000비트를 커밋합니다.

tokenmt 매개변수에 대한 설명은 465 페이지 “두 속도 측정기로 tokenmt 구성”을 참조하십시오.

4 트래픽 준수 우선권을 구성하는 매개변수를 추가합니다.

```
red_action markAF31
green_action_name markAF22
global_stats TRUE
}
```

red\_action\_name markAF31      ftp 클래스의 트래픽 흐름이 약정된 속도를 초과할 경우 패킷이 markAF31 표시기 action 명령문으로 보내짐을 나타냅니다.

green\_action\_name markAF22      ftp의 트래픽 흐름이 약정된 속도를 준수할 경우 패킷이 markAF22 작업 명령문으로 보내짐을 나타냅니다.

global\_stats TRUE      ftp 클래스에 대한 측정 통계를 사용으로 설정합니다.

트래픽 준수에 대한 자세한 내용은 463 페이지 “측정기 모듈”을 참조하십시오.

5 흐름별 동작을 ftp 클래스의 비준수 트래픽 흐름에 지정하는 표시기 action 명령문을 추가합니다.

```
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
    }
}
```

```

        next_action continue
    }
}
module dscpmk                표시기 모듈 dscpmk를 호출합니다.
name markAF31                markAF31 이름을 action 명령문에 제공합니다.
global_stats TRUE            ftp 클래스에 대한 통계를 사용으로 설정합니다.
dscp_map{0-63:26}            이 트래픽이 약정된 속도를 초과할 때마다 트래픽 클래스
                                ftp의 패킷 헤더에 DSCP 26을 지정합니다.
next_action continue        트래픽 클래스 ftp의 패킷에 추가 처리가 필요하지 않음을
                                나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로
                                돌아갈 수 있습니다.

```

DSCP 26은 표시기가 dscp 맵의 모든 항목을 십진수 값 26(이진수 011010)으로 설정하도록 지시합니다. DSCP 26은 AF31 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 26의 ftp 트래픽 클래스 패킷을 표시합니다.

AF31은 DSCP 26의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 3 우선 순위를 가집니다. 따라서 비준수 FTP 트래픽이 삭제될 가능성은 낮습니다. 가능한 AF 코드 포인트 표는 [표 32-2](#)를 참조하십시오.

## 6 약정된 속도를 준수하는 ftp 트래픽 흐름에 홉별 동작을 지정하는 표시기 action 명령문을 추가합니다.

```

action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
name markAF22                markAF22 이름을 marker 작업에 제공합니다.
dscp_map{0-63:20}            ftp 트래픽이 구성된 속도를 준수할 때마다 트래픽 클래스 ftp의
                                패킷 헤더에 DSCP 20을 지정합니다.

```

DSCP 20은 표시기가 dscp 맵의 모든 항목을 십진수 값 20(이진수 010100)으로 설정하도록 지시합니다. DSCP 20은 AF22 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 20의 ftp 트래픽 클래스 패킷을 표시합니다.

AF22는 DSCP 20의 모든 패킷이 클래스 2 우선 순위로 중간 삭제 우선권을 받도록 보장합니다. 따라서 준수하는 FTP 트래픽은 IPQoS 시스템에서 동시에 전송되는 흐름 중에서 중간 삭제 우선권이 보장됩니다. 하지만 라우터는 클래스 1 중간 삭제 우선권 표시 이상의 트래픽 클래스에 더 높은 전달 우선 순위를 제공합니다. 가능한 AF 코드 포인트 표는 [표 32-2](#)를 참조하십시오.

## 7 애플리케이션 서버에 대해 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

- 참조
- IPQoS 구성 파일을 활성화하려면 448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
  - 웹 서버에 대한 구성 정보를 추가하려면 425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.
  - 흐름 계산을 구성하려면 432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
  - 라우터에 대한 전달 동작을 구성하려면 445 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.

## 라우터에서 차별화 서비스 제공

진정한 차별화 서비스를 제공하려면 402 페이지 “Diffserv 네트워크에 대한 하드웨어 전략”에 설명된 대로 네트워크 토폴로지에 Diffserv 인식 라우터를 포함시켜야 합니다. 라우터에서 Diffserv를 구성하고 해당 라우터의 파일을 업데이트하기 위한 실제 단계는 이 설명서의 범위를 벗어납니다.

이 절에서는 네트워크의 다양한 IPQoS 사용 시스템 및 Diffserv 라우터 사이에서 전달 정보를 조정하기 위한 일반적인 단계를 설명합니다.

### ▼ IPQoS 사용 네트워크에서 라우터를 구성하는 방법

다음 절차에서는 그림 28-4의 토폴로지를 예로 사용합니다.

**시작하기 전에** 다음 절차에서는 이 장의 이전 작업을 수행하여 네트워크에서 IPQoS 시스템을 이미 구성했다고 가정합니다.

#### 1 네트워크의 모든 IPQoS 사용 시스템에 대한 구성 파일을 검토합니다.

#### 2 QoS 다양한 정책에서 사용되는 각 코드 포인트를 식별합니다.

코드 포인트 및 코드 포인트가 적용되는 시스템과 클래스를 나열합니다. 다음 표는 동일한 코드 포인트를 사용했을 수 있는 영역을 나타냅니다. 이 연습은 수용할 수 있습니다. 하지만 동일하게 표시된 클래스의 우선권을 결정하려면 IPQoS 구성 파일에서 다른 조건(예: precedence 선택기)을 제공해야 합니다.

예를 들어, 이 장의 절차에서 사용된 샘플 네트워크의 경우 다음 코드 포인트 표를 만들 수 있습니다.

시스템	클래스	PHB	DS 코드 포인트
Goldweb	video	EF	46 (101110)
Goldweb	goldweb	AF11	10 (001010)
Userweb	webout	AF12	12 (001100)
BigAPPS	smtp	AF13	14 (001110)
BigAPPS	news	AF18	18 (010010)
BigAPPS	ftp 준수 트래픽	AF22	20 (010100)
BigAPPS	ftp 비준수 트래픽	AF31	26 (011010)

**3 네트워크의 IPQoS 구성 파일에서 코드 포인트를 Diffserv 라우터의 해당하는 파일에 추가합니다.**

제공하는 코드 포인트는 라우터의 Diffserv 일정 예약 방식을 구성하는 데 도움이 되어야 합니다. 자세한 내용은 라우터 제조업체의 설명서 및 웹 사이트를 참조하십시오.

## IPQoS 시작 및 유지 관리(작업)

이 장에는 IPQoS 구성 파일을 활성화하고 IPQoS 관련 이벤트를 기록하기 위한 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

- 447 페이지 “IPQoS 관리(작업 맵)”
- 448 페이지 “IPQoS 구성 적용”
- 449 페이지 “IPQoS 메시지에 대한 **syslog** 로깅 사용”
- 450 페이지 “IPQoS 오류 메시지를 사용하여 문제 해결”

### IPQoS 관리(작업 맵)

이 절에서는 Oracle Solaris 시스템에서 IPQoS를 시작하고 유지 관리하기 위한 작업을 나열합니다. 이 작업을 사용하기 전에 421 페이지 “IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)”에 설명된 대로 완성된 IPQoS 구성 파일을 가지고 있어야 합니다.

다음 표에서는 이러한 작업을 나열하고 설명하며 이러한 작업을 완료하는 방법을 자세히 설명하는 링크를 제공합니다.

작업	설명	수행 방법
1. 시스템에서 IPQoS를 구성합니다.	<code>ipqosconf</code> 명령을 사용하여 시스템에서 IPQoS 구성 파일을 활성화합니다.	448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”
2. 각 시스템 부트 후 Oracle Solaris 시작 스크립트가 디버깅된 IPQoS 구성 파일을 적용하도록 합니다.	시스템이 재부트할 때마다 IPQoS 구성이 적용되도록 합니다.	449 페이지 “재부트 때마다 IPQoS 구성이 적용되도록 하는 방법”.
3. IPQoS에 대해 <b>syslog</b> 로깅을 사용으로 설정합니다.	항목을 추가하여 IPQoS 메시지의 <b>syslog</b> 로깅을 사용으로 설정합니다.	449 페이지 “부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법”.

작업	설명	수행 방법
4. 발생하는 IPQoS 문제를 해결합니다.	오류 메시지를 사용하여 IPQoS 문제를 해결합니다.	표 30-1의 오류 메시지를 참조하십시오.

## IPQoS 구성 적용

ipqosconf 명령을 사용하여 IPQoS 구성을 활성화하고 조작합니다.

### ▼ IPQoS 커널 모듈에 새 구성을 적용하는 방법

ipqosconf 명령을 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. 다음 절차에서는 423 페이지 “웹 서버에 대한 IPQoS 구성 파일 만들기”에서 만든 /var/ipqos/Goldweb.qos 파일을 예로 사용합니다. 자세한 내용은 ipqosconf (1M) 매뉴얼 페이지를 참조하십시오.

#### 1 새 구성을 적용합니다.

```
# /usr/sbin/ipqosconf -a/var/ipqos/Goldweb.qos
```

ipqosconf는 지정된 IPQoS 구성 파일의 정보를 Oracle Solaris 커널의 IPQoS 모듈에 씁니다. 이 예에서는 /var/ipqos/Goldweb.qos의 내용이 현재 Oracle Solaris 커널에 적용됩니다.

---

주 --a 옵션을 사용하여 IPQoS 구성 파일을 적용할 경우 파일의 작업이 현재 세션에 대해서만 활성화됩니다.

---

#### 2 새 IPQoS 구성을 테스트하고 디버깅합니다.

UNIX 유틸리티를 사용하여 IPQoS 동작을 추적하고 IPQoS 구현에 대한 통계를 수집합니다. 이 정보는 구성이 예상한 대로 작동하는지 여부를 결정하는 데 도움이 됩니다.

- 참조
- IPQoS 모듈이 어떻게 작동하는지에 대한 통계를 보려면 458 페이지 “통계 정보 수집”을 참조하십시오.
  - ipqosconf 메시지를 기록하려면 449 페이지 “IPQoS 메시지에 대한 syslog 로깅 사용”을 참조하십시오.
  - 각 부트 후 현재 IPQoS 구성이 적용되도록 하려면 449 페이지 “재부트 때마다 IPQoS 구성이 적용되도록 하는 방법”을 참조하십시오.



## ▼ 재부트 때마다 IPQoS 구성이 적용되도록 하는 방법

재부트되더라도 IPQoS 구성을 명시적으로 유지해야 합니다. 그렇지 않으면 시스템이 재부트할 때까지만 현재 구성이 적용됩니다. IPQoS가 시스템에서 올바르게 작동하는 경우 다음을 수행하여 재부트되더라도 구성이 유지되도록 하십시오.

- 1 커널 모듈에 IPQoS 구성이 존재하는지 테스트합니다.

```
# ipqosconf -l
```

구성이 존재하는 경우 ipqosconf가 화면에 구성을 표시합니다. 출력을 받지 못할 경우 448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”에 설명된 대로 구성을 적용합니다.

- 2 IPQoS 시스템이 재부트될 때마다 기존 IPQoS 구성이 적용되도록 합니다.

```
# /usr/sbin/ipqosconf -c
```

-c 옵션은 현재 IPQoS 구성이 부트 시 구성 파일 /etc/inet/ipqosinit.conf에 다시 나타나도록 합니다.

## IPQoS 메시지에 대한 syslog 로깅 사용

IPQoS 부트 시 메시지를 기록하려면 다음 절차에 나온 대로 /etc/syslog.conf 파일을 수정해야 합니다.

## ▼ 부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법

- 1 /etc/syslog.conf 파일을 엽니다.
- 2 다음 텍스트를 파일에 최종 항목으로 추가합니다.

```
user.info /var/adm/messages
```

열 사이에는 공백 대신 탭을 사용합니다.

이 항목은 IPQoS로 생성되는 모든 부트 시 메시지를 /var/adm/messages 파일에 기록합니다.

- 3 시스템을 재부트하여 메시지를 적용합니다.

### 예 30-1 /var/adm/messages의 IPQoS 출력

시스템이 재부트된 후 /var/adm/messages를 보면 출력에 다음과 유사한 IPQoS 로깅 메시지가 포함되어 있습니다.

```
May 14 10:44:33 ipqos-14 ipqosconf: [ID 815575 user.info]
New configuration applied.
May 14 10:44:46 ipqos-14 ipqosconf: [ID 469457 user.info]
Current configuration saved to init file.
May 14 10:44:55 ipqos-14 ipqosconf: [ID 435810 user.info]
Configuration flushed.
```

또한 IPQoS 시스템의 `/var/adm/messages` 파일에서 다음과 유사한 IPQoS 오류 메시지도 볼 수 있습니다.

```
May 14 10:56:47 ipqos-14 ipqosconf: [ID 123217 user.error]
Missing/Invalid config file fmt_version.
May 14 10:58:19 ipqos-14 ipqosconf: [ID 671991 user.error]
No ipgpc action defined.
```

이러한 오류 메시지에 대한 설명은 [표 30-1](#)을 참조하십시오.

# IPQoS 오류 메시지를 사용하여 문제 해결

이 절에서는 IPQoS로 생성되는 오류 메시지 및 가능한 해결 방법에 대한 표가 포함되어 있습니다.

표 30-1 IPQoS 오류 메시지

오류 메시지	설명	해결 방법
Undefined action in parameter <i>parameter-name's</i> action <i>action-name</i>	IPQoS 구성 파일에서 <i>parameter-name</i> 에 지정한 작업 이름이 구성 파일에 존재하지 않습니다.	작업을 만듭니다. 또는 매개변수의 다른 기존 작업을 참조합니다.
action <i>action-name</i> involved in cycle	IPQoS 구성 파일에서 <i>action-name</i> 이 작업 순환의 일부이며, 이는 IPQoS에서 허용되지 않습니다.	작업 순환을 확인합니다. 그런 다음 IPQoS 구성 파일에서 순환 참조 중 하나를 제거합니다.
Action <i>action-name</i> isn't referenced by any other actions	비ipgpc 작업 정의가 IPQoS에서 정의된 다른 작업에 의해 참조되지 않으며, 이는 IPQoS에서 허용되지 않습니다.	참조되지 않는 작업을 제거합니다. 또는 다른 작업이 현재 참조되지 않는 작업을 참조하도록 만듭니다.
Missing/Invalid config file <i>fmt_version</i>	구성 파일의 형식이 파일의 첫번째 항목으로 지정되지 않았으며, 이는 IPQoS에서 필수입니다.	<a href="#">425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”</a> 에 설명된 대로 형식 버전을 추가합니다.
Unsupported config file format version	구성 파일에 지정된 형식 버전이 IPQoS에서 지원되지 않습니다.	IPQoS의 Solaris 9/02 릴리스로 시작하는 데 필요한 <i>fmt_version 1.0</i> 으로 형식 버전을 변경합니다.
No ipgpc action defined.	구성 파일에서 ipgpc 분류기에 대한 작업을 정의하지 않았으며, 이는 IPQoS 필수 사항입니다.	<a href="#">425 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”</a> 에 나온 대로 ipgpc에 대한 작업을 정의합니다.

표 30-1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
Can't commit a null configuration	ipqosconf -c를 실행하여 구성을 커밋할 때 해당 구성이 비어 있었으며, 이는 IPQoS에서 허용되지 않습니다.	구성 커밋을 시도하기 전에 구성 파일을 적용합니다. 자세한 내용은 <a href="#">448 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”</a> 을 참조하십시오.
Invalid CIDR mask on line <i>line-number</i>	구성 파일에서 CIDR 마스크를 IP 주소에 대한 유효한 범위를 벗어난 IP 주소의 일부로 사용했습니다.	마스크 값이 IPv4의 경우 1-32 및 IPv6의 경우 1-128 범위에 있도록 변경합니다.
Address masks aren't allowed for host names line <i>line-number</i>	구성 파일에서 호스트 이름에 대한 CIDR 마스크를 정의했으며, 이는 IPQoS에서 허용되지 않습니다.	마스크를 제거하거나 호스트 이름을 IP 주소로 변경합니다.
Invalid module name line <i>line-number</i>	구성 파일에서 작업 명령문에 지정한 모듈 이름이 잘못되었습니다.	모듈 이름의 철자를 확인합니다. IPQoS 모듈 목록은 <a href="#">표 32-5</a> 를 참조하십시오.
ipgpc action has incorrect name line <i>line-number</i>	구성 파일에서 ipgpc 작업에 제공한 이름이 필요한 ipgpc.classify가 아닙니다.	ipgpc.classify 작업 이름을 바꿉니다.
Second parameter clause not supported line <i>line-number</i>	구성 파일에서 단일 작업에 대해 두 매개변수 절을 지정했으며, 이는 IPQoS에서 허용되지 않습니다.	작업에 대한 모든 매개변수를 단일 매개변수 절로 합칩니다.
Duplicate named action	구성 파일에서 두 작업에 동일한 이름을 제공했습니다.	작업 중 하나의 이름을 바꾸거나 제거합니다.
Duplicate named filter/class in action <i>action-name</i>	동일 작업에서 두 필터 또는 두 클래스에 동일한 이름을 제공했으며, 이는 IPQoS 구성 파일에서 허용되지 않습니다.	필터 또는 클래스 중 하나의 이름을 바꾸거나 제거합니다.
Undefined class in filter <i>filter-name</i> in action <i>action-name</i>	구성 파일에서 필터가 작업에 정의되지 않은 클래스를 참조합니다.	클래스를 만들거나 기존 클래스에 대한 필터 참조를 변경합니다.
Undefined action in class <i>class-name</i> action <i>action-name</i>	클래스가 구성 파일에서 정의되지 않은 작업을 참조합니다.	작업을 만들거나 기존 작업에 대한 참조를 변경합니다.
Invalid parameters for action <i>action-name</i>	구성 파일에서 매개변수 중 하나가 잘못되었습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 의 모듈 항목을 참조합니다. 또는 ipqosconf(1M) 매뉴얼 페이지를 참조할 수 있습니다.
Mandatory parameter missing for action <i>action-name</i>	구성 파일에서 작업에 대한 필수 매개변수를 정의하지 않았습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 의 모듈 항목을 참조합니다. 또는 ipqosconf(1M) 매뉴얼 페이지를 참조할 수 있습니다.

표 30-1 IPQoS 오류 메시지

(계속)

오류 메시지	설명	해결 방법
Max number of classes reached in ipgpc	IPQoS 구성 파일의 <code>ipgpc</code> 작업에서 허용되는 것보다 많은 클래스를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 클래스를 제거합니다. 또는 <code>/etc/system</code> 파일에 <code>ipgpc_max_classesclass-number</code> 항목을 추가하여 최대 클래스 수를 늘릴 수 있습니다.
Max number of filters reached in action ipgpc	IPQoS 구성 파일의 <code>ipgpc</code> 작업에서 허용되는 것보다 많은 필터를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 필터를 제거합니다. 또는 <code>/etc/system</code> 파일에 <code>ipgpc_max_filtersfilter-number</code> 항목을 추가하여 최대 필터 수를 늘릴 수 있습니다.
Invalid/missing parameters for filter <i>filter-name</i> in action ipgpc	구성 파일에 <i>filter-name</i> 필터에 잘못되거나 누락된 매개변수가 있습니다.	유효한 매개변수 목록은 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조하십시오.
Name not allowed to start with '!', line <i>line-number</i>	작업, 필터 또는 클래스 이름을 느낌표(!)로 시작했으며, 이는 IPQoS 파일에서 허용되지 않습니다.	느낌표를 제거하거나 작업, 클래스 또는 필터 이름을 바꿉니다.
Name exceeds the maximum name length line <i>line-number</i>	최대 길이 23자를 초과하는 작업, 클래스 또는 필터 이름을 구성 파일에 정의했습니다.	작업, 클래스 또는 필터에 더 짧은 이름을 제공합니다.
Array declaration line <i>line-number</i> is invalid	구성 파일에서 <i>line-number</i> 행의 매개변수에 대한 배열 선언이 잘못되었습니다.	잘못된 배열의 <code>action</code> 명령문으로 호출되는 배열 선언의 올바른 구문은 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조하십시오.
Quoted string exceeds line, <i>line-number</i>	동일 행에서 문자열에 닫는 인용 부호가 없으며, 이는 구성 파일에서 필수입니다.	구성 파일에서 인용 문자열은 동일 행에서 시작되고 끝나야 합니다.
Invalid value, line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 제공된 값이 매개변수에 대해 지원되지 않습니다.	<code>action</code> 명령문으로 호출되는 모듈에 대해 허용되는 값은 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 의 모듈 설명을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.
Unrecognized value, line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 대한 값이 해당 매개변수에 대해 지원되는 열거 값이 아닙니다.	열거 값이 매개변수에 대해 올바른지 확인합니다. 인식할 수 없는 행 번호의 <code>action</code> 명령문으로 호출되는 모듈에 대한 설명은 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.
Malformed value list line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 지정된 열거가 사양 구문을 준수하지 않습니다.	값 목록 형식이 잘못된 <code>action</code> 명령문으로 호출되는 모듈에 대한 올바른 구문은 <a href="#">461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”</a> 의 모듈 설명을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.

표 30-1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
Duplicate parameter line <i>line-number</i>	<i>line-number</i> 에 지정된 매개변수가 중복되었으며, 이는 구성 파일에서 허용되지 않습니다.	중복된 매개변수 중 하나를 제거합니다.
Invalid action name line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 대한 작업에 사전 정의된 이름 “continue” 또는 “drop”을 사용하는 이름을 제공했습니다.	작업에서 사전 정의된 이름을 사용하지 않도록 작업 이름을 바꿉니다.
Failed to resolve src/dst host name for filter at line <i>line-number</i> , ignoring filter	ipqosconf가 구성 파일에 제공된 필터에 대해 정의된 소스 또는 대상 주소를 확인할 수 없습니다. 따라서 필터가 무시되었습니다.	필터가 중요한 경우 나중에 구성 적용을 시도합니다.
Incompatible address version line <i>line-number</i>	<i>line-number</i> 에서 주소의 IP 버전이 이전에 지정된 IP 주소 또는 ip_version 매개변수의 버전과 호환되지 않습니다.	두 충돌 항목이 호환되도록 변경합니다.
Action at line <i>line-number</i> has the same name as currently installed action, but is for a different module	시스템의 IPQoS 구성에 존재하는 작업의 모듈을 변경하려고 시도했으며, 이는 허용되지 않습니다.	새 구성을 적용하기 전에 현재 구성을 지웁니다.



## 흐름 계산 및 통계 수집 사용(작업)

이 장에서는 IPQoS 시스템이 처리하는 트래픽에 대한 계산 및 통계 정보를 얻는 방법을 설명합니다. 다음 내용으로 구성되어 있습니다.

- 455 페이지 “흐름 계산 설정(작업 맵)”
- 456 페이지 “트래픽 흐름에 대한 정보 기록”
- 458 페이지 “통계 정보 수집”

### 흐름 계산 설정(작업 맵)

다음 작업 맵에서는 flowacct 모듈을 사용하여 트래픽 흐름에 대한 정보를 얻는 일반적인 작업을 나열합니다. 이 맵에서는 해당 작업을 수행하는 절차와 관련된 링크도 제공합니다.

작업	설명	수행 방법
1. 트래픽 흐름에 대한 계산 정보를 포함할 파일을 만듭니다.	acctadm 명령을 사용하여 flowacct를 통한 처리 결과를 보관할 파일을 만듭니다.	456 페이지 “흐름 계산 데이터에 대한 파일을 만드는 방법”
2. IPQoS 구성 파일에서 flowacct 매개변수를 정의합니다.	timer, timeout 및 max_limit 매개변수에 대한 값을 정의합니다.	432 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”

## 트래픽 흐름에 대한 정보 기록

IPQoS flowacct 모듈을 사용하여 트래픽 흐름에 대한 정보를 수집할 수 있습니다. 예를 들어, 소스 및 대상 주소, 흐름의 패킷 수 및 유사한 데이터를 수집할 수 있습니다. 흐름에 대한 정보를 수집하고 기록하는 프로세스를 **흐름 계산**이라고 합니다.

특정 클래스의 트래픽에 대한 흐름 계산 결과는 **흐름 레코드** 테이블에 기록됩니다. 각 흐름 레코드는 일련의 속성으로 구성됩니다. 이러한 속성에는 일정한 시간 간격에 해당하는 특정 클래스의 트래픽 흐름에 대한 데이터가 포함되어 있습니다. flowacct 속성 목록은 [표 32-4](#)를 참조하십시오.

흐름 계산은 SLA(서비스 단계 계약)에 정의된 청구 클라이언트에 특히 유용합니다. 흐름 계산을 사용하여 중요한 응용 프로그램에 대한 흐름 통계를 얻을 수도 있습니다. 이 절에는 Oracle Solaris 확장 계산 기능에 flowacct를 사용하여 트래픽 흐름에 대한 데이터를 얻는 작업이 포함되어 있습니다.

다음 정보는 이 장에서 다루지 않는 소스에서 제공됩니다.

- IPQoS 구성 파일에서 flowacct에 대한 작업문을 만드는 지침은 [442 페이지](#) “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”을 참조하십시오.
- flowacct 작동 방식에 대해 알아보려면 [461 페이지](#) “분류기 모듈”을 참조하십시오.
- 기술 정보는 flowacct(7ipp) 매뉴얼 페이지를 참조하십시오.

### ▼ 흐름 계산 데이터에 대한 파일을 만드는 방법

IPQoS 구성 파일에 flowacct 작업을 추가하기 전에 flowacct 모듈에서 흐름 레코드에 대한 파일을 만들어야 합니다. 이 용도로 acctadm 명령을 사용합니다. acctadm은 파일에 기본 속성 또는 확장 속성을 기록할 수 있습니다. 모든 flowacct 속성은 [표 32-4](#)에 나열됩니다. acctadm에 대한 자세한 내용은 [acctadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

#### 1 기본 흐름 계산 파일을 만듭니다.

다음 예에서는 [예 29-1](#)에서 구성된 고급 웹 서버에 대한 기본 흐름 계산 파일을 만드는 방법을 보여 줍니다.

```
# /usr/sbin/acctadm -e basic -f /var/ipqos/goldweb/account.info flow
```

acctadm -e                    -e 옵션과 함께 acctadm을 호출합니다. -e 옵션 뒤에 인수가 올 수 있습니다.

basic                        여덟 개의 기본 flowacct 속성에 대한 데이터만 파일에 기록되도록 지정합니다.

/var/ipqos/goldweb/account.info    flowacct의 흐름 레코드를 보관할 파일의 정규화된 경로 이름을 지정합니다.

flow                        흐름 계산을 사용으로 설정하도록 acctadm에 지시합니다.



**2 인수 없이 acctadm을 입력하여 IPQoS 시스템에서 흐름 계산에 대한 정보를 확인합니다.**

acctadm은 다음 출력을 생성합니다.

```
Task accounting: inactive
  Task accounting file: none
  Tracked task resources: none
  Untracked task resources: extended
    Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
  Untracked process resources: extended,host,mstate
    Flow accounting: active
    Flow accounting file: /var/ipqos/goldweb/account.info
  Tracked flow resources: basic
  Untracked flow resources: dsfield,ctime,lseen,projid,uid
```

마지막 네 개를 제외한 모든 항목은 Oracle Solaris 자원 관리자 기능에 사용됩니다. 다음 표에서는 IPQoS와 관련된 항목에 대해 설명합니다.

항목	설명
Flow accounting: active	흐름 계산이 켜져 있음을 나타냅니다.
Flow accounting file: /var/ipqos/goldweb/account.info	현재 흐름 계산 파일의 이름을 지정합니다.
Tracked flow resources: basic	기본 흐름 속성만 추적됨을 나타냅니다.
Untracked flow resources: dsfield,ctime,lseen,projid,uid	파일에서 추적되지 않는 flowacct 속성을 나열합니다.

**3 (옵션) 계산 파일에 확장 속성을 추가합니다.**

```
# acctadm -e extended -f /var/ipqos/goldweb/account.info flow
```

**4 (옵션) 계산 파일에 기본 속성만 기록하도록 되돌립니다.**

```
# acctadm -d extended -e basic -f /var/ipqos/goldweb/account.info
-d 옵션은 확장 계산을 사용 안함으로 설정합니다.
```

**5 흐름 계산 파일의 내용을 확인합니다.**

흐름 계산 파일 내용 확인 지침은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리의 “libexacct에 대한 Perl 인터페이스”**에서 확인할 수 있습니다.

- 참조**
- 확장 계산 기능에 대한 자세한 내용은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리의 4장, “확장 계정(개요)”**을 참조하십시오.
  - IPQoS 구성 파일에서 flowacct 매개변수를 정의하려면 432 페이지 **“IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”**을 참조하십시오.

- acctadm으로 만들어진 파일의 데이터를 인쇄하려면 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리의 “libexacct에 대한 Perl 인터페이스”**를 참조하십시오.

## 통계 정보 수집

kstat 명령을 사용하여 IPQoS 모듈에서 통계 정보를 생성할 수 있습니다. 다음 구문을 사용하십시오.

```
/bin/kstat -m ipqos-module-name
```

표 32-5와 같이 유효한 IPQoS 모듈 이름을 지정할 수 있습니다. 예를 들어, dscpmk 표시자가 생성한 통계를 보려면 다음 형식의 kstat를 사용합니다.

```
/bin/kstat -m dscpmk
```

자세한 기술 정보는 kstat(1M) 매뉴얼 페이지를 참조하십시오.

예 31-1 IPQoS에 대한 kstat 통계

다음은 flowacct 모듈에 대한 통계를 얻기 위해 kstat를 실행하여 발생할 수 있는 결과의 예입니다.

```
# kstat -m flowacct
module: flowacct                instance: 3
name:   Flowacct statistics      class:   flacct
        bytes_in_tbl             84
        crtime                   345728.504106363
        epackets                 0
        flows_in_tbl             1
        nbytes                   84
        npackets                 1
        snaptime                 345774.031843301
        usedmem                  256
```

class: flacct      트래픽 흐름이 속한 클래스의 이름(이 예의 경우 flacct)을 지정합니다.

bytes\_in\_tbl      흐름 테이블의 총 바이트 수입입니다. 총 바이트 수는 현재 흐름 테이블에 상주하는 모든 흐름 레코드의 합계(바이트)입니다. 이 흐름 테이블의 총 바이트 수는 84입니다. 테이블에 흐름이 없을 경우 bytes\_in\_tbl에 대한 값은 0입니다.

crtime            마지막으로 이 kstat 출력이 만들어진 시간입니다.

epackets          처리 중 오류가 발생한 패킷 수(이 예의 경우 0)입니다.

flows\_in\_tbl      흐름 테이블의 흐름 레코드 수(이 예의 경우 1)입니다. 테이블에 레코드가 없을 경우 flows\_in\_tbl에 대한 값은 0입니다.

예 31-1 IPQoS에 대한 kstat 통계 (계속)

nbytes	이 flowacct 작업 인스턴스가 확인한 총 바이트 수(이 예의 경우 84)입니다. 값에는 현재 흐름 테이블에 있는 바이트가 포함됩니다. 또한 값에는 시간이 초과되었으며 흐름 테이블에 더 이상 존재하지 않는 바이트가 포함됩니다.
npackets	이 flowacct 작업 인스턴스가 확인한 총 패킷 수(이 예의 경우 1)입니다. npackets에는 현재 흐름 테이블에 있는 패킷이 포함됩니다. 또한 npackets에는 시간이 초과되었으며 흐름 테이블에 더 이상 존재하지 않는 패킷이 포함됩니다.
usedmem	이 flowacct 인스턴스가 유지 관리하는 흐름 테이블에서 사용 중인 메모리(바이트)입니다. 이 예의 경우 usedmem 값은 256입니다. 흐름 테이블에 흐름 레코드가 없을 경우 usedmem에 대한 값은 0입니다.



## IPQoS 세부 정보(참조)

---

이 장에는 다음 IPQoS 항목에 대한 세부 정보를 제공하는 참조 자료가 포함되어 있습니다.

- 461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”
- 473 페이지 “IPQoS 구성 파일”
- 477 페이지 “ipqosconf 구성 유틸리티”

개요는 27 장, “IPQoS 소개(개요)”를 참조하십시오. 계획 정보는 28 장, “IPQoS 사용 네트워크 계획(작업)”을 참조하십시오. IPQoS 구성 절차는 29 장, “IPQoS 구성 파일 만들기(작업)”를 참조하십시오.

## IPQoS 아키텍처 및 Diffserv 모델

이 절에서는 IPQoS 아키텍처 및 IPQoS가 RFC 2475, *An Architecture for Differentiated Services* (<http://www.ietf.org/rfc/rfc2475.txt?number=2475>)에 정의된 차별화 서비스(Diffserv) 모델을 어떻게 구현하는지 설명합니다. IPQoS에는 Diffserv 모델의 다음 요소가 포함됩니다.

- 분류기
- 측정기
- 표시기

또한 IPQoS에는 흐름 계산 모듈 및 VLAN(virtual local area network) 장치와 함께 사용하기 위한 dlcsmk 표시기가 포함됩니다.

## 분류기 모듈

Diffserv 모델에서 **분류기**는 선택된 트래픽 흐름을 서로 다른 서비스 레벨이 적용되는 그룹으로 구성하기 위한 작업을 수행합니다. RFC 2475에서 정의된 분류기는 원래 경계 라우터를 위해 고안되었습니다. 반면, IPQoS 분류기 ipgpc은 로컬 네트워크의 내부에

있는 호스트의 트래픽 흐름을 처리하기 위해 고안되었습니다. 그러므로 IPQoS 시스템과 Diffserv 라우터가 모두 있는 네트워크는 더욱 뛰어난 차별화 서비스를 제공할 수 있습니다. `ipgpc`의 기술적인 설명은 `ipgpc(7ipp)` 매뉴얼 페이지를 참조하십시오.

`ipgpc` 분류기는 다음을 수행합니다.

1. IPQoS 사용 시스템의 IPQoS 구성 파일에 지정된 조건을 충족하는 트래픽 흐름을 선택합니다.  
QoS 정책은 패킷 헤더에 있어야 하는 다양한 조건을 정의합니다. 이러한 조건을 **선택기**라고 합니다. `ipgpc` 분류기는 이러한 선택기를 IPQoS 시스템에서 수신한 패킷의 헤더와 비교한 다음 `ipgpc`는 모든 일치하는 패킷을 선택합니다.
2. IPQoS 구성 파일에 정의된 대로 패킷 흐름을 동일 특성을 가진 네트워크 트래픽인 **클래스**로 구분합니다.
3. 패킷의 DS(차별화 서비스) 필드 값에 DSCP(차별화 서비스 코드 포인트)가 있는지 검사합니다.  
DSCP가 있으면 수신 트래픽이 전달 동작으로 발신자에 의해 표시되었는지 여부를 나타냅니다.
4. 특정 클래스의 패킷에 대해 IPQoS 구성 파일에서 지정된 추가 작업을 확인합니다.
5. 패킷을 IPQoS 구성 파일에서 지정된 다음 IPQoS 모듈에 전달하거나 패킷을 네트워크 스트림으로 돌려 보냅니다.

분류기의 개요는 [392 페이지 “분류기\(`ipgpc`\) 개요”](#)를 참조하십시오. IPQoS 구성 파일에서 분류기 호출에 대한 자세한 내용은 [473 페이지 “IPQoS 구성 파일”](#)을 참조하십시오.

## IPQoS 선택기

`ipgpc` 분류기는 IPQoS 구성 파일의 `filter` 절에서 사용할 수 있는 다양한 선택기를 지원합니다. 필터를 정의할 경우 항상 특정 클래스의 트래픽을 성공적으로 검색하는 데 필요한 최소 수의 선택기를 사용하십시오. 정의하는 필터 수에 따라 IPQoS 성능이 영향을 받을 수 있습니다.

다음 표는 `ipgpc`에 대해 사용 가능한 선택기를 나열합니다.

표 32-1 IPQoS 분류기에 대한 필터 선택기

선택기	인수	선택되는 정보
<code>saddr</code>	IP 주소 번호.	소스 주소입니다.
<code>daddr</code>	IP 주소 번호.	대상 주소입니다.
<code>sport</code>	<code>/etc/services</code> 에서 정의된 포트 번호 또는 서비스 이름.	트래픽 클래스가 발생한 소스 포트.

표 32-1 IPQoS 분류기에 대한 필터 선택기 (계속)

선택기	인수	선택되는 정보
dport	/etc/services에서 정의된 포트 번호 또는 서비스 이름.	트래픽 클래스가 향하는 대상 포트.
protocol	/etc/protocols에서 정의된 프로토콜 번호 또는 프로토콜 이름.	이 트래픽 클래스에서 사용할 프로토콜.
dsfield	0-63 값의 DSCP(DS 코드 포인트).	패킷에 적용할 전달 동작을 정의하는 DSCP. 이 매개변수가 지정되면 dsfield_mask 매개변수도 지정되어야 합니다.
dsfield_mask	0-255 값의 비트 마스크.	dsfield 선택기와 함께 사용됨. dsfield_mask는 dsfield 선택기에 적용되어 일치시킬 해당 비트를 결정합니다.
if_name	인터페이스 이름.	특정 클래스의 수신 또는 송신 트래픽에 사용될 인터페이스.
user	선택할 UNIX 사용자 ID 또는 사용자 이름 수. 패킷에 사용자 ID 또는 사용자 이름이 없으면 기본값 -1이 사용됩니다.	응용 프로그램에 제공된 사용자 ID.
projid	선택할 프로젝트 ID 수.	응용 프로그램에 제공된 프로젝트 ID.
priority	우선 순위 번호. 가장 낮은 우선 순위는 0입니다.	이 클래스의 패킷에 제공된 우선 순위. 우선 순위는 동일 클래스에 대해 필터의 중요도 순서를 정렬하는 데 사용됩니다.
direction	인수는 다음 중 하나가 될 수 있습니다.	IPQoS 시스템에서 패킷 흐름의 방향.
	LOCAL_IN	IPQoS 시스템에 로컬 입력 트래픽.
	LOCAL_OUT	IPQoS 시스템에 로컬 출력 트래픽.
	FWD_IN	전달할 입력 트래픽.
	FWD_OUT	전달할 출력 트래픽.
precedence	우선권 값. 가장 높은 우선권은 0입니다.	우선권은 동일 우선 순위를 가진 필터 순서를 정렬하는 데 사용됩니다.
ip_version	V4 또는 V6	패킷에서 사용되는 주소 지정 체계(IPv4 또는 IPv6).

## 측정기 모듈

측정기는 패킷당 기준으로 흐름의 전송 속도를 추적합니다. 그런 다음 측정기는 패킷이 구성된 매개변수를 준수하는지 여부를 확인합니다. 측정기 모듈은 패킷 크기, 구성된 매개변수 및 흐름 속도에 의존하는 작업 집합에서 패킷에 대한 다음 작업을 결정합니다.

측정기는 `tokenmt` 및 `tswtclmt`의 두 측정 모듈로 구성되며, IPQoS 구성 파일에서 구성합니다. 한 클래스에 대해 둘 중 하나의 모듈 또는 둘 다 구성할 수 있습니다.

측정 모듈을 구성할 때 속도에 대해 두 매개변수를 정의할 수 있습니다.

- `committed-rate` - 특정 클래스의 패킷에 대해 수용할 만한 전송 속도(bps, 초당 비트)를 정의합니다.
- `peak-rate` - 특정 클래스의 패킷에 대해 허용되는 최대 전송 속도(bps, 초당 비트)를 정의합니다.

패킷에 대한 측정 작업 결과는 세 가지 결과 중 하나가 될 수 있습니다.

- `green` - 패킷으로 인해 흐름이 약정된 속도 내에서 유지됩니다.
- `yellow` - 패킷으로 인해 흐름이 약정된 속도를 초과하지만 최대 속도를 초과하지는 않습니다.
- `red` - 패킷으로 인해 흐름이 최대 속도를 초과합니다.

IPQoS 구성 파일에서 서로 다른 작업으로 각 결과를 구성할 수 있습니다. 약정된 속도 및 최대 속도는 다음 절에서 설명합니다.

## tokenmt 측정 모듈

`tokenmt` 모듈은 **토큰 버킷**을 사용하여 흐름의 전송 속도를 측정합니다. `tokenmt`가 단일 속도 또는 두 가지 속도 측정기로 작동하도록 구성할 수 있습니다. `tokenmt` 작업 인스턴스는 트래픽 흐름이 구성된 매개변수를 준수하는지 여부를 결정하는 두 토큰 버킷을 유지 관리합니다.

[tokenmt\(7ipp\)](#) 매뉴얼 페이지에서 IPQoS가 토큰 측정기 패러다임을 구현하는 방식을 설명합니다. 토큰 버킷에 대한 일반적인 정보는 Kalevi Kilkki의 *Differentiated Services for the Internet* 및 여러 웹 사이트에서 찾을 수 있습니다.

`tokenmt`에 대한 구성 매개변수는 다음과 같습니다.

- `committed_rate` - 흐름의 약정된 속도 bps(초당 비트)로 지정합니다.
- `committed_burst` - 약정된 버스트 크기를 비트로 지정합니다. `committed_burst` 매개변수는 특정 클래스의 나가는 패킷이 약정된 속도로 네트워크에 전달될 수 있는 크기를 정의합니다.
- `peak_rate` - 최대 속도를 bps(초당 비트)로 지정합니다.
- `peak_burst` - 최대 또는 초과 버스트 크기를 비트로 지정합니다. `peak_burst` 매개변수는 약정된 속도를 초과하는 최대 버스트 크기를 트래픽 클래스에 부여합니다.
- `color_aware` - `tokenmt`에 대한 인식 모드를 설정합니다.
- `color_map` - DSCP 값을 녹색, 노란색 또는 빨간색으로 매핑하는 정수 배열을 정의합니다.



## 단일 속도 측정기로 tokenmt 구성

tokenmt를 단일 속도 측정기로 구성하려면 IPQoS 구성 파일에서 tokenmt에 대해 `peak_rate` 매개변수를 지정하지 마십시오. 단일 속도 tokenmt 인스턴스가 빨간색, 녹색 또는 노란색 결과를 가지도록 구성하려면 `peak_burst` 매개변수를 지정해야 합니다. `peak_burst` 매개변수를 사용하지 않을 경우 tokenmt가 빨간색 결과 또는 녹색 결과만 가지도록 구성할 수 있습니다. 두 결과를 가지는 단일 속도 tokenmt의 예는 [예 29-3](#)을 참조하십시오.

tokenmt가 단일 속도 측정기로 작동하는 경우 `peak_burst` 매개변수는 실제로 초과 버스트 크기입니다. `committed_rate` 및 `committed_burst` 또는 `peak_burst`는 0이 아닌 양의 정수여야 합니다.

## 두 속도 측정기로 tokenmt 구성

tokenmt를 두 속도 측정기로 구성하려면 IPQoS 구성 파일에서 tokenmt 작업에 대해 `peak_rate` 매개변수를 지정합니다. 두 속도 tokenmt는 항상 빨간색, 노란색 및 녹색의 세 가지 결과를 가집니다. `committed_rate`, `committed_burst` 및 `peak_burst` 매개변수는 0이 아닌 양의 정수여야 합니다.

## 색상을 인식하도록 tokenmt 구성

두 속도 tokenmt가 색상을 인식하도록 구성하려면 “색상 인식”을 구체적으로 추가하는 매개변수를 추가해야 합니다. 다음은 색상을 인식하도록 tokenmt를 구성하는 예제 작업 명령문입니다.

예 32-1 IPQoS 구성 파일에 대한 색상 인식 tokenmt 작업

```
action {
  module tokenmt
  name meter1
  params {
    committed_rate 4000000
    peak_rate 8000000
    committed_burst 4000000
    peak_burst 8000000
    global_stats true
    red_action_name continue
    yellow_action_name continue
    green_action_name continue
    color_aware true
    color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
  }
}
```

`color_aware` 매개변수를 `true`로 설정하여 색상 인식을 사용으로 설정해야 합니다. 색상 인식 측정기로서 tokenmt는 패킷이 이전 tokenmt 작업에 의해 이미 빨간색, 노란색 또는 녹색으로 표시되었다고 간주합니다. 색상 인식 tokenmt는 두 속도 측정기에 대한 매개변수와 함께 패킷 헤더의 DSCP를 사용하여 패킷을 평가합니다.

`color_map` 매개변수에는 패킷 헤더의 DSCP가 매핑되는 배열이 포함됩니다. 다음 `color_map` 배열을 고려하십시오.

```
color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
```

DSCP 0-20 및 22의 패킷은 녹색으로 매핑됩니다. DSCP 21 및 23-42의 패킷은 빨간색으로 매핑됩니다. DSCP 43-63의 패킷은 노란색으로 매핑됩니다. `tokenmt`는 기본 색상 맵을 유지 관리합니다. 하지만 `color_map` 매개변수를 사용하여 필요에 따라 기본값을 변경할 수 있습니다.

`color_action_name` 매개변수에서 `continue`를 지정하여 패킷 처리를 완료할 수 있습니다. 또는 패킷을 표시기 작업에 보내는 인수를 추가할 수 있습니다(예: `yellow_action_name mark22`).

## tswtclmt 측정 모듈

tswtclmt 측정 모듈은 시간 기반 **속도 추정기**를 사용하여 트래픽 클래스에 대한 평균 대역폭을 추정합니다. tswtclmt는 항상 세 가지 결과 측정기로 작동합니다. 속도 추정기는 흐름의 도착 추정 속도를 제공합니다. 이 속도는 지정된 기간(**시간 창**) 동안 트래픽 스트림의 실행 평균 대역폭에 근접해야 합니다. 속도 추정 알고리즘은 RFC 2859, *A Time Sliding Window Three Colour Marker*에서 가져옵니다.

다음 매개변수를 사용하여 tswtclmt를 구성합니다.

- `committed_rate` - 약정된 속도를 bps(초당 비트)로 지정합니다.
- `peak_rate` - 최대 속도를 bps(초당 비트)로 지정합니다.
- `window` - 평균 대역폭 내역이 보관되는 시간 창을 밀리초로 정의합니다.

tswtclmt에 대한 기술적인 세부 정보는 [tswtclmt\(7ipp\)](#) 매뉴얼 페이지를 참조하십시오. tswtclmt와 유사한 속도 샤퍼(Shaper)에 대한 일반적인 정보는 [RFC 2963, A Rate Adaptive Shaper for Differentiated Services](#) (<http://www.ietf.org/rfc/rfc2963.txt?number=2963>)를 참조하십시오.

## 표시기 모듈

IPQoS에는 `dscpmk` 및 `dlcosmk`의 두 표시기 모듈이 포함됩니다. 이 절에서는 두 표시기 사용에 대한 정보가 포함되어 있습니다. `dlcosmk`만 VLAN 장치가 있는 IPQoS 시스템에 대해 사용할 수 있으므로 일반적으로는 `dscpmk`를 사용해야 합니다.

`dscpmk`에 대한 기술적인 정보는 `dscpmk(7ipp)` 매뉴얼 페이지를 참조하십시오. `dlcosmk`에 대한 기술적인 정보는 `dlcosmk(7ipp)` 매뉴얼 페이지를 참조하십시오.

## 패킷 전달을 위해 dscpmk 표시기 사용

표시기는 흐름이 분류기 또는 측정 모듈에서 처리된 후 트래픽 흐름을 수신합니다. 표시기는 트래픽을 전달 동작으로 표시합니다. 이 전달 동작은 흐름이 IPQoS 시스템을 떠난 후 수행할 작업입니다. 트래픽 클래스에 대해 수행할 전달 동작은 **PHB(휴별**

동작)에서 정의됩니다. PHB는 다른 트래픽 클래스와 관련하여 해당 클래스의 우선권 흐름을 나타내는 우선 순위를 트래픽 클래스에 지정합니다. PHB는 IPQoS 시스템의 인접 네트워크에 대한 전달 동작만 제어합니다. PHB에 대한 자세한 내용은 [397 페이지 “흡별 동작”](#)을 참조하십시오.

**패킷 전달**은 특정 클래스의 트래픽을 네트워크의 다음 대상으로 보내는 프로세스입니다. IPQoS 시스템과 같은 호스트의 경우, 패킷은 호스트에서 로컬 네트워크 스트림으로 전달됩니다. Diffserv 라우터의 경우, 패킷은 로컬 네트워크에서 라우터의 다음 홉으로 전달됩니다.

표시기는 패킷 헤더의 DS 필드를 IPQoS 구성 파일에서 정의된 잘 알려진 전달 동작으로 표시합니다. 그러면 IPQoS 시스템 및 후속 Diffserv 인식 시스템은 표시가 바뀔 때까지 DS 필드에 나타난 대로 트래픽을 전달합니다. PHB를 지정하기 위해 IPQoS 시스템은 패킷 헤더의 DS 필드에 값을 표시합니다. 이 값을 DSCP(차별화 서비스 코드 포인트)라고 합니다. Diffserv 아키텍처는 서로 다른 DSCP를 사용하는 두 가지 유형의 전달 동작인 EF 및 AF를 정의합니다. DSCP에 대한 개요는 [397 페이지 “DS 코드 포인트”](#)를 참조하십시오.

IPQoS 시스템은 트래픽 흐름에 대해 DSCP를 읽고 다른 송신 트래픽 흐름과 관련하여 흐름의 우선권을 평가합니다. 그런 다음 IPQoS 시스템은 모든 동시 트래픽 흐름에 우선 순위를 지정하고 각 흐름을 우선 순위에 따라 네트워크로 보냅니다.

Diffserv 라우터는 송신 트래픽 흐름을 수신하고 패킷 헤더의 DS 필드를 읽습니다. DSCP는 라우터가 동시 트래픽 흐름에 우선 순위를 지정하고 일정을 예약하도록 합니다. 라우터는 PHB로 지정된 우선 순위에 따라 각 흐름을 전달합니다. 후속 홉의 Diffserv 인식 시스템도 동일한 PHB를 인식하지 못하면 PHB는 네트워크의 경계 라우터를 벗어나서 적용할 수 없습니다.

## EF(빠른 전달) PHB

**빠른 전달(EF)**은 권장 EF 코드 포인트 46(101110)의 패킷이 네트워크로 전송 시 사용할 가능한 가장 좋은 취급을 받도록 보장합니다. 빠른 전달은 임대 회선과 비교되기도 합니다. 46(101110) 코드 포인트의 패킷은 패킷의 대상으로 향하는 모든 Diffserv 경로에서 선호 취급이 보장됩니다. EF에 대한 기술적인 정보는 RFC 2598, *An Expedited Forwarding PHB*를 참조하십시오.

## AF(보장 전달) PHB

**보장 전달(AF)**은 표시기에 지정할 수 있는 네 가지 서로 다른 클래스의 전달 동작을 제공합니다. 다음 표는 클래스, 각 클래스에 제공되는 세 가지 삭제 우선권 및 각 우선권과 연결된 권장 DSCP를 보여줍니다. 각 DSCP는 해당 AF 값, 십진수 값 및 이진수 값으로 표시됩니다.

표 32-2 보장 전달 코드 포인트

	클래스 1	클래스 2	클래스 3	클래스 4
낮은 삭제 우선권	AF11 = 10 (001010)	AF21 = 18 (010010)	AF31 = 26 (011010)	AF41 = 34 (100010)
중간 삭제 우선권	AF12 = 12 (001100)	AF22 = 20 (010100)	AF32 = 28 (011100)	AF42 = 36 (100100)
높은 삭제 우선권	AF13 = 14 (001110)	AF23 = 22 (010110)	AF33 = 30 (011110)	AF43 = 38 (100110)

모든 Diffserv 인식 시스템에서는 AF 코드 포인트를 기준으로 사용하여 서로 다른 클래스의 트래픽에 차별화된 전달 동작을 제공할 수 있습니다.

이러한 패킷이 Diffserv 라우터에 도달하면 라우터는 대기열에 있는 다른 트래픽의 DSCP와 함께 패킷의 코드 포인트를 평가합니다. 그런 다음 라우터는 사용 가능한 대역폭 및 패킷의 DSCP로 지정된 우선 순위에 따라 패킷을 전달하거나 삭제합니다. EF PHB로 표시된 패킷은 다양한 AF PHB로 표시된 패킷에 비해 대역폭이 보장됩니다.

패킷이 예상한 대로 전달되도록 하려면 네트워크의 IPQoS 시스템과 Diffserv 라우터 사이에 패킷 표시를 조정하십시오. 예를 들어, 네트워크의 IPQoS 시스템이 AF21(010010), AF13(001110), AF43(100110) 및 EF(101110) 코드 포인트로 패킷을 표시한다고 가정해 보겠습니다. 그러면 AF21, AF13, AF43 및 EF DSCP를 Diffserv 라우터의 해당 파일에 추가해야 합니다.

AF 코드 포인트 표시의 기술적인 설명은 RFC 2597을 참조하십시오. 라우터 제조업체 Cisco Systems 및 Juniper Networks의 회사 웹 사이트에 가면 자세한 AF PHB 설정 정보가 제공됩니다. 이러한 정보를 활용하여 IPQoS 시스템 및 라우터에 대한 AF PHB를 정의할 수 있습니다. 또한 라우터 제조업체의 설명서에는 자사 장비에서 DS 코드 포인트 설정에 대한 지침이 포함되어 있습니다.

## 표시기에 DSCP 제공

DSCP는 6비트 길이입니다. DS 필드는 1바이트 길이입니다. DSCP를 정의할 때 표시기는 패킷 헤더의 처음 중요 6비트를 DS 코드 포인트로 표시합니다. 나머지 덜 중요한 2비트는 사용되지 않습니다.

DSCP를 정의하려면 표시기 작업 매개변수 내에서 다음 매개변수를 사용합니다.

```
dscp_map{0-63:DS_codepoint}
```

dscp\_map 매개변수는 (DSCP) 값으로 채우는 64 요소 배열입니다. dscp\_map은 들어오는 DSCP를 dscpmk 표시기에 의해 적용된 나가는 DSCP로 매핑하는 데 사용됩니다.

DSCP 값은 십진수 형식의 `dscp_map`으로 지정해야 합니다. 예를 들어, EF 코드 포인트 101110은 십진수 값 46으로 변환해야 하며, 결과적으로 `dscp_map{0-63:46}`이 됩니다. AF 코드 포인트의 경우, 표 32-2에 나온 다양한 코드 포인트를 `dscp_map`에서 사용할 십진수 형식으로 변환해야 합니다.

## VLAN 장치에서 `dlcosmk` 표시기 사용

`dlcosmk` 표시기 모듈은 데이터그램의 MAC 헤더에 전달 동작을 표시합니다. VLAN 인터페이스가 있는 IPQoS 시스템에서만 `dlcosmk`를 사용할 수 있습니다.

`dlcosmk`는 VLAN 태그로 알려진 4바이트를 MAC 헤더에 추가합니다. VLAN 태그에는 IEEE 801.D 표준에서 정의된 3비트 사용자 우선 순위 값이 포함됩니다. VLAN을 이해하는 Diffserv 인식 스위치는 데이터그램의 사용자 우선 순위 필드를 읽을 수 있습니다. 801.D 사용자 우선 순위 값은 잘 알려지고 상용 스위치에서 이해할 수 있는 CoS(서비스 클래스) 표시를 구현합니다.

다음 표에 나열된 서비스 클래스 표시를 정의하여 `dlcosmk` 표시기 작업에서 사용자 우선 순위 값을 사용할 수 있습니다.

표 32-3 801.D 사용자 우선 순위 값

서비스 클래스	정의
0	최선 조건
1	백그라운드
2	여분
3	최우선 조건
4	제어 로드
5	100ms 대기 시간 미만의 비디오
6	10ms 대기 시간 미만의 비디오
7	네트워크 제어

`dlcosmk`에 대한 자세한 내용은 `dlcosmk(7ipp)` 매뉴얼 페이지를 참조하십시오.

## VLAN 장치가 있는 시스템에 대한 IPQoS 구성

이 절에서는 VLAN 장치가 있는 시스템에서 IPQoS를 구현하는 방법을 보여주는 단순한 네트워크 시나리오를 소개합니다. 시나리오에는 스위치로 연결된 `machine1` 및 `machine2`의 두 IPQoS 시스템이 포함됩니다. `machine1`의 VLAN 장치는 IP 주소 10.10.8.1을 가집니다. `machine2`의 VLAN 장치는 IP 주소 10.10.8.3을 가집니다.

`machine1`에 대한 다음 IPQoS 구성 파일은 스위치를 거쳐 `machine2`로 이동하는 트래픽을 표시하기 위한 간단한 솔루션을 보여줍니다.

예 32-2 VLAN 장치가 있는 시스템에 대한 IPQoS 구성 파일

```

fmt_version 1.0
action {
    module ipgpc
    name ipgpc.classify

    filter {
        name myfilter2
        daddr 10.10.8.3
        class myclass
    }

    class {
        name myclass
        next_action mark4
    }
}

action {
    name mark4
    module dlcsmk
    params {
        cos 4
        next_action continue
    }
    global_stats true
}

```

이 구성에서 machine2의 VLAN 장치를 대상으로 하는 machine1의 모든 트래픽은 dlcsmk 표시기로 전달됩니다. mark4 표시기 작업은 dlcsmk가 VLAN 표시를 CoS가 4인 myclass 클래스의 데이터그램에 추가하도록 지시합니다. 사용자 우선 순위 값 4는 두 시스템 사이에 있는 스위치가 machine1의 myclass 트래픽 흐름에 제어 로드 전달을 제공해야 한다는 것을 나타냅니다.

## flowacct 모듈

IPQoS flowacct 모듈은 트래픽 흐름에 대한 정보를 기록하며, 이 프로세스를 **흐름 계산**이라고 합니다. 흐름 계산은 고객 청구 또는 특정 클래스에 대한 트래픽의 양 평가 목적으로 사용될 수 있는 데이터를 생성합니다.

흐름 계산은 선택 사항입니다. flowacct는 일반적으로 측정되거나 표시된 트래픽 흐름이 네트워크 스트림으로 보내지기 전에 만날 수 있는 마지막 모듈입니다. Diffserv 모델에서 flowacct의 위치에 대한 그림은 [그림 27-1](#)을 참조하십시오. flowacct에 대한 기술적인 세부 정보는 flowacct(7ipp) 매뉴얼 페이지를 참조하십시오.

흐름 계산을 사용으로 설정하려면 flowacct와 함께 Oracle Solaris exacct 계산 기능 및 acctadm 명령을 사용해야 합니다. 흐름 계산 설정에 대한 전체 단계는 [455 페이지 “흐름 계산 설정\(작업 맵\)”](#)을 참조하십시오.

## flowacct 매개변수

flowacct 모듈은 흐름 레코드로 구성된 흐름 테이블에서 흐름에 대한 정보를 수집합니다. 테이블의 각 항목은 하나의 흐름 레코드를 포함합니다. 흐름 테이블을 표시할 수는 없습니다.

IPQoS 구성 파일에서 다음 flowacct 매개변수를 정의하여 흐름 레코드를 측정하고 레코드를 흐름 테이블에 기록합니다.

- **timer** - 시간 초과된 흐름이 흐름 테이블에서 제거되고 acctadm으로 만들어진 파일에 기록되는 간격을 밀리초로 정의합니다.
- **timeout** - 흐름이 시간 초과되기 전에 패킷 흐름이 비활성화되어야 하는 기간을 밀리초로 정의합니다.

---

주 - timer 및 timeout이 서로 다른 값을 가지도록 구성할 수 있습니다.

---

- **max\_limit** - 흐름 테이블에 저장할 수 있는 흐름 레코드 수에 대한 상한 제한을 둡니다.

flowacct 매개변수가 IPQoS 구성 파일에서 사용되는 예는 [442 페이지](#) “IPQoS 구성 파일에서 흐름 제어를 구성하는 방법”을 참조하십시오.

## 흐름 테이블

flowacct 모듈은 flowacct 인스턴스에서 본 모든 패킷 흐름을 기록하는 흐름 테이블을 유지 관리합니다. 흐름은 flowacct 8-튜플을 포함하는 다음 매개변수로 식별됩니다.

- 소스 주소
- 대상 주소
- 소스 포트
- 대상 포트
- DSCP
- 사용자 ID
- 프로젝트 ID
- 프로토콜 번호

흐름에 대한 8-튜플의 모든 매개변수가 동일하게 유지될 경우 흐름 테이블은 하나의 항목만 포함합니다. max\_limit 매개변수는 흐름 테이블에 포함될 수 있는 항목 수를 결정합니다.

흐름 테이블은 timer 매개변수에 대해 IPQoS 구성 파일에서 정의된 간격으로 검사됩니다. 기본값은 15초입니다. 흐름은 IPQoS 구성 파일의 timeout 간격 이상 동안 IPQoS 시스템에서 해당 패킷을 볼 수 없을 때 “시간 초과”됩니다. 기본 시간 초과 간격은 60초입니다. 그런 다음 시간 초과된 항목은 acctadm 명령으로 만들어진 계산 파일에 기록됩니다.

## flowacct 레코드

flowacct 레코드에는 다음 표에 설명된 속성이 포함됩니다.

표 32-4 flowacct 레코드의 속성

속성 이름	속성 내용	유형
src-addr-address-type	발신자의 소스 주소. <i>address-type</i> 은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
dest-addr-address-type	패킷에 대한 대상 주소. <i>address-type</i> 은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
src-port	흐름이 발생한 소스 포트.	기본
dest-port	이 흐름이 향하는 대상 포트 번호.	기본
protocol	흐름에 대한 프로토콜 번호.	기본
total-packets	흐름의 패킷 수.	기본
total-bytes	흐름의 바이트 수.	기본
action-name	이 흐름을 기록한 flowacct 작업의 이름.	기본
creation-time	흐름에 대한 패킷이 flowacct에 의해 처음으로 목격된 시간.	확장 전용
last-seen	흐름의 패킷이 마지막으로 목격된 시간.	확장 전용
diffserv-field	흐름의 나가는 패킷 헤더에 있는 DSCP.	확장 전용
user	응용 프로그램에서 가져온 UNIX 사용자 ID 또는 사용자 이름.	확장 전용
projid	응용 프로그램에서 가져온 프로젝트 ID.	확장 전용

## flowacct 모듈에서 acctadm 사용

acctadm 명령을 사용하여 flowacct로 생성된 다양한 흐름 레코드를 저장할 파일을 만듭니다. acctadm은 확장 계산 기능과 함께 작동합니다. acctadm에 대한 기술적인 정보는 [acctadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

flowacct 모듈은 흐름을 관찰하고 흐름 테이블을 레코드로 채웁니다. 그런 다음 flowacct는 timer에서 지정된 간격으로 해당 매개변수 및 속성을 평가합니다. 패킷이 last\_seen + timeout 값 이상 동안 보이지 않으면 패킷은 시간 초과됩니다. 모든 시간 초과된 항목은 흐름 테이블에서 삭제됩니다. 그런 다음 이러한 항목은 timer 매개변수에 지정된 간격이 경과할 때마다 계산 파일에 기록됩니다.

flowacct 모듈에서 사용할 acctadm을 호출하려면 다음 구문을 사용합니다.



```
acctadm -e file-type -f filename flow
```

`acctadm -e`     -e 옵션과 함께 `acctadm`을 호출합니다. -e는 리소스 목록이 있음을 나타냅니다.

`file-type`     수집할 속성을 지정합니다. `file-type`은 `basic` 또는 `extended`로 바뀌어야 합니다. 각 파일 유형의 속성 목록은 [표 32-4](#)를 참조하십시오.

`-f file-name`   흐름 레코드를 보관할 `file-name` 파일을 만듭니다.

`flow`           `acctadm`이 IPQoS에서 실행됨을 나타냅니다.

## IPQoS 구성 파일

이 절에서는 IPQoS 구성 파일의 부분에 대한 전체 세부 정보가 포함되어 있습니다. IPQoS 부트 시 활성화되는 정책은 `/etc/inet/ipqosinit.conf` 파일에 저장됩니다. 이 파일을 편집할 수 있지만 새 IPQoS 시스템의 경우 가장 좋은 방법은 다른 이름으로 구성 파일을 만드는 것입니다. IPQoS 구성을 적용하고 디버깅하는 작업은 [29 장, “IPQoS 구성 파일 만들기\(작업\)”](#)를 참조하십시오.

IPQoS 구성 파일의 구문은 [예 32-3](#)을 참조하십시오. 예에서는 다음 규약을 사용합니다.

- **컴퓨터 스타일 유형** - 구성 파일의 부분을 설명하기 위해 제공되는 구문 정보입니다. 컴퓨터 스타일 유형으로 나타나는 텍스트는 입력하지 않습니다.
- **굵은체 유형** - IPQoS 구성 파일에 입력해야 하는 리터럴 텍스트입니다. 예를 들어, IPQoS 구성 파일은 항상 **fmt\_version**으로 시작해야 합니다.
- **기울임꼴 유형** - 구성에 대한 설명 정보로 바꾸는 변수 텍스트입니다. 예를 들어, *action-name* 또는 *module-name*은 항상 조직에 해당하는 정보로 바뀌어야 합니다.

예 32-3 IPQoS 구성 파일의 구문

```
file_format_version ::= fmt_version version

action_clause ::= action {
    name action-name
    module module-name
    params_clause | ""
    cf-clauses
}
action_name ::= string
module_name ::= ipgpc | dlcosmk | dscpmk | tswtclmt | tokenmt | flowacct

params_clause ::= params {
    parameters
    params-stats | ""
}
parameters ::= prm-name-value parameters | ""
prm_name_value ::= param-name param-value

params_stats ::= global-stats boolean
```

## 예 32-3 IPQoS 구성 파일의 구문 (계속)

```

cf_clauses ::= class-clause cf-clauses |
               filter-clause cf-clauses | ""

class_clause ::= class {
    name class-name
    next_action next-action-name
    class-stats | ""
}
class_name   ::= string
next_action_name ::= string
class_stats  ::= enable_stats boolean
boolean      ::= TRUE | FALSE

filter_clause ::= filter {
    name filter-name
    class class-name
    parameters
}
filter_name   ::= string

```

IPQoS 구성 파일의 각 주요 부분을 설명하는 나머지 텍스트입니다.

## action 명령문

action 명령문을 사용하여 461 페이지 “IPQoS 아키텍처 및 Diffserv 모델”에 설명된 다양한 IPQoS 모듈을 호출합니다.

IPQoS 구성 파일을 만들 때는 항상 버전 번호로 시작해야 합니다. 그리고 다음 action 명령문을 추가하여 분류기를 호출합니다.

```

fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
}

```

분류기 action 명령문 다음에는 params 절 또는 class 절이 옵니다.

기타 모든 action 명령문에 대해 다음 구문을 사용합니다.

```

action {
    name action-name
    module module-name
    params-clause | ""
    cf-clauses
}

```

<code>name</code> <i>action_name</i>	작업에 이름을 지정합니다.
<code>module</code> <i>module_name</i>	호출할 IPQoS 모듈을 식별합니다. 표 32-5의 모듈 중 하나이어야 합니다.
<i>params_clause</i>	분류기가 처리할 매개변수가 될 수 있습니다(예: 전역 통계 또는 처리할 다음 작업).
<i>cf_clauses</i>	0개 이상의 <code>class</code> 절 또는 <code>filter</code> 절 집합입니다.

## 모듈 정의

모듈 정의는 `action` 명령문에서 매개변수를 처리할 모듈을 나타냅니다. IPQoS 구성 파일에는 다음 모듈이 포함될 수 있습니다.

표 32-5 IPQoS 모듈

모듈 이름	정의
<code>ipgpc</code>	IP 분류기
<code>dscpmk</code>	IP 패킷에서 DSCP를 만드는 데 사용할 표시기
<code>dlcosmk</code>	VLAN 장치에서 사용할 표시기
<code>tokenmt</code>	토큰 버킷 측정기
<code>tswtclmt</code>	시간별 창 측정기
<code>flowacct</code>	흐름 계산 모듈

## class 절

각 트래픽 클래스에 대해 `class` 절을 정의합니다.

이 구문을 사용하여 IPQoS 구성의 나머지 절을 정의합니다.

```
class {
    name class-name
    next_action next-action-name
}
```

특정 클래스에 대한 통계 수집을 사용으로 설정하려면 먼저 `ipgpc.classify action` 명령문에서 전역 통계를 사용으로 설정해야 합니다. 자세한 내용은 474 페이지 “[action 명령문](#)”을 참조하십시오.

클래스에 대한 통계 수집을 설정할 때는 항상 `enable_stats TRUE` 명령문을 사용합니다. 클래스에 대한 통계를 수집할 필요가 없는 경우 `enable_stats FALSE`를 지정할 수 있습니다. 또는 `enable_stats` 명령문을 제거할 수 있습니다.

명시적으로 정의하지 않은 IPQoS 사용 네트워크에 대한 트래픽은 **기본 클래스**로 들어갑니다.

## filter 절

**필터**는 트래픽 흐름을 클래스로 그룹화하는 선택기로 구성됩니다. 이러한 선택기는 `class` 절에서 만들어진 클래스의 트래픽에 적용될 조건을 구체적으로 정의합니다. 패킷이 가장 높은 우선 순위 필터의 모든 선택기와 일치할 경우 해당 패킷은 필터 클래스의 멤버로 간주됩니다. `ipgpc` 분류기에서 사용할 수 있는 전체 선택기 목록은 [표 32-1](#)을 참조하십시오.

다음 구문을 가지는 **filter 절**을 사용하여 IPQoS 구성 파일에서 필터를 정의합니다.

```
filter {
    name filter-name
    class class-name
    parameters (selectors)
}
```

## params 절

**params 절**에는 작업 명령문에서 정의된 모듈에 대한 처리 지침이 포함됩니다. **params 절**에 대해 다음 구문을 사용합니다.

```
params {
    parameters
    params-stats | ""
}
```

**params 절**에서 모듈에 적용 가능한 매개변수를 사용합니다.

**params 절**의 `params-stats` 값은 `global_stats TRUE` 또는 `global_stats FALSE`입니다. `global_stats TRUE` 지침은 전역 통계가 호출되는 **action** 명령문에 대해 UNIX 스타일 통계를 설정합니다. 통계는 `kstat` 명령을 사용하여 볼 수 있습니다. 클래스별 통계를 사용으로 설정하려면 먼저 **action** 명령문 통계를 사용으로 설정해야 합니다.

## ipqosconf 구성 유틸리티

ipqosconf 유틸리티를 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. ipqosconf는 다음 작업을 수행합니다.

- 구성 파일을 IPQoS 커널 모듈에 적용합니다(ipqosconf -a *filename*).
- 커널에 현재 상주하는 IPQoS 구성 파일을 나열합니다(ipqosconf -l).
- 시스템이 재부트될 때마다 현재 IPQoS 구성을 읽고 적용되도록 합니다(ipqosconf -c).
- 현재 IPQoS 커널 모듈을 비웁니다(ipqosconf -f).

기술적인 정보는 [ipqosconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.



# 용어집

---

3DES	3중 DES를 참조하십시오.
3중 DES	3중 데이터 암호화 표준(Triple-Data Encryption Standard). 대칭 키 암호화 방법입니다. 3중 DES는 168비트의 키 길이가 필요합니다. 3중 DES를 3DES로 쓰기도 합니다.
AES	고급 암호화 표준(Advanced Encryption Standard). 대칭 128비트 블록 데이터 암호화 기술입니다. 미국 정부는 2000년 10월 알고리즘의 Rijndael 변형을 암호화 표준으로 채택했습니다. AES가 정부 표준으로 DES 암호화를 대체합니다.
Blowfish	32-448비트의 가변 길이 키를 사용하는 대칭 블록 암호화 알고리즘입니다. 저작자인 Bruce Schneier에 따르면, Blowfish는 키를 자주 바꾸지 않는 응용 프로그램에 최적화되어 있습니다.
CA	인증 기관(CA)을 참조하십시오.
CIDR (classless inter-domain routing) 주소	네트워크 클래스(클래스 A, B, C)에 기반하지 않는 IPv4 주소 형식입니다. CIDR 주소는 32비트 길이입니다. 표준 IPv4의 점으로 구분된 십진수 표기법 형식에 네트워크 접두어가 추가됩니다. 이 접두어는 네트워크 번호 및 네트워크 마스크를 정의합니다.
CRL (인증서 해지 목록)	CA에 의해 해지된 공개 키 인증서 목록입니다. CRL은 IKE를 통해 유지 관리하는 CRL 데이터베이스에 저장됩니다.
DES	데이터 암호화 표준(Data Encryption Standard). 1975년에 개발되고 1981년에 ANSI에 의해 ANSI X.3.92로 표준화된 대칭 키 암호화 방법입니다. DES에서는 56비트 키를 사용합니다.
Diffie-Hellman 알고리즘	공개 키 암호화라고도 합니다. 1976년에 Diffie와 Hellman이 공동 개발한 비대칭 암호화 기법인 것입니다. 이 프로토콜을 사용하면 어떤 예비 보안 없이도 두 사용자가 비보안 매체를 통해 보안 키를 교환할 수 있습니다. Diffie-Hellman은 IKE 프로토콜에서 사용됩니다.
diffserv 모델	IP 네트워크에서 차등화 서비스를 구현하기 위한 IETF(Internet Engineering Task Force) 구조 표준입니다. 주 모듈에는 분류자, 측정자, 표시자, 스케줄러, 삭제자가 있습니다. IPQoS는 분류자, 측정자, 표시자 모듈을 구현합니다. diffserv 모델은 RFC 2475 <i>An Architecture for Differentiated Services</i> 에 설명됩니다.
DOI (Domain of Interpretation)	DOI는 데이터 형식, 네트워크 트래픽 교환 유형 및 보안 관련 정보의 이름 지정 규약을 정의합니다. 보안 관련 정보의 예로 보안 정책, 암호화 알고리즘, 암호화 모드 등이 있습니다.
DS 코드점 (DSCP)	IP 헤더의 DS 필드에 포함될 때 패킷의 전달 방법을 나타내는 6비트 값입니다.

<b>DSA</b>	디지털 서명 알고리즘(Digital Signature Algorithm). 512-4096비트의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 1024비트까지 지원합니다. DSA는 입력에 <a href="#">SHA-1</a> 을 사용합니다.
<b>ESP (보안 페이로드 캡슐화)</b>	데이터그램에 무결성 및 기밀성을 제공하는 확장 헤더입니다. ESP는 IP 보안 구조(IPsec)의 5개 구성 요소 중 하나입니다.
<b>HMAC</b>	메시지 인증을 위해 입력한 해싱 방법입니다. HMAC는 보안 키 인증 알고리즘입니다. HMAC는 비밀 공유 키와 조합하여 MD5 또는 SHA-1과 같은 반복 암호화 해시 기능과 함께 사용합니다. 기본 해시 기능의 등록 정보에 따라 HMAC의 암호화 강도가 달라집니다.
<b>ICMP</b>	인터넷 제어 메시지 프로토콜(Internet Control Message Protocol). 오류를 처리하고 제어 메시지를 교환하는 데 사용됩니다.
<b>ICMP 에코 요청 패킷</b>	인터넷에서 응답을 간청하기 위해 시스템으로 보낸 패킷입니다. 이러한 패킷을 흔히 "ping" 패킷이라고 합니다.
<b>IKE</b>	인터넷 키 교환(Internet Key Exchange). IKE는 IPsec 보안 연관(SA)에 대한 인증된 키 관련 자료의 프로비전을 자동화합니다.
<b>IP</b>	<a href="#">IP(인터넷 프로토콜)</a> , <a href="#">IPv4</a> , <a href="#">IPv6</a> 을 참조하십시오.
<b>IP-in-IP 캡슐화</b>	IP 패킷 안에 IP 패킷을 터널링하는 방식입니다.
<b>IP 데이터그램</b>	IP를 통해 전달된 정보의 패킷입니다. IP 데이터그램은 헤더 및 데이터를 포함합니다. 헤더는 데이터그램의 소스 및 대상 주소를 포함합니다. 헤더의 다른 필드를 통해 대상에서 데이터와 동반 데이터그램을 식별하고 재검파일할 수 있습니다.
<b>IP 링크</b>	링크 계층에서 노드가 통신할 수 있는 통신 설비 또는 매체입니다. 링크 계층은 IPv4/IPv6 바로 아래의 계층입니다. 그 예로 이더넷(단순/브릿지된) 또는 ATM 네트워크가 있습니다. 하나 이상의 IPv4 서브넷 번호 또는 접두어가 IP 링크에 지정됩니다. 서브넷 번호 또는 접두어를 여러 개의 IP 링크에 지정할 수 없습니다. ATM LANE에서 IP 링크는 단일 에뮬레이트된 LAN입니다. ARP를 사용할 때 ARP 프로토콜의 범위는 단일 IP 링크입니다.
<b>IP 스택</b>	TCP/IP를 종종 "스택"이라고도 합니다. 이것은 데이터 교환의 클라이언트측과 서버측 양쪽에서 모든 데이터가 전달되는 계층(TCP, IP 및 기타)을 가리킵니다.
<b>IP (인터넷 프로토콜)</b>	인터넷을 통해 한 컴퓨터에서 다른 컴퓨터로 데이터를 보내는 방법 또는 규약입니다.
<b>IP 헤더</b>	인터넷 패킷을 고유하게 식별하는 20바이트의 데이터입니다. 헤더는 패킷의 소스 및 대상 주소를 포함합니다. 헤더 내에는 바이트를 더 추가할 수 있는 옵션이 존재합니다.
<b>IPQoS</b>	<a href="#">diffserv 모델</a> 표준 구현과 더불어, 가상 LAN에 대한 흐름 계산 및 802.1D 표시를 제공하는 소프트웨어 기능입니다. IPQoS를 사용하면 IPQoS 구성 파일에 정의된 대로 여러 수준의 네트워크 서비스를 고객 및 응용 프로그램에 제공할 수 있습니다.
<b>IPsec</b>	IP 보안. IP 데이터그램에 대한 보호를 제공하는 보안 구조입니다.
<b>IPv4</b>	인터넷 프로토콜, 버전 4. IPv4를 종종 IP라고도 합니다. 이 버전은 32비트 주소 공간을 지원합니다.



IPv6	인터넷 프로토콜, 버전 6. IPv6은 128비트 주소 공간을 지원합니다.
link-local 주소	IPv6에서 자동 주소 구성과 같은 목적으로 단일 링크에 주소 배정을 위해 사용되는 지정입니다. 기본적으로 link-local 주소는 시스템의 MAC 주소에서 생성됩니다.
local-use 주소	(서브넷 내에 또는 가입자 네트워크 내에) 로컬 경로 지정 가능성 범위만 갖는 유니캐스트 주소입니다. 이 주소는 로컬 또는 전역 고유성 범위를 가질 수도 있습니다.
MAC (메시지 인증 코드)	MAC는 데이터 무결성을 보증하고 데이터 발신을 인증합니다. MAC는 도청에 대해 보호되지 않습니다.
MD5	디지털 서명을 포함하여 메시지 인증용으로 사용되는 반복적인 암호화 해시 함수입니다. 이 기능은 1991년 Rivest가 개발했습니다.
MTU	최대 전송 단위(Maximum Transmission Unit). 링크를 통해 전송할 수 있는 옥텟 단위의 크기입니다. 예를 들어, 이더넷의 MTU는 1500 옥텟입니다.
NAT	<a href="#">네트워크 주소 변환</a> 을 참조하십시오.
NIC (네트워크 인터페이스 카드)	네트워크에 인터페이스로 연결된 네트워크 어댑터 카드입니다. 일부 NIC는 igb 카드와 같은 여러 물리적 인터페이스를 가질 수 있습니다.
PFS (완전 순방향 비밀성)	PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송을 보호하는 키의 소스도 추가 키를 파생하는 데 사용되지 않습니다.  PFS는 인증된 키 교환에만 적용됩니다. <a href="#">Diffie-Hellman 알고리즘</a> 도 참조하십시오.
PHB (홉별 동작)	트래픽 클래스에 지정된 우선 순위입니다. PHB는 다른 트래픽 클래스와 비교하여 해당 클래스의 어떤 흐름이 우선권을 갖는지 나타냅니다.
PKI	공개 키 기반구조(Public Key Infrastructure). 인터넷 트랜잭션에 관여한 해당자의 유효성을 확인 및 인증하는 디지털 인증서, 인증 기관 및 기타 등록 기관의 시스템제입니다.
RSA	디지털 서명 및 공개 키 암호화 체계를 얻기 위한 방법입니다. 1978년에 개발자 Rivest, Shamir, Adleman이 처음 기술했습니다.
SA	<a href="#">SA(보안 연관)</a> 를 참조하십시오.
SA (보안 연관)	한 호스트에서 두번째 호스트로 보안 등록 정보를 지정하는 연관입니다.
SADB	보안 연관 데이터베이스(Security Associations Database). 암호화 키 및 암호화 알고리즘을 지정하는 테이블입니다. 키 및 알고리즘은 보안 데이터 전송에 사용됩니다.
SCTP	흐름 제어 전송 프로토콜을 참조하십시오.
SHA-1	보안 해시 알고리즘(Secure Hashing Algorithm). 알고리즘은 $2^{64}$ 미만의 입력 길이에서 작동하여 메시지 다이제스트를 생성합니다. SHA-1 알고리즘은 DSA로 입력됩니다.
site-local-use address	단일 링크에 주소 배정을 위해 사용되는 지정입니다.
SPD	<a href="#">SPD(보안 정책 데이터베이스)</a> 를 참조하십시오.

SPD (보안 정책 데이터베이스)	패킷에 적용할 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷을 폐기할지, 일반 텍스트로 전달할지, IPsec로 보호할지 결정합니다.
SPI	SPI(보안 매개변수 색인)를 참조하십시오.
SPI (보안 매개변수 색인)	수신자가 받은 패킷을 해독하기 위해 사용할 보안 연관 데이터베이스(SADB)의 행을 지정하는 정수입니다.
stateful 패킷 필터	활성 연결의 상태를 모니터링하여 얻은 정보를 바탕으로 네트워크 패킷이 <b>방화벽</b> 을 통과할지 여부를 확인할 수 있는 <b>패킷 필터</b> 입니다. 요청 및 회신을 추적하고 일치시키면 stateful 패킷 필터가 요청과 일치하지 않는 회신을 차단할 수 있습니다.
stateless 자동 구성	호스트가 로컬 IPv6 라우터에서 보급한 MAC 주소와 IPv6 접두어를 결합하여 고유의 IPv6 주소를 생성하는 프로세스입니다.
TCP/IP	TCP/IP(Transmission Control Protocol/Internet Protocol)는 인터넷의 기본 통신 언어 또는 규약입니다. 또한 인트라넷 또는 엑스트라넷과 같은 사설망에서 통신 프로토콜로 사용할 수 있습니다.
VPN (가상 사설망)	인터넷과 같은 공중망에서 터널을 사용하는 단일의 안전한 논리적 네트워크입니다.
가상 LAN (VLAN) 장치	이더넷(datalink) 레벨의 IP 프로토콜 스택에서 트래픽 전달을 제공하는 네트워크 인터페이스입니다.
가상 네트워크	소프트웨어 및 하드웨어 네트워크 리소스 및 기능의 조합으로, 단일 소프트웨어 엔티티로 함께 관리됩니다. <b>내부</b> 가상 네트워크는 네트워크 리소스를 단일 시스템으로 통합하며, 이를 때때로 “일체형 네트워크”라고도 합니다.
가상 네트워크 인터페이스 (VNIC)	물리적 네트워크 인터페이스에 구성되었는지 여부에 관계없이 가상 네트워크 연결을 제공하는 의사 인터페이스입니다. 배타적 IP 영역과 같은 컨테이너에서 위의 VNIC이 가상 네트워크를 형성하도록 구성됩니다.
개인 주소	인터넷을 통해 경로를 지정할 수 없는 IP 주소입니다. 개인 주소는 인터넷 연결이 필요하지 않은 호스트의 내부 네트워크에서 사용할 수 있습니다. 이러한 주소는 <a href="http://www.ietf.org/rfc/rfc1918.txt?number=1918">Address Allocation for Private Internets (http://www.ietf.org/rfc/rfc1918.txt?number=1918)</a> 에 정의되며 종종 “1918” 주소라고도 합니다.
결과	트래픽 측정 결과로 취할 조치입니다. IPQoS 측정자에는 IPQoS 구성 파일에서 정의한 빨강, 노랑, 녹색의 세 가지 결과가 있습니다.
공개 키 암호화	두 개의 다른 키를 사용하는 암호화 시스템입니다. 공개 키는 모든 사람이 알 수 있습니다. 개인 키는 메시지의 수신자만 알 수 있습니다. IKE는 IPsec에 공개 키를 제공합니다.
네트워크 주소 변환	NAT. 한 네트워크 내에 사용된 IP 주소를 다른 네트워크 내에 알려진 다른 IP 주소로 변환합니다. 필요한 전역 IP 주소 수를 제한하는 데 사용됩니다.
노드	IPv6에서 호스트든 라우터든 관계없이 IPv6이 사용으로 설정된 시스템입니다.
대기	다른 물리적 인터페이스가 실패하지 않는 한, 데이터 트래픽 전달에 사용되지 않는 물리적 인터페이스입니다.

대칭 키 암호화	메시지의 발신자 및 수신자가 단일의 공통 키를 공유하는 암호화 시스템입니다. 이 공통 키는 메시지를 암호화 및 해독하는 데 사용됩니다. 대칭 키를 사용하면 IPsec에서 데이터 전송을 대량으로 암호화할 수 있습니다. 대칭 키 시스템의 한 가지 예로 DES가 있습니다.
데이터그램	IP 데이터그램을 참조하십시오.
동적 재구성 (DR)	진행 중인 작업에 거의 또는 전혀 영향을 주지 않고 시스템이 실행 중인 동안 시스템을 재구성할 수 있는 기능입니다. Oracle의 모든 Sun 플랫폼이 DR을 지원하지는 않습니다. Oracle의 일부 Sun 플랫폼은 NIC와 같은 특정 유형의 하드웨어에만 DR을 지원할 수도 있습니다.
동적 패킷 필터	stateful 패킷 필터를 참조하십시오.
디지털 서명	발신자를 고유하게 식별하는, 전자적으로 전송된 메시지에 첨부된 디지털 코드입니다.
라우터	대개 여러 개의 인터페이스가 있고 경로 지정 프로토콜을 실행하며 패킷을 전달하는 시스템입니다. 시스템이 PPP 링크의 끝점인 경우 하나의 인터페이스만 있는 시스템을 라우터로 구성할 수 있습니다.
라우터 간청	호스트가 다음 일정이 잡힌 시간이 아닌, 즉시 라우터 알림을 생성하도록 라우터에 요청하는 프로세스입니다.
라우터 검색	호스트가 연결된 링크에 상주하는 라우터를 찾는 프로세스입니다.
라우터 알림	정기적으로 또는 라우터 간청 메시지의 응답으로, 라우터가 다양한 링크 및 인터넷 매개변수를 함께 사용하여 자신의 존재를 알리는 프로세스입니다.
로드 확산	인터페이스를 통해 인바운드 또는 아웃바운드 트래픽을 분배하는 프로세스입니다. 로드 확산을 사용하면 더 높은 처리량을 달성할 수 있습니다. 로드 확산은 네트워크 트래픽이 다중 연결을 사용하는 여러 대상으로 흐르고 있을 때만 발생합니다. 두 가지 유형의 로드 확산이 존재합니다. 인바운드 트래픽에는 인바운드 로드 확산을 사용하고 아웃바운드 트래픽에는 아웃바운드 로드 확산을 사용합니다.
링크 계층	IPv4/IPv6 바로 아래의 계층입니다.
멀티캐스트 주소	특수한 방법으로 인터페이스 그룹을 식별하는 IPv6 주소입니다. 멀티캐스트 주소로 보낸 패킷은 그룹의 모든 인터페이스로 전달됩니다. IPv6 멀티캐스트 주소는 IPv4 브로드캐스트 주소와 기능상 비슷합니다.
멀티홈 호스트	패킷 전달을 수행하지 않는 여러 개의 물리적 인터페이스가 있는 시스템입니다. 멀티홈 호스트는 경로 지정 프로토콜을 실행할 수 있습니다.
물리적 인터페이스	시스템의 링크 연결입니다. 이 연결은 종종 장치 드라이버와 NIC(네트워크 인터페이스 카드)로 구현됩니다. 일부 NIC는 여러 연결 지점(예: igb)을 가질 수 있습니다.
방문자 목록	외래 에이전트를 방문 중인 모바일 노드 목록입니다.
방문한 네트워크	모바일 노드가 현재 연결되어 있는, 모바일 노드의 홈 네트워크가 아닌 다른 네트워크입니다.

방화벽	조직의 사설망이나 인트라넷을 인터넷에서 격리시켜서 외부 침입으로부터 보호할 수 있는 장치 또는 소프트웨어입니다. 방화벽은 패킷 필터링, 프록시 서버 및 NAT(네트워크 주소 변환)를 포함할 수 있습니다.
복구 감지	NIC 또는 NIC에서 어떤 layer-3 장치로의 경로가 실패 후에 올바르게 작동을 시작하는지 감지하는 프로세스입니다.
브로드캐스트 주소	주소의 호스트 부분이 모두 제로(10.50.0.0) 또는 모두 한 비트(10.50.255.255)인 IPv4 네트워크 주소입니다. 로컬 네트워크의 시스템에서 브로드캐스트 주소로 보낸 패킷은 해당 네트워크의 모든 시스템에 전달됩니다.
비대칭 키 암호화	메시지를 암호화 및 해독하기 위해 메시지의 발신자 및 수신자가 서로 다른 키를 사용하는 암호화 시스템입니다. 비대칭 키는 대칭 키 암호화에 대한 보안 채널을 설정하는 데 사용됩니다. <a href="#">Diffie-Hellman 알고리즘</a> 은 비대칭 키 프로토콜의 예입니다. <a href="#">대칭 키 암호화</a> 와 대조됩니다.
사용자 우선 순위	class-of-service 표시를 구현하는 3비트 값으로, VLAN 장치의 네트워크에서 이더넷 데이터그램의 전달 방법을 정의합니다.
선택기	네트워크 시스템에서 트래픽을 선택하기 위해 특정 클래스의 패킷에 적용할 기준을 특별히 정의하는 요소입니다. IPQoS 구성 파일의 <a href="#">filter</a> 절에 선택기를 정의합니다.
속임수	메시지가 신뢰된 호스트에서 들어오고 있음을 나타내는 메시지를 IP 주소와 함께 보내어 컴퓨터에 허용되지 않은 액세스를 얻는 것입니다. IP 속임수에 관여하려면 먼저 해커가 다양한 기법을 사용하여 신뢰된 호스트의 IP 주소를 찾는 다음, 패킷이 해당 호스트에서 들어오고 있다고 나타나도록 패킷 헤더를 수정해야 합니다.
스니프	컴퓨터 네트워크에서 도청하는 것입니다. 일반 텍스트 암호, 유선 끄기와 같은 정보를 조사하기 위해 자동화된 프로그램의 일부로 자주 사용됩니다.
스머프 공격	원격 위치에서 IP <a href="#">브로드캐스트 주소</a> 또는 다중 브로드캐스트 주소로 지정된 ICMP 에코 요청 패킷을 사용하여 심각한 네트워크 혼잡 또는 정전을 일으킵니다.
스택	<a href="#">IP 스택</a> 을 참조하십시오.
애니캐스트 그룹	동일한 애니캐스트 IPv6 주소를 가진 인터페이스 그룹입니다. Oracle Solaris IPv6 구현은 애니캐스트 주소 및 그룹의 생성을 지원하지 않습니다. 그러나 Oracle Solaris IPv6 노드가 애니캐스트 그룹으로 트래픽을 보낼 수 있습니다.
애니캐스트 주소	(일반적으로 서로 다른 노드에 속하는) 인터페이스 그룹에 지정된 IPv6 주소입니다. 애니캐스트 주소로 보낸 패킷은 해당 주소를 가진 <a href="#">가장 가까운</a> 인터페이스로 경로가 지정됩니다. 패킷의 경로는 경로 지정 프로토콜의 거리 측정을 준수합니다.
양방향 터널	데이터그램을 양방향으로 전송할 수 있는 터널입니다.
역방향 터널	모바일 노드의 care-of 주소에서 시작해서 홈 에이전트에서 끝나는 터널입니다.
유니캐스트 주소	IPv6 사용 노드의 단일 인터페이스를 식별하는 IPv6 주소입니다. 유니캐스트 주소의 부분은 사이트 접두어, 서브넷 ID, 인터페이스 ID입니다.
이웃 간청	이웃의 link-layer 주소를 결정하기 위해 노드에서 보낸 간청입니다. 또한 이웃 간청은 캐시된 link-layer 주소에서 이웃에 아직 연결할 수 있는지 확인합니다.

이웃 검색	호스트가 연결된 링크에 상주하는 다른 호스트를 찾을 수 있는 IP 방식입니다.
이웃 알림	이웃 간청 메시지에 대한 응답 또는 link-layer 주소 변경을 공지하기 위해 노드가 청하지 않은 이웃 알림을 보내는 프로세스입니다.
이중 스택	네트워크 계층에 IPv4 및 IPv6이 모두 있는 TCP/IP 프로토콜 스택입니다(스택의 나머지는 동일함). Oracle Solaris 설치 중 IPv6을 사용으로 설정하면 호스트가 TCP/IP의 이중 스택 버전을 수신합니다.
인증 기관 (CA)	디지털 서명 및 공개-개인 키 쌍을 만드는 데 사용된 디지털 인증서를 발행하는 신뢰된 타사 조직 또는 회사입니다. CA는 고유한 인증서를 부여받은 개인의 신원을 보증합니다.
인증 헤더	IP 데이터그램에 (기밀성 없이) 인증 및 무결성을 제공하는 확장 헤더입니다.
자동 구성	호스트가 사이트 접두어 및 로컬 MAC 주소로부터 해당 IPv6 주소를 자동으로 구성하는 프로세스입니다.
재전송 공격	IPsec에서 침입자가 패킷을 캡처하는 공격입니다. 그런 다음 저장된 패킷이 나중에 원본을 대체하거나 반복합니다. 이러한 공격으로부터 보호하려면 패킷을 보호 중인 보안 키의 수명 주기 동안 증분하는 필드를 포함할 수 있습니다.
재지정	라우터에서 특정 대상에 연결하기 위해 더 좋은 첫번째 홉 노드를 호스트에 알려주는 것입니다.
최소 캡슐화	홉 에이전트, 외래 에이전트, 모바일 노드에서 지원할 수 있는 선택적 형태의 IPv4-in-IPv4 터널링입니다. 최소 캡슐화는 IP-in-IP 캡슐화보다 8 또는 12바이트 정도 오버헤드가 적습니다.
측정자	특정 클래스에 대한 트래픽 흐름의 비율을 측정하는 diffserv 구조의 모듈입니다. IPQoS 구현에는 tokenmt 및 tswtclmt의 두 측정자가 포함됩니다.
캡슐화	헤더 및 페이로드를 첫번째 패킷에 넣고, 이어서 두번째 패킷의 페이로드에 넣는 프로세스입니다.
클래스	IPQoS에서 비슷한 특성을 공유하는 네트워크 흐름 그룹입니다. IPQoS 구성 파일에 클래스를 정의합니다.
키 관리	보안 연관(SA)을 관리하는 방법입니다.
키 저장소 이름	NIC(네트워크 인터페이스 카드)의 저장소 영역 또는 키 저장소에 관리자가 부여하는 이름입니다. 키 저장소 이름을 토큰 또는 토큰 ID라고도 합니다.
터널	캡슐화된 동안 데이터그램에 이어지는 경로입니다. 캡슐화를 참조하십시오.
패킷	통신 회선을 통해 한 단위로 전송되는 정보 그룹입니다. IP 헤더와 페이로드를 포함합니다.
패킷 필터	방화벽을 통해 지정된 패킷을 허용하도록 구성하거나 허용하지 않도록 구성할 수 있는 방화벽 기능입니다.
패킷 헤더	IP 헤더를 참조하십시오.
페이로드	패킷에 전달된 데이터입니다. 페이로드는 패킷을 대상으로 가져오는 데 필요한 헤더 정보를 포함하지 않습니다.

표시자	<p>1. 패킷의 전달 방법을 나타내는 값으로 IP 패킷의 DS 필드를 표시하는 diffserv 구조 및 IPQoS의 모듈입니다. IPQoS 구현에서 표시자 모듈은 dscpmk입니다.</p> <p>2. 이더넷 데이터그램의 가상 LAN 태그를 사용자 우선 순위 값으로 표시하는 IPQoS 구현의 모듈입니다. 사용자 우선 순위 값은 VLAN 장치가 포함된 네트워크에서 데이터그램의 전달 방법을 나타냅니다. 이 모듈을 dlcosmk라고 합니다.</p>
프로토콜 스택	IP 스택을 참조하십시오.
프록시 서버	클라이언트 응용 프로그램(예: 웹 브라우저)과 다른 서버 사이에 앉은 서버입니다. 요청을 필터링하는 데 사용됩니다. 예를 들어, 특정 웹 사이트에 액세스를 금지할 수 있습니다.
필터	IPQoS 구성 파일에 클래스의 특성의 정의하는 규칙 세트입니다. IPQoS 시스템이 IPQoS 구성 파일에서 필터를 준수하는 트래픽 흐름을 처리하기 위해 선택합니다. <b>패킷 필터</b> 를 참조하십시오.
해시 값	텍스트의 문자열에서 생성된 숫자입니다. 해시 함수를 사용하여 전송된 메시지가 변조되지 않았는지 확인할 수 있습니다. MD5 및 SHA-1은 단방향 해시 함수의 예입니다.
헤더	IP 헤더를 참조하십시오.
호스트	패킷 전달을 수행하지 않는 시스템입니다. Oracle Solaris 설치 시 시스템은 기본적으로 호스트가 됩니다. 즉 시스템이 패킷을 전달할 수 없습니다. 호스트는 다중 인터페이스를 가질 수 있지만 일반적으로 하나의 물리적 인터페이스를 가집니다.
홉	두 호스트를 구분하는 라우터 수를 식별하는 데 사용되는 측정값입니다. 3개의 라우터가 소스 및 대상을 구분하는 경우 호스트가 서로 4홉씩 떨어져 있습니다.
흐름 계산	IPQoS에서 트래픽 흐름에 대한 정보를 누적하고 기록하는 프로세스입니다. IPQoS 구성 파일에 flowacct 모듈의 매개변수를 정의하여 흐름 계산을 설정합니다.
흐름 제어 전송 프로토콜	TCP와 비슷한 방법으로 연결 지향적 통신을 제공하는 전송 계층 프로토콜입니다. 추가적으로, SCTP는 멀티홉 기능을 지원하므로 연결 끝점 중 하나가 여러 개의 IP 주소를 가질 수 있습니다.

# 색인

---

## 번호와 기호

3DES 암호화 알고리즘, IPsec 및, 208

6to4 릴레이 라우터

6to4 터널, 146

보안 문제, 116–118, 133–134

터널 구성 작업, 126, 127

터널 토폴로지, 117

6to4 알림, 125

6to4 터널

참조 터널, 유형

6to4 릴레이 라우터, 126

샘플 토폴로지, 114

패킷 흐름, 115, 117

6to4relay 명령, 126

구문, 147

예제, 147

정의, 146

터널 구성 작업, 126

## A

-A 옵션

ikecert certlocal 명령, 258

ikecert 명령, 285

-a 옵션

ikecert certdb 명령, 259, 263

ikecert certrldb 명령, 272

ikecert 명령, 267

AAAA 레코드, 87, 159–160

acctadm 명령, 흐름 계산, 472

acctadm 명령, 흐름 계산용, 457

action 명령문, 474

AES 암호화 알고리즘, IPsec 및, 208

AF(보장 전달), 398, 467

AF 코드 포인트 표, 467

표시기 action 명령문, 430

AH, 참조 AH(authentication header)

AH(authentication header)

IP 데이터그램 보호, 206

IP 패킷 보호, 199

IPsec 보호 방식, 206–208

보안 고려 사항, 207

ARP(Address Resolution Protocol), Neighbor

Discovery 프로토콜과 비교, 156–158

ATM 지원, IPv6 over, 160

## B

Blowfish 암호화 알고리즘, IPsec 및, 208

BOOTP 프로토콜, 및 DHCP, 163

## C

-c 옵션

in.iked 데몬, 253

ipseckey 명령, 241

cert\_root 키워드

IKE 구성 파일, 264, 269

cert\_trust 키워드

IKE 구성 파일, 261, 268

ikecert 명령, 285

CIDR 표기법, 27



ciphers, 참조 encryption 알고리즘

class 절, IPQoS 구성 파일, 426

class 절, IPQoS 구성 파일, 475

CoS(서비스클래스) 표시, 394

CRL

ike/crls 데이터베이스, 286

ikecert certrldb 명령, 286

나열, 270

무시, 265

중앙 위치에서 액세스, 270

CRL에 대한 HTTP 액세스, use\_http 키워드, 271

## D

-D 옵션

ikecert certlocal 명령, 258

ikecert 명령, 285

defaultrouter 파일, 로컬 파일 모드 구성, 53

DES 암호화 알고리즘, IPsec 및, 208

DHCP 네트워크 테이블, 설명, 195

DHCP 명령줄 유틸리티, 권한, 171

DHCP 이벤트, 189-191

DHCP 임대 연장, 184

DHCP 클라이언트

IP 주소 삭제, 185

IP 주소 해제, 184

관리, 184

구성 해제, 183-184

논리적 인터페이스, 186

다중 네트워크 인터페이스, 186

매개변수, 185

사용 안함, 183-184

사용으로 설정, 183

시작, 179, 184

이벤트 스크립트, 189-191

임대 없이 네트워크 정보, 184

임대 연장, 184

정의, 169

종료, 182

프로그램 실행, 189-191

호스트 이름

지정, 187

DHCP 프로토콜

Oracle Solaris 구현의 이점, 164

DHCP 프로토콜 (계속)

개요, 163

이벤트 순서, 165

dhcpageant 데몬, 179

dhcpageant 데몬, 매개변수 파일, 195

dhcpageant 명령, 설명, 194

dhcpageant 파일, 설명, 195

dhcpcconfig 명령, 설명, 194

dhcpcd 데몬, 설명, 193

dhcpcd4.conf 파일, 설명, 195

dhcpcd6.conf 파일, 설명, 195

dhcpcinfo 명령, 설명, 194

dhcpcmgr 명령, 설명, 193

dhcpsvc.conf 파일, 195

dhcptab 테이블, 설명, 195

DHCPv4 및 DHCPv6 비교, 176

DHCPv4 클라이언트, 네트워크 인터페이스의  
관리, 181

DHCPv6, 클라이언트 이름, 177

DHCPv6 관리 모델, 176

DHCPv6 및 DHCPv4 비교, 176

DHCPv6 클라이언트, 네트워크 인터페이스의  
관리, 181

dhcrelay 명령, 설명, 193

dhtadm 명령, 설명, 194

Diffie-Hellman 그룹, IKE 미리 공유한 키, 249-251

Diffserv 모델

IPQoS 구현, 392, 394, 395

분류기 모듈, 392

측정기 모듈, 394

표시기 모듈, 394

흐름 예, 395

Diffserv 인식 라우터

DS 코드 포인트 평가, 468

계획, 407

dladm 명령

IP 터널 삭제, 130

터널 구성 수정, 128-129

터널 만들기, 120-124

터널 정보 표시, 129

dlcosmk 표시기, 394

VLAN 태그, 469

사용자 우선 순위 값, 표, 469

dlcosmk 표시자, 데이터그램 전달 계획, 414



## DNS(Domain Name System)

- IPv6에 대한 확장, 159-160
- 역순 영역 파일, 86
- 영역 파일, 86
- 이름 서비스로 선택, 30
- 준비, IPv6 지원, 40
- DS 코드점(DSCP), 계획, QoS 정책에서, 415
- DSCP(DS 코드 포인트), 394, 397
  - AF 전달 코드 포인트, 398, 467
  - dscp\_map 매개변수, 468
  - EF 전달 코드 포인트, 398, 467
  - PHB 및 DSCP, 397
  - 구성, diffserv 라우터, 445, 467
  - 색상 인식 구성, 466
  - 정의, IPQoS 구성 파일, 430
- dscpmk 표시기, 394
  - 패킷 전달을 위한 PHB, 466
  - 호출, 표시기 action 명령문, 429, 435, 441, 443
- dscpmk 표시자, 패킷 전달 계획, 414
- DSS 인증 알고리즘, 285

**E**

- EF(빠른 전달), 398, 467
  - 정의, IPQoS 구성 파일, 431
- ESP, 참조 ESP(encapsulating security payload)
- ESP(encapsulating security payload)
  - IP 패킷 보호, 199
  - IPsec 보호 방식, 206-208
  - 보안 고려 사항, 207
  - 설명, 207-208
- /etc/bootparams 파일, 설명, 135
- /etc/default/dhcpagent 파일, 185
- /etc/default/dhcpagent 파일, 설명, 195
- /etc/default/inet\_type 파일, 99
  - DEFAULT\_IP 값, 148
- /etc/defaulttrouter 파일
  - 로컬 파일 모드 구성, 53
  - 설명, 135
- /etc/dhcp/dhcptags 파일, 설명, 196
- /etc/dhcp/eventhook 파일, 190
  - 설명, 195
- /etc/dhcp/inittab 파일, 설명, 196
- /etc/dhcp/interface.dh\* 파일, 설명, 195
- /etc/ethers 파일, 설명, 135
- /etc/inet/dhcpd4.conf 파일, 설명, 195
- /etc/inet/dhcpd6.conf 파일, 설명, 195
- /etc/inet/dhcpsvc.conf 파일, 설명, 195
- /etc/inet/hosts 파일, 219
  - 네트워크 클라이언트 모드 구성, 54
  - 로컬 파일 모드 구성, 53
  - 설명, 135
- /etc/inet/ike/config 파일
  - cert\_root 키워드, 264, 269
  - cert\_trust 키워드, 261, 268
  - ignore\_crls 키워드, 265
  - ikecert 명령, 285
  - ldap-list 키워드, 271
  - PKCS#11 라이브러리 항목, 284
  - pkcs11\_path 키워드, 267, 284
  - proxy 키워드, 271
  - use\_http 키워드, 271
  - 공개 키 인증서, 264, 269
  - 미리 공유한 키, 252
  - 보안 고려 사항, 283
  - 샘플, 252
  - 설명, 245, 282
  - 요약, 247
  - 자체 서명된 인증서, 261
  - 하드웨어에 인증서 넣기, 268
- /etc/inet/ike/crls 디렉토리, 286
- /etc/inet/ike/publickeys 디렉토리, 286
- /etc/inet/ipaddrsel.conf 파일, 109, 145
- /etc/inet/ipsecinit.conf 파일, 238-239
- /etc/inet/ndpd.conf 파일, 78, 149
  - 6to4 라우터 알림, 125
  - 만들기, 78
  - 인터페이스 구성 변수, 142
  - 임시 주소 구성, 81
  - 접두어 구성 변수, 144
  - 키워드, 142-145, 150
- /etc/inet/secret/ike.privatekeys 디렉토리, 286
- /etc/ipf/ipf.conf 파일, 참조 IP 필터
- /etc/ipf/ipnat.conf 파일, 참조 IP 필터
- /etc/ipf/ippool.conf 파일, 참조 IP 필터
- /etc/netmasks 파일, 설명, 135
- /etc/networks 파일, 설명, 135
- /etc/protocols 파일, 설명, 135

/etc/services 파일, 설명, 135  
ethers 데이터베이스, 항목 검사, 132  
eventhook 파일, 190

## F

-F 옵션, ikecert certlocal 명령, 258  
-f 옵션, in.iked 데몬, 253  
filter 절, IPQoS 구성 파일, 428, 476  
flowacct 모듈, 395, 470  
    acctadm 명령, 흐름 계산 파일 만들기, 472  
    flowacct에 대한 action 명령문, 432  
    매개변수, 471  
    흐름 레코드, 456  
    흐름 레코드 테이블, 471  
    흐름 레코드의 속성, 472

## H

hosts 데이터베이스  
    /etc/inet/hosts 파일  
        로컬 파일 모드 구성, 53  
        항목 검사, 132  
hosts 파일, 219

## I

ICMP 프로토콜  
    메시지, Neighbor Discovery 프로토콜, 153  
    통계 표시, 91  
    호출, ping 사용, 97  
ID 연관, 177  
ignore\_crls 키워드, IKE 구성 파일, 265  
IKE  
    1단계 알고리즘 및 그룹 보기, 249-251  
    crls 데이터베이스, 286  
    ike.preshared 파일, 284  
    ike.privatekeys 데이터베이스, 286  
    ikeadm 명령, 283  
    ikecert certdb 명령, 263  
    ikecert certrldb 명령, 272  
    ikecert tokens 명령, 279

## IKE (계속)

    ikecert 명령, 284  
    in.iked 데몬, 282  
    ISAKMP SA, 244  
    NAT 및, 275-276, 277-278  
    PFS(완전 순방향 비밀성), 244  
    Phase 1 교환, 244  
    Phase 2 교환, 245  
    publickeys 데이터베이스, 286  
    RFC, 201  
    SMF 서비스 설명, 246-247  
    SMF를 사용하여 관리, 234-235  
    SMF의 서비스, 281-282  
    Sun Crypto Accelerator 6000 보드 사용, 279-280  
    Sun Crypto Accelerator 보드 사용, 284, 286  
    개요, 243  
    구성  
        CA 인증서 사용, 262-266  
        공개 키 인증서 사용, 256  
        모바일 시스템용, 272-278  
        미리 공유한 키 사용, 251  
    구성 파일, 246-247  
    구현, 251  
    권한 레벨  
        변경, 283  
        설명, 283  
    데몬, 282  
    데이터베이스, 284-286  
    명령 설명, 246-247  
    모바일 시스템 및, 272-278  
    미리 공유한 키, 245  
        1단계 알고리즘 및 그룹 보기, 249-251  
    변경  
        권한 레벨, 283  
    보기  
        1단계 알고리즘 및 그룹, 249-251  
    보안 연관, 282  
    사용 가능한 알고리즘 표시, 249-251  
    유효한 구성인지 여부 확인, 253  
    인증서, 246  
    인증서 요청 생성, 262  
    자체 서명된 인증서 만들기, 257  
    자체 서명된 인증서 추가, 257  
    참조, 281

**IKE (계속)**

- 키 관리, 244
- 키의 저장소 위치, 246-247
- ike/config 파일, **참조** /etc/inet/ike/config 파일
- ike.preshared 파일, 253, 284
  - 샘플, 255
- ike.privatekeys 데이터베이스, 286
- IKE 구성(작업 맵), 251
- ike 서비스
  - 사용, 220
  - 설명, 206, 237
- ikeadm 명령
  - dump 하위 명령, 249-251
  - 설명, 282, 283
- ikecert certdb 명령
  - a 옵션, 259, 263
- ikecert certlocal 명령
  - kc 옵션, 262
  - ks 옵션, 257
- ikecert certrldb 명령, -a 옵션, 272
- ikecert tokens 명령, 279
- ikecert 명령
  - A 옵션, 285
  - a 옵션, 267
  - T 옵션, 267
  - t 옵션, 285
  - 설명, 282, 284
- in.dhcpd 데몬, 설명, 193
- in.iked 데몬
  - c 옵션, 253
  - f 옵션, 253
  - 설명, 244
  - 활성화, 282
- in.ndpd 데몬
  - 로그 만들기, 100-101
  - 상태 확인, 132
  - 옵션, 149
- in.rdisc 프로그램, 설명, 139
- in.ripngd 데몬, 78, 150
- in.routed 데몬
  - 공간 절약 모드, 139
  - 로그 만들기, 100
  - 설명, 139
- in.tftpd 데몬, 55

- in.tftpd 데몬, 켜기, 55
- inet\_type 파일, 99
- inetd 데몬
  - IPv6 서비스, 150-152
- inetd 데몬, 상태 확인, 132
- inetd 데몬
  - 서비스 관리, 136
  - 서비스 시작 데몬, 69
- IP 데이터그램, IPsec로 보호, 199
- IP 보안 아키텍처, **참조** IPsec
- IP 인터페이스
  - 터널을 경유하여 구성됨, 118-119, 122, 125
- IP 전달
  - IPv4 VPN, 227
  - VPN, 211
- IP 주소
  - CIDR 표기법, 27
  - 네트워크 클래스
  - 네트워크 번호 관리, 27
  - 주소 체계 설계, 27
- IP 터널, **참조** 터널
- IP 프로토콜
  - 통계 표시, 91
  - 호스트 연결 확인, 97, 98
- IP 필터
  - /etc/ipf/ipf.conf 파일, 326-327
  - /etc/ipf/ipf6.conf 파일, 297-298
  - /etc/ipf/ipnat.conf 파일, 326-327
  - /etc/ipf/ippool.conf 파일, 326-327
  - ipadm 명령, 290-291
  - ipf.conf 파일, 292-294
  - ipf 명령, 303-304
    - 6 옵션, 297-298
  - ipf6.conf 파일, 297-298
  - ipfstat 명령
    - 6 옵션, 297-298
  - ipmon 명령
    - IPv6 및, 297-298
  - IPMP에서, 290-291
  - ipnat.conf 파일, 295-296
  - ipnat 명령, 303-304
  - ippool.conf 파일, 296-297
  - ippool 명령, 317
    - IPv6 및, 297-298

## IP 필터 (계속)

IPv6, 297-298

NAT 규칙

보기, 315

추가, 316

NAT 및, 295-296

개요, 287-288

구성 파일 만들기, 326-327

구성 파일 예, 291

규칙 세트

다른 항목 활성화, 309-310

비활성, 309

비활성 제거, 314

비활성에 추가, 312-313

전환, 313-314

제거, 311

활성, 308-309

활성에 추가, 311-312

규칙 세트 및, 292-297

기록된 패킷을 파일에 저장, 325

다시 사용으로 설정, 303-304

로그 파일 비우기, 324

루프백 필터링, 304-305

만들기

로그 파일, 322-323

보기

NAT 통계, 321

로그 파일, 323-324

상태 테이블, 319-320

상태 통계, 320

주소 풀 통계, 321

비활성화, 306-307

NAT, 306

사용 지침, 290-291

오픈 소스, 288

제거

NAT 규칙, 315-316

주소 풀

보기, 317

제거, 317

추가, 318

주소 풀 및, 296-297

패킷 필터 후크, 297, 302-303

패킷 필터링 개요, 292-294

## IP 필터 (계속)

패킷 필터링 규칙 세트 관리, 308-314

IP 필터 비활성화, 306-307

ipaddrsel.conf 파일, 109, 145

ipaddrsel 명령, 109, 145-146

ipadm 명령, 290-291

DHCP 클라이언트 제어, 184

hostmodel 매개변수, 227

멀티홈 호스트, 62

문제 해결 도구로 사용, 131

엄격한 다중 홈 지정, 227

인터페이스 배관, 48

ipdam 명령, DHCP, 194

ipf.conf 파일, 292-294

참조 IP 필터

ipf 명령

참조 IP 필터

-6 옵션, 297-298

-a 옵션, 309-310

-D 옵션, 306-307

-E 옵션, 303-304

-F 옵션, 305-306, 309-310, 311, 314

-f 옵션, 303-304, 309-310, 311-312, 312-313

-I 옵션, 312-313, 314

-s 옵션, 313-314

명령줄에서 규칙 추가, 311-312

ipfstat 명령, 319-320

참조 IP 필터

-6 옵션, 297-298

-I 옵션, 309

-i 옵션, 308-309, 309

-o 옵션, 308-309, 309

-s 옵션, 320

-t 옵션, 319-320

ipgpc 분류기, 참조 분류기 모듈

ipmon 명령

참조 IP 필터

-a 옵션, 323-324

-F 옵션, 324

IPv6 및, 297-298

-o 옵션, 323-324

IPMP, 패킷 필터링을 사용으로 설정, 290-291

ipnat.conf 파일, 295-296

참조 IP 필터

- ipnat 명령
  - 참조 IP 필터
  - C 옵션, 306
  - F 옵션, 306, 315-316
  - f 옵션, 303-304, 316
  - l 옵션, 315
  - s 옵션, 321
  - 명령줄에서 규칙 추가, 316
- ippool.conf 파일, 296-297
  - 참조 IP 필터
- ippool 명령
  - 참조 IP 필터
  - F 옵션, 317
  - f 옵션, 318
  - IPv6 및, 297-298
  - l 옵션, 317
  - s 옵션, 321
  - 명령줄에서 규칙 추가, 318
- IPQoS, 387
  - Diffserv 모델 구현, 392
  - IPQoS 네트워크의 라우터, 445
  - IPv6 지원 네트워크에 대한 정책, 40
  - QoS 정책 계획, 405
  - VLAN 장치 지원, 469
  - 관련 RFC, 388
  - 구성 계획, 401
  - 구성 예, 417-419
  - 구성 파일, 423, 473
    - action 명령문 구문, 474
    - class 절, 426
    - filter 절, 428
    - IPQoS 모듈 목록, 475
    - 구문, 473
    - 초기 action 명령문, 474
    - 초기 작업 명령문, 426
    - 표시기 action 명령문, 429
  - 기능, 388
  - 네트워크 예, 423
  - 매뉴얼 페이지, 389
  - 메시지 로깅, 449
  - 오류 메시지, 450
  - 지원되는 네트워크 토폴로지, 402, 403, 404
  - 통계 생성, 458
  - 트래픽 관리 기능, 391, 392
  - IPQoS 네트워크의 가상 LAN(VLAN) 장치, 469
  - IPQoS 사용 네트워크에 대한 하드웨어, 402
- ipqosconf, 422
- ipqosconf 명령
  - 구성 적용, 448, 449
  - 명령 옵션, 477
  - 현재 구성 나열, 449
- IPQoS에 대한 syslog.conf 파일 로깅, 449
- IPQoS에 대한 네트워크 예, 423
- IPQoS에 대한 네트워크 토폴로지, 402
  - IPQoS 사용 방화벽의 LAN, 404
  - IPQoS 사용 서버 팜의 LAN, 402
  - IPQoS 사용 호스트의 LAN, 403
  - 구성 예, 417
- IPQoS에 대한 오류 메시지, 450
- IPQoS에 대한 통계
  - 수집, kstat 명령 사용, 458
  - 전역 통계 사용, 426, 475
  - 클래스 기반 통계 사용, 476
- IPsec
  - ESP(encapsulating security payload), 206-208
  - /etc/hosts 파일, 219
  - in.iked 데몬, 206
  - ipsecalgs 명령, 208, 240
  - ipsecconf 명령, 209, 238
  - ipsecinit.conf 파일
    - LAN 우회, 228
    - 구성, 219
    - 설명, 238-239
    - 웹 서버 보호, 221
    - 정책 파일, 209
  - ipseckey 명령, 206, 241-242
  - IPv4 VPN, 및, 227-230
  - NAT 및, 212-213
  - RBAC 및, 217
  - RFC, 201
  - route 명령, 230
  - SA(보안 연결), 200, 205-206
  - SA(보안 연결) 추가, 219, 228
  - SADB(보안 연결 데이터베이스), 200, 240
  - SCTP 프로토콜 및, 213, 217
  - SMF를 사용하여 관리, 234-235
  - SMF의 서비스, 237
  - snoop 명령, 242

## IPsec (계속)

- SPD(보안 정책 데이터베이스), 200, 201, 238
- SPI(보안 매개변수 색인), 205-206
- Trusted Extensions 레이블 및, 218
- VPN(virtual private networks), 211, 227-230
- VPN 보호, 223-230
- 개요, 199
- 구성, 209, 238
- 구성 요소, 200
- 구성 파일, 214-215
- 구현, 218
- 논리적 도메인 및, 213
- 데이터 캡슐화, 207
- 레이블이 있는 패킷 및, 218
- 명령, 목록, 214-215
- 보안 방식, 200
- 보안 역할, 232-234
- 보안 원격 로그인을 위해 ssh 사용, 220
- 보안 프로토콜, 200, 205-206
- 보호
  - VPN, 227-230
  - 모바일 시스템, 272-278
  - 웹 서버, 221-222
  - 패킷, 199
- 보호 방식, 206-208
- 보호 정책, 209
- 서비스
  - ipsecalgs, 215
  - manual-key, 214
  - policy, 214
- 서비스, 목록, 214-215
- 수동으로 SA 만들기, 231-232
- 아웃바운드 패킷 프로세스, 202
- 알고리즘 소스, 240
- 암호화 알고리즘, 208
- 암호화 프레임워크 및, 240
- 영역 및, 213, 217
- 용어, 201-202
- 우회, 209, 221
- 원격 로그인 보안, 219
- 유틸리티에 대한 확장
  - snoop 명령, 242
- 인바운드 패킷 프로세스, 202
- 인증 알고리즘, 208

## IPsec (계속)

- 전송 모드, 209-211
- 정책 명령
  - ipseccnf, 238
- 정책 설정
  - 영구적으로, 238-239
  - 임시로, 238
- 정책 파일, 238-239
- 정책 표시, 222-223
- 키 관련 유틸리티
  - IKE, 244
- 키 관리, 205-206
- 키 입력 유틸리티
  - ipseckey 명령, 241-242
- 터널, 211
- 터널 모드, 209-211
- 트래픽 보호, 218-221
- 패킷 보호 확인, 235-236
- 활성화, 214
- IPsec 정책, 터널 구문의 예, 223-225
- ipsecalgs 서비스, 설명, 237
- ipseccnf 명령
  - IPsec 정책 구성, 238
  - IPsec 정책 보기, 238-239
  - IPsec 정책 표시, 221-222, 222-223
  - 보안 고려 사항, 239
  - 설명, 214
  - 용도, 209
  - 터널 설정, 210
- ipseccinit.conf 파일
  - LAN 우회, 228
  - 구문 확인, 220, 228
  - 보안 고려 사항, 239
  - 샘플, 238
  - 설명, 214
  - 용도, 209
  - 웹 서버 보호, 221
  - 위치 및 범위, 213
- ipseckey 명령
  - 보안 고려 사항, 241-242
  - 설명, 214, 241-242
  - 용도, 206
- ipseckey 파일
  - IPsec 키 저장, 214

**ipseckeys 파일 (계속)**

구문 확인, 232

IPsec를 사용하여 트래픽 보호(작업 맵), 218

IPv4 네트워크, 구성 파일, 135

IPv4 터널, **참조** 터널, 유형**IPv6**

ATM 지원, 160

DNS AAAA 레코드, 87

DNS 지원 준비, 40

in.ndpd 데몬, 149

in.ndpd의 상태 확인, 132

in.ripngd 데몬, 150

IPv4와 비교, 156-158

Neighbor Discovery 프로토콜, 152-158

nslookup 명령, 87

Stateless 주소 자동 구성, 154

경로 지정, 158

기본 주소 선택 정책 테이블, 145

라우터 검색, 149, 157

라우터 알림, 153, 154, 157, 159

라우터 요청, 153, 154

링크 로컬 주소, 154, 158

멀티캐스트 주소, 157

및 IP 필터, 297-298

보안 고려 사항, 41

사용, 서버에서, 85-86

이웃 연결 불가 감지, 157

이웃 요청, 153

이웃 요청 및 연결 불가, 155

일반 IPv6 문제 해결, 132-134

임시 주소 구성, 80-83

재지정, 153, 157

주소 자동 구성, 149, 153

주소 지정 계획, 37-38

추가

DNS 지원, 86

트래픽 모니터링, 105

프로토콜 개요, 153

IPv6 주소, 고유성, 154

IPv6 터널, **참조** 터널, 유형

ISAKMP(Internet Security Association and Key

Management Protocol) SA

설명, 244

저장소 위치, 284

**K**

-kc 옵션

ikecert certlocal 명령, 262, 285

-ks 옵션

ikecert certlocal 명령, 257, 285

kstat 명령, IPQoS에서 사용, 458

**L**

-L 옵션, ipsecconf 명령, 223

-l 옵션

ikecert certddb 명령, 259

ipsecconf 명령, 223

ldap-list 키워드, IKE 구성 파일, 271

**M**

-m 옵션, ikecert certlocal 명령, 257

MAC 주소, 177

manual-key 서비스

사용, 232

설명, 206, 237

metaslot, 키 저장소, 280

MTU(최대 전송 단위), 157

**N**

name-service/switch SMF 서비스, 137

NAT

IPsec 및 IKE 사용, 275-276, 277-278

IPsec 제한 사항, 212-213

NAT 규칙

보기, 315

추가, 316

NAT 규칙 제거, 315-316

개요, 295-296

규칙 구성, 295-296

비활성화, 306

통계 보기, 321

NAT(Network Address Translation), **참조** NAT

ndpd.conf 파일

6to4 알림, 125

## ndpd.conf 파일 (계속)

만들기, IPv6 라우터, 78

## ndpd.conf 파일

인터페이스 구성 변수, 142

## ndpd.conf 파일

임시 주소 구성, 81

## ndpd.conf 파일

접두어 구성 변수, 144

키워드 목록, 142-145

## Neighbor Discovery 프로토콜

라우터 검색, 154

비교ARP, 156-158

이웃 요청, 155

접두어 검색, 154

주소 자동 구성, 153

주요 기능, 152-158

중복 주소 감지 알고리즘, 155

## netmasks 데이터베이스, 서버넷 추가, 53

## netstat 명령

-a 옵션, 94

-f 옵션, 94

inet 옵션, 94

inet6 옵션, 94

IPv6 확장, 148

-r 옵션, 96-97

구문, 91

설명, 91

소프트웨어 검사 실행, 132

알려진 경로의 상태 표시, 96-97

프로토콜별 통계 표시, 91

/network/dhcp/relay SMF 서비스, 설명, 196

/network/dhcp-server SMF 서비스, 설명, 196

/network/dhcp/server SMF 서비스, 설명, 196

/network/dns/client SMF 서비스, DHCP에서  
사용, 196

Network IPsec Management 권한 프로파일, 233

Network Management 권한 프로파일, 233

Network Security 권한 프로파일, 232-234

NIS, 이름 서비스로 선택, 30

nis/tdomain SMF 서비스, 로컬 파일 모드 구성, 53

## nslookup 명령, 160

IPv6, 87

## O

omshell 명령, 설명, 194

/opt/SUNWconn/lib/libpkcs11.so 항목, ike/config  
파일, 284

## P

## params 절

action 측정, 443

flowacct action, 432

구문, 476

전역 통계 정의, 426, 476

표시기 action, 430

## PF\_KEY 소켓 인터페이스

IPsec, 205, 214

PFS, 참조 PFS(완전 순방향 비밀성)

PFS(완전 순방향 비밀성)

IKE, 244

설명, 244

## PHB(홉별 동작), 397

AF 전달, 398

EF 전달, 398

사용, dscpmk 표시기, 466

정의, IPQoS 구성 파일, 444

## ping 명령, 98

description, 97

IPv6에 대한 확장, 148

-s 옵션, 98

구문, 97

실행, 98

PKCS #11 라이브러리, ike/config 파일, 284

## pkcs11\_path 키워드

사용, 267

설명, 284

## pntadm 명령, 설명, 194

policy files, ike/config 파일, 215

## policy 서비스

사용, 220, 229

설명, 237

## PPP 링크

문제 해결

패킷 흐름, 102

proxy 키워드, IKE 구성 파일, 271

publickeys 데이터베이스, 286



**Q**

-q 옵션, in.routed 데몬, 139  
 QoS(quality of service), QoS 정책, 390  
 QoS(서비스 품질), 작업, 387  
 QoS 정책, 390  
   계획 작업 맵, 406  
   구현, IPQoS 구성 파일, 421  
   정책 구성 템플릿, 405  
   필터 만들기, 409

**R**

RARP 프로토콜, 이더넷 주소 검사, 132  
 RBAC, IPsec 및, 217  
 RDISC, 설명, 139  
 RDISC(ICMP Router Discovery) 프로토콜, 139  
 RFC(Requests for Comments)  
   IKE, 201  
   IPQoS, 388  
   IPsec, 201  
 RIP(routing information protocol), 설명, 139  
 route 명령  
   inet6 옵션, 148  
   IPsec, 230  
 routeadm 명령  
   IP 전달, 227  
   IPv6 라우터 구성, 78  
 Router Advertisement, 180  
 RSA 암호화 알고리즘, 285

**S**

-S 옵션  
   ikecert certlocal 명령, 258  
   in.routed 데몬, 139  
 -s 옵션, ping 명령, 98  
 SA(보안 연결)  
   IPsec, 205-206, 219, 228  
   IPsec 데이터베이스, 240  
   IPsec 추가, 219, 228  
   수동으로 만들기, 231-232  
   정의, 200  
 SADB(보안 연결 데이터베이스), 240

SADB(보안 연결 데이터베이스)(계속)

  IPsec, 200  
 SCTP 프로토콜  
   IPsec 및, 217  
   IPsec 제한 사항, 213  
   SCTP 사용 서비스 추가, 70-73  
   상태 표시, 93  
   통계 표시, 91  
 services 데이터베이스, 업데이트, SCTP용, 71  
 SLA(서비스 단계 계약), 390  
   서로 다른 서비스 클래스 우선 순위 지정, 391  
   서비스 클래스, 393  
   청구 클라이언트, 흐름 계산 기반, 456  
 SMF 서비스, DHCP에서 사용, 196  
 SMF(서비스 관리 기능)  
   IKE 서비스  
     ike 서비스, 206, 246  
     구성 가능한 등록 정보, 281  
     다시 시작, 220  
     사용, 220  
     사용으로 설정, 275, 282  
     새로 고침, 232  
     설명, 281-282  
   IPsec 서비스, 237  
     ipsecalgs 서비스, 240  
     manual-key 사용, 232  
     manual-key 서비스, 241  
     manual-key 설명, 206  
     policy 서비스, 214  
     목록, 214-215  
     사용하여 IKE 관리, 234-235  
     사용하여 IPsec 관리, 234-235  
 snoop 명령  
   DHCP, 194  
   IP 계층에서 패킷 확인, 105-108  
   ip6 프로토콜 키워드, 148  
   IPv6 트래픽 모니터링, 105  
   IPv6에 대한 확장, 148  
   보호된 패킷 보기, 242  
   패킷 보호 확인, 235-236  
   패킷 콘텐츠 표시, 103  
   패킷 흐름 확인, 102  
 snoop명령, 서버와 클라이언트 간 패킷 확인, 104  
 sockets, netstat로 소켓 상태 표시, 94

softtoken 키 저장소, metaslot이 포함된 키  
저장소, 284  
SPD(보안 정책 데이터베이스)  
IPsec, 200, 201  
구성, 238  
SPI(보안 매개변수 색인), 설명, 205-206  
Stateless 주소 자동 구성, 154  
Sun Crypto Accelerator 6000 보드, IKE에서  
사용, 279-280  
/system/name-service/switch SMF 서비스,  
DHCP에서 사용, 196

## T

-T 옵션  
ikecert 명령, 267, 286  
ikecert certlocal 명령, 258  
-t 옵션  
ikecert certlocal 명령, 258  
ikecert 명령, 285  
inetd 데몬, 69  
TCP/IP 네트워크  
ESP로 보호, 207  
구성  
name-service/switch SMF 서비스, 137  
표준 TCP/IP 서비스, 69  
문제 해결, 104  
netstat 명령, 91  
ping 명령, 97, 98  
소프트웨어 검사, 131  
일반 방법, 131  
타사 진단 프로그램, 131  
패킷 손실, 98  
패킷 콘텐츠 표시, 103  
TCP/IP 프로토콜 모음, 통계 표시, 91  
TCP/IP 프로토콜 제품군, 표준 서비스, 69  
TCP 래퍼, 사용으로 설정, 73  
TCP 프로토콜, 통계 표시, 91  
/tftpboot 디렉토리 만들기, 55  
tokenmt 측정기, 394  
단일 속도 측정기, 465  
두 속도 측정기, 465  
색상 인식 구성, 394, 465  
속도 매개변수, 464

tokenmt 측정기 (계속)  
측정 속도, 464  
tokens 인수, ikecert 명령, 284  
traceroute 명령  
IPv6 확장, 149  
경로 추적, 102  
정의, 101-102  
Trusted Extensions, IPsec 및, 218  
tswtclmt 측정기, 394, 466  
측정 속도, 466  
tunnel 키워드  
IPsec 정책, 210, 224, 228  
tunnels, 터널 만들기 및 구성, 120-124

## U

UDP 프로토콜, 통계 표시, 91  
URI(Uniform Resource Indicator), CRL  
액세스용, 270  
use\_http 키워드, IKE 구성 파일, 271  
/usr/lib/inet/dhcdpd 데몬, 설명, 193  
/usr/lib/inet/dhcrelay 명령, 설명, 193  
/usr/lib/inet/in.dhcdpd 데몬, 설명, 193  
/usr/sadm/admin/bin/dhcdpmgr 명령, 설명, 193  
/usr/sbin/6to4relay 명령, 126  
/usr/sbin/dhcdpagent 명령, 설명, 194  
/usr/sbin/dhcdpconfig 명령, 설명, 194  
/usr/sbin/dhcdpinfo 명령, 설명, 194  
/usr/sbin/dhtadm 명령, 설명, 194  
/usr/sbin/in.rdisc 프로그램, 설명, 139  
/usr/sbin/in.routed 데몬  
공간 절약 모드, 139  
설명, 139  
/usr/sbin/inetd 데몬  
inetd의 상태 확인, 132  
서비스 시작 데몬, 69  
/usr/sbin/ipdam 명령, DHCP, 194  
/usr/sbin/omshell 명령, 설명, 194  
/usr/sbin/ping 명령, 98  
구문, 97  
설명, 97  
실행, 98  
/usr/sbin/pntadm 명령, 설명, 194  
/usr/sbin/snoop 명령, DHCP, 194

**V**

-V 옵션, snoop 명령, 242  
 /var/inet/ndpd\_state.interface 파일, 149  
 VPN, 참조 VPN(virtual private networks)  
 VPN(virtual private networks)  
   IPsec로 보호, 227-230  
   IPsec로 생성, 211  
   IPv4 예, 227-230  
   routeadm 명령으로 구성, 227

**개**

개인 키, 저장(IKE), 285

**계**

게이트웨이, 네트워크 토폴로지 내, 59

**경**

경계 라우터, 46  
 경계 라우터, 6to4 사이트, 115  
 경로 설정표  
   모든 경로 추적, 102  
   표시, 131  
 경로 지정  
   IPv6, 158  
   게이트웨이, 59  
   단일 인터페이스 호스트에서, 64  
   동적 경로 지정, 59  
   정적 경로 지정, 59  
   정적 구성, 64  
 경로 지정 테이블, 59  
   in.routed 데몬 만들기, 139  
   공간 절약 모드, 139  
   수동 구성, 60  
 경로 지정 프로토콜  
   RDISC  
     설명, 139  
   RIP  
     설명, 139  
   설명, 139

경로 지정 프로토콜 (계속)  
 연결된 경로 지정 데몬, 140

**계**

계산, 하드웨어에서 IKE 속도 향상, 279-280

**공**

공간 절약 모드, in.routed 데몬 옵션, 139  
 공개 키, 저장(IKE), 286  
 공개 키 인증서, 참조 인증서  
 공개 키 인증서로 IKE 구성(작업 맵), 256

**관**

관리 모델, 176

**구**

구성  
   CA 인증서로 IKE, 262-266  
   DHCP 클라이언트, 175  
   IKE, 251  
   ike/config 파일, 282  
   IPsec, 238  
   ipsecinit.conf 파일, 238-239  
   IPsec로 보호되는 VPN, 227-230  
   IPsec를 사용하여 터널 모드의 VPN, 227-230  
   IPv6 지원 라우터, 78  
   IPv6에 대해 인터페이스를 수동으로, 76-77  
   NAT 규칙, 295-296  
   TCP/IP 구성 파일, 135  
   TCP/IP 네트워크  
     name-service/switch SMF 서비스, 137  
     표준 TCP/IP 서비스, 69  
   공개 키 인증서로 IKE, 256, 257-262  
   라우터, 56, 139  
   개요, 56  
   모바일 시스템에서 IKE, 272-278  
   역할을 가진 네트워크 보안, 232-234

## 구성 (계속)

- 자체 서명된 인증서로 IKE, 257-262
- 주소 풀, 296-297
- 터널
  - 참조 터널
- 패킷 필터링 규칙, 292-294
- 하드웨어에서 인증서로 IKE, 266-270

## 구성 파일

- IP 필터 예, 291
- IP 필터에 대해 만들기, 326-327
- IPv6
  - /etc/inet/ipaddrsel.conf 파일, 145
  - /etc/inet/ndpd.conf 파일, 142-145, 144

## 권

## 권한 프로파일

- Network IPsec Management, 233
- Network Management, 233

## 규

## 규칙 세트

- 참조 IP 필터 참조
- NAT, 295-296
- 비활성
  - 참조 IP 필터
- 패킷 필터링, 292-297

## 기

- 기록된 패킷, 파일에 저장, 325
- 기본 라우터, 정의, 47
- 기본 주소 선택, 145-146
  - IPv6 주소 선택 정책 테이블, 109-110
  - 정의, 108-110

## 나

## 나열

- CRL(IPsec), 270

## 나열 (계속)

- metaslot의 토큰 ID, 280
- 알고리즘(IPsec), 208
- 인증서(IPsec), 259, 270
- 토큰 ID(IPsec), 279
- 하드웨어(IPsec), 279

## 네

## 네트워크 계획

- IP 주소 지정 체계, 27
- 네트워크 등록, 29
- 설계 결정, 25

## 네트워크 관리

- 네트워크 설계, 25
- 호스트 이름, 30

## 네트워크 구성

- IPv4 네트워크 구성 작업, 45
- IPv6 라우터, 78
- IPv6 사용 멀티홈 호스트, 76-77
- 구성
  - 서비스, 69

- 네트워크 구성 서버 설정, 55

## 라우터, 56

## 보안 구성, 197

## 호스트에서 IPv6 사용, 80-86

## 네트워크 구성 서버, 설정, 55

## 네트워크 데이터베이스

## ethers 데이터베이스

## 항목 검사, 132

## hosts 데이터베이스

## 항목 검사, 132

## name-service/switch SMF 서비스, 137

## name-service/switch SMF 서비스 및, 137

## 이름 서비스, 138

## 네트워크 보안, 구성, 197

## 네트워크 설계

## IP 주소 지정 체계, 27

## 개요, 25

## 도메인 이름 선택, 31

## 호스트 이름 지정, 30

## 네트워크 토폴로지, 자율 시스템, 45

**논**

- 논리적 도메인, IPsec 및, 213
- 논리적 인터페이스, 177, 178
  - DHCP 클라이언트 시스템, 186

**다**

- 다음 홉, 157
- 다중 네트워크 인터페이스, DHCP 클라이언트 시스템, 186

**대**

- 대역폭 규제, 391
  - 계획, QoS 정책에서, 408

**데**

- 데몬
  - in.iked 데몬, 244, 246, 282
  - in.ndpd 데몬, 149
  - in.ripngd 데몬, 78, 150
  - inetd 인터넷 서비스, 136
- 데이터 링크, 링크를 통해 IP 인터페이스 구성, 48
- 데이터그램, IP, 199
- 데이터베이스
  - IKE, 284-286
  - ike/crls 데이터베이스, 286
  - ike.privatekeys 데이터베이스, 285, 286
  - ike/publickeys 데이터베이스, 285, 286
  - SADB(보안 연결 데이터베이스), 240
  - SPD(보안 정책 데이터베이스), 200

**도**

- 도메인 이름
  - nis/domain SMF 서비스, 53, 54
  - 선택, 31

**동**

- 동적 경로 지정, 최적 사례, 60

**등**

- 등록, 네트워크, 29

**디**

- 디렉토리
  - /etc/inet, 247
  - /etc/inet/ike, 247
  - /etc/inet/publickeys, 286
  - /etc/inet/secret, 247
  - /etc/inet/secret/ike.privatekeys, 285
  - 개인 키(IKE), 285
  - 공개 키(IKE), 286
  - 미리 공유한 키(IKE), 284
  - 인증서(IKE), 286
- 디렉토리 이름(DN), CRL 액세스용, 270
- 디지털 서명
  - DSA, 285
  - RSA, 285

**라**

- 라우터
  - IPv6에 대한 업그레이드 문제, 132
  - 경로 지정 프로토콜
    - 설명, 139
  - 구성, 139
    - IPv6, 78
    - 로컬 파일 모드 구성, 53
    - 역할, 6to4 토폴로지, 114
    - 정의, 56, 139
    - 패킷 전달 라우터, 47
  - 라우터 검색, IPv6, 149, 154, 157
  - 라우터 구성, IPv4 라우터, 56
  - 라우터 알림
    - IPv6, 153, 154, 157, 158-159
    - 접두어, 154

## 라우터 요청

IPv6, 153, 154

## 래

래퍼, TCP, 73

## 로

## 로그 파일

IP 필터에 대해 만들기, 322-323

IP 필터에 대해 보기, 323-324

IP 필터에서 비우기, 324

## 로드 균형 조정

IPQoS 사용 네트워크에서, 403

IPv6 지원 네트워크에서, 156

로컬 파일, 이름 서비스로 선택, 31

로컬 파일 이름 서비스, /etc/inet/hosts 파일, 219

## 릴

릴레이 라우터, 6to4 터널 구성, 126, 127

## 링

링크 계층 주소 변경, 156

## 링크로컬 주소

IPv6, 154, 158

수동 구성, 토큰 사용, 85

## 만

## 만들기

IPsec SA, 219, 231-232

ipsecinit.conf 파일, 219

보안 관련 역할, 232-234

인증서 요청, 262

자체 서명된 인증서(IKE), 257

## 멀

멀티캐스트 주소, IPv6, 브로드캐스트 주소와  
비교, 157

멀티홉 시스템, 정의, 47

## 멀티홉 호스트

IPv6에 대해 사용, 76-77

정의, 62

## 메

메시지, 라우터 알림, 159

## 명

## 명령

IKE, 284-286

ikeadm 명령, 247, 282, 283

ikecert 명령, 247, 282, 284

in.iked 데몬, 282

## IPsec

in.iked 명령, 206

ipsecalgs 명령, 208, 240

ipsecconf 명령, 214, 238

ipseckey 명령, 214, 241-242

snoop 명령, 242

목록, 214-215

보안 고려 사항, 241-242

## 모

모바일 시스템에 대한 IKE 구성(작업 맵), 272

## 문

## 문제 해결

IKE 페이로드, 266

IPv6 문제, 132-134

PPP 링크 확인

패킷 흐름, 102

TCP/IP 네트워크

in.ndpd 작업 추적, 100-101

## 문제 해결, TCP/IP 네트워크 (계속)

in.routed 작업 추적, 100  
 IP 계층에서 패킷 전송 모니터링, 105-108  
 netstat 명령으로 네트워크 상태  
   모니터링, 91  
 ping 명령, 98  
 snoop 명령으로 패킷 전송 모니터링, 102  
 traceroute 명령, 101-102  
 소프트웨어 검사, 131  
 알려진 경로의 상태 표시, 96-97  
 원격 호스트 검사 ping 명령, 97  
 인터페이스에서 전송 관찰, 93-94  
 일반 방법, 131  
 전송 프로토콜 상태 표시, 92-93  
 클라이언트와 서버 간 패킷 확인, 104  
 타사 진단 프로그램, 131  
 패킷 손실, 98  
 프로토콜별 통계 표시, 91-92

## 미

## 미리 공유한 키(IKE)

1단계 알고리즘 및 그룹 보기, 249-251  
 바꾸기, 254  
 설명, 245  
 작업 맵, 251  
 저장, 284

미리 공유한 키로 IKE 구성(작업 맵), 251

## 바

바꾸기, 미리 공유한 키(IKE), 254

## 보

## 보기

IPsec 구성, 238-239  
 IPsec 정책, 222-223

## 보안

IKE, 282  
 IPsec, 199

## 보안 고려 사항

6to4 릴레이 라우터 문제, 133-134  
 AH(authentication header), 207  
 ESP(encapsulating security payload), 207  
 ike/config 파일, 282  
 ipsecconf 명령, 239  
 ipsecinit.conf 파일, 239  
 ipseckey 명령, 241-242  
 ipseckey 파일, 232  
 IPv6 지원 네트워크, 41  
 구성

  IPsec, 219

  미리 공유한 키, 245  
   보안 프로토콜, 207  
   잠긴 소켓, 239

## 보안 연관(SA)

IKE, 282  
 ISAKMP, 244  
 난수 생성, 245

## 보안 정책

ike/config 파일(IKE), 215  
 IPsec, 209  
 ipsecinit.conf 파일(IPsec), 238-239

## 보안 프로토콜

AH(authentication header), 206  
 ESP(encapsulating security payload), 207-208  
 IPsec 보호 방식, 206  
 개요, 200  
 보안 고려 사항, 207

## 보호

IPsec 트래픽, 199  
 IPsec로 모바일 시스템, 272-278  
 IPsec를 사용하여 웹 서버, 221-222  
 두 시스템 사이의 패킷, 218-221  
 터널 모드에서 IPsec 터널로 VPN, 227-230

보호 방식, IPsec, 206-208

## 분

## 분류기 모듈, 392

  action 명령문, 426  
   분류기의 기능, 462

## 비

비우기, **참조** 삭제  
비활성 규칙 세트, **참조** IP 필터

## 사

사용자 우선 순위 값, 394  
사이트 접두어, IPv6  
    알림, 라우터에, 79  
    확인 방법, 37  
사전 공유된 키(IPsec), 만들기, 231–232

## 삭

삭제 또는 손실된 패킷, 98

## 삼

삼중 DES 암호화 알고리즘, IPsec 및, 208

## 상

상태 테이블, 보기, 319–320  
상태 통계, 보기, 320

## 새

새 기능  
    DHCP 이벤트 스크립트, 189–191  
    논리적 인터페이스의 DHCP, 186  
새로 고침, 미리 공유한 키(IKE), 254  
새로운 기능  
    inetconv 명령, 55  
    IPv6의 임시 주소, 80–83  
    routeadm 명령, 78  
    SCTP 프로토콜, 70–73  
    SMF(Service Management Facility), 56  
    기본 주소 선택, 108–110  
    링크 로컬 주소 수동 구성, 83–85

## 색

색상 인식, 394, 465

## 서

서버, DHCPv6, 176  
서버, IPv6  
    IPv6 사용, 85–86  
    작업 계획, 36  
서브넷, 31  
    IPv4  
        넷마스크 구성, 53  
    IPv4 네트워크에 추가, 67–69  
    IPv6  
        6to4 토폴로지, 115  
        번호 지정 제안 사항, 38  
서비스 클래스, **참조** 클래스

## 선

선택기, 393  
    IPQoS 5-튜플, 393  
    계획, QoS 정책에서, 409  
    선택기, 목록, 462

## 소

소켓, IPsec 보안, 239  
소프트 토큰 키 저장소, metaslot이 있는 키  
    저장소, 280

## 속

속도 향상, IKE 계산, 279

## 손

손실 또는 삭제된 패킷, 98



**슬**

슬롯, 하드웨어, 286

**시**

시스템

통신 보호, 218-221

**암**

암호화 알고리즘

IKE 미리 공유한 키, 249-251

IPsec

3DES, 208

AES, 208

Blowfish, 208

DES, 208

암호화 프레임워크, IPsec, 및, 240

**애**

애니캐스트 그룹, 6to4 릴레이 라우터, 126

애니캐스트 주소, 126

애플리케이션 서버, IPQoS에 대한 구성, 436

**역**

역순 영역 파일, 86

역할, 네트워크 보안 역할 만들기, 232-234

**영**

영역

IPsec 및, 213, 217

키 관리 및, 217

영역 파일, 86

**예**

예제 IPQoS 구성 파일

VLAN 장치 구성, 469

색상 인식 세그먼트, 465

애플리케이션 서버, 436

최선 조건 웹 서버, 425

프리미엄 웹 서버, 423

**옵**

옵션 요청, 178

**우**

우회

IPsec 정책, 209

LAN의 IPsec, 228

**웹**

웹 서버

IPQoS에 대한 구성, 423, 425, 433, 434

IPsec를 사용하여 보호, 221-222

**이**

이름 서비스

네트워크 데이터베이스 및, 138

데이터베이스 검색 순서 지정, 137

서비스 선택, 30

이름/이름 지정

노드 이름

로컬 호스트, 54

이웃 연결 불가 감지

IPv6, 155, 157

이웃 요청, IPv6, 153

**인**

인바운드 로드 균형 조정, 156

## 인증 알고리즘

IKE 미리 공유한 키, 249-251

IKE 인증서, 285

## 인증서

CA에서, 263

CRL 무시, 265

IKE, 246

ike/config 파일, 268

나열, 259

데이터베이스에 추가, 263

설명, 263

## 요청

CA에서, 262

하드웨어에서, 267

자체 서명 만들기(IKE), 257

## 저장

IKE, 286

컴퓨터에서, 257

하드웨어에, 279

하드웨어의 CA에서, 269

## 인증서 요청

CA에서, 262

사용, 285

하드웨어에서, 267

## 인증서 해지 목록, 참조 CRL

## 인터넷 초안, IPsec에서 SCTP, 201

## 인터페이스

## 구성

IPv6에 대해 수동으로, 76-77

데이터 링크를 통해, 48

임시 주소, 80-83

지속 구성 만들기, 50

패킷 확인, 103

인터페이스 ID, 수동으로 구성된 토큰 사용, 85

## 임

임시 주소, IPv6

구성, 81-83

정의, 80-83

## 자

자율 시스템(AS), 참조 네트워크 토폴로지

## 작

## 작업 맵

IKE 구성(작업 맵), 251

## IPQoS

QoS 정책 계획, 406

구성 계획, 401

구성 파일 만들기, 421

흐름 계산 설정, 455

IPsec를 사용하여 트래픽 보호(작업 맵), 218

## IPv6

계획, 33-34

공개 키 인증서로 IKE 구성(작업 맵), 256

네트워크 관리 작업, 90

모바일 시스템에 대한 IKE 구성(작업 맵), 272

미리 공유한 키로 IKE 구성(작업 맵), 251

## 재

## 재 지정

IPv6, 153, 157

## 저

## 저장

디스크의 IKE 키, 286

하드웨어에 IKE 키, 279-280

## 전

## 전송 계층

## TCP/IP

SCTP 프로토콜, 70-73

전송 프로토콜 상태 표시, 92-93

## 전송 모드

AH로 데이터 보호, 210

ESP로 보호된 데이터, 210

IPsec, 209-211

**접**

## 접두어

라우터 알람, 154, 157, 159

**정**

정렬, 디스크의 IKE 키, 263

## 정적 경로 지정

구성 예, 61

정적 경로 지정 추가, 60-61

최적 사례, 60

호스트에서 수동 구성, 64

정책, IPsec, 209

## 정책 파일

ike/config 파일, 247, 282

ipsecinit.conf 파일, 238-239

보안 고려 사항, 239

**주**

## 주소

기본 주소 선택, 108-110

입시, IPv6, 80-83

## 주소 자동 구성

IPv6, 149, 153

## 주소 풀

개요, 296-297

구성, 296-297

보기, 317

제거, 317

추가, 318

통계 보기, 321

**중**

중복 주소 감지, 알고리즘, 155

**지**

지속 링크 구성, 만들기, 50

**차**

차별화 서비스, 387

서로 다른 서비스 클래스 제공, 391

차별화 서비스 모델, 392

차별화된 서비스, 네트워크 토폴로지, 402

**추**

## 추가

CA 인증서(IKE), 262-266

IPsec SA, 219, 231-232

공개 키 인증서(IKE), 262-266

미리 공유한 키(IKE), 254-256

수동으로 키(IPsec), 231-232

자체 서명된 인증서(IKE), 257

**측**

## 측정 모듈

참조 tokenmt 측정기

참조 tswtclmt 측정기

소개, 394

측정 결과, 394, 464

호출, IPQoS 구성 파일, 443

**클**

클라이언트 ID, 177

클라이언트 구성, 176

클래스, 393

class 절의 구문, 475

선택기, 목록, 462

정의, IPQoS 구성 파일, 434, 438

클래스 A, B 및 C 네트워크 번호, 27

**키**

## 키

ike.privatekeys 데이터베이스, 286

ike/publickeys 데이터베이스, 286

IPsec SA에 대해 만들기, 231-232

## 키 (계속)

- IPsec 관리, 205-206
- 미리 공유(IKE), 245
- 수동 관리, 241-242
- 자동 관리, 244
- 저장(IKE)
  - 개인, 285
  - 공개 키, 286
  - 인증서, 286
- 키 관련 유틸리티, IKE 프로토콜, 243
- 키 관리
  - IKE, 244
  - ike 서비스, 206
  - IPsec, 205-206
  - manual-key 서비스, 206
  - 수동, 241-242
  - 영역 및, 217
  - 자동, 244
- 키 입력 유틸리티
  - ike 서비스, 206
  - ipseckey 명령, 206
  - manual-key 서비스, 206
- 키 저장소
  - IPsec SA, 214
  - ISAKMP SA, 284
  - metaslot의 토큰 ID, 280
  - softtoken, 284
  - 소프트 토큰 키 저장소, 280
- 키 저장소 이름, **참조** 토큰 ID

## 터

- 터널, 111-130
  - 6to4 터널, 113
    - 토폴로지, 114
    - 패킷 흐름, 115, 117
  - 6to4 터널 구성, 125
  - dladm 명령
    - create-iptun, 120-124
    - delete-iptun, 130
    - modify-iptun, 128-129
    - show-iptun, 129
  - 터널 구성을 위한 하위 명령, 119-120
  - dladm 명령으로 구성, 119-130

## 터널 (계속)

- encaplimit, 122
- hoplimit, 122
- IP 터널 삭제, 130
- IPsec, 211
- IPsec의 모드, 209-211
- IPv4, 112-113
- IPv4 over IPv4 터널 만들기, 123
- IPv6, 112-113
- IPv6 over IPv4 터널 만들기, 123
- IPv6 over IPv6 터널 만들기, 124
- IPv6 구성
  - 6to4 릴레이 라우터에 대한, 126
- IPv6 터널링 방식, 112
- VPN
  - 참조** VPN(Virtual Private Network)
- 계획, IPv6, 40-41
- 로컬 및 원격 주소, 128
- 만들기 요구 사항, 118-119
- 배치, 118-119
- 유형, 111
  - 6to4, 112
  - IPv4, 112
  - IPv4 over IPv4, 112
  - IPv4 over IPv6, 112
  - IPv6, 112
  - IPv6 over IPv4, 112
  - IPv6 over IPv6, 112
- 전송 모드, 209
- 터널 구성 수정, 128-129
- 터널 대상 주소
  - 참조** 터널, *tdst*
- 터널 모드, 209
- 터널 소스 주소
  - 참조** 터널, *tsrc*
- 터널 정보 표시, 129
- 토폴로지, 6to4 릴레이 라우터, 117
- 패킷 보호, 211
- 패킷 캡슐화, 111
- 필요한 IP 인터페이스, 118-119
- 터널 모드
  - IPsec, 209-211
  - 전체 내부 IP 패킷 보호, 211

**토**

토큰 ID, 하드웨어, 286

**통****통계**

패킷 전송(ping), 98  
프로토콜별(netstat), 91

**트****트래픽 관리**

네트워크 토폴로지 계획, 402  
대역폭 규제, 391  
트래픽 전달, 397, 398, 399  
트래픽 흐름 우선 순위 지정, 391  
흐름 제어, 394

**트래픽 전달**

Diffserv 네트워크를 통한 트래픽 흐름, 398  
IP 패킷 전달, DSCP 사용, 397  
계획, QoS 정책에서, 408  
데이터그램 전달, 469  
패킷 전달에 대한 PHB의 효과, 466

**트래픽 준수**

결과, 394, 464  
계획  
QoS 정책 결과, 413  
QoS 정책의 속도, 412  
속도 매개변수, 464  
정의, 443

**파****파일****IKE**

crIs 디렉토리, 247, 286  
ike/config 파일, 215, 245, 247, 282  
ike.preshared 파일, 247, 284  
ike.privatekeys 디렉토리, 247, 286  
publickeys 디렉토리, 247, 286

**IPsec**

ipseccinit.conf 파일, 214, 238-239

**파일, IPsec (계속)**

ipseckey 파일, 214

**패****패킷**

IP 계층에서 관찰, 105-108  
보호

IKE, 244  
IPsec 사용, 202, 206-208  
아웃바운드 패킷, 202  
인바운드 패킷, 202

보호 확인, 235-236  
삭제 또는 손실됨, 98  
컨텐츠 표시, 103  
흐름 확인, 102

패킷 전달 라우터, 47

패킷 필터 후크, 297

**패킷 필터링**

구성, 292-294  
규칙 세트 간 전환, 313-314  
규칙 세트 관리, 308-314  
다른 규칙 세트 활성화, 309-310  
비활성화, 305-306

**제거**

비활성 규칙 세트, 314  
활성 규칙 세트, 311

**추가**

비활성 세트에 규칙, 312-313  
활성 세트에 규칙, 311-312  
현재 규칙 세트 업데이트 후 재로드, 309-310

**패킷 흐름**

릴레이 라우터, 117  
터널 경유, 115

**패킷 흐름, IPv6**

6to4 및 원시 IPv6, 117  
6to4 터널 경유, 115

**표**

표시, IPsec 정책, 222-223

표시기 모듈, 394

참조 dlcosmk 표시기

표시기 모듈 (계속)

- 참조 dscpmk 표시기
- DS 코드 포인트 지정, 468
- PHB, IP 패킷 전달, 397
- VLAN 장치 지원, 469

프

- 프로토콜 통계 표시, 91

필

- 필터, 393
  - filter 절 구문, 476
  - 계획, QoS 정책에서, 409
  - 만들기, IPQoS 구성 파일, 434, 439
  - 선택기, 목록, 462

하

- 하드웨어
  - IKE 계산 속도 향상, 279
  - IKE 키 저장, 279–280
  - 연결된 하드웨어 찾기, 279

호

- 호스트
  - IP 연결 확인, 98
  - IPv6에 대한 구성, 80–86
  - 멀티홈
    - 구성, 62
  - 일반 문제 해결, 131
  - 임시 IPv6 주소, 80–83
  - 호스트 연결 확인 ping, 97
  - 호스트 이름
    - 관리, 30
- 호스트 이름, 클라이언트 요청 사용, 187

화

- 확인
  - ipseccinit.conf 파일
    - 구문, 220, 228
  - ipseckeykeys 파일
    - 구문, 232
  - 패킷 보호, 235–236

활

- 활성 규칙 세트, 참조 IP 필터

흐

- 흐름 계산, 456, 470
  - 흐름 레코드 테이블, 471
- 흐름 정산에 대한 acctadm 명령, 395
- 흐름 제어, 측정 모듈을 통해, 394