Oracle® Solaris 11.2의 네트워크 보안

부품 번호: E53812-02
2014년 8월

ORACLE®

목차

표

코드 예

이 설명서 사용

- **개요** - 네트워크 보안을 제공하는 방법을 설명합니다. 웹 서버에 대한 링크 보호, 조정 가능한 네트워크 매개변수, 방화벽 보호, IPsec 및 IKE, SSL 커널 보호 등을 포함합니다.
- **대상** - 네트워크 보안 관리자
- **필요한 지식** - 사이트 보안 요구 사항

제품 설명서 라이브러리

이 제품에 대한 최신 정보 및 알려진 문제는 설명서 라이브러리(http://www.oracle.com/pls/topic/lookup?ctx=E56343)에서 확인할 수 있습니다.

Oracle 지원 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info를 참조하거나, 청각 장애가 있는 경우 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs를 방문하십시오.

피드백

http://www.oracle.com/goto/docfeedback에서 이 설명서에 대한 피드백을 보낼 수 있습니다.

♦ ♦ ♦ 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 항목을 다룹니다.

Oracle Solaris 11.2에서 네트워크 보안의 새로운 기능

이 절에서는 기존 고객을 위해 이 릴리스의 중요한 새 네트워크 보안 기능에 대해 주로 설명합니다.

IKE 버전 2(IKEv2)에서는 최신 버전의 IKE 프로토콜을 사용하여 IPsec에 대한 자동 키 관리 기능을 제공합니다. IKEv2 및 IPsec에서는 Oracle Solaris 암호화 프레임워크 기능의 암호화 알고리즘을 사용합니다.

참고 - Oracle Solaris의 암호화 프레임워크 기능은 FIPS 140-2, 레벨 1에 대해 검증되었습니다. IKE에서 FIPS 140 모드를 사용하려면 표 8-1. “Oracle Solaris에서 IKEv2 및 IKEv1 구현”을 참조하십시오. 하드웨어 및 소프트웨어 세부 사항은 Oracle FIPS 140 Software Validations (http://www.oracle.com/technetwork/topics/security/fips140-software-validations-1703049.html)를 참조하십시오.

IKE 버전 1(IKEv1) 지원도 계속 제공됩니다. 자세한 내용은 8장. IKE(Internet Key Exchange)를 참조하십시오.

링크 보호 정보

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 VM이 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

참고 - 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 형식을 지정해야 합니다.

`restricted`

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

참고 - 링크 보호로 인해 삭제되는 패킷은 `mac_spoofed`, `dhcp_spoofed`, `ip_spoofed` 및 `restricted`의 네 가지 보호 유형에 대한 커널 통계를 통해 추적합니다. 이러한 링크별 통계를 검색하려면 링크 보호 구성 및 통계를 확인하는 방법 [20]을 참조하십시오.

이러한 보호 유형에 대한 자세한 설명은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

링크 보호 구성

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

참고 - 전역 영역에 링크 보호를 구성해야 합니다.

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

표 1-1 링크 보호 구성 작업 맵

작업	설명	지침
링크 보호를 사용으로 설정합니다.	링크에서 보내는 패킷을 제한하고 스푸핑으로부터 링크를 보호합니다.	링크 보호를 사용으로 설정하는 방법 [17]
링크 보호를 사용 안함으로 설정합니다.	링크 보호를 제거합니다.	링크 보호를 사용 안함으로 설정하는 방법 [18]
IP 링크 보호 유형을 지정합니다.	링크 보호 방식을 통과할 수 있는 IP 주소를 지정합니다.	IP 스푸핑으로부터 보호할 IP 주소를 지정하는 방법 [19]
DHCP 링크 보호 유형을 지정합니다.	링크 보호 방식을 통과할 수 있는 DHCP 주소를 지정합니다.	DHCP 스푸핑으로부터 보호할 DHCP 클라이언트를 지정하는 방법 [19]
링크 보호 구성을 확인합니다.	보호되는 링크와 예외를 나열하고 적용 통계를 표시합니다.	링크 보호 구성 및 통계를 확인하는 방법 [20]

▼ 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **사용 가능한 링크 보호 유형을 확인합니다.**

```
# dladm show-linkprop -p protection
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     protection      rw   --           --           --        mac-nospoof,
```

```
                                                    restricted,
                                                    ip-nospoof,
                                                    dhcp-nospoof
```

가능한 유형에 대한 설명은 “링크 보호 유형” [16]과 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 개 링크 보호 유형을 모두 사용으로 설정되어 있습니다.

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

주의 - 각 보호 값을 사용으로 설정하기 전에 하나씩 테스트합니다. 시스템을 잘못 구성하면 통신하지 못할 수 있습니다.

3. **링크 보호가 사용으로 설정되어 있는지 확인합니다.**

```
# dladm show-linkprop -p protection vnic0
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     protection      rw   mac-nospoof  mac-nospoof  --        mac-nospoof,
                              restricted   restricted   --        restricted,
                              ip-nospoof   ip-nospoof   --        ip-nospoof,
                              dhcp-nospoof dhcp-nospoof --        dhcp-nospoof
```

▼ 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 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”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        EFFECTIVE    DEFAULT   POSSIBLE
net0     protection      rw   --           --           --        mac-nospoof,
                                                                  restricted,
                                                                  ip-nospoof,
                                                                  dhcp-nospoof
```

▼ IP 스푸핑으로부터 보호할 IP 주소를 지정하는 방법

시작하기 전에 `How to Enable Link Protection`에 나온 대로 링크 보호를 사용으로 설정하는 방법 [17] 보호 유형이 사용으로 설정되어 있습니다.

Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **IP 스푸핑으로부터 보호하도록 설정했는지 확인합니다.**

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

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 `10.0.0.2`를 추가하는 방법을 보여줍니다.

```
# 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 `How to Enable Link Protection`에 나온 대로 링크 보호를 사용으로 설정하는 방법 [17] 보호 유형이 사용으로 설정되어 있습니다.

Network Link Security 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **DHCP 스푸핑으로부터 보호하도록 설정했는지 확인합니다.**

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

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 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 보여줍니다.

```
# dladm show-linkprop -p protection,allowed-ips,allowed-dhcp-cids vnic0
LINK    PROPERTY            PERM  VALUE         EFFECTIVE    DEFAULT POSSIBLE
vnic0   protection          rw    mac-nospoof   mac-nospoof  --      mac-nospoof,
                                  restricted    restricted           restricted,
                                  ip-nospoof    ip-nospoof           ip-nospoof,
                                  dhcp-nospoof  dhcp-nospoof         dhcp-nospoof
vnic0   allowed-ips         rw    10.0.0.1,     10.0.0.1,    --      --
                                  10.0.0.2      10.0.0.2
vnic0   allowed-dhcp-cids   rw    hello         hello        --      --
```

참고 - allowed-ips 등록 정보는 EFFECTIVE 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 표시합니다.**

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

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

네트워크 조정

표 2-1 네트워크 조정 작업 맵

작업	설명	지침
네트워크 경로 지정 데몬을 사용 안함으로 설정합니다.	잠재적인 네트워크 스니퍼에 의한 시스템 액세스를 제한합니다.	네트워크 경로 지정 데몬을 사용 안함으로 설정하는 방법 [24]
네트워크 토폴로지 정보에 대한 배포를 방지합니다.	패킷 브로드캐스트를 방지합니다.	브로드캐스트 패킷 전달을 사용 안함으로 설정하는 방법 [24]
	브로드캐스트 에코 요청 및 멀티캐스트 에코 요청에 대한 응답을 방지합니다.	에코 요청에 대한 응답을 사용 안함으로 설정하는 방법 [25]
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.	엄격한 다중 홈 지정을 설정하는 방법 [26]
완전하지 않은 시스템 연결 개수를 제한하여 DOS 공격을 방지합니다.	TCP 리스너에 대해 완전하지 않은 TCP 연결의 허용 가능한 개수를 제한합니다.	완전하지 않은 TCP 연결의 최대 개수를 설정하는 방법 [26]
허용된 수신 연결 개수를 제한하여 DOS 공격을 방지합니다.	TCP 리스너에 대한 보류 중인 TCP 연결의 기본 최대 개수를 지정합니다.	보류 중인 TCP 연결의 최대 개수를 설정하는 방법 [27]
초기 TCP 연결에 대해 강력한 난수가 생성되었는지 확인합니다.	RFC 6528에 의해 지정된 시퀀스 번호 생성 값을 준수합니다.	초기 TCP 연결에 대한 높은 수준의 난수를 지정하는 방법 [27]
ICMP 재지정을 방지합니다.	네트워크 토폴로지의 표시기를 제거합니다.	ICMP 재지정을 방지하는 방법 [28]
네트워크 매개변수를 해당 보안 기본값으로 반환합니다.	관리 작업으로 줄어든 보안을 늘립니다.	네트워크 매개변수를 보안 값으로 재설정하는 방법 [29]

▼ 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.

참고 - 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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, 2014 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, 2014 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.

참고 - `_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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2 조정 가능 매개변수 참조 설명서 "의 "_respond_to_echo_broadcast 및 _respond_to_echo_multicast(ipv4 또는 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 멀티홈 시스템에 대한 IP 패킷의 전송 및 수신 동작을 제어합니다.

시작하기 전에 Network Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2 조정 가능 매개변수 참조 설명서 "의 "hostmodel(ipv4 또는 ipv6)" 및 `ipadm`(1M) 매뉴얼 페이지를 참조하십시오.

엄격한 멀티홈 사용에 대한 자세한 내용은 터널 모드에서 IPsec을 사용하여 두 LAN 사이의 연결을 보호하는 방법 [109]을 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2 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 TCP 초기 시퀀스 번호 생성 매개변수가 RFC 6528 (http://www.ietf.org/rfc/rfc6528.txt)을 준수하는지 확인합니다.

시작하기 전에 `solaris.admin.edit/etc.default/inetinit` 권한 부여가 지정된 관리자여야 합니다. 기본적으로 `root` 역할에는 이 권한 부여가 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **`TCP_STRONG_ISS` 변수에 대한 기본값이 2인지 확인합니다.**

```
# grep TCP_STRONG /etc/default/inetinit
```

```
# TCP_STRONG_ISS sets the TCP initial sequence number generation parameters.
# Set TCP_STRONG_ISS to be:
TCP_STRONG_ISS=2
```

2. **TCP_STRONG_ISS가 2가 아닌 경우 2로 변경합니다.**

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

3. **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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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=off ipv4
# ipadm set-prop -p send_redirects=off ipv6
# ipadm show-prop -p send_redirects ipv4
PROTO PROPERTY          PERM CURRENT  PERSISTENT   DEFAULT  POSSIBLE
ipv4  send_redirects    rw   off      off          on       on,off

# ipadm show-prop -p send_redirects ipv6
PROTO  PROPERTY        PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
ipv6  send_redirects   rw   off       off          on       on,off
```

자세한 내용은 “Oracle Solaris 11.2 조정 가능 매개변수 참조 설명서 ”의 “send_redirects(ipv4 또는 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 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”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 11.2 조정 가능 매개변수 참조 설명서 ”의 “forwarding(ipv4 또는 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.

   참고 - 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 11.2 조정 가능 매개변수 참조 설명서 "의 "_rev_src_routes"를 참조하십시오.

참조 ipadm(1M) 매뉴얼 페이지

♦ ♦ ♦ 3 장

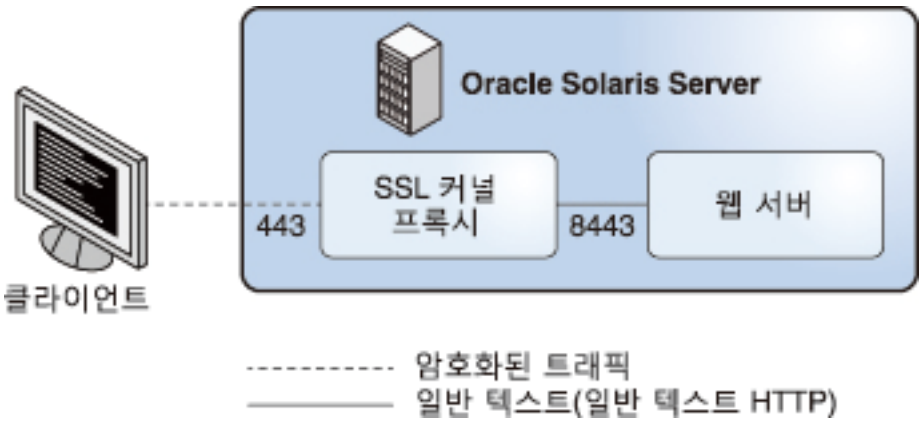
3 웹 서버 및 Secure Sockets Layer 프로토콜

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 보여줍니다.

그림 3-1 커널로 암호화된 웹 서버 통신



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

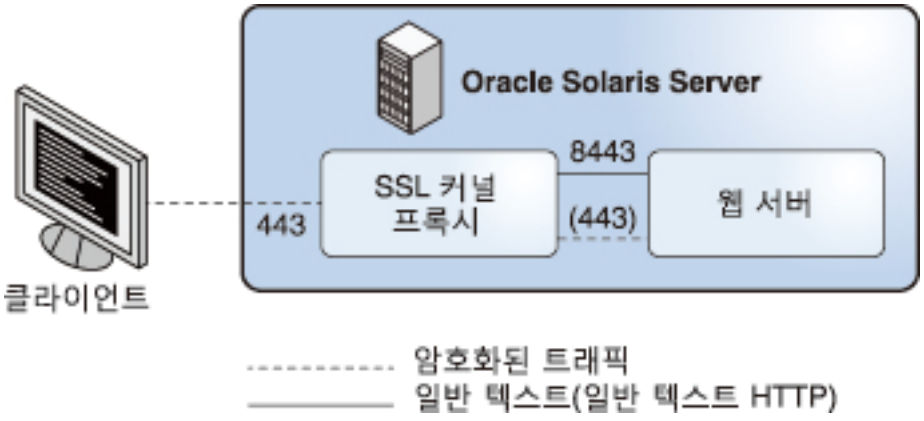
- 프록시는 웹 서버 같은 서버 응용 프로그램의 SSL 성능 속도를 향상시키므로 사용자 레벨의 SSL 라이브러리를 사용하는 응용 프로그램보다 더 나은 성능을 제공합니다. 성능 향상은 응용 프로그램의 작업 부하에 따라 35% 이상이 될 수 있습니다.
- SSL 커널 프록시는 투명합니다. 지정된 IP 주소가 없으므로 웹 서버에 실제 클라이언트 IP 주소와 TCP 포트가 표시됩니다.
- SSL 커널 프록시 및 웹 서버는 함께 작동하도록 설계되었습니다.

  그림 3-1. "커널로 암호화된 웹 서버 통신"은 SSL 커널 프록시를 사용하는 웹 서버가 있는 기본 시나리오를 보여줍니다. SSL 커널 프록시는 443 포트에 구성되어 있는 반면 웹 서버는 8443 포트에 구성되어 있으며, 이 포트에서 암호화 해제된 HTTP 통신을 수신합니다.
- SSL 커널 프록시는 요청된 암호화를 지원하지 않는 경우 사용자 레벨의 암호로 폴백하도록 구성할 수 있습니다.

  그림 3-2. "사용자 레벨의 폴백 옵션이 있는 커널로 암호화된 웹 서버 통신"는 더 복잡한 시나리오를 보여줍니다. 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

다음 절차에서는 SSL 커널 프록시를 사용하도록 웹 서버를 구성하는 방법을 보여줍니다.

▼ 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 3-1. “커널로 암호화된 웹 서버 통신”에서 보여주는 간단한 시나리오를 구현합니다.

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

`root` 역할을 맡아야 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. 제공해야 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 토큰 인증을 받습니다. `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 PKCS11-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 3-1. “커널로 암호화된 웹 서버 통신”에서 보여주는 간단한 시나리오를 구현합니다.

시작하기 전에 Oracle iPlanet 웹 서버를 설치 및 구성했습니다. 서버는 Oracle iPlanet 웹 서버 (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 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”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. 제공해야 하는 매개변수 목록은 SSL 커널 프록시를 사용하도록 Apache 2.2 웹 서버를 구성하는 방법 [33]의 3 단계를 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 PKCS11-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:webserver7`이라고 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 3-2. “사용자 레벨의 폴백 옵션이 있는 커널로 암호화된 웹 서버 통신”에서 보여주는 복잡한 시나리오를 구현합니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”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
```

참고 - FIPS 140 준수를 위한 RSA 키의 최소 길이는 2048입니다. 자세한 내용은 "Using a FIPS 140 Enabled System in Oracle Solaris 11.2 "을 참조하십시오.

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
```

참고 - 문장암호에는 공백 문자를 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. **8443 포트에서 암호화되지 않은 통신을 수신하도록 웹 서버를 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.conf`를 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 허용합니다.**

   a. **`kssl.pass` 파일을 읽는 셸 스크립트를 만듭니다.**

   ```
   # pfedit /root/put-passphrase.sh
   #!/usr/bin/ksh -p
   ## Reads SSL 커널 프록시 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_handshakes`에서 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_connections에서 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. 전역 영역에서 ksslcfg 명령을 사용하여 서비스 인스턴스를 구성한 후 비전역 영역에서 웹 서버를 시작할 수 있습니다.
- 인스턴스를 구성하는 경우 ksslcfg 명령과 함께 특정 호스트 이름이나 IP 주소를 지정해야 합니다. 특히 인스턴스에서 IP 주소로 INADDR_ANY를 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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-zone에 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-zone에서 웹 서비스를 사용으로 설정합니다.

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

♦ ♦ ♦ 4 장

4 Oracle Solaris의 IP 필터 정보

이 장에서는 Oracle Solaris의 IP 필터 기능에 대해 간략히 설명합니다. IP 필터 작업은 5장. IP 필터 구성을 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.

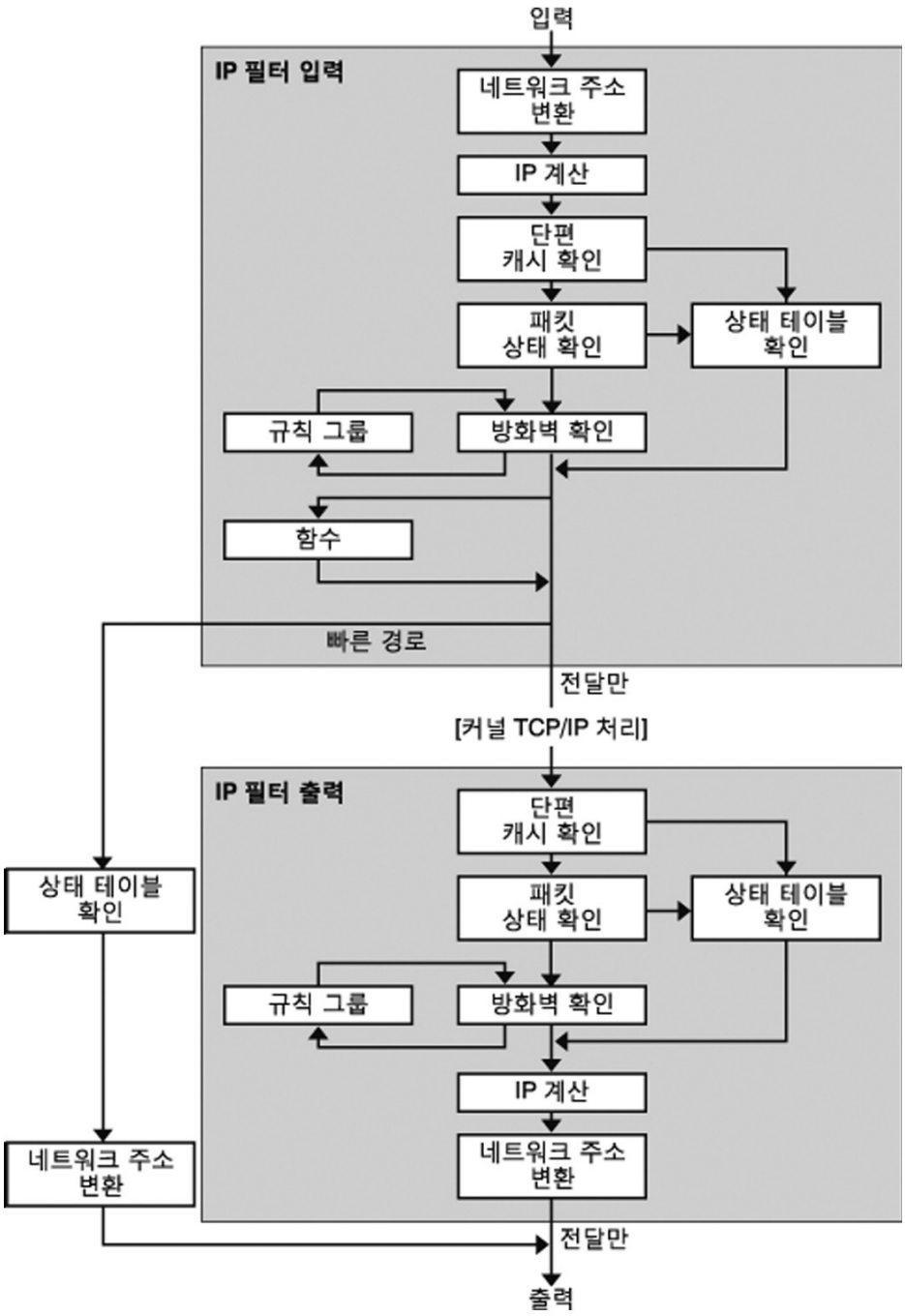
오픈 소스 IP 필터에 대한 정보 소스

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 보여줍니다.

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 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오. SMF와 연관된 단계별 절차에 대한 정보는 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 3 장, "서비스 관리"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 프로파일에서는 이 방화벽을 사용으로 설정합니다. 자동으로 네트워크에 연결되는 시스템의 사용자 정의 구성에서는 IP 필터 서비스가 사용으로 설정되지 않습니다. 서비스를 사용으로 설정하는 것과 관련된 작업은 "IP 필터 서비스 구성" [53]을 참조하십시오.
- IP 필터를 관리하려면 root 역할이거나 IP 필터 관리 권한 프로파일을 지정받아야 합니다. IP Filter Management 권한 프로파일은 만든 역할이나 사용자에 지정할 수 있습니다. 역할을 만들고 역할을 사용자에 지정하려면 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "역할 만들기"를 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 "IP 필터 구성 파일 예" [77]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 주소의 참조 그룹에 대한 주소 풀을 만들 수 있습니다. 그런 다음 나중에 규칙 세트에서 이러한 풀을 사용할 수 있습니다. 주소 풀을 사용하면 규칙 처리 속도가 증가할 수 있습니다. 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.

참고 - 규칙의 `dup-to` 옵션을 사용하면 네트워크 관리자가 네트워크 탭을 만들 수 있습니다. 이 옵션은 Oracle Solaris에서 여전히 지원되지만 대체로 덜 중요해졌습니다. 네트워크 탭을 수행할 포트를 직접 구성할 수 있는 최신 스위치가 제공되므로, 규칙에서 이 함수를 정의할 필요가 없습니다. 네트워크를 탭할 포트를 구성하는 방법은 스위치 설명서를 참조하십시오.

`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.

참고 - 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 보여줍니다. IP 주소 `192.168.0.0/16`의 수신 트래픽을 차단하려면 규칙 목록에 다음 규칙을 포함합니다.

```
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 "IP 필터에 대한 패킷 필터링 규칙 세트 관리" [60]를 참조하십시오. 예에 표시된 IP 주소 체계(`192.168.0.0/16`)에 대한 설명은 "Oracle Solaris 11.2의 네트워크 배치 계획 "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.conf입니다. 자세한 내용은 ipnat(1M) 명령을 참조하십시오.

NAT 규칙은 IPv4 및 IPv6 주소 모두에 적용할 수 있습니다. 그러나 각 주소 유형에 대해 규칙을 별도로 만들어야 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	ipmask가 변환되는 주소를 지정합니다.
mapport	포트 번호 범위와 함께 tcp, udp 또는 tcp/udp 프로토콜을 지정합니다.

다음 예에서는 NAT 규칙을 구성하는 방법을 보여줍니다. 소스 주소가 192.168.1.0/24인 net2 장치에서 송신되는 패킷을 재작성하고 외부적으로 소스 주소를 10.1.0.0/16으로 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 풀에서는 주소 그룹/넷마스크 쌍에 대한 단일 참조를 설정합니다. 주소 풀을 사용하면 더 적은 시간으로 IP 주소와 규칙을 일치시킬 수 있습니다. 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. 참조할 수 있는 역할은 `ipf`뿐입니다.

`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(점보그램이라고도 함)에는 65,535바이트 이상의 데이터그램이 포함되어 있습니다. IP 필터는 IPv6 점보그램을 지원하지 않습니다.

참고 - Jumbogram에 대한 자세한 내용은 IPv6 Jumbograms, RFC 2675 (http://www.ietf.org/rfc/rfc2675.txt)를 참조하십시오.

IPv6과 관련된 IP 필터 작업은 IPv4와 유사합니다. 가장 큰 차이는 특정 명령에 `-6` 옵션을 사용한다는 점입니다. `ipf` 명령과 `ipfstat` 명령 모두에는 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 개선되어 IPv6 패킷 로깅이 가능합니다. `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.

IP 필터와 관련된 작업은 5장. IP 필터 구성을 참조하십시오.

IP 필터 매뉴얼 페이지

다음과 같은 매뉴얼 페이지에서 IP 필터를 다룹니다.

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

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 구성

이 장에서는 IP 필터 작업에 대한 단계별 지침을 제공합니다. 개요 정보는 4장. Oracle Solaris의 IP 필터 정보를 참조하십시오.

이 장에서는 다음 내용을 다룹니다.

IP 필터 서비스 구성

다음 작업 맵에는 IP 필터 규칙을 만들고 서비스를 사용 및 사용 안함으로 설정하는 절차가 나옵니다.

표 5-1 IP 필터 서비스 구성 작업 맵

작업	지침
IP 필터에서 사용되는 파일과 서비스 상태를 확인합니다.	IP 필터 서비스 기본값을 표시하는 방법 [54]
네트워크 트래픽, NAT를 통한 패킷 및 주소 풀에 대한 패킷 필터링 규칙 세트를 사용자 정의합니다.	IP 필터 구성 파일을 만드는 방법 [55]
IP 필터 서비스를 사용 또는 사용 안함으로 설정하거나 새로 고칩니다.	IP 필터를 사용으로 설정하고 새로 고치는 방법 [56]
단편에 도달하는 패킷의 기본 설정을 수정합니다.	패킷 재어셈블을 사용 안함으로 설정하는 방법 [57]
시스템에서 영역 사이의 트래픽을 필터링합니다.	루프백 필터링을 사용으로 설정하는 방법 [58]
IP 필터 사용을 중지합니다.	패킷 필터링을 사용 안함으로 설정하는 방법 [59]

▼ IP 필터 서비스 기본값을 표시하는 방법

시작하기 전에 `ipfstat` 명령을 실행하려면 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 파일 등록 정보 3개에는 기본 파일 위치가 포함됩니다. 이러한 파일을 만든 경우에만 해당 파일이 존재합니다. 구성 파일의 위치를 변경하는 경우 해당 파일의 등록 정보 값을 변경해야 합니다. 절차는 IP 필터 구성 파일을 만드는 방법 [55]을 참조하십시오.

사용자 고유의 패킷 필터링 규칙을 사용자 정의하는 경우 네번째 파일 등록 정보를 수정합니다. 1단계의 2단계 및 IP 필터 구성 파일을 만드는 방법 [55]을 참조하십시오.

2. **IP 필터 서비스가 사용으로 설정되었는지 확인합니다.**

   - 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 보려면 `/etc/nwam/loc/NoNet/ipf.conf`를 읽습니다. 이 파일은 보기 전용입니다.
     - 정책을 수정하려면 IP 필터 구성 파일을 만드는 방법 [55]을 참조하십시오.

참고 - 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **IP 필터 서비스에 대한 정책 파일의 파일 위치를 지정합니다.**
   이 파일에는 패킷 필터링 규칙 세트가 포함되어 있습니다.

   a. **우선 정책 파일을 `custom`으로 설정합니다.**

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

   b. **그런 다음 위치를 지정합니다.**
   예를 들어, `/etc/ipf/myorg.ipf.conf`를 패킷 필터링 규칙 세트의 위치로 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 "IP 필터의 패킷 필터링 기능 사용" [46]을 참조하십시오. 구성 파일의 예는 "IP 필터 구성 파일 예" [77] 및 `/etc/nwam/loc/NoNet/ipf.conf` 파일을 참조하십시오.

   참고 - 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 NAT 규칙 파일을 기본 파일 이름 `/etc/ipf/ipnat.conf`를 사용하여 만듭니다. 다른 이름을 사용하는 경우 다음과 같은 `config/ipnat_config_file` 서비스 등록 정보 값의 값을 변경해야 합니다.

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

   NAT에 대한 자세한 내용은 "IP 필터의 NAT 기능 사용" [48]을 참조하십시오.

4. **(옵션) 주소 풀 구성 파일을 만듭니다.**

   주소 그룹을 단일 주소 풀로 참조하려면 풀 파일을 기본 파일 이름 `/etc/ipf/ippool.conf`로 만듭니다. 다른 이름을 사용하는 경우 다음과 같은 `config/ippool_config_file` 서비스 등록 정보 값의 값을 변경해야 합니다.

   ```
   # 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 “IP 필터의 주소 풀 기능 사용” [50]을 참조하십시오.

5. **(옵션) 루프백 트래픽의 필터링을 사용으로 설정합니다.**

   시스템에서 구성된 영역 간의 트래픽을 필터링하려면 루프백 필터링을 사용으로 설정해야 합니다. 루프백 필터링을 사용으로 설정하는 방법 [58]을 참조하십시오. 영역에 적용할 규칙 세트도 정의해야 합니다.

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

   기본적으로 단편은 IP 필터에서 재어셈블됩니다. 기본값을 수정하려면 패킷 재어셈블을 사용 안함으로 설정하는 방법 [57]을 참조하십시오.

▼ IP 필터를 사용으로 설정하고 새로 고치는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

IP 필터 구성 파일을 만드는 방법 [55]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
   ```

참고 - `refresh` 명령은 간단하게 방화벽을 사용 안함으로 설정합니다. 방화벽을 유지하려면 규칙을 추가하거나 새 구성 파일을 추가합니다. 예와 함께 절차를 보려면 “IP 필터 규칙 세트 작업” [59]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 필터 관리 권한 프로파일 및 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 권한 부여가 지정된 관리자로 로그인해야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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` 값이 `0`이면 단편이 재어셈블되는 중이 아닙니다. `current`가 `1`이면 단편이 재어셈블됩니다.

▼ 루프백 필터링을 사용으로 설정하는 방법

시작하기 전에 IP 필터 관리 권한 프로파일 및 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 권한 부여가 지정된 관리자로 로그인해야 합니다. `root` 역할에는 이러한 권한이 모두 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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` 값이 `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 IP 필터를 사용으로 설정하고 새로 고치는 방법 [56]을 참조하십시오.

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 "IP 필터 규칙 세트 작업" [59]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 맵에는 IP 필터 규칙 세트와 관련된 절차가 나옵니다.

표 5-2 IP 필터 규칙 세트 작업 작업 맵

작업	지침
활성 패킷 필터링 규칙 세트를 확인합니다.	활성 패킷 필터링 규칙 세트 확인 방법 [60]
비활성 패킷 필터링 규칙 세트를 확인합니다.	비활성 패킷 필터링 규칙 세트 확인 방법 [60]
다른 활성 규칙 세트를 활성화합니다.	다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법 [61]
규칙 세트를 제거합니다.	패킷 필터링 규칙 세트 제거 방법 [62]
규칙 세트에 규칙을 추가합니다.	활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법 [62] 비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법 [63]
활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.	활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법 [64]
커널에서 비활성 규칙 세트를 삭제합니다.	커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법 [65]
활성 NAT 규칙을 확인합니다.	IP 필터에서 활성 NAT 규칙을 확인하는 방법 [66]
NAT 규칙을 제거합니다.	IP 필터에서 NAT 규칙을 비활성화하는 방법 [66]
규칙을 추가하여 NAT 규칙을 활성화합니다.	NAT 패킷 필터링 규칙에 규칙을 추가하는 방법 [67]

작업	지침
활성 주소 풀을 확인합니다.	활성 주소 풀 확인 방법 [68]
주소 풀을 제거합니다.	주소 풀 제거 방법 [68]
주소 풀에 규칙을 추가합니다.	주소 풀에 규칙을 추가하는 방법 [69]

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.

참고 - 다음 절차는 IPv4 네트워크의 예를 제공합니다. IPv6 패킷의 경우 `-How to Display IP Filter Service Defaults`의 2단계에 설명된 대로 IP 필터 서비스 기본값을 표시하는 방법 [54] 옵션을 사용합니다.

▼ 활성 패킷 필터링 규칙 세트 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **활성 패킷 필터링 규칙 세트를 확인합니다.**
  다음 예에서는 커널에서 로드된 활성 패킷 필터링 규칙 세트의 출력을 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **비활성 패킷 필터링 규칙 세트를 확인합니다.**
  다음 예에서는 비활성 패킷 필터링 규칙 세트의 출력을 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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.

참고 - 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 보여줍니다.

```
# 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 보여줍니다.

필요에 따라 활성 규칙 세트를 나열합니다.

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

그런 다음 `/etc/ipf/myorg.ipf.conf` 구성 파일을 편집하고 서비스를 새로 고치고 활성 규칙 세트를 다시 나열합니다.

```
# svcadm refresh network/ipfilter
# ipfstat -io
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any
block in quick on net1 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 보여줍니다.

```
# 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 테스트나 문제 해결 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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.

    1. 선택한 파일에 규칙 세트를 만듭니다.
    2. 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 보여줍니다.

```
# 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 테스트나 문제 해결 시 유용할 수 있습니다. IP 필터 서비스를 중지하지 않고도 해당 규칙 세트를 활성 규칙 세트로 전환할 수 있습니다. 하지만 서

비스를 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 우선순위가 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 보여줍니다.

```
# 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 전환하는 기능은 테스트나 문제 해결 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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.

참고 - 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 보여줍니다.

- `ipf -s` 명령을 실행하기 전에 `ipfstat -I -io` 명령의 출력은 비활성 규칙 세트의 규칙을 보여줍니다. `ipfstat -io` 명령의 출력은 활성 규칙 세트의 규칙을 보여줍니다.

```
# 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 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **"모두 비우기" 명령에 비활성 규칙 세트를 지정합니다.**

```
# ipf -I -Fa
```

참고 - 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 보여줍니다.

```
# 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

다음은 IP 필터의 NAT 규칙을 관리하고, 보고, 수정하는 절차입니다.

▼ IP 필터에서 활성 NAT 규칙을 확인하는 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **활성 NAT 규칙을 확인합니다.**

  다음 예에서는 활성 NAT 규칙 세트의 출력을 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **커널에서 NAT 규칙을 제거합니다.**

```
# ipnat -FC
```

  -C 옵션은 현재 NAT 규칙 목록의 모든 항목을 제거합니다. -F 옵션은 현재 활성 NAT 매핑을 보여주는 현재 NAT 변환 테이블의 모든 활성 항목을 제거합니다.

예 5-8 NAT 규칙 제거

다음 예에서는 현재 NAT 규칙의 항목을 제거하는 방법을 보여줍니다.

```
# 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 테스트나 문제 해결 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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.

    1. 선택한 파일에 추가 NAT 규칙을 만듭니다.
    2. 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 보여줍니다.

```
# 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

다음은 주소 풀을 관리하고, 보고, 수정하는 절차입니다.

▼ 활성 주소 풀 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **활성 주소 풀을 확인합니다.**

  다음 예에서는 활성 주소 풀의 컨텐츠를 확인하는 방법을 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **현재 주소 풀의 항목을 제거합니다.**

```
# ippool -F
```

예 5-10 주소 풀 제거

다음 예에서는 주소 풀 제거 방법을 보여줍니다.

```
# 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 테스트나 문제 해결 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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.

     1. 선택한 파일에 추가 주소 풀을 만듭니다.
     2. 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.**
   활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법 [62]의 지침을 따릅니다.

참고 - IP 필터 서비스를 새로 고치거나 다시 시작하지 마십시오. 그럴 경우 추가한 주소 풀 규칙이 손실됩니다.

예 5-11 주소 풀에 규칙 추가

다음 예에서는 명령줄에서 주소 풀 규칙 세트에 주소 풀을 추가하는 방법을 보여줍니다.

```
# 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	지침
상태 테이블을 확인합니다.	IP 필터에 대한 상태 테이블 확인 방법 [70]
패킷 상태에 대한 통계를 확인합니다.	IP 필터에 대한 상태 통계 확인 방법 [71]
IP 필터 조정 가능 매개변수를 나열합니다.	IP 필터 조정 가능 매개변수를 확인하는 방법 [72]
NAT 통계를 확인합니다.	IP 필터에 대한 NAT 통계 확인 방법 [72]
주소 풀 통계를 확인합니다.	IP 필터에 대한 주소 풀 통계 확인 방법 [72]

▼ IP 필터에 대한 상태 테이블 확인 방법

시작하기 전에 IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **상태 테이블을 확인합니다.**

```
# ipfstat
```

참고 - `-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 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **상태 통계를 확인합니다.**

```
# ipfstat -s
```

예 5-13 IP 필터에 대한 상태 통계 보기

다음 예에서는 상태 통계 출력을 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **NAT 통계를 확인합니다.**

```
# ipnat -s
```

예 5-14 IP 필터에 대한 NAT 통계 보기

다음 예에서는 NAT 통계를 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **주소 풀 통계를 확인합니다.**

```
# ippool -s
```

예 5-15 IP 필터에 대한 주소 풀 통계 보기

다음 예에서는 주소 풀 통계를 보여줍니다.

```
# 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	지침
별도의 IP 필터 로그 파일을 만듭니다.	IP 필터 로그 파일 설정 방법 [73]
상태, NAT 및 일반 로그 파일을 확인합니다.	IP 필터 로그 파일 확인 방법 [74]
패킷 로그 버퍼를 비웁니다.	패킷 로그 버퍼를 비우는 방법 [75]
나중에 참조할 수 있도록 기록된 패킷을 파일에 저장합니다.	기록된 패킷을 파일에 저장하는 방법 [76]

▼ IP 필터 로그 파일 설정 방법

기본적으로 IP 필터에 대한 모든 로그 정보는 `syslog`로 기록됩니다. `syslog` 로그 파일에 기록될 수 있는 다른 데이터와 별도로 IP 필터 트래픽 정보를 기록할 로그 파일을 만드는 것이 좋습니다.

시작하기 전에 `root` 역할을 맡아야 합니다.

1. **사용으로 설정된 `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
```

참고 - `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
```

참고 - 입력 시 스페이스바가 아닌 Tab 키를 사용하여 `local0.debug`와 `/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

참고 - `rsyslog` 서비스가 사용으로 설정되어 있으면 `system-log:rsyslog` 서비스 인스턴스를 새로 고칩니다.

예 5-16 IP 필터 로그 만들기

다음 예에서는 IP 필터 정보를 아카이브할 `ipmon.log`를 만드는 방법을 보여줍니다.

`syslog.conf`를 편집합니다.

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

그런 다음 명령줄에서 파일을 만들고 서비스를 다시 시작합니다.

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

▼ IP 필터 로그 파일 확인 방법

시작하기 전에 IP 필터 로그 파일 설정 방법 [73]을 완료했습니다.

IP Filter Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"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
  ```

참고 - 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. 그런 다음 두 복사본은 동일한 로그 정보를 읽지만 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 보여줍니다.

```
# 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 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **패킷 로그 버퍼를 비웁니다.**

  ```
  # ipmon -F
  ```

예 5-18 패킷 로그 버퍼 비우기

다음 예에서는 로그 파일 제거 시 출력을 보여줍니다. 이 예에 나온 대로 로그 파일이 비어 있는 경우에도 보고서가 제공됩니다.

```
# 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

문제를 해결하는 동안이나 트래픽을 수동으로 감사하려는 경우 패킷을 파일에 저장할 수 있습니다.

시작하기 전에 `root` 역할을 맡아야 합니다.

- **기록된 패킷을 파일에 저장합니다.**

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

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

예 5-19 기록된 패킷을 파일에 저장

다음 예에서는 기록된 패킷을 파일에 저장한 후의 결과를 보여줍니다.

```
# 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 보여줍니다. 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(₩)가 있으면 규칙이 다음 행에서 계속됨을 나타냅니다.

자세한 구문 정보는 “패킷 필터링 규칙 구성” [46]을 참조하십시오.

예 5-20 IP 필터 호스트 구성

이 예에서는 `net0` 네트워크 인터페이스를 사용하는 호스트 시스템의 구성을 보여줍니다.

```
# 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 `10.0.0.0` 및 `172.16.0.0`의 수신 패킷이 방화벽에 들어오는 것을 차단합니다. 다음 규칙 세트는 호스트 시

스템의 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 서버 역할을 하는 호스트 시스템에 대한 구성을 보여줍니다. 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

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 인터페이스가 `net0`이고 외부 인터페이스가 `net1`인 라우터에 대한 구성을 보여줍니다.

```
# 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
```

```
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 are not 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
```

```
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 Security Architecture 정보

IPsec(IP Security Architecture)는 IPv4 및 IPv6 네트워크에서 IP 패킷에 대한 암호화 보호를 제공합니다.

이 장에서는 다음 내용을 다룹니다.

네트워크에서 IPsec을 구현하려면 7장. IPsec 구성을 참조하십시오. 참조 정보는 12장. IPsec 및 키 관리 참조를 참조하십시오.

IPsec 소개

IPsec은 암호화를 사용하여 IP 패킷의 내용을 보호하고 패킷 내용을 인증하여 무결성 검사를 제공합니다. IPsec은 네트워크 계층에서 수행되므로, 네트워크 응용 프로그램에서 IPsec을 사용하도록 자체적으로 구성하지 않아도 IPsec을 활용할 수 있습니다. 제대로 사용되면 IPsec는 네트워크 트래픽을 보호하는 효과적인 도구가 될 수 있습니다.

IPsec에서는 다음과 같은 용어를 사용합니다.

- **보안 프로토콜** - IP 패킷에 적용되는 보호입니다. authentication header(인증 헤더)(AH)는 IP 헤더를 포함하는 전체 패킷의 해시인 ICV(Integrity Check Vector)를 추가

하여 IP 패킷을 보호합니다. 수신자는 패킷이 수정되지 않았다고 확신할 수 있습니다. 암호화를 통한 기밀성은 제공하지 않습니다.

ESP(보안 페이로드 캡슐화)는 IP 패킷의 페이로드를 보호합니다. 패킷의 페이로드는 암호화되어 기밀성을 제공할 수 있으며 ICV를 사용하여 데이터 무결성을 보장할 수 있습니다.

- **SA(보안 연관)** - 암호화 매개변수, 키, IP 보안 프로토콜, IP 주소, IP 프로토콜, 포트 번호 및 기타 매개변수로, 특정 SA를 특정 트래픽 플로우와 일치시키는 데 사용됩니다.
- **SADB(보안 연관 데이터베이스)** - 보안 연관을 저장하는 데이터베이스입니다. SA는 SPI(보안 매개변수 색인), 보안 프로토콜 및 대상 IP 주소에서 참조합니다. 이러한 세 가지 요소는 IPsec SA를 고유하게 식별합니다. 시스템에서 IPsec 헤더(ESP 또는 AH)가 있는 IP 패킷을 수신하는 경우 시스템은 SADB를 검색하여 일치하는 SA를 찾습니다. 일치하는 SA를 찾으면 이 SA를 사용하여 IPsec에서 패킷을 해독하고 확인할 수 있습니다. 검증에 실패하거나 일치하는 SA가 없으면 패킷이 무시됩니다.
- **키 관리** - 암호화 알고리즘에서 사용되는 키를 안전하게 생성 및 배포하고 키를 저장하는 데 사용되는 SA를 생성하는 작업입니다.
- **SPD(보안 정책 데이터베이스)** - IP 트래픽에 적용되는 보호 정책을 지정하는 데이터베이스입니다. SPD는 트래픽을 필터링하여 패킷이 어떻게 처리되어야 하는지 결정합니다. 패킷은 무시되거나 투명하게 전달될 수 있습니다. 또는 패킷이 IPsec으로 보호될 수 있습니다(즉, 보안 정책이 적용됨).

  아웃바운드 패킷의 경우 IPsec 정책에 따 IPsec이 IP 패킷에 적용될지 여부가 결정됩니다. IPsec이 적용되는 경우 IP 모듈에서는 SADB를 검색하여 일치하는 SA를 찾고 이 SA를 사용하여 정책을 적용합니다.

  인바운드 패킷의 경우 IPsec 정책에서는 수신된 패킷의 보호 레벨이 적절한지 확인합니다. 정책에서 특정 IP 주소의 패킷을 IPsec으로 보호해야 하는 경우 보호되지 않는 패킷은 모두 무시됩니다. 인바운드 패킷이 IPsec으로 보호되는 경우 IP 모듈에서는 SADB를 검색하여 일치하는 SA를 찾고 SA를 패킷에 적용합니다.

응용 프로그램에서는 IPsec을 호출하여 소켓별 레벨에서도 IP 패킷에 보안 방식을 적용할 수 있습니다. 포트의 소켓이 연결되고 나중에 해당 포트에 IPsec 정책이 적용될 경우 해당 소켓을 사용하는 트래픽은 IPsec으로 보호되지 않습니다. 물론, IPsec 정책이 포트에 적용된 이후 포트에서 열린 소켓은 IPsec 정책으로 보호됩니다.

IPsec 패킷 플로우

그림 6-1. “아웃바운드 패킷 프로세스에 적용된 IPsec”에서는 IPsec이 아웃바운드 패킷에서 호출된 경우 IP packet(IP 패킷)이 진행되는 방식을 보여줍니다. 플로우 다이어그램은 AH(authentication header) 및 ESP(encapsulating security payload) 엔티티를 어디에서 패킷에 적용할 수 있는지 보여줍니다. 이후 절에서는 이러한 엔티티를 적용하는 방법 및 알고리즘을 선택하는 방법을 설명합니다.

그림 6-2. “인바운드 패킷 프로세스에 적용된 IPsec”에서는 IPsec 인바운드 프로세스를 보여줍니다.

그림 6-1 아웃바운드 패킷 프로세스에 적용된 IPsec

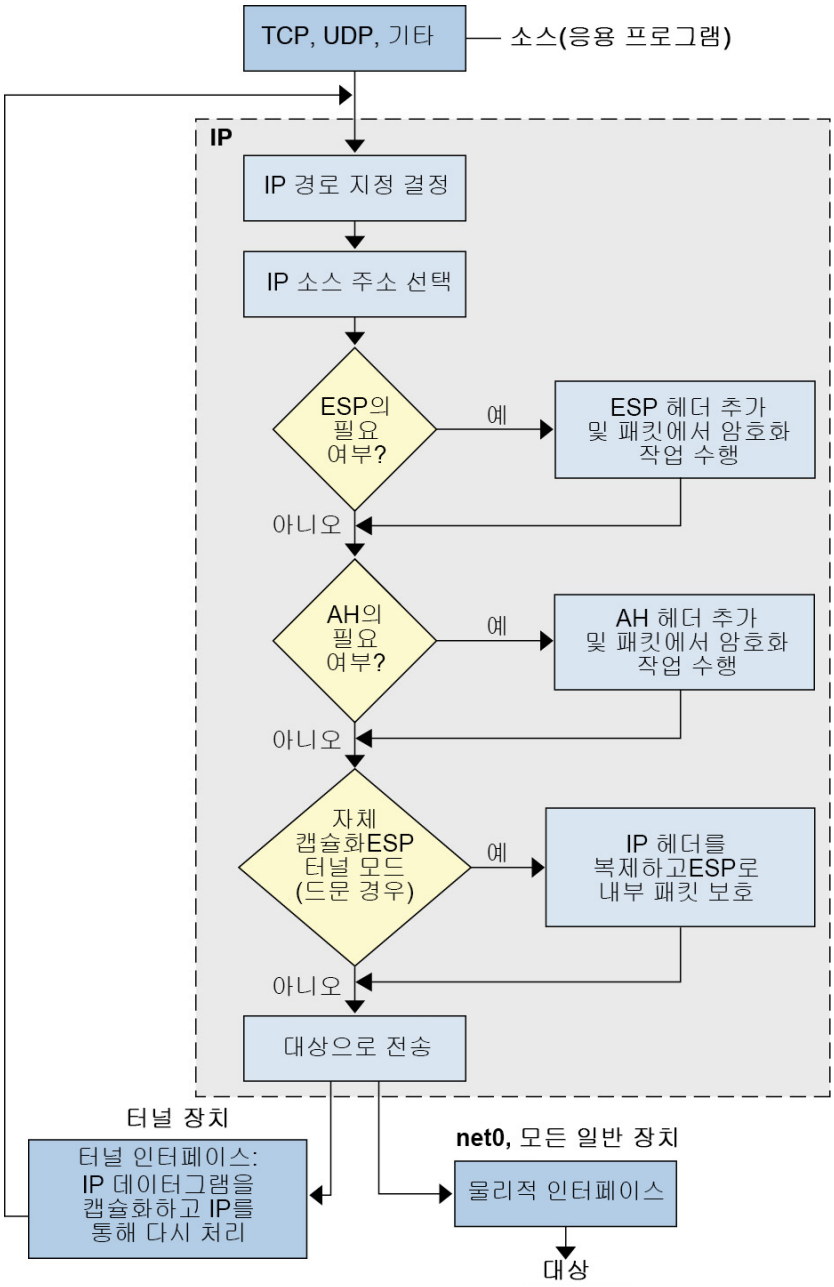
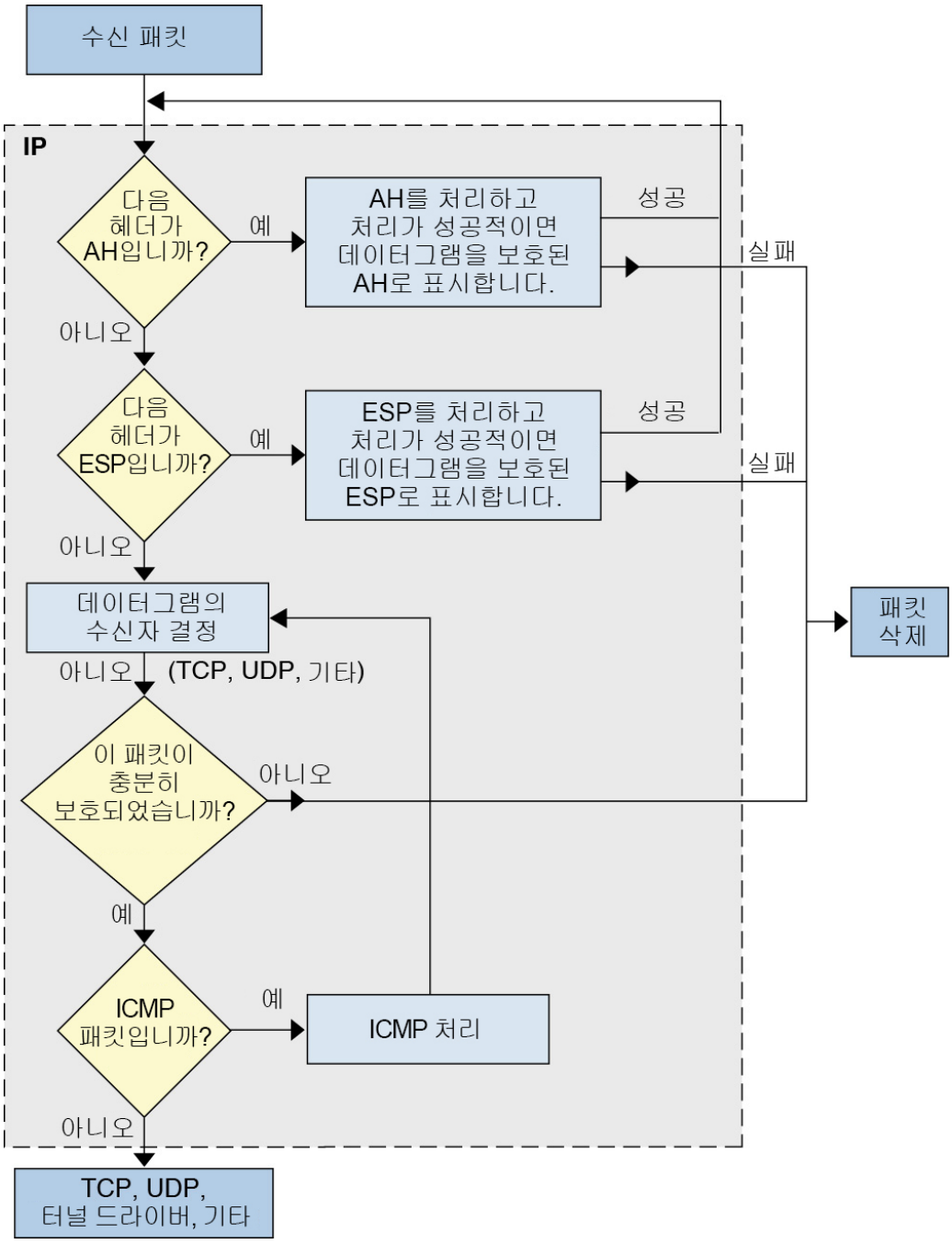


그림 6-2 인바운드 패킷 프로세스에 적용된 IPsec



IPsec 보안 연관

IPsec 보안 연관 SA는 SA에도 저장되는 IP 매개변수와 일치하는 IP 패킷에 적용할 보안 등록 정보를 정의합니다. 각 SA 단방향입니다. 통신은 대부분 양방향이므로 단일 연결에 SA가 두 개 필요합니다.

다음 세 가지 요소가 함께 IPsec SA를 고유하게 식별합니다.

- 보안 프로토콜(AH 또는 ESP)
- 대상 IP 주소
- SPI(보안 매개변수 색인)

SA에 대한 SPI는 보호를 추가로 제공하며 IPsec으로 보호되는 패킷의 AH 또는 ESP 헤더로 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 연관은 SADB(보안 연관 데이터베이스)에 저장됩니다. 소켓 기반 관리 인터페이스인 PF_KEY를 사용하면 권한이 부여된 응용 프로그램이 데이터베이스를 프로그래밍 방식으로 관리할 수 있습니다. 예를 들어, IKE 데몬 및 `ipseckey` 명령은 PF_KEY 소켓 인터페이스를 사용합니다.

IPsec SADB에 대한 자세한 설명은 "IPsec에 대한 보안 연관 데이터베이스" [212]를 참조하십시오.

SADB를 관리하는 방법에 대한 자세한 내용은 `pf_key`(7P) 및 `ipseckey`(1M) 매뉴얼 페이지를 참조하십시오.

IPsec 보안 연관에 대한 키 관리

SA(보안 연관)에는 인증 및 암호화를 위한 키 입력 자료가 필요합니다. 이 키 입력 도구를 관리하는 작업을 키 관리라고 합니다. Oracle Solaris에서는 IPsec SA에 대한 키를 IKE와 수동 키 관리라는 두 가지 방법으로 관리합니다.

IPsec SA 생성을 위한 IKE

IKE(Internet Key Exchange) 프로토콜에서는 키 관리를 자동으로 처리합니다. Oracle Solaris 11.2에서는 IKE 프로토콜의 IKE 버전 2(IKEv2) 및 IKE 버전 1(IKEv1)을 지원합니다.

IKE를 사용하여 IPsec SA를 관리하는 것이 좋습니다. 이러한 키 관리 프로토콜을 사용하면 다음과 같은 이점이 있습니다.

- 간단한 구성
- 강력한 피어 인증 제공
- 고품질의 임의 키 소스를 사용하여 SA를 자동으로 생성
- 관리자 개입 없이 새 SA 생성 가능

자세한 내용은 "IKE 작동 방식" [124]을 참조하십시오.

IKE를 구성하려면 9장. IKEv2 구성을 참조하십시오. IKEv2 프로토콜을 지원하지 않는 시스템과 통신하는 경우에는 10장. IKEv1 구성의 지침을 따르십시오.

IPsec SA 생성을 위한 수동 키

수동 키를 사용하면 IKE보다 더 복잡하며 위험할 수도 있습니다. 시스템 파일 `/etc/inet/secret/ipseckeys`에는 암호 키가 포함됩니다. 이러한 키는 손상되는 경우 기록된 네트워크 트래픽을 해독하는 데 사용될 수 있습니다. IKE에서는 키를 자주 변경하므로 이러한 손상에 노출되는 기간이 훨씬 작습니다. IKE를 지원하지 않는 시스템에서만 `ipseckeys` 파일이나 해당 명령 인터페이스 `ipseckey`를 사용하는 것이 좋습니다.

`ipseckey` 명령에는 제한된 수의 일반 옵션만 있지만 명령은 풍부한 명령 언어를 지원합니다. 수동 키 입력에 대한 프로그래밍 인터페이스로 해당 요청이 전달되도록 지정할 수 있습니다. 자세한 내용은 `ipseckey`(1M) 및 `pf_key`(7P) 매뉴얼 페이지를 참조하십시오.

일반적으로 수동 SA 생성은 사정상 IKE를 사용할 수 없을 때 사용됩니다. 하지만 SPI 값이 고유한 경우 수동 SA 생성과 IKE를 동시에 사용할 수 있습니다.

IPsec 보호 프로토콜

IPsec은 데이터 보호를 위한 두 가지 보안 프로토콜을 제공합니다.

- AH(인증 헤더)
- ESP(Encapsulating Security Payload)

AH에서는 인증 알고리즘을 사용하여 데이터 무결성을 제공합니다. 패킷을 암호화하지는 않습니다.

ESP에서는 일반적으로 암호화 알고리즘을 사용하여 패킷을 보호하고 인증 알고리즘을 사용하여 데이터 무결성을 제공합니다. AES GCM과 같은 일부 암호화 알고리즘은 암호화와 인증을 모두 제공합니다.

AH 프로토콜은 NAT(Network Address Translation)와 함께 사용할 수 없습니다.

인증 헤더

authentication header(인증 헤더)는 IP 패킷에 데이터 인증, 강력한 무결성 및 재생 보호 기능을 제공합니다. AH는 IP 패킷의 많은 부분을 보호합니다. 다음 그림에 나온 대로 AH는 IP 헤더와 전송 헤더 사이에 삽입됩니다.

전송 헤더는 TCP, UDP, SCTP 또는 ICMP가 될 수 있습니다. tunnel(터널)이 사용되는 경우 전송 헤더는 다른 IP 헤더가 될 수 있습니다.

ESP(Encapsulating Security Payload)

ESP(보안 페이로드 캡슐화) 프로토콜은 ESP가 캡슐화하는 항목에 대한 기밀성을 제공합니다. 또한 ESP는 AH가 제공하는 서비스도 제공합니다. 그러나 ESP는 외부 IP 헤더는 보호하지 않습니다. ESP는 보호된 패킷의 무결성을 위해 인증 서비스를 제공합니다. ESP는 암호화 지원 기술을 사용하므로 ESP를 제공하는 시스템은 가져오기 및 내보내기 제어 규칙에 종속될 수 있습니다.

ESP 헤더 및 트레일러는 IP 페이로드를 캡슐화합니다. ESP와 함께 암호화가 사용되는 경우 다음 그림에 나온 대로 IP 페이로드 데이터에만 암호화가 적용됩니다.

TCP 패킷에서는 ESP 헤더가 인증되고 TCP 헤더 및 해당 데이터를 캡슐화합니다. 패킷이 IP-in-IP 패킷인 경우 ESP는 내부 IP 패킷을 보호합니다. 소켓별 정책에서는 자체 캡슐화를 허용하므로 ESP에서 필요할 때 IP 옵션을 캡슐화할 수 있습니다.

자체 캡슐화는 `setsockopt()`를 사용하는 프로그램을 작성하여 사용할 수 있습니다. 자체 캡슐화가 설정되면 IP 헤더의 복사본이 IP-in-IP 패킷을 생성하게 됩니다. 예를 들어, 자체 캡슐화가 TCP 소켓에서 설정되지 않은 경우 패킷은 다음 형식으로 전송됩니다.

`[ IP(a -> b)` *options* `+ TCP + data ]`

자체 캡슐화가 TCP 소켓에서 설정된 경우 패킷은 다음 형식으로 전송됩니다.

```
[ IP(a -> b) + ESP [ IP(a -> b) options + TCP + data ] ]
```

자세한 내용은 "IPsec의 전송 및 터널 모드" [91]를 참조하십시오.

AH 및 ESP를 사용할 때 보안 고려 사항

다음 표에서는 AH와 ESP에서 제공하는 보호 기능을 비교합니다.

표 6-1 IPsec에서 AH 및 ESP로 제공되는 보호 기능

프로토콜	패킷 범위	보호	공격 방어
AH	IP 헤더에서 전송 데이터 끝 사이의 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	ESP 헤더에서 전송 데이터 끝 사이의 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 보안에서는 인증과 암호화라는 두 가지 알고리즘 유형을 사용합니다. AH 프로토콜은 인증 알고리즘을 사용합니다. ESP 프로토콜은 인증 알고리즘과 함께 암호화를 사용할 수 있습니다. 시스템의 알고리즘 및 해당 등록 정보 목록은 `ipsecalgs` 명령을 사용하여 확인할 수 있습니다. 자세한 내용은 `ipsecalgs`(1M) 매뉴얼 페이지를 참조하십시오. 또한 `getipsecalgbyname`(3NSL) 매뉴얼 페이지에 설명된 기능을 사용하여 알고리즘의 등록 정보를 검색할 수도 있습니다.

IPsec에서는 암호화 프레임워크를 사용하여 암호화와 인증을 수행합니다. IPsec은 암호화 프레임워크를 통해 하드웨어에서 지원하는 경우 하드웨어 가속을 활용할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- "Oracle Solaris 11.2의 암호화 및 인증서 관리 "의 1 장, "암호화 프레임워크"
- "Developer's Guide to Oracle Solaris 11 Security "의 8 장, "Introduction to the Oracle Solaris Cryptographic Framework"

IPsec 보호 정책

IPsec 보호 정책은 다음 레벨에서 적용할 수 있습니다.

- 시스템 차원 레벨
- 소켓별 레벨

IPsec은 IPsec 정책 규칙과 일치하는 아웃바운드 패킷 및 인바인드 패킷에 시스템 차원 정책을 적용합니다. 규칙에서는 특정 알고리즘을 지정하거나 여러 알고리즘 중 하나를 허용할 수 있습니다. 아웃바운드 패킷의 경우 시스템에서 데이터를 추가로 인식하므로 추가 규칙을 적용할 수 있습니다.

인바운드 패킷은 수락되거나 삭제됩니다. 인바운드 패킷을 삭제할지 또는 수락할지는 여러 기준에 따라 결정됩니다. 기준이 겹치거나 충돌하는 경우 먼저 구문 분석된 규칙이 사용됩니다.

IPsec 정책이 대부분의 패킷에 적용되지 않는 경우인 예외를 지정할 수 있습니다. 즉, IPsec 정책을 우회할 수 있습니다. 우회는 시스템 전체 또는 소켓별로 가능합니다.

공유 IP 주소의 영역을 포함하는 시스템 내의 트래픽인 경우 정책은 적용되지만 실제 보안 방식은 적용되지 않습니다. 대신 시스템간 패킷에 대한 아웃바운드 정책은 해당 방식이 적용된 인바운드 패킷으로 변환됩니다. 배타적 IP 영역인 경우 정책이 적용되고 실제 보안 방식도 적용됩니다.

`ipsecinit.conf` 파일 및 `ipsecconf` 명령을 사용하여 IPsec 정책을 구성합니다. 자세한 내용과 예는 `ipsecconf`(1M) 매뉴얼 페이지와 7장. IPsec 구성을 참조하십시오.

IPsec의 전송 및 터널 모드

IPsec 표준에서는 전송 모드 및 터널 모드의 두 가지 고유 IPsec 작업 모드를 정의합니다. 전송 모드와 터널 모드의 주된 차이점은 정책이 적용되는 위치입니다. 터널 모드에서는 원본 패킷이 다른 IP 헤더에 캡슐화됩니다. 헤더가 다르면 주소가 다를 수 있습니다.

패킷은 각 모드에서 AH, ESP 또는 둘 다로 보호될 수 있습니다. 두 모드는 정책 적용에서 다음과 같이 차이가 납니다.

- 전송 모드에서는 외부 헤더의 IP 주소를 사용하여 패킷에 적용할 IPsec 정책을 결정합니다.
- 터널 모드에서는 두 IP 헤더가 적송됩니다. 내부 IP 패킷은 해당 내용을 보호하는 IPsec 정책을 결정합니다.

  터널 모드는 엔드 시스템 및 보안 게이트웨이와 같은 중간 시스템이 임의로 혼합된 경우에 적용할 수 있습니다.

전송 모드에서는 IP 헤더, 다음 헤더 및 다음 헤더가 지원하는 모든 포트를 사용하여 IPsec 정책을 결정할 수 있습니다. 실제로 IPsec는 두 IP 주소 사이에 서로 다른 전송 모드 정책을 적

용하여 단일 포트를 세분화할 수 있습니다. 예를 들어, 다음 헤더가 포트를 지원하는 TCP인 경우 IPsec 정책을 외부 IP 주소의 TCP 정책에 대해 설정할 수 있습니다.

터널 모드는 IP-in-IP 패킷에 대해서만 작동합니다. 터널 모드에서는 IPsec 정책이 내부 IP 패킷의 내용에 적용됩니다. 서로 다른 내부 IP 주소에 대해 서로 다른 IPsec 정책을 적용할 수 있습니다. 즉, 내부 IP 헤더, 다음 헤더 및 다음 헤더가 지원하는 포트에서 정책을 적용할 수 있습니다. 전송 모드와 달리 터널 모드에서는 외부 IP 헤더에 따라 내부 IP 패킷의 정책이 결정되지 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 “IPsec를 사용하여 VPN 보호” [106]를 참조하십시오.

Oracle Solaris에서는 IP 터널링 네트워크 인터페이스에서만 터널 모드를 적용할 수 있습니다. 터널링 인터페이스에 대한 자세한 내용은 “Oracle Solaris 11.2의 TCP/IP 네트워크, IPMP 및 IP 터널 관리 ”의 4 장, “IP 터널 관리 정보”를 참조하십시오. IPsec 정책에서는 IP 터널링 네트워크 인터페이스를 선택하는 `tunnel` 키워드를 제공합니다. `tunnel` 키워드가 규칙에 존재하는 경우 해당 규칙에서 지정된 모든 선택기가 내부 패킷에 적용됩니다.

다음 그림은 보호되지 않는 TCP 패킷의 IP 헤더를 보여줍니다.

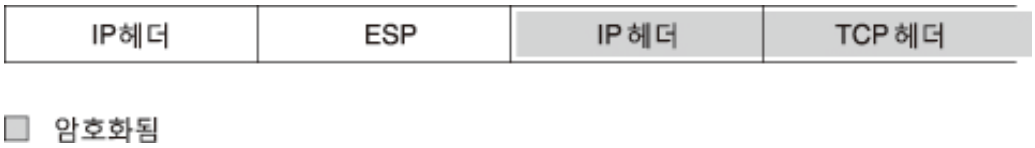
그림 6-3 TCP 정보를 전달하는 보호되지 않는 IP 패킷

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

그림 6-4 TCP 정보를 전달하는 보호된 IP 패킷

터널 모드에서는 전체 패킷이 ESP 헤더 내부에 있습니다. 그림 6-3. “TCP 정보를 전달하는 보호되지 않는 IP 패킷”의 패킷은 다음 그림에 나온 대로 터널 모드에서 외부 IPsec 헤더(이 경우 ESP)로 보호됩니다.

그림 6-5 터널 모드에서 보호된 IPsec 패킷



IPsec 정책에서는 터널 모드 및 전송 모드에 대한 키워드를 제공합니다. 자세한 내용은 다음을 참조하십시오.

- 소켓별 정책에 대한 자세한 내용은 `ipsec`(7P) 매뉴얼 페이지를 참조하십시오.
- 소켓별 정책의 예는 IPsec을 사용하여 웹 서버와 다른 서버의 통신을 보호하는 방법 [104]을 참조하십시오.
- 터널에 대한 자세한 내용은 `ipsecconf`(1M) 매뉴얼 페이지를 참조하십시오.
- 터널 구성의 예는 터널 모드에서 IPsec을 사용하여 두 LAN 사이의 연결을 보호하는 방법 [109]을 참조하십시오.

VPN(Virtual Private Networks) 및 IPsec

VPN(가상 사설망)이라는 용어는 인터넷과 같은 공개 네트워크 위에 구축된 안전한 지점 간 개인 네트워크를 설명하는 데 자주 사용됩니다. 지점 간 네트워크 또는 VPN은 개인 네트워크의 시스템 또는 개인 네트워크의 시스템 네트워크를 연결하는 데 사용될 수 있습니다.

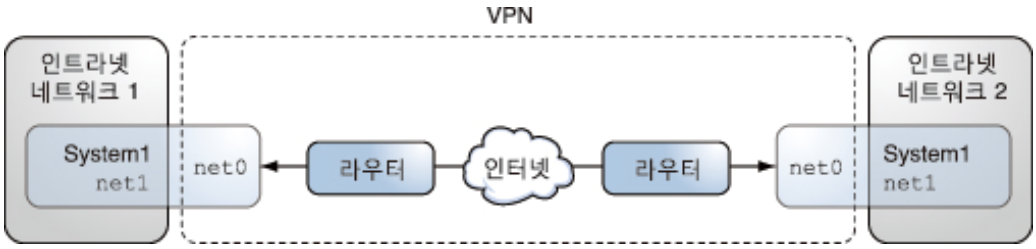
구성된 tunnel(터널)은 지점 간 인터페이스입니다. 터널을 통해 한 IP 패킷을 다른 IP 패킷 내부에 캡슐화할 수 있습니다. 올바르게 구성된 터널에는 터널 소스와 터널 대상이 필요합니다. 자세한 내용은 “Oracle Solaris 11.2의 TCP/IP 네트워크, IPMP 및 IP 터널 관리 ”의 “IP 터널을 만들고 구성하는 방법”을 참조하십시오.

터널은 IP에 대한 명백한 physical interface(물리적 인터페이스)를 만듭니다. IP 터널 인터페이스를 통해 전달되는 IP 트래픽은 IPsec으로 보호될 수 있습니다.

Oracle Solaris의 터널 인터페이스를 사용하면 시스템 간의 IP 패킷을 캡슐화 또는 터널링할 수 있습니다. 터널링된 패킷에서는 원래 IP 헤더 앞에 IP 헤더가 추가됩니다. 추가된 헤더는 공개 네트워크에서 경로 지정 가능한 주소를 사용합니다. 이러한 주소는 다음 다이어그램의 `net0` 인터페이스로 표시됩니다.

다음 그림에서는 두 사이트에서 IPsec을 사용하여 사이트 사이에 VPN을 만드는 방법을 보여 줍니다. `Intranet 1`과 `Intranet 2` 간의 트래픽은 인터넷을 통해 IP-in-ESP 캡슐화를 사용하여 터널링됩니다. 이 경우 `net0` 주소는 외부 IP 헤더에서 사용되지만, 내부 IP 주소는 인트라넷 네트워크에서 터널링된 패킷의 주소입니다. 내부 IP 주소는 ESP로 보호되므로 트래픽이 인터넷을 통과할 때 검사되지 않습니다.

그림 6-6 VPN(가상 사설망)



설정 절차의 자세한 예는 터널 모드에서 IPsec을 사용하여 두 LAN 사이의 연결을 보호하는 방법 [109]을 참조하십시오.

IPsec 및 FIPS 140

FIPS 140 사용 시스템에서 FIPS 140 요구 사항을 준수하도록 IPsec를 쉽게 구성할 수 있습니다. 키 및 인증서를 만들기 위해서는 FIPS 140 검증 알고리즘만 선택해야 합니다. 이 안내서의 절차 및 예제에는 `any` 알고리즘이 지정되지 않은 한 FIPS 140 승인 알고리즘이 사용됩니다.

참고 - FIPS 140-2 검증 암호화만 사용하도록 요구 사항이 엄격한 경우, Oracle Solaris 11.1 SRU 5.5 릴리스 또는 Oracle Solaris 11.1 SRU 3 릴리스를 실행해야 합니다. Oracle은 이 두 가지 릴리스에서 암호화 프레임워크에 대한 FIPS 140-2 검증을 마쳤습니다. Oracle Solaris 11.2는 이러한 검증을 기초로 제작되었으며 성능, 기능 및 안정성 문제를 해결하는 소프트웨어 향상 기능을 포함합니다. 이러한 향상 기능을 활용하기 위해서는 가능한 모든 경우에 Oracle Solaris 11.2를 FIPS 140-2 모드로 구성해야 합니다.

다음 방식을 IPsec에 사용할 수 있으며, FIPS 140 모드의 Oracle Solaris에서 사용하도록 승인되어 있습니다.

- 키 길이가 128~256비트인 CBC, CCM, GCM 및 GMAC 모드의 AES
- 3DES
- SHA1

- 키 길이가 256비트 및 512비트인 SHA2

Oracle Solaris에 대한 FIPS 140 검증 알고리즘의 최종 목록은 http://www.oracle.com/technetwork/topics/security/140sp2061-2082028.pdf를 참조하십시오. 자세한 내용은 "Using a FIPS 140 Enabled System in Oracle Solaris 11.2 "을 참조하십시오.

IPsec 및 NAT 순회

IKE는 NAT 장치에 걸쳐 IPsec SA를 협상할 수 있습니다. 시스템이 NAT 장치 뒤에 있더라도 이 기능을 통해 원격 네트워크에서 안전하게 연결할 수 있습니다. 예를 들어, 집에서 작업하거나 회의실에서 로그온하는 직원이 IPsec을 사용하여 트래픽을 보호할 수 있습니다.

NAT 장치는 개인 내부 주소를 고유한 인터넷 주소로 변환합니다. NAT는 호텔과 같은 인터넷 공용 액세스 지점에서 매우 일반적입니다.

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. RFC 사본은 http://www.rfc-editor.org에서 제공됩니다.

- 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

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 않습니

다. 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 Stream Control Transmission Protocol (SCTP) RFC를 참조하십시오.

IPsec 및 Oracle Solaris 영역

IPsec은 영역에서 지원됩니다. 각 영역에는 고유한 IPsec 정책 및 IKE 구성이 있을 수 있습니다. 영역을 별도의 호스트처럼 처리할 수 있습니다.

예외적으로 공유 IP 영역의 경우 자체 IP 스택이 없습니다. 공유 IP 영역의 경우 IPsec 정책과 IKE 구성이 전역 영역에서 수행됩니다. 공유 IP 영역의 IPsec 정책 규칙에서는 해당 영역에 지정된 IP 주소를 사용합니다.

자세한 내용은 "Oracle Solaris 영역 소개 "의 1 장, "Oracle Solaris 영역 소개"을 참조하십시오.

IPsec 및 가상 머신

IPsec은 VM(가상 머신)에서 작동합니다. SPARC 시스템에서 VM을 만들려면 Oracle VM Server를 사용합니다. x86 시스템에서는 Oracle VM VirtualBox를 사용할 수 있습니다. 구성에 대한 자세한 내용은 사용 중인 Oracle VM 버전의 관리 설명서를 참조하십시오.

IPsec 구성 명령 및 파일

표 6-2. "선택한 IPsec 구성 명령 및 파일"에서는 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 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오.

네트워크에서 IPsec 구현에 대한 지침은 "IPsec을 사용하여 네트워크 트래픽 보호" [99]를 참조하십시오.

IPsec 유틸리티 및 파일에 대한 자세한 내용은 12장. IPsec 및 키 관리 참조를 참조하십시오.

표 6-2 선택한 IPsec 구성 명령 및 파일

IPsec 명령, 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:ikev2, svc:/network/ipsec/ike:default	IKE를 사용하여 IPsec SA를 자동으로 관리하는 SMF 서비스 인스턴스입니다.	smf(5), in.ikev2d(1M), in.iked(1M)
/etc/inet/ipsecinit.conf 파일	IPsec 정책 파일입니다. SMF policy 서비스에서 시스템 부트 시 IPsec 정책을 구성하는 데 사용됩니다.	ipsecconf(1M)
ipsecconf 명령	IPsec 정책 명령입니다. 현재 IPsec 정책을 보고 수정하며 테스트하는 데 유용합니다. SMF policy 서비스에서 시스템 부트 시 IPsec 정책을 구성하는 데 사용됩니다.	ipsecconf(1M)
PF_KEY 소켓 인터페이스	SADB(보안 연관 데이터베이스)에 대한 인터페이스입니다. 수동 키 관리 및 자동 키 관리를 처리합니다.	pf_key(7P)
ipseckey 명령	IPsec SA 키 입력 명령입니다. ipseckey는 PF_KEY 인터페이스에 대한 명령줄 프런트 엔드입니다. ipseckey로 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.	
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 방식 및 알고리즘 정의를 포함합니다. 이 파일은 ipsecalgs 명령으로 관리되며 수동으로 편집하면 안됩니다.	
/etc/inet/ike/ikev2.config 파일	IKEv2 구성 및 정책 파일입니다. 키 관리는 이 파일의 규칙 및 전역 매개변수를 기반으로 이루어집니다. "IKEv2 유틸리티 및 파일" [212]을 참조하십시오.	ikev2.config(4)
/etc/inet/ike/config 파일	IKEv1 구성 및 정책 파일입니다. 기본적으로 이 파일은 존재하지 않습니다. 키 관리는 이 파일의 규칙 및 전역 매개변수를 기반으로 이루어집니다. "IKEv1 유틸리티 및 파일" [216]을 참조하십시오. 이 파일이 있으면 svc:/network/ipsec/ike:default 서비스에서는 IKEv1 데몬 in.iked를 시작합니다.	ike.config(4)

♦ ♦ ♦ 7 장

7 IPsec 구성

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

- "IPsec을 사용하여 네트워크 트래픽 보호" [99]
- "IPsec를 사용하여 VPN 보호" [106]
- "추가 IPsec 작업" [113]

IPsec에 대한 내용은 6장. IP Security Architecture 정보를 참조하십시오. IPsec에 대한 참조 정보는 12장. IPsec 및 키 관리 참조를 참조하십시오.

참고 - 이러한 작업에서는 시스템에 정적 IP 주소가 지정되어 있고 네트워크 구성 프로파일 `DefaultFixed`가 실행 중이라고 가정합니다. `netadm list` 명령에서 `Automatic`을 반환하는 경우 자세한 내용은 `netcfg`(1M) 매뉴얼 페이지를 참조하십시오.

IPsec을 사용하여 네트워크 트래픽 보호

이 절의 절차를 따르면 두 시스템 간의 트래픽을 보호하고 웹 서버를 보호할 수 있습니다. VPN을 보호하려면 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. IPsec을 관리하고 IPsec 및 IKE에서 SMF 명령을 사용하는 추가 절차는 "추가 IPsec 작업" [113]을 참조하십시오.

다음 정보는 모든 IPsec 구성 작업에 적용됩니다.

- **IPsec 및 영역** - 각 시스템은 전역 영역이나 배타적 IP 영역입니다. 자세한 내용은 "IPsec 및 Oracle Solaris 영역" [96]을 참조하십시오.
- **IPsec 및 FIPS 140 모드** - IPsec 관리자는 Oracle Solaris에 대한 FIPS 140 검증 알고리즘을 선택해야 합니다. 이 장의 절차 및 예제에는 `any` 알고리즘이 지정되지 않은 한 FIPS 140 승인 알고리즘이 사용됩니다.
- **IPsec 및 RBAC** - 역할을 사용하여 IPsec을 관리하려면 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 3 장, "Oracle Solaris에서 권한 지정"을 참조하십시오. 예는 네트워크 보안에 대한 역할을 구성하는 방법 [116]을 참조하십시오.

- **IPsec 및 SCTP** -IPsec을 사용하여 SCTP(Streams Control Transmission Protocol) 연관을 보호할 수 있지만, 주의해야 합니다. 자세한 내용은 "IPsec 및 SCTP" [95]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 유형의 구성을 사용하면 IPv4 네트워크를 통해 IPv6을 터널링할 수 있습니다.

다음 작업 맵에서는 하나 이상의 시스템 사이에 IPsec을 설정하는 절차를 안내합니다. `ipsecconf`(1M), `ipseckey`(1M) 및 `ipadm`(1M) 매뉴얼 페이지의 해당 예제 절에서도 유용한 절차를 설명합니다.

표 7-1 IPsec을 사용하여 네트워크 트래픽 보호 작업 맵

작업	설명	지침
두 시스템 사이의 트래픽을 보호합니다.	한 시스템에서 다른 시스템으로의 패킷을 보호합니다.	IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]
IPsec 정책을 사용하여 웹 서버를 보호합니다.	비웹 트래픽에서 IPsec를 사용하도록 합니다. 웹 클라이언트는 IPsec 검사를 우회하는 특정 포트로 식별됩니다.	IPsec을 사용하여 웹 서버와 다른 서버의 통신을 보호하는 방법 [104]
IKE를 사용하여 IPsec SA에 대한 키 입력 자료를 자동으로 만듭니다.	IPsec SA를 만들 때 권장되는 방법입니다.	"IKEv2 구성" [133] 및 "IKEv1 구성" [159]
보안 VPN(virtual private network)을 설정합니다.	인터넷을 거치는 두 시스템 사이에 IPsec를 설정합니다.	"IPsec를 사용하여 VPN 보호" [106]
수동 키 관리를 설정합니다.	IKE를 사용하지 않고 IPsec SA에 대한 원시 데이터를 제공합니다.	IPsec 키를 수동으로 만드는 방법 [114]

▼ IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법

이 절차에서는 다음 설정을 가정합니다.

- 시스템에 정적 IP 주소가 지정되어 있고 네트워크 구성 프로파일 `DefaultFixed`가 실행 중입니다. `netadm list` 명령에서 `Automatic`을 반환하는 경우 자세한 내용은 `netcfg`(1M) 매뉴얼 페이지를 참조하십시오.
- 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. 이 절차에서는 IPv4 주소를 사용합니다.

- 각 시스템은 전역 영역이나 배타적 IP 영역입니다. 자세한 내용은 “IPsec 및 Oracle Solaris 영역” [96]을 참조하십시오.
- 각 시스템에서는 트래픽을 AES 알고리즘을 사용하여 암호화하고 SHA-2를 사용하여 인증합니다.

참고 - 일부 사이트에서는 SHA-2 알고리즘이 필요할 수 있습니다.

- 각 시스템은 공유 보안 연관을 사용합니다.

  공유 SA를 사용하여 두 시스템을 보호하는 데 한 쌍의 SA만 필요합니다.

참고 - 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 특정 권한이 있는 사용자는 `root`가 아니어도 다음 명령을 실행할 수 있습니다.

- 구성 명령을 실행하려면 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다.
- 이 관리 역할에서는 `pfedit` 명령을 사용하여 IPsec 관련 시스템 파일을 편집하고 키를 만들 수 있습니다.
- `hosts` 파일을 편집하려면 `root` 역할이거나 해당 파일을 편집할 수 있는 명시적 권한이 있어야 합니다. 예 7-7. “신뢰할 수 있는 사용자가 IPsec을 구성 및 관리하도록 허용”을 참조하십시오.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

원격으로 관리하는 경우 예 7-1. “ssh 연결을 사용하여 IPsec 정책을 원격으로 구성” 및 “Oracle Solaris 11.2의 보안 셸 액세스 관리 ”의 “보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법”에서 보안 원격 로그인 지침을 참조하십시오.

1. **각 시스템에서 호스트 항목을 `/etc/inet/hosts` 파일에 추가합니다.**

   이 단계를 사용하면 네트워크 이름 지정 서비스를 사용하지 않고도 로컬 이름 지정 서비스에서 시스템 이름을 IP 주소로 확인할 수 있습니다.

   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.

   ```
   # pfedit /etc/inet/ipsecinit.conf
   ```

3. **IPsec 정책 항목을 `ipsecinit.conf` 파일에 추가합니다.**

   IPsec 정책 항목의 구문과 몇 가지 예는 `ipsecconf`(1M) 매뉴얼 페이지를 참조하십시오.

   a. **`enigma` 시스템에서 다음 정책을 추가합니다.**

      ```
      {laddr enigma raddr partym} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
      ```

      `dir` 키워드를 사용하지 않았으므로 정책이 아웃바운드 및 인바운드 패킷 모두에 적용됩니다.

   b. **`partym` 시스템에서 동일한 정책을 추가합니다.**

      ```
      {laddr partym raddr enigma} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
      ```

4. **각 시스템에서 IKE를 구성하여 IPsec SA를 관리합니다.**

   "IKEv2 구성" [133]에 있는 구성 절차 중 하나를 따릅니다. IKE 구성 파일의 구문은 `ikev2.config`(4) 매뉴얼 페이지를 참조하십시오. IKEv1 프로토콜만 지원하는 시스템과 통신하는 경우 "IKEv1 구성" [159] 및 `ike.config`(4) 매뉴얼 페이지를 참조하십시오.

   참고 - 키를 수동으로 생성하고 유지 관리해야 하는 경우 IPsec 키를 수동으로 만드는 방법 [114]을 참조하십시오.

5. **IPsec 정책 파일의 구문을 확인합니다.**

   ```
   % pfbash
   # /usr/sbin/ipsecconf -c /etc/inet/ipsecinit.conf
   ```

   오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

6. **IPsec 정책을 새로 고칩니다.**

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

   IPsec 정책은 기본적으로 사용으로 설정되므로 새로 고칩니다. IPsec 정책을 사용 안함으로 설정한 경우 사용으로 설정합니다.

   ```
   # svcadm enable ipsec/policy:default
   ```

7. **IPsec에 대한 키를 활성화합니다.**

   - **`ike` 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.**

참고 - IKEv1 프로토콜만 실행할 수 있는 시스템과 통신하는 경우에는 `ike:default` 인스턴스를 지정합니다.

```
# svcadm enable ipsec/ike:ikev2
```

■ **ike 서비스가 사용으로 설정된 경우 다시 시작합니다.**

```
# svcadm restart ike:ikev2
```

4단계에서 키를 수동으로 구성한 경우 IPsec 키를 수동으로 만드는 방법 [114]의 절차를 완료하여 키를 활성화합니다.

8. **패킷이 보호되고 있는지 확인합니다.**
   절차는 IPsec로 패킷이 보호되는지 확인하는 방법 [119]을 참조하십시오.

예 7-1 ssh 연결을 사용하여 IPsec 정책을 원격으로 구성

이 예에서는 `root` 역할의 관리자가 두 시스템에서 `ssh` 명령으로 두번째 시스템에 연결하여 IPsec 정책 및 키를 구성합니다. 관리자는 두 시스템에서 동일하게 정의됩니다. 자세한 내용은 `ssh`(1) 매뉴얼 페이지를 참조하십시오.

1. 관리자는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]의 1단계 ~ 5단계를 수행하여 첫번째 시스템을 구성합니다.
2. 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: xxxxxxxx
other-system #
```

3. `ssh` 세션의 터미널 창에서 관리자는 1단계 ~ 7단계를 완료하여 두번째 시스템의 IPsec 정책 및 키를 구성합니다.
4. 관리자는 `ssh` 세션을 종료합니다.

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

5. 관리자는 6단계 및 7단계를 완료하여 첫번째 시스템에서 IPsec 정책을 사용으로 설정합니다.

다음에 `ssh` 연결을 사용하여 통신하는 경우를 비롯해 두 시스템이 통신할 때 통신이 IPsec으로 보호됩니다.

예 7-2 FIPS 140 모드로 실행하도록 IPsec 정책 구성

이 예제에서 관리자는 키 길이가 최소 192비트인 대칭적 알고리즘이 필요한 사이트 보안 정책을 따르도록 FIPS 140 사용 시스템에서 IPsec 정책을 구성합니다.

관리자는 두 가지 가능한 IPsec 정책을 지정합니다. 첫번째 정책은 암호화 및 인증을 위해 CCM 모드로 AES를 지정하고, 두번째 정책은 암호화를 위해 키 길이가 192비트 및 256비트인 AES를 지정하고 인증을 위해서는 SHA384를 지정합니다.

```
{laddr machine1 raddr machine2} ipsec {encr_algs aes-ccm(192...) sa shared} or ipsec
{laddr machine1 raddr machine2} ipsec {encr_algs aes(192...) encr_auth_algs sha2(384) sa
shared}
```

▼ IPsec을 사용하여 웹 서버와 다른 서버의 통신을 보호하는 방법

웹 서비스를 실행하는 시스템에서 IPsec을 사용하여 웹 클라이언트 요청을 제외한 모든 트래픽을 보호할 수 있습니다. 일반적으로 웹 서버와 다른 백엔드 서버 간의 네트워크 트래픽이 보호됩니다.

이 절차의 IPsec 정책에서는 웹 클라이언트가 IPsec을 우회하도록 허용할 뿐 아니라 서버가 DNS 클라이언트 요청을 하도록 허용합니다. 다른 트래픽은 모두 IPsec으로 보호됩니다.

시작하기 전에 이 절차에서는 두 서버에서 IPsec을 구성하는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]의 단계를 완료했다고 가정하므로 다음 조건이 적용됩니다.

- 각 시스템은 주소가 고정된 전역 영역이나 배타적 IP 영역입니다. 자세한 내용은 “IPsec 및 Oracle Solaris 영역” [96]을 참조하십시오.
- 웹 서버와의 통신은 이미 IPsec으로 보호되고 있습니다.
- 키 입력 자료가 IKE에 의해 생성됩니다.
- 패킷이 보호되고 있는지 확인했습니다.

특정 권한이 있는 사용자는 `root`가 아니어도 이러한 명령을 실행할 수 있습니다.

- 구성 명령을 실행하려면 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다.
- IPsec 관련 시스템 파일을 편집하고 키를 만들려면 `pfedit` 명령을 사용합니다.
- `hosts` 파일을 편집하려면 `root` 역할이거나 해당 파일을 편집할 수 있는 명시적 권한이 있어야 합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

원격으로 관리하는 경우 예 7-1. “ssh 연결을 사용하여 IPsec 정책을 원격으로 구성” 및 “Oracle Solaris 11.2의 보안 셸 액세스 관리 ”의 “보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법”에서 보안 원격 로그인 지침을 참조하십시오.

1. **IPsec 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. **IPsec 정책 파일에 웹 서버 정책을 추가합니다.**
   `ipsecinit.conf` 파일에 다음 라인을 추가합니다.

   ```
   # pfedit /etc/inet/ipsecinit.conf
   ...
   # 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 1단계에서 설명한 우회 예외 사항과 함께 보안 트래픽만 시스템에 액세스할 수 있도록 허용합니다.

3. **IPsec 정책 파일의 구문을 확인합니다.**

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

4. **IPsec 정책을 새로 고칩니다.**

   ```
   # svcadm refresh ipsec/policy
   ```

5. **IPsec에 대한 키를 새로 고칩니다.**
   `ike` 서비스를 다시 시작합니다.

   ```
   # svcadm restart ike:ikev2
   ```

   참고 - IKEv1 프로토콜만 실행할 수 있는 시스템과 통신하는 경우에는 `ike:default` 인스턴스를 지정합니다.

   키를 수동으로 구성한 경우 IPsec 키를 수동으로 만드는 방법 [114]의 지침을 따릅니다.
   설정이 완료되었습니다.

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
##
{raddr webserver} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

구문을 확인한 다음 IPsec 정책을 새로 고쳐 활성화합니다.

```
remote-system # ipsecconf -c /etc/inet/ipsecinit.conf
remote-system # svcadm refresh ipsec/policy
```

원격 시스템은 시스템의 IPsec 정책이 일치할 경우에만 비웹 트래픽에 대해 웹 서버와 안전하게 통신할 수 있습니다.

7. **(옵션) 터널별 항목을 포함하여 일치하는 순서대로 IPsec 정책 항목을 표시합니다.**

```
# ipsecconf -L -n
```

IPsec를 사용하여 VPN 보호

IPsec을 사용하여 VPN을 보호할 수 있습니다. 배경 정보는 "IPsec의 전송 및 터널 모드" [91]를 참조하십시오. 이 절의 예와 절차에서는 IPv4 주소를 사용하지만, 예와 절차는 IPv6 VPN에도 적용됩니다. 간략한 설명은 "IPsec을 사용하여 네트워크 트래픽 보호" [99]를 참조하십시오.

터널 모드에 대한 IPsec 정책의 예는 "터널 모드를 사용하여 IPsec로 VPN을 보호하는 예" [106]를 참조하십시오.

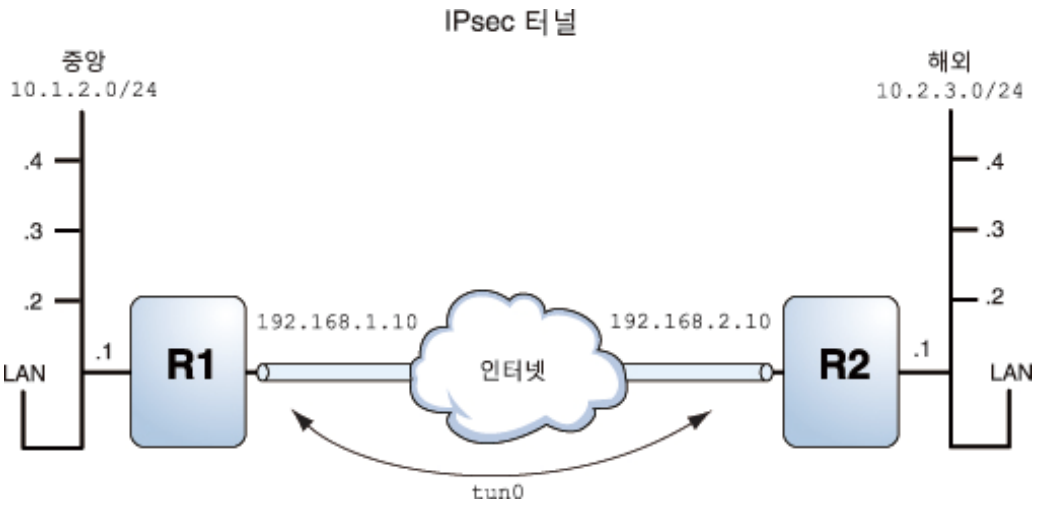
터널 모드를 사용하여 IPsec로 VPN을 보호하는 예

다음 그림의 터널은 다음과 같이 LAN의 모든 서브넷에 대해 구성됩니다.

```
## Tunnel configuration for ##
# 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
```

그림 7-1 IPsec로 보호되는 터널



다음 예는 위 그림을 기반으로 합니다.

예 7-3 모든 서브넷에서 사용할 수 있는 터널 만들기

그림 7-1. "IPsec로 보호되는 터널"에 나온 Central LAN 로컬 LAN의 모든 트래픽이 Router 1을 거쳐 Router 2로 터널링된 다음 Overseas LAN의 모든 로컬 LAN에 전달될 수 있습니다. 이 트래픽은 AES로 암호화됩니다.

```
## IPsec policy ##
{tunnel tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

예 7-4 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

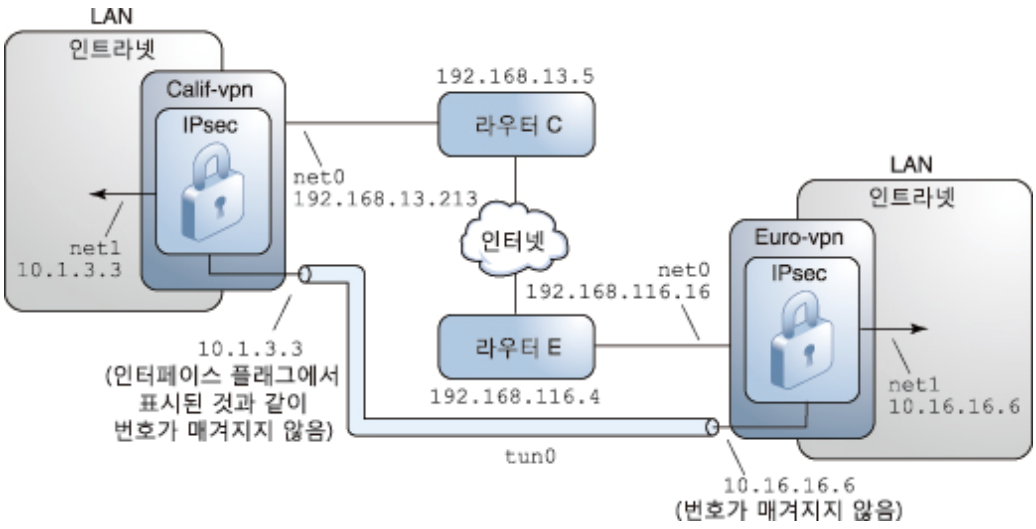
이 절의 절차에서는 다음 설정을 가정합니다. 네트워크 그림은 그림 7-2. “인터넷으로 연결된 사무실 사이의 샘플 VPN”를 참조하십시오.

- 각 시스템은 IPv4 주소 공간을 사용합니다.

  이러한 절차는 IPv6 주소 또는 IPv4 및 IPv6 주소의 조합에서도 작동합니다.
- 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 AES 알고리즘을 사용하는 ESP 암호화가 필요합니다. AES 알고리즘은 128비트 또는 256비트 키를 사용합니다.
- 각 시스템에는 SHA-2 알고리즘을 사용하는 ESP 인증이 필요합니다. 이 예에서 SHA-2 알고리즘은 512비트 키를 사용합니다.
- 각 시스템은 인터넷에 직접 액세스되는 라우터에 연결할 수 있습니다.
- 각 시스템은 공유 보안 연관을 사용합니다.

다음 그림에서는 절차에 사용되는 구성 매개변수를 보여줍니다.

그림 7-2 인터넷으로 연결된 사무실 사이의 샘플 VPN



구성 매개변수는 다음 표와 같습니다.

매개변수	유럽	캘리포니아
시스템 이름	euro-vpn	calif-vpn
시스템 인트라넷 인터페이스	net1	net1
시스템 인트라넷 주소, 다른 네트워크의 기본 경로	10.16.16.6	10.1.3.3
시스템 인트라넷 주소 객체	net1/inside	net1/inside
시스템 인터넷 인터페이스	net0	net0
시스템 인터넷 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 11.2의 TCP/IP 네트워크, IPMP 및 IP 터널 관리 ”의 “IP 터널 관리”를 참조하십시오. 주소 객체에 대한 자세한 내용은 “Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리 ”의 “IPv4 인터페이스를 구성하는 방법” 및 `ipadm`(1M) 매뉴얼 페이지를 참조하십시오.

▼ 터널 모드에서 IPsec을 사용하여 두 LAN 사이의 연결을 보호하는 방법

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

이 절차는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100] 절차를 확장합니다. 설정은 “VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명” [108]에 설명되어 있습니다.

특정 명령을 실행하는 이유에 대한 자세한 설명은 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]에서 해당하는 단계를 참조하십시오.

참고 - 두 시스템에서 이 절차의 단계를 수행하십시오.

두 시스템 연결과 함께 이러한 두 시스템에 연결되는 두 인트라넷을 연결하게 됩니다. 이 절차에서 시스템은 게이트웨이로 작동합니다.

참고 - 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 각 시스템은 전역 영역이나 배타적 IP 영역입니다. 자세한 내용은 “IPsec 및 Oracle Solaris 영역” [96]을 참조하십시오.

특정 권한이 있는 사용자는 root가 아니어도 이러한 명령을 실행할 수 있습니다.

- 구성 명령을 실행하려면 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다.
- IPsec 관련 시스템 파일을 편집하고 키를 만들려면 pfedit 명령을 사용합니다.
- hosts 파일을 편집하려면 root 역할이거나 해당 파일을 편집할 수 있는 명시적 권한이 있어야 합니다.

자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 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 사용 안함으로 설정하면 패킷이 이 시스템을 통해 한 네트워크에서 다른 네트워크로 전달되지 않습니다. routeadm 명령에 대한 설명은 routeadm(1M) 매뉴얼 페이지를 참조하십시오.

   b. **IP 엄격한 멀티홈을 사용으로 설정합니다.**

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

   IP 엄격한 멀티홈을 사용으로 설정하면 시스템의 대상 주소 중 하나에 대한 패킷이 올바른 대상 주소에 도달해야 합니다.

   hostmodel 매개변수가 strong으로 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.**

   ssh 서비스가 실행 중인지 확인합니다.

   ```
   % svcs | grep network
   …
   online         Aug_09   svc:/network/ssh:default
   ```

2. **VPN에 대한 IPsec 정책을 /etc/inet/ipsecinit.conf 파일에 추가합니다.**

   추가 예는 "터널 모드를 사용하여 IPsec로 VPN을 보호하는 예" [106]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 추가합니다.**

```
# 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 추가합니다.**

```
# 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.**

   "IKEv2 구성" [133]의 구성 절차 중 하나에 따라 IKE를 구성합니다. IKE 구성 파일의 구문은 ikev2.config(4) 매뉴얼 페이지를 참조하십시오. IKEv1 프로토콜만 지원하는 시스템과 통신하는 경우 "IKEv1 구성" [159] 및 ike.config(4) 매뉴얼 페이지를 참조하십시오.

참고 - 키를 수동으로 생성하고 유지 관리해야 하는 경우 IPsec 키를 수동으로 만드는 방법 [114]을 참조하십시오.

---

4. **IPsec 정책 파일의 구문을 확인합니다.**

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

   오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

5. **IPsec 정책을 새로 고칩니다.**

```
# svcadm refresh ipsec/policy
```

   IPsec 정책은 기본적으로 사용으로 설정되므로 새로 고칩니다. IPsec 정책을 사용 안함으로 설정한 경우 사용으로 설정합니다.

```
# svcadm enable ipsec/policy
```

6. **터널 tun0을 만들고 구성합니다.**

   다음 명령은 내부 및 외부 인터페이스를 구성하고, tun0 터널을 만들며, IP 주소를 터널에 지정합니다.

a. **calif-vpn 시스템에서 터널을 만들고 구성합니다.**

```
# ipadm create-ip net1
# ipadm create-addr -T static -a local=10.1.3.3 net1/inside
# dladm create-iptun -T ipv4 -a local=192.168.13.213,remote=192.168.116.16 tun0
# ipadm create-ip tun0
# ipadm create-addr -T static \
-a local=10.1.3.3,remote=10.16.16.6 tun0/v4tunaddr
```

첫번째 명령은 IP 인터페이스 net1을 만듭니다. 두번째 명령은 net1에 주소를 추가합니다. 세번째 명령은 IP 인터페이스 tun0을 만듭니다. 네번째 명령은 캡슐화된 IP 주소를 터널 링크에 추가합니다. 자세한 내용은 dladm(1M) 및 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

b. **euro-vpn 시스템에서 터널을 만들고 구성합니다.**

```
# ipadm create-ip net1
# ipadm create-addr -T static -a local=10.16.16.6 net1/inside
# dladm create-iptun -T ipv4 -a local=192.168.116.16,remote=192.168.13.213 tun0
# ipadm create-ip tun0
# ipadm create-addr -T static \
-a local=10.16.16.6,remote=10.1.3.3 tun0/v4tunaddr
```

참고 - 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 11.2 네트워크 구성 요소의 구성 및 관리 "의 "IPv4 인터페이스를 구성하는 방법"을 참조하십시오. 사용자 정의 이름에 대한 자세한 내용은 "Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리 "의 "Oracle Solaris의 네트워크 장치 및 데이터 링크 이름 지정"을 참조하십시오.

7. **각 시스템에서 전달을 구성합니다.**

```
# ipadm set-ifprop -m ipv4 -p forwarding=on net1
# ipadm set-ifprop -m ipv4 -p forwarding=on tun0
# 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 인터페이스에서 모두 IP 전달을 사용으로 설정해야 합니다.

net1 인터페이스는 인트라넷 내부에 있으므로 net1에 대해 IP 전달이 사용으로 설정되어 있어야 합니다. tun0은 인터넷을 통해 두 시스템을 연결하므로 tun0에 대해 IP 전달이 사용으로 설정되어 있어야 합니다. net0 인터페이스의 경우 인터넷의 외부 공격자가 보호된 인트라넷에 패킷을 주입하지 못하도록 IP 전달이 해제되어 있습니다.

8. **각 시스템에서 개인 인터페이스의 알림을 막습니다.**

```
# ipadm set-addrprop -p private=on net0
```

net0에 IP 전달이 사용 안함으로 설정되어 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.

추가 IPsec 작업

다음 작업 맵에는 IPsec을 관리할 때 사용할 수 있는 작업이 나옵니다.

표 7-2 추가 IPsec 작업 작업 맵

작업	설명	지침
IPsec SA를 수동으로 만들거나 대체합니다.	IPsec SA에 대한 원시 데이터를 제공합니다.	IPsec 키를 수동으로 만드는 방법 [114]
네트워크 보안 역할을 만듭니다.	보안 네트워크를 설정할 수 있지만 root 역할보다 권한이 적은 역할을 만듭니다.	네트워크 보안에 대한 역할을 구성하는 방법 [116]
네트워크 관리 작업을 모두 처리할 수 있는 권한 프로파일을 만듭니다.	네트워크 관리 작업은 수행할 수 있지만 root 역할보다 권한이 적은 역할을 만듭니다.	예 7-7. “신뢰할 수 있는 사용자가 IPsec을 구성 및 관리하도록 허용”
IPsec이 패킷을 보호하고 있는지 확인합니다.	IP 패킷이 어떻게 보호되는지를 나타내는 특정 헤더에 대한 snoop 출력을 검사합니다.	IPsec로 패킷이 보호되는지 확인하는 방법 [119]

작업	설명	지침
IPsec 및 키 입력 자료를 SMF 서비스의 일부로 관리합니다.	서비스를 사용으로 설정하고, 사용 안함으로 설정하고, 새로 고치고, 다시 시작합니다. 서비스의 등록 정보 값도 변경합니다.	"IPsec 및 수동 키 서비스 등록 정보 보기" [199]

▼ IPsec 키를 수동으로 만드는 방법

다음 절차에서는 키 관리에 IKE만 사용하지 않는 경우에 대한 IPsec 키를 제공합니다.

`ipseckey` 명령을 사용하여 추가된 IPsec SA는 시스템을 재부트하면 없어집니다. 지속 IPsec SA가 필요하면 항목을 `/etc/inet/secret/ipseckeys` 파일에 추가합니다.

주의 - 수동 키 입력을 사용해야 하는 경우에는 생성하는 키를 안전하게 보관하는 데 많은 주의가 필요합니다. 이러한 키는 데이터를 보호하는 데 실제로 사용됩니다.

시작하기 전에 공유 IP 영역에서 키 입력 도구를 수동으로 관리하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 해당 배타적 IP 영역에서 키 입력 도구를 구성합니다.

`root` 역할을 맡아야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오.

1. **IPsec SA에 대한 키를 생성합니다.**

   이 키는 `ipsecinit.conf` 파일의 특정 정책을 지원해야 합니다. 예를 들어, IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]의 정책을 사용할 수 있습니다.

   ```
   {laddr enigma raddr partym} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
   ```

   이 정책에서는 AES 및 SHA-2 알고리즘을 사용합니다.

   a. **필요한 키를 결정합니다.**

      SA에 대해 `aes`, `sha512` 및 SPI(보안 매개변수 색인)용 키를 생성해야 합니다.

      - SPI에 대한 값으로 2개의 16진수 난수. 하나는 아웃바운드 트래픽용입니다. 다른 하나는 인바운드 트래픽용입니다. 각 숫자 모두 최대 8자까지만 허용됩니다.
      - SHA-2 인증 알고리즘에 대한 2개의 16진수 난수. 각 숫자 모두 512자까지만 허용됩니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.
      - AES 암호화 알고리즘에 대한 2개의 16진수 난수. 각 숫자의 길이는 128자여야 합니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.

참고 - `ipsecalgs -l` 명령을 실행하면 알고리즘의 키 크기가 표시됩니다. 수동 키, 즉 SHA512 및 AES 알고리즘을 사용할 때는 이 절차를 따릅니다. 약한 알고리즘, 결합 모드 알고리즘 또는 GMAC 알고리즘은 수동 키에 사용하지 마십시오.

b. **필요한 키를 생성합니다.**

- 사이트에 임의 숫자 생성기가 있을 경우 생성기를 사용하십시오.
- "Oracle Solaris 11.2의 암호화 및 인증서 관리 "의 "pktool 명령을 사용하여 대칭 키를 생성하는 방법" 및 해당 절의 IPsec 예에 나온 대로 `pktool` 명령을 사용합니다.

2. **키를 IPsec의 수동 키 파일에 추가합니다.**

a. **`enigma` 시스템에서 `/etc/inet/secret/ipseckeys` 파일을 다음과 유사하게 표시되도록 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 aes \
   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 aes \
   auth_alg sha512  \
   encrkey dd325c5c137fb4739a55c9b3a1747baa... \
   authkey ad9ced7ad5f255c9a8605fba5eb4d2fd...
```

b. **읽기 전용 권한으로 파일을 보호합니다.**

```
# chmod 400 /etc/inet/secret/ipseckeys
```

`pfedit -s` 명령을 사용하여 `ipseckeys` 파일을 만든 경우 권한이 올바르게 설정되어 있습니다. 자세한 내용은 `pfedit`(1M) 매뉴얼 페이지를 참조하십시오.

c. **파일의 구문을 확인합니다.**

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

참고 - 두 시스템의 키는 동일해야 합니다.

3. **IPsec에 대한 키를 활성화합니다.**

■ **manual-key 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.**

```
% svcs manual-key
STATE          STIME    FMRI
disabled       Apr_10   svc:/network/ipsec/manual-key:default
# svcadm enable ipsec/manual-key
```

■ **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. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ 네트워크 보안에 대한 역할을 구성하는 방법

Oracle Solaris의 권한 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
...
Network Management:RO::Manage the host and network configuration...
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.**

```
% profiles -p "Network Management" info
```

```
...
cmd=/usr/sbin/dladm
cmd=/usr/sbin/dlstat
...
cmd=/usr/sbin/svcadm
cmd=/usr/sbin/svccfg
cmd=/usr/sbin/dumpcap
```

3. **사이트에서 네트워크 보안 역할의 범위를 결정합니다.**

   1단계의 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.
   - 네트워크 관리 및 보안을 처리하는 역할을 만들려면 Network Management 프로파일과 함께 Network Security 또는 Network IPsec Management 권한 프로파일을 사용합니다.

4. **역할을 만들고 하나 이상의 사용자에게 지정합니다.**

   단계는 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "역할 만들기" 및 예 7-7. "신뢰할 수 있는 사용자가 IPsec을 구성 및 관리하도록 허용"을 참조하십시오.

예 7-5 네트워크 관리 및 보안 역할 만들기 및 지정

이 예에서는 관리자가 역할에 Network Management 및 Network Security 등 두 개의 권한 프로파일을 지정합니다. 그런 다음 관리자는 신뢰할 수 있는 사용자에게 역할을 지정합니다.

```
# roleadd -c "Network Mgt and Security" \
-S ldap -K profiles="Network Management Plus" netmgtsec
# passwd netmgtsec
New Password: xxxxxxxx
Confirm password: xxxxxxxx
# usermod -R netmgtsec jdoe
```

사용자 `jdoe`가 `netmgtsec` 역할을 맡으면 `jdoe`에게 프로파일의 권한이 제공됩니다.

```
% su - netsecmgt
Password: xxxxxxxx
#
```

예 7-6 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.

1. 관리자는 첫번째 역할 이름을 LinkWifi로 지정합니다.
2. 관리자는 Network Wifi, Network Link Security 및 Network Management 권한 프로파일을 역할에 지정합니다.
3. 관리자는 LinkWifi 역할을 적절한 사용자에게 지정합니다.
4. 관리자는 두번째 역할 이름을 IPsec Administrator로 지정합니다.
5. 관리자는 Network IPsec Management 및 Network Management 권한 프로파일을 역할에 지정합니다.
6. 관리자는 IPsec Administrator 역할을 적절한 사용자에게 지정합니다.

예 7-7 신뢰할 수 있는 사용자가 IPsec을 구성 및 관리하도록 허용

이 예에서는 관리자가 한 사용자에게 IPsec을 구성 및 관리하는 책임을 줍니다.

관리자는 Network Management 및 IPsec Network Management 권한 프로파일뿐 아니라 `hosts` 파일을 편집할 수 있는 기능과 로그를 읽을 수 있는 기능을 사용자에게 제공합니다.

1. 관리자는 권한 프로파일을 두 개 만듭니다. 하나는 파일 편집용이고 다른 하나는 로그 읽기용입니다.

```
# profiles -p -S LDAP "Hosts Configuration"
profiles:Network Configuration> set desc="Edits root-owned network files"
...Configuration> add auth=solaris.admin.edit/etc/hosts
...Configuration> commit
...Configuration> end
...Configuration> exit

# profiles -p -S LDAP "Read Network Logs"
profiles:Read Network Logs> set desc="Reads root-owned network log files"
...Logs> add cmd=/usr/bin/more
...Logs:more>set privs={file_dac_read}:/var/user/ikeuser/*
...Logs:more>set privs={file_dac_read}:/var/log/ikev2/*
...Logs:more> set privs={file_dac_read}:/etc/inet/ike/*
...Logs:more> set privs={file_dac_read}:/etc/inet/secret/*
...Logs:more>end
...Logs> add cmd=/usr/bin/tail
...Logs:tail>set privs={file_dac_read}:/var/user/ikeuser/*
...Logs:tail>set privs={file_dac_read}:/var/log/ikev2/*
...Logs:tail>set privs={file_dac_read}:/etc/inet/ike/*
...Logs:tail> set privs={file_dac_read}:/etc/inet/secret/*
...Logs:tail>end
...Logs> add cmd=/usr/bin/page
...Logs:page>set privs={file_dac_read}:/var/user/ikeuser/*
...Logs:page>set privs={file_dac_read}:/var/log/ikev2/*
```

```
...Logs:page>set privs={file_dac_read}:/etc/inet/ike/*
...Logs:page> set privs={file_dac_read}:/etc/inet/secret/*
...Logs:page>end
...Logs> exit
```

권한 프로파일을 통해 사용자는 more, tail 및 page 명령을 사용하여 로그를 읽을 수 있습니다. cat 및 head 명령을 사용할 수 없습니다.

2. 관리자는 사용자가 IPsec 및 해당 키 입력 서비스에 대한 구성 및 관리 작업을 모두 수행하도록 허용하는 권한 프로파일을 만듭니다.

```
# profiles -p "Site Network Management"
profiles:Site Network Management> set desc="Handles all network files and logs"
...Management> add profiles="Network Management"
...Management> add profiles="Network IPsec Management"
...Management> add profiles="Hosts Configuraton"
...Management> add profiles="Read Network Logs"
...Management> commit; end; exit
```

3. 관리자는 프로파일에 대한 역할을 만들고, 역할에 암호를 지정하고, 네트워킹 및 보안에 대해 잘 아는 신뢰할 수 있는 사용자에게 역할을 지정합니다.

```
# roleadd -S LDAP -c "Network Management Guru" \
-m -K profiles="Site Network Management" netadm
# passwd netadm
Password: xxxxxxxx
Confirm password: xxxxxxxx
# usermod -S LDAP -R +netadm jdoe
```

4. 관리자는 아웃오브밴드로 jdoe에 역할 암호를 제공합니다.

▼ 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 이 접두어가 표시됩니다.
- ESP: 접두어는 암호화된 데이터가 보내지고 있음을 나타냅니다. encr_auth_alg 또는 encr_alg를 사용하여 트래픽을 보호하는 경우 이 접두어가 표시됩니다.

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

snoop 출력을 만들려면 root 역할을 있어야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **partym과 같은 한 시스템에서 root 역할을 맡습니다.**

```
% su -
Password: xxxxxxxx
#
```

2. **(옵션) SA에 대한 세부 정보를 표시합니다.**

   ```
   # ipseckey dump
   ```

   이 출력에는 사용되는 SA와 일치하는 SPI 값, 사용된 알고리즘, 키 등이 표시됩니다.

3. **이 시스템에서 원격 시스템의 패킷을 스누핑할 준비를 합니다.**

   partym의 터미널 창에서 enigma 시스템으로부터 패킷을 스누핑합니다.

   ```
   # snoop -d net0 -o /tmp/snoop_capture enigma
   Using device /dev/e1000g (promiscuous mode)
   ```

4. **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.

   ```
   partym% ssh enigma
   Password: xxxxxxxx
   enigma% su -
   Password: xxxxxxxx
   enigma# ping partym
   ```

5. **snoop 출력을 검사합니다.**

   ```
   partym# snoop -i /tmp.snoop_capture -v
   ```

   snoop 출력을 Wireshark 응용 프로그램으로 로드할 수도 있습니다. 자세한 내용은 문제 해결을 위해 IPsec 및 IKE 시스템을 준비하는 방법 [191] 및 “snoop 명령 및 IPsec” [211]를 참조하십시오.

   파일에서 초기 IP 헤더 정보 이후에 AH 및 ESP 정보가 포함된 출력이 표시되어야 합니다. AH 및 ESP 정보가 다음과 유사하면 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 IKE(Internet Key Exchange)

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

최신 버전의 IKE 프로토콜을 구현하는 방법에 대한 지침은 9장. IKEv2 구성을 참조하십시오. IKEv1을 계속 사용하려면 10장. IKEv1 구성을 참조하십시오. 참조 정보는 12장. IPsec 및 키 관리 참조를 참조하십시오. IPsec에 대한 내용은 6장. IP Security Architecture 정보를 참조하십시오.

IKE 소개

IPsec 보안 연관(SA)에 대한 키 관련 자료를 관리하는 것을 키 관리라고 합니다. 자동 키 관리를 위해서는 키 생성, 인증, 교환을 위한 통신 보안 채널이 필요합니다. Oracle Solaris는 IKE(Internet Key Exchange)를 사용하여 키 관리를 자동화합니다. IKE는 보안 키를 수동으로 배포할 때 발생하는 관리 오버헤드와 보안 위험을 제거합니다.

IKE는 사용 가능한 하드웨어 암호화 가속 및 키 저장소를 활용할 수 있습니다. 하드웨어 암호화 가속기를 사용하면 CPU를 많이 사용하는 키 작업을 시스템 외부에서 처리할 수 있습니다. 하드웨어의 키 저장소는 추가적 보호 계층을 제공합니다.

Oracle Solaris는 두 가지 버전의 IKE 프로토콜을 지원합니다.

- IKE 버전 2(IKEv2) - Internet Key Exchange 프로토콜 버전 2(IKEv2), RFC 5996을 기반으로 합니다.
- IKE 버전 1(IKEv1) - IKE(Internet Key Exchange), RFC 2409를 기반으로 합니다.

IKE 개념 및 용어

다음 개념 및 용어는 두 버전의 IKE에 공통됩니다. 이들 개념과 용어는 두 버전에서 다르게 구현될 수 있습니다.

- **키 협상 및 교환** - 피어의 ID에 대한 키 입력 도구 및 인증을 안전한 방식으로 교환하는 작업입니다. 이 프로세스에서는 비대칭 암호화 알고리즘을 사용합니다. 두 가지 주요 방법은 RSA 및 Diffie-Hellman 프로토콜입니다.

  IKE는 IKE 데몬을 실행 중인 시스템 사이에서 IPsec SA를 만들고 관리합니다. IKE는 키 입력 도구의 전송을 보호하는 보안 채널을 협상합니다. 데몬은 `/dev/random` 장치를 사용하여 난수 생성기로부터 키를 만듭니다. 데몬이 구성 가능한 비율로 키를 변경합니다. IPsec 정책용 구성 파일인 `ipsecinit.conf`에 지정된 알고리즘에서 키 관련 자료를 사용할 수 있습니다.
- **DH(Diffie-Hellman) 알고리즘** - 두 시스템이 비보안 채널을 통해 공유 암호를 안전하게 생성할 수 있는 키 교환 알고리즘입니다.
- **RSA 알고리즘** - 일반적으로 X.509 인증서의 소유권을 제공하여 피어 시스템의 ID를 인증하는 데 사용되는 비대칭 키 알고리즘입니다. 알고리즘 이름은 Rivest, Shamir, Adleman 등 3인의 저작자 이름에서 따왔습니다.

  또는 DSA나 ECDSA 알고리즘이 이 용도로 사용될 수 있습니다.
- **PFS(Perfect Forward Secrecy)** - PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송을 보호하는 키의 소스도 추가 키를 파생하는 데 사용되지 않습니다. 따라서 PFS는 이전에 기록된 트래픽의 해독을 방지할 수 있습니다.
- **Oakley 그룹** - PFS를 협상하는 데 사용됩니다. The Internet Key Exchange (IKE) RFC의 6절을 참조하십시오.
- **IKE 정책** - 피어 시스템과의 보안 키 교환 채널을 설정할 때 IKE 데몬이 사용하는 허용 가능한 매개변수를 정의하는 IKE 규칙 세트입니다. IKEv2에서는 IKE SA, IKEv1에서는 1단계라고 합니다.

  매개변수에는 알고리즘, 키 크기, Oakley 그룹 및 인증 방법이 포함됩니다. Oracle Solaris IKE 데몬은 미리 공유한 키 및 인증서를 인증 방법으로 지원합니다.

IKE 작동 방식

IKE 데몬을 실행 중인 시스템은 이 시스템과 IKE 데몬을 실행 중인 다른 시스템 간의 보안 연관(SA)을 만드는 데 필요한 매개변수를 협상할 수 있습니다. 이 SA 및 후속 IPsec SA를 협상하는 데 사용되는 프로토콜을 IKE라고 합니다. 이 Oracle Solaris 버전에서는 IKE 프로토콜의 버전 1(IKEv1) 및 버전 2(IKEv2)를 지원합니다.

IKE 보안 연관(IKEv1에서는 ISAKMP 또는 1단계 SA라고도 함)을 통해 이러한 두 IKE 시스템 간의 프로토콜 교환이 추가로 보호됩니다. 이러한 교환에서는 암호화 알고리즘, IPsec 정책 및 IPsec SA를 만드는 데 필요한 기타 매개변수를 협상합니다.

IKE 데몬을 실행 중인 시스템은 다른 시스템 대신 IPsec SA를 협상하도록 구성할 수도 있습니다. 이 방법으로 구성한 시스템을 보안 게이트웨이라고 합니다. IKE 협상이 성공하면 IPsec SA를 사용하여 네트워크 패킷을 보호할 수 있습니다.

참고 - Oracle Solaris 11.2에서, IKEv2는 FIPS 140-2, 레벨 1에 대해 검증된 암호화 프레임워크의 암호화 알고리즘을 사용하지만 IKEv1의 경우는 해당되지 않습니다. 기본적으로 FIPS 140은 사용으로 설정되어 있지 않습니다. 두 버전의 기능을 비교하려면 “IKEv2 및 IKEv1 비교” [128]를 참조하십시오. FIPS 140-2 모드를 사용으로 설정하려면 “Oracle Solaris 11.2의 암호화 및 인증서 관리 ”의 “FIPS 140이 사용으로 설정된 부트 환경을 만드는 방법”을 참조하십시오.

FIPS 140-2 검증 암호화만 사용하도록 요구 사항이 엄격한 경우, Oracle Solaris 11.1 SRU 5.5 릴리스 또는 Oracle Solaris 11.1 SRU 3 릴리스를 실행해야 합니다. Oracle은 이 두 가지 릴리스에서 암호화 프레임워크에 대한 FIPS 140-2 검증을 마쳤습니다. Oracle Solaris 11.2는 이러한 검증을 기초로 제작되었으며 성능, 기능 및 안정성 문제를 해결하는 소프트웨어 향상 기능을 포함합니다. 이러한 향상 기능을 활용하기 위해서는 가능한 모든 경우에 Oracle Solaris 11.2를 FIPS 140-2 모드로 구성해야 합니다.

IKE SA를 만들기 위해 협상하는 매개변수에는 IKE 교환 및 일부 인증 자료를 보호하는 암호화 알고리즘이 포함됩니다. 인증 자료는 IKE 프로토콜 교환을 포함하는 패킷을 신뢰할 수 있는지 여부를 결정하는 데 사용됩니다. 신뢰할 수 있는 경우 신뢰할 수 있는 시스템으로 가장하는 시스템이 아니라 신뢰할 수 있는 시스템에서 패킷을 가져왔음을 의미합니다.

Oracle Solaris에서는 IKE에 대해 미리 공유한 키와 공개 키 인증서라는 두 가지 유형의 인증 재료를 지원합니다.

IKE와 미리 공유한 키 인증

미리 공유한 키는 두 IKE 시스템만 아는 16진수 또는 ASCII 문자의 문자열입니다. 미리 공유한 키라고 하는 이유는 두 끝점이 모두 키 값을 알고 있어야만 IKE 교환이 이루어지기 때문입니다. 두 시스템 모두에서 이 키가 IKE 구성에 포함되어야 합니다. 미리 공유한 키는 IKE 페이로드 생성 시 사용되며, 이 페이로드는 IKE 프로토콜을 구현하는 패킷을 구성합니다. 이러한 IKE 페이로드를 처리하는 시스템은 수신하는 페이로드를 동일한 키를 사용하여 인증합니다.

미리 공유한 키는 IKE 끝점 간에 IKE 프로토콜을 사용하여 교환되지 않습니다. 일반적으로 이 키는 전화 통화와 같은 다른 매체를 통해 피어 시스템과 공유됩니다.

이 인증 방법을 사용하는 피어의 미리 공유한 키는 동일해야 합니다. 키는 각 시스템에서 파일에 저장됩니다.

IKE와 공개 키 인증서

공개 키 인증서 및 해당 트러스트 체인에서는 보안 정보를 수동으로 교환하지 않고도 시스템을 디지털 방식으로 식별하는 방식을 제공합니다. 따라서 공개 키 인증서가 미리 공유한 키보다 더 안전합니다.

공개 키 인증서는 공개 키 값, 인증서 생성에 대한 일부 정보(예: 이름 및 인증서에 서명한 사람), 인증서의 해시 또는 체크섬 및 해시의 디지털 서명을 인코딩하는 데이터 blob입니다. 이러한 값이 결합되어 인증서가 형성됩니다. 디지털 서명이 있으면 인증서가 수정되지 않은 것입니다.

공개 키는 개인 키라는 다른 값에서 수학적으로 파생되는 값입니다. 수학 알고리즘을 통해 개인 키에서 공개 키를 파생하므로 공개 키에서 개인 키를 검색하는 작업은 비실용적입니다. 따라서 공개 키 인증서는 자유롭게 공유할 수 있습니다. 공개 키를 파생하는 데 사용되는 알고리즘의 예로는 RSA, 타원 곡선 등이 있습니다.

디지털 서명은 RSA, DSA 또는 ECDSA와 같은 디지털 서명 알고리즘을 통해 인증서 내용을 전달하면 생성됩니다. 이러한 알고리즘에서는 인증서에 포함되지 않은 개인 서명 키를 사용하고 디지털 서명을 생성합니다. 서명은 인증서에 추가됩니다. 마찬가지로 인증서 내용 및 서명에서 서명 키를 계산하는 작업은 비실용적입니다. 그보다는 인증서 서명이 더 중요하므로 서명 키에서 파생된 공개 키를 사용하여 인증서 내용을 쉽게 확인할 수 있습니다.

인증서에 자체 서명될 수 있으며, 이 경우 인증서 서명은 인증서의 공개 키로 확인할 수 있거나 다른 엔티티가 인증서에 서명할 수 있습니다. 다른 엔티티가 인증서에 서명하는 경우 인증서를 확인하는 데 사용되는 공개 키 값은 공개 키 인증서로도 배포됩니다. 이 두번째 인증서에는 신뢰할 수 있는 certificate authority(CA, 인증 기관)나 중개자가 서명합니다. 중개자는 궁극적으로 서명 엔티티, 즉 루트 CA 또는 trust anchor(트러스트 앵커)의 신뢰를 받습니다.

이러한 공개 키 인증서 구성 요소와 이 구성 요소를 구현하는 절차 및 구조를 대개 PKI(공개 키 기반구조)라고 합니다. 조직에 따라 PKI의 범위가 다를 수 있습니다. 단순한 PKI는 로컬 사용을 위한 인증서 몇 개에 서명하는 CA로 구성될 수 있습니다. 보다 광범위한 PKI에서는 전역으로 인식되는 트러스트 앵커를 권한 있는 CA로 사용합니다.

IKE에서 공개 키 인증서 사용

이 섹션에서는 IKE에서 공개 키 인증서를 만들고 사용하는 전체 단계를 설명합니다. 구체적인 절차는 “미리 공유한 키로 IKEv2 구성” [134] 및 “미리 공유한 키로 IKEv1 구성” [160]을 참조하십시오.

1. 자체 서명된 인증서 또는 CA(인증 기관)의 인증서를 사용하려면 먼저 공개/개인 키 쌍을 생성해야 합니다.
   - 자체 서명된 인증서의 경우에는 IKE 피어가 이러한 인증서를 교환하고 아웃오브밴드에서 정품 인증서인지 확인한 다음 피어의 인증서를 로컬 키 저장소로 가져옵니다. 그러면 키 저장소에는 원래 자체 서명된 인증서와 가져온 인증서가 포함됩니다.
   - CA의 인증서인 경우 몇 가지 단계를 더 수행합니다. 공개/개인 키 쌍을 생성할 때 인증서 서명 요청(CSR)도 생성합니다. CSR에는 공개 키와 식별자가 포함됩니다. 일반 식별자는 distinguished name(DN, 고유 이름)입니다. 예를 들면 다음과 같습니다.

     ```
     DN=”O=Example\, Inc, OU=qa, L=Silicon Valley, ST=CA, CN=enigma”
     ```

작은 정보 - 최대한 구체적인 DN 또는 기타 식별자를 만들어 다른 인증서의 식별자와 일치하지 않도록 하십시오.

2. CSR을 CA에 보내 서명을 받습니다.

   일반적으로 CSR을 웹 양식에 붙여 넣고 양식을 CA에 제출합니다. CA에서는 사용자에게 서명된 인증서를 두 개 이상 보낼 수 있습니다.
3. 서명된 인증서를 CA에서 가져온 다음 IKEv2 키 저장소나 IKEv1 데이터베이스로 가져옵니다.

   CA가 보내는 인증서를 모두 가져와야 합니다. 이러한 인증서는 트러스트 앵커나 루트 CA에서 개별적으로 식별된 서명 인증서에 이르는 "트러스트 체인"을 구성합니다.
4. IKE 피어에서 프로세스를 반복합니다.
5. IKE 규칙에서 인증서를 사용합니다.

   인증서를 지정할 때는 DN과 같은 식별자를 사용합니다. CA 서명된 인증서의 경우 특정 CA에서 서명한 인증서를 사용하도록 IKE를 구성할 수 있습니다.

해지된 인증서 처리

서명된 인증서는 서명 기관이 유효성을 보증하기 때문에 유효하다고 신뢰할 수 있습니다. 인증서가 손상되었거나 잘못된 상태라고 확인된 경우 CA는 인증서를 해지합니다.

CA에서는 해지한 인증서의 목록을 유지하며, 이 목록은 종종 CRL(인증서 해지 목록)이라고 합니다. OCSP(온라인 인증서 상태 프로토콜)를 사용하여 인증서의 상태를 동적으로 확인할 수 있습니다. 일부 공개 키 인증서에는 URI가 포함되어 있습니다. 이 URI는 CRL을 확인할 수 있는 웹 위치나 OCSP 서버의 웹 위치를 식별합니다.

자세한 내용은 RFC 2459: Certificate and CRL Profile 및 RFC 2560: Online Certificate Status Protocol - OCSP를 참조하십시오.

공개 인증서를 사용하는 시스템에서 시간 조정

공개 키 인증서에는 실행 날짜 및 시간과 인증서가 유효한 상태로 유지되는 시간이 포함되어 있습니다. 따라서 인증서를 생성하고 사용하는 시스템의 시계가 정확해야 합니다. NTP(Network Time Protocol) 소프트웨어를 사용하여 시스템의 시계를 동기화할 수 있습니다. Oracle Solaris 릴리스에는 University of Delaware의 NTP 공용 도메인 소프트웨어가 포함되어 있습니다. 설명서는 NTP Documentation 웹 사이트에서 사용할 수 있습니다. `service/network/ptp` 패키지를 설치하여 PTP(Precision Time Protocol) 서비스를 구성할 수도 있습니다. IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems를 참조하십시오.

IKEv2 및 IKEv1 비교

다음 표에서는 Oracle Solaris 시스템에서 IKEv2 및 IKEv1 버전의 구현을 비교합니다.

표 8-1 Oracle Solaris에서 IKEv2 및 IKEv1 구현

특징	IKEv2	IKEv1
인증서 chain of trust(트러스트 체인)	키 저장소의 객체를 기준으로 암시적	`ike/config` 파일의 `cert_trust` 매개변수
인증서 만들기	`ikev2cert` 명령	`ikecert certlocal` 명령
인증서 가져오기	`ikev2cert import` 명령으로 인증서와 키를 PKCS #11 키 저장소로 가져올 수 있음	`ikecert certdb` 명령으로 독립형 인증서를 IKE 키 저장소로 가져올 수 있음
인증서 소유자	`ikeuser`	`root`
인증서 정책 파일	`kmf-policy.xml`	`ike/config` 파일의 일부 정책
인증서 저장소	PKCS #11 softtoken 라이브러리	로컬 IKEv1 데이터베이스
구성 파일 디렉토리	`/etc/inet/ike/`	`/etc/inet/ike/` 및 `/etc/inet/secret/`
구성 소유자	`ikeuser` 계정	`root` 계정
데몬	`in.ikev2d`	`in.iked`
데몬 간 트래픽을 위한 FIPS 140 알고리즘 †	IKE SA에서 암호화 프레임워크 사용	일부 교환에서 암호화 프레임워크를 사용하지 않음
IPsec 트래픽에 대한 FIPS 140 알고리즘†	암호화 프레임워크 사용	암호화 프레임워크 사용
IKE 정책 파일	`ike/ikev2.config`	`ike/config`
IKE 미리 공유한 키	`ike/ikev2.preshared`	`secret/ike.preshared`
NAT 포트	UDP 포트 `4500`	UDP 포트 `4500`
포트	UDP 포트 `500`	UDP 포트 `500`
권한 프로파일	네트워크 IPsec 관리	네트워크 IPsec 관리
서비스 이름(FMRI)	`svc:/ipsec/ike:ikev2`	`svc:/ipsec/ike:default`

†Oracle Solaris 11.1 SRU 5.5 및 SRU 3의 암호화 프레임워크 기능은 FIPS 140-2, 레벨 1에 대해 검증되었습니다. FIPS 140 모드가 사용으로 설정되었고 암호화 프레임워크를 사용 중인 경우에는 FIPS 140에서 검증된 알고리즘이 사용됩니다. 기본적으로 FIPS 140 모드는 사용으로 설정되어 있지 않습니다.

IKEv2 프로토콜

이 섹션에서는 IKEv2 구현을 다룹니다. IKEv1 정보는 "IKEv1 프로토콜" [129]을 참조하십시오. 비교는 "IKEv2 및 IKEv1 비교" [128]를 참조하십시오. 두 프로토콜 모두에 적용되는 정보는 "IKE 소개" [123]를 참조하십시오. Oracle Solaris에서는 IKE 프로토콜의 두 버전을 동시에 지원합니다.

IKEv2 데몬 `in.ikev2d`는 IPsec SA에 대한 키 입력 도구를 협상하고 인증합니다. `in.ikev2d`(1M) 매뉴얼 페이지를 참조하십시오.

IKEv2 구성 선택

/etc/inet/ike/ikev2.config 구성 파일에는 in.ikev2d 데몬에 대한 구성이 포함됩니다. 구성은 여러 규칙으로 구성됩니다. 각 항목에는 이 시스템이 유사하게 구성된 IKEv2 피어에서 사용할 수 있는 알고리즘 및 인증 데이터와 같은 매개변수가 포함됩니다.

in.ikev2d 데몬에서는 미리 공유한 키(PSK) 및 공개 키 인증서를 ID로 사용할 수 있습니다.

ikev2.config(4) 매뉴얼 페이지에 샘플 규칙이 나옵니다. 각 규칙에는 고유한 레이블이 있어야 합니다. 다음은 매뉴얼 페이지에 있는 샘플 규칙에 대한 설명이 포함된 레이블의 목록입니다.

- IP identities and PSK auth
- IP address prefixes and PSK auth
- IPv6 address prefixes and PSK auth
- Certificate auth with DN identities
- Certificate auth with many peer ID types
- Certificate auth with wildcard peer IDs
- Override transforms
- Mixed auth types
- Wildcard with required signer

참고 - 미리 공유한 키는 IP 주소, DN, FQDN 및 전자 메일 주소를 비롯한 여러 피어 ID 유형 중 하나와 함께 사용할 수 있습니다.

공개 인증서에 대한 IKEv2 정책

kmf-policy.xml 파일에는 IKEv2에 대한 인증서 검증 정책이 포함됩니다. kmfcfg dbfile=/etc/inet/ike/kmf-policy.xml policy=default 명령은 인증서 검증 정책을 수정하는 데 사용됩니다. 일반적으로 수정하는 항목은 OCSP 및 CRL 사용, 인증서 검증 중 네트워크 시간 초과 기능 등입니다. 또한 정책을 사용하여 관리자는 유효 날짜 적용 및 키 사용 요구 사항과 같은 인증서 검증의 다양한 측면을 수정할 수 있습니다. 인증서 검증에 대한 기본 요구 사항은 완화하지 않는 것이 좋습니다.

IKEv1 프로토콜

다음 절에서는 IKEv1에 대해 간략히 살펴봅니다. IKEv1은 더 빠르고 안전한 키 관리를 제공하는 IKEv2로 교체되었습니다. IKEv2에 대한 자세한 내용은 "IKEv2 프로토콜" [128]을 참조하십시오. 비교는 "IKEv2 및 IKEv1 비교" [128]를 참조하십시오. 두 프로토콜에 공통

된 정보는 “IKE 소개” [123]를 참조하십시오. IKEv1과 IKEv2가 다른 시스템에서 동시에 실행되어 해당하는 피어 프로토콜과 협상할 수 있습니다.

IKEv1 키 협상

IKEv1 데몬 `in.iked`는 안전한 방식으로 키를 협상하고 IPsec SA를 인증합니다. IKEv1에서는 PFS(Perfect Forward Secrecy)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.

IKEv1 1단계 교환

IKEv1 프로토콜에는 2가지 단계가 있습니다. Oracle Solaris에서는 주 모드 1단계 교환을 지원합니다. 주 모드 교환에서는 허용 가능한 매개변수를 협상하여 두 피어 간의 ISAKMP 보안 연관(SA)을 만듭니다. 이 ISAKMP SA는 비대칭 암호화를 사용하여 키 입력 도구를 교환하고 미리 공유한 키나 공개 키 인증서를 사용하여 피어를 인증합니다. IPsec SA와 달리, ISAKMP SA는 양방향이므로 하나의 보안 연관만 필요합니다.

IKEv1이 1단계에서 ISAKAMP SA를 협상하는 방법을 구성할 수 있습니다. IKEv1은 `/etc/inet/ike/config` 파일에서 구성 정보를 읽습니다. 구성 정보는 다음과 같습니다.

- 공개 키 인증서 이름과 같은 전역 매개변수
- PFS(Perfect Forward Secrecy)가 필요한지 여부
- 이 시스템의 IKE 피어
- 1단계 교환을 보호하는 알고리즘
- 인증 방법

  두 가지 인증 방법은 미리 공유한 키와 공개 키 인증서입니다. 공개 키 인증서는 자체 서명되거나 certificate authority(CA, 인증 기관)에서 발행할 수 있습니다.

자세한 내용은 `ike.config`(4) 매뉴얼 페이지를 참조하십시오.

IKEv1 2단계 교환

2단계 교환은 빠른 모드라고 합니다. 빠른 모드 교환에서는 IPsec SA를 만드는 데 필요한 IPsec 알고리즘과 키 입력 도구를 협상합니다. 이 교환은 1단계에서 협상된 ISAKMP SA로 보호(암호화)됩니다.

빠른 모드 교환의 알고리즘 및 보안 프로토콜은 IPsec 정책 파일 `/etc/inet/ipsecinit.conf`에서 가져옵니다.

IPsec SA는 만료되면 다시 입력됩니다. SA의 수명은 in.iked 데몬이 IPsec SA를 만들 때 설정합니다. 이 값은 구성할 수 있습니다.

자세한 내용은 ipsecconf(1M) 및 in.iked(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 구성 선택

/etc/inet/ike/config 구성 파일에는 in.iked 데몬에 대한 구성이 포함됩니다. 구성은 여러 규칙으로 구성됩니다. 각 항목에는 이 시스템이 유사하게 구성된 IKEv1 피어에서 사용할 수 있는 알고리즘 및 인증 데이터와 같은 매개변수가 포함됩니다. in.iked 데몬에서는 미리 공유한 키 및 공개 키 인증서를 ID로 사용할 수 있습니다.

auth_method preshared 항목은 미리 공유한 키가 사용됨을 나타냅니다. auth_method의 값이 preshared가 아니면 공개 키 인증서가 사용됩니다.

IKEv1에서 미리 공유한 키는 특정 IP 주소나 주소 범위에 연결됩니다. 각 시스템의 /etc/inet/secret/ike.preshared 파일에 키가 저장됩니다.

자세한 내용은 "IKE 작동 방식" [124]과 ike.config(4) 및 ike.preshared(4) 매뉴얼 페이지를 참조하십시오.

♦ ♦ ♦ 9 장

9 IKEv2 구성

이 장에서는 시스템의 IKE(Internet Key Exchange) 버전 2(IKEv2)를 구성하는 방법에 대해 설명합니다. IKEv2를 구성하고 사용으로 설정하면 지정하는 IPsec 끝점의 키 입력 도구를 자동으로 생성합니다. 이 장은 다음 정보를 포함합니다.

IKE에 대한 개요 정보는 8장. IKE(Internet Key Exchange)를 참조하십시오. IKE에 대한 참조 정보는 12장. IPsec 및 키 관리 참조를 참조하십시오. 추가 절차는 `ikeadm`(1M), `pktool`(1), `ikev2cert`(1M), `ikev2.config`(4), `in.ikev2d`(1M) 및 `kmfcfg`(1) 매뉴얼 페이지의 예를 참조하십시오.

IKEv2 구성

미리 공유한 키, 자체 서명된 인증서 및 CA(인증 기관)의 인증서를 사용하여 IKE를 인증할 수 있습니다. 규칙에서는 특정 인증 방법을 보호되는 끝점과 연결합니다. 따라서 시스템에서 인증 방법 중 하나 또는 전체를 사용할 수 있습니다. IKEv2 시스템에서 IKEv1을 실행할 수도 있습니다. 일반적으로 IKEv2를 지원하지 않는 시스템과의 통신은 IKEv1을 실행하여 보호합니다. IKEv2에서는 키 및 인증서 저장소에 PKCS #11 하드웨어 토큰을 사용할 수도 있습니다.

참고 - 이러한 작업에서는 시스템에 정적 IP 주소가 지정되어 있고 네트워크 구성 프로파일 `DefaultFixed`를 실행 중이라고 가정합니다. `netadm list` 명령에서 `Automatic`을 반환하는 경우 자세한 내용은 `netcfg`(1M) 매뉴얼 페이지를 참조하십시오.

IKEv2를 구성한 후에는 7장. IPsec 구성에 나온 대로 이러한 IKEv2 규칙을 사용하여 키를 관리하는 IPsec 절차를 완료합니다. 다음 절에서는 특정 IKEv2 구성을 주로 설명합니다.

미리 공유한 키로 IKEv2 구성

IKEv2를 사용하도록 피어 시스템 또는 서브넷을 구성 중이며, 해당 서브넷의 관리자라면 미리 공유한 키를 사용하는 것이 좋습니다. 테스트할 때에도 미리 공유한 키를 사용할 수 있습니다. 자세한 내용은 "IKE와 미리 공유한 키 인증" [125]을 참조하십시오.

▼ 미리 공유한 키로 IKEv2를 구성하는 방법

이 절차에서는 시스템 이름을 `enigma` 및 `partym`으로 바꿉니다. IKE 끝점을 모두 구성합니다.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **각 시스템에서 `/etc/inet/ike/ikev2.config` 파일을 편집합니다.**

```
# pfedit /etc/inet/ike/ikev2.config
```

2. **파일에서 미리 공유한 키를 사용하는 규칙을 만듭니다.**

참고 - 키는 4단계에서 만듭니다.

이 파일의 규칙 및 전역 매개변수는 시스템의 `ipsecinit.conf` 파일에 있는 IPsec 정책의 키를 관리해야 합니다. 다음 IKEv2 구성 예에서는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]에 있는 `ipsecinit.conf` 예의 키를 관리합니다.

   a. **예를 들어 `enigma` 시스템에서 `ikev2.config` 파일을 수정합니다.**

참고 - 이 예에서는 전역 매개변수 섹션의 변환 두 가지를 보여줍니다. 이러한 변환 중 하나로 피어를 구성할 수 있습니다. 특정 변환을 요청하려면 해당 변환을 규칙에 포함하십시오.

```
### ikev2.config file on enigma, 192.168.116.16

## Global parameters
# This default value will apply to all transforms that follow
#
ikesa_lifetime_secs 3600
#
# Global transform definitions.  The algorithm choices are
```

```
# based on RFC 4921.
#
## Two transforms are acceptable to this system, Group 20 and Group 19.
## A peer can be configured with 19 or 20.
## To ensure that a particular peer uses a specific transform,
## include the transform in the rule.
##
# Group 20 is 384-bit ECP - Elliptic Curve over Prime
ikesa_xform { encr_alg aes(256..256) auth_alg sha384 dh_group 20 }
# Group 19 is 256-bit ECP
ikesa_xform { encr_alg aes(128..128) auth_alg sha256 dh_group 19 }
#
## The rule to communicate with partym
##  Label must be unique
{ label "enigma-partym"
  auth_method preshared
  local_addr  192.168.116.16
  remote_addr 192.168.13.213
}
```

b. **partym 시스템에서 ikev2.config 파일을 수정합니다.**

```
## ikev2.config file on partym, 192.168.13.213
## Global Parameters
#
...
ikesa_xform { encr_alg aes(256..256) auth_alg sha384 dh_group 20 }
ikesa_xform { encr_alg aes(128..128) auth_alg sha256 dh_group 19 }
...
## The rule to communicate with enigma
##  Label must be unique
{ label "partym-enigma"
  auth_method preshared
  local_addr  192.168.13.213
  remote_addr 192.168.116.16
}
```

3. **각 시스템에서 파일의 구문을 확인합니다.**

```
# /usr/lib/inet/in.ikev2d -c
```

4. **각 시스템에서 미리 공유한 키를 /etc/inet/ike/ikev2.preshared 파일에 넣습니다.**

주의 - 이 파일은 사용 권한이 특수하며 ikeuser가 소유합니다. 이 파일을 삭제하거나 대체하지 마십시오. 대신, pfedit 명령으로 내용을 편집하여 파일의 원래 등록 정보가 유지되도록 하십시오.

a. **예를 들어, enigma 시스템에서 ikev2.preshared 파일은 다음과 유사하게 표시됩니다.**

```
# pfedit -s /etc/inet/ike/ikev2.preshared
## ikev2.preshared on enigma, 192.168.116.16
```

```
#…
## label must match the rule that uses this key
{ label "enigma-partym"
## The preshared key can also be represented in hex
## as in 0xf47cb0f432e14480951095f82b
   key "This is an ASCII Cqret phrAz, use str0ng p@ssword tekniques"
}
```

pfedit 명령의 옵션에 대한 자세한 내용은 pfedit(1M) 매뉴얼 페이지를 참조하십시오.

b. **partym 시스템에서 ikev2.preshared 파일은 고유한 레이블을 제외하고는 다음과 유사합니다.**

```
## ikev2.preshared on partym, 192.168.13.213
#…
## label must match the label of the rule that uses this key
{ label "partym-enigma"
## 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. **IKEv2 서비스 인스턴스를 사용으로 설정합니다.**

```
# svcadm enable ipsec/ike:ikev2
```

미리 공유한 키를 대체할 경우 피어 시스템에서 미리 공유한 키 파일을 편집하고 ikev2 서비스를 다시 시작합니다.

```
# svcadm restart ikev2
```

예 9-1 다른 로컬 및 원격 IKEv2 미리 공유한 키 사용

이 예에서는 IKEv2 관리자가 시스템별로 미리 공유한 키를 만들고, 키를 교환하고, 각 키를 미리 공유한 키 파일에 추가합니다. 미리 공유한 키 항목의 레이블은 ikev2.config 파일에 있는 규칙의 레이블과 일치합니다. 그런 다음 in.ikev2d 데몬을 다시 시작합니다.

다른 시스템의 미리 공유한 키를 수신한 후 관리자는 ikev2.preshared 파일을 편집합니다. partym의 파일은 다음과 같습니다.

```
# pfedit -s /etc/inet/ike/ikev2.preshared
#…
{ label "partym-enigma"
## local and remote preshared keys
local_key  "P-LongISH key Th@t m^st Be Ch*angEd \'reguLarLy)"
remote_key "E-CHaNge lEyeGhtB+lBs et KeeS b4 2LoOoOoOoOng"
}
```

따라서 enigma의 ikev2.preshared 키 파일은 다음과 같아야 합니다.

```
#…
```

```
{ label "enigma-partym"
## local and remote preshared keys
local_key  "E-CHaNge lEyeGhtB+lBs et KeeS b4 2LoOoOoOoOng"
remote_key "P-LongISH key Th@t m^st Be Ch*angEd \'reguLarLy)"
}
```

관리자가 각 시스템에서 IKEv2 서비스 인스턴스를 다시 시작합니다.

```
# svcadm restart ikev2
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

더 많은 예는 `ikev2.config`(4) 및 `ikev2.preshared`(4) 매뉴얼 페이지를 참조하십시오.

▼ IKEv2에서 미리 공유한 키를 사용할 때 새 피어를 추가하는 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 IKEv2 구성도 수정해야 합니다.

시작하기 전에 `ipsecinit.conf` 파일을 업데이트했으며 피어 시스템에 대한 IPsec 정책을 새로 고쳤습니다.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **IPsec을 사용 중인 새 시스템의 키를 관리할 IKEv2에 대한 규칙을 만듭니다.**

   a. **예를 들어, `enigma` 시스템에서 `/etc/inet/ike/ikev2.config` 파일에 다음 규칙을 추가합니다.**

   ```
   # pfedit ikev2.config
   ## ikev2.config file on enigma, 192.168.116.16
   ...
   ## The rule to communicate with ada
   ```

```
##  Label must be unique
{label "enigma-ada"
 auth_method preshared
 local_addr  192.168.116.16
 remote_addr 192.168.15.7
}
```

pfedit 명령의 옵션에 대한 자세한 내용은 pfedit(1M) 매뉴얼 페이지를 참조하십시오.

b. **ada 시스템에서 다음 규칙을 추가합니다.**

```
## ikev2.config file on ada, 192.168.15.7
...
## The rule to communicate with enigma
{label "ada-enigma"
 auth_method preshared
 local_addr  192.168.15.7
 remote_addr 192.168.116.16
}
```

2. **(옵션) 각 시스템에서 파일의 구문을 확인합니다.**

```
# /usr/lib/inet/in.ikev2d -c -f /etc/inet/ike/ikev2.config
```

3. **피어 시스템에 대해 IKEv2 미리 공유한 키를 만듭니다.**

a. **enigma 시스템에서 /etc/inet/ike/ikev2.preshared 파일에 다음 정보를 추가합니다.**

```
# pfedit -s /etc/inet/ike/ikev2.preshared
## ikev2.preshared on enigma for the ada interface
...
## The rule to communicate with ada
##  Label must match the label of the rule
{ label "enigma-ada"
  # enigma and ada's shared key
  key "Twas brillig and the slivey toves did *s0mEtHiNg* be CareFULL hEEEr"
}
```

pfedit 명령의 옵션에 대한 자세한 내용은 pfedit(1M) 매뉴얼 페이지를 참조하십시오.

b. **ada 시스템에서 ikev2.preshared 파일에 다음 정보를 추가합니다.**

```
# ikev2.preshared on ada for the enigma interface
#
{ label "ada-enigma"
  # ada and enigma's shared key
  key "Twas brillig and the slivey toves did *s0mEtHiNg* be CareFULL hEEEr"
}
```

4. **각 시스템에서 변경 사항을 커널로 읽어들입니다.**

- **서비스를 사용으로 설정한 경우 새로 고칩니다.**

  ```
  # svcadm refresh ikev2
  ```

- **서비스를 사용할 수 없는 경우 사용으로 설정합니다.**

  ```
  # svcadm enable ikev2
  ```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

IKEv2에 대한 공개 키 인증서를 저장하도록 키 저장소 초기화

IKEv2에서 공개 인증서를 사용하려면 PKCS #11 키 저장소를 만들어야 합니다. 가장 흔히 사용되는 키 저장소는 Oracle Solaris의 암호화 프레임워크 기능에서 제공하는 `pkcs11_softtoken`입니다.

IKEv2에 대한 `pkcs11_softtoken` 키 저장소는 특수 사용자 `ikeuser`가 소유하는 디렉토리에 있습니다. 기본 디렉토리는 `/var/user/ikeuser`입니다. 사용자 ID `ikeuser`는 시스템과 함께 제공되지만, 키 저장소는 직접 만들어야 합니다. 키 저장소를 만들 때 키 저장소의 PIN을 만듭니다. IKEv2 서비스를 이용하려면 이 PIN을 사용하여 키 저장소에 로그인해야 합니다.

`pkcs11_softtoken` 키 저장소에는 IKEv2에서 사용하는 개인 키, 공개 키 및 공개 인증서가 유지됩니다. 이러한 키 및 인증서는 `pktool` 명령의 래퍼인 `ikev2cert` 명령을 사용하여 관리합니다. 이 래퍼를 사용하면 `ikeuser`가 소유하는 `pkcs11_softtoken` 키 저장소에 키 및 인증서 작업이 모두 적용됩니다.

PIN을 `ikev2` 서비스의 등록 정보 값으로 추가하지 않은 경우 `/var/log/ikev2/in.ikev2d.log` 파일에 다음과 같은 메시지가 표시됩니다.

```
date: (n)  No PKCS#11 token "pin" property defined
for the smf(5) service: ike:ikev2
```

공개 키 인증서를 사용하지 않는 경우에는 이 메시지를 무시해도 됩니다.

▼ IKEv2 공개 키 인증서에 대한 키 저장소를 만들고 사용하는 방법

IKEv2와 함께 공개 인증서를 사용하려는 경우 키 저장소를 만들어야 합니다. 키 저장소를 사용하려면 키 저장소에 로그인해야 합니다. `in.ikev2d` 데몬이 시작되면 데몬에 PIN을 사용자

가 제공하거나 자동 프로세스에서 제공합니다. 사이트 보안에서 자동 로그인을 허용하는 경우 자동 로그인을 구성해야 합니다. 기본값은 키 저장소를 사용하는 대화식 로그인입니다.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **IKEv2 키 저장소에 대한 PIN을 설정합니다.**

   `ikev2cert setpin` 명령을 사용하여 IKEv2 키 저장소를 만듭니다. 이 명령은 PKCS #11 키 저장소의 소유자를 `ikeuser`로 설정합니다.

   PIN에 공백을 사용하지 마십시오. 예를 들어, 값 `WhatShouldIWrite`는 유효하지만 값 `"What Should"`는 유효하지 않습니다.

```
%  pfbash
# /usr/sbin/ikev2cert setpin
Enter token passphrase: changeme
Create new passphrase:      Type strong passphrase
Re-enter new passphrase: xxxxxxxx
Passphrase changed.
```

주의 - 이 문장암호를 안전한 위치에 저장하십시오. 키 저장소를 사용하려면 필요합니다.

2. **키 저장소에 자동으로 또는 대화식으로 로그인합니다.**

   자동 로그인을 사용하는 것이 좋습니다. 사이트 보안 정책에서 자동 로그인을 허용하지 않는 경우 `in.ikev2d` 데몬이 다시 시작될 때 키 저장소에 대화식으로 로그인해야 합니다.

   ■ **키 저장소를 구성하여 자동 로그인을 사용으로 설정합니다.**

   a. **PIN을 `pkcs11_softtoken/pin` 서비스 등록 정보의 값으로 추가합니다.**

```
# svccfg -s ike:ikev2 editprop
```

   임시 편집 창이 열립니다.

   b. **`setprop pkcs11_token/pin =` 행의 주석 처리를 해제합니다.**

```
# setprop pkcs11_token/pin = astring: ()      Original entry
setprop pkcs11_token/pin = astring: ()  Uncommented entry
```

   c. **괄호를 1단계의 PIN으로 대체합니다.**

```
setprop pkcs11_token/pin = astring: PIN-from-Step-1
```

   콜론과 PIN 사이에 공백을 그대로 둡니다.

   d. **파일 맨 아래의 `refresh` 행을 주석 해제한 다음 변경 사항을 저장합니다.**

```
# refresh
refresh
```

e. **(옵션) pkcs11_token/pin 등록 정보의 값을 확인합니다.**

pkcs11_token/pin 등록 정보에는 ikeuser가 소유하는 키 저장소에 액세스할 때 확인되는 값이 포함됩니다.

```
# svccfg -s ike:ikev2 listprop pkcs11_token/pin
pkcs11_token/pin     astring      PIN
```

■ **자동 키 저장소 로그인을 구성하지 않은 경우 키 저장소에 수동으로 로그인합니다.**

in.ikev2d 데몬이 시작될 때마다 이 명령을 실행합니다.

```
# pfbash
# ikeadm -v2 token login "Sun Metaslot"
Enter PIN for PKCS#11 token 'Sun Metaslot':      Type the PIN from Step 1
ikeadm: PKCS#11 operation successful
```

3. **(옵션) 키 저장소에서 PIN이 설정되었는지 확인합니다.**

```
# ikev2cert tokens
Flags: L=Login required  I=Initialized  X=User PIN expired  S=SO PIN expired
Slot ID     Slot Name                   Token Name                      Flags
-------     ---------                   ----------                      -----
1           Sun Crypto Softtoken        Sun Software PKCS#11 softtoken    LI
```

Flags 열에 LI가 있으면 PIN이 설정된 것입니다.

4. **pkcs11_softtoken에서 수동으로 로그아웃하려면 ikeadm 명령을 사용합니다.**

```
# ikeadm -v2 token logout "Sun Metaslot"
ikeadm: PKCS#11 operation successful
```

로그아웃하여 두 사이트 간의 통신을 유한 기간으로 제한할 수 있습니다. 로그아웃하면 개인 키를 사용할 수 없게 되므로 새 IKEv2 세션을 시작할 수 없습니다. 기존 IKEv2 세션은 ikeadm delete ikesa 명령으로 세션 키를 삭제하지 않는 한 계속됩니다. 미리 공유한 키 규칙은 계속 작동합니다. ikeadm(1M) 매뉴얼 페이지를 참조하십시오.

공개 키 인증서로 IKEv2 구성

대규모 배포에서는 공개 인증서가 좋은 선택일 수 있습니다. 자세한 내용은 “IKE와 공개 키 인증서” [125]를 참조하십시오.

공개 키 인증서는 암호화 프레임워크에서 softtoken 키 저장소에 저장합니다. 연결된 하드웨어가 있는 시스템에서는 하드웨어에서도 인증서가 생성되어 저장될 수 있습니다. 절차는 하

드웨어에서 IKEv2에 대한 공개 키 인증서를 생성하고 저장하는 방법 [154]을 참조하십시오.

배경 정보는 "IKE 작동 방식" [124]을 참조하십시오.

다음 작업 맵에는 IKEv2에 대한 공개 키 인증서를 만드는 절차가 나옵니다. 이 절차에는 시스템에 Sun Crypto Accelerator 6000 보드가 연결된 경우 하드웨어 저장소에 인증서를 저장하는 방법이 포함됩니다.

표 9-1 공개 키 인증서로 IKEv2 구성 작업 맵

작업	설명	지침
인증서에 대한 키 저장소를 만듭니다.	IKEv2에 대한 인증서가 저장된 PKCS #11 키 저장소를 초기화합니다.	"IKEv2에 대한 공개 키 인증서를 저장하도록 키 저장소 초기화" [139]
자체 서명된 공개 키 인증서로 IKEv2를 구성합니다.	사용자가 서명한 공개 키 인증서를 만듭니다. 인증서를 피어로 내보내고 피어의 인증서를 가져옵니다.	자체 서명된 공개 키 인증서로 IKEv2를 구성하는 방법 [142]
CA에서 발행한 인증서로 IKEv2를 구성합니다.	CSR을 만든 다음 반환된 인증서를 모두 키 저장소로 가져와야 합니다. 그런 다음 IKE 피어의 인증서를 확인하고 가져옵니다.	CA가 서명한 인증서로 IKEv2를 구성하는 방법 [148]
해지된 인증서를 처리하는 방법을 구성합니다.	네트워크 지연을 처리하는 방법을 비롯해 CRL이 사용되는지 여부와 OCSP 서버가 폴링되는지 여부를 결정합니다.	IKEv2에서 인증서 검증 정책을 설정하는 방법 [150]
연결된 하드웨어의 키 저장소에서 인증서 저장을 구성합니다.	Sun Crypto Accelerator 6000 보드를 찾고 IKEv2에서 이 보드를 사용하도록 구성합니다.	하드웨어에서 IKEv2에 대한 공개 키 인증서를 생성하고 저장하는 방법 [154]

▼ 자체 서명된 공개 키 인증서로 IKEv2를 구성하는 방법

이 절차에서는 공개 키 인증서를 만들고 이 인증서에 서명합니다. 개인 키와 인증서는 IKEv2의 PKCS #11 softtoken 키 저장소에 저장됩니다. 공개 키 인증서를 IKE 피어로 보내면 IKE 피어에서 자체의 공개 인증서를 보내옵니다.

자체 서명된 인증서를 사용하는 IKE 시스템 모두에서 이 절차를 수행합니다.

시작하기 전에 인증서를 사용하려면 IKEv2 공개 키 인증서에 대한 키 저장소를 만들고 사용하는 방법 [139]을 완료해야 합니다.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸을 사용하는 중이어야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **키 저장소에서 자체 서명된 인증서를 만듭니다.**

`ikev2cert gencert` 명령의 인수에 대한 설명은 `pktool(1)` 매뉴얼 페이지에서 `pktool gencert keystore=pkcs11` 하위 명령을 검토하십시오.

`subject` 인수의 형식은 "IKE에서 공개 키 인증서 사용" [126]을 참조하십시오.

참고 - 인증서에 레이블을 지정합니다. 레이블은 로컬 키 저장소에서 인증서와 해당 키를 식별합니다.

---

a. **예를 들어, `partym` 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# pfbash
 # ikev2cert gencert \
 label="ITpartym" \
 subject="O=exampleco, OU=IT, C=US, CN=partym" \
 serial=0x87654321
    keytype=rsa
    keylen=2048
 Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
```

다음 오류 메시지는 PIN을 잘못 입력하거나 키 저장소가 초기화되지 않았음을 나타냅니다.

```
Error creating certificate and keypair:
 keystore error: CKR_PIN_INCORRECT
 libkmf error: KMF_ERR_AUTH_FAILED

 Error creating certificate and keypair:
 keystore error: CKR_PIN_EXPIRED: PIN expired and must be changed
 libkmf error: KMF_ERR_BAD_PARAMETER: invalid parameter
```

작은 정보 - `pktool` 명령 구문이 표시되면 인증서 항목 일부를 잘못 입력했음을 나타냅니다. 명령을 검토하여 허용되지 않는 알고리즘을 사용하거나, 큰따옴표와 등호가 누락되거나, 기타 오타가 있는지 확인하십시오. 잘못된 인수를 찾는 전략 중 하나로 명령을 검색한 다음 인수를 한 번에 하나씩 제거해 보십시오.

---

b. **`enigma` 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# ikev2cert gencert \
 label=ITenigma \
 subject="O=exampleco, OU=IT, C=US, CN=enigma" \
 serial=0x86428642
    keytype=rsa
    keylen=2048
 Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
```

2. **(옵션) 키 및 인증서를 나열합니다.**

```
enigma # /usr/sbin/ikev2cert list objtype=both
 Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
```

```
No.     Key Type        Key Len.        Key Label
-------------------------------------------------
Asymmetric private keys:
1)      RSA                             ITenigma
Asymmetric public keys:
1)      RSA                             ITenigma
Certificates:
1) X.509 certificate
 Label: ITenigma
 Subject: C=US, O=exampleco, OU=IT, CN=enigma
 Issuer: C=US, O=exampleco, OU=IT, CN=enigma
 Not Before: April 10 21:49:00 2014 GMT
 Not After: April 10 21:49:00 2015 GMT
 Serial: 0x86426420
 Signature Algorithm: sha1WithRSAEncryption
 X509v3 Subject Key Identifier:
  34:7a:3b:36:c7:7d:4f:60:ed:ec:4a:96:33:67:f2:ac:87:ce:35:cc
 SHA1 Certificate Fingerprint:
  68:07:48:65:a2:4a:bf:18:f5:5b:95:a5:01:42:c0:26:e3:3b:a5:30
```

작은 정보 - 기본 해싱 알고리즘은 SHA1입니다. 강력한 서명 알고리즘으로 인증서를 만들려면 keytype 옵션과 다른 해시 알고리즘(예: keytype=rsa hash=sha384)을 사용하십시오. 옵션은 pktool(1) 매뉴얼 페이지를 참조하십시오.

3. **인증서를 다른 시스템으로 배달합니다.**

   a. **각 시스템에서 인증서만 파일로 내보냅니다.**

   outformat=pem 옵션을 지정하면 공개 인증서가 직접 가져오기 적합한 형식의 파일에 저장됩니다. 레이블은 키 저장소에서 인증서를 식별합니다.

   ```
   # cd /tmp
    # ikev2cert export objtype=cert outformat=pem outfile=filename label=label
    Enter PIN for Sun Software PKCS#11 softtoken:xxxxxxxx
   ```

   b. **인증서를 전자 메일, sftp 또는 ssh로 다른 시스템에 보냅니다.**

   예를 들어, 두 시스템을 모두 관리하는 경우 sftp 명령을 사용하여 다른 시스템에서 인증서를 가져옵니다.

   ```
   enigma # sftp jdoe@partym:/tmp/ITpartym.pem /tmp/ITpartym.pem.cert
    partym # sftp jdoe@enigma:/tmp/ITenigma.pem /tmp/ITenigma.pem.cert
   ```

   암호를 입력하라는 메시지가 표시됩니다. 이 예에서 jdoe가 암호를 제공해야 합니다.

4. **인증서가 동일한지 확인합니다.**

   인증서를 키 저장소로 로드하기 전에 적절한 인증서를 받았는지 확인하려고 합니다.

   a. **각 시스템에서 내보낸 파일의 다이제스트를 만듭니다.**

예를 들어, partym 관리자는 partym의 인증서가 포함된 파일의 다이제스트를 다른 관리자에게 전자 메일로 보냅니다. enigma 관리자는 enigma 인증서 파일의 다이제스트를 전자 메일로 보냅니다.

```
partym # digest -a sha1 /tmp/ITpartym.pem
 c6dbef4136c0141ae62110246f288e5546a59d86

 enigma # digest -a sha1 ITenigma.pem
 6b288a6a6129d53a45057065bd02b35d7d299b3a
```

b. **다른 시스템에서 첫번째 시스템의 인증서가 포함된 파일에 대해 digest 명령을 실행합니다.**

```
enigma # digest -a sha1 /tmp/ITpartym.pem.cert
 c6dbef4136c0141ae62110246f288e5546a59d86

 partym # digest -a sha1 /tmp/ITenigma.pem.cert
 6b288a6a6129d53a45057065bd02b35d7d299b3a
```

다이제스트가 일치해야 합니다. 일치하지 않는 경우 파일을 키 저장소로 가져오지 마십시오. 인증서 유효성을 확인하는 다른 방법은 예 9-3. “지문으로 공개 키 인증서 확인”을 참조하십시오.

5. **확인 후 다른 시스템의 인증서를 키 저장소로 가져옵니다.**

인증서를 키 저장소로 가져올 때 시스템에서 인증서를 고유하게 식별하는 레이블을 지정해야 합니다. 레이블은 공개 키를 공개 키 인증서와 연결합니다.

```
enigma# ikev2cert import label=ITpartym1 infile=/tmp/ITpartym.pem.cert

partym# ikev2cert import label=ITenigma1 infile=/tmp/ITenigma.pem.cert
```

6. **(옵션) 키 저장소의 객체를 나열합니다.**

목록을 2단계의 목록과 비교합니다. 예를 들어, enigma 키 저장소에서는 partym 인증서가 추가되었습니다.

```
enigma # /usr/sbin/ikev2cert list objtype=both
 Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
 No.     Key Type        Key Len.        Key Label
 -------------------------------------------------
 Asymmetric private keys:
 1)      RSA                             ITenigma
 Asymmetric public keys:
 1)      RSA                             ITenigma
 Certificates:
 1) X.509 certificate
  Label: ITenigma
  Subject: C=US, O=exampleco, OU=IT, CN=enigma
  Issuer: C=US, O=exampleco, OU=IT, CN=enigma
  Not Before: April 10 21:49:00 2014 GMT
  Not After: April 10 21:49:00 2015 GMT
```

```
 Serial: 0x86426420
 Signature Algorithm: sha1WithRSAEncryption
 X509v3 Subject Key Identifier:
  34:7a:3b:36:c7:7d:4f:60:ed:ec:4a:96:33:67:f2:ac:87:ce:35:cc
 SHA1 Certificate Fingerprint:
  68:07:48:65:a2:4a:bf:18:f5:5b:95:a5:01:42:c0:26:e3:3b:a5:30

2) X.509 certificate
 Label: ITpartym1
 Subject: C=US, O=exampleco, OU=IT, CN=partym
 Issuer: C=US, O=exampleco, OU=IT, CN=partym
 Not Before: April 10 21:40:00 2014 GMT
 Not After: April 10 21:40:00 2015 GMT
 Serial: 0x87654321
 Signature Algorithm: sha1WithRSAEncryption
 X509v3 Subject Key Identifier:
  ae:d9:c8:a4:19:68:fe:2d:6c:c2:9a:b6:06:55:b5:b5:d9:d9:45:c6
 SHA1 Certificate Fingerprint:
  83:26:44:29:b4:1f:af:4a:69:0d:87:c2:dc:f4:a5:1b:4f:0d:36:3b
```

7. **각 시스템에서 인증서를 IKEv2 규칙에 사용합니다.**

   `pfedit` 명령을 사용하여 `/etc/inet/ike/ikev2.config` 파일을 편집합니다.

   a. **예를 들어, `partym` 시스템에서 `ikev2.config` 파일의 규칙은 다음과 비슷하게 표시됩니다.**

   ```
   ##  ... Global transform that applies to any rule without a declared transform
    ikesa_xform { dh_group 21 auth_alg sha512 encr_alg aes }
    ##  ... Any self-signed
    ## end-entity certificates must be present in the keystore or
    ## they will not be trusted.
    {
     label "partym-enigma"
     auth_method cert
     local_id DN = "O=exampleco, OU=IT, C=US, CN=partym"
     remote_id DN = "O=exampleco, OU=IT, C=US, CN=enigma"
    }
    ...
   ```

   b. **`enigma` 시스템에서 `ikev2.config` 파일에서 `local_id` 값에 `enigma` 인증서의 DN을 사용합니다.**

      원격 매개변수에는 `partym` 인증서의 DN을 값으로 사용합니다. `label` 키워드가 로컬 시스템에서 고유한지 확인합니다.

   ```
   …
    ikesa_xform { dh_group 21 auth_alg sha512 encr_alg aes }
    …
    {
     label "enigma-partym"
     auth_method cert
     local_id DN = "O=exampleco, OU=IT, C=US, CN=enigma"
   ```

```
 remote_id DN = "O=exampleco, OU=IT, C=US, CN=partym"
}
...
```

8. **(옵션) 각 시스템에서 `ikev2.config` 파일의 유효성을 확인합니다.**

```
# /usr/lib/inet/inikev2.d -c
```

계속하기 전에 오타나 오류를 수정합니다.

9. **각 시스템에서 IKEv2 서비스 인스턴스의 상태를 확인합니다.**

```
# svcs ikev2
 STATE          STIME    FMRI
 disabled       Sep_07   svc:/network/ipsec/ike:ikev2
```

10. **각 시스템에서 IKEv2 서비스 인스턴스를 사용으로 설정합니다.**

```
partym # svcadm enable ipsec/ike:ikev2

 enigma # svcadm enable ipsec/ike:ikev2
```

예 9-2 수명이 제한된 자체 서명된 인증서 만들기

이 예에서 관리자는 인증서가 2년간 유효하도록 지정합니다.

```
# ikev2cert gencert \
  > label=DBAuditV \
  > serial=0x12893467235412 \
  > subject="O=exampleco, OU=DB, C=US, CN=AuditVault" \
  > altname=EMAIL=auditV@example.com \
  > keytype=ec curve=secp521r1 hash=sha512 \
     > lifetime=2-year
```

예 9-3 지문으로 공개 키 인증서 확인

이 예에서 관리자는 인증서 지문을 사용하여 인증서를 확인합니다. 이 방법의 단점은 관리자가 피어의 인증서를 키 저장소로 가져와야만 지문을 볼 수 있다는 점입니다.

관리자는 인증서를 가져오고, `ikev2cert list objtype=cert` 명령으로 인증서를 나열한 다음 인증서 지문을 출력에서 복사하여 다른 시스템 관리자에게 보냅니다.

```
SHA1 Certificate Fingerprint:
            83:26:44:29:b4:1f:af:4a:69:0d:87:c2:dc:f4:a5:1b:4f:0d:36:3b
```

확인에 실패하는 경우 인증서를 가져온 관리자가 인증서와 인증서의 공개 키를 키 저장소에서 제거해야 합니다.

```
# ikev2cert delete label=label-name
```

```
Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
1 public key(s) found, do you want to delete them (y/N) ? y
1 certificate(s) found, do you want to delete them (y/N) ? y
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ CA가 서명한 인증서로 IKEv2를 구성하는 방법

많은 수의 통신 시스템을 보호하는 조직은 일반적으로 CA(인증 기관)에서 발행한 공개 인증서를 사용합니다. 배경 정보는 "IKE와 공개 키 인증서" [125]를 참조하십시오.

이 절차는 CA의 인증서를 사용하는 모든 IKE 시스템에서 수행합니다.

시작하기 전에 인증서를 사용하려면 IKEv2 공개 키 인증서에 대한 키 저장소를 만들고 사용하는 방법 [139]을 완료해야 합니다.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **쓰기 가능한 디렉토리로 변경합니다.**

   다음 오류 메시지는 CSR 파일을 디스크에 쓸 수 없음을 나타낼 수 있습니다.

   ```
   Warning: error accessing "CSR-file"
   ```

   예를 들어, `/tmp` 디렉토리를 사용합니다.

   ```
   # cd /tmp
   ```

2. **인증서 서명 요청을 만듭니다.**

   `ikev2cert gencsr` 명령을 사용하여 CSR(인증서 서명 요청)을 만듭니다. 명령의 인수에 대한 설명은 `pktool`(1) 매뉴얼 페이지에서 `pktool gencsr keystore=pkcs11` 하위 명령을 검토하십시오.

   예를 들어, 다음 명령은 `partym` 시스템에 대한 CSR을 포함하는 파일을 만듭니다.

   ```
   # pfbash
   # /usr/sbin/ikev2cert gencsr \
   keytype=rsa
   ```

```
keylen=2048
label=Partym1 \
outcsr=/tmp/Partymcsr1 \
subject="C=US, O=PartyCompany\, Inc., OU=US-Partym, CN=Partym"
Enter PIN for Sun Software PKCS#11 softtoken: xxxxxxxx
```

3. **(옵션) CA의 웹 양식에 붙여 넣기 위해 CSR의 내용을 복사합니다.**

```
# cat /tmp/Partymcsr1
-----BEGIN CERTIFICATE REQUEST-----
MIICkDCCAXoCAQAwTzELMAkGA1UEBhMCVVMxGzAZBgNVBAoTElBhcnR5Q29tcGFu
eSwgSW5jLjESMBAGA1UECxMJVVMtUGFydHltMQ8wDQYDVQQDEwZQYXJ0eW0wggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCMbINmgZ4XWUv2q1fshZUN/SLb
WNLXZxdKwt5e71o0owjyby69eL7HE0QBUij73nTkXE3n4gxojBZE+hvJ6GOCbREA
jgSquP2US7Bn9XEcXRrsOc7MCFPrsA+hVIcNHpKNseUOU/rg+wzoo5hA1ixtWuXH
bYDeEWQi5tlZgDZoCWGrdHEjwVyHfvz+a0WBjyZBYOueBhXaa68QqSOSnRVDX56Q
3p4H/AR4h0dcSja72XmMKPU5p3RVb8n/hrfKjiDjiGYXD4D+WZxQ65xxCcnALvVH
nZHUlAtP7QHX4RXlQVNNwEsY6C95RX9297rNWLsYvp/86xWrQkTlNqVAeUKhAgMB
AAEwCwYJKoZIhvcNAQEFA4IBAQB3R6rmZdqcgN8Tomyjp2CFTdyAWixkIATXpLM1
GL5ghrnDvadD61M+vS1yhFlIcSNM8fLRrCHIKtAmB8ITnggJ//rzbHq3jdla/iQt
kgGoTXfz8j6B57Ud6l+MBLiBSBy0QK4GIg8Ojlb9Kk5HsZ48mIoI/Qb7FFW4p9dB
JEUn0eYhkaGtwJ21YNNvKgOeOcnSZy+xP9Wa9WpfdsBO4TicLDw0Yq7koNnfL0IB
Fj2bt/wI7iZ1Dcpwphsiwn W9K9YynAJZzHd1ULVpn5Kd7vSRz9youLLzSb+9ilgO
E43DW0hRk6P/Uq0N4e1Zca4otezNxyEqlPZI7pJ5uOo0sbiw
-----END CERTIFICATE REQUEST-----
```

4. **CSR을 certificate authority(CA, 인증 기관)에 제출합니다.**

   CA에서 CSR을 제출하는 방법을 알려 줄 수 있습니다. 대부분 조직에는 제출 양식을 제공하는 웹 사이트가 있습니다. 양식을 사용하려면 제출이 적합한지 증명해야 합니다. 일반적으로 양식에 CSR을 붙여 넣습니다.

작은 정보 - 일부 웹 양식에는 인증서를 붙여 넣을 수 있는 고급 버튼이 있습니다. CSR은 PKCS#10 형식으로 생성됩니다. 따라서 웹 양식에서 PKCS#10을 언급하는 부분을 찾으십시오.

---

5. **CA에서 받는 각 인증서를 키 저장소로 가져옵니다.**

   `ikev2cert import`는 인증서를 키 저장소로 가져옵니다.

   a. **CA에서 받은 인증서와 공개 키를 가져옵니다.**

```
# ikev2cert import objtype=cert label=Partym1 infile=/tmp/Partym1Cert
```

작은 정보 - 관리하기 편하도록 원래 CSR의 레이블과 동일한 레이블을 가져온 인증서에 지정하십시오.

---

   b. **CA에서 루트 인증서를 가져옵니다.**

```
# ikev2cert import objtype=cert infile=/tmp/Partym1CAcert
```

c. **중간 CA 인증서를 키 저장소로 가져옵니다.**

작은 정보 - 관리하기 편하도록 원래 CSR의 레이블과 동일한 레이블을 가져온 중간 인증서에 지정하십시오.

CA에서 각 중간 인증서의 파일을 별도로 보낸 경우 앞의 인증서를 가져올 때처럼 파일을 가져옵니다. 그러나 CA에서 인증서 체인을 PKCS#7 파일로 제공하는 경우에는 파일에서 개별 인증서를 추출한 다음 앞의 인증서를 가져올 때처럼 각 인증서를 가져와야 합니다.

참고 - `openssl` 명령을 실행하려면 `root` 역할이 있어야 합니다. `openssl`(5) 매뉴얼 페이지를 참조하십시오.

```
# openssl pkcs7 -in pkcs7-file -print_certs
# ikev2cert import objtype=cert label=Partym1 infile=individual-cert
```

6. **인증서 검증 정책을 설정합니다.**
인증서에 CRL 또는 OCSP에 대한 섹션이 포함된 경우 사이트 요구 사항에 따라 인증서 검증 정책을 구성해야 합니다. 지침은 IKEv2에서 인증서 검증 정책을 설정하는 방법 [150]을 참조하십시오.

7. **인증서를 사용하는 모든 IKE 시스템에서 절차를 완료한 후 모든 시스템에서 `ikev2` 서비스를 사용으로 설정합니다.**
피어 시스템에는 trust anchor(트러스트 앵커) 인증서 및 구성된 `ikev2.config` 파일이 필요합니다.

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ IKEv2에서 인증서 검증 정책을 설정하는 방법

IKEv2 시스템에 대해 인증서가 처리되는 방법을 여러 측면에서 구성할 수 있습니다.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **기본 인증서 검증 정책을 검토합니다.**
   인증서 정책은 설치 시 `/etc/inet/ike/kmf-policy.xml` 파일에서 설정됩니다. 이 파일은 `ikeuser`가 소유하며 `kmfcfg` 명령을 사용하여 수정합니다. 기본 인증서 검증 정책을 적용하면 CRL이 `/var/user/ikeuser/crls` 디렉토리로 다운로드됩니다. OCSP 사용도 기본적으로 사용으로 설정됩니다. 사이트에서 인터넷에 연결하는 데 프록시가 필요한 경우 프록시를 구성해야 합니다. IKEv2에서 해지된 인증서를 처리하는 방법 [152]을 참조하십시오.

```
# pfbash
# kmfcfg list dbfile=/etc/inet/ike/kmf-policy.xml policy=default
Policy Name: default
Ignore Certificate Validity Dates: false      Unknown purposes or applications for the certificate
Ignore Unknown EKUs: false
Ignore Trust Anchor in Certificate Validation: false
Trust Intermediate CAs as trust anchors: false
Maximum Certificate Path Length: 32
Certificate Validity Period Adjusted Time leeway: [not set]
Trust Anchor Certificate: Search by Issuer
Key Usage Bits: 0      Identifies critical parts of certificate
Extended Key Usage Values: [not set]      Purposes or applications for the certificate
HTTP Proxy (Global Scope): [not set]
Validation Policy Information:
    Maximum Certificate Revocation Responder Timeout: 10
    Ignore Certificate Revocation Responder Timeout: true
    OCSP:
        Responder URI: [not set]
        OCSP specific proxy override: [not set]
        Use ResponderURI from Certificate: true
        Response lifetime: [not set]
        Ignore Response signature: false
        Responder Certificate: [not set]
    CRL:
        Base filename: [not set]
        Directory: /var/user/ikeuser/crls
        Download and cache CRL: true
        CRL specific proxy override: [not set]
        Ignore CRL signature: false
        Ignore CRL validity date: false
IPsec policy bypass on outgoing connections: true
Certificate to name mapper name: [not set]
Certificate to name mapper pathname: [not set]
Certificate to name mapper directory: [not set]
Certificate to name mapper options: [not set]
```

2. **인증서를 검토하여 수정할 검증 옵션을 나타내는 기능이 있는지 확인합니다.**
   예를 들어, CRL 또는 OCSP URI를 포함하는 인증서는 인증서 해지 상태를 확인하는 데 사용할 URI를 지정하는 검증 정책을 사용할 수 있습니다. 시간 초과를 구성할 수도 있습니다.

3. **`kmfcfg(1)` 매뉴얼 페이지에서 구성 가능한 옵션을 검토하십시오.**

4. **인증서 검증 정책을 구성합니다.**
   샘플 정책은 IKEv2에서 해지된 인증서를 처리하는 방법 [152]을 참조하십시오.

▼ IKEv2에서 해지된 인증서를 처리하는 방법

해지된 인증서는 일정한 이유로 손상된 인증서입니다. 해지된 인증서를 사용하면 보안상 위험합니다. 인증서 해지 확인 시 옵션을 선택할 수 있습니다. 정적 목록을 사용할 수도 있고 HTTP 프로토콜을 통해 해지를 동적으로 확인할 수도 있습니다.

시작하기 전에 CA에서 인증서를 받아 설치한 상태여야 합니다.

해지된 인증서를 확인하는 CRL 및 OSCP 방법에 대해 잘 알아야 합니다. 정보 및 포인터는 "IKE와 공개 키 인증서" [125]를 참조하십시오.

Network IPsec Management 권한 프로파일이 지정된 관리자여야 하며 프로파일 셸을 사용해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **CA에서 받은 인증서에서 CRL 및 OCSP 섹션을 찾습니다.**
   CSR의 레이블을 통해 인증서를 식별할 수 있습니다.

```
# pfbash
# ikev2cert list objtype=cert | grep Label:
Enter PIN for Sun Software PKCS#11 softtoken:
        Label: Partym1
```

   예를 들어, 잘린 다음 출력에서는 인증서의 CRL 및 OCSP URI이 강조 표시됩니다.

```
# ikev2cert list objtype=cert label=Partym1
X509v3 extensions:
    ...
    X509v3 CRL Distribution Points:
         Full Name:
       URI:http://onsitecrl.PKI.example.com/OCCIPsec/LatestCRL.crl
    X509v3 Authority Key Identifier:
         ...
    Authority Information Access:
        OCSP - URI:http://ocsp.PKI.example.com/revokes/
    X509v3 Certificate Policies:
         Policy: 2.16.840.1.113733.1.7.23.2
```

   `CRL Distribution Points` 항목 아래에 URI 값이 있으면 이 조직의 CRL이 웹의 파일로 제공됨을 나타냅니다. OCSP 항목은 개별 인증서의 상태를 서버에서 동적으로 확인할 수 있음을 나타냅니다.

2. **프록시를 지정하여 CRL 또는 OCSP 서버를 사용할 수 있도록 설정합니다.**

```
# kmfcfg modify \
dbfile=/etc/inet/ike/kmf-policy.xml \
policy=default \
http-proxy=www-proxy.ja.example.com:80
```

   프록시가 선택 사항인 사이트에서는 프록시를 지정하지 않아도 됩니다.

3. **인증서 검증 정책이 업데이트되었는지 확인합니다.**
   예를 들어, OCSP가 업데이트되었는지 확인합니다.

```
# kmfcfg list \
dbfile=/etc/inet/ike/kmf-policy.xml \
policy=default
...
    OCSP:
        Responder URI: [not set]
        Proxy: www-proxy.ja.example.com:80
        Use ResponderURI from Certificate: true
        Response lifetime: [not set]
        Ignore Response signature: false
        Responder Certificate: [not set]
```

4. **IKEv2 서비스를 다시 시작합니다.**

```
# svcadm restart ikev2
```

5. **(옵션) CRL 또는 OCSP 사용을 중지합니다.**

   - **CRL 사용을 중지하려면 다음을 입력합니다.**

```
# pfexec kmfcfg modify \
dbfile=/etc/inet/ike/kmf-policy.xml policy=default \
crl-none=true
```

     `crl-none=true` 인수를 지정하면 시스템이 로컬 캐시에서 다운로드한 CRL을 사용합니다.

   - **OCSP 사용을 중지하려면 다음을 입력합니다.**

```
# pfexec kmfcfg modify \
dbfile=/etc/inet/ike/kmf-policy.xml policy=default \
ocsp-none=true
```

예 9-4 시스템이 IKEv2 인증서 검증을 대기하는 시간 변경

이 예에서는 관리자가 인증서 검증 대기 시간을 20초로 제한합니다.

```
# kmfcfg modify dbfile=/etc/inet/ike/kmf-policy.xml policy=default \
    cert-revoke-responder-timeout=20
```

기본적으로 응답 시간이 초과되면 피어 인증이 성공합니다. 여기서, 관리자는 인증이 실패하면 연결이 거부되는 정책을 구성합니다. 이 구성에서는 OCSP 또는 CRL 서버가 응답하지 않게 되는 경우 인증서 검증이 실패합니다.

```
# kmfcfg modify dbfile=/etc/inet/ike/kmf-policy.xml policy=default \
    ignore-cert-revoke-responder-timeout=false
```

정책을 활성화하려면 관리자가 IKEv2 서비스를 다시 시작합니다.

```
# svcadm restart ikev2
```

▼ 하드웨어에서 IKEv2에 대한 공개 키 인증서를 생성하고 저장하는 방법

공개 키 인증서를 연결된 하드웨어에 저장할 수도 있습니다. Sun Crypto Accelerator 6000 보드는 저장소를 제공하고 공개 키 작업이 시스템에서 보드로 오프로드될 수 있도록 합니다.

공개 키 인증서를 생성하여 하드웨어에 저장하는 작업은 시스템에서 공개 키 인증서를 생성하여 저장하는 작업과 유사합니다. 하드웨어에서 `ikev2cert gencert token=`*hw-keystore* 명령을 사용하여 하드웨어 키 저장소를 식별할 수 있습니다.

시작하기 전에 이 절차에서는 Sun Crypto Accelerator 6000 보드가 시스템에 연결되었다고 간주합니다. 또한 보드용 소프트웨어가 설치되었으며 하드웨어 키 저장소가 구성되었다고 간주합니다. 지침은 Sun Crypto Accelerator 6000 Board Product Library Documentation (http://docs.oracle.com/cd/E19321-01/index.html)을 참조하십시오. 이러한 지침에는 키 저장소를 설정하는 작업도 포함됩니다.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **연결된 Sun Crypto Accelerator 6000 보드에 대한 토큰 ID가 있는지 확인합니다.**

```
# pfbash
# ikev2cert tokens

Flags: L=Login required I=Initialized X=User PIN expired S=SO PIN expired
Slot ID  Slot Name                        Token Name                       Flags
-------  ---------                        ----------                       -----
1        sca6000                          sca6000                          LI
2        n2cp/0 Crypto Accel Bulk 1.0     n2cp/0 Crypto Accel Bulk 1.0
3        ncp/0 Crypto Accel Asym 1.0      ncp/0 Crypto Accel Asym 1.0
4        n2rng/0 SUNW_N2_Random_Number_Ge n2rng/0 SUNW_N2_RNG
5        Sun Crypto Softtoken             Sun Software PKCS#11 softtoken   LI
```

2. **자체 서명된 인증서 또는 CSR을 생성하고 토큰 ID를 지정합니다.**

참고 - Sun Crypto Accelerator 6000 보드는 RSA에 대해 최대 2048비트의 키를 지원합니다. DSA의 경우 이 보드는 최대 1024비트의 키를 지원합니다.

다음 옵션 중 하나를 선택합니다.

- **자체 서명된 인증서의 경우 다음 구문을 사용합니다.**

```
# ikev2cert gencert token=sca6000 keytype=rsa \
hash=sha256 keylen=2048 \
subject="CN=FortKnox, C=US" serial=0x6278281232 label=goldrepo
Enter PIN for sca6000:      See Step 3
```

- **인증서 서명 요청의 경우 다음 구문을 사용합니다.**

```
# ikev2cert gencsr token=sca6000 -i
> keytype=
> hash=
> keylen=
> subject=
> serial=
> label=
> outcsr=
Enter PIN for sca6000 token:      See Step 3
```

`ikev2cert` 명령의 인수에 대한 설명은 `pktool(1)` 매뉴얼 페이지를 참조하십시오.

3. **PIN에 대한 프롬프트에서 Sun Crypto Accelerator 6000 사용자 이름, 콜론 및 사용자 암호를 입력합니다.**

참고 - 키 저장소의 사용자 이름과 암호를 알아야 합니다.

Sun Crypto Accelerator 6000 보드가 암호가 `inThe%4ov`인 `admin` 사용자로 구성된 경우 다음을 입력합니다.

```
Enter PIN for sca6000 token: admin:inThe%4ov
-----BEGIN X509 CERTIFICATE-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
…
oKUDBbZ9O/pLWYGr
-----END X509 CERTIFICATE-----
```

4. **상대방이 사용할 인증서를 보냅니다.**

   다음 옵션 중 하나를 선택합니다.

   - **원격 시스템에 자체 서명된 인증서를 보냅니다.**

     이 인증서를 전자 메일 메시지에 붙여 넣을 수 있습니다.

   - **인증서 서명 요청을 CA에 보냅니다.**

     CA의 지침에 따라 CSR을 제출합니다. 자세한 설명은 "IKE에서 공개 키 인증서 사용" [126]을 참조하십시오.

5. **인증서를 하드웨어 키 저장소로 가져옵니다.**

CA에서 받은 인증서를 가져오고 3단계에서 사용자와 PIN을 지정합니다.

```
# ikev2cert import token=sca6000 infile=/tmp/DCA.ACCEL.CERT1
Enter PIN for sca6000 token:      Type user:password
# ikev2cert import token=sca6000 infile=/tmp/DCA.ACCEL.CA.CERT
Enter PIN for sca6000 token:      Type user:password
```

6. **하드웨어 키 저장소가 자동으로 또는 대화식으로 사용되도록 설정합니다.**

   자동 로그인을 사용하는 것이 좋습니다. 사이트 보안 정책에서 자동 로그인을 허용하지 않는 경우 `in.ikev2d` 데몬이 다시 시작될 때 키 저장소에 대화식으로 로그인해야 합니다.

   ■ **키 저장소로의 자동 로그인을 구성합니다.**

   a. **PIN을 `pkcs11_token/uri` 서비스 등록 정보의 값으로 추가합니다.**

      이 등록 정보에 대한 설명은 "IKEv2 서비스" [213]를 참조하십시오.

      ```
      # svccfg -s ike:ikev2 editprop
      ```

      임시 편집 창이 열립니다.

   b. **`setprop pkcs11_token/uri =` 행을 주석 해제하고 괄호를 다음 형식의 토큰 이름으로 대체합니다.**

      ```
      # setprop pkcs11_token/uri = ()      Original entry
      setprop pkcs11_token/uri = pkcs11:token=sca6000
      ```

   c. **`setprop pkcs11_token/uri =` 행을 주석 해제하고 괄호를 3단계의 *username:PIN*으로 대체합니다.**

      ```
      # setprop pkcs11_token/uri = ()      Original entry
      setprop pkcs11_token/uri = admin:PIN-from-Step-3
      ```

   d. **파일 맨 아래에 있는 `refresh` 행을 주석 해제한 다음 변경 사항을 저장합니다.**

      ```
      # refresh
      refresh
      ```

   e. **(옵션) `pkcs11_token` 등록 정보 값을 확인합니다.**

      ```
      # svccfg -s ikev2 listprop pkcs11_token
      pkcs11_token/pin     astring      username:PIN
      pkcs11_token/uri     astring  pkcs11:token=sca6000
      ```

   ■ **자동 로그인을 구성하지 않은 경우 하드웨어 키 저장소에 수동으로 로그인합니다.**

   `in.ikev2d` 데몬이 시작될 때마다 이 명령을 실행합니다.

   ```
   # pfexec ikeadm -v2 token login sca6000
   Enter PIN for sca6000 token: admin:PIN-from-Step-3
   ```

```
ikeadm: sca6000 operation successful
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

♦ ♦ ♦ 10 장

10 IKEv1 구성

이 장에서는 시스템의 IKE(Internet Key Exchange) 버전 1(IKEv1)을 구성하는 방법에 대해 설명합니다. IKEv1이 구성되면 네트워크에서 IPsec에 대한 키 입력 도구가 자동으로 생성됩니다. 이 장은 다음 정보를 포함합니다.

참고 - IKEv2만 구현하려는 경우 9장. IKEv2 구성을 진행합니다.

IKE에 대한 개요 정보는 8장. IKE(Internet Key Exchange)를 참조하십시오. IKE에 대한 참조 정보는 12장. IPsec 및 키 관리 참조를 참조하십시오. 자세한 절차는 `ikeadm`(1M), `ikecert`(1M) 및 `ike.config`(4) 매뉴얼 페이지의 Examples 절을 참조하십시오.

참고 - 이러한 작업에서는 시스템에 정적 IP 주소가 지정되어 있고 네트워크 구성 프로파일 `DefaultFixed`를 실행 중이라고 가정합니다. `netadm list` 명령에서 `Automatic`을 반환하는 경우 자세한 내용은 `netcfg`(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 구성

미리 공유한 키, 자체 서명된 인증서 및 CA(인증 기관)의 인증서를 사용하여 IKE를 인증할 수 있습니다. `ike/config` 파일의 규칙으로 특정 IKEv1 인증 방법을 IKEv1 피어와 연결합니다. 따라서 시스템에서 IKE 인증 방법 중 하나 또는 전체를 사용할 수 있습니다. PKCS #11 라이브러리에 대한 포인터를 통해 IKEv1은 연결된 하드웨어 가속기를 사용할 수 있습니다.

IKEv1을 구성한 후 7장. IPsec 구성에서 IKEv1을 사용하는 IPsec 작업을 완료합니다.

미리 공유한 키로 IKEv1 구성

IKEv1을 사용하도록 피어 시스템 또는 서브넷을 구성 중이며, 해당 서브넷의 관리자라면 미리 공유한 키를 사용하는 것이 좋습니다. 테스트할 때에도 미리 공유한 키를 사용할 수 있습니다. 자세한 내용은 "IKE와 미리 공유한 키 인증" [125]을 참조하십시오.

▼ 미리 공유한 키로 IKEv1을 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.

참고 - 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 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "`ssh` 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 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 IKEv1 구성 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]의 `ipsecinit.conf` 예와 함께 사용합니다.

   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 미리 공유한 키를 /etc/inet/secret/ike.preshared 파일에 넣습니다.**

a. **예를 들어, enigma 시스템에서는 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 시스템에서는 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. **IKEv1 서비스를 사용으로 설정합니다.**

```
# svcadm enable ipsec/ike:default
```

예 10-1 IKEv1 미리 공유한 키 새로 고침

IKEv1 관리자가 미리 공유한 키를 새로 고치려고 할 경우, 피어 시스템에서 이 파일을 편집하고 `in.iked` 데몬을 다시 시작합니다.

먼저, 미리 공유한 키를 사용하는 두 서브넷의 모든 시스템에서 관리자는 미리 공유한 키 항목을 변경합니다.

```
# pfedit -s /etc/inet/secret/ike.preshared
…
{ localidtype IP
 localid 192.168.116.0/24
 remoteidtype IP
 remoteid 192.168.13.0/24
 # The two subnet's shared passphrase for keying material
key "LOooong key Th@t m^st Be Ch*angEd \'reguLarLy)"
 }
```

그런 다음 관리자는 모든 시스템에서 IKEv1 서비스를 다시 시작합니다.

`pfedit` 명령의 옵션에 대한 자세한 내용은 `pfedit`(1M) 매뉴얼 페이지를 참조하십시오.

```
# svcadm enable ipsec/ike:default
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ 새 피어 시스템에 대한 IKEv1을 업데이트하는 방법

같은 피어 간의 작업 구성에 IPsec 정책 항목을 추가할 경우에는 IPsec 정책 서비스를 새로 고쳐야 합니다. IKEv1은 재구성하거나 다시 시작하지 않아도 됩니다.

IPsec 정책에 새 피어를 추가할 경우 IPsec 변경 외에 IKEv1 구성도 수정해야 합니다.

시작하기 전에 `ipsecinit.conf` 파일을 업데이트했으며 피어 시스템에 대한 IPsec 정책을 새로 고쳤습니다.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **IPsec을 사용 중인 새 시스템의 키를 관리할 IKEv1에 대한 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 IKEv1 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:default
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 “IPsec를 사용하여 VPN 보호” [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

공개 키 인증서로 IKEv1 구성

공개 키 인증서를 사용하면 통신 시스템이 아웃오브밴드에서 보안 키 입력 도구를 공유할 필요가 없습니다. CA(인증 기관)의 공개 인증서를 사용하려면 일반적으로 외부 조직과의 협상이 필요합니다. 간편한 인증서 확장을 통해 통신하는 여러 시스템을 보호할 수 있습니다.

또한 공개 키 인증서를 생성하여 연결된 하드웨어에 저장할 수 있습니다. 절차는 “연결된 하드웨어를 찾도록 IKEv1 구성” [188]을 참조하십시오.

모든 인증서에는 X.509 distinguished name(DN, 고유 이름) 형식의 고유한 이름이 있습니다. 또한 인증서에는 전자 메일 주소, DNS 이름, IP 주소 등과 같은 주체 대체 이름이 하나 이상 있을 수 있습니다. 인증서의 전체 DN이나 주체 대체 이름 중 하나로 IKEv1 구성에서 인증서를 식별할 수 있습니다. 이러한 대체 이름의 형식은 *tag=value*입니다. 여기서 값의 형식은 태그 유형에 해당합니다. 예를 들어, `email` 태그의 형식은 *name@domain.suffix*입니다.

다음 작업 맵에는 IKEv1에 대한 공개 키 인증서를 만드는 절차가 나와 있습니다. 이 절차에서는 인증서를 빠르게 만들고 연결된 하드웨어에 저장하는 방법을 설명합니다.

표 10-1 공개 키 인증서로 IKEv1 구성 작업 맵

작업	설명	지침
자체 서명된 공개 키 인증서로 IKEv1을 구성합니다.	다음과 같은 키와 두 개의 인증서를 만들어 각 시스템에 저장합니다. ■ 자체 서명된 인증서와 해당 키 ■ 피어 시스템의 공개 키 인증서	자체 서명된 공개 키 인증서로 IKEv1을 구성하는 방법 [165]
인증 기관과 함께 IKEv1을 구성합니다.	인증서 서명 요청을 만든 다음 CA에서 받은 인증서를 각 시스템에 배치합니다. "IKE에서 공개 키 인증서 사용" [126]을 참조하십시오.	CA가 서명한 인증서로 IKEv1을 구성하는 방법 [170]
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 서명 요청을 생성한 다음 CA의 공개 키 인증서를 하드웨어에 추가합니다.	하드웨어에서 IKEv1에 대한 공개 키 인증서를 생성하고 저장하는 방법 [175]
CA의 CRL(인증서 해지 목록)을 업데이트합니다.	중앙 배포 지점에서 CRL에 액세스합니다.	IKEv1에서 해지된 인증서를 처리하는 방법 [178]

참고 - Trusted Extensions 시스템에서 패킷 및 IKE 협상에 레이블을 지정하려면 "Trusted Extensions 구성 및 관리 "의 "레이블이 있는 IPsec 구성" 절차를 따르십시오.

공개 키 인증서는 Trusted Extensions 시스템의 전역 영역에서 관리됩니다. Trusted Extensions는 인증서 관리 및 저장 방법을 변경하지 않습니다.

▼ 자체 서명된 공개 키 인증서로 IKEv1을 구성하는 방법

이 절차에서는 인증서 쌍이라고 하는 공개/개인 키와 인증서를 만듭니다. 개인 키는 로컬 인증서 데이터베이스의 디스크에 저장되며 `ikecert certlocal` 명령을 사용하여 참조할 수 있습니다. 공개 키와 인증서는 공개 인증서 데이터베이스에 저장됩니다. 이는 `ikecert certdb` 명령을 사용하여 참조할 수 있습니다. 피어 시스템과 공개 인증서를 교환합니다. 두 인증서는 IKEv1 전송을 인증하는 데 사용됩니다.

자체 서명된 인증서는 CA의 공개 인증서보다 오버헤드가 적지만 확장이 어렵습니다. CA에서 발행한 인증서와 달리 자체 서명된 인증서는 인증서를 교환한 두 관리자가 확인해야 합니다.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. “ssh 연결을 사용하여 IPsec 정책을 원격으로 구성” 및 “Oracle Solaris 11.2의 보안 셸 액세스 관리 ”의 “보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법”에서 보안 원격 로그인 지침을 참조하십시오.

1. **각 IKEv1 시스템에서 `ike.privatekeys` 데이터베이스에 자체 서명된 인증서를 만듭니다.**

   `ikecert certlocal` 명령의 인수는 `ikecert`(1M) 매뉴얼 페이지를 참조하십시오.

   a. **예를 들어, `partym` 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha512 \
-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-----
```

   구문 설명

   `-ks` 자체 서명된 인증서를 만듭니다.

   `-m` *keysize* 키의 크기를 지정합니다.

   `-t` *keytype* 사용할 알고리즘의 유형을 지정합니다.

   `-D` *dname* 인증서 주체의 X.509 DN(고유 이름)을 지정합니다. 예는 “IKE에서 공개 키 인증서 사용” [126]을 참조하십시오.

   `-A` *altname* 인증서의 대체 이름 또는 별명을 지정합니다. *altname*은 *tag=value* 형식입니다. 유효한 태그는 `IP`, `DNS`, `email` 및 `DN`입니다.

   참고 - `-D` 및 `-A` 옵션 값은 시스템이 아니라 인증서만 식별하는 이름입니다(예: `192.168.13.213`). 실제로 이러한 값은 인증서 별칭이므로 피어 시스템에 올바른 인증서가 설치되어 있는지 아웃오브밴드로 확인해야 합니다.

   b. **`enigma` 시스템의 명령은 다음과 유사하게 표시됩니다.**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha512 \
-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 메일 메시지에 붙여 넣어도 안전합니다. 수신자는 4단계와 같이 올바른 인증서를 설치했는지 대역 외 연결에서 확인해야 합니다.

   a. **예를 들어, `partym` 인증서의 공개 부분을 `enigma` 관리자에게 보냅니다.**

   ```
   To: admin@enigma.ja.example.com
   From: admin@party.us.example.com
   Message: -----BEGIN X509 CERTIFICATE-----
   MIIC1TCCAb2gAwIBAgIEfdZgKjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
   a...+
   zBGi4QkNdI3f
   -----END X509 CERTIFICATE------
   ```

   b. **`enigma` 관리자로부터 `enigma` 인증서의 공개 부분을 받습니다.**

   ```
   To: admin@party.us.example.com
   From: admin@enigma.ja.example.com
   Message: ----BEGIN X509 CERTIFICATE-----
   MIIC1TCCAb2gAwIBAgIEBl5JnjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
   ...
   y85m6LHJYtC6
   -----END X509 CERTIFICATE-----
   ```

3. **각 시스템에서 공개 키 데이터베이스에 수신한 인증서를 추가합니다.**

   a. **`root`가 읽는 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 관리자에게 인증서를 보냈는지 확인합니다.**

   예를 들어, 다른 관리자와 전화 통화를 통해 수신한 공개 인증서의 해시가 해당 관리자만 가진 개인 인증서의 해시와 일치하는지 확인할 수 있습니다.

   a. **`partym`에 저장된 인증서를 나열합니다.**

   다음 예에서 Note 1은 슬롯 0에 있는 인증서의 distinguished name(DN, 고유 이름)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 합

니다. 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-sha512
        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 모두 신뢰합니다.**

   인증서가 인식되도록 /etc/inet/ike/config 파일을 편집합니다.

   원격 시스템의 관리자가 cert_trust, remote_addr 및 remote_id 매개변수의 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-enigma"
 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-enigma 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

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

6. **피어 시스템에서 IKEv1을 사용으로 설정합니다.**

```
partym # svcadm enable ipsec/ike:default
enigma # svcadm enable ipsec/ike
```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ CA가 서명한 인증서로 IKEv1을 구성하는 방법

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **`ikecert certlocal -kc` 명령을 사용하여 CSR(인증서 서명 요청)을 만듭니다.**
   명령 인수에 대한 설명은 자체 서명된 공개 키 인증서로 IKEv1을 구성하는 방법 [165]의 1단계를 참조하십시오.

```
# ikecert certlocal -kc -m keysize -t keytype \
-D dname -A altname
```

   a. **예를 들어, 다음 명령은 `partym` 시스템에서 CSR을 만듭니다.**

```
# ikecert certlocal -kc -m 2048 -t rsa-sha384 \
> -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 CSR을 만듭니다.**

```
# ikecert certlocal -kc -m 2048 -t rsa-sha384 \
> -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. **CSR을 CA에 제출합니다.**

CA에서 CSR 제출 방법을 알려 줄 수 있습니다. 대부분 조직에는 제출 양식을 제공하는 웹 사이트가 있습니다. 양식을 사용하려면 제출이 적합한지 증명해야 합니다. 일반적으로 양식에 CSR을 붙여 넣습니다. 조직에서 요청을 확인하면 서명된 인증서를 발행합니다. 자세한 내용은 “IKE에서 공개 키 인증서 사용” [126]을 참조하십시오.

3. **각 인증서를 시스템에 추가합니다.**

`ikecert certdb -a`에 대한 `-a` 옵션은 붙여 넣은 객체를 시스템의 적합한 인증서 데이터베이스에 추가합니다. 자세한 내용은 “IKE와 공개 키 인증서” [125]를 참조하십시오.

a. **관리자로 로그인합니다.**

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오. 원격으로 관리하는 경우 예 7-1 “ssh 연결을 사용하여 IPsec 정책을 원격으로 구성” 및 “Oracle Solaris 11.2의 보안 셸 액세스 관리 ”의 “보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법”에서 보안 원격 로그인 지침을 참조하십시오.

b. **CA에서 받은 공개 키와 해당 인증서를 추가합니다.**

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

c. **CA의 공개 인증서를 추가합니다.**

중간 인증서도 추가해야 할 수 있습니다.

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

d. **CA에서 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
Press Control-D
```

4. **`/etc/inet/ike/config` 파일의 `cert_root` 키워드를 사용하여 인증서를 발행한 CA를 식별합니다.**

   CA 인증서의 DN(고유 이름)을 사용합니다.

   a. **예를 들어, `partym` 시스템의 `ike/config` 파일은 다음과 유사하게 표시될 수 있습니다.**

```
# Trusted root cert
# This certificate is from Example CA
# This is the X.509 distinguished name for the CA's cert

cert_root "C=US, O=ExampleCA\, Inc., OU=CA-Example, CN=Example CA"

## 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-enigma - Example CA"
 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}
}
```

참고 - `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=ExampleCA\, Inc., OU=CA-Example, CN=Example CA"
…
{
 label "JA-enigma to US-partym - Example CA"
 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. **해지된 인증서를 처리하는 IKEv1 정책을 설정합니다.**

   적합한 옵션을 선택합니다.

   - **OCSP를 사용할 수 없음**

     공개 키 인증서에 OCSP 서버에 연결하는 URI가 제공되지만 시스템에서 인터넷에 연결할 수 없는 경우 `ike/config` 파일에 `ignore_ocsp` 키워드를 추가합니다.

     ```
     # Trusted root cert
     …
     cert_root "C=US, O=ExampleCA\, Inc., OU=CA-Example,…
     ignore_ocsp
     …
     ```

     `ignore_ocsp` 키워드를 지정하면 IKEv1에서는 인증서가 유효하다고 간주합니다.

   - **CRL을 사용할 수 없음**

     CA에서 CRL에 대한 신뢰할 수 있는 소스를 제공하지 않거나 시스템에서 인터넷에 연결하여 CRL을 검색할 수 없는 경우 `ike/config` 파일에 `ignore_crls` 키워드를 추가합니다.

     ```
     # Trusted root cert
     …
     cert_root "C=US, O=ExampleCA\, Inc., OU=CA-Example,…
     ignore_crls
     …
     ```

   - **CRL 또는 OCSP에 대한 URI를 사용할 수 있음**

     CA에서 해지된 인증서에 대한 중앙 배포 지점을 제공하는 경우 URI를 사용하도록 `ike/config` 파일을 수정할 수 있습니다.

     예는 IKEv1에서 해지된 인증서를 처리하는 방법 [178]을 참조하십시오.

예 10-2 IKEv1 구성 시 rsa_encrypt 사용

ike/config 파일의 auth_method rsa_encrypt를 사용할 경우 publickeys 데이터베이스에 피어의 인증서를 추가해야 합니다.

1. 원격 시스템의 관리자에게 인증서를 보냅니다.

   이 인증서를 전자 메일 메시지에 붙여 넣을 수 있습니다.

   예를 들어, partym 관리자는 다음 메시지를 보냅니다.

   ```
   To: admin@enigma.ja.example.com
   From: admin@party.us.example.com
   Message: -----BEGIN X509 CERTIFICATE-----
   MII…
   ----END X509 CERTIFICATE-----
   ```

   enigma 관리자는 다음 메시지를 보냅니다.

   ```
   To: admin@party.us.example.com
   From: admin@enigma.ja.example.com
   Message: -----BEGIN X509 CERTIFICATE-----
   MII
   …
   -----END X509 CERTIFICATE-----
   ```

2. 각 시스템에서 전자 메일로 전송된 인증서를 로컬 publickeys 데이터베이스에 추가합니다.

   ```
   # ikecert certdb -a < /tmp/saved.cert.eml
   ```

RSA 암호화에 대한 인증 방법은 IKE에서 도청자에게 ID를 숨깁니다. rsa_encrypt 메소드는 피어의 ID를 숨기므로 IKEv1이 피어의 인증서를 검색할 수 없습니다. 즉, rsa_encrypt 메소드를 사용하려면 IKEv1 피어가 상대의 공개 키를 알고 있어야 합니다.

따라서 /etc/inet/ike/config 파일에 있는 rsa_encrypt의 auth_method를 사용할 경우 publickeys 데이터베이스에 피어의 인증서를 추가해야 합니다. 그러면 publickeys 데이터베이스에는 통신하는 시스템 쌍의 각각에 대해 다음 세 개의 인증서가 포함됩니다.

- 공개 키 인증서
- CA의 인증서 체인
- 피어의 공개 키 인증서

문제 해결 - IKEv1 페이로드는 인증서 세 개 이상을 포함하므로 너무 커져서 rsa_encrypt를 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. VPN을 보호하는 IPsec 정책의 예는 “IPsec를 사용하여 VPN 보

호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ 하드웨어에서 IKEv1에 대한 공개 키 인증서를 생성하고 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 하드웨어에서는 `/usr/lib/libpkcs11.so` 라이브러리를 사용합니다. 라이브러리는 RSA Security Inc. PKCS #11 암호화 토큰 인터페이스(Cryptoki), 즉 PKCS #11 라이브러리 표준에 따라 구현되어 있어야 합니다.

  설정 지침은 Sun Crypto Accelerator 6000 보드를 찾도록 IKEv1을 구성하는 방법 [188]을 참조하십시오.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **자체 서명된 인증서 또는 CSR을 생성하고 토큰 ID를 지정합니다.**

참고 - Sun Crypto Accelerator 6000 보드는 RSA에 대해 최대 2048비트의 키를 지원합니다. DSA의 경우 이 보드는 최대 1024비트의 키를 지원합니다.

다음 옵션 중 하나를 선택합니다.

- **자체 서명된 인증서의 경우 다음 구문을 사용합니다.**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha512 \
> -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.

- **CSR의 경우 다음 구문을 사용합니다.**

```
# ikecert certlocal -kc -m 2048 -t rsa-sha512 \
> -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 이름, 콜론 및 사용자 암호를 입력합니다.**

   Sun Crypto Accelerator 6000 보드에 암호가 `rgm4tigt`인 사용자 `ikemgr`이 있을 경우 다음을 입력합니다.

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
```

참고 - `ikecert` 명령을 입력할 때 `-p` 옵션을 지정하면 PKCS #11 토큰이 디스크에 일반 텍스트로 저장되며 `root` 권한으로 보호됩니다. PIN을 디스크에 저장하지 않는 경우 `in.iked` 명령을 실행한 후 `ikeadm` 명령을 사용하여 토큰을 잠금 해제해야 합니다.

   암호를 입력한 후 인증서에서 다음과 같은 출력을 인쇄합니다.

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
-----BEGIN X509 CERTIFICATE-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
…
oKUDBbZ9O/pLWYGr
-----END X509 CERTIFICATE-----
```

3. **인증서를 상대방에게 보냅니다.**

   다음 옵션 중 하나를 선택합니다.

   - **원격 시스템에 자체 서명된 인증서를 보냅니다.**

     이 인증서를 전자 메일 메시지에 붙여 넣을 수 있습니다.

   - **CSR을 certificate authority(CA, 인증 기관)에 보냅니다.**

     CA의 지침에 따라 인증서 요청을 제출합니다. 자세한 설명은 CA가 서명한 인증서로 IKEv1을 구성하는 방법 [170]의 2단계를 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-enigma 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**

CA에서 `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 CA
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExampleCA\, Inc., OU=CA-Example, CN=Example CA"

…
{
 label "JA-enigma to US-partym - Example CA"
 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.**

   2단계에서 응답한 대로 PIN 요청에 응답합니다.

참고 - 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.

- **CA의 인증서**

  CA가 인증서 요청에서 생성한 인증서와 조직의 인증서를 추가합니다.

  중간 인증서도 추가해야 할 수 있습니다.

  ```
  # 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
  ```

  CA의 CRL(인증서 해지 목록)을 추가하려면 IKEv1에서 해지된 인증서를 처리하는 방법 [178]을 참조하십시오.

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

▼ IKEv1에서 해지된 인증서를 처리하는 방법

해지된 인증서는 일정한 이유로 손상된 인증서입니다. 해지된 인증서를 사용하면 보안상 위험합니다. 인증서 해지 확인 시 옵션을 선택할 수 있습니다. 정적 목록을 사용할 수도 있고 HTTP 프로토콜을 통해 해지를 동적으로 확인할 수도 있습니다. 해지된 인증서를 처리하는 방법은 4가지입니다.

- URI(Uniform Resource Indicator)가 인증서에 포함된 CRL 또는 OCSP를 무시하도록 IKEv1에 알릴 수 있습니다. 이 옵션은 5단계에 나옵니다.
- CA의 공개 키 인증서에 주소가 포함된 URI(Uniform Resource Indicator)를 통해 CRL 또는 OCSP에 액세스하도록 IKEv1에 알릴 수 있습니다.

- CA의 공개 키 인증서에 DN(디렉토리 이름) 항목이 포함된 LDAP 서버에서 CRL에 액세스하도록 IKEv1에 알릴 수 있습니다.
- CRL을 `ikecert certrldb` 명령의 인수로 지정할 수 있습니다. 예제는 예 10-3. “CRL을 IKEv1에 대한 로컬 `certrldb` 데이터베이스에 붙여 넣기”를 참조하십시오.

시작하기 전에 Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **CA에서 수신한 인증서를 표시합니다.**

   `ikecert certdb` 명령의 인수에 대한 자세한 내용은 `ikecert`(1M) 매뉴얼 페이지를 참조하십시오.

   예를 들어, 다음은 회사의 PKI에서 발행한 인증서입니다. 세부 정보는 변경되었습니다.

```
# ikecert certdb -lv cert-protect.example.com
Certificate Slot Name: 0   Type: dsa-sha256
   (Private key in certlocal slot )
 Subject Name: <O=Example, CN=cert-protect.example.com>
 Issuer Name: <CN=ExampleCo CO (Cl B), O=Example>
 SerialNumber: 14000D93
   Validity:
      Not Valid Before: 2013 Sep 19th, 21:11:11 GMT
      Not Valid After:  2017 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 = cert-protect.example.com
      Key Usage: DigitalSignature KeyEncipherment
      [CRITICAL]
   CRL Distribution Points:
      Full Name:
         URI = #Ihttp://www.example.com/pki/pkismica.crl#i
         DN = <CN=ExampleCo CO (Cl B), O=Example>
      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 or OCSP from organization's URI
     use_http
     …
     ```

   - **웹 프록시 사용**

     `proxy` 키워드를 `ike/config` 파일에 추가합니다. `proxy` 키워드는 다음에서와 같이 URL을 인수로 사용합니다.

     ```
     # Use web proxy to reach CRLs or OCSP
     proxy "http://proxy1:8080"
     ```

   - **LDAP 서버 사용**

     호스트의 `/etc/inet/ike/config` 파일에서 LDAP 서버를 `ldap-list` 키워드의 인수로 지정합니다. 조직에서 LDAP 서버의 이름을 제공합니다. `ike/config` 파일의 항목은 다음과 유사하게 표시됩니다.

     ```
     # Use CRL from organization's LDAP
     ldap-list "ldap1.example.com:389,ldap2.example.com"
     …
     ```

   IKE가 CRL을 검색하고 인증서가 만료될 때까지 CRL을 캐시합니다.

예 10-3 CRL을 IKEv1에 대한 로컬 `certrldb` 데이터베이스에 붙여 넣기

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

```
# ikecert certrldb -a < ExampleCo.Cert.CRL
```

모바일 시스템에 대한 IKEv1 구성

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 고

유한 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 CA의 인증서를 통해 중앙 사이트와 통신할 수 있습니다.

다음 작업 맵에서는 원격으로 중앙 사이트에 로그인하는 시스템을 처리하도록 IKEv1을 구성하는 절차에 대해 설명합니다.

표 10-2 모바일 시스템에 대한 IKEv1 구성 작업 맵

작업	설명	지침
오프사이트의 중앙 사이트와 통신합니다.	오프사이트 시스템이 중앙 사이트와 통신할 수 있도록 합니다. 오프사이트 시스템은 모바일일 수 있습니다.	오프사이트 시스템에 대해 IKEv1을 구성하는 방법 [181]
모바일 시스템의 트래픽을 허용하는 중앙 시스템에서 CA의 공개 인증서 및 IKEv1을 사용합니다.	고정 IP 주소가 없는 시스템의 IPsec 트래픽을 승인하도록 게이트웨이 시스템을 구성합니다.	예 10-4. "IKEv1을 사용하여 모바일 시스템의 보호된 트래픽을 허용하도록 중앙 컴퓨터 구성"
고정 IP 주소가 없는 시스템에서 CA의 공개 인증서 및 IKEv1을 사용합니다.	회사 본사 등의 중앙 사이트에 대한 트래픽을 보호하도록 모바일 시스템을 구성합니다.	예 10-5. "NAT 뒤에서 IPsec 및 IKEv1을 사용하여 시스템 구성"
모바일 시스템의 트래픽을 허용하는 중앙 시스템에서 자체 서명된 인증서 및 IKEv1을 사용합니다.	모바일 시스템의 IPsec 트래픽을 승인하도록 자체 서명된 인증서로 게이트웨이 시스템을 구성합니다.	예 10-6. "모바일 시스템의 자체 서명된 인증서 승인"
고정 IP 주소가 없는 시스템에서 자체 서명된 인증서 및 IKEv1을 사용합니다.	중앙 사이트에 대한 트래픽을 보호하도록 자체 서명된 인증서로 모바일 시스템을 구성합니다.	예 10-7. "자체 서명된 인증서를 사용하여 중앙 시스템에 연결"

▼ 오프사이트 시스템에 대해 IKEv1을 구성하는 방법

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오. 원격으로 관리하는 경우 예 7-1. "`ssh` 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 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. **IKEv1 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` 파일에 모바일 시스템의 주소가 없어도 되지만 지정해도 됩니다. 중앙 시스템 *central*에 대한 공용 IP 주소는 포함해야 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. **IKEv1 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 시스템 *mobile*을 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:default` 서비스를 사용으로 설정합니다.**

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

예 10-4 IKEv1을 사용하여 모바일 시스템의 보호된 트래픽을 허용하도록 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 10-5. “NAT 뒤에서 IPsec 및 IKEv1을 사용하여 시스템 구성”를 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=ExampleCA Inc, OU=CA-Example, CN=Example CA"
#
# 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 NAT 뒤에서 IPsec 및 IKEv1을 사용하여 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 10-4. “IKEv1을 사용하여 모바일 시스템의 보호된 트래픽을 허용하도록 중앙 컴퓨터 구성”를 참조하십시오.

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExampleCA Inc, OU=CA-Example, CN=Example CA"
#
# 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 10-7. “자체 서명된 인증서를 사용하여 중앙 시스템에 연결”을 참조하십시오.

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Self-signed certificates - trust me and enumerated others
cert_trust    "DNS=main1.domain.org"
cert_trust    "jdoe@domain.org"
cert_trust    "user2@domain.org"
cert_trust    "user3@domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site systems with trusted certificates"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# Trust the self-signed certificates
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}
```

예 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 10-6. “모바일 시스템의 자체 서명된 인증서 승인”을 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. VPN을 보호하는 IPsec 정책의 예는 “IPsec를 사용하여 VPN 보호” [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

연결된 하드웨어를 찾도록 IKEv1 구성

공개 키 인증서를 연결된 하드웨어에 저장할 수도 있습니다. Sun Crypto Accelerator 6000 보드는 저장소를 제공하고 공개 키 작업이 시스템에서 보드로 오프로드될 수 있도록 합니다.

▼ Sun Crypto Accelerator 6000 보드를 찾도록 IKEv1을 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 Product Library Documentation (http://docs.oracle.com/cd/E19321-01/index.html)을 참조하십시오.

Network IPsec Management 권한 프로파일에 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

원격으로 관리하는 경우 예 7-1. "ssh 연결을 사용하여 IPsec 정책을 원격으로 구성" 및 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법"에서 보안 원격 로그인 지침을 참조하십시오.

1. **PKCS #11 라이브러리가 연결되어 있는지 확인합니다.**

   IKEv1에서는 라이브러리 루틴을 사용하여 키를 생성하고 Sun Crypto Accelerator 6000 보드에 키를 저장합니다.

   ```
   $ ikeadm get stats
   …
   PKCS#11 library linked in from /usr/lib/libpkcs11.so
   $
   ```

2. **연결된 Sun Crypto Accelerator 6000 보드의 토큰 ID를 찾습니다.**

   ```
   $ ikecert tokens
   Available tokens with library "/usr/lib/libpkcs11.so":

   "Sun Metaslot                    "
   ```

   라이브러리가 32자의 토큰 ID(keystore name(키 저장소 이름)이라고도 함)를 반환합니다. 이 예에서는 `ikecert` 명령에 `Sun Metaslot` 토큰을 사용하여 IKEv1 키를 저장하고 속도를 향상시킬 수 있습니다.

   토큰 사용 방법에 대한 지침은 하드웨어에서 IKEv1에 대한 공개 키 인증서를 생성하고 저장하는 방법 [175]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. VPN을 보호하는 IPsec 정책의 예는 "IPsec를 사용하여 VPN 보호" [106]를 참조하십시오. 다른 IPsec 정책 예는 IPsec을 사용하여 두 서버 간의 네트워크 트래픽을 보호하는 방법 [100]을 참조하십시오.

♦ ♦ ♦ 11 장

11 IPsec 및 해당 키 관리 서비스 문제 해결

이 장에서는 IPsec 및 해당 키 관련 문제를 해결하는 방법, 구성 정보를 보는 방법, 활성 IPsec, IKE 및 수동 키 서비스에 대한 정보를 보는 방법에 대해 설명합니다.

이 장은 다음 정보를 포함합니다.

IPsec 및 해당 키 관리 구성 문제 해결

해결해야 하는 문제가 발생하기 전이나 문제를 해결하는 동안 시스템에서 문제 해결 관련 설정을 할 수 있습니다.

문제 해결할 때 프로파일 셸에서 네트워크 IPsec 관리 권한 프로파일을 보유한 관리자로 여러 명령을 실행할 수 있습니다. 그러나 로그를 읽으려면 `root` 역할이 있어야 합니다.

문제 해결 섹션의 프롬프트를 보면 권한이 있어야만 명령을 실행할 수 있는지 여부를 알 수 있습니다.

- `#` 프롬프트 - 적절한 관리 권한이 있는 사용자나 이러한 권한이 있는 역할이 명령을 실행할 수 있습니다.
- `%` 프롬프트 - 일반 사용자가 명령을 실행할 수 있습니다.

▼ 문제 해결을 위해 IPsec 및 IKE 시스템을 준비하는 방법

IPsec 및 해당 키 관리 서비스를 사용으로 설정하기 전에 시스템에서 문제 해결에 도움이 되는 로그 및 도구를 설정할 수 있습니다.

1. **IPsec 및 IKEv2 서비스에 대한 도구를 찾습니다.**

-L 옵션을 사용하면 로그의 전체 경로가 제공됩니다. 이러한 로그에는 정보 메시지 및 오류 메시지가 포함됩니다.

```
% svcs -L policy
/var/svc/log/network-ipsec-policy:default.log

% svcs -L ikev2
/var/svc/log/network-ipsec-ike:ikev2.log
```

2. **IKEv2에 대한 디버그 로그 파일을 구성합니다.**
root 역할은 이러한 로그를 읽을 수 있습니다.

```
% svccfg -s ikev2 listprop | grep debug
config/debug_level            astring     op
config/debug_logfile          astring     /var/log/ikev2/in.ikev2d.log
```

디버깅 레벨에 대한 설명은 ikeadm(1M) 매뉴얼 페이지를 참조하십시오. 문제 해결할 때 verbose 및 all 값을 사용하면 도움이 됩니다.

3. **(옵션) 디버그 레벨을 구성합니다.**
다음 명령은 디버그 레벨을 영구적으로 설정합니다. 디버그 레벨을 일시적으로 설정하려면 예 11-3. “실행 중인 IKE 데몬에서 새 디버그 레벨 설정”을 참조하십시오.

```
# svccfg -s ikev2 setprop config/debug_level = all
```

ikev2 서비스를 사용으로 설정한 경우 새로 고쳐야만 새 디버그 레벨을 사용할 수 있습니다.

```
# svcadm refresh ikev2
```

4. **(옵션) wireshark 패키지를 설치합니다.**
Wireshark 응용 프로그램에서는 snoop 출력을 읽을 수 있습니다.

```
% pkg info -r wireshark
          Name: diagnostic/wireshark
       Summary: Graphical network protocol analyzer
      Category: Applications/Internet
         State: Not installed
     Publisher: solaris
...
          FMRI: pkg://solaris/diagnostic/wireshark@version
# pkg install diagnostic/wireshark
```

▼ IPsec 및 IKE를 실행하기 전에 시스템 문제를 해결하는 방법

서비스를 실행하기 전에 IPsec 구성 파일의 구문, IPsec 키 파일 및 키 저장소에 있는 인증서의 유효성을 확인할 수 있습니다.

1. **IPsec 구성 파일의 구문을 확인합니다.**

```
# ipsecconf -c /etc/inet/ipsecinit.conf
ipsecconf: Invalid pattern on line 5: ukp
ipsecconf: form_ipsec_conf error
ipsecconf: Malformed command (fatal):
{ ukp 58 type 133-137 dir out} pass {}

ipsecconf: 1 policy rule(s) contained errors.
ipsecconf: Fatal error - exiting.
```

   출력에 오류가 표시되는 경우 검증이 성공할 때까지 오류를 수정하고 명령을 실행합니다.

2. **ipseckeys 파일의 구문을 확인합니다.**

```
# ipseckey -c /etc/inet/secret/ipseckeys
Config file /etc/inet/secret/ipseckeys has insecure permissions,
will be rejected in permanent config.
```

   출력에 오류가 표시되는 경우 오류를 수정한 다음 서비스를 새로 고칩니다.

```
# svcadm refresh ipsec/policy
```

   참고 - IKE 구성 파일 및 IKE 미리 공유한 키 파일은 실행 중인 IKE 데몬에 의해 검증됩니다.

3. **인증서의 유효성을 확인합니다.**

   - IKEv2에서 자체 서명된 인증서의 유효성을 확인하려면 자체 서명된 공개 키 인증서로 IKEv2를 구성하는 방법 [142]의 4단계를 수행합니다.
   - IKEv2에서 공개 키 인증서가 해지되지 않았는지 확인하려면 IKEv2에서 인증서 검증 정책을 설정하는 방법 [150] 절차를 따릅니다.
   - IKEv1에서 자체 서명된 인증서의 유효성을 확인하려면 자체 서명된 공개 키 인증서로 IKEv1을 구성하는 방법 [165]의 4단계를 수행합니다.
   - IKEv1에서 공개 키 인증서가 해지되지 않았는지 확인하려면 IKEv1에서 해지된 인증서를 처리하는 방법 [178] 절차를 따릅니다.

다음 순서 IPsec 및 해당 키 입력 서비스를 사용으로 설정할 때 구성이 작동하지 않는 경우에는 서비스가 실행 중인 동안 문제를 해결해야 합니다.

▼ IPsec이 실행 중일 때 시스템 문제를 해결하는 방법

IKE를 사용하여 패킷을 교환 중이거나 교환하려고 하는 실행 중인 시스템에서 `ikeadm` 명령을 사용하여 통계, 규칙, 미리 공유한 키 및 기타 항목을 볼 수 있습니다. 또한 Wireshark 응용 프로그램과 같은 선택한 도구 및 로그 파일도 사용할 수 있습니다.

1. **다음 항목을 조사합니다.**

■ **policy 및 적절한 키 관리 서비스를 사용으로 설정했는지 확인합니다.**

다음 테스트 시스템에서는 manual-key 서비스가 키 관리에 사용됩니다.

```
% svcs -a  | grep ipsec
online         Feb_04   svc:/network/ipsec/manual-key:default
online         Feb_04   svc:/network/ipsec/ipsecalgs:default
online         Feb_04   svc:/network/ipsec/policy:default
disabled       Feb_28   svc:/network/ipsec/ike:ikev2
disabled       Feb_28   svc:/network/ipsec/ike:default
```

서비스를 사용할 수 없는 경우 사용으로 설정합니다.

두 IKE 서비스를 동시에 사용할 수 있습니다. 수동 키와 IKE를 동시에 사용할 수도 있지만, 이 구성을 사용할 경우 해결하기 어려운 특이한 문제가 발생할 수 있습니다.

■ **IKEv2 서비스에 대한 로그 파일의 끝을 봅니다.**

```
# svcs -xL ikev2
svc:/network/ipsec/ike:ikev2 (IKEv2 daemon)
 State: disabled since October  10, 2013 10:10:40 PM PDT
Reason: Disabled by an administrator.
   See: http://support.oracle.com/msg/SMF-8000-05
   See: in.ikev2d(1M)
   See: /var/svc/log/network-ipsec-ike:ikev2.log
Impact: This service is not running.
   Log:
Oct 01 13:20:20: (1)  Property "debug_level" set to: "op"
Oct 01 13:20:20: (1)  Errors and debug messages will be written to:
                        /var/log/ikev2/in.ikev2d.log
[ Oct 10 10:10:10 Method "start" exited with status 0. ]
[ Oct 10 10:10:40 Stopping because service disabled. ]
[ Oct 10 10:10:40 Executing stop method (:kill). ]

   Use: 'svcs -Lv svc:/network/ipsec/ike:ikev2' to view the complete log.
```

■ **(옵션) 실행 중인 데몬의 디버그 레벨로 임시 값을 설정할 수 있습니다.**

```
# ikeadm set debug verbose /var/log/ikev2/in.ikev2d.log
Successfully changed debug level from 0x80000000 to 0x6204
Debug categories enabled:
        Operational / Errors
        Config file processing
        Interaction with Audit
        Verbose Operational
```

2. **ipsecconf 명령의 출력이 정책 파일의 내용과 일치하는지 확인합니다.**

```
# ipsecconf
#INDEX 14
...
{  laddr 10.133.66.222 raddr 10.133.64.77 }
```

```
 ipsec   { encr_algs aes(256) encr_auth_algs sha512 sa shared }
...
{  laddr 10.134.66.122 raddr 10.132.55.55 }
 ipsec   { encr_algs aes(256) encr_auth_algs sha512 sa shared }

# cat /etc/inet/ipsecinit.conf
...
{  laddr 10.133.66.222 raddr 10.133.64.77 }
 ipsec   { encr_algs aes(256) encr_auth_algs sha512 sa shared }

{  laddr 10.134.66.122 raddr 10.132.55.55 }
 ipsec   { encr_algs aes(256) encr_auth_algs sha512 sa shared }
```

참고 - 와일드카드 주소를 사용하면 일치 항목을 찾기가 어려울 수 있으므로, `ipsecconf`의 출력에서 `ipsecinit.conf` 파일의 특정 주소가 와일드카드 주소 범위 내에 있는지 확인합니다.

`ipsecconf` 명령의 출력이 인쇄되지 않는 경우에는 정책 서비스가 사용으로 설정되었는지 확인하고 서비스를 새로 고칩니다.

```
% svcs policy
STATE          STIME    FMRI
online         Apr_10   svc:/network/ipsec/policy:default
```

출력에 오류가 표시되는 경우에는 `ipsecinit.conf` 파일을 편집하여 오류를 수정한 다음 서비스를 새로 고칩니다.

3. **IKEv2 구성을 검증합니다.**

   구성 출력을 수정해야 하는 경우 예 11-1. “잘못된 IKEv2 구성 수정” 및 예 11-2. “일치하는 규칙 없음 메시지 수정”를 참조하십시오. 다음 예의 출력에는 구성이 유효하다고 나옵니다.

```
# /usr/lib/inet/in.ikev2d -c
Feb 04 12:08:25: (1)    Reading service properties from smf(5) repository.
Feb 04 12:08:25: (1)    Property "config_file" set to: "/etc/inet/ike/ikev2.config"
Feb 04 12:08:25: (1)    Property "debug_level" set to: "all"
Feb 04 12:08:25: (1)    Warning: debug output being written to stdout.
Feb 04 12:08:25: (1)    Checking IKE rule #1: "Test 104 to 113"
Feb 04 12:08:25: (1)    Configuration file /etc/inet/ike/ikev2.config is valid.
Feb 04 12:08:25: (1)    Pre-shared key file /etc/inet/ike/ikev2.preshared is valid.
```

참고 - 디버그 출력에 대한 경고는 디버그 로그 파일을 지정한 후에도 변경되지 않습니다. `debug_logfile` 등록 정보 값을 지정하는 경우 경고는 디버그 출력이 해당 파일로 제공됨을 의미합니다. 그렇지 않은 경우 디버그 출력이 콘솔로 제공됩니다.

- `Checking IKE rule` 행에서 IKE 규칙에서 적절한 IP 주소를 연결하는지 확인합니다. 예를 들어, 다음 항목은 일치합니다. `ipsecinit.conf` 파일의 `laddr` 값이 `ikev2.config` 파일의 `local_addr` 값과 일치하고 원격 주소가 일치합니다.

```
{  laddr 10.134.64.104 raddr 10.134.66.113 }      /** ipsecinit.conf **/
                ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

```
local_addr   10.134.64.104                          /** ikev2.config **/
remote_addr  10.134.66.113                          /** ikev2.config **/
```

항목이 일치하지 않는 경우 구성을 수정하여 올바른 IP 주소를 식별합니다.

참고 - 규칙에서는 주소 범위를 포함하는 `10.134.0.0/16`과 같은 와일드카드 주소를 사용할 수 있습니다. 특정 주소에 대해 범위를 확인하십시오.

- `Pre-shared key file` 행에 파일이 유효하지 않다고 표시되는 경우 파일을 수정합니다.

  오타가 있는지 확인합니다. 또한 IKEv2에서 `ikev2.config`에 있는 규칙의 레이블 값이 `ikev2.preshared` 파일의 레이블 값과 일치하는지 확인합니다. 그런 다음 두 키를 사용 중인 경우 한 시스템의 로컬 미리 공유한 키가 피어의 원격 미리 공유한 키와 일치하고 원격 키가 피어의 로컬 키와 일치하는지 확인합니다.

  그래도 구성이 작동하지 않는 경우에는 "IPsec 및 IKE 의미 오류 문제 해결" [197]을 참조하십시오.

예 11-1 잘못된 IKEv2 구성 수정

다음 출력에서는 IKE SA의 수명이 너무 짧습니다.

```
# /usr/lib/inet/in.ikev2d -c
...
May 08 08:52:49: (1) WARNING: Problem in rule "Test 104 to 113"
May 08 08:52:49: (1)  HARD lifetime too small (60 < 100)
May 08 08:52:49: (1)   -> Using 100 seconds (minimum)
May 08 08:52:49: (1) Checking IKE rule #1: "config 10.134.13.113 to 10.134.13.104"
...
```

이 값은 `ikev2.config` 파일에서 명시적으로 설정했습니다. 경고를 제거하려면 수명 값을 `100` 이상으로 변경하고 서비스를 새로 고칩니다.

```
# pfedit /etc/inet/ike/ikev2.config
...
## childsa_lifetime_secs   60
childsa_lifetime_secs   100
...
# /usr/lib/inet/in.ikev2d -c
...
# svcadm refresh ikev2
```

예 11-2 일치하는 규칙 없음 메시지 수정

다음 출력에서는 미리 공유한 키가 정의되었지만 규칙에서 사용되지 않습니다.

```
# /usr/lib/inet/in.ikev2d -c
Feb 4 12:58:31: (1)  Reading service properties from smf(5) repository.
Feb 4 12:58:31: (1)  Property "config_file" set to: "/etc/inet/ike/ikev2.config"
```

```
Feb 4 12:58:31: (1)  Property "debug_level" set to: "op"
Feb 4 12:58:31: (1)  Warning: debug output being written to stdout.
Feb 4 12:58:31: (1)  Checking IKE rule #1: "Test 104 to 113"
Feb 4 12:58:31: (1)  Configuration file /etc/inet/ike/ikev2.config is valid.
Feb 4 12:58:31: (1)  No matching IKEv2 rule for pre-shared key ending on line 12
Feb 4 12:58:31: (1)  Pre-shared key file /etc/inet/ike/ikev2.preshared is valid.
```

출력에는 규칙이 하나만 있다고 나옵니다.

- 규칙에 미리 공유한 키가 필요한 경우 미리 공유한 키의 레이블이 규칙의 레이블과 일치하지 않습니다. `ikev2.config` 규칙 레이블과 `ikev2.preshared` 키 레이블을 일치하도록 수정합니다.
- 규칙에서 인증서를 사용하는 경우에는 `ikev2.preshared` 파일의 12행에서 끝나는 미리 공유한 키를 제거하거나 주석 처리하여 `No matching` 메시지가 표시되지 않게 할 수 있습니다.

예 11-3 실행 중인 IKE 데몬에서 새 디버그 레벨 설정

다음 출력에서는 `ikev2` 서비스에서 디버그 출력이 `all`로 설정되어 있습니다.

```
# /usr/lib/inet/in.ikev2d -c
Feb 4 12:58:31: (1)  Reading service properties from smf(5) repository.
...
Feb 4 12:58:31: (1)  Property "debug_level" set to: "all"
...
```

IPsec 및 IKE를 실행하기 전에 시스템 문제를 해결하는 방법 [192]의 2단계를 완료했는데도 디버그 출력이 `all`이 아닌 `op`인 경우 `ikeadm` 명령을 사용하여 실행 중인 IKE 데몬에서 디버그 레벨을 설정합니다.

```
# ikeadm set debug_level all
```

IPsec 및 IKE 의미 오류 문제 해결

IPsec이 실행 중일 때 시스템 문제를 해결하는 방법 [193]의 조사로 문제를 처리하지 못한 경우 파일의 구문이나 서비스 구성이 문제가 아니라 구성의 의미가 문제일 수 있습니다.

- `ike:default` 및 `ike:ikev2` 서비스 인스턴스를 모두 사용으로 설정한 경우 IKEv2 및 IKEv1 규칙이 겹치지 않는지 확인합니다. 동일한 네트워크 끝점에 규칙을 적용하면 IPsec SA가 중복되어 경우에 따라 연결이 끊어질 수도 있습니다.

  IKE 규칙을 변경하는 경우 규칙을 커널로 읽어들입니다.

  ```
  # ikeadm -v[1|2] read rule
  ```

- IKEv1을 실행 중인 경우 연결하는 IKEv1 시스템에서 규칙의 알고리즘 방식을 사용할 수 있는지 확인합니다. 사용 가능한 알고리즘을 확인하려면 IKEv2를 지원하지 않는 시스템에서 `ikeadm dump` *algorithms* 명령을 실행합니다.

  ```
  # ikeadm dump groups      Available Diffie-Hellman groups
  ```

```
# ikeadm dump encralgs      All IKE encryption algorithms
# ikeadm dump authalgs      All IKE authentication algorithms
```

IPsec 및 IKEv1 시스템 모두에서 사용 가능한 알고리즘을 사용하도록 두 시스템의 정책 파일을 모두 수정합니다. 그럼 다음 IKEv1 서비스를 다시 시작하고 IPsec 서비스를 새로 고칩니다.

```
# svcadm restart ike:default; svcadm refresh ipsec/policy
```

- IKEv1에서 미리 공유한 키를 사용 중이고 원격 IKEv1 시스템을 재부팅한 경우 로컬 시스템에서 `ipseckey flush` 명령을 실행합니다.
- 자체 서명된 인증서를 사용 중인 경우 다른 관리자와 함께 DN이 동일한 인증서를 다시 만들지 않았는지, 그리고 인증서의 해시 값이 일치하는지 확인합니다. 검증 단계는 자체 서명된 공개 키 인증서로 IKEv2를 구성하는 방법 [142]의 4단계를 참조하십시오.

  인증서가 업데이트된 경우 새 인증서를 가져온 다음 IKEv2 서비스를 새로 고치고 다시 시작합니다.
- `ikeadm -v2 dump | get` 명령을 사용하여 현재 IKEv2 구성을 봅니다. 사용법 요약은 "IKE 정보 보기" [199]를 참조하십시오.
- `kstat` 명령을 사용하여 IPsec 관련 통계를 표시합니다. 자세한 내용은 `kstat`(1M) 매뉴얼 페이지를 참조하십시오.

```
# kstat -m ipsecesp
# kstat -m ipsecah
# kstat -m ip
```

다음 예의 `kstat` 출력에는 `ipsecesp` 모듈에 문제가 없다고 나옵니다.

```
# kstat -m ipsecesp
module: ipsecesp                        instance: 0
name:   esp_stat                        class:    net
        acquire_requests                18
        bad_auth                        0
        bad_decrypt                     0
        bad_padding                     0
        bytes_expired                   0
        crtime                          4.87974774
        crypto_async                    0
        crypto_failures                 0
        crypto_sync                     172
        good_auth                       86
        keysock_in                      135
        num_aalgs                       9
        num_ealgs                       13
        out_discards                    0
        out_requests                    86
        replay_early_failures           0
        replay_failures                 0
```

```
        sa_port_renumbers               0
        snaptime                        5946769.7947628
```

- snoop 명령을 사용하여 보호되지 않는 트래픽을 봅니다. Wireshark 응용 프로그램에서는 snoop 출력을 읽을 수 있습니다. snoop 출력의 예는 IPsec로 패킷이 보호되는지 확인하는 방법 [119]을 참조하십시오.

IPsec 및 해당 키 입력 서비스에 대한 정보 보기

참고 - 대부분의 명령을 실행하려면 Network IPsec Management 권한 프로파일이 지정된 관리자여야 합니다. 프로파일 셸에서 입력해야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

IPsec 및 수동 키 서비스 등록 정보 보기

수동 키가 저장된 파일 및 IPsec 정책 파일의 이름을 볼 수 있습니다.

- IPsec 구성 파일의 이름을 표시하려면

```
% svccfg -s policy listprop config/config_file
config/config_file         astring     /etc/inet/ipsecinit.conf
```

- IPsec용 수동 키가 저장된 파일의 이름을 표시하려면

```
% svccfg -s manual-key listprop config/config_file
config/config_file         astring     /etc/inet/secret/ipseckeys
```

IKE 정보 보기

IKE 서비스의 등록 정보, IKE 상태 및 IKE 데몬 객체의 요소, 인증서 검증 정책을 볼 수 있습니다. 두 IKE 서비스를 모두 실행 중인 경우 서비스별로 정보를 표시하거나 두 서비스 정보를 모두 표시할 수 있습니다. 이러한 명령은 테스트, 문제 해결 및 모니터링 중에 유용할 수 있습니다.

- IKE 서비스 인스턴스의 등록 정보 보기 - 출력에는 구성 파일의 이름을 비롯해 IKEv2 서비스에 대해 구성 가능한 속성이 표시됩니다.

참고 - ipsecconf(1M). in.ikev2d(1M) 및 in.iked(1M) 매뉴얼 페이지를 검토하여 IPsec, IKEv2 또는 IKEv1 서비스의 config 그룹에서 등록 정보를 수정할 수 있는지 또는 수정해야 하는지 확인합니다. 예를 들어, IKEv2 구성 파일은 특수한 권한으로 만들어지며 ikeuser가 소유합니다. 이 권한과 파일 소유자는 변경하면 안됩니다.

```
% svccfg -s ipsec/ike:ikev2 listprop config
config                      application
config/allow_keydump        boolean     false
config/config_file          astring     /etc/inet/ike/ikev2.config
config/ignore_errors        boolean     false
config/kmf_policy           astring     /etc/inet/ike/kmf-policy.xml
config/max_child_sas        integer     0
config/max_threads          integer     0
config/min_threads          integer     0
config/preshared_file       astring     /etc/inet/ike/ikev2.preshared
config/response_wait_time   integer     30
config/value_authorization  astring     solaris.smf.value.ipsec
config/debug_logfile        astring
config/debug_level          astring     op
```

다음 예의 출력에는 IKEv1 서비스에 대해 구성 가능한 등록 정보가 표시됩니다. :default 서비스 인스턴스를 지정하지 마십시오.

```
% svccfg -s ipsec/ike listprop config
config                      application
config/admin_privilege      astring     base
config/config_file          astring     /etc/inet/ike/config
config/debug_level          astring     op
config/debug_logfile        astring     /var/log/in.iked.log
config/ignore_errors        boolean     false
config/value_authorization  astring     solaris.smf.value.ipsec
```

- IKE 데몬의 현재 상태 보기 - 다음 예의 출력에는 ikeadm 명령의 인수가 표시됩니다. 이러한 인수를 사용하면 데몬의 현재 상태가 표시됩니다.

참고 - ikeadm 명령을 사용하려면 IKE 데몬이 실행 중이어야 합니다.

```
% ikeadm help
...
        get   debug|priv|stats|p1|ikesa|rule|preshared|defaults [identifier]
        dump  p1|ikesa|rule|preshared|certcache|groups|encralgs|authalgs
        read  rule|preshared [filename]
        help  [get|set|add|del|dump|flush|read|write|token|help]
```

- ikeadm 명령의 특정 인수에 대한 구문 표시 - help 하위 명령을 사용하여 명령 인수 구문을 표시합니다. 예를 들면 다음과 같습니다.

```
% ikeadm help read
This command reads a new configuration file into
in.iked, discarding the old configuration info.

Sets of data that may be read include:
        rule            all phase 1/ikesa rules
        preshared       all preshared keys

A filename may be provided to specify a source file
other than the default.
```

- 미리 공유한 키 보기 - IKEv1 및 IKEv2에 대한 미리 공유한 키를 볼 수 있습니다.

참고 - IKE 버전을 하나만 실행 중인 경우 -v 옵션을 생략해도 됩니다.

IKEv2의 경우:

```
# ikeadm -v2 dump preshared
```

IKEv1의 경우:

```
# ikeadm set priv keymat
# ikeadm -v1 dump preshared

PSKEY: Rule label: "Test PSK 197 to 56"
PSKEY: Local pre-shared key (80 bytes): 74206272696c6c696720...3/584
PSKEY: Remote pre-shared key (80 bytes): 74206272696c6c696720...3/584

Completed dump of preshared keys
```

- IKE SA 보기 - 출력에는 SA, 변환, 로컬 및 원격 시스템에 대한 정보와 기타 세부 정보가 포함됩니다. 통신을 요청하지 않은 경우 SA가 없으므로 표시할 정보가 없습니다.

```
# ikeadm -v2 dump ikesa
IKESA: SPIs: Local 0xd3db95689459cca4  Remote 0xb5878717f5cfa877
...
XFORM: Encryption alg: aes-cbc(256..256); Authentication alg: hmac-sha512
...
LOCIP: AF_INET: port 500, 10.1.2.3 (example-3).
...
REMIP: AF_INET: port 500, 10.1.4.5 (ex-2).
...
LIFTM: SA expires in 11459 seconds (3.18 hours)
...
```

```
STATS: 0 IKE SA rekeys since initial AUTH.
LOCID: Initiator identity, type FQDN
...
CHILD: ESP Inbound SPI: 0x94841ca3, Outbound SPI 0x074ae1e5
...
Completed dump of IKE SA info
```

- 활성 IKE 규칙 보기 - 나열된 IKE 규칙은 사용 중이 아닐 수 있지만 사용 가능한 상태입니다.

```
# ikeadm -v2 dump rule

GLOBL: Label 'Test Rule1 for PSK', key manager cookie 1
GLOBL: Local auth method=pre-shared key
GLOBL: Remote auth method=pre-shared key

GLOBL: childsa_pfs=false
GLOBL: authentication_lifetime=86400 seconds (1.00 day)
GLOBL: childsa_lifetime=120 seconds (2.00 minutes)
GLOBL: childsa_softlife=108 seconds (1.80 minute)
GLOBL: childsa_idletime=60 seconds
GLOBL: childsa_lifetime_kb=122880 kilobytes (120.00 MB)
GLOBL: childsa_softlife_kb=110592 kilobytes (108.00 MB)
LOCIP: IP address range(s):
LOCIP: 10.142.245.197
REMIP: IP address range(s):
REMIP: 10.134.64.56
LOCID: Identity descriptors:
LOCID: Includes:
LOCID:       fqdn="gloria@ms.mag"
REMID: Identity descriptors:
REMID: Includes:
REMID:       fqdn="gloria@ms.mag"
XFRMS: Available Transforms:

XF  0: Encryption alg: aes-cbc(128..256); Authentication alg: hmac-sha512
XF  0: PRF: hmac-sha512 ; Diffie-Hellman Group: 2048-bit MODP (group 14)
XF  0: IKE SA lifetime before rekey: 14400 seconds (4.00 hours)

Completed dump of policy rules
```

- IKEv2에서 인증서 검증 정책 보기 - `dbfile` 값과 `policy` 값을 지정해야 합니다.
  - 동적으로 다운로드되는 CRL에서는 관리자가 개입하여 응답자 시간 초과를 조정해야 합니다.

    다음 예의 출력에서는 인증서에 포함된 URI에서 CRL이 다운로드된 다음 목록이 캐시됩니다. 캐시에 만료된 CRL이 포함된 경우 새 CRL이 다운로드되어 이전 CRL을 대체합니다.

```
# kmfcfg list dbfile=/etc/inet/ike/kmf-policy.xml policy=default
…
Validation Policy Information:
    Maximum Certificate Revocation Responder Timeout: 10
    Ignore Certificate Revocation Responder Timeout: true
…
    CRL:
        Base filename: [not set]
        Directory: /var/user/ikeuser/crls
        Download and cache CRL: true
        CRL specific proxy override: www-proxy.cagate.example.com:80
        Ignore CRL signature: false
        Ignore CRL validity date: false
IPsec policy bypass on outgoing connections: true
…
```

- 정적으로 다운로드되는 CRL은 관리자가 주기적으로 주의를 기울여야 합니다.

  관리자가 CRL 항목을 다음 값으로 설정하는 경우 관리자는 CRL을 수동으로 다운로드하고, 디렉토리를 채우고, 현재 CRL을 유지 관리해야 합니다.

```
…
        Directory: /var/user/ikeuser/crls
        Download and cache CRL: false
        Proxy: [not set]
…
```

IPsec 및 해당 키 입력 서비스 관리

IPsec 정책은 기본적으로 사용으로 설정되지만 구성 정보를 충분히 포함하고 있지 않습니다.

키 관리는 기본적으로 사용으로 설정되지 않습니다. IKE나 수동 키 관리 중 하나 또는 둘 다 구성할 수 있습니다. 각 IKE 규칙에서는 어떤 키 관리 서비스를 사용하는지 지정합니다. `ikeadm` 명령으로 실행 중인 IKE 데몬을 수정할 수 있습니다.

IPsec 및 해당 키 입력 서비스 구성 및 관리

- IPsec을 구성하고 새로 고친 후 정책 보기:

```
# pfedit /etc/inet/ipsecinit.conf
# ipsecconf -c /etc/inet/ipsecinit.conf
# svcadm refresh ipsec/policy
# ipsecconf -Ln
```

- IPsec에 대한 수동 키 구성 및 사용으로 설정:

```
# pfedit -s /etc/inet/secret/ipseckeys
# svcadm enable ipsec/manual-key
```

- IKEv2 구성 및 사용으로 설정:

```
# pfedit /etc/inet/ike/ikev2.config
# /usr/lib/inet/in.ikev2d -c
# svcadm enable ipsec/ike:ikev2
```

- IKEv1 구성 및 사용으로 설정:

```
# pfedit /etc/inet/ike/config
# /usr/lib/inet/in.iked -c
# svcadm enable ipsec/ike:default
```

- 서비스를 사용으로 설정한 시스템에서 IPsec 및 IKE가 구성되어 있는지 확인:

```
# ipsecconf -Ln
# ikeadm -v2 dump rule
# ikeadm set priv keymat
# ikeadm -v1 dump rule
```

- 키 관리 수정:

  IKEv2의 경우:

```
# pfedit /etc/inet/ike/ikev2.config
# /usr/lib/inet/in.ikev2d -c
# svcadm restart ipsec/ike:ikev2
```

  IKEv1의 경우:

```
# pfedit /etc/inet/ike/config
# /usr/lib/inet/in.iked -c
# svcadm restart ipsec/ike:default
```

  수동 키 관리의 경우:

```
# pfedit -s /etc/inet/secret/ipseckeys
# ipseckey -c /etc/inet/secret/ipseckeys
# svcadm refresh ipsec/manual-key
```

- IPsec 및 IKE의 구성 가능한 등록 정보 수정:

  IPsec 서비스:

```
# svccfg -s ipsec/policy setprop config/property = value
# svcadm refresh ipsec/policy; svcadm restart ipsec/policy
```

  IKEv2 서비스:

```
# svccfg -s ike:ikev2 editprop
# svcadm refresh ipsec/ike:ikev2; svcadm restart ipsec/ike:ikev2
```

IKEv1 서비스:

```
# svccfg -s ipsec/ike setprop config/property = value
# svcadm refresh ipsec/ike:ikev2; svcadm restart ipsec/ike:ikev2
```

수동 키 서비스:

```
# svccfg -s ipsec/manual-key setprop config/property = value
# svcadm refresh ipsec/manual-key; svcadm restart ipsec/manual-key
```

- IKEv2에 대한 미리 공유한 키 구성:

```
# pfedit -s /etc/inet/ike/ikev2.preshared
# /usr/lib/inet/in.ikev2d -c
# svcadm restart ikev2
```

- IKEv1에 대한 미리 공유한 키 구성:

```
# pfedit -s /etc/inet/secret/ike.preshared
# svcadm restart ike
```

실행 중인 IKE 데몬 관리

자세한 내용은 ikeadm(1M) 매뉴얼 페이지를 참조하십시오. 이 섹션의 명령은 IKEv2 또는 IKEv1 데몬이 실행 중인 동안에만 사용할 수 있습니다.

- 실행 중인 IKE 데몬 수정:

  다음 출력에는 데몬의 현재 상태를 수정할 수 있는 ikeadm 명령의 인수가 표시됩니다. 일부 인수는 IKEv2 또는 IKEv1 데몬에만 적용됩니다.

```
% ikeadm help
...
        set   priv level
        set   debug level [filename]
        add   rule|preshared {definition}|filename
        del   p1|ikesa|rule|preshared identifier
        flush p1|ikesa|certcache
        write rule|preshared filename
        token login|logout PKCS#11-Token-Object
```

- ikeadm 명령과 관련된 인수의 구문 표시:

```
% ikeadm help add
This command adds items to in.iked's tables.

Objects that may be set include:
```

```
        rule            a phase 1 or IKE SA policy rule
        preshared       a preshared key

    Objects may be entered on the command-line, as a
    series of keywords and tokens contained in curly
    braces ('{', '}'); or the name of a file containing
    the object definition may be provided.

    For security purposes, preshared keys may only be
    entered on the command-line if ikeadm is running in
    interactive mode.
```

- `ikeadm` 명령을 사용하여 IKEv2 데몬 수정:

```
# ikeadm add rule | preshared {definition} | filename
# ikeadm flush ikesa
# ikeadm del ikesa | rule | preshared identifier
# ikeadm set debug level
# ikeadm token login | logout PKCS#11-Token-Object
# ikeadm write rule | preshared filename
```

- `ikeadm` 명령을 사용하여 IKEv1 데몬 수정:

```
# ikeadm set debug level
# ikeadm set privlevel
# ikeadm add rule | preshared {definition} | filename
# ikeadm del p1 | rule | preshared identifier
# ikeadm flush p1 | certcache
# ikeadm del rule | preshared id
# ikeadm write rule | preshared filename
```

♦ ♦ ♦ 12 장

12 IPsec 및 키 관리 참조

이 장에는 IPsec, IKEv2 및 IKEv1에 대한 참조 정보가 포함되어 있습니다.

네트워크에서 IPsec을 구현하는 방법에 대한 지침은 7장. IPsec 구성을 참조하십시오. IPsec의 개요는 6장. IP Security Architecture 정보를 참조하십시오.

IKE 구현 지침은 9장. IKEv2 구성을 참조하십시오. 개요 정보는 8장. IKE(Internet Key Exchange)를 참조하십시오.

IPsec 참조

IPsec 서비스, 파일 및 명령

이 섹션에는 IPsec 서비스, 선택된 IPsec RFC, IPsec과 관련된 파일 및 명령 등이 나옵니다.

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 값으로 `ipsecinit.conf` 파일의 위치를 결정합니다. DefaultFixed 네트워크 구성 프로파일을 실행 중인 시스템의 초기 값은 `/etc/inet/ipsecinit.conf`입니다. 이 프로파일을 실행하고 있지 않은 시스템에서는 등록 정보 값이 비어 있습니다.
- `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 값으로 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 "IKEv2 서비스" [213] 및 "IKEv1 서비스" [217]를 참조하십시오.

SMF에 대한 자세한 내용은 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오. 또한 smf(5), svcadm(1M) 및 svccfg(1M) 매뉴얼 페이지를 참조하십시오.

ipsecconf 명령

ipsecconf 명령을 사용하여 호스트의 IPsec 정책을 구성합니다. 명령을 실행하여 정책을 구성할 때 시스템은 커널에 IPsec 정책 항목을 만듭니다. 시스템에서는 이러한 항목을 사용하여 모든 인바운드 및 아웃바운드 IP 데이터그램에 대한 정책을 확인합니다. 터널링 및 전달되지 않은 패킷은 이 명령을 사용하여 추가한 정책 확인에 영향을 받지 않습니다. ipsecconf 명령으로 SPD(보안 정책 데이터베이스)의 IPsec 항목도 관리합니다. IPsec 정책 옵션은 ipsecconf(1M) 매뉴얼 페이지를 참조하십시오.

ipsecconf 명령을 호출하려면 root 역할이 있어야 합니다. 이 명령을 통해 양방향에서 트래픽을 보호하는 항목을 구성할 수 있습니다. 한 방향에서만 트래픽을 보호하는 항목도 구성할 수 있습니다.

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 명명된 호스트에 대해 방향을 지정하지 않은 경우 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 "IPsec을 사용하여 네트워크 트래픽 보호" [99]를 참조하십시오.

ipsecinit.conf 구성 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 do not match the above rules to pass up the IP
# network stack. ,,,
```

ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항

설정된 연결에 대한 IPsec 정책은 변경할 수 없습니다. 정책을 변경할 수 없는 소켓을 잠긴 소켓이라고 합니다. 새 정책 항목은 이미 잠긴 소켓을 보호하지 않습니다. 자세한 내용은 connect(3SOCKET) 및 accept(3SOCKET) 매뉴얼 페이지를 참조하십시오. 의심스러운 경우 연결을 다시 시작하십시오. 자세한 내용은 ipsecconf(1M) 매뉴얼 페이지의 SECURITY 섹션을 참조하십시오.

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 없으며 직접 편집하지 않아야 합니다. 그러나 파일을 수정해야 하는 경우 ipsecalgs 명령을 사용합니다. 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. 특히, 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 11.2의 암호화 및 인증서 관리 ”의 1 장, “암호화 프레임워크”를 참조하십시오.

ipseckey 명령

ipseckey 명령에 다양한 옵션을 사용하여 IPsec의 키를 수동으로 관리합니다. ipseckey 명령에 대한 설명은 ipseckey(1M) 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.

참고 - 가능하면 수동 키 입력보다는 IKE를 사용하십시오.

자세한 내용은 ipseckey(1M) 매뉴얼 페이지의 SECURITY 섹션을 참조하십시오.

kstat 명령

kstat 명령으로 ESP, AH 및 기타 IPsec 데이터에 대한 통계를 표시할 수 있습니다. IPsec 관련 옵션은 “IPsec 및 IKE 의미 오류 문제 해결” [197]에 나옵니다. kstat(1M) 매뉴얼 페이지도 참조하십시오.

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 않으므로 AH로 보호되는 트래픽은 snoop 명령을 사용하여 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 IPsec로 패킷이 보호되는지 확인하는 방법 [119]을 참조하십시오.

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

IPsec RFC

IETF(Internet Engineering Task Force)는 IP 계층에 대한 보안 아키텍처를 설명하는 여러 RFC(Requests for Comment)를 게시했습니다. 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 (IKEv1),” 1998년 11월
- RFC 5996, “Internet Key Exchange Protocol Version 2 (IKEv2),” 2010년 9월
- RFC 3554, “On the Use of Stream Control Transmission Protocol (SCTP) with IPsec,” 2003년 7월

IPsec에 대한 보안 연관 데이터베이스

IPsec 보안 서비스에 대한 키 자료 정보는 보안 연관 데이터베이스(SADB)에서 유지 관리됩니다. SA(보안 연관)는 인바운드 패킷 및 아웃바운드 패킷을 보호합니다.

`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 키 관리

IKE(Internet Key Exchange) 프로토콜은 IPsec에 대한 키 관리를 자동으로 처리합니다. `ipseckey` 명령을 사용하여 IPsec SA를 수동으로 관리할 수도 있지만 IKE를 사용하는 것이 좋습니다. 자세한 내용은 "IPsec 보안 연관에 대한 키 관리" [87]를 참조하십시오.

Oracle Solaris의 SMF(서비스 관리 기능) 기능은 IPsec에 대한 다음 키 관리 서비스를 제공합니다.

- `svc:/network/ipsec/ike` 서비스 - 자동 키 관리를 위한 SMF 서비스입니다. `ike` 서비스는 인스턴스가 두 개입니다. `ike:ikev2` 서비스 인스턴스는 `in.ikev2d` 데몬(IKEv2)을 실행하여 자동 키 관리를 제공합니다. `ike:default` 서비스는 `in.iked` 데몬(IKEv1)을 실행합니다. IKE에 대한 설명은 8장. IKE(Internet Key Exchange)를 참조하십시오. 데몬에 대한 자세한 내용은 `in.ikev2d`(1M) 및 `in.iked`(1M) 매뉴얼 페이지를 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 `ipseckey`(1M) 매뉴얼 페이지를 참조하십시오.

IKEv2 참조

IKEv2는 IKEv1을 대체합니다. 비교는 "IKEv2 및 IKEv1 비교" [128]를 참조하십시오.

IKEv2 유틸리티 및 파일

다음 표에서는 IKEv2 정책의 구성 파일, IKEv2 키의 저장소 위치 및 IKEv2를 구현하는 다양한 명령과 서비스를 요약합니다. 서비스에 대한 자세한 내용은 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오.

표 12-1 IKEv2 서비스 이름, 명령, 구성 및 키 저장소 위치, 하드웨어 장치

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
`svc:/network/ipsec/ike:ikev2`	IKEv2를 관리하는 SMF 서비스입니다.	`smf`(5)
`/usr/lib/inet/in.ikev2d`	IKE(Internet Key Exchange) 데몬입니다. `ike:ikev2` 서비스를 사용으로 설정할 때 자동 키 관리를 활성화합니다.	`in.ikev2d`(1M)
`/usr/sbin/ikeadm [-v 2]`	IKEv2 정책을 확인하고 일시적으로 수정하기 위한 IKE 관리 명령입니다. 사용 가능한 Diffie-Hellman 그룹과 같은 IKEv2 관리 객체를 확인할 수 있습니다.	`ikeadm`(1M)
`/usr/sbin/ikev2cert`	구성 소유자 `ikeuser`로 공개 키 인증서를 만들고 저장하는 인증서 데이터베이스 관리 명령입니다. `pktool` 명령을 호출합니다.	`ikev2cert`(1M) `pktool`(1)
`/etc/inet/ike/ikev2.config`	IKEv2 정책의 기본 구성 파일입니다. 인바운드 IKEv2 요청을 일치시키고 아웃바운드 IKEv2 요청을 준비하기 위한 사이트 규칙을 포함합니다. 이 파일이 있으면 `ike:ikev2` 서비스를 사용으로 설정할 때 `in.ikev2d` 데몬이 시작됩니다. `svccfg` 명령을 사용하여 이 파일의 위치를 변경할 수 있습니다.	`ikev2.config`(4)
`/etc/inet/ike/ikev2.preshared`	인증서 기반 인증을 사용하지 않는 두 IKEv2 인스턴스가 각 사용자를 인증하는 데 사용할 수 있는 보안 키를 포함합니다.	`ikev2.preshared`(4)
softtoken 키 저장소	`ikeuser`가 소유하는 IKEv2에 대한 개인 키 및 공개 키를 포함합니다.	`pkcs11_softtoken`(5)

IKEv2 서비스

SMF(서비스 관리 기능)에서는 IKEv2를 관리하는 `svc:/network/ipsec/ike:ikev2` 서비스 인스턴스를 제공합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 이 서비스를 사용으로 설정하려면 먼저 `/etc/inet/ike/ikev2.config` 파일에서 IKEv2 구성을 만들어야 합니다.

다음과 같은 `ike:ikev2` 서비스 등록 정보를 구성할 수 있습니다.

- `config_file` **등록 정보** - IKEv2 구성 파일의 위치를 지정합니다. 초기 값은 `/etc/inet/ike/ikev2.config`입니다. 이 파일은 사용 권한이 특수하며 `ikeuser`가 소유해야 합니다. 다른 파일을 사용하지 마십시오.
- `debug_level` **등록 정보** - `in.ikev2d` 데몬의 디버깅 레벨을 설정합니다. 초기 값은 `op` 또는 operational입니다. 가능한 값은 `ikeadm`(1M) 매뉴얼 페이지에서 객체 유형 아래의 디버그 레벨 테이블을 참조하십시오.
- `debug_logfile` **등록 정보** - IKEv2를 디버깅하기 위한 로그 파일의 위치를 지정합니다. 초기 값은 `/var/log/ikev2/in.ikev2d.log`입니다.
- `kmf_policy` **등록 정보** - 인증서 정책에 대한 로그 파일의 위치를 설정합니다. 기본값은 `/etc/inet/ike/kmf-policy.xml`입니다. 이 파일은 사용 권한이 특수하며 `ikeuser`가 소유해야 합니다. 다른 파일을 사용하지 마십시오.

- `pkcs11_token/pin` **등록 정보** - IKEv2 데몬이 시작될 때 키 저장소에 로그인하는 데 사용할 PIN을 설정합니다. 이 값은 `ikev2cert setpin` 명령에서 토큰에 설정한 값과 일치해야 합니다.
- `pkcs11_token/uri` **등록 정보** - 키 저장소에 대한 PKCS #11 URI를 설정합니다. 암호화 가속기 카드에서 하드웨어 저장소를 사용하려면 이 값을 지정해야 합니다.

SMF에 대한 자세한 내용은 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오. 또한 `smf`(5), `svcadm`(1M) 및 `svccfg`(1M) 매뉴얼 페이지를 참조하십시오.

IKEv2 데몬

`in.ikev2d` 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. IPsec을 사용하여 IKEv2 프로토콜을 통한 통신을 보호하려는 모든 시스템에서 이 데몬이 실행되고 있어야 합니다.

기본적으로 `svc:/network/ipsec/ike:ikev2` 서비스는 사용으로 설정되지 않습니다. `/etc/inet/ike/ikev2.config` 파일을 구성하고 `ike:ikev2` 서비스 인스턴스를 사용으로 설정한 경우 시스템 부트 시 SMF에 의해 `in.ikev2d` 데몬이 시작됩니다.

IKEv2 데몬이 실행되면 시스템은 자신을 피어 IKEv2 엔티티에 인증하고 세션 키를 설정합니다. 구성 파일에 지정된 간격으로 IKE 키가 자동으로 대체됩니다. `in.ikev2d` 데몬은 네트워크에서 들어오는 IKE 요청과 `PF_KEY` 소켓을 통과하는 아웃바운드 트래픽 요청을 수신합니다. 자세한 내용은 `pf_key`(7P) 매뉴얼 페이지를 참조하십시오.

두 가지 명령이 IKEv2 데몬을 지원합니다. `ikeadm` 명령을 사용하여 IKE 정책을 볼 수 있습니다. 자세한 내용은 "IKEv2에 대한 `ikeadm` 명령" [215]을 참조하십시오. `ikev2cert` 명령을 사용하면 공개 및 개인 키 인증서를 보고 관리할 수 있습니다. 자세한 내용은 "IKEv2 `ikev2cert` 명령" [215]을 참조하십시오.

IKEv2 구성 파일

IKEv2 구성 파일 `/etc/inet/ike/ikev2.config`는 IPsec 정책 파일 `/etc/inet/ipsecinit.conf`에서 보호되는 지정된 네트워크 끝점에 대한 키를 협상하는 데 사용되는 규칙을 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 IKEv2 SA가 다시 입력되기 전까지의 기본 시간 `ikesa_lifetime_secs`와 같은 항목이 포함됩니다. IKEv2 구성 파일의 예는 "미리 공유한 키로 IKEv2 구성" [134]을 참조하십시오. IKEv2 정책 항목의 예 및 설명은 `ikev2.config`(4) 매뉴얼 페이지를 참조하십시오.

IKEv2가 지원하는 IPsec SA는 IPsec 구성 파일 /etc/inet/ipsecinit.conf의 정책에 따라 IP 패킷을 보호합니다.

ike/ikev2.config 파일에 대한 보안 고려 사항은 ipsecinit.conf 파일의 고려 사항과 비슷합니다. 세부 정보는 “ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항” [209]을 참조하십시오.

IKEv2에 대한 ikeadm 명령

in.ikev2d 데몬이 실행 중인 경우 ikeadm [-v2] 명령을 사용하여 다음을 수행할 수 있습니다.

- IKEv2 상태의 여러 측면을 봅니다.
- 정책 규칙, 미리 공유한 키, 사용 가능한 Diffie-Hellman 그룹, 암호화 및 인증 알고리즘, 기존의 활성 IKEv2 SA 등과 같은 IKEv2 데몬 객체를 표시합니다.

이 명령의 옵션에 대한 예제 및 전체 설명은 ikeadm(1M) 매뉴얼 페이지를 참조하십시오.

ikeadm 명령에 대한 보안 고려 사항은 ipseckey 명령의 고려 사항과 비슷합니다. 세부 정보는 “ipseckey에 대한 보안 고려 사항” [210]을 참조하십시오.

IKEv2 미리 공유한 키 파일

/etc/inet/ike/ikev2.preshared 파일에는 IKEv2 서비스에서 사용하는 미리 공유한 키가 포함됩니다. 이 파일은 ikeuser가 소유하며 0600에서 보호됩니다.

미리 공유한 키가 필요한 ike/ikev2.config 파일에서 규칙을 구성하는 경우 기본 ikev2.preshared 파일을 사용자 정의해야 합니다. IKEv2에서는 이러한 미리 공유한 키를 사용하여 IKEv2 피어를 인증하므로 이러한 파일이 유효해야만 in.ikev2d 데몬에서 미리 공유한 키가 필요한 규칙을 읽습니다.

IKEv2 ikev2cert 명령

ikev2cert 명령을 사용하여 공개 및 개인 키와 인증서를 생성, 저장 및 관리합니다. ike/ikev2.config 파일에 공개 키 인증서가 필요할 때 이 명령을 사용합니다. IKEv2에서는 이러한 인증서를 사용하여 IKEv2 피어를 인증하므로 인증서가 마련되어 있어야만 in.ikev2d 데몬에서 인증서가 필요한 규칙을 읽습니다.

ikev2cert 명령은 pktool 명령을 ikeuser로 호출합니다.

다음 ikev2cert 명령으로 IKEv2에 대한 인증서를 관리합니다. 이 명령은 ikeuser 계정이 실행해야 합니다. 결과는 PKCS #11 softtoken 키 저장소에 저장됩니다.

- ikev2cert setpin - ikeuser 사용자에 대한 PIN을 생성합니다. 이 PIN은 인증서를 사용할 때 필요합니다.
- ikev2cert gencert - 자체 서명된 인증서를 생성합니다.
- ikev2cert gencsr - CSR(인증서 서명 요청)을 생성합니다.
- ikev2cert list - 키 저장소의 인증서를 나열합니다.
- ikev2cert export - 인증서를 파일로 내보냅니다.
- ikev2cert import - 인증서 또는 CRL을 가져옵니다.

ikev2cert 하위 명령의 구문에 대한 자세한 내용은 pktool(1) 매뉴얼 페이지를 참조하십시오. 예는 ikev2cert(1M) 매뉴얼 페이지를 참조하십시오. softtoken 키 저장소에 대한 내용은 cryptoadm(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 참조

다음 절에서는 IKEv1에 대한 참조 정보를 제공합니다. IKEv1은 더 빠른 자동 키 관리를 제공하는 IKEv2로 대체되었습니다. IKEv2에 대한 자세한 내용은 "IKEv2 참조" [212]를 참조하십시오. 비교는 "IKEv2 및 IKEv1 비교" [128]를 참조하십시오.

IKEv1 유틸리티 및 파일

다음 표에서는 IKEv1 정책의 구성 파일, IKEv1 키의 저장소 위치 및 IKEv1을 구현하는 다양한 명령과 서비스를 요약합니다. 서비스에 대한 자세한 내용은 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오.

표 12-2 IKEv1 서비스 이름, 명령, 구성 및 키 저장소 위치, 하드웨어 장치

서비스, 명령, 파일 또는 장치	설명	매뉴얼 페이지
svc:/network/ipsec/ike:default	IKEv1을 관리하는 SMF 서비스입니다.	smf(5)
/usr/lib/inet/in.iked	Internet Key Exchange(IKEv1) 데몬입니다. ike 서비스를 사용으로 설정할 때 자동화된 키 관리를 활성화합니다.	in.iked(1M)
/usr/sbin/ikeadm [-v1]	IKE 정책을 확인하고 일시적으로 수정하기 위한 IKE 관리 명령입니다. 1단계 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	IKEv1 정책의 기본 구성 파일입니다. 인바운드 IKEv1 요청을 일치시키고 아웃바운드 IKEv1 요청을 준비하기 위한 사이트 규칙을 포함합니다.	ike.config(4)

서비스, 명령, 파일 또는 장치	설명	매뉴얼 페이지
	이 파일이 존재하면 ike 서비스를 사용으로 설정할 때 in.iked 데몬을 시작합니다. svccfg 명령을 사용하여 이 파일의 위치를 변경할 수 있습니다.	
ike.preshared	/etc/inet/secret 디렉토리의 미리 공유한 키 파일입니다. 1단계 교환에서 인증을 위한 보안 키를 포함합니다. 미리 공유한 키로 IKEv1을 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	/etc/inet/ike 디렉토리에서 공개 키 및 인증서 파일이 저장되는 디렉토리입니다. 공개-개인 키 쌍 중 공개 키 부분을 포함합니다.	ikecert(1M)
crls 디렉토리	/etc/inet/ike 디렉토리에서 공개 키 및 인증서 파일에 대한 해지 목록이 저장되는 디렉토리입니다.	ikecert(1M)
Sun Crypto Accelerator 6000 보드	운영 체제에서 작업을 오프로드하여 공개 키 작업의 속도를 늘리는 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)

IKEv1 서비스

SMF(서비스 관리 기능)에서는 IKEv1을 관리하는 svc:/network/ipsec/ike:default 서비스를 제공합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 이 서비스를 사용으로 설정하기 전에 IKEv1 구성 파일 /etc/inet/ike/config를 만들어야 합니다.

다음 ike 서비스 등록 정보를 구성할 수 있습니다.

- config_file **등록 정보** - IKEv1 구성 파일의 위치를 지정합니다. 초기 값은 /etc/inet/ike/config입니다.
- debug_level **등록 정보** - in.iked 데몬의 디버깅 레벨을 설정합니다. 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 레벨을 설정합니다. 초기 값은 base입니다. 다른 값으로 modkeys 및 keymat가 있습니다. 세부 정보는 "IKEv1 ikeadm 명령" [219]을 참조하십시오.

SMF에 대한 자세한 내용은 "Oracle Solaris 11.2의 시스템 서비스 관리 "의 1 장, "서비스 관리 기능 소개"를 참조하십시오. 또한 smf(5), svcadm(1M) 및 svccfg(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 데몬

in.iked 데몬은 IPsec을 사용하는 패킷을 보호하는 암호화 키가 포함된 IPsec SA에 대한 관리를 자동화합니다. 이 데몬은 IKEv1 프로토콜을 실행 중인 피어 시스템과 ISAKMP SA 및 IPsec SA를 안전하게 협상합니다.

기본적으로 svc:/network/ipsec/ike:default 서비스는 사용으로 설정되지 않습니다. /etc/inet/ike/config 파일을 구성하고 ike:default 서비스를 사용으로 설정한 후에는 SMF에서 시스템 부트 시 in.iked 데몬을 시작합니다. /etc/inet/ike/config 파일 외에 추가 구성이 다른 파일 및 데이터베이스에 저장되거나 SMF 등록 정보로 저장됩니다. 자세한 내용은 "IKEv1 유틸리티 및 파일" [216]과 ike.preshared(4), ikecert(1M) 및 in.iked(1M) 매뉴얼 페이지를 참조하십시오.

ike:default 서비스를 사용으로 설정하면 in.iked 데몬은 구성 파일을 읽고 IKE 피어의 외부 요청과 IPsec for SA의 내부 요청을 수신합니다.

IKEv1 피어가 보내는 외부 요청의 경우 ike:default 서비스의 구성에 따라 데몬이 응답하는 방식이 결정됩니다. 내부 요청은 PF_KEY 인터페이스를 통해 라우팅됩니다. 이 인터페이스는 IPsec SA를 저장하고 패킷 암호화 및 해독을 수행하는 IPsec의 커널 부분과 userland에서 실행되는 키 관리 데몬 in.iked 사이의 통신을 처리합니다. 커널에서 패킷을 보호하는 데 SA가 필요한 경우 PF_KEY 인터페이스를 통해 in.iked 데몬에 메시지를 보냅니다. 자세한 내용은 pf_key(7P) 매뉴얼 페이지를 참조하십시오.

두 가지 명령으로 IKEv1 데몬을 지원합니다. ikeadm 명령은 실행 중인 데몬에 명령줄 인터페이스를 제공합니다. ikecert 명령은 디스크 및 하드웨어에서 인증서 데이터베이스 ike.privatekeys 및 publickeys를 관리합니다.

이러한 명령에 대한 자세한 내용은 in.iked(1M), ikeadm(1M) 및 ikecert(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 구성 파일

IKEv1 구성 파일 /etc/inet/ike/config에서는 IPsec 보호가 필요한 네트워크 패킷의 SA를 IPsec 구성 파일 /etc/inet/ipsecinit.conf의 정책에 따라 관리합니다.

IKE의 키 관리에는 규칙 및 전역 매개변수가 관여합니다. IKEv1 규칙은 다른 IKEv1 데몬을 실행 중인 시스템을 식별합니다. 또한 규칙은 인증 방법을 지정합니다. 전역 매개변수에는 연결된 하드웨어 가속기의 경로와 같은 항목이 포함됩니다. IKEv1 정책 파일의 예는 "미리 공유한 키로 IKEv2 구성" [134]을 참조하십시오. IKEv1 정책 항목의 예 및 설명은 ike.config(4) 매뉴얼 페이지를 참조하십시오.

/etc/inet/ike/config 파일에는 RSA Security Inc. PKCS #11 Cryptographic Token Interface(Cryptoki) 표준에 따라 구현된 라이브러리의 경로가 포함될 수 있습니다. IKEv1에서는 이 PKCS #11 라이브러리를 사용하여 키 가속과 키 저장소를 위해 하드웨어에 액세스합니다.

ike/config 파일에 대한 보안 고려 사항은 ipsecinit.conf 파일의 고려 사항과 비슷합니다. 세부 정보는 "ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항" [209]을 참조하십시오.

IKEv1 ikeadm 명령

ikeadm 명령을 사용하여 다음을 수행할 수 있습니다.

- IKE 상태의 여러 측면을 봅니다
- IKE 데몬의 등록 정보를 변경합니다
- 1단계 교환 중 SA 생성에 대한 통계를 표시합니다
- IKE 프로토콜 교환을 디버그합니다
- 모든 1단계 SA, 정책 규칙, 미리 공유한 키, 사용 가능한 Diffie-Hellman 그룹, 1단계 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. 다음과 같은 3가지 권한 레벨이 가능합니다.

base 레벨	키를 보거나 수정할 수 없습니다. base 레벨이 기본 권한 레벨입니다.
keymat 레벨	ikeadm 명령을 사용하여 실제 키를 볼 수 있습니다.
modkeys 레벨	미리 공유한 키를 제거, 변경, 추가할 수 있습니다.

일시적 권한 변경은 ikeadm 명령을 사용할 수 있습니다. 영구적 변경은 ike 서비스의 admin_privilege 등록 정보를 변경합니다. 임시 권한 변경에 대해서는 “실행 중인 IKE 데몬 관리” [205]를 참조하십시오.

ikeadm 명령에 대한 보안 고려 사항은 ipseckey 명령의 고려 사항과 비슷합니다. “ipseckey에 대한 보안 고려 사항” [210]을 참조하십시오. ikeadm 명령과 관련된 세부 사항은 ikeadm(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 미리 공유한 키 파일

미리 공유한 키를 수동으로 만들면 키가 /etc/inet/secret 디렉토리의 파일에 저장됩니다. ike.preshared 파일에는 미리 공유한 키를 사용하도록 ike/config의 규칙을 구성하는 경우 1단계 교환에 대한 미리 공유한 키가 포함됩니다. ipseckeys 파일에는 IP 패킷을 보호하는 데 사용되는 미리 공유한 키가 포함됩니다. 파일은 0600에서 보호됩니다. secret 디렉토리는 0700에서 보호됩니다.

미리 공유한 키를 사용하여 1단계 교환을 인증하므로 in.iked 데몬을 시작하기 전에 파일이 유효해야 합니다.

IPsec 키를 수동으로 관리하는 예는 IPsec 키를 수동으로 만드는 방법 [114]을 참조하십시오.

IKEv1 공개 키 데이터베이스 및 명령

ikecert 명령으로는 로컬 시스템의 공개/개인 키, 공개 인증서 및 정적 CRL 데이터베이스를 관리합니다. IKEv1 구성 파일에 공개 키 인증서가 필요한 경우 이 명령을 사용합니다. IKEv1은 이러한 데이터베이스를 사용하여 1단계 교환을 인증하므로 데이터베이스를 채워야만 in.iked 데몬을 활성화할 수 있습니다. 세 가지 하위 명령 certlocal, certdb, certrldb로 각각 세 데이터베이스를 처리합니다.

시스템에 Sun Crypto Accelerator 6000 보드가 연결된 경우 ikecert 명령에서는 PKCS #11 라이브러리를 사용하여 하드웨어 키 및 인증서 저장소에 액세스합니다.

자세한 내용은 ikecert(1M) 매뉴얼 페이지를 참조하십시오. metaslot 및 softtoken 키 저장소에 대한 내용은 cryptoadm(1M) 매뉴얼 페이지를 참조하십시오.

IKEv1 ikecert tokens 명령

tokens 인수를 지정하면 사용 가능한 토큰 ID가 나열됩니다. ikecert certlocal 및 ikecert certdb 명령에 토큰 ID를 사용하여 공개 키 인증서 및 인증서 요청을 생성할 수 있습니다. 이 키와 인증서도 연결된 Sun Crypto Accelerator 6000 보드에 저장될 수 있습니다. ikecert 명령에서는 PKCS #11 라이브러리를 사용하여 하드웨어 키 저장소에 액세스합니다.

IKEv1 ikecert certlocal 명령

certlocal 하위 명령으로는 개인 키 데이터베이스를 관리합니다. 이 하위 명령의 옵션을 사용하여 개인 키를 추가, 보기, 제거할 수 있습니다. 또한 이 하위 명령은 자체 서명된 인증서 또는 CSR을 만듭니다. -ks 옵션은 자체 서명된 인증서를 만듭니다. -kc 옵션은 CSR을 만듭니다. 키는 시스템의 /etc/inet/secret/ike.privatekeys 디렉토리에 저장되거나, -T 옵션을 사용하여 연결된 하드웨어에 저장됩니다.

개인 키를 만들 때는 ikecert certlocal 명령의 옵션과 관련된 항목이 ike/config 파일에 있어야 합니다. ikecert 옵션과 대응하는 ike/config 항목은 다음 표에 나와 있습니다.

표 12-3 IKEv1에서 ikecert 옵션과 ike/config 항목 사이의 관계

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 | dsa-sha256	auth_method dsa_sig	RSA보다 약간 느린 인증 방법입니다.
-t rsa-md5 및 -t rsa-sha1 | rsa-sha256 | rsa-sha384 | rsa-sha512	auth_method rsa_sig	DSA보다 약간 빠른 인증 방법입니다.

ikecert 옵션	ike/config 항목	설명
		RSA 공개 키는 가장 큰 payload(페이로드)를 암호화할 만큼 충분히 커야 합니다. 일반적으로 X.509 식별 이름과 같은 신원 페이로드가 가장 큰 페이로드입니다.
-t rsa-md5 및 -t rsa-sha1 | rsa-sha256 | rsa-sha384 | rsa-sha512	auth_method rsa_encrypt	RSA 암호화에서는 도청자가 찾을 수 없도록 IKE의 신원을 숨기지만 이 암호화를 위해서는 IKE 피어가 서로의 공개 키를 알아야 합니다.

ikecert certlocal -kc 명령으로 CSR을 실행하는 경우 명령의 출력을 CA(인증 기관)에 보냅니다. 회사에서 자체의 PKI(공개 키 기반구조)를 실행하는 경우에는 출력을 PKI 관리자에게 보냅니다. 그런 다음 CA 또는 PKI 관리자가 인증서를 만듭니다. 반환되는 인증서는 certdb 하위 명령의 입력입니다. CA가 반환하는 CRL(인증서 해지 목록)은 certrldb 하위 명령의 입력입니다.

IKEv1 ikecert certdb 명령

certdb 하위 명령으로는 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 자체 서명된 공개 키 인증서로 IKEv1을 구성하는 방법 [165]을 참조하십시오. 또한 이 명령은 CA에서 받은 인증서를 입력으로 받아들입니다. 절차는 CA가 서명한 인증서로 IKEv1을 구성하는 방법 [170]을 참조하십시오.

인증서 및 공개 키는 시스템의 /etc/inet/ike/publickeys 디렉토리에 저장됩니다. -T 옵션은 연결된 하드웨어에 인증서, 개인 키, 공개 키를 저장합니다.

IKEv1 ikecert certrldb 명령

certrldb 하위 명령으로는 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. CA에서 CRL을 제공하는 경우 ikecert certrldb 명령을 사용하여 CRL 데이터베이스에 CRL을 설치할 수 있습니다. 절차는 IKEv1에서 해지된 인증서를 처리하는 방법 [178]을 참조하십시오.

IKEv1 /etc/inet/ike/publickeys 디렉토리

/etc/inet/ike/publickeys 디렉토리에는 공개-개인 키 쌍의 공개 부분과 해당 인증서가 파일이나 슬롯으로 포함됩니다. 디렉토리는 0755에서 보호됩니다. ikecert certdb 명령은 디렉토리를 채웁니다. -T 옵션은 publickeys 디렉토리가 아닌 Sun Crypto Accelerator 6000 보드에 키를 저장합니다.

슬롯에는 다른 시스템에서 생성된 인증서의 X.509 고유 이름이 인코딩된 형식으로 포함됩니다. 자체 서명된 인증서를 사용하는 경우 원격 시스템의 관리자로부터 받은 인증서를 명령

의 입력으로 사용합니다. CA의 인증서를 사용하는 경우 CA에서 서명한 두 인증서를 이 데이터베이스로 설치합니다. CA에 보낸 CSR에 준하는 인증서를 설치합니다. 또한 CA의 인증서를 설치합니다.

IKEv1 /etc/inet/secret/ike.privatekeys 디렉토리

/etc/inet/secret/ike.privatekeys 디렉토리에는 공개-개인 키 쌍의 일부인 개인 키 파일이 저장됩니다. 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.

IKEv1 /etc/inet/ike/crls 디렉토리

/etc/inet/ike/crls 디렉토리에는 CRL(인증서 해지 목록) 파일이 포함됩니다. 각 파일은 /etc/inet/ike/publickeys 디렉토리에 있는 공개 인증서 파일에 해당합니다. CA에서는 자신이 발행한 인증서에 대한 CRL을 제공합니다. ikecert certrldb 명령을 사용하여 데이터베이스를 채울 수 있습니다.

네트워크 보안 용어 해설

3DES Triple-DES를 참조하십시오.

가상 LAN(VLAN) 장치 이더넷(datalink) 레벨의 IP 프로토콜 스택에서 트래픽 전달을 제공하는 네트워크 인터페이스입니다.

라우터 간청 호스트가 다음 일정이 잡힌 시간이 아닌, 즉시 라우터 알림을 생성하도록 라우터에 요청하는 프로세스입니다.

라우터 검색 호스트가 연결된 링크에 상주하는 라우터를 찾는 프로세스입니다.

AES Advanced Encryption Standard의 머리글자어로, 고급 암호화 표준입니다. 대칭 블록 데이터 암호화 기술입니다. 미국 정부는 2000년 10월 알고리즘의 Rijndael 변형을 암호화 표준으로 채택했습니다. AES가 정부 표준으로 DES 암호화를 대체합니다.

asymmetric key cryptography (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 algorithm (Diffie-Hellman 알고리즘)은 비대칭 키 프로토콜의 예입니다. symmetric key cryptography (대칭 키 암호화)와 대조됩니다.

authentication header(인증 헤더) IP 패킷에 기밀성 없이 인증 및 무결성을 제공하는 확장 헤더입니다.

bidirectional tunnel(양방향 터널) 패킷을 양방향으로 전송할 수 있는 터널입니다.

Blowfish 32-448비트의 가변 길이 키를 사용하는 대칭 블록 암호화 알고리즘입니다. 저작자인 Bruce Schneier에 따르면, Blowfish는 키를 자주 바꾸지 않는 응용 프로그램에 최적화되어 있습니다.

broadcast address(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.

certificate authority(CA, 인증 기관) 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.

chain of trust(트러스트 체인) X.509 인증서에서 인증 기관이 trust anchor(트러스트 앵커)에서 사용자의 인증서에 이르는 인증서가 끊어지지 않은 인증 체인을 제공한다는 보증입니다.

CRL(인증서 해지 목록) CA에 의해 해지된 공개 키 인증서 목록입니다. CRL은 IKE를 통해 유지 관리하는 CRL 데이터베이스에 저장됩니다.

DES Data Encryption Standard의 머리글자어로, 데이터 암호화 표준입니다. 1975년에 개발되고 1981년에 ANSI에 의해 ANSI X.3.92로 표준화된 대칭 키 암호화 방법입니다. DES에서는 56비트 키를 사용합니다.

Diffie-Hellman algorithm (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.

digital signature(디지털 서명) 발신자를 고유하게 식별하는, 전자적으로 전송된 메시지에 첨부된 디지털 코드입니다.

distinguished name(DN, 고유 이름) 일반 문자열을 사용하여 공유 정보를 나타내는 표준화된 방법입니다. 고유 이름은 LDAP 및 X.509 인증서뿐 아니라 다른 기술에서도 사용됩니다. 자세한 내용은 A String Representation of Distinguished Names (http://www.ietf.org/rfc/rfc1779.txt)를 참조하십시오.

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.

DSA Digital Signature Algorithm의 머리글자어로, 디지털 서명 알고리즘입니다. 512-4096비트의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 1024비트까지 지원합니다. DSA는 입력에 SHA-1을 사용합니다.

dynamic packet filter(동적 패킷 필터) stateful packet filter(stateful 패킷 필터)를 참조하십시오.

ECDSA Elliptic Curve Digital Signature Algorithm의 머리글자어로, 타원 곡선 디지털 서명 알고리즘입니다. 타원 곡선 수학을 기반으로 하는 공개 키 알고리즘입니다. ECDSA 키 크기는 동일한 길이의 서명을 생성하는 데 필요한 DSA 공개 키의 크기보다 많이 작습니다.

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

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

firewall(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.

hash value(해시 값) 텍스트의 문자열에서 생성된 숫자입니다. 해시 함수를 사용하여 전송된 메시지가 변조되지 않았는지 확인할 수 있습니다. MD5 및 SHA-1은 단방향 해시 함수의 예입니다.

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

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

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

IP Internet Protocol(IP, 인터넷 프로토콜), IPv4, IPv6을 참조하십시오.

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

IP link(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 packet(IP 패킷) IP를 통해 전달된 정보의 패킷입니다. IP 패킷에는 헤더와 데이터가 포함됩니다. 헤더는 패킷의 소스 및 대상 주소를 포함합니다. 헤더의 다른 필드는 대상에서 데이터를 식별하고 함께 표시되는 패킷과 재결합하는 데 도움이 됩니다.

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

IPsec IP 보안입니다. IP 패킷을 보호하는 보안 아키텍처입니다.

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.

key management(키 관리) 보안 연관(SA)을 관리하는 방법입니다.

keystore name(키 저장소 이름) NIC(네트워크 인터페이스 카드)의 저장소 영역 또는 키 저장소에 관리자가 부여하는 이름입니다. 키 저장소 이름을 토큰 또는 토큰 ID라고도 합니다.

label(레이블) 1. IKEv2 규칙의 키워드로, `auth_method`가 `preshared`인 경우 이 키워드 값은 미리 공유한 키 파일의 `label` 키워드 값과 일치해야 합니다.

2. IKEv2 인증서를 만들 때 사용되는 키워드입니다. 이 값은 키 저장소에서 인증서의 모든 부분(개인 키, 공개 키 및 공개 키 인증서)을 찾을 때 편리합니다.

3. 객체 또는 프로세스의 민감도 수준에 대한 MAC(필수 액세스 제어) 표시입니다. Confidential 및 Top Secret가 샘플 레이블입니다. 레이블 지정 네트워크 전송에는 MAC 레이블이 포함됩니다.

4. IKEv1 규칙의 키워드로, 해당 값은 규칙을 가져오는 데 사용됩니다.

link layer(링크 계층) IPv4/IPv6 바로 아래의 계층입니다.

link-local address(링크 로컬 주소) IPv6에서 자동 주소 구성과 같은 목적으로 단일 링크에 주소 배정을 위해 사용되는 지정입니다. 기본적으로 link-local 주소는 시스템의 MAC 주소에서 생성됩니다.

MAC(메시지 인증 코드) MAC는 데이터 무결성을 보증하고 데이터 발신을 인증합니다. MAC는 도청에 대해 보호되지 않습니다.

marker(표시자) 1. 패킷의 전달 방법을 나타내는 값으로 IP 패킷의 DS 필드를 표시하는 diffserv 구조 및 IPQoS의 모듈입니다. IPQoS 구현에서 표시자 모듈은 `dscpmk`입니다.

2. 사용자 우선 순위 값으로 이더넷 패킷의 가상 LAN 태그를 표시하는 IPQoS 구현의 한 모듈입니다. 사용자 우선 순위 값은 VLAN 장치가 있는 네트워크에서 패킷을 전달하는 방식을 나타냅니다. 이 모듈을 `dlcosmk`라고 합니다.

MD5 디지털 서명을 포함하여 메시지 인증용으로 사용되는 반복적인 암호화 해시 함수입니다. 이 기능은 1991년 Rivest가 개발했습니다.

multicast address(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.

multihomed host(멀티홈 호스트) 패킷 전달을 수행하지 않는 여러 개의 물리적 인터페이스가 있는 시스템입니다. 멀티홈 호스트는 경로 지정 프로토콜을 실행할 수 있습니다.

NAT network address translation (NAT, 네트워크 주소 변환)를 참조하십시오.

network address translation (NAT, 네트워크 주소 변환) 한 네트워크 내에 사용된 IP 주소를 다른 네트워크 내에 알려진 다른 IP 주소로 변환합니다. 필요한 전역 IP 주소 수를 제한하는 데 사용됩니다.

NIC(네트워크 인터페이스 카드) 네트워크에 인터페이스로 연결된 네트워크 어댑터 카드입니다. 일부 NIC는 `igb` 카드와 같은 여러 물리적 인터페이스를 가질 수 있습니다.

packet filter(패킷 필터) 방화벽을 통해 지정된 패킷을 허용하도록 구성하거나 허용하지 않도록 구성할 수 있는 방화벽 기능입니다.

packet header(패킷 헤더) IP header(IP 헤더)를 참조하십시오.

packet(패킷) IP packet(IP 패킷)을 참조하십시오.

packet(패킷) 통신 회선을 통해 한 단위로 전송되는 정보 그룹입니다. IP header(IP 헤더)와 payload(페이로드)를 포함합니다.

payload(페이로드) 패킷에 전달된 데이터입니다. 페이로드는 패킷을 대상으로 가져오는 데 필요한 헤더 정보를 포함하지 않습니다.

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. 따라서 PFS는 이전에 기록된 트래픽의 해독을 방지할 수 있습니다.

PFS는 인증된 키 교환에만 적용됩니다. Diffie-Hellman algorithm (Diffie-Hellman 알고리즘)도 참조하십시오.

physical interface(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.

PKI Public Key Infrastructure의 머리글자어로, 공개 키 기반구조입니다. 인터넷 트랜잭션에 관여한 해당자의 유효성을 확인 및 인증하는 디지털 인증서, 인증 기관 및 기타 등록 기관의 시스템제입니다.

proxy server(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.

public key cryptography 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.

(공개 키 암호화)

replay attack(재생 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.

router advertisement (라우터 알림) 정기적으로 또는 라우터 간청 메시지의 응답으로, 라우터가 다양한 링크 및 인터넷 매개변수를 함께 사용하여 자신의 존재를 알리는 프로세스입니다.

router(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.

RSA 디지털 서명 및 공개 키 암호화 체계를 얻기 위한 방법입니다. 1978년에 개발자 Rivest, Shamir, Adleman이 처음 기술했습니다.

SADB Security Associations Database의 머리글자어로, 보안 연관 데이터베이스입니다. 암호화 키 및 암호화 알고리즘을 지정하는 테이블입니다. 키 및 알고리즘은 보안 데이터 전송에 사용됩니다.

security association(SA, 보안 연관) 한 호스트에서 두번째 호스트로 보안 등록 정보를 지정하는 연관입니다.

SHA-1 Secure Hashing Algorithm의 머리글자어로, 보안 해시 알고리즘입니다. 이 알고리즘은 2^{64} 미만의 입력 길이에서 작동하여 메시지 다이제스트를 생성합니다. SHA-1 알고리즘은 DSA로 입력됩니다.

smurf attack(스머프 공격) 원격 위치에서 IP broadcast address(브로드캐스트 주소) 또는 다중 브로드캐스트 주소로 지정된 ICMP 에코 요청 패킷을 사용하여 심각한 네트워크 혼잡 또는 정전을 일으킵니다.

sniff(스니프) 컴퓨터 네트워크에서 도청하는 것입니다. 일반 텍스트 암호, 유선 끄기와 같은 정보를 조사하기 위해 자동화된 프로그램의 일부로 자주 사용됩니다.

SPD(보안 정책 데이터베이스) 패킷에 적용할 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷을 폐기할지, 일반 텍스트로 전달할지, IPsec로 보호할지 결정합니다.

SPI(보안 매개변수 색인) 수신자가 받은 패킷을 해독하기 위해 사용할 보안 연관 데이터베이스(SADB)의 행을 지정하는 정수입니다.

spoof(스푸핑) 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.

stateful packet filter(stateful 패킷 필터)
활성 연결의 상태를 모니터하여 얻은 정보를 바탕으로 네트워크 패킷이 packet filter(패킷 필터)를 통과할지 여부를 확인할 수 있는 firewall(방화벽)입니다. Stateful 패킷 필터는 요청과 응답을 추적하고 일치시켜 요청과 일치하지 않는 응답을 걸러낼 수 있습니다.

Stream Control Transport Protocol(SCTP)
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.

symmetric key cryptography(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. AES는 대칭 키의 한 예입니다.

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.

Triple-DES
Triple-Data Encryption Standard입니다. 대칭 키 암호화 방법입니다. 3중 DES는 168비트의 키 길이가 필요합니다. 3중 DES를 3DES로 쓰기도 합니다.

trust anchor(트러스트 앵커)
X.509 인증서에서 인증 기관의 루트 인증서입니다. 루트 인증서에서 최종 인증서까지의 인증서로 트러스트 체인이 설정됩니다.

tunnel(터널)
캡슐화되는 동안 뒤에 packet(패킷)이 오는 경로입니다. encapsulation (캡슐화)을 참조하십시오.

IPsec에서 구성되는 터널은 지점 간 인터페이스입니다. 터널을 통해 한 IP 패킷을 다른 IP 패킷 내부에 캡슐화할 수 있습니다.

virtual network interface(VNIC, 가상 네트워크 인터페이스)
물리적 네트워크 인터페이스에 구성되었는지 여부에 관계없이 가상 네트워크 연결을 제공하는 의사 인터페이스입니다. 배타적 IP 영역과 같은 컨테이너에서 위의 VNIC이 가상 네트워크를 형성하도록 구성됩니다.

virtual network(가상 네트워크)
소프트웨어 및 하드웨어 네트워크 리소스 및 기능의 조합으로, 단일 소프트웨어 엔티티로 함께 관리됩니다. 내부 가상 네트워크는 네트워크 리소스를 단일 시스템에 통합하며 "시스템 내 네트워크"라고도 합니다.

VPN(가상 사설망)
인터넷과 같은 공중망에서 터널을 사용하는 단일의 안전한 논리적 네트워크입니다.

색인

ㅎ