Oracle® Solaris 11.1의 네트워크 보안

부품 번호: E36647–02
2013년 2월

ORACLE®

Copyright © 1999, 2013, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이센스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이센스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이센스, 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 주기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이센스한 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, 라이센스 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 라이센스에 의거하여 사용되며SPARC 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 등록상표입니다.

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

130326@25097

목차

머리말

이 설명서에서는 Oracle Solaris 운영 체제(Oracle Solaris OS)가 설치되어 있고 네트워크를 보호할 준비가 완료되었다고 가정합니다.

주 – 본 Oracle Solaris 릴리스는 프로세서 아키텍처의 SPARC 및 x86 제품군을 사용하는 시스템을 지원합니다. 지원되는 시스템은 **Oracle Solaris OS: 하드웨어 호환성 목록**을 참조하십시오. 이 설명서에서는 플랫폼 유형에 따른 구현 차이가 있는 경우 이에 대하여 설명합니다.

이 설명서의 대상

본 설명서는 Oracle Solaris를 실행하는 네트워크에 연결된 시스템을 관리하는 사용자를 대상으로 작성되었습니다. 이 설명서를 사용하려면 적어도 2년의 UNIX 시스템 관리 경험이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 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	설명	예
`AaBbCc123`	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	`.login` 파일을 편집하십시오. 모든 파일 목록을 보려면 `ls -a` 명령을 사용하십시오. `machine_name% you have mail.`
`AaBbCc123`	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	`machine_name%` **`su`** `Password:`
AaBbCc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	`rm` *filename* 명령을 사용하여 파일을 제거합니다.
AaBbCc123	책 제목, 장, 절	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 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 UNIX 시스템 프롬프트 및 수퍼유저 프롬프트가 나와 있습니다. 명령 예에서 셸 프롬프트는 명령을 일반 사용자가 실행해야 하는지 또는 권한 있는 사용자가 실행해야 하는지 나타냅니다.

표 P–2 쉘 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	`$`
수퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	`#`
C 셸	`machine_name%`
수퍼유저용 C 셸	`machine_name#`

◆ ◆ ◆ 1 장

1 가상화된 환경에서 링크 보호 사용

이 장에서는 링크 보호 및 Oracle Solaris 시스템에서 링크 보호를 구성하는 방법에 대해 설명합니다. 이 장에서는 다음 주제를 다룹니다.

링크 보호 개요

시스템 구성에서 가상화를 채택하는 경우가 많아지면서 호스트 관리자에 의해 게스트 VM(가상 컴퓨터)이 물리적 링크 또는 가상 링크에 배타적으로 액세스할 수 있게 되었습니다. 이렇게 구성하면 가상 환경의 네트워크 트래픽이 호스트 시스템에서 수신 또는 전송되는 더 넓은 트래픽에서 격리될 수 있으므로 네트워크 성능이 향상됩니다. 동시에 이 구성으로 인해 시스템과 전체 네트워크가 게스트 환경에서 생성될 수 있는 유해한 패킷의 위험에 노출될 수 있습니다.

링크 보호는 잠재적으로 악의적인 게스트 VM에서 네트워크에 초래할 수 있는 손상을 방지하기 위한 것입니다. 이 기능은 다음의 기본적인 위협으로부터의 보호를 제공합니다.

- IP, DHCP 및 MAC 스푸핑
- BPDU(Bridge Protocol Data Unit) 공격과 같은 L2 프레임 스푸핑

주 – 링크 보호는 특히 복잡한 필터링 요구 사항이 있는 구성에 있어 방화벽 배포를 대체하지 않습니다.

링크 보호 유형

Oracle Solaris의 링크 보호 방식은 다음 보호 유형을 제공합니다.

`mac-nospoof`
시스템의 MAC 주소 스푸핑에 대한 보호를 제공할 수 있습니다. 링크가 특정 영역에 속해 있는 경우 `mac-nospoof`를 사용하면 영역의 소유자가 해당 링크의 MAC 주소를 수정할 수 없습니다.

`ip-nospoof`
IP 스푸핑에 대한 보호를 제공할 수 있습니다. 기본적으로 DHCP 주소 및 링크 로컬 IPv6 주소가 있는 아웃바운드 패킷이 허용됩니다.

`allowed-ips` 링크 등록 정보를 사용하여 주소를 추가할 수 있습니다. IP 주소의 경우 패킷의 소스 주소가 `allowed-ips` 목록의 주소와 일치해야 합니다. ARP 패킷의 경우 패킷의 발신자 프로토콜 주소가 `allowed-ips` 목록에 있어야 합니다.

`dhcp-nospoof`
DHCP 클라이언트 스푸핑에 대한 보호를 제공할 수 있습니다. 기본적으로 ID가 시스템의 MAC 주소와 일치하는 DHCP 패킷이 허용됩니다.

`allowed-dhcp-cids` 링크 등록 정보를 사용하여 허용되는 클라이언트를 추가할 수 있습니다. `allowed-dhcp-cids` 목록의 항목은 `dhcpagent(1M)` 매뉴얼 페이지에 지정된 대로 형식이 지정되어야 합니다.

`restricted`
IPv4, IPv6 및 ARP로 나가는 패킷을 제한합니다. 이 보호 유형은 링크에서 잠재적으로 유해한 L2 컨트롤 프레임이 생성되지 못하도록 방지하기 위해 설계되었습니다.

주 – 링크 보호로 인해 삭제되는 패킷은 `mac_spoofed`, `dhcp_spoofed`, `ip_spoofed` 및 `restricted`의 네 가지 보호 유형에 대한 커널 통계에서 추적됩니다. 이러한 링크별 통계를 검색하려면 15 페이지 “링크 보호 구성 및 통계를 확인하는 방법”을 참조하십시오.

링크 보호 구성(작업 맵)

링크 보호를 사용하려면 링크의 `protection` 등록 정보를 설정합니다. 보호 유형이 다른 구성 파일과 작동(예: `ip-nospoof`는 `allowed-ips` 또는 `dhcp-nospoof`는 `allowed-dhcp-cids`)하는 경우 두 가지 일반 작업을 수행합니다. 우선 링크 보호를 사용으로 설정합니다. 그런 다음 구성 파일을 사용자 정의하여 통과하도록 허용할 기타 패킷을 식별합니다.

주 – 전역 영역에 링크 보호를 구성해야 합니다.

다음 작업 맵에서는 Oracle Solaris 시스템에서 링크 보호를 구성하기 위한 절차를 안내합니다.

작업	설명	수행 방법
링크 보호를 사용으로 설정합니다.	링크에서 보내는 패킷을 제한하고 스푸핑으로부터 링크를 보호합니다.	13 페이지 "링크 보호를 사용으로 설정하는 방법"
링크 보호를 사용 안함으로 설정합니다.	링크 보호를 제거합니다.	14 페이지 "링크 보호를 사용 안함으로 설정하는 방법"
IP 링크 보호 유형을 지정합니다.	링크 보호 방식을 통과할 수 있는 IP 주소를 지정합니다.	14 페이지 "IP 스푸핑으로부터 보호할 IP 주소를 지정하는 방법"
DHCP 링크 보호 유형을 지정합니다.	링크 보호 방식을 통과할 수 있는 DHCP 주소를 지정합니다.	15 페이지 "DHCP 스푸핑으로부터 보호할 DHCP 클라이언트를 지정하는 방법"
링크 보호 구성을 확인합니다.	보호되는 링크와 예외를 나열하고 적용 통계를 표시합니다.	15 페이지 "링크 보호 구성 및 통계를 확인하는 방법"

▼ 링크 보호를 사용으로 설정하는 방법

이 절차에서는 나가는 패킷 유형을 제한하고 링크 스푸핑을 방지합니다.

시작하기 전에 Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 사용 가능한 링크 보호 유형을 확인합니다.

```
# dladm show-linkprop -p protection
LINK      PROPERTY      PERM VALUE    DEFAULT    POSSIBLE
vnic0     protection    rw   --       --         mac-nospoof,
                                                  restricted,
                                                  ip-nospoof,
                                                  dhcp-nospoof
```

가능한 유형에 대한 설명은 11 페이지 "링크 보호 유형" 및 `dladm`(1M) 매뉴얼 페이지를 참조하십시오.

2 보호 유형을 하나 이상 지정하여 링크 보호를 사용으로 설정합니다.

```
# dladm set-linkprop -p protection=value[,value,...] link
```

다음 예에서는 `vnic0` 링크에서 네 개의 모든 링크 보호가 사용으로 설정되어 있습니다.

```
# dladm set-linkprop \
-p protection=mac-nospoof,restricted,ip-nospoof,dhcp-nospoof vnic0
```

3 링크 보호가 사용으로 설정되어 있는지 확인합니다.

```
# dladm show-linkprop -p protection vnic0
LINK    PROPERTY       PERM VALUE         DEFAULT      POSSIBLE
vnic0   protection     rw   mac-nospoof   --           mac-nospoof,
                            restricted                 restricted,
                            ip-nospoof                 ip-nospoof,
                            dhcp-nospoof               dhcp-nospoof
```

VALUE 아래의 링크 보호 유형은 해당 보호가 사용으로 설정되어 있음을 나타냅니다.

▼ 링크 보호를 사용 안함으로 설정하는 방법

이 절차에서는 링크 보호를 기본값인 링크 보호 안함으로 재설정합니다.

시작하기 전에 Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 protection 등록 정보를 기본값으로 재설정하여 링크 보호를 사용 안함으로 설정합니다.

```
# dladm reset-linkprop -p protection link
```

2 링크 보호가 사용 안함으로 설정되어 있는지 확인합니다.

```
# dladm show-linkprop -p protection vnic0
LINK      PROPERTY      PERM VALUE    DEFAULT    POSSIBLE
vnic0     protection    rw   --       --         mac-nospoof,
                                                 restricted,
                                                 ip-nospoof,
                                                 dhcp-nospoof
```

VALUE 아래 목록에 링크 보호 유형이 없으면 해당 링크 보호가 사용 안함으로 설정되어 있음을 나타냅니다.

▼ IP 스푸핑으로부터 보호할 IP 주소를 지정하는 방법

시작하기 전에 13 페이지 "링크 보호를 사용으로 설정하는 방법"에 나온 대로 ip-nospoof 보호 유형이 사용으로 설정되어 있습니다.

Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 스푸핑으로부터 보호하도록 설정했는지 확인합니다.

```
# dladm show-linkprop -p protection link
LINK     PROPERTY       PERM    VALUE          DEFAULT      POSSIBLE
link     protection     rw      ...
                                ip-nospoof                  ip-nospoof
```

VALUE 아래 ip-nospoof 목록은 이 보호 유형이 사용으로 설정되어 있음을 나타냅니다.

2 allowed-ips 링크 등록 정보의 기본값 목록에 IP 주소를 추가합니다.

```
# dladm set-linkprop -p allowed-ips=IP-addr[,IP-addr,...] link
```

다음 예에서는 vnic0 링크의 allowed-ips 등록 정보에 IP 주소 10.0.0.1 및 10.0.0.2를 추가하는 방법을 보여 줍니다.

```
# dladm set-linkprop -p allowed-ips=10.0.0.1,10.0.0.2 vnic0
```

자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

▼ DHCP 스푸핑으로부터 보호할 DHCP 클라이언트를 지정하는 방법

시작하기 전에 13 페이지 "링크 보호를 사용으로 설정하는 방법"에 나온 대로 dhcp-nospoof 보호 유형이 사용으로 설정되어 있습니다.

Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 DHCP 스푸핑으로부터 보호하도록 설정했는지 확인합니다.

```
# dladm show-linkprop -p protection link
LINK     PROPERTY        PERM    VALUE          DEFAULT       POSSIBLE
link     protection      rw      ...
                                 dhcp-nospoof                 dhcp-nospoof
```

VALUE 아래 dhcp-nospoof 목록은 이 보호 유형이 사용으로 설정되어 있음을 나타냅니다.

2 allowed-dhcp-cids 링크 등록 정보에 대해 ASCII 구문을 지정합니다.

```
# dladm set-linkprop -p allowed-dhcp-cids=CID-or-DUID[,CID-or-DUID,...] link
```

다음 예에서는 hello 문자열을 vnic0 링크의 allowed-dhcp-cids 등록 정보 값으로 지정하는 방법을 보여 줍니다.

```
# dladm set-linkprop -p allowed-dhcp-cids=hello vnic0
```

자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

▼ 링크 보호 구성 및 통계를 확인하는 방법

시작하기 전에 Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 링크 보호 등록 정보 값을 확인합니다.

```
# dladm show-linkprop -p protection,allowed-ips,allowed-dhcp-cids link
```

다음 예에서는 vnic0 링크의 protection, allowed-ips 및 allowed-dhcp-cids 등록 정보 값을 보여 줍니다.

```
# dladm show-linkprop -p protection,allowed-ips,allowed-dhcp-cids vnic0
LINK     PROPERTY            PERM     VALUE           DEFAULT     POSSIBLE
vnic0    protection          rw       mac-nospoof     --          mac-nospoof,
                                      restricted                  restricted,
                                      ip-nospoof                  ip-nospoof,
                                      dhcp-nospoof                dhcp-nospoof
```

```
vnic0   allowed-ips          rw       10.0.0.1,       --          --
                                      10.0.0.2
vnic0   allowed-dhcp-cids    rw       hello           --          --
```

주 – allowed-ips 등록 정보는 VALUE 아래 나열된 것과 같이 ip-nospoof가 사용으로 설정되어 있는 경우에만 사용됩니다. allowed-dhcp-cids 등록 정보는 dhcp-nospoof가 사용으로 설정되어 있는 경우에만 사용됩니다.

2 링크 보호 통계를 확인합니다.

dlstat 명령의 출력은 커밋되므로 이 명령은 스크립트에 적합합니다.

```
# dlstat -A
...
 vnic0
  mac_misc_stat
               multircv                      0
              brdcstrcv                      0
               multixmt                      0
              brdcstxmt                      0
           multircvbytes                     0
            bcstrcvbytes                     0
           multixmtbytes                     0
            bcstxmtbytes                     0
                txerrors                     0
              macspoofed                     0  <----------
               ipspoofed                     0  <----------
             dhcpspoofed                     0  <----------
              restricted                     0  <----------
                ipackets                     3
                  rbytes                   182
...
```

출력은 스푸핑되거나 제한된 패킷이 통과를 시도하지 않았음을 나타냅니다.

kstat 명령을 사용할 수 있지만 해당 출력은 커밋되지 않습니다. 예를 들어, 다음 명령은 dhcpspoofed 통계를 찾습니다.

```
# kstat vnic0:0:link:dhcpspoofed
module: vnic0                           instance: 0
name:   link                            class:    vnic
        dhcpspoofed                     0
```

자세한 내용은 dlstat(1M) 및 kstat(1M) 매뉴얼 페이지를 참조하십시오.

◆ ◆ ◆ 2 장

2 네트워크 조정(작업)

이 장에서는 Oracle Solaris에서 보안에 영향을 미치는 네트워크 매개변수를 조정하는 방법에 대해 설명합니다.

네트워크 조정(작업 맵)

작업	설명	수행 방법
네트워크 경로 지정 데몬을 사용 안함으로 설정합니다.	잠재적인 네트워크 스니퍼에 의한 시스템 액세스를 제한합니다.	18 페이지 "네트워크 경로 지정 데몬을 사용 안함으로 설정하는 방법"
네트워크 토폴로지 정보에 대한 배포를 방지합니다.	패킷 브로드캐스트를 방지합니다.	19 페이지 "브로드캐스트 패킷 전달을 사용 안함으로 설정하는 방법"
	브로드캐스트 에코 요청 및 멀티캐스트 에코 요청에 대한 응답을 방지합니다.	19 페이지 "에코 요청에 대한 응답을 사용 안함으로 설정하는 방법"
다른 도메인에 대한 게이트웨이인 시스템(예: 방화벽 또는 VPN 노드)의 경우 엄격한 소스 및 대상 다중 홈 지정을 설정합니다.	헤더의 게이트웨이 주소를 포함하지 않는 패킷이 게이트웨이 외부로 이동하지 않도록 방지합니다.	20 페이지 "엄격한 다중 홈 지정을 설정하는 방법"
완전하지 않은 시스템 연결 개수를 제한하여 DOS 공격을 방지합니다.	TCP 리스너에 대해 완전하지 않은 TCP 연결의 허용 가능한 개수를 제한합니다.	21 페이지 "완전하지 않은 TCP 연결의 최대 개수를 설정하는 방법"
허용된 수신 연결 개수를 제한하여 DOS 공격을 방지합니다.	TCP 리스너에 대한 보류 중인 TCP 연결의 기본 최대 개수를 지정합니다.	21 페이지 "보류 중인 TCP 연결의 최대 개수를 설정하는 방법"
초기 TCP 연결에 대한 높은 수준의 난수를 생성합니다.	RFC 6528에 의해 지정된 시퀀스 번호 생성 값을 준수합니다.	22 페이지 "초기 TCP 연결에 대한 높은 수준의 난수를 지정하는 방법"

작업	설명	수행 방법
ICMP 재지정을 방지합니다.	네트워크 토폴로지의 표시기를 제거합니다.	22 페이지 "ICMP 재지정을 방지하는 방법"
네트워크 매개변수를 해당 보안 기본값으로 반환합니다.	관리 작업으로 줄어든 보안을 늘립니다.	23 페이지 "네트워크 매개변수를 보안 값으로 재설정하는 방법"

▼ 네트워크 경로 지정 데몬을 사용 안함으로 설정하는 방법

이 절차에 따라 기본 라우터를 지정하여 설치한 후 네트워크 경로 지정을 방지합니다. 그렇지 않으면 경로 지정을 수동으로 구성한 후 이 절차를 수행하십시오.

주 – 여러 네트워크 구성 절차에서는 경로 지정 데몬을 사용 안함으로 설정해야 합니다. 따라서 대규모 구성 절차에서는 이 데몬이 사용 안함으로 설정되었을 수 있습니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 경로 지정 데몬이 실행 중인지 확인합니다.

```
# svcs -x svc:/network/routing/route:default
svc:/network/routing/route:default (in.routed network routing daemon)
 State: online since April 10, 2011 05:15:35 AM PDT
   See: in.routed(1M)
   See: /var/svc/log/network-routing-route:default.log
Impact: None.
```

서비스가 실행 중이 아니면 여기에서 중지할 수 있습니다.

2 경로 지정 데몬을 사용 안함으로 설정합니다.

```
# routeadm -d ipv4-forwarding -d ipv6-forwarding
# routeadm -d ipv4-routing -d ipv6-routing
# routeadm -u
```

3 경로 지정 데몬이 사용 안함으로 설정되어 있는지 확인합니다.

```
# svcs -x routing/route:default
svc:/network/routing/route:default (in.routed network routing daemon)
 State: disabled since April 11, 2011 10:10:10 AM PDT
Reason: Disabled by an administrator.
   See: http://support.oracle.com/msg/SMF-8000-05
   See: in.routed(1M)
Impact: This service is not running.
```

참조 routeadm(1M) 매뉴얼 페이지

▼ 브로드캐스트 패킷 전달을 사용 안함으로 설정하는 방법

기본적으로 Oracle Solaris는 브로드캐스트 패킷을 전달합니다. 사이트 보안 정책에 따라 브로드캐스트 범람 가능성을 줄여야 하는 경우 이 절차를 사용하여 기본값을 변경하십시오.

주 – `_forward_directed_broadcasts` 네트워크 등록 정보를 사용 안함으로 설정하면 브로드캐스트 핑이 사용 안함으로 설정됩니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 패킷에 대해 브로드캐스트 패킷 전달 등록 정보를 0으로 설정합니다.

```
# ipadm set-prop -p _forward_directed_broadcasts=0 ip
```

2 현재 값을 확인합니다.

```
# ipadm show-prop -p _forward_directed_broadcasts ip
PROTO  PROPERTY                   PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ip     _forward_directed_broadcasts  rw   0         --           0         0,1
```

참조 `ipadm(1M)` 매뉴얼 페이지

▼ 에코 요청에 대한 응답을 사용 안함으로 설정하는 방법

이 절차를 사용하여 네트워크 토폴로지에 대한 정보 배포를 방지합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 브로드캐스트 에코 요청 등록 정보에 대한 응답을 IP 패킷에 대해 0으로 설정하고 현재 값을 확인합니다.

```
# ipadm set-prop -p _respond_to_echo_broadcast=0 ip

# ipadm show-prop -p _respond_to_echo_broadcast ip
PROTO  PROPERTY                 PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ip     _respond_to_echo_broadcast rw   0         --           1         0,1
```

2 멀티캐스트 에코 요청 등록 정보에 대한 응답을 IP 패킷에 대해 0으로 설정하고 현재 값을 확인합니다.

```
# ipadm set-prop -p _respond_to_echo_multicast=0 ipv4
# ipadm set-prop -p _respond_to_echo_multicast=0 ipv6

# ipadm show-prop -p _respond_to_echo_multicast ipv4
PROTO  PROPERTY                 PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4   _respond_to_echo_multicast rw   0         --           1         0,1
# ipadm show-prop -p _respond_to_echo_multicast ipv6
PROTO  PROPERTY                 PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv6   _respond_to_echo_multicast rw   0         --           1         0,1
```

참조 자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "_respond_to_echo_broadcast 및 _respond_to_echo_multicast (ipv4 or ipv6)" 및 `ipadm`(1M) 매뉴얼 페이지를 참조하십시오.

▼ 엄격한 다중 홈 지정을 설정하는 방법

다른 시스템에 대한 게이트웨이인 시스템(예: 방화벽 또는 VPN 노드)의 경우 이 절차를 사용하여 엄격한 다중 홈 지정을 설정합니다. `hostmodel` 등록 정보는 다중 홈 지정 시스템에 대한 IP 패킷의 전송 및 수신 동작을 제어합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 `hostmodel` 등록 정보를 IP 패킷에 대해 `strong`으로 설정합니다.

```
# ipadm set-prop -p hostmodel=strong ipv4
# ipadm set-prop -p hostmodel=strong ipv6
```

2 현재 값을 확인하고 가능한 값을 표시합니다.

```
# ipadm show-prop -p hostmodel ip
PROTO  PROPERTY    PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv6   hostmodel   rw   strong    strong       weak      strong,src-priority,weak
ipv4   hostmodel   rw   strong    strong       weak      strong,src-priority,weak
```

참조 자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "hostmodel (ipv4 or ipv6)" 및 `ipadm`(1M) 매뉴얼 페이지를 참조하십시오.

엄격한 다중 홈 지정 사용에 대한 자세한 내용은 터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법을 참조하십시오.

▼ 완전하지 않은 TCP 연결의 최대 개수를 설정하는 방법

이 절차에 따라 완전하지 않은 보류 중인 연결 개수를 제어하여 서비스 거부(DOS) 공격을 방지합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1. **수신 중인 연결의 최대 개수를 설정합니다.**

```
# ipadm set-prop -p _conn_req_max_q0=4096 tcp
```

2. **현재 값을 확인합니다.**

```
# ipadm show-prop -p _conn_req_max_q0 tcp
PROTO  PROPERTY         PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
tcp    _conn_req_max_q0 rw   4096      --           128       1-4294967295
```

참조 자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "_conn_req_max_q0" 및 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 보류 중인 TCP 연결의 최대 개수를 설정하는 방법

이 절차에 따라 허용된 수신 중인 연결 개수를 제어하여 DOS 공격을 방지합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1. **수신 중인 연결의 최대 개수를 설정합니다.**

```
# ipadm set-prop -p _conn_req_max_q=1024 tcp
```

2. **현재 값을 확인합니다.**

```
# ipadm show-prop -p _conn_req_max_q tcp
PROTO  PROPERTY         PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
tcp    _conn_req_max_q  rw   1024      --           128       1-4294967295
```

참조 자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "_conn_req_max_q" 및 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 초기 TCP 연결에 대한 높은 수준의 난수를 지정하는 방법

이 절차에서는 RFC 6528 (http://www.ietf.org/rfc/rfc6528.txt)을 준수하는 TCP 초기 시퀀스 번호 생성 매개변수를 설정합니다.

시작하기 전에 solaris.admin.edit/etc.default/inetinit 권한 부여가 지정된 관리자여야 합니다. 기본적으로 root 역할에 이 권한 부여가 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 TCP_STRONG_ISS 변수에 대한 기본값을 변경합니다.

```
# pfedit /etc/default/inetinit
# TCP_STRONG_ISS=1
TCP_STRONG_ISS=2
```

2 시스템을 재부트합니다.

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

▼ ICMP 재지정을 방지하는 방법

라우터는 ICMP 재지정 메시지를 사용하여 대상에 더 직접적인 경로를 호스트에 알립니다. 불법적인 ICMP 재지정 메시지는 중간 전달자의 공격을 초래할 수 있습니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 패킷에 대해 재지정 무시 등록 정보를 1로 설정한 후 현재 값을 확인하십시오.

ICMP 재지정 메시지는 호스트의 경로 테이블을 수정하며 인증되지 않습니다. 또한 재지정된 패킷을 처리하려면 시스템의 CPU가 더 많이 필요합니다.

```
# ipadm set-prop -p _ignore_redirect=1 ipv4
# ipadm set-prop -p _ignore_redirect=1 ipv6
# ipadm show-prop -p _ignore_redirect ipv4
PROTO  PROPERTY         PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4   _ignore_redirect  rw   1         1            0         0,1
# ipadm show-prop -p _ignore_redirect ipv6
PROTO  PROPERTY         PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv6   _ignore_redirect  rw   1         1            0         0,1
```

2 ICMP 재지정 메시지를 보내지 않도록 방지합니다.

이러한 메시지에는 네트워크 토폴로지 부분을 노출시킬 수 있는 경로 테이블 정보가 포함되어 있습니다.

```
# ipadm set-prop -p _send_redirects=0 ipv4
# ipadm set-prop -p _send_redirects=0 ipv6
# ipadm show-prop -p _send_redirects ipv4
```

```
PROTO PROPERTY          PERM CURRENT  PERSISTENT   DEFAULT  POSSIBLE
ipv4  _send_redirects   rw   0        0            1        0,1

# ipadm show-prop -p _send_redirects ipv6
PROTO  PROPERTY        PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
ipv6  _send_redirects  rw   0         0            1        0,1
```

자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "_send_redirects (ipv4 or ipv6)" 및 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

▼ 네트워크 매개변수를 보안 값으로 재설정하는 방법

기본적으로 보안되는 여러 네트워크 매개변수는 조정 가능하며 기본값에서 변경되었을 수 있습니다. 사이트 조건에서 허용하는 경우 다음과 같은 튜닝 가능한 매개변수를 해당 기본값으로 반환합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 패킷에 대해 소스 패킷 전달 등록 정보를 0으로 설정한 후 현재 값을 확인하십시오.

기본값은 허위로 제공된 패킷으로부터의 DOS 공격을 방지합니다.

```
# ipadm set-prop -p _forward_src_routed=0 ipv4
# ipadm set-prop -p _forward_src_routed=0 ipv6
# ipadm show-prop -p _forward_src_routed ipv4
PROTO  PROPERTY             PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4  _forward_src_routed   rw   0         --           0         0,1
# ipadm show-prop -p _forward_src_routed ipv6
PROTO  PROPERTY             PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv6  _forward_src_routed   rw   0         --           0         0,1
```

자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "forwarding (ipv4 or ipv6)"을 참조하십시오.

2 IP 패킷에 대해 netmask 응답 등록 정보를 0으로 설정한 후 현재 값을 확인하십시오.

기본값은 네트워크 토폴로지 정보의 배포를 방지합니다.

```
# ipadm set-prop -p _respond_to_address_mask_broadcast=0 ip
# ipadm show-prop -p _respond_to_address_mask_broadcast ip
PROTO PROPERTY                          PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ip    _respond_to_address_mask_broadcast rw   0         --           0         0,1
```

3 IP 패킷에 대해 시간 기록 응답 등록 정보를 0으로 설정한 후 현재 값을 확인하십시오.

기본값은 시스템에서 추가 CPU 요구를 제거하고 네트워크 정보의 배포를 방지합니다.

```
# ipadm set-prop -p _respond_to_timestamp=0 ip
# ipadm show-prop -p _respond_to_timestamp ip
PROTO  PROPERTY                 PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ip    _respond_to_timestamp     rw   0         --           0         0,1
```

4 IP 패킷에 대해 브로드캐스트 시간 기록 응답 등록 정보를 0으로 설정한 후 현재 값을 확인하십시오.

기본값은 시스템에서 추가 CPU 요구를 제거하고 네트워크 정보의 배포를 방지합니다.

```
# ipadm set-prop -p _respond_to_timestamp_broadcast=0 ip
# ipadm show-prop -p _respond_to_timestamp_broadcast ip
PROTO  PROPERTY                        PERM CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ip     _respond_to_timestamp_broadcast rw   0         --           0         0,1
```

5 IP 소스 경로 지정을 방지합니다.

기본값은 패킷이 네트워크 보안 조치를 무시하지 못하도록 합니다. 소스 경로가 지정된 패킷의 경우 패킷 소스가 라우터에 구성된 경로와 다른 경로를 표시하도록 허용합니다.

주 – 진단을 위해 이 매개변수를 1로 설정할 수 있습니다. 진단이 완료되면 이 값을 0으로 되돌립니다.

```
# ipadm set-prop -p _rev_src_routes=0 tcp
# ipadm show-prop -p _rev_src_routes tcp
PROTO PROPERTY          PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp   _rev_src_routes   rw   0        --          0        0,1
```

자세한 내용은 **Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서**의 "_rev_src_routes"를 참조하십시오.

참조 ipadm(1M) 매뉴얼 페이지

◆ ◆ ◆ 3 장

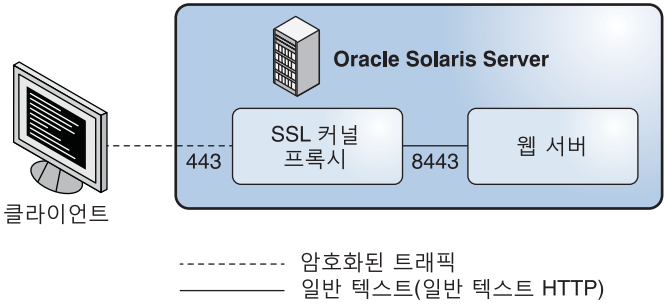
3 웹 서버 및 Secure Sockets Layer 프로토콜

이 장에서는 Oracle Solaris 시스템에서 SSL(Secure Sockets Layer) 프로토콜을 사용하여 웹 서버 통신을 암호화하고 속도를 향상시키는 방법에 대해 설명합니다.

SSL 커널 프록시로 웹 서버 통신 암호화

Oracle Solaris에서 실행되는 모든 웹 서버는 커널 레벨의 SSL 프로토콜 즉, SSL 커널 프록시를 사용하도록 구성할 수 있습니다. 각 웹 서버의 예로는 Apache 2.2 웹 서버 및 Oracle iPlanet 웹 서버가 있습니다. SSL 프로토콜에서는 기밀성, 메시지 무결성 및 두 응용 프로그램 간의 끝점 인증을 제공합니다. 웹 서버에서 SSL 커널 프록시가 실행되면 통신 속도가 향상됩니다. 다음 그림은 기본 구성을 보여 줍니다.

그림 3–1 커널로 암호화된 웹 서버 통신



SSL 커널 프록시는 SSL 프로토콜의 서버측을 구현합니다. 프록시는 여러 이점을 제공합니다.

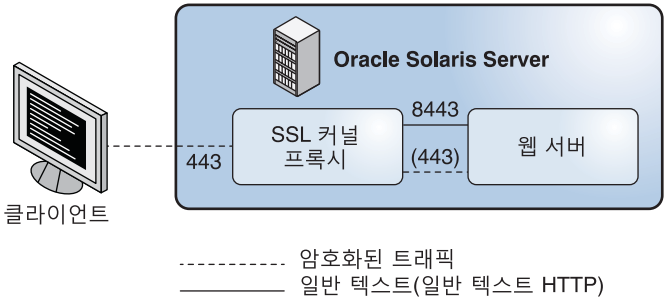
- 프록시는 웹 서버 같은 서버 응용 프로그램의 SSL 성능 속도를 향상시키므로 사용자 레벨의 SSL 라이브러리를 사용하는 응용 프로그램보다 더 나은 성능을 제공합니다. 성능 향상은 응용 프로그램의 작업 부하에 따라 35% 이상이 될 수 있습니다.
- SSL 커널 프록시는 투명합니다. 지정된 IP 주소가 없으므로 웹 서버에 실제 클라이언트 IP 주소와 TCP 포트가 표시됩니다.
- SSL 커널 프록시 및 웹 서버는 함께 작동하도록 설계되었습니다.

  그림 3–1은 SSL 커널 프록시를 사용하는 웹 서버가 있는 기본 시나리오를 보여 줍니다. SSL 커널 프록시는 443 포트에 구성되어 있는 반면 웹 서버는 8443에 구성되어 있으며, 이 웹 서버에서 암호화 해제된 HTTP 통신을 수신합니다.
- SSL 커널 프록시는 요청된 암호화를 지원하지 않는 경우 사용자 레벨의 암호로 폴백하도록 구성할 수 있습니다.

  그림 3–2는 더 복잡한 시나리오를 보여 줍니다. 웹 서버 및 SSL 커널 프록시가 사용자 레벨의 웹 서버 SSL로 폴백하도록 구성되어 있습니다.

  SSL 커널 프록시는 443 포트에 구성되어 있습니다. 웹 서버는 두 개 포트에 구성되어 있습니다. 8443 포트는 암호화 해제된 HTTP 통신을 수신하며 443 포트는 폴백 포트입니다. 폴백 포트는 SSL 커널 프록시에서 지원되지 않는 암호 슈트에 대한 암호화된 SSL 트래픽을 수신합니다.

그림 3–2 사용자 레벨의 폴백 옵션이 있는 커널로 암호화된 웹 서버 통신



SSL 커널 프록시는 가장 일반적인 암호 슈트 외에도 SSL 3.0 및 TLS 1.0 프로토콜을 지원합니다. `ksslcfg(1M)` 매뉴얼 페이지에서 전체 목록을 참조하십시오. 지원되지 않는 모든 암호 슈트에 대한 사용자 레벨 SSL 서버를 폴백하도록 프록시를 구성할 수 있습니다.

SSL 커널 프록시를 통한 웹 서버 보호(작업)

다음 절차에서는 SSL 커널 프록시를 사용하도록 웹 서버를 구성하는 방법을 보여 줍니다.

▼ SSL 커널 프록시를 사용하도록 Apache 2.2 웹 서버를 구성하는 방법

SSL 커널 프록시는 Apache 2.2 웹 서버에서 SSL 패킷 처리 속도를 높일 수 있습니다. 이 절차에서는 그림 3–1에서 보여 주는 간단한 시나리오를 구현합니다.

시작하기 전에 Apache 2.2 웹 서버를 구성했습니다. 이 웹 서버는 Oracle Solaris에 포함되어 있습니다.

`root` 역할이 있어야 합니다.

1 웹 서버를 중지합니다.

```
# svcadm disable svc:/network/http:apache22
```

2 서버 개인 키와 서버 인증서를 한 파일에 저장합니다.

`ssl.conf` 파일에서 `SSLCertificateFile` 매개변수만 지정한 경우 SSL 커널 프록시에 대해 지정한 파일을 직접 사용할 수 있습니다.

`SSLCertificateKeyFile` 매개변수도 지정한 경우 인증서 파일과 개인 키 파일을 결합해야 합니다. 다음과 유사한 명령을 실행하여 파일을 결합합니다.

```
# cat cert.pem key.pem > cert-and-key.pem
```

3 `ksslcfg` 명령과 함께 사용할 매개변수를 결정합니다.

`ksslcfg(1M)` 매뉴얼 페이지에서 전체 옵션 목록을 참조하십시오. 제공**해야** 하는 매개변수는 다음과 같습니다.

- *key-format* – 인증서 및 키 형식을 정의하기 위해 `-f` 옵션과 함께 사용합니다. SSL 커널 프록시의 경우 지원되는 형식은 `pkcs11`, `pem` 및 `pkcs12`입니다.
- *key-and-certificate-file* – `pem` 및 `pkcs12` *key-format* 옵션에 대한 서버 키 및 인증서를 저장하는 파일의 위치를 설정하기 위해 `-i` 옵션과 함께 사용합니다.
- *password-file* – `pem` 또는 `pkcs12` *key-format* 옵션에 대한 개인 키를 암호화하는 데 사용되는 암호를 가져오기 위해 `-p` 옵션과 함께 사용합니다. `pkcs11`의 경우 암호를 사용하여 PKCS #11 토큰에 인증합니다. `0400` 권한으로 암호 파일을 보호해야 합니다. 이 파일은 무인 재부트에 필요합니다.
- *token-label* – PKCS #11 토큰을 지정하기 위해 `-T` 옵션과 함께 사용합니다.

- *certificate-label* – PKCS #11 토큰에서 인증서 객체의 레이블을 선택하기 위해 `-C` 옵션과 함께 사용합니다.
- *proxy-port* – SSL 프록시 포트를 설정하기 위해 `-x` 옵션과 함께 사용합니다. 표준 포트 `80`과 다른 포트를 지정해야 합니다. 웹 서버는 암호화 해제된 일반 텍스트 트래픽을 위한 SSL 프록시 포트에서 수신 대기합니다. 일반적으로 값은 `8443`입니다.
- *ssl-port* – SSL 커널 프록시에 대한 수신 포트를 지정합니다. 일반적으로 값은 `443`입니다.

4 SSL 커널 프록시에 대한 서비스 인스턴스를 만듭니다.

다음 형식 중 하나를 사용하여 SSL 프록시 포트와 관련 매개변수를 지정합니다.

- **PEM 또는 PKCS #12를 키 형식으로 지정합니다.**

```
# ksslcfg create -f key-format -i key-and-certificate-file \
-p password-file -x proxy-port ssl-port
```

- **PKCS #11을 키 형식으로 지정합니다.**

```
# ksslcfg create -f pkcs11 -T PKCS#11-token -C certificate-label \
-p password-file -x proxy-port ssl-port
```

5 서비스 인스턴스가 온라인 상태인지 확인합니다.

```
# svcs svc:/network/ssl/proxy
STATE          STIME    FMRI
online         02:22:22 svc:/network/ssl/proxy:default
```

다음 출력은 감사 서비스 인스턴스가 만들어지지 않았음을 나타냅니다.

```
svcs: Pattern 'svc:/network/ssl/proxy' doesn't match any instances
STATE          STIME    FMRI
```

6 웹 서버를 SSL 프록시 포트에서 수신 대기하도록 구성합니다.

`/etc/apache2/2.2/http.conf` 파일을 편집하고 SSL 프록시 포트를 정의하도록 행을 추가합니다. 서버 IP 주소를 사용하는 경우 웹 서버는 해당 인터페이스에 대해서만 수신합니다. 행은 다음과 유사합니다.

```
Listen proxy-port
```

7 웹 서버에 대한 SMF 종속성을 설정합니다.

웹 서버 서비스는 SSL 커널 프록시 인스턴스가 시작된 후에만 시작할 수 있습니다. 다음 명령은 이러한 종속성을 설정합니다.

```
# svccfg -s svc:/network/http:apache22
svc:/network/http:apache22> addpg kssl dependency
...apache22> setprop kssl/entities = fmri:svc:/network/ssl/proxy:kssl-INADDR_ANY-443
...apache22> setprop kssl/grouping = astring: require_all
...apache22> setprop kssl/restart_on = astring: refresh
...apache22> setprop kssl/type = astring: service
...apache22> end
```

8 **웹 서버 서비스를 사용으로 설정합니다.**

```
# svcadm enable svc:/network/http:apache22
```

▼ SSL 커널 프록시를 사용하도록 Oracle iPlanet 웹 서버를 구성하는 방법

SSL 커널 프록시는 Oracle iPlanet 웹 서버에서 SSL 패킷 처리 속도를 높일 수 있습니다. 이 절차에서는 그림 3–1에서 보여 주는 간단한 시나리오를 구현합니다.

시작하기 전에 Oracle iPlanet 웹 서버를 설치 및 구성했습니다. 서버는 Oracle iPlanet Web Server (`http://www.oracle.com/technetwork/middleware/iplanetwebserver-098726.html?ssSourceSiteId=ocomen`)에서 다운로드할 수 있습니다. 지침은 Oracle iPLANET WEB SERVER 7.0.15 (`http://docs.oracle.com/cd/E18958_01/index.htm`)를 참조하십시오.

Network Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 **웹 서버를 중지합니다.**

관리자 웹 인터페이스를 사용하여 서버를 중지합니다. 지침은 Oracle iPLANET WEB SERVER 7.0.15 (`http://docs.oracle.com/cd/E18958_01/index.htm`)를 참조하십시오.

2 **`ksslcfg` 명령과 함께 사용할 매개변수를 결정합니다.**

`ksslcfg`(1M) 매뉴얼 페이지에서 전체 옵션 목록을 참조하십시오. 제공**해야** 하는 매개변수 목록은 단계 3 in 27 페이지 "SSL 커널 프록시를 사용하도록 Apache 2.2 웹 서버를 구성하는 방법"를 참조하십시오.

3 **SSL 커널 프록시에 대한 서비스 인스턴스를 만듭니다.**

다음 형식 중 하나를 사용하여 SSL 프록시 포트와 관련 매개변수를 지정합니다.

- **PEM 또는 PKCS #12를 키 형식으로 지정합니다.**

```
# ksslcfg create -f key-format -i key-and-certificate-file \
-p password-file -x proxy-port ssl-port
```

- **PKCS #11을 키 형식으로 지정합니다.**

```
# ksslcfg create -f pkcs11 -T PKCS#11-token -C certificate-label \
-p password-file -x proxy-port ssl-port
```

4 **인스턴스가 온라인 상태인지 확인합니다.**

```
# svcs svc:/network/ssl/proxy
STATE          STIME    FMRI
online         02:22:22 svc:/network/ssl/proxy:default
```

5 **웹 서버를 SSL 프록시 포트에서 수신 대기하도록 구성합니다.**

지침은 Oracle iPLANET WEB SERVER 7.0.15 (http://docs.oracle.com/cd/E18958_01/index.htm)를 참조하십시오.

6 **웹 서버에 대한 SMF 종속성을 설정합니다.**

웹 서버 서비스는 SSL 커널 프록시 인스턴스가 시작된 후에만 시작할 수 있습니다. 다음 명령은 웹 서버 서비스의 FMRI가 svc:/network/http:webserver7이라고 가정하여 이러한 종속성을 설정합니다.

```
# svccfg -s svc:/network/http:webserver7
svc:/network/http:webserver7> addpg kssl dependency
...webserver7> setprop kssl/entities = fmri:svc:/network/ssl/proxy:kssl-INADDR_ANY-443
...webserver7> setprop kssl/grouping = astring: require_all
...webserver7> setprop kssl/restart_on = astring: refresh
...webserver7> setprop kssl/type = astring: service
...webserver7> end
```

7 **웹 서버 서비스를 사용으로 설정합니다.**

```
# svcadm enable svc:/network/http:webserver7
```

▼ Apache 2.2 SSL로 폴백하도록 SSL 커널 프록시를 구성하는 방법

이 절차에서는 Apache 2.2 웹 서버를 처음부터 구성하고 SSL 커널 프록시를 기본 SSL 세션 처리 방식으로 구성합니다. 클라이언트가 제공하는 SSL 암호 세트에 SSL 커널 프록시에서 제공되는 암호가 없으면 Apache 2.2 웹 서버가 폴백 방식으로 사용됩니다. 이 절차에서는 그림 3–2에서 보여 주는 복잡한 시나리오를 구현합니다.

시작하기 전에 root 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 **Apache 2.2 웹 서버에서 서버의 SSL 커널 프록시에서 사용되는 키 인증서를 만듭니다.**

a. **CSR(인증서 서명 요청)을 생성합니다.**

다음 명령은 SSL 커널 프록시에 대한 CSR 및 관련 개인 키를 생성합니다.

```
# cd /root
# openssl req \
> -x509 -new \
> -subj "/C=CZ/ST=Prague region/L=Prague/CN=‘hostname‘" \
> -newkey rsa:2048 -keyout webkey.pem \
> -out webcert.pem
Generating a 2048 bit RSA private key
.+++
........+++
writing new private key to ’webkey.pem’
Enter PEM pass phrase: JohnnyCashIsCool
```

```
Verifying - Enter PEM pass phrase:  JohnnyCashIsCool
#
# chmod 440 /root/webcert.pem ; chown root:webservd /root/webcert.pem
```

자세한 내용은 openssl(5) 매뉴얼 페이지를 참조하십시오.

b. **CSR을 CA(인증 기관)에 보냅니다.**

c. **webcert.pem 파일을 CA에서 받은 서명된 인증서로 바꿉니다.**

2 **문장암호 및 공개/개인 키 인증서로 SSL 커널 프록시를 구성합니다.**

a. **문장암호를 만들고 저장하며 보호합니다.**

```
# echo "RefrigeratorsAreCool" > /root/kssl.pass
# chmod 440 /root/kssl.pass; chown root:webservd /root/kssl.pass
```

주 – 문장암호에는 공백을 포함할 수 없습니다.

---

b. **개인 키 및 공개 키 인증서를 한 파일로 결합합니다.**

```
# cat /root/webcert.pem /root/webkey.pem > /root/webcombo.pem
```

c. **공개/개인 키 인증서 및 문장암호로 SSL 커널 프록시를 구성합니다.**

```
# ksslcfg create -f pem -i /root/webcombo.pem -x 8443 -p /root/kssl.pass 443
```

3 **웹 서버가 8443 포트에서 일반 텍스트를 수신 대기하도록 구성합니다.**

/etc/apache2/2.2/httpd.conf 파일에서 Listen 행을 편집합니다.

```
# pfedit /etc/apache2/2.2/httpd.conf
...
## Listen 80
Listen 8443
```

4 **SSL 모듈 템플리트인 ssl.conf를 Apache 구성 디렉토리에 추가합니다.**

```
# cp /etc/apache2/2.2/samples-conf.d/ssl.conf /etc/apache2/2.2/ssl.conf
```

이 모듈은 암호화된 연결을 위해 443 수신 대기 포트를 추가합니다.

5 **웹 서버가 /root/kssl.pass 파일의 문장암호를 해독할 수 있도록 합니다.**

a. **kssl.pass 파일을 읽는 셸 스크립트를 만듭니다.**

```
# pfedit /root/put-passphrase.sh
#!/usr/bin/ksh -p
## Reads SSL kernel proxy passphrase
/usr/bin/cat /root/kssl.pass
```

b. **스크립트를 실행 가능하게 설정하고 파일을 보호합니다.**

```
# chmod 500 /root/put-passphrase.sh
# chown webservd:webservd /root/put-passphrase.sh
```

c. ssl.conf 파일의 SSLPassPhraseDialog 매개변수를 수정하여 이 셸 스크립트를 호출합니다.

```
# pfedit /etc/apache2/2.2/ssl.conf
...
## SSLPassPhraseDialog  builtin
SSLPassPhraseDialog exec:/root/put-passphrase.sh
```

6 웹 서버의 공개 및 개인 키 인증서를 올바른 위치에 저장합니다.

ssl.conf 파일의 SSLCertificateFile 및 SSLCertificateKeyFile 매개변수 값에 올바른 위치 및 이름이 포함됩니다. 올바른 위치로 인증서를 복사하거나 연결할 수 있습니다.

```
# ln -s  /root/webcert.pem /etc/apache2/2.2/server.crt      SSLCertificateFile default location
# ln -s /root/webkey.pem /etc/apache2/2.2/server.key        SSLCertificateKeyFile default location
```

7 Apache 서비스를 사용으로 설정합니다.

```
# svcadm enable apache22
```

8 (옵션) 두 개의 포트가 작동 중인지 확인합니다.

openssl s_client 및 kstat 명령을 사용하여 패킷을 확인합니다.

a. SSL 커널 프록시에 사용할 수 있는 암호를 사용합니다.

```
# openssl s_client -cipher RC4-SHA -connect web-server:443
```

kstat 카운터 kssl_full_handshakes에 1이 증가하면 SSL 커널 프록시에서 SSL 세션이 처리되었는지 확인합니다.

```
# kstat -m kssl -s kssl_full_handshakes
```

b. SSL 커널 프록시에 사용할 수 없는 암호를 사용합니다.

```
# openssl s_client -cipher CAMELLIA256-SHA -connect web-server:443
```

kstat 카운터 kssl_fallback_connections에 1이 증가하면 패킷이 도달했지만 SSL 세션이 Apache 웹 서버에서 처리되었는지 확인합니다.

```
# kstat -m kssl -s kssl_fallback_connections
```

예 3–1 SSL 커널 프록시를 사용하도록 Apache 2.2 웹 서버 구성

다음 명령은 pem 키 형식을 사용하는 SSL 커널 프록시 서비스 인스턴스를 만듭니다.

```
# ksslcfg create -f pem -i cert-and-key.pem -p kssl.pass -x 8443 443
```

▼ 영역에서 SSL 커널 프록시를 사용하는 방법

SSL 커널 프록시는 다음 제한 사항과 함께 영역에서 작동합니다.

- 모든 커널 SSL 관리는 전역 영역에서 수행해야 합니다. 전역 영역 관리자는 로컬 영역 인증서 및 키 파일에 액세스해야 합니다. 로컬 영역 웹 서버는 전역 영역에서 `ksslcfg` 명령을 사용하여 서비스 인스턴스를 구성한 후 시작할 수 있습니다.
- 인스턴스를 구성하는 경우 `ksslcfg` 명령과 함께 특정 호스트 이름이나 IP 주소를 지정해야 합니다. 특히 인스턴스에서 IP 주소로 `INADDR_ANY`를 지정할 수 없습니다.

시작하기 전에 웹 서버 서비스는 비전역 영역에서 구성되고 사용으로 설정됩니다.

Network Security 및 Zone Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 비전역 영역에서 먼저 웹 서버를 중지합니다.

예를 들어 `apache-zone` 영역에서 Apache 웹 서버를 중지하려면 다음 명령을 실행합니다.

```
apache-zone # svcadm disable svc:/network/http:apache22
```

2 전역 영역에서 영역의 SSL 커널 프록시에 대한 서비스 인스턴스를 만듭니다.

`apache-zone`에 대한 서비스 인스턴스를 만들려면 다음과 유사한 명령을 사용합니다.

```
# ksslcfg create -f pem -i /zone/apache-zone/root/keypair.pem \
-p /zone/apache-zone/root/skppass -x 8443 apache-zone 443
```

3 비전역 영역에서 웹 서비스 인스턴스를 사용으로 설정합니다.

예를 들어 `apache-zone`에서 웹 서비스를 사용으로 설정합니다.

```
apache-zone # svcadm enable svc:/network/http:apache22
```

◆ ◆ ◆ 4 장

4 Oracle Solaris의 IP 필터(개요)

이 장에서는 Oracle Solaris 기능인 IP 필터의 개요를 제공합니다. IP 필터 작업은 5 장, “IP 필터(작업)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

IP 필터 소개

Oracle Solaris의 IP 필터 기능은 Stateful 패킷 필터링 및 NAT(Network Address Translation)를 제공하는 방화벽입니다. 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 라이센스 약관, 직권 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.

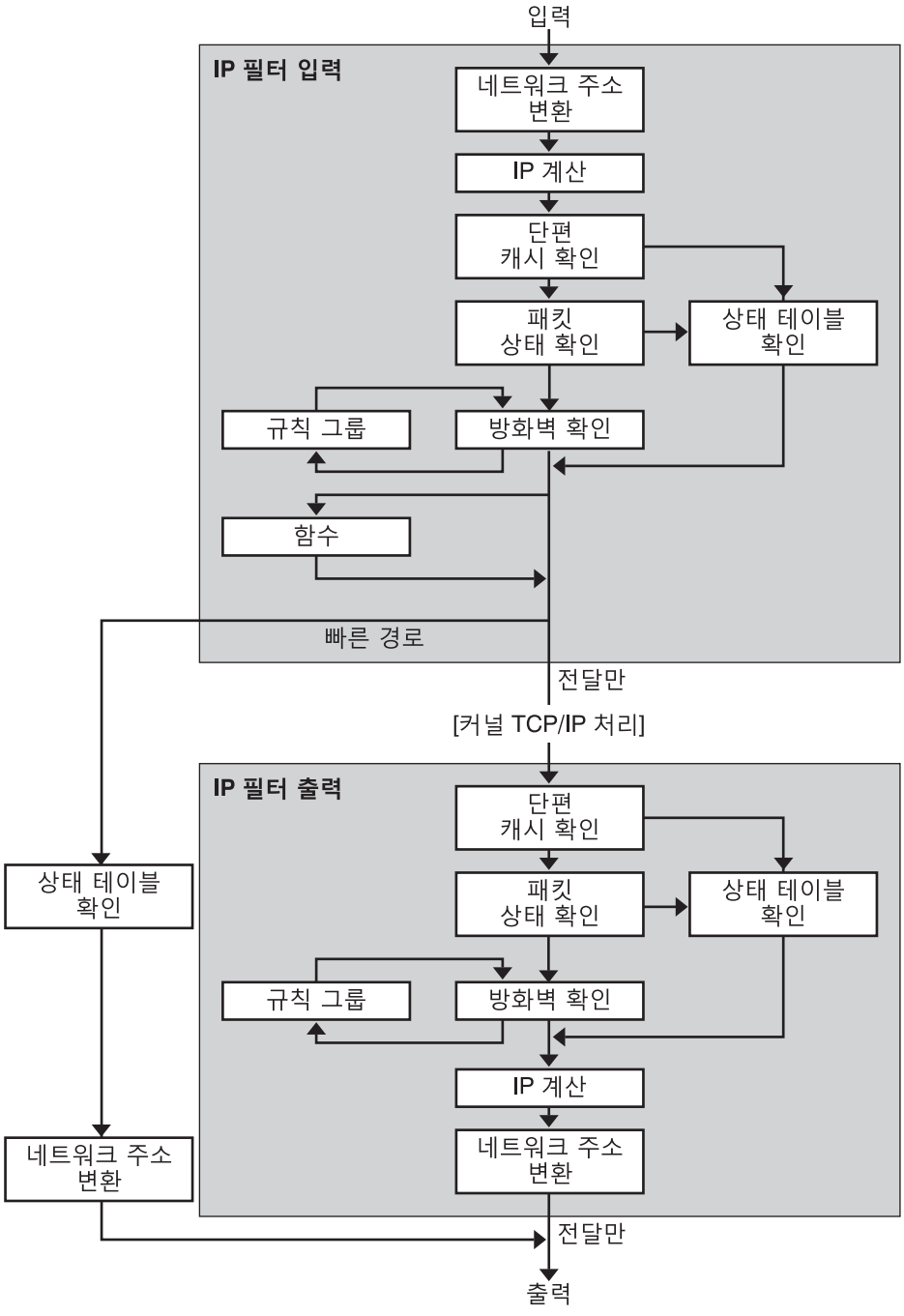
오픈 소스 IP 필터에 대한 정보 소스

Darren Reed의 오픈 소스 IP 필터 소프트웨어 홈 페이지는 http://coombs.anu.edu.au/~avalon/ip-filter.html에서 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 Oracle Solaris에서의 IP 필터 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 4–1 패킷 처리 순서


입력
IP 필터 입력
네트워크 주소 변환
IP 계산
단편 캐시 확인
패킷 상태 확인
상태 테이블 확인
규칙 그룹
방화벽 확인
함수
빠른 경로
전달만
[커널 TCP/IP 처리]
IP 필터 출력
단편 캐시 확인
패킷 상태 확인
상태 테이블 확인
규칙 그룹
방화벽 확인
IP 계산
네트워크 주소 변환
상태 테이블 확인
네트워크 주소 변환
전달만
출력

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**

  기본적으로 단편화된 패킷은 캐시됩니다. 특정 패킷에 대한 단편이 모두 도착한 경우 필터링 규칙이 적용되며 단편이 허용되거나 차단됩니다. 규칙 파일에 `set defrag off`가 나타나면 단편이 캐시되지 않습니다.

- **패킷 상태 확인**

  `keep state`가 규칙에 포함된 경우 규칙이 `pass`를 의미하는지 아니면 `block`을 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 response`가 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**

  이중 처리를 방지하기 위해 인증된 패킷은 방화벽 루프를 통해 한 번만 전달됩니다.

IP 필터 사용 지침

- IP 필터는 SMF 서비스 `svc:/network/ipfilter`를 통해 관리됩니다. SMF에 대한 전체 개요는 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"를 참조하십시오. SMF와 관련된 단계별 절차에 대한 자세한 내용은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 2 장, "서비스 관리(작업)"를 참조하십시오.
- IP 필터를 사용하려면 구성 파일을 직접 편집해야 합니다.

- IP 필터는 Oracle Solaris의 일부로 설치됩니다. 시스템이 자동 네트워킹을 사용하도록 구성된 경우 기본적으로 IP 필터 서비스가 사용으로 설정됩니다. `nwam(5)` 및 `netadm(1M)` 매뉴얼 페이지에 설명된 대로 자동 네트워크 프로파일이 이 방화벽을 사용으로 설정합니다. 자동으로 네트워크에 연결되는 시스템의 사용자 정의 구성에서는 IP 필터 서비스가 사용으로 설정되지 않습니다. 서비스를 사용으로 설정하는 것과 관련된 작업은 47 페이지 "IP 필터 구성"을 참조하십시오.
- IP 필터를 관리하려면 `root` 역할이 있거나 IP Filter Management 권한 프로파일이 포함된 역할이 있어야 합니다. 만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 11.1 관리: 보안 서비스**의 "RBAC 초기 구성(작업 맵)"을 참조하십시오.
- Oracle Solaris Cluster 소프트웨어의 경우 확장 가능한 서비스에 대해서는 IP 필터를 통한 필터링을 지원하지 않지만 페일오버 서비스에 대해서는 IP 필터를 지원합니다. 클러스터에서 IP 필터를 구성하는 경우의 지침 및 제한 사항은 **Oracle Solaris Cluster Software Installation Guide**에서 "Oracle Solaris OS Feature Restrictions"를 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. 방화벽 및 NAT에 대한 규칙은 기본적으로 제공되지 않습니다. 사용자 정의 구성 파일을 만들고 이러한 파일의 경로 이름을 IP 필터 서비스 등록 정보 값으로 설정해야 합니다. 서비스가 사용으로 설정된 후 시스템이 재부트되면 이러한 파일이 자동으로 로드됩니다. 샘플 구성 파일은 70 페이지 "IP 필터 구성 파일 예"를 참조하십시오. 자세한 내용은 `svc.ipfd(1M)` 매뉴얼 페이지를 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. 구성 파일을 로드하려면 파일을 만든 다음 IP 필터 서비스에 해당 경로 이름을 제공해야 합니다.

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.

`block`	패킷이 필터를 통과하지 못하도록 합니다.
`pass`	패킷이 필터를 통과할 수 있도록 합니다.
`log`	패킷을 기록하되 패킷 차단 또는 통과를 결정하지 않습니다. `ipmon` 명령을 사용하여 로그를 확인할 수 있습니다.
`count`	필터 통계에 패킷을 포함합니다. `ipfstat` 명령을 사용하여 통계를 확인할 수 있습니다.
`skip` *number*	필터가 *number*개의 필터링 규칙을 건너 뛸 수 있도록 합니다.

`auth` 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.

`log`	규칙이 마지막 일치 규칙인 경우 패킷을 기록합니다. `ipmon` 명령을 사용하여 로그를 확인할 수 있습니다.
`quick`	패킷 일치가 있을 경우 `quick` 옵션이 포함된 규칙을 실행합니다. 모든 후속 규칙 확인이 중지됩니다.
`on` *interface-name*	패킷이 지정된 인터페이스 내부 또는 외부로 이동되고 있는 경우에만 규칙을 적용합니다.
`dup-to` *interface-name*	패킷을 복사하고 *interface-name*의 중복 출력을 선택적으로 지정된 IP 주소로 보냅니다.
`to` *interface-name*	패킷을 *interface-name*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.

`tos`	16진수 또는 십진수 정수로 표시되는 type-of-service 값을 기준으로 패킷을 필터링합니다.
`ttl`	time-to-live 값을 기준으로 패킷을 일치시킵니다. 패킷에 저장된 time-to-live 값은 패킷을 폐기하기 전에 네트워크에 보관할 수 있는 기간을 나타냅니다.
`proto`	특정 프로토콜을 일치시킵니다. `/etc/protocols` 파일에 지정된 프로토콜 이름을 사용할 수도 있고, 십진수를 사용하여 프로토콜을 나타낼 수도 있습니다. `tcp/udp` 키워드를 사용하여 TCP 또는 UDP 패킷을 일치시킬 수 있습니다.
`from/to/all/ any`	소스 IP 주소, 대상 IP 주소, 포트 번호 중 일부 또는 전체와 일치시킵니다. `all` 키워드는 모든 소스에서 수신되고 모든 대상으로 송신되는 패킷을 승인할 수 있습니다.
`with`	패킷과 연관되어 있는 지정된 속성을 일치시킵니다. 옵션이 없는 경우에만 패킷을 일치시키려면 키워드 앞에 `not` 또는 `no` 단어를 삽입하십시오.

`flags`	설정된 TCP 플래그를 기준으로 필터링할 TCP에 사용됩니다. TCP 플래그에 대한 자세한 내용은 `ipf(4)` 매뉴얼 페이지를 참조하십시오.
`icmp-type`	ICMP 유형에 따라 필터링합니다. 이 키워드는 `proto` 옵션이 `icmp`로 설정된 경우에만 사용되며 `flags` 옵션이 설정된 경우 사용되지 않습니다.
`keep` *keep-options*	패킷에 대해 보관되는 정보를 결정합니다. 사용 가능한 *keep-options*로는 `state` 옵션이 있습니다. `state` 옵션은 세션에 대한 정보를 보관하며 TCP, UDP 및 ICMP 패킷에 대해 보관될 수 있습니다.
`head` *number*	*number* 번호로 표시되는 필터링 규칙에 대한 새 그룹을 만듭니다.
`group` *number*	기본 그룹 대신 그룹 번호 *number*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 53 페이지 “IP 필터에 대한 패킷 필터링 규칙 세트 관리”를 참조하십시오. 예에 표시된 IP 주소 체계(`192.168.0.0/16`)에 대한 설명은 **Oracle Solaris 11.1 네트워크 구성 및 관리**의 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.

명령줄에서 `ipnat` 명령을 사용하거나 NAT 구성 파일에서 NAT 규칙을 만들 수 있습니다. NAT 구성 파일을 만들고 해당 경로 이름을 서비스의 `config/ipnat_config_file` 등록 정보 값으로 설정해야 합니다. 기본값은 `/etc/ipf/ipnat.conf`입니다. 자세한 내용은 `ipnat(1M)` 명령을 참조하십시오.

NAT 규칙은 IPv4 및 IPv6 주소 모두에 적용할 수 있습니다. 하지만 두 유형의 주소를 단일 규칙으로 지정할 수 없습니다. 대신 각 주소 유형에 대해 별도의 규칙을 설정해야 합니다. IPv6 주소가 포함된 NAT 규칙에서는 `mapproxy` 및 `rdrproxy` NAT 명령을 동시에 사용할 수 없습니다.

NAT 규칙 구성

다음 구문을 사용하여 NAT 규칙을 만들 수 있습니다.

command interface-name parameters

1. 각 규칙은 다음 명령 중 하나로 시작합니다.

   `map` 제한되지 않은 라운드 로빈 프로세스에서 특정 IP 주소 또는 네트워크를 다른 IP 주소 또는 네트워크에 매핑합니다.

   `rdr` 특정 IP 주소와 포트 쌍의 패킷을 다른 IP 주소와 포트 쌍으로 재지정합니다.

   `bimap` 외부 IP 주소와 내부 IP 주소 간에 양방향 NAT를 설정합니다.

   `map-block` 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.

   `ipmask` 네트워크 마스크를 지정합니다.

   `dstipmask` `ipmask`가 변환되는 주소를 지정합니다.

   `mapport` 포트 번호 범위와 함께 `tcp`, `udp` 또는 `tcp/udp` 프로토콜을 지정합니다.

다음 예에서는 NAT 규칙을 구성하는 방법을 보여 줍니다. 소스 주소가 `192.168.1.0/24`인 `net2` 장치에서 송신되는 패킷을 재작성하고 외부적으로 소스 주소를 `10.1.0.0/16`으로 표시하려면 NAT 규칙 세트에 다음 규칙을 포함합니다.

```
map net2 192.168.1.0/24 -> 10.1.0.0/16
```

IPv6 주소에는 다음 규칙이 적용됩니다.

```
map net3 fec0:1::/64 -> 2000:1:2::/72 portmap tcp/udp 1025:65000
map-block net3 fe80:0:0:209::/64 -> 209:1:2::/72 ports auto
rdr net0 209::ffff:fe13:e43e port 80 -> fec0:1::e,fec0:1::f port 80 tcp round-robin
```

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 IP 필터 서비스에서 로드되는 파일에 상주할 수 있습니다. 파일을 만들고 해당 경로 이름을 서비스의 `config/ippool_config_file` 등록 정보 값으로 설정해야 합니다. 기본값은 `/etc/ipf/ippool.conf`입니다.

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 저장소 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 `13`으로 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 `13`을 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 다시 로드해야 합니다.

전체 문법 및 구문은 `ippool(4)` 매뉴얼 페이지를 참조하십시오.

IP 필터용 IPv6

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 및 IPv6 주소에 대해 개별 풀을 만들거나 IPv4 및 IPv6 주소가 모두 포함된 풀을 만들 수 있습니다.

다시 사용 가능한 IPv6 패킷 필터링 규칙을 만들려면 특정 IPv6 파일을 만들어야 합니다. 그런 다음 해당 경로 이름을 IP 필터 서비스의 `config/ip6_config_file` 등록 정보 값으로 설정합니다. 기본값은 `/etc/ipf/ip6.conf`입니다.

IPv6에 대한 자세한 내용은 **System Administration Guide: IP Services**의 3 장, "Introducing IPv6 (Overview)"를 참조하십시오. IP 필터와 관련된 작업은 5 장, "IP 필터(작업)"를 참조하십시오.

IP 필터 매뉴얼 페이지

다음 표에서는 IP 필터와 관련된 매뉴얼 페이지에 대해 설명합니다.

매뉴얼 페이지	설명
`ipf(1M)`	IP 필터 규칙을 관리하고 조정 가능 매개변수를 표시하며 기타 작업을 수행합니다.

매뉴얼 페이지	설명
ipf(4)	IP 필터 패킷 필터링 규칙 생성 문법 및 구문을 포함합니다.
ipfilter(5)	IP 필터 소프트웨어를 설명합니다.
ipfs(1M)	재부트 시 NAT 정보 및 상태 테이블 정보를 저장하고 복원합니다.
ipfstat(1M)	패킷 처리에 관한 통계를 검색하고 표시합니다.
ipmon(1M)	로그 장치를 열고 패킷 필터링 및 NAT에 대해 기록된 패킷을 확인합니다.
ipnat(1M)	NAT 규칙을 관리하고 NAT 통계를 표시합니다.
ipnat(4)	NAT 규칙 생성 문법 및 구문을 포함합니다.
ippool(1M)	주소 풀을 만들고 관리합니다.
ippool(4)	IP 필터 주소 풀 생성 문법 및 구문을 포함합니다.
svc.ipfd(1M)	IP 필터 서비스 구성에 관한 정보를 제공합니다.

◆ ◆ ◆ 5 장

5 IP 필터(작업)

이 장에서는 단계별 작업 지침을 제공합니다. IP 필터에 대한 개요 정보는 4 장, “Oracle Solaris의 IP 필터(개요)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

IP 필터 구성

다음 작업 맵에서는 IP 필터 규칙을 만들고 서비스를 사용으로 설정 및 사용 안함으로 설정하는 것과 관련된 절차를 식별합니다.

표 5–1 IP 필터 구성(작업 맵)

작업	수행 방법
IP 필터에서 사용되는 파일과 서비스 상태를 확인합니다.	48 페이지 “IP 필터 서비스 기본값을 표시하는 방법”
네트워크 트래픽, NAT를 통한 패킷 및 주소 풀에 대한 패킷 필터링 규칙 세트를 사용자 정의합니다.	49 페이지 “IP 필터 구성 파일을 만드는 방법”
IP 필터 서비스를 사용 또는 사용 안함으로 설정하거나 새로 고칩니다.	50 페이지 “IP 필터를 사용으로 설정하고 새로 고치는 방법”
단편에 도달하는 패킷의 기본 설정을 수정합니다.	50 페이지 “패킷 재어셈블을 사용 안함으로 설정하는 방법”
시스템에서 영역 사이의 트래픽을 필터링합니다.	51 페이지 “루프백 필터링을 사용으로 설정하는 방법”
IP 필터 사용을 중지합니다.	52 페이지 “패킷 필터링을 사용 안함으로 설정하는 방법”

▼ IP 필터 서비스 기본값을 표시하는 방법

시작하기 전에 `ipfstat` 명령을 실행하려면 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 필터 서비스에 대한 구성 파일 이름과 위치를 확인합니다.

```
% svccfg -s ipfilter:default listprop   | grep file
config/ipf6_config_file                      astring     /etc/ipf/ipf6.conf
config/ipnat_config_file                     astring     /etc/ipf/ipnat.conf
config/ippool_config_file                    astring     /etc/ipf/ippool.conf
firewall_config_default/custom_policy_file   astring     none
```

처음 세 개의 파일 등록 정보는 파일 위치를 나타냅니다. 이러한 파일을 만든 경우에만 해당 파일이 존재합니다. 해당 파일의 등록 정보 값을 변경하면 구성 파일의 위치를 변경할 수 있습니다. 절차는 49 페이지 "IP 필터 구성 파일을 만드는 방법"을 참조하십시오.

사용자 고유의 패킷 필터링 규칙을 사용자 정의하는 경우 네번째 파일 등록 정보를 수정합니다. 49 페이지 "IP 필터 구성 파일을 만드는 방법"의 단계 1 및 단계 2를 참조하십시오.

2 IP 필터 서비스가 사용으로 설정되었는지 여부를 결정합니다.

- 수동으로 네트워크에 연결된 시스템에서는 기본적으로 IP 필터가 사용으로 설정되어 있지 않습니다.

  ```
  % svcs -x ipfilter:default
  svc:/network/ipfilter:default (IP Filter)
   State: disabled since Mon Sep 10 10:10:50 2012
  Reason: Disabled by an administrator.
     See: http://oracle.com/msg/SMF-8000-05
     See: ipfilter(5)
  Impact: This service is not running.
  ```

- IPv4 네트워크에서 자동으로 네트워크에 연결된 시스템의 경우 다음 명령을 실행하여 IP 필터 정책을 확인합니다.

  ```
  $ ipfstat -io
  ```

  정책을 만든 파일을 확인하려면 `/etc/nwam/loc/NoNet/ipf.conf`를 검토합니다. 이 파일은 보기 전용입니다. 정책을 수정하려면 49 페이지 "IP 필터 구성 파일을 만드는 방법"을 참조하십시오.

주 – IPv6 네트워크에서 IP 필터 정책을 확인하려면 `ipfstat -6io`에서와 같이 `-6` 옵션을 추가합니다. 자세한 내용은 `ipfstat(1M)` 매뉴얼 페이지를 참조하십시오.

▼ IP 필터 구성 파일을 만드는 방법

자동으로 구성된 네트워크 구성의 IP 필터 정책을 수정하거나 수동으로 구성된 네트워크에서 IP 필터를 사용하려면 구성 파일을 만들고 서비스에 이러한 파일을 알린 다음 서비스를 사용으로 설정합니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 필터 서비스에 대한 정책 파일의 파일 위치를 지정합니다.

이 파일에는 패킷 필터링 규칙 세트가 포함되어 있습니다.

a. 우선 정책 파일을 custom으로 설정합니다.

```
$ svccfg -s ipfilter:default setprop firewall_config_default/policy = astring: "custom"
```

b. 그런 다음 위치를 지정합니다.

예를 들어 패킷 필터링 규칙 세트의 위치를 /etc/ipf/myorg.ipf.conf로 지정합니다.

```
$ svccfg -s ipfilter:default \
setprop firewall_config_default/custom_policy_file = astring: "/etc/ipf/myorg.ipf.conf"
```

2 패킷 필터링 규칙 세트를 만듭니다.

패킷 필터링에 대한 자세한 내용은 40 페이지 "IP 필터의 패킷 필터링 기능 사용"을 참조하십시오. 구성 파일의 예는 70 페이지 "IP 필터 구성 파일 예" 및 /etc/nwam/loc/NoNet/ipf.conf 파일을 참조하십시오.

주 – 지정한 정책 파일이 비어 있으면 필터링이 수행되지 않습니다. 비어 있는 패킷 필터링 파일은 다음과 같은 규칙 세트가 있는 것과 같습니다.

```
pass in all
pass out all
```

---

3 (옵션) IP 필터에 대한 NAT(Network Address Translation) 구성 파일을 만듭니다.

NAT를 통해 패킷을 필터링하려면 적절한 이름을 사용하여 NAT규칙 파일을 만듭니다(예: /etc/ipf/ipnat.conf). 이 이름을 변경하려면 config/ipnat_config_file 서비스 등록 정보 값을 다음과 같이 변경합니다.

```
$ svccfg -s ipfilter:default \
setprop config/ipnat_config_file = astring: "/etc/ipf/myorg.ipnat.conf"
```

NAT에 대한 자세한 내용은 42 페이지 "IP 필터의 NAT 기능 사용"을 참조하십시오.

4 **(옵션) 주소 풀 구성 파일을 만듭니다.**

주소 그룹을 단일 주소 풀로 나타내려면 적절한 이름을 사용하여 풀 파일을 만듭니다(예: `/etc/ipf/ippool.conf`). 이 이름을 변경하려면 `config/ippool_config_file` 서비스 등록 정보 값을 다음과 같이 변경합니다.

```
$ svccfg -s ipfilter:default \
setprop config/ippool_config_file = astring: "/etc/ipf/myorg.ippool.conf"
```

주소 풀에는 IPv4 및 IPv6 주소의 조합이 포함될 수 있습니다. 주소 풀에 대한 자세한 내용은 44 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. 51 페이지 "루프백 필터링을 사용으로 설정하는 방법"을 참조하십시오. 영역에 적용할 규칙 세트도 정의해야 합니다.

6 **(옵션) 단편화된 패킷의 재어셈블을 사용 안함으로 설정합니다.**

기본적으로 단편은 IP 필터에서 재어셈블됩니다. 기본값을 수정하려면 50 페이지 "패킷 재어셈블을 사용 안함으로 설정하는 방법"을 참조하십시오.

▼ IP 필터를 사용으로 설정하고 새로 고치는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

49 페이지 "IP 필터 구성 파일을 만드는 방법"을 완료했습니다.

1 **IP 필터를 사용으로 설정합니다.**

초기에 IP 필터를 사용으로 설정하려면 다음 명령을 입력하십시오.

```
$ svcadm enable network/ipfilter
```

2 **서비스가 실행 중인 경우 IP 필터 구성 파일을 수정한 후 서비스를 새로 고칩니다.**

```
$ svcadm refresh network/ipfilter
```

주 – `refresh` 명령은 간단하게 방화벽을 사용 안함으로 설정합니다. 방화벽을 유지하려면 규칙을 추가하거나 새 구성 파일을 추가합니다. 예와 함께 절차를 보려면 53 페이지 "IP 필터 규칙 세트 작업"을 참조하십시오.

▼ 패킷 재어셈블을 사용 안함으로 설정하는 방법

기본적으로 단편은 IP 필터에서 재어셈블됩니다. 이 재어셈블을 사용 안함으로 설정하려면 정책 파일의 시작 부분에 규칙을 삽입합니다.

시작하기 전에 IP Filter Management 권한 프로파일 및 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 권한 부여가 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 필터를 사용 안함으로 설정합니다.

```
$ svcadm disable network/ipfilter
```

2 IP 필터 정책 파일의 시작 부분에 다음 규칙을 추가합니다.

```
set defrag off;
```

다음과 같이 `pfedit` 명령을 사용합니다.

```
$ pfedit /etc/ipf/myorg.ipf.conf
```

이 규칙은 파일에서 정의된 모든 `block` 및 `pass` 규칙 앞에 와야 합니다. 단, 다음 예와 유사하게 행 앞에 주석을 삽입할 수 있습니다.

```
# Disable fragment reassembly
#
set defrag off;
# Define policy
#
block in all
block out all
other rules
```

3 IP 필터를 사용으로 설정합니다.

```
$ svcadm enable network/ipfilter
```

4 패킷이 재어셈블되고 있지 않은지 확인합니다.

```
$ ipf -T defrag
defrag  min 0   max 0x1 current 0
```

`current`가 `0`이면 단편이 재어셈블되지 않습니다. `current`가 `1`이면 단편이 재어셈블됩니다.

▼ 루프백 필터링을 사용으로 설정하는 방법

시작하기 전에 IP Filter Management 권한 프로파일 및 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 권한 부여가 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 IP 필터가 실행 중인 경우 중지합니다.

```
$ svcadm disable network/ipfilter
```

2 **IP 필터 정책 파일의 시작 부분에 다음 규칙을 추가합니다.**

```
set intercept_loopback true;
```

다음과 같이 `pfedit` 명령을 사용합니다.

```
$ pfedit /etc/ipf/myorg.ipf.conf
```

이 행은 파일에서 정의된 모든 `block` 및 `pass` 규칙 앞에 와야 합니다. 단, 다음 예와 유사하게 행 앞에 주석을 삽입할 수 있습니다.

```
...
#set defrag off;
#
# Enable loopback filtering to filter between zones
#
set intercept_loopback true;
#
# Define policy
#
block in all
block out all
other rules
```

3 **IP 필터를 사용으로 설정합니다.**

```
$ svcadm enable network/ipfilter
```

4 **루프백 필터링 상태를 확인하려면 다음 명령을 사용합니다.**

```
$ ipf -T ipf_loopback
ipf_loopback    min 0   max 0x1 current 1
$
```

`current`가 `0`이면 루프백 필터링이 사용 안함으로 설정됩니다. `current`가 `1`이면 루프백 필터링이 사용으로 설정됩니다.

▼ 패킷 필터링을 사용 안함으로 설정하는 방법

이 절차에서는 커널에서 규칙을 모두 제거하고 서비스를 사용 안함으로 설정합니다. 이 절차를 사용하는 경우 패킷 필터링 및 NAT를 다시 시작하려면 적절한 구성 파일과 함께 IP 필터를 사용으로 설정해야 합니다. 자세한 내용은 50 페이지 “IP 필터를 사용으로 설정하고 새로 고치는 방법”을 참조하십시오.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.

- **서비스를 사용 안함으로 설정하려면 `svcadm` 명령을 사용합니다.**

```
$ svcadm disable network/ipfilter
```

서비스를 테스트하거나 디버그하려면 서비스가 실행 중인 동안 규칙 세트를 제거할 수 있습니다. 자세한 내용은 53 페이지 “IP 필터 규칙 세트 작업”을 참조하십시오.

IP 필터 규칙 세트 작업

다음과 같은 경우 패킷 필터링 및 NAT 규칙을 수정하거나 비활성화할 수 있습니다.

- 테스트 용도로 사용하려는 경우
- 문제의 원인이 IP 필터인 것으로 간주되어 시스템 문제를 해결하려는 경우

다음 작업 맵에서는 IP 필터 규칙 세트와 관련된 절차를 식별합니다.

표 5–2 IP 필터 규칙 세트 작업(작업 맵)

작업	수행 방법
활성 패킷 필터링 규칙 세트를 확인합니다.	54 페이지 "활성 패킷 필터링 규칙 세트 확인 방법"
비활성 패킷 필터링 규칙 세트를 확인합니다.	54 페이지 "비활성 패킷 필터링 규칙 세트 확인 방법"
다른 활성 규칙 세트를 활성화합니다.	54 페이지 "다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법"
규칙 세트를 제거합니다.	55 페이지 "패킷 필터링 규칙 세트 제거 방법"
규칙 세트에 규칙을 추가합니다.	56 페이지 "활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법" 57 페이지 "비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법"
활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.	58 페이지 "활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법"
커널에서 비활성 규칙 세트를 삭제합니다.	59 페이지 "커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법"
활성 NAT 규칙을 확인합니다.	59 페이지 "IP 필터에서 활성 NAT 규칙을 확인하는 방법"
NAT 규칙을 제거합니다.	60 페이지 "IP 필터에서 NAT 규칙을 비활성화하는 방법"
규칙을 추가하여 NAT 규칙을 활성화합니다.	60 페이지 "NAT 패킷 필터링 규칙에 규칙을 추가하는 방법"
활성 주소 풀을 확인합니다.	61 페이지 "활성 주소 풀 확인 방법"
주소 풀을 제거합니다.	61 페이지 "주소 풀 제거 방법"
주소 풀에 규칙을 추가합니다.	62 페이지 "주소 풀에 규칙을 추가하는 방법"

IP 필터에 대한 패킷 필터링 규칙 세트 관리

IP 필터에서는 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.

주 – 다음 절차는 IPv4 네트워크의 예를 제공합니다. IPv6 패킷의 경우 48 페이지 "IP 필터 서비스 기본값을 표시하는 방법"의 단계 2에 설명된 대로 -6 옵션을 사용합니다.

▼ 활성 패킷 필터링 규칙 세트 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **활성 패킷 필터링 규칙 세트를 확인합니다.**

  다음 예에서는 커널에서 로드된 활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
$ ipfstat -io
empty list for ipfilter(out)
pass in quick on net1 from 192.168.1.0/24 to any
pass in all
block in on net1 from 192.168.1.10/32 to any
```

▼ 비활성 패킷 필터링 규칙 세트 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **비활성 패킷 필터링 규칙 세트를 확인합니다.**

  다음 예에서는 비활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
$ ipfstat -I -io
pass out quick on net1 all
pass in quick on net1 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 다시 로드합니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1. **다음 단계 중 하나를 선택합니다.**
   - 완전히 다른 규칙 세트를 활성화하려면 별도의 파일에 새 규칙 세트를 만듭니다.
   - 구성 파일에서 현재 규칙 세트를 업데이트합니다.
2. **현재 규칙 세트를 제거하고 새 규칙 세트를 로드합니다.**

```
$ ipf -Fa -f filename
```

*filename*의 규칙이 활성 규칙 세트를 대체합니다.

주 – 업데이트된 규칙 세트를 로드하려면 `ipf -D` 또는 `svcadm restart` 등의 명령을 사용하지 마십시오. 새 규칙 세트를 로드하기 전에 먼저 방화벽을 사용 안함으로 설정하므로 해당 명령으로 인해 네트워크가 노출됩니다.

예 5–1 다른 패킷 필터링 규칙 세트 활성화

다음 예에서는 특정 패킷 필터링 규칙 세트를 다른 규칙 세트로 바꾸는 방법을 보여 줍니다.

```
$ ipfstat -io
empty list for ipfilter(out)
pass in quick on net0 all
$ ipf -Fa -f /etc/ipf/ipfnew.conf
$ ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
```

예 5–2 업데이트된 패킷 필터링 규칙 세트 다시 로드

다음 예에서는 현재 활성 상태이며 업데이트된 패킷 필터링 규칙 세트를 다시 로드하는 방법을 보여 줍니다.

```
$ ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any

(Edit the /etc/ipf/myorg.ipf.conf configuration file.)

$ svcadm refresh network/ipfilter
$ ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any
block in quick on net11 from 192.168.0.0/12 to any
```

▼ 패킷 필터링 규칙 세트 제거 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **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 5–3 패킷 필터링 규칙 세트 제거

다음 예에서는 활성 필터링 규칙 세트에서 모든 필터링 규칙을 제거하는 방법을 보여 줍니다.

```
$ ipfstat -io
block out log on net0 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

기존 규칙 세트에 규칙을 추가하면 테스트 또는 디버그 시 유용할 수 있습니다. 규칙이 추가된 경우 IP 필터 서비스는 계속 사용으로 설정됩니다. 하지만 서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, IP 필터 서비스의 등록 정보 파일에 규칙이 없으면 해당 규칙이 손실됩니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.**
  - `ipf -f -` 명령을 사용하여 명령줄에서 규칙 세트에 규칙을 추가합니다.

    ```
    $ echo "block in on net1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
    ```

    서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, 추가된 이 규칙은 IP 필터 구성의 일부가 아닙니다.
  - 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.

*filename*이 IP 필터 구성 파일 등록 정보 중 하나의 값이면 서비스를 사용으로 설정하거나 다시 시작하거나 새로 고치는 경우 해당 규칙이 다시 로드됩니다. 그렇지 않은 경우 추가된 규칙이 임시 규칙 세트를 제공합니다.

예 5–4 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 net1 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 net1 proto tcp from 10.1.1.1/32 to any
```

▼ 비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법

커널에서 비활성 규칙 세트를 만들면 테스트 또는 디버그 시 유용할 수 있습니다. IP 필터 서비스를 중지하지 않고도 해당 규칙 세트를 활성 규칙 세트로 전환할 수 있습니다. 하지만 서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, 비활성 규칙 세트를 추가해야 합니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 선택한 파일에 규칙 세트를 만듭니다.

2 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 5–5 비활성 규칙 세트에 규칙 추가

다음 예에서는 파일에서 비활성 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
$ ipfstat -I -io
pass out quick on net1 all
pass in quick on net1 all
$ ipf -I -f /etc/ipf/ipftrial.conf
$ ipfstat -I -io
pass out quick on net1 all
pass in quick on net1 all
block in log quick from 10.0.0.0/8 to any
```

▼ 활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법

커널에서 다른 규칙 세트로 전환하면 테스트 또는 디버그 시 유용할 수 있습니다. IP 필터 서비스를 중지하지 않고도 해당 규칙 세트를 활성화할 수 있습니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.**

  ```
  $ ipf -s
  ```

  이 명령을 사용하면 커널에서 활성 규칙 세트와 비활성 규칙 세트 간에 전환할 수 있습니다. 비활성 규칙 세트가 비어 있을 경우 패킷 필터링이 없는 것입니다.

주 – IP 필터 서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하면 IP 필터 서비스의 등록 정보 파일에 있는 규칙이 복원됩니다. 비활성 규칙 세트는 복원되지 않습니다.

예 5–6 활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환

다음 예에서는 `ipf -s` 명령을 사용하여 비활성 규칙 세트를 활성 규칙 세트로 전환하고 활성 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 활성 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 net1 proto tcp from 10.1.1.1/32 to any
  $ ipfstat -I -io
  pass out quick on net1 all
  pass in quick on net1 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 net1 all
  pass in quick on net1 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 net1 proto tcp from 10.1.1.1/32 to any
```

▼ 커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **"모두 비우기" 명령에 비활성 규칙 세트를 지정합니다.**

```
$ ipf -I -Fa
```

주 – 나중에 `ipf -s`를 실행할 경우 비어 있는 비활성 규칙 세트가 활성 규칙 세트로 전환됩니다. 활성 규칙 세트가 비어 있을 경우 필터링이 수행되지 **않습니다**.

예 5–7 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 net1 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 IP 필터의 NAT 규칙을 관리, 확인 및 수정할 수 있습니다.

▼ IP 필터에서 활성 NAT 규칙을 확인하는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **활성 NAT 규칙을 확인합니다.**

  다음 예에서는 활성 NAT 규칙 세트의 출력을 보여 줍니다.

```
$ ipnat -l
List of active MAP/Redirect filters:
map net0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

▼ IP 필터에서 NAT 규칙을 비활성화하는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **커널에서 NAT 규칙을 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.

예 5–8 NAT 규칙 제거

다음 예에서는 현재 NAT 규칙의 항목을 제거하는 방법을 보여 줍니다.

```
$ ipnat -l
List of active MAP/Redirect filters:
map net0 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 패킷 필터링 규칙에 규칙을 추가하는 방법

기존 규칙 세트에 규칙을 추가하면 테스트 또는 디버그 시 유용할 수 있습니다. 규칙이 추가된 경우 IP 필터 서비스는 계속 사용으로 설정됩니다. 하지만 서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, IP 필터 서비스의 등록 정보 파일에 NAT 규칙이 없으면 해당 규칙이 손실됩니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.**
  - `ipnat -f -` 명령을 사용하여 명령줄에서 NAT 규칙 세트에 규칙을 추가합니다.

    ```
    $ echo "map net0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
    ```

    서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, 추가된 이 규칙은 IP 필터 구성의 일부가 아닙니다.
  - 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.

*filename*이 IP 필터 구성 파일 등록 정보 중 하나의 값이면 서비스를 사용으로 설정하거나 다시 시작하거나 새로 고치는 경우 해당 규칙이 다시 로드됩니다. 그렇지 않은 경우 추가된 규칙이 임시 규칙 세트를 제공합니다.

예 5–9 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 net0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
$ ipnat -l
List of active MAP/Redirect filters:
map net0 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

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **활성 주소 풀을 확인합니다.**

  다음 예에서는 활성 주소 풀의 컨텐츠를 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

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **현재 주소 풀의 항목을 제거합니다.**

  ```
  $ ippool -F
  ```

예 5–10 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

기존 규칙 세트에 규칙을 추가하면 테스트 또는 디버그 시 유용할 수 있습니다. 규칙이 추가된 경우 IP 필터 서비스는 계속 사용으로 설정됩니다. 하지만 서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, IP 필터 서비스의 등록 정보 파일에 주소 풀 규칙이 없으면 해당 규칙이 손실됩니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

1 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 -
  ```

  서비스를 새로 고치거나 다시 시작하거나 사용으로 설정하는 경우, 추가된 이 규칙은 IP 필터 구성의 일부가 아닙니다.

- 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.

2 규칙에 원래 규칙 세트에 없는 풀이 포함되어 있으면 다음 단계를 수행합니다.

a. 새 패킷 필터링 규칙에 풀을 추가합니다.

b. 현재 규칙 세트에 새 패킷 필터링 규칙을 추가합니다.

56 페이지 "활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법"의 지침을 따릅니다.

주 – IP 필터 서비스를 새로 고치거나 다시 시작하지 마십시오. 추가한 주소 풀 규칙이 손실됩니다.

예 5–11 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 5–3 IP 필터 통계 및 정보 표시(작업 맵)

작업	수행 방법
상태 테이블을 확인합니다.	63 페이지 "IP 필터에 대한 상태 테이블 확인 방법"
패킷 상태에 대한 통계를 확인합니다.	64 페이지 "IP 필터에 대한 상태 통계 확인 방법"
IP 필터 조정 가능 매개변수를 나열합니다.	65 페이지 "IP 필터 조정 가능 매개변수를 확인하는 방법"
NAT 통계를 확인합니다.	65 페이지 "IP 필터에 대한 NAT 통계 확인 방법"
주소 풀 통계를 확인합니다.	66 페이지 "IP 필터에 대한 주소 풀 통계 확인 방법"

▼ IP 필터에 대한 상태 테이블 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **상태 테이블을 확인합니다.**

```
$ ipfstat
```

주 – -t 옵션을 사용하여 UNIX top 유틸리티 형식으로 상태 테이블을 확인할 수 있습니다.

예 5–12 IP 필터에 대한 상태 테이블 보기

다음 예에서는 상태 테이블 출력을 보여 줍니다.

```
$ ipfstat
bad packets:            in 0    out 0
 IPv6 packets:          in 56286 out 63298
 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  not fragmented 0
fragment reassembly(in):bad v6 hdr 0     bad v6 ehdr 0  failed reassembly 0
fragment state(out):    kept 0  lost 0  not fragmented 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

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **상태 통계를 확인합니다.**

```
$ ipfstat -s
```

예 5–13 IP 필터에 대한 상태 통계 보기

다음 예에서는 상태 통계 출력을 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 필터 조정 가능 매개변수를 확인하는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **IP 필터에 대한 커널 조정 가능 매개변수를 확인합니다.**

  다음 출력은 잘립니다.

```
$ ipf -T list
fr_flags    min 0    max 0xffffffff    current 0
fr_active    min 0    max 0    current 0
...
ipstate_logging    min 0    max 0x1    current 1
...
fr_authq_ttl    min 0x1    max 0x7fffffff    current sz = 0
fr_enable_rcache    min 0    max 0x1    current 0
```

▼ IP 필터에 대한 NAT 통계 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **NAT 통계를 확인합니다.**

```
$ ipnat -s
```

예 5–14 IP 필터에 대한 NAT 통계 보기

다음 예에서는 NAT 통계를 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

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **주소 풀 통계를 확인합니다.**

```
$ ippool -s
```

예 5–15 IP 필터에 대한 주소 풀 통계 보기

다음 예에서는 주소 풀 통계를 보여 줍니다.

```
$ ippool -s
Pools:  3
Hash Tables:    0
Nodes:  0
```

IP 필터 로그 파일 작업

표 5–4 IP 필터 로그 파일 작업(작업 맵)

작업	수행 방법
별도의 IP 필터 로그 파일을 만듭니다.	66 페이지 "IP 필터 로그 파일 설정 방법"
상태, NAT 및 일반 로그 파일을 확인합니다.	67 페이지 "IP 필터 로그 파일 확인 방법"
패킷 로그 버퍼를 비웁니다.	68 페이지 "패킷 로그 버퍼를 비우는 방법"
나중에 참조할 수 있도록 기록된 패킷을 파일에 저장합니다.	69 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 만드는 것이 좋습니다.

시작하기 전에 `root` 역할이 있어야 합니다.

1 **온라인 상태의 system-log 서비스 인스턴스를 확인합니다.**

```
# svcs system-log
STATE          STIME    FMRI
disabled       13:11:55 svc:/system/system-log:rsyslog
online         13:13:27 svc:/system/system-log:default
```

주 – rsyslog 서비스 인스턴스가 온라인이면 rsyslog.conf 파일을 수정합니다.

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 – 입력 시 스페이스바가 아닌 Tab 키를 사용하여 local0.debug와 /var/log/*log-name*을 구분합니다. 자세한 내용은 syslog.conf(4) 및 syslogd(1M) 매뉴얼 페이지를 참조하십시오.

3 **새 로그 파일을 만듭니다.**

```
# touch /var/log/log-name
```

4 **system-log 서비스에 대한 구성 정보를 새로 고칩니다.**

```
# svcadm refresh system-log:default
```

주 – rsyslog 서비스가 온라인이면 system-log:rsyslog 서비스 인스턴스를 새로 고칩니다.

예 5–16 IP 필터 로그 만들기

다음 예에서는 IP 필터 정보를 아카이브할 ipmon.log를 만드는 방법을 보여 줍니다.

/etc/syslog.conf에서 다음을 입력합니다.

```
## Save IP Filter log output to its own file
local0.debug<Tab>/var/log/ipmon.log
```

명령줄에서 다음을 입력합니다.

```
# touch /var/log/ipmon.log
# svcadm restart system-log
```

▼ IP 필터 로그 파일 확인 방법

시작하기 전에 66 페이지 "IP 필터 로그 파일 설정 방법"을 완료했습니다.

IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **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*

  - **ipmon 데몬을 중지한 후 ipmon 명령을 사용하여 상태, NAT 및 IP 필터 로그 파일을 표시할 수 있습니다.**

    # pkill ipmon
    # 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 ipmon의 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 5–17 IP 필터 로그 파일 보기

다음 예에서는 /var/ipmon.log의 출력을 보여 줍니다.

```
# ipmon -o SNI /var/ipmon.log
02/09/2012 15:27:20.606626 net0 @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
02/09/2012 15:27:20.606626 net0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

▼ 패킷 로그 버퍼를 비우는 방법

이 절차에서는 버퍼를 지우고 화면에 출력을 표시합니다.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **패킷 로그 버퍼를 비웁니다.**

  ```
  # ipmon -F
  ```

예 5–18 패킷 로그 버퍼 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

디버그하는 동안 또는 트래픽을 수동으로 감사하는 경우 패킷을 파일에 저장할 수 있습니다.

시작하기 전에 root 역할이 있어야 합니다.

- **기록된 패킷을 파일에 저장합니다.**

  ```
  # cat /dev/ipl > filename
  ```

  명령줄 프롬프트를 다시 가져올 Ctrl-C를 입력하여 프로시저를 중단할 때까지 *filename* 파일에 패킷이 계속 기록됩니다.

예 5–19 기록된 패킷을 파일에 저장

다음 예에서는 기록된 패킷을 파일에 저장한 후의 결과를 보여 줍니다.

```
# cat /dev/ipl > /tmp/logfile
^C#

# ipmon -f /tmp/logfile
02/09/2012 15:30:28.708294 net0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 52 -S IN
02/09/2012 15:30:28.708708 net0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2012 15:30:28.792611 net0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 70 -AP IN
02/09/2012 15:30:28.872000 net0 @0:1 p 129.146.157.149,33923 ->
 129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2012 15:30:28.872142 net0 @0:1 p 129.146.157.149,33923 ->
```

```
  129.146.157.145,23 PR tcp len 20 43 -AP IN
02/09/2012 15:30:28.872808 net0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2012 15:30:28.872951 net0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 47 -AP IN
02/09/2012 15:30:28.926792 net0 @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 예

다음 예에서는 단일 호스트, 서버 및 라우터에 적용되는 패킷 필터링 규칙을 보여 줍니다.

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.

자세한 구문 정보는 40 페이지 “패킷 필터링 규칙 구성”을 참조하십시오.

예 5–20 IP 필터 호스트 구성

이 예에서는 `net0` 네트워크 인터페이스가 있는 호스트 시스템에 대한 구성을 보여 줍니다.

```
# pass and log everything by default
pass in log on net0 all
pass out log on net0 all

# block, but don't log, incoming packets from other reserved addresses
block in quick on net0 from 10.0.0.0/8 to any
block in quick on net0 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 IP Filter.
block in log quick from 192.168.1.15 to <thishost>
block in log quick from 192.168.1.43 to <thishost>

# block and log X11 (port 6000) and remote procedure call
# and portmapper (port 111) attempts
block in log quick on net0 proto tcp from any to net0/32 port = 6000 keep state
block in log quick on net0 proto tcp/udp from any to net0/32 port = 111 keep state
```

이 규칙 세트는 `net0` 인터페이스에서 모든 항목을 주고받을 수 있도록 허용하는 제한되지 않은 두 개의 규칙으로 시작합니다. 두번째 규칙 세트는 개인 주소 공간

예 5–20 IP 필터 호스트 구성 *(계속)*

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 5–21 IP 필터 서버 구성

이 예에서는 웹 서버로 사용되는 호스트 시스템에 대한 구성을 보여 줍니다. 이 시스템에는 net0 네트워크 인터페이스가 있습니다.

```
# web server with an net0 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 net0 from any to any head 100
block out log on net0 from any to any head 200


# web rules that get hit most often
pass in quick on net0 proto tcp from any \
to net0/32 port = http flags S keep state group 100
pass in quick on net0 proto tcp from any \
to net0/32 port = https flags S keep state group 100


# inbound traffic - ssh, auth
pass in quick on net0 proto tcp from any \
to net0/32 port = 22 flags S keep state group 100
pass in log quick on net0 proto tcp from any \
to net0/32 port = 113 flags S keep state group 100
pass in log quick on net0 proto tcp from any port = 113 \
to net0/32 flags S keep state group 100

# outbound traffic - DNS, auth, NTP, ssh, WWW, smtp
pass out quick on net0 proto tcp/udp from net0/32 \
to any port = domain flags S keep state group 200
pass in quick on net0 proto udp from any \
port = domain to net0/32 group 100

pass out quick on net0 proto tcp from net0/32 \
to any port = 113 flags S keep state group 200
pass out quick on net0 proto tcp from net0/32 port = 113 \
to any flags S keep state group 200
```

예 5–21 IP 필터 서버 구성 *(계속)*

```
pass out quick on net0 proto udp from net0/32 to any \
port = ntp group 200
pass in quick on net0 proto udp from any \
port = ntp to net0/32 port = ntp group 100

pass out quick on net0 proto tcp from net0/32 \
to any port = ssh flags S keep state group 200

pass out quick on net0 proto tcp from net0/32 \
to any port = http flags S keep state group 200
pass out quick on net0 proto tcp from net0/32 \
to any port = https flags S keep state group 200

pass out quick on net0 proto tcp from net0/32 \
to any port = smtp flags S keep state group 200


# pass icmp packets in and out
pass in quick on net0 proto icmp from any to net0/32  keep state group 100
pass out quick on net0 proto icmp from net0/32 to any keep state group 200

# block and ignore NETBIOS packets
block in quick on net0 proto tcp from any \
to any port = 135 flags S keep state group 100

block in quick on net0 proto tcp from any port = 137 \
to any flags S keep state group 100
block in quick on net0 proto udp from any to any port = 137 group 100
block in quick on net0 proto udp from any port = 137 to any group 100

block in quick on net0 proto tcp from any port = 138 \
to any flags S keep state group 100
block in quick on net0 proto udp from any port = 138 to any group 100

block in quick on net0 proto tcp from any port = 139 to any flags S keep state
group 100
block in quick on net0 proto udp from any port = 139 to any group 100
```

예 5–22 IP 필터 라우터 구성

이 예에서는 내부 인터페이스 `net0` 및 외부 인터페이스 `net1`이 있는 라우터에 대한 구성을 보여 줍니다.

```
# internal interface is net0 at 192.168.1.1
# external interface is net1 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 net0 all
```

예 5–22 IP 필터 라우터 구성 *(계속)*

```
block in log on net1 all
block out log on net0 all
block out log on net1 all


# Packets going in/out of network interfaces that aren't on the loopback
# interface should not exist.
block in log quick on net0 from 127.0.0.0/8 to any
block in log quick on net0 from any to 127.0.0.0/8
block in log quick on net1 from 127.0.0.0/8 to any
block in log quick on net1 from any to 127.0.0.0/8


# Deny reserved addresses.
block in quick on net1 from 10.0.0.0/8 to any
block in quick on net1 from 172.16.0.0/12 to any
block in log quick on net1 from 192.168.1.0/24 to any
block in quick on net1 from 192.168.0.0/16 to any


# Allow internal traffic
pass in quick on net0 from 192.168.1.0/24 to 192.168.1.0/24
pass out quick on net0 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 net1 proto tcp/udp from net1/32 to any port = domain keep state
pass in quick on net0 proto tcp/udp from 192.168.1.2 to any port = domain keep state
pass in quick on net0 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 net0 proto udp from 192.168.1.0/24 to any port = 123 keep state
pass out quick on net1 proto udp from any to any port = 123 keep state


# Allow incoming mail
pass in quick on net1 proto tcp from any to net1/32 port = smtp keep state
pass in quick on net1 proto tcp from any to net1/32 port = smtp keep state
pass out quick on net1 proto tcp from 192.168.1.0/24 to any port = smtp keep state


# Allow outgoing connections: SSH, WWW, NNTP, mail, whois
pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = 22 keep state
pass out quick on net1 proto tcp from 192.168.1.0/24 to any port = 22 keep state

pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass out quick on net1 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = 443 keep state
pass out quick on net1 proto tcp from 192.168.1.0/24 to any port = 443 keep state

pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = nntp keep state
block in quick on net1 proto tcp from any to any port = nntp keep state
pass out quick on net1 proto tcp from 192.168.1.0/24 to any port = nntp keep state
```

예 5–22 IP 필터 라우터 구성 *(계속)*

```
pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = smtp keep state

pass in quick on net0 proto tcp from 192.168.1.0/24 to any port = whois keep state
pass out quick on net1 proto tcp from any to any port = whois keep state


# Allow ssh from offsite
pass in quick on net1 proto tcp from any to net1/32 port = 22 keep state


# Allow ping out
pass in quick on net0 proto icmp all keep state
pass out quick on net1 proto icmp all keep state


# allow auth out
pass out quick on net1 proto tcp from net1/32 to any port = 113 keep state
pass out quick on net1 proto tcp from net1/32 port = 113 to any keep state


# return rst for incoming auth
block return-rst in quick on net1 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 net1 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
```

◆ ◆ ◆ 6 장

6 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.

네트워크에서 IPsec를 구현하려면 7 장, “IPsec 구성(작업)”을 참조하십시오. 참조 정보는 8 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 컨텐츠는 암호화되지 않지만, 수신자에게 패킷 컨텐츠가 변경되지 않았음을 보장합니다. 또한 패킷이 발신자에 의해 보내졌음을 수신자에게 보장합니다. ESP(보안 페이로드 캡슐화)는 IP 데이터를 암호화하므로 패킷 전송 중 컨텐츠를 숨깁니다. 또한 ESP는 인증 알고리즘 옵션을 통해 데이터 무결성을 보장할 수 있습니다.
- **SA(보안 연결)** – 네트워크 트래픽의 특정 플로우에 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 적법한 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 적법한지 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 9–1을 참조하십시오.

표 6–1 IPsec 용어, 약어 및 사용

IPsec 용어	머리글자어	정의
보안 연결	SA	네트워크 트래픽의 특정 플로우에 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 6–1 IPsec 용어, 약어 및 사용 *(계속)*

IPsec 용어	머리글자어	정의
Diffie-Hellman	DH	키 생성 및 키 인증을 허용하는 키 교환 알고리즘입니다. **인증된 키 교환**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 플로우

그림 6–1은 IPsec가 아웃바운드 패킷에서 호출될 때 IP 주소 지정된 패킷이 IP 데이터그램의 일부로 진행되는지 보여줍니다. 플로우 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 6–2는 IPsec 인바운드 프로세스를 보여줍니다.

그림 6–1 아웃바운드 패킷 프로세스에 적용된 IPsec

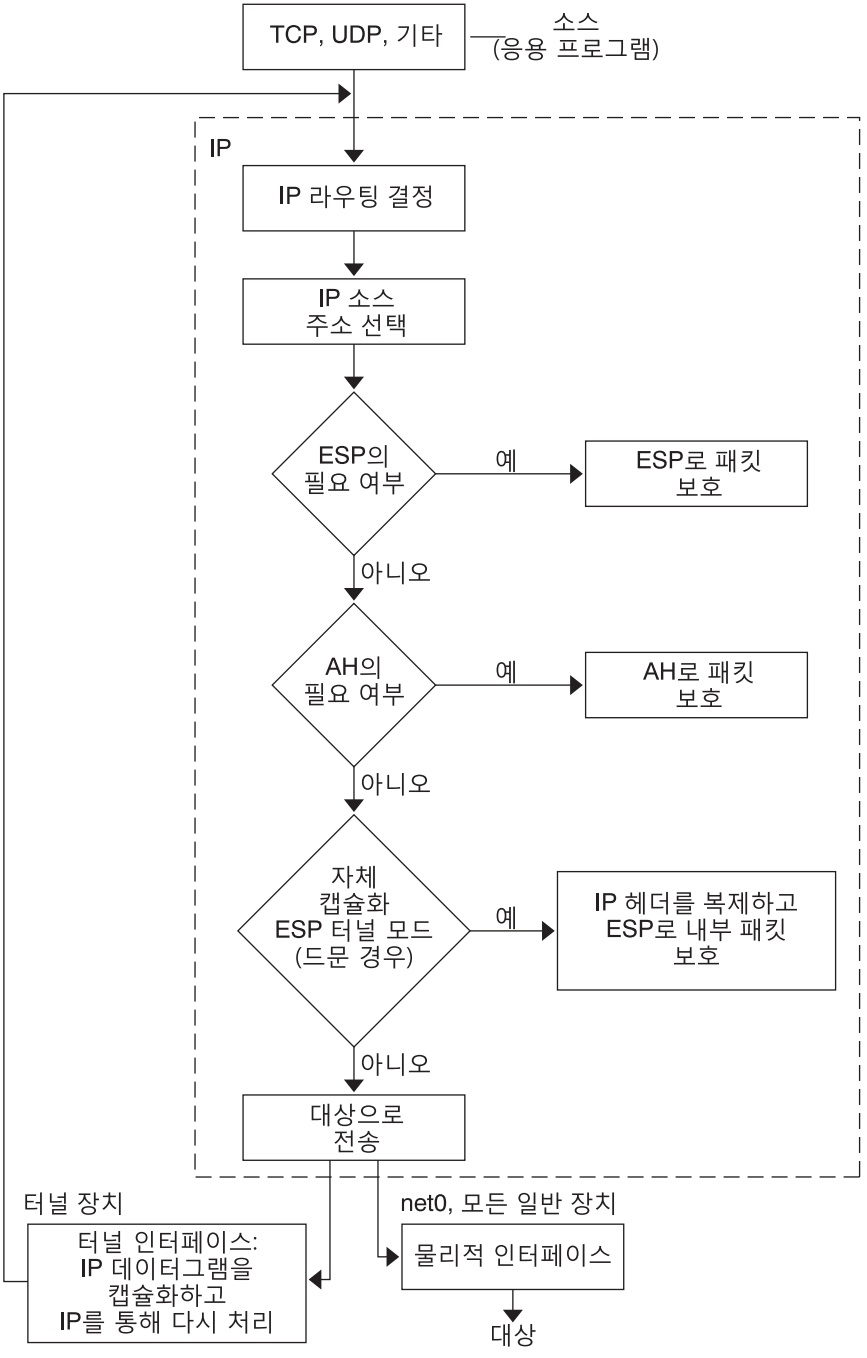
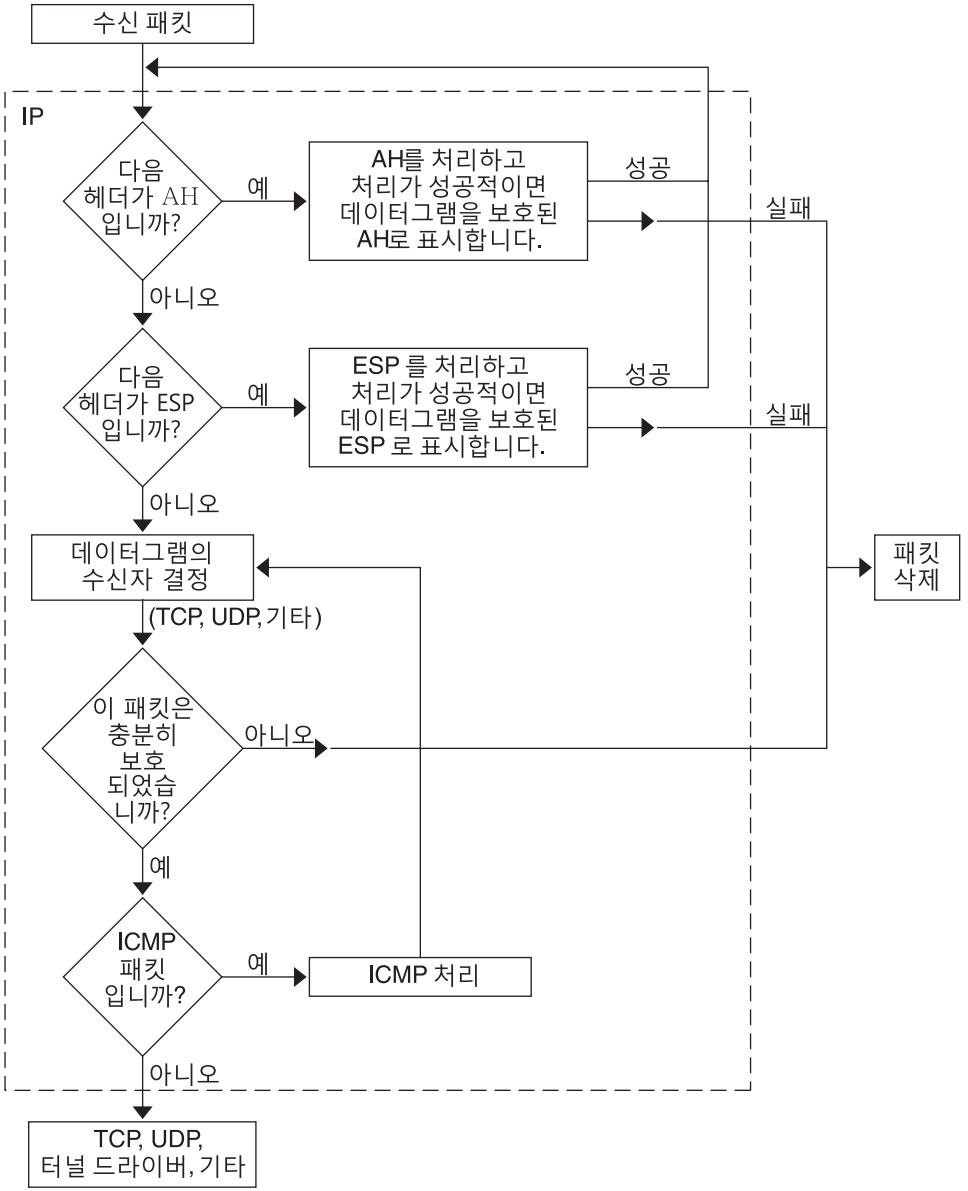


그림 6–2 인바운드 패킷 프로세스에 적용된 IPsec



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 ESP로 보호되는 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_KEY`는 권한이 부여된 응용 프로그램이 데이터베이스를 관리하도록 합니다. 예를 들어, IKE 응용 프로그램 및 `ipseckeys` 명령은 `PF_KEY` 소켓 인터페이스를 사용합니다.

- IPsec SADB에 대한 자세한 설명은 116 페이지 “IPsec에 대한 보안 연결 데이터베이스”를 참조하십시오.
- SADB를 관리하는 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 관리**라고 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 9 장, “Internet Key Exchange(개요)”를 참조하십시오. `in.iked` 데몬에 대한 자세한 내용은 `in.iked(1M)` 매뉴얼 페이지를 참조하십시오. `ike` 서비스에 대한 자세한 내용은 159 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 117 페이지 “IPsec에서 SA 생성을 위한 유틸리티”를 참조하십시오. `ipseckey` 명령 옵션에 대한 자세한 설명은 `ipseckey(1M)` 매뉴얼 페이지를 참조하십시오.

IPsec 보호 방식

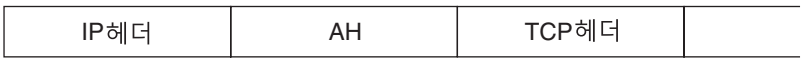
IPsec는 데이터 보호를 위한 두 가지 보안 프로토콜을 제공합니다.

- 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.

인증 헤더

인증 헤더는 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.



전송 헤더는 TCP, UDP, SCTP 또는 ICMP가 될 수 있습니다. 터널이 사용되는 경우 전송 헤더는 다른 IP 헤더가 될 수 있습니다.

ESP(Encapsulating Security Payload)

ESP(보안 페이로드 캡슐화) 모듈은 ESP가 캡슐화하는 컨텐츠에 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 85 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 6–2 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. ■ 발신자가 보낸 컨텐츠를 그대로 수신자가 수신할 수 있도록 합니다. ■ AH에서 재생 보호를 사용으로 설정하지 않을 경우 재생 공격에 취약합니다.	재생, 잘라내기 및 붙여넣기
ESP	데이터그램에서 ESP 시작 이후의 패킷을 보호합니다.	암호화 옵션을 사용하여 IP 페이로드를 암호화합니다. 기밀성을 유지합니다.	도청
		인증 옵션을 사용하여 AH와 동일한 페이로드 보호 기능을 제공합니다.	재생, 잘라내기 및 붙여넣기
		두 옵션을 모두 사용하면 강력한 무결성, 데이터 인증 및 기밀성을 제공할 수 있습니다.	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 `ipsecalgs` 명령을 사용하여 얻을 수 있습니다. 자세한 내용은 `ipsecalgs(1M)` 매뉴얼 페이지를 참조하십시오. 또한 `getipsecalgbyname(3NSL)` 매뉴얼 페이지에 설명된 기능을 사용하여 알고리즘의 등록 정보를 검색할 수 있습니다.

IPsec는 암호화 프레임워크를 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 11.1 관리: 보안 서비스**의 11 장, “암호화 프레임워크(개요)”
- **Developer’s Guide to Oracle Solaris 11 Security**의 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 시스템간 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 컨텐츠를 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 컨텐츠에 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 99 페이지 “IPsec를 사용하여 VPN 보호”를 참조하십시오.

Oracle Solaris에서 터널 모드는 IP 터널링 네트워크 인터페이스에만 적용할 수 있습니다. 터널링 인터페이스에 대한 자세한 내용은 **Oracle Solaris 11.1 네트워크 구성 및 관리**의 6장, “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 6–3 TCP 정보를 전달하는 보호되지 않는 IP 패킷

전송 모드에서 ESP가 다음 그림에 나온 대로 데이터를 보호합니다. 음영 영역은 패킷의 암호화된 부분을 나타냅니다.

그림 6–4 TCP 정보를 전달하는 보호된 IP 패킷

암호화됨

전송 모드에서 AH가 다음 그림에 나온 대로 데이터를 보호합니다.

그림 6–5 인증 헤더로 보호된 패킷

AH 보호는 전송 모드라도 IP 헤더의 대부분을 포함합니다.

터널 모드에서 전체 데이터그램은 IPsec 헤더의 보호 **내부**에 있습니다. 그림 6–3의 데이터그램은 다음 그림에 나온 대로 외부 IPsec 헤더(이 경우 ESP)로 터널 모드에서 보호됩니다.

그림 6–6 터널 모드에서 보호된 IPsec 패킷

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 97 페이지 “IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법”을 참조하십시오.
- 터널에 대한 자세한 내용은 `ipsecconf(1M)` 매뉴얼 페이지를 참조하십시오.
- 터널 구성의 예는 102 페이지 “터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법”을 참조하십시오.

VPN(Virtual Private Networks) 및 IPsec

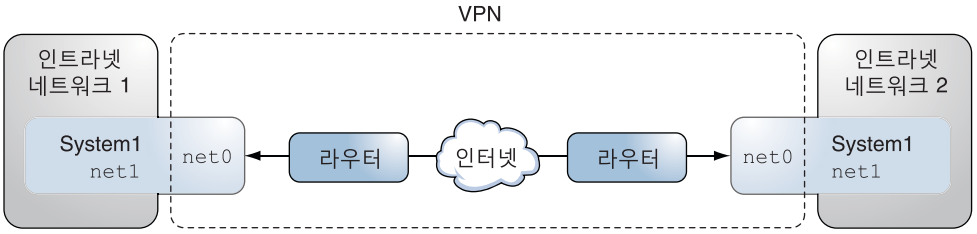
구성된 터널은 지점간 인터페이스입니다. 터널을 통해 한 IP 패킷을 다른 IP 패킷 내부에 캡슐화할 수 있습니다. 올바르게 구성된 터널에는 터널 소스와 터널 대상이 필요합니다. 자세한 내용은 **Oracle Solaris 11.1 네트워크 구성 및 관리**의 “IP 터널을 만들고 구성하는 방법”을 참조하십시오.

터널은 IP에 대한 분명한 물리적 인터페이스를 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 6–7 VPN(Virtual Private Networks)



설정 절차의 자세한 예는 102 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 로그온하는 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 42 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 149 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”을 참조하십시오.

IPsec 및 SCTP

Oracle Solaris는 SCTP(Streams Control Transmission Protocol)를 지원합니다. IPsec 정책 지정을 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 93 페이지 “IPsec를 사용하여 트래픽 보호”를 참조하십시오. 영역에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리**의 15 장, “Oracle Solaris 영역 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 도메인을 구성하는 방법에 대한 자세한 내용은 **Oracle VM Server for SPARC 2.2 관리자 설명서**를 참조하십시오.

IPsec 유틸리티 및 파일

표 6–3에서는 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 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"를 참조하십시오.

- 네트워크에서 IPsec 구현에 대한 지침은 93 페이지 "IPsec를 사용하여 트래픽 보호"를 참조하십시오.
- IPsec 유틸리티 및 파일에 대한 자세한 내용은 8 장, "IP 보안 아키텍처(참조)"를 참조하십시오.

표 6–3 일부 IPsec 유틸리티 및 파일 목록

IPsec 유틸리티, 파일 또는 서비스	설명	매뉴얼 페이지
`svc:/network/ipsec/ipsecalgs`	IPsec 알고리즘을 관리하는 SMF 서비스입니다.	`ipsecalgs(1M)`
`svc:/network/ipsec/manual-key`	키 입력 IPsec SA를 수동으로 관리하는 SMF 서비스입니다.	`ipseckey(1M)`
`svc:/network/ipsec/policy`	IPsec 정책을 관리하는 SMF 서비스입니다.	`smf(5)`, `ipsecconf(1M)`
`svc:/network/ipsec/ike`	IKE를 사용하여 IPsec SA의 자동 관리를 위한 SMF 서비스입니다.	`smf(5)`, `in.iked(1M)`
`/etc/inet/ipsecinit.conf` 파일	IPsec 정책 파일입니다.	`ipsecconf(1M)`
	SMF `policy` 서비스에서는 이 파일을 사용하여 시스템 부트 시 IPsec 정책을 구성합니다.	
`ipsecconf` 명령	IPsec 정책 명령입니다. 현재 IPsec 정책을 보고 수정하며 테스트하는 데 유용합니다.	`ipsecconf(1M)`
	SMF `policy` 서비스에서 시스템 부트 시 IPsec 정책을 구성하는 데 사용됩니다.	
`PF_KEY` 소켓 인터페이스	SADB(보안 연결 데이터베이스)에 대한 인터페이스입니다. 수동 키 관리 및 자동 키 관리를 처리합니다.	`pf_key(7P)`
`ipseckey` 명령	IPsec SA 키 입력 명령. `ipseckey`는 `PF_KEY` 인터페이스에 대한 명령줄 프론트 엔드입니다. `ipseckey`는 SA를 만들거나 삭제하거나 수정할 수 있습니다.	`ipseckey(1M)`
`/etc/inet/secret/ipseckeys` 파일	수동으로 키를 입력한 SA가 포함됩니다.	
	SMF `manual-key` 서비스에서 시스템 부트 시 SA를 수동으로 구성하는 데 사용됩니다.	

표 6–3 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 `ipsecalgs` 서비스에서 시스템 부트 시 알려진 IPsec 알고리즘을 커널과 동기화하는 데 사용됩니다.	`ipsecalgs(1M)`
`/etc/inet/ipsecalgs` 파일	구성된 IPsec 프로토콜 및 알고리즘 정의를 포함합니다. 이 파일은 `ipsecalgs` 명령으로 관리되며 수동으로 편집하면 안됩니다.	
`/etc/inet/ike/config` 파일	IKE 구성 및 정책 파일입니다. 기본적으로 이 파일은 존재하지 않습니다. 키 관리는 `/etc/inet/ike/config` 파일의 규칙 및 전역 매개변수를 기준으로 합니다. 122 페이지 “IKE 유틸리티 및 파일”을 참조하십시오. 이 파일이 존재하는 경우 `svc:/network/ipsec/ike` 서비스가 IKE 데몬 `in.iked`를 시작하여 자동 키 관리를 제공합니다.	`ike.config(4)`

◆ ◆ ◆ 7 장

7 IPsec 구성(작업)

이 장에서는 네트워크에서 IPsec를 구현하기 위한 절차를 설명합니다. 관련 절차는 다음 절에서 설명합니다.

IPsec에 대한 개요 정보는 6 장, “IP 보안 아키텍처(개요)”를 참조하십시오. IPsec에 대한 참조 정보는 8 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 99 페이지 “IPsec를 사용하여 VPN 보호”를 참조하십시오. IPsec를 관리하고 IPsec 및 IKE에서 SMF 명령을 사용하는 추가 절차는 106 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 11.1 관리: 보안 서비스**의 9 장, “역할 기반 액세스 제어 사용(작업)”을 참조하십시오. 예는 108 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”을 참조하십시오.
- **IPsec 및 SCTP** – IPsec는 SCTP(Streams Control Transmission Protocol) 연결을 보호하는 데 사용할 수 있지만 주의해야 합니다. 자세한 내용은 89 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.	94 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"
IPsec 정책을 사용하여 웹 서버를 보호합니다.	비웹 트래픽에서 IPsec를 사용하도록 합니다. 웹 클라이언트는 IPsec 검사를 우회하는 특정 포트로 식별됩니다.	97 페이지 "IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법"
IPsec 정책을 표시합니다.	현재 적용 중인 IPsec 정책을 적용 순서대로 표시합니다.	98 페이지 "IPsec 정책을 표시하는 방법"
IKE를 사용하여 IPsec SA에 대한 키 입력 자료를 자동으로 만듭니다.	보안 연결을 위한 원시 데이터를 제공합니다.	127 페이지 "IKE 구성(작업 맵)"
보안 VPN(virtual private network)을 설정합니다.	인터넷을 거치는 두 시스템 사이에 IPsec를 설정합니다.	99 페이지 "IPsec를 사용하여 VPN 보호"

▼ 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.

구성 명령을 실행하려면 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 시스템 파일을 편집하고 키를 만들려면 `root` 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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 `partym`인 시스템에서 `hosts` 파일에 다음을 입력합니다.

```
# Secure communication with enigma
192.168.116.16 enigma
```

b. 이름이 `enigma`인 시스템에서 `hosts` 파일에 다음을 입력합니다.

```
# Secure communication with partym
192.168.13.213 partym
```

2 각 시스템에서 IPsec 정책 파일을 만듭니다.

파일 이름은 `/etc/inet/ipsecinit.conf`입니다. 예는 `/etc/inet/ipsecinit.sample` 파일을 참조하십시오.

3 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.

4 **각 시스템에서 IKE를 구성하여 두 시스템 사이에 IPsec SA 쌍을 추가합니다.**

127 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 106 페이지 “IPsec 키를 수동으로 만드는 방법”을 참조하십시오.

---

5 **IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -f -c /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 **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 4에서 키를 수동으로 구성한 경우 106 페이지 “IPsec 키를 수동으로 만드는 방법”을 완료하여 키를 활성화합니다.

8 **패킷이 보호되고 있는지 확인합니다.**

절차는 111 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

예 7–1 `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. 관리자는 두 시스템에서 동일하게 정의됩니다. 자세한 내용은 `ssh(1)` 매뉴얼 페이지를 참조하십시오.

- 먼저 관리자는 위 절차의 단계 1 ~ 단계 5를 수행하여 첫번째 시스템을 구성합니다.
- 그런 다음 다른 터미널 창에서 관리자는 동일하게 정의된 사용자 이름 및 ID를 사용하여 `ssh` 명령을 통해 원격으로 로그인합니다.

```
local-system $ ssh -l jdoe other-system
other-system $ su - root
```

```
Enter password:
other-system #
```

- ssh 세션의 터미널 창에서 관리자는 단계 1 ~ 단계 7을 완료하여 두번째 시스템의 IPsec 정책 및 키를 구성합니다.
- 그런 다음 관리자는 ssh 세션을 종료합니다.

```
other-system # exit
local-system $ exit
```

- 마지막으로 관리자는 단계 6 및 단계 7를 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.

94 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.

구성 명령을 실행하려면 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 시스템 파일을 편집하려면 root 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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 53이 서버에 포함되어야 할 수도 있습니다.

2 **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 1에서 설명한 우회 예외 사항과 함께 보안 트래픽만 시스템에 액세스할 수 있도록 허용합니다.

3 **IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -f -c /etc/inet/ipsecinit.conf
```

4 **IPsec 정책을 새로 고칩니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

5 **IPsec에 대한 키를 새로 고칩니다.**

ike 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/ipsec/ike
```

키를 수동으로 구성한 경우 106 페이지 “IPsec 키를 수동으로 만드는 방법”의 지침을 따릅니다.

설정이 완료되었습니다. 선택적으로 단계 6을 수행할 수 있습니다.

6 **(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 -f -c /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.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

- **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
    ```

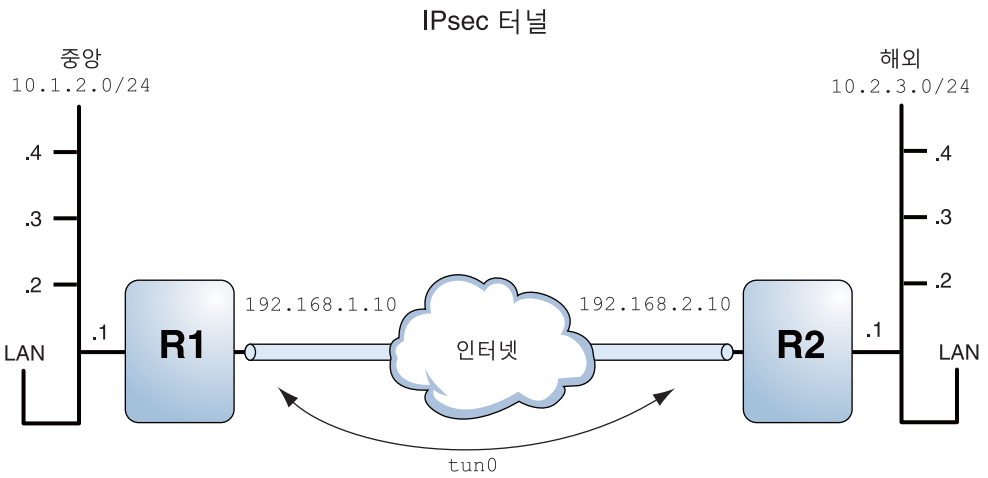
IPsec를 사용하여 VPN 보호

Oracle Solaris는 IPsec로 보호되는 VPN을 구성할 수 있습니다. 터널은 **터널 모드** 또는 **전송 모드**에서 만들 수 있습니다. 자세한 내용은 85 페이지 "IPsec의 전송 및 터널 모드"를 참조하십시오. 이 절의 예와 절차에서는 IPv4 주소를 사용하지만, 예와 절차는 IPv6 VPN에도 적용됩니다. 추가 정보는 93 페이지 "IPsec를 사용하여 트래픽 보호"를 참조하십시오.

터널 모드에서 터널에 대한 IPsec 정책의 예는 99 페이지 "터널 모드를 사용하여 IPsec로 VPN을 보호하는 예"를 참조하십시오.

터널 모드를 사용하여 IPsec로 VPN을 보호하는 예

그림 7–1 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 7–2 모든 서브넷에서 사용할 수 있는 터널 만들기

이 예에서는 그림 7–1에 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 7–3 두 서브넷만 연결하는 터널 만들기

이 예에서는 Central LAN의 서브넷 `10.1.2.0/24`와 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.

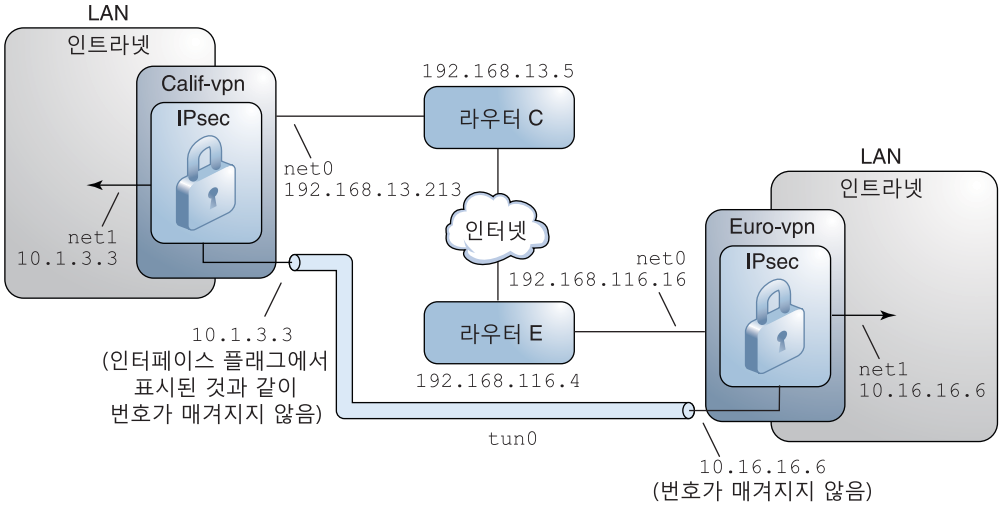
```
## IPsec policy ##
{tunnel tun0 negotiate tunnel laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs aes encr_auth_algs sha512 shared}
```

VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명

이 절에 나오는 절차에서는 다음 설정을 가정합니다. 네트워크 그림은 그림 7–2를 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 7–2 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 6의 *-point* 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 6의 *tsrc* 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 자세한 내용은 **Oracle Solaris 11.1 네트워크 구성 및 관리**의 "dladm 명령을 통한 터널 구성 및 관리"를 참조하십시오. 주소 객체에 대한 자세한 내용은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법" 및 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

▼ 터널 모드에서 IPsec를 사용하여 VPN을 보호하는 방법

터널 모드에서 내부 IP 패킷은 해당 컨텐츠를 보호하는 IPsec 정책을 결정합니다.

이 절차는 절차 94 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"을 확장합니다. 설정은 101 페이지 "VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명"에 설명되어 있습니다.

특정 명령을 실행하는 이유에 대한 자세한 설명은 94 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.

구성 명령을 실행하려면 Network Management 및 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 시스템 파일을 편집하려면 `root` 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 IPsec를 구성하기 전에 패킷의 플로우를 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 홈 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
```

2 IPsec 정책을 추가합니다.

`/etc/inet/ipsecinit.conf` 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다. 추가 예는 99 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}
```

3 각 시스템에서 IKE를 구성하여 두 시스템 사이에 IPsec SA 쌍을 추가합니다.

127 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 106 페이지 “IPsec 키를 수동으로 만드는 방법”을 참조하십시오.

---

4 IPsec 정책 파일의 구문을 확인합니다.

```
# ipsecconf -f -c /etc/inet/ipsecinit.conf
```

오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

5 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
```

6 *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 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오. 사용자 정의된 이름에 대한 자세한 내용은 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화**의 "네트워크 장치 및 데이터 링크 이름"을 참조하십시오.

7 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.

8 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.

9 네트워크 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/initial:default
```

10 **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. ■ IPsec 알고리즘 이름 및 키 입력 자료 ■ SPI(보안 매개변수 색인) ■ IP 소스 및 대상 주소와 기타 매개변수	106 페이지 "IPsec 키를 수동으로 만드는 방법"
네트워크 보안 역할을 만듭니다.	보안 네트워크를 설정할 수 있지만 root 역할보다 권한이 적은 역할을 만듭니다.	108 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.	110 페이지 "IPsec 및 IKE 서비스를 관리하는 방법"
IPsec가 패킷을 보호하고 있는지 확인합니다.	IP 데이터그램이 어떻게 보호되는지 나타내는 특정 헤더에 대한 snoop 출력을 검사합니다.	111 페이지 "IPsec로 패킷이 보호되는지 확인하는 방법"

▼ IPsec 키를 수동으로 만드는 방법

다음 절차에서는 94 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"의 단계 4에 대한 키 입력 자료를 제공합니다. partym 및 enigma의 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.

root 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

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 enigma용입니다. 다른 하나는 dst partym용입니다.
- ESP의 3DES 알고리즘에 대한 2개의 16진수 임의 숫자. 각 숫자의 길이는 168자여야 합니다. 하나는 dst enigma용입니다. 다른 하나는 dst partym용입니다.

b. 필요한 키를 생성합니다.

- 사이트에 임의 숫자 생성기가 있을 경우 생성기를 사용하십시오.
- **Oracle Solaris 11.1 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/ipseckeys
```

c. 파일의 구문을 확인합니다.

```
# ipseckey -c -f /etc/inet/secret/ipseckeys
```

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.

시작하기 전에 역할을 만들고 지정하려면 `root` 역할이 있어야 합니다. 일반 사용자는 사용 가능한 권한 프로파일 내용을 표시하여 확인할 수 있습니다.

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 11.1 관리: 보안 서비스**의 "RBAC 초기 구성(작업 맵)"을 참조하십시오.

예 7–4 역할 간 네트워크 보안 책임 구분

이 예에서는 관리자가 두 역할 간에 네트워크 보안 책임을 구분합니다. 한 역할은 Wifi 및 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 `ipsecalgs` 서비스는 사용으로 설정됩니다. 또한 기본적으로 `ike` 및 `manual-key` 서비스는 사용 안함으로 설정됩니다.

시작하기 전에 `root` 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

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/ipseckeys 파일에 항목을 추가한 후 manual-key 서비스를 사용으로 설정합니다.**

```
# svcadm enable svc:/network/ipsec/manual-key:default
```

- **ipseckeys 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_alg를 사용하여 트래픽을 보호하는 경우 AH:를 보게 됩니다.
- ESP: 접두어는 암호화된 데이터가 보내지고 있음을 나타냅니다. encr_auth_alg 또는 encr_alg를 사용하여 트래픽을 보호하는 경우 ESP:를 보게 됩니다.

시작하기 전에 연결을 테스트하려면 두 시스템에 대한 액세스 권한이 있어야 합니다.

snoop 출력을 만들려면 root 역할을 가진 사용자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

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 -----
...
```

◆ ◆ ◆ 8 장

8 IP 보안 아키텍처(참조)

이 장에는 다음 참조 정보가 포함되어 있습니다.

네트워크에서 IPsec를 구현하는 방법에 대한 지침은 7 장, “IPsec 구성(작업)”을 참조하십시오. IPsec의 개요는 6 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.conf입니다.
- 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/ipseckeys입니다.
- svc:/network/ipsec/ike 서비스 – IKE를 관리합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 구성 가능한 등록 정보는 159 페이지 “IKE 서비스”를 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"를 참조하십시오. 또한 smf(5), svcadm(1M) 및 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 역할이 있어야 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 93 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 템플리트로 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 ipsecconf(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 ipsecconf에 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. ipsecconf 명령을 사용할 때는 주의해야 합니다. 가장 안전한 작업 모드를 위해서는 ssh 또는 콘솔 또는 기타 하드 연결된 TTY를 사용합니다.

ipsecalgs 명령

암호화 프레임워크는 IPsec에 인증 및 암호화 알고리즘을 제공합니다. ipsecalgs 명령은 각 IPsec 프로토콜이 지원하는 알고리즘을 나열할 수 있습니다. ipsecalgs 구성은 /etc/inet/ipsecalgs 파일에 저장됩니다. 일반적으로 이 파일은 수정할 필요가 없습니다. 하지만 파일을 수정해야 하는 경우 ipsecalgs 명령을 사용합니다. 파일을 직접 편집하면 안됩니다. 지원되는 알고리즘은 시스템 부트 시 svc:/network/ipsec/ipsecalgs: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/ipsecalgs 파일에 있습니다. 알고리즘 및 프로토콜 번호는 IANA(Internet Assigned Numbers Authority)에 의해 정의됩니다. ipsecalgs 명령은 IPsec에 대한 알고리즘 목록을 확장할 수 있도록 합니다.

알고리즘에 대한 자세한 내용은 ipsecalgs(1M) 매뉴얼 페이지를 참조하십시오. 암호화 프레임워크에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 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 10 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/ipseckeys 파일에 추가한 다음 svc:/network/ipsec/manual-key:default 서비스를 사용으로 설정합니다. 절차는 106 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 ipseckey로 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 111 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

이 릴리스에 포함된 무료 오픈 소스 소프트웨어인 Wireshark (http://www.wireshark.org/about.html) 등의 타사 네트워크 분석기도 사용할 수 있습니다.

◆ ◆ ◆ 9 장

9 Internet Key Exchange(개요)

IKE(Internet Key Exchange)는 IPsec의 키 관리를 자동화합니다. Oracle Solaris는 IKEv1을 구현합니다. 이 장은 IKE에 대한 다음 정보를 포함합니다.

IKE 구현 지침은 10 장, "IKE 구성(작업)"을 참조하십시오. 참고 사항은 11 장, "Internet Key Exchange(참조)"를 참조하십시오. IPsec에 대한 내용은 6 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 9–1 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. **인증된 키 교환**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 그룹은 PFS를 협상하는 데 사용됩니다. The Internet Key Exchange (IKE) (`http://www.faqs.org/rfcs/rfc2409.html`)의 6절을 참조하십시오.

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(인증 기관(CA)) 조직에서 PKI에 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.conf에 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. preshared가 아닌 auth_method의 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 개 이상의 피어 시스템을 인증하는 데 사용됩니다. 미리 공유한 키는 한 시스템에서 관리자가 만든 16진수 또는 ASCII 문자열입니다. 그런 다음 대역 외 연결에서 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 저장됩니다.

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 11.1에서 서비스 및 결함 관리**의 1 장, “서비스 관리(개요)”를 참조하십시오.

표 9–2 IKE 구성 파일, 키 저장소 위치, 명령 및 서비스

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
svc:/network/ipsec/ike	IKE를 관리하는 SMF 서비스입니다.	smf(5)

표 9–2 IKE 구성 파일, 키 저장소 위치, 명령 및 서비스 (계속)

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
/usr/lib/inet/in.iked	IKE(Internet Key Exchange) 데몬입니다. ike 서비스를 사용으로 설정할 때 자동화된 키 관리를 활성화합니다.	in.iked(1M)
/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.	ikeadm(1M)
/usr/sbin/ikecert	공개 키 인증서를 보유하는 로컬 데이터베이스를 조작하기 위한 인증서 데이터베이스 관리 명령입니다. 데이터베이스를 연결된 하드웨어에 저장할 수도 있습니다.	ikecert(1M)
/etc/inet/ike/config	IKE 정책의 기본 구성 파일입니다. 인바운드 IKE 요청을 일치시키고 아웃바운드 IKE 요청을 준비하기 위한 사이트 규칙을 포함합니다. 이 파일이 존재하면 ike 서비스를 사용으로 설정할 때 in.iked 데몬을 시작합니다. 이 파일의 위치는 svccfg 명령으로 변경할 수 있습니다.	ike.config(4)
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.	ike.preshared(4)
ike.privatekeys	/etc/inet/secret 디렉토리의 개인 키 디렉토리입니다. 공개-개인 키 쌍의 일부인 개인 키를 포함합니다.	ikecert(1M)
publickeys 디렉토리	공개 키 및 인증서 파일을 보유하는 /etc/inet/ike 디렉토리 안의 디렉토리입니다. 공개-개인 키 쌍 중 공개 키 부분을 포함합니다.	ikecert(1M)
crls 디렉토리	공개 키 및 인증서 파일에 대한 해지 목록을 보유하는 /etc/inet/ike 디렉토리 안의 디렉토리입니다.	ikecert(1M)
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.	ikecert(1M)

◆ ◆ ◆ 10 장

10 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.

- 125 페이지 "IKE 정보 표시"
- 127 페이지 "IKE 구성(작업 맵)"
- 127 페이지 "미리 공유한 키로 IKE 구성(작업 맵)"
- 132 페이지 "공개 키 인증서로 IKE 구성(작업 맵)"
- 149 페이지 "모바일 시스템에 대한 IKE 구성(작업 맵)"
- 156 페이지 "연결된 하드웨어를 찾도록 IKE 구성"

IKE에 대한 개요 정보는 9 장, "Internet Key Exchange(개요)"를 참조하십시오. IKE에 대한 참조 정보는 11 장, "Internet Key Exchange(참조)"를 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.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

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.

```
p1_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.

```
p1_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.

```
p1_xform
  { auth_method preshared oakley_group 15 auth_alg sha256 encr_alg aes }
```

참조 이러한 값이 필요한 IKE 규칙을 구성하는 작업은 127 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.	127 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.	132 페이지 "공개 키 인증서로 IKE 구성(작업 맵)"
NAT 경계를 벗어납니다.	모바일 시스템과 통신하도록 IPsec 및 IKE를 구성합니다.	149 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.	156 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.	128 페이지 "미리 공유한 키로 IKE를 구성하는 방법"

작업	설명	수행 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.	131 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 관리**의 "다중 레벨 Trusted Extensions 네트워크에서 IPsec 보호를 적용하는 방법"을 참조하십시오.

시작하기 전에 `solaris.admin.edit/etc/inet/ike/config` 권한 부여 외에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1. **각 시스템에서 `/etc/inet/ike/config` 파일을 만듭니다.**

   `/etc/inet/ike/config.sample`을 템플리트로 사용할 수 있습니다.

2 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 94 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
}
```

3 각 시스템에서 파일의 구문을 확인합니다.

```
# /usr/lib/inet/in.iked -c -f /etc/inet/ike/config
```

4 **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 tekniques"
}
```

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"
    }
```

5 **IKE 서비스를 사용으로 설정합니다.**

```
# svcadm enable ipsec/ike
```

예 10–1 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.

`solaris.admin.edit/etc/inet/ike/config` 권한 부여 외에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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
 p1_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
 p1_xform
 {auth_method preshared oakley_group 5 auth_alg sha256 encr_alg aes}
 p2_pfs 5
}
```

2 **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"
}
```

3 **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. ■ 자체 서명된 인증서 ■ 피어 시스템의 공개 키 인증서	133 페이지 "자체 서명된 공개 키 인증서로 IKE를 구성하는 방법"
PKI 인증 기관으로 IKE를 구성합니다.	인증서 요청을 만들고 각 시스템에 다음 세 개의 인증서를 배치합니다. ■ 인증 기관(CA)이 요청에 따라 만든 인증서 ■ CA의 공개 키 인증서 ■ CA의 CRL	138 페이지 "CA가 서명한 인증서로 IKE를 구성하는 방법"
로컬 하드웨어에서 공개 키 인증서를 구성합니다.	다음 작업 중 하나를 수행합니다. ■ 로컬 하드웨어에서 자체 서명된 인증서를 생성한 다음 원격 시스템의 공개 키를 하드웨어에 추가합니다. ■ 로컬 하드웨어에서 인증서 요청을 생성한 다음 CA의 공개 키 인증서를 하드웨어에 추가합니다.	143 페이지 "공개 키 인증서를 생성하여 하드웨어에 저장하는 방법"
PKI에서 인증서 해지 목록(CRL)을 업데이트합니다.	중앙 배포 지점에서 CRL에 액세스합니다.	147 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 156 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.

시작하기 전에 `solaris.admin.edit/etc/inet/ike/config` 권한 부여 외에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 `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` *keysize*	키의 크기입니다. *keysize*는 512, 1024, 2048, 3072 또는 4096일 수 있습니다.
`-t` *keytype*	사용할 알고리즘의 유형을 지정합니다. *keytype*은 `rsa-sha1`, `rsa-md5` 또는 `dsa-sha1`일 수 있습니다.
`-D` *dname*	인증서 주체에 대한 X.509 식별 이름입니다. 일반적으로 *dname*의 형식은 C=*country* , O=*organization*, OU=*organizational unit*, CN=*common name*입니다. 유효한 태그는 C, O, OU 및 CN입니다.
`-A` *altname*	인증서의 대체 이름입니다. *altname*의 형식은 `tag=value`입니다. 유효한 태그는 `IP`, `DNS`, `email` 및 `DN`입니다.
`-S` *validity-start-time*	인증서 시작 시간을 유효한 절대 또는 상대 시작 시간으로 지정합니다.
`-F` *validity-end-time*	인증서 종료 시간을 유효한 절대 또는 상대 종료 시간으로 지정합니다.
`-T` *token-ID*	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-----
MIIC1TCCAb2gAwIBAgIEBl5JnjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
...
y85m6LHJYtC6
-----END X509 CERTIFICATE-----
```

2 인증서를 저장하여 원격 시스템으로 보냅니다.

출력은 인증서 공개 부분의 인코딩된 버전입니다. 이 인증서는 전자 메일에 안전하게 첨부할 수 있습니다. 수신자는 단계 4와 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-----
```

3 각 시스템에서 공개 키 데이터베이스에 수신한 인증서를 추가합니다.

a. root가 읽을 수 있는 파일에 관리자의 전자 메일을 저장합니다.

b. ikecert 명령에 파일을 재지정합니다.

```
# ikecert certdb -a < /tmp/certificate.eml
```

이 명령은 BEGIN 태그와 END 태그 사이의 텍스트를 가져옵니다.

4 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. partym에 저장된 인증서를 나열합니다.

다음 예에서 Note 1은 슬롯 0에 있는 인증서의 식별 이름(DN)을 나타냅니다. 슬롯 0에 있는 개인 인증서가 동일한 해시(주 3 참조)를 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 # 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=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: FEA65C5387BBF3B2C8F16C019FEBC388
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: FEA65C5387BBF3B2C8F16C019FEBC388

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: FEA65C5387BBF3B2C8F16C019FEBC388

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 인증서와 일치합니다.

5 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
...
```

6 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.

시작하기 전에 `solaris.admin.edit/etc/inet/ike/config` 권한 부여 외에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 ikecert certlocal -kc 명령을 사용하여 인증서 요청을 만듭니다.

명령 인수에 대한 설명은 133 페이지 “자체 서명된 공개 키 인증서로 IKE를 구성하는 방법”의 단계 1을 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-----
MIIByjCCATMCAQAwUzELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zckO80mO9X
-----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
...
8qlqdjaStLGfhDOO
-----END CERTIFICATE REQUEST-----
```

2 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 147 페이지 "인증서 해지 목록 처리 방법"을 참조하십시오.

3 시스템에 각 인증서를 추가합니다.

`ikecert certdb -a`에 대한 `-a` 옵션은 붙여 넣은 객체를 시스템의 적합한 인증서 데이터베이스에 추가합니다. 자세한 내용은 122 페이지 "IKE와 공개 키 인증서"를 참조하십시오.

a. 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 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 `certrldb` 데이터베이스에 CRL을 추가합니다.

```
# ikecert certrldb -a
     Press the Return key
     Paste the CRL:
-----BEGIN CRL-----
...
-----END CRL----
     Press the Return key
<Control>-D
```

4 `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-partym - 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
...
```

5 **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 147 페이지 “인증서 해지 목록 처리 방법”을 참조하십시오.

예 10–2 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-----
MII
...
-----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 156 페이지 "Sun Crypto Accelerator 6000 보드를 찾도록 IKE를 구성하는 방법"을 참조하십시오.

`solaris.admin.edit/etc/inet/ike/config` 권한 부여 외에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 ssh 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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.**

```
# ikecert 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.**

```
# ikecert 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
```

ikecert 명령 인수에 대한 설명은 ikecert(1M) 매뉴얼 페이지를 참조하십시오.

2 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
...
oKUDBbZ9O/pLWYGr
-----END X509 CERTIFICATE-----
```

3 **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 138 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”의 단계 2를 참조하십시오.

4 **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-partym"
 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

 p1_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-partym - 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

 p1_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha256 encr_alg aes}
}
```

5 **하드웨어에서 상대방의 인증서를 배치합니다.**

단계 2에서 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 147 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 138 페이지 "CA가 서명한 인증서로 IKE를 구성하는 방법"의 단계 5에서 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 certrldb` 명령에 대한 인수로 CRL을 제공할 수 있습니다. 예는 예 10–3을 참조하십시오.

다음 절차에서는 중앙 배포 지점의 CRL을 사용하도록 IKE에 알리는 방법에 대해 설명합니다.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

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 10–3 로컬 certrldb 데이터베이스에 CRL 붙여넣기

중앙 배포 지점에서 PKI 조직의 CRL을 사용할 수 없을 경우 수동으로 로컬 `certrldb` 데이터베이스에 CRL을 추가할 수 있습니다. PKI 조직의 지침에 따라 CRL을 파일에 추출한 다음 `ikecert certrldb -a` 명령을 사용하여 데이터베이스에 CRL을 추가합니다.

```
# ikecert certrldb -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.	149 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 10–4
고정 IP 주소가 없는 시스템에서 CA의 공개 인증서 및 IKE를 사용합니다.	회사 본사 등의 중앙 사이트에 대한 트래픽을 보호하도록 모바일 시스템을 구성합니다.	예 10–5
모바일 시스템의 트래픽을 승인하는 중앙 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	모바일 시스템의 IPsec 트래픽을 승인하도록 자체 서명된 인증서로 게이트웨이 시스템을 구성합니다.	예 10–6
고정 IP 주소가 없는 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	중앙 사이트에 대한 트래픽을 보호하도록 자체 서명된 인증서로 모바일 시스템을 구성합니다.	예 10–7

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.

시작하기 전에 `root` 역할이 있어야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오. 원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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 }
}
```

2 각 모바일 시스템에 로그인하고 중앙 시스템을 찾도록 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 }
}
```

3 ike 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/ike
```

예 10–4 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. `main1`은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 예 10–5를 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 10–5 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 10–4를 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 10–6 모바일 시스템의 자체 서명된 인증서 승인

다음 예에서는 자체 서명된 인증서가 발급되었으며 모바일 및 중앙 시스템에 배치됩니다. `main1`은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 예 10–7을 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 10–7 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 10–6을 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 사용자 설명서** (`http://download.oracle.com/docs/cd/E19321-01/820-4144-12/820-4144-12.pdf`)를 참조하십시오.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

원격으로 로그인할 경우 안전한 원격 로그인을 위해 `ssh` 명령을 사용합니다. 예는 예 7–1을 참조하십시오.

1 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
$
```

2 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 이름이라고도 함)를 반환합니다. 이 예에서는 `ikecert` 명령에 `Sun Metaslot` 토큰을 사용하여 IKE 키를 저장하고 속도를 향상시킬 수 있습니다.

토큰 사용 방법에 대한 지침은 143 페이지 "공개 키 인증서를 생성하여 하드웨어에 저장하는 방법"을 참조하십시오.

`ikecert` 명령을 통해 자동으로 후행 공백이 채워집니다.

예 10–8 Metaslot 토큰 찾기 및 사용

토큰은 디스크, 연결된 보드 또는 암호화 프레임워크가 제공하는 소프트 토큰 키 저장소에 저장할 수 있습니다. 소프트 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 문장암호를 만들려면 pktool(1) 매뉴얼 페이지를 참조하십시오.

다음과 유사한 명령이 소프트 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.

◆ ◆ ◆ 11 장

11 Internet Key Exchange(참조)

이 장은 IKE에 대한 다음 참조 정보를 포함합니다.

IKE 구현 지침은 10 장, “IKE 구성(작업)”을 참조하십시오. 개요 정보는 9 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/config를 만들어야 합니다.

다음 ike 서비스 등록 정보를 구성할 수 있습니다.

- config_file **등록 정보** – IKE 구성 파일의 위치입니다. 초기 값은 /etc/inet/ike/config입니다.
- debug_level **등록 정보** – in.iked 데몬의 디버깅 레벨입니다. 초기 값은 op 또는 operational입니다. 가능한 값은 ikeadm(1M) 매뉴얼 페이지에서 **객체 유형** 아래의 디버그 레벨 테이블을 참조하십시오.
- admin_privilege **등록 정보** – in.iked 데몬의 권한 레벨입니다. 초기 값은 base입니다. 다른 값으로 modkeys 및 keymat가 있습니다. 세부 정보는 161 페이지 “ikeadm 명령”을 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"를 참조하십시오. 또한 `smf(5)`, `svcadm(1M)` 및 `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 110 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 127 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.conf의 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 115 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 110 페이지 “IPsec 및 IKE 서비스를 관리하는 방법”을 참조하십시오.

ikeadm 명령에 대한 보안 고려 사항은 ipseckey 명령의 고려 사항과 비슷합니다. 세부 정보는 117 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. ipseckeys 파일은 IPsec SA에 대한 미리 공유한 키를 포함합니다. 파일은 0600에서 보호됩니다. secret 디렉토리는 0700에서 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.
- ipseckeys 파일은 IPsec SA에 대한 키 관련 자료를 포함합니다. 파일 수동 관리의 예는 106 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, certrldb가 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 metaslot을 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 11–1 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 -t rsa-sha1	auth_method rsa_sig	DSA보다 약간 빠른 인증 방법입니다. 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 -t rsa-sha1	auth_method rsa_encrypt	RSA 암호화는 도청자로부터 IKE의 신원을 숨기지만 IKE 피어가 서로의 공개 키를 알아야 합니다.

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 certrldb 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 133 페이지 “자체 서명된 공개 키 인증서로 IKE를 구성하는 방법”을

참조하십시오. 또한 이 명령은 PKI 또는 CA로부터 받은 인증서를 입력으로 받아들입니다. 절차는 138 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 certrldb 명령

certrldb 하위 명령은 CRL(인증서 해지 목록) 데이터베이스인 /etc/inet/ike/crls를 관리합니다. CRL 데이터베이스는 공개 키에 대한 해지 목록을 유지 관리합니다. 이 목록에는 더 이상 유효하지 않은 인증서가 있습니다. PKI에서 CRL을 제공할 때 ikecert certrldb 명령을 사용하여 CRL 데이터베이스에 CRL을 설치할 수 있습니다. 절차는 147 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 **슬롯**에 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 서명된 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 서명된 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 certrldb 명령을 사용하여 데이터베이스를 채울 수 있습니다.

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. 1976년 Diffie와 Hellman이 개발한 비대칭 암호화 키 계약 프로토콜입니다. 이 프로토콜을 사용하면 두 사용자가 사전 보안 없이 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 *An Architecture for Differentiated Services*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.

DSA 디지털 서명 알고리즘(Digital Signature Algorithm)입니다. 512-4096비트의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 1024비트까지 지원합니다. DSA는 입력에 SHA-1을 사용합니다.

ESP (보안 페이로드 캡슐화) 데이터그램에 무결성 및 기밀성을 제공하는 확장 헤더입니다. ESP는 IP 보안 구조(IPsec)의 5개 구성 요소 중 하나입니다.

HMAC 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.

ICMP 인터넷 제어 메시지 프로토콜(Internet Control Message Protocol). 오류를 처리하고 제어 메시지를 교환하는 데 사용됩니다.

ICMP 에코 요청 패킷 인터넷에서 응답을 간청하기 위해 시스템으로 보낸 패킷입니다. 이러한 패킷을 흔히 "ping" 패킷이라고 합니다.

IKE 인터넷 키 교환(Internet Key Exchange). IKE는 IPsec 보안 연관(SA)에 대한 인증된 키 관련 자료의 프로비전을 자동화합니다.

IP IP(인터넷 프로토콜), IPv4, IPv6을 참조하십시오.

IP-in-IP 캡슐화 IP 패킷 안에 IP 패킷을 터널링하는 방식입니다.

IP 데이터그램 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 재컴파일할 수 있습니다.

IP 링크 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.

IP 스택 TCP/IP를 종종 "스택"이라고도 합니다. 이것은 데이터 교환의 클라이언트측과 서버측 양쪽에서 모든 데이터가 전달되는 계층(TCP, IP 및 기타)을 가리킵니다.

IP (인터넷 프로토콜) 인터넷을 통해 한 컴퓨터에서 다른 컴퓨터로 데이터를 보내는 방법 또는 규약입니다.

IP 헤더 인터넷 패킷을 고유하게 식별하는 20바이트의 데이터입니다. 헤더는 패킷의 소스 및 대상 주소를 포함합니다. 헤더 내에는 바이트를 더 추가할 수 있는 옵션이 존재합니다.

IPQoS diffserv 모델 표준 구현과 더불어, 가상 LAN에 대한 플로우 계산 및 802.1 D 표시를 제공하는 소프트웨어 기능입니다. IPQoS를 사용하면 IPQoS 구성 파일에 정의된 대로 여러 레벨의 네트워크 서비스를 고객 및 응용 프로그램에 제공할 수 있습니다.

IPsec IP 보안. IP 데이터그램에 대한 보호를 제공하는 보안 구조입니다.

IPv4 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 네트워크 주소 변환을 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. Diffie-Hellman 알고리즘도 참조하십시오.

PHB (홉별 동작) 트래픽 클래스에 지정된 우선 순위입니다. PHB는 다른 트래픽 클래스와 비교하여 해당 클래스의 어떤 플로우가 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 SA(보안 연관)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)입니다. 이 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 SPD(보안 정책 데이터베이스)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 모니터하여 얻은 정보를 바탕으로 네트워크 패킷이 방화벽을 통과할지 여부를 확인할 수 있는 패킷 필터입니다. 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. **내부** 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 Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`)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 한가지 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.

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. Diffie-Hellman 알고리즘은 비대칭 키 프로토콜의 예입니다. 대칭 키 암호화와 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 filter 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 찾은 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 브로드캐스트 주소 또는 다중 브로드캐스트 주소로 지정된 ICMP 에코 요청 패킷을 사용하여 심각한 네트워크 혼잡 또는 정전을 일으킵니다.

스택 IP 스택을 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 **가장 가까운** 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 홈 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 플로우의 비율을 측정하는 diffserv 구조의 모듈입니다. IPQoS 구현에는 `tokenmt` 및 `tswtclmt`의 두 측정자가 포함됩니다.

캡슐화 헤더 및 페이로드를 첫번째 패킷에 넣고, 이어서 두번째 패킷의 페이로드에 넣는 프로세스입니다.

클래스 IPQoS에서 비슷한 특성을 공유하는 네트워크 플로우 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 1. 패킷의 전달 방법을 나타내는 값으로 IP 패킷의 DS 필드를 표시하는 diffserv 구조 및 IPQoS의 모듈입니다. IPQoS 구현에서 표시자 모듈은 `dscpmk`입니다.

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 `dlcosmk`라고 합니다.

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.

플로우 계산 IPQoS에서 트래픽 플로우에 대한 정보를 누적하고 기록하는 프로세스입니다. IPQoS 구성 파일에 `flowacct` 모듈의 매개변수를 정의하여 플로우 계산을 설정합니다.

필터 IPQoS 구성 파일에 클래스의 특성의 정의하는 규칙 세트입니다. IPQoS 시스템이 IPQoS 구성 파일에서 필터를 준수하는 트래픽 플로우를 처리하기 위해 선택합니다. 패킷 필터를 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 제어 전송 프로토콜 TCP와 비슷한 방법으로 연결 지향적 통신을 제공하는 전송 계층 프로토콜입니다. 추가적으로, SCTP는 멀티홈 기능을 지원하므로 연결 끝점 중 하나가 여러 개의 IP 주소를 가질 수 있습니다.

색인

E

F

H

I

T

U

V

W

X

개

계

공

구

확

활