在 Oracle® Solaris 11.2 中确保网络安全

文件号码 E53813-02
2014 年 9 月

ORACLE

目录

表

示例

使用本文档

- **概述** – 介绍如何提供网络安全性。包括 Web 服务器的链路保护、可调网络参数、防火墙保护、IPsec 和 IKE 以及 SSL 内核保护。
- **目标读者** – 网络安全管理员。
- **必备知识** – 站点安全要求。

产品文档库

位于 http://www.oracle.com/pls/topic/lookup?ctx=E56344 的文档库中包含此产品的最新信息和已知问题。

获得 Oracle 支持

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info；如果您听力受损，请访问 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs。

反馈

可以在 http://www.oracle.com/goto/docfeedback 上提供有关本文档的反馈。

♦ ♦ ♦ 第 1 章

1 在虚拟化环境中使用链路保护

本章介绍链路保护以及如何在 Oracle Solaris 系统上对其进行配置。本章涵盖以下主题：

Oracle Solaris 11.2 中新增的网络安全功能

本节向现有客户重点介绍此发行版中新增的重要网络安全功能。

IKE 版本 2 (IKEv2) 使用 IKE 协议的最新版本为 IPsec 提供自动密钥管理。IKEv2 和 IPsec 都使用 Oracle Solaris 的加密框架功能中的加密算法。

注 - Oracle Solaris 的加密框架功能通过了第 1 级 FIPS 140-2 验证。有关 IKE 如何使用 FIPS 140 模式，请参见表 8-1 “Oracle Solaris 中的 IKEv2 和 IKEv1 实现”。有关硬件和软件的详细信息，请参见 Oracle FIPS 140 Software Validations (http://www.oracle.com/technetwork/topics/security/fips140-software-validations-1703049.html)（Oracle FIPS 140 软件验证）。

IKE 版本 1 (IKEv1) 支持依然可用。有关更多信息，请参见第 8 章 关于 Internet 密钥交换。

关于链路保护

随着在系统配置中越来越多地采用虚拟化，主机管理员可能对来宾虚拟机 (virtual machine, VM) 授予对某物理或虚拟链路的独占访问权限。这种配置可以将虚拟环境的网络通信流量与主机系统接收或发送的更广泛的通信流量隔离开来，从而提高网络性能。

同时，这种配置会使系统和整个网络暴露于来宾环境可能生成的有害包，带来一定的风险。

链路保护旨在防止潜在的恶意来宾 VM 可能对网络造成损害。该功能提供了针对以下基本威胁的保护：

- IP、DHCP 和 MAC 欺骗
- L2 帧欺骗，例如网桥协议数据单元 (Bridge Protocol Data Unit, BPDU) 攻击

注 - 链路保护不能取代防火墙部署，特别是对于具有复杂的过滤要求的配置。

链路保护类型

Oracle Solaris 中的链路保护机制提供以下保护类型：

`mac-nospoof`

启用针对系统 MAC 地址欺骗的保护。如果链路属于某个区域，启用 `mac-nospoof` 将防止该区域的所有者修改该链路的 MAC 地址。

`ip-nospoof`

启用针对 IP 欺骗的保护。缺省情况下，允许包含 DHCP 地址和链路本地 IPv6 地址的传出包。

您可以使用 `allowed-ips` 链路属性添加地址。对于 IP 地址，包的源地址必须匹配 `allowed-ips` 列表中的地址。对于 ARP 包，包的发送方协议地址必须位于 `allowed-ips` 列表中。

`dhcp-nospoof`

启用针对 DHCP 客户机欺骗的保护。缺省情况下，允许其 ID 与系统的 MAC 地址相匹配的 DHCP 包。

您可以使用 `allowed-dhcp-cids` 链路属性添加允许的客户机。必须按 `dhcpagent(1M)` 手册页中指定的方式设置 `allowed-dhcp-cids` 列表项的格式。

`restricted`

将传出包限制为 IPv4、IPv6 和 ARP。这种保护类型的目的是阻止链路生成可能有害的 L2 控制帧。

注 - 针对以下四种保护类型的内核统计数据会跟踪由于链路保护而被丢弃的包：`mac_spoofed`、`dhcp_spoofed`、`ip_spoofed` 和 `restricted`。要检索这些基于链路的统计数据，请参见如何查看链路保护配置和统计信息 [20]。

有关这些保护类型的更完整的说明，请参见 `dladm(1M)` 手册页。

配置链路保护

要使用链路保护，请设置链路的 `protection` 属性。如果保护类型与其他配置文件（例如 `ip-nospoof` 和 `allowed-ips` 或 `dhcp-nospoof` 和 `allowed-dhcp-cids`）一起使用，则您要执行两个常规操作。首先，启用链路保护。然后，定制配置文件来识别允许传递的其他包。

注 - 您必须在全局区域中配置链路保护。

以下任务列表列出了在 Oracle Solaris 系统上配置链路保护的过程。

表 1-1 配置链路保护任务列表

任务	说明	参考
启用链路保护。	限制链路发送的包并保护链路不受欺骗。	如何启用链路保护 [17]
禁用链路保护。	删除链路保护。	如何禁用链路保护 [18]
指定 IP 链路保护类型。	指定可以通过链路保护机制的 IP 地址。	如何指定 IP 地址以防止受到 IP 欺骗 [18]
指定 DHCP 链路保护类型。	指定可以通过链路保护机制的 DHCP 地址。	如何指定 DHCP 客户机以防止受到 DHCP 欺骗 [19]
查看链路保护配置。	列出受保护的链路和例外情况，并显示执行统计数据。	如何查看链路保护配置和统计信息 [20]

▼ 如何启用链路保护

此过程可限制传出包类型并阻止链路欺骗。

开始之前 您必须成为分配有 "Network Link Security"（网络链路安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **查看可用链路保护类型。**

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

有关可能的类型的说明，请参见“链路保护类型” [16] 和 `dladm`(1M) 手册页。

2. **通过指定一种或多种保护类型启用链路保护。**

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

在以下示例中，系统将在 vnic0 链路上启用所有四种链路保护类型：

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

注意 - 在启用前单独测试每个保护值。错误配置的系统会妨碍通信。

3. 验证是否已启用链路保护。

```
# 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
```

▼ 如何禁用链路保护

此过程将链路保护重置为缺省值，而无链路保护。

开始之前 您必须成为分配有 "Network Link Security"（网络链路安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 通过将 **protection** 属性重置为其缺省值禁用链路保护。

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

2. 验证是否已禁用链路保护。

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

▼ 如何指定 IP 地址以防止受到 IP 欺骗

开始之前 已启用 ip-nospoof 保护类型，如如何启用链路保护 [17]中所示。

您必须成为分配有 "Network Link Security"（网络链路安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 验证是否已启用针对 IP 欺骗的保护。

```
# 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. 将 IP 地址添加到 **allowed-ips** 链路属性的缺省值列表。

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

以下示例说明如何将 IP 地址 10.0.0.1 和 10.0.0.2 添加到 vnic0 链路的 allowed-ips 属性：

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

有关更多信息，请参见 dladm(1M) 手册页。

▼ 如何指定 DHCP 客户机以防止受到 DHCP 欺骗

开始之前 已启用 dhcp-nospoof 保护类型，如如何启用链路保护 [17]中所示。

您必须成为分配有 "Network Link Security"（网络链路安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 验证是否已启用针对 DHCP 欺骗的保护。

```
# 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. 为 **allowed-dhcp-cids** 链路属性指定 ASCII 短语。

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

以下示例说明如何指定字符串 hello 作为 vnic0 链路的 allowed-dhcp-cids 属性值：

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

有关更多信息，请参见 dladm(1M) 手册页。

▼ 如何查看链路保护配置和统计信息

开始之前 您必须成为分配有 "Network Link Security"（网络链路安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 查看链路保护属性值。

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

以下示例显示了 vnic0 链路的 protection、allowed-ips 和 allowed-dhcp-cids 属性的值。

```
# 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       --      --
```

注 - 仅当 ip-nospoof 已启用，列在 EFFECTIVE 下时，才使用 allowed-ips 属性。仅当 dhcp-nospoof 已启用时，才使用 allowed-dhcp-cids 属性。

2. 查看链路保护统计信息。

已提交 dlstat 命令输出，因此该命令适用于脚本。

```
# 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
...
```

该输出表明没有受欺骗或受限制的包已尝试通过。

可使用 kstat 命令，但其输出尚未提交。例如，以下命令可找到 dhcpspoofed 统计数据：

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

有关更多信息，请参见 dlstat(1M) 和 kstat(1M) 手册页。

♦ ♦ ♦ 第 2 章

2 调优网络

本章说明如何调优影响 Oracle Solaris 安全的网络参数。

调优网络

表 2-1 调优网络任务列表

任务	说明	参考
禁用网络路由选择守护进程。	限制可能存在的网络探查器访问系统。	如何禁用网络路由选择守护进程 [23]
防止散播有关网络拓扑的信息。	防止广播包。	如何禁用广播包转发 [24]
	阻止对广播回显请求和多播回显请求的响应。	如何禁用回显请求的响应 [25]
对于充当其他域的网关的系统（例如防火墙或 VPN 节点），打开严格的源和目标多宿主。	阻止其标头中没有网关地址的包在网关外移动。	如何设置严格多宿主 [26]
通过控制不完整系统连接的数量阻止 DOS 攻击。	限制 TCP 侦听器所允许的不完整 TCP 连接数。	如何设置不完整 TCP 连接的最大数目 [26]
通过控制允许的传入连接数阻止 DOS 攻击。	指定 TCP 侦听器的缺省最大暂挂 TCP 连接数。	如何设置暂挂 TCP 连接的最大数目 [27]
验证是否为初始 TCP 连接生成强随机数。	符合 RFC 6528 指定的序列号生成值。	如何为初始 TCP 连接指定强随机数 [27]
防止 ICMP 重定向。	删除网络拓扑的指示符。	如何禁止 ICMP 重定向 [28]
将网络参数恢复为安全的缺省值。	提高因管理操作而降低的安全性。	如何将网络参数重置为安全值 [29]

▼ 如何禁用网络路由选择守护进程

使用此过程可在安装后阻止网络路由，方法是指定缺省路由器。否则，请在手动配置路由后执行此过程。

注 - 许多网络配置过程都要求禁用路由选择守护进程。因此，您可能已在某个大型配置过程中禁用此守护进程。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 检验路由选择守护进程是否正在运行。

```
$ 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.
```

如果服务未运行，则可在此处停止。

2. 禁用路由选择守护进程。

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

3. 检验路由选择守护进程是否已被禁用。

```
$ 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.
```

另请参见 routeadm(1M) 手册页

▼ 如何禁用广播包转发

缺省情况下，Oracle Solaris 将转发广播包。如果您的站点安全策略要求您降低广播泛洪的可能性，请使用此过程更改缺省设置。

注 - 在禁用 _forward_directed_broadcasts 网络属性时，将禁用广播 ping。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 将 IP 包的广播包转发属性设置为 0。

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

2. 检验当前值。

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

另请参见 ipadm(1M) 手册页

▼ 如何禁用回显请求的响应

使用此过程可防止散播有关网络拓扑的信息。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 将 IP 包对广播回显请求的响应属性设置为 0，然后检验当前值。

```
# 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. 将 IP 包对多播回显请求的响应属性设置为 0，然后检验当前值。

```
# 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
```

另请参见 有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“_respond_to_echo_broadcast 和 _respond_to_echo_multicast（ipv4 或 ipv6）”和 ipadm(1M) 手册页。

▼ 如何设置严格多宿主

对于充当其他域的网关的系统（例如防火墙或 VPN 节点），使用此过程可打开严格多宿主。`hostmodel` 属性可控制 IP 包在多宿主系统上的发送和接收行为。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **将 IP 包的 `hostmodel` 属性设置为 `strong`。**

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

2. **检验当前值并注意可能的值。**

```
# 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
```

另请参见 有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“hostmodel（ipv4 或 ipv6）”和 `ipadm`(1M) 手册页。

有关严格多宿主使用情况的更多信息，请参见如何在隧道模式下使用 IPsec 保护两个 LAN 之间的连接 [105]。

▼ 如何设置不完整 TCP 连接的最大数目

使用此过程可通过控制不完整的暂挂连接数阻止拒绝服务 (denial of service, DOS) 攻击。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **设置最大传入连接数。**

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

2. **检验当前值。**

```
# 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
```

另请参见 有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“_conn_req_max_q0”和 ipadm(1M) 手册页。

▼ 如何设置暂挂 TCP 连接的最大数目

使用此过程可通过控制允许的传入连接数阻止 DOS 攻击。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 设置最大传入连接数。

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

2. 检验当前值。

```
# 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
```

另请参见 有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“_conn_req_max_q”和 ipadm(1M) 手册页。

▼ 如何为初始 TCP 连接指定强随机数

以下过程可确保 TCP 初始序列号生成参数符合 RFC 6528 (http://www.ietf.org/rfc/rfc6528.txt) 标准。

开始之前 您必须是指定有 solaris.admin.edit/etc.default/inetinit 授权的管理员。缺省情况下，root 角色拥有此授权。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 验证 **TCP_STRONG_ISS** 变量的缺省值是否为 **2**。

```
# 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. 如果 **TCP_STRONG_ISS** 的值不是 **2**，请将其更改为 **2**。

```
# pfedit /etc/default/inetinit
```

```
TCP_STRONG_ISS=2
```

3. 重新引导系统。

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

▼ 如何禁止 ICMP 重定向

路由器使用 ICMP 重定向消息通知主机更多指向目标的直接路由。非法的 ICMP 重定向消息可能导致 "man-in-the-middle"（中间人）攻击。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 将 IP 包的忽略重定向属性设置为 1，然后检验当前值。

   ICMP 重定向消息可修改主机的路由表且未通过验证。此外，重定向包的处理可增加系统 CPU 需求。

```
# 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. 防止发送 ICMP 重定向消息。

   这些消息包括可显示网络拓扑的一部分的路由表信息。

```
# 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
```

   有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“send_redirects（ipv4 或 ipv6）”和 `ipadm`(1M) 手册页。

▼ 如何将网络参数重置为安全值

许多缺省情况下安全的网络参数是可调的，因此可能已发生变化，不再是缺省值。如果站点条件允许，可将以下可调参数恢复为缺省值。

开始之前 您必须成为分配有 "Network Management"（网络管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **将 IP 包的源包转发属性设置为 0，然后检验当前值。**
   缺省值可阻止来自欺骗性包的 DOS 攻击。

```
# 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
```

   有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“forwarding（ipv4 或 ipv6）”。

2. **将 IP 包的网络掩码响应属性设置为 0，然后检验当前值。**
   缺省值可防止散播有关网络拓扑的信息。

```
# 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. **将 IP 包的时间戳响应属性设置为 0，然后检验当前值。**
   缺省值可删除系统上的其他 CPU 需求，并防止散播有关网络的信息。

```
# 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. **将 IP 包的广播时间戳响应属性设置为 0，然后检验当前值。**
   缺省值可删除系统上的其他 CPU 需求，并防止散播有关网络的信息。

```
# 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. **阻止 IP 源路由。**

缺省值可防止包绕过网络安全措施。源路由包允许包的源建议路由器上配置的路径以外的其他路径。

注 - 可将该参数设置为 1 以用于诊断目的。诊断完成后，将该值变回 0。

```
# 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
```

有关更多信息，请参见《Oracle Solaris 11.2 可调参数参考手册 》中的“_rev_src_routes”。

另请参见 ipadm(1M) 手册页

♦ ♦ ♦ 第 3 章

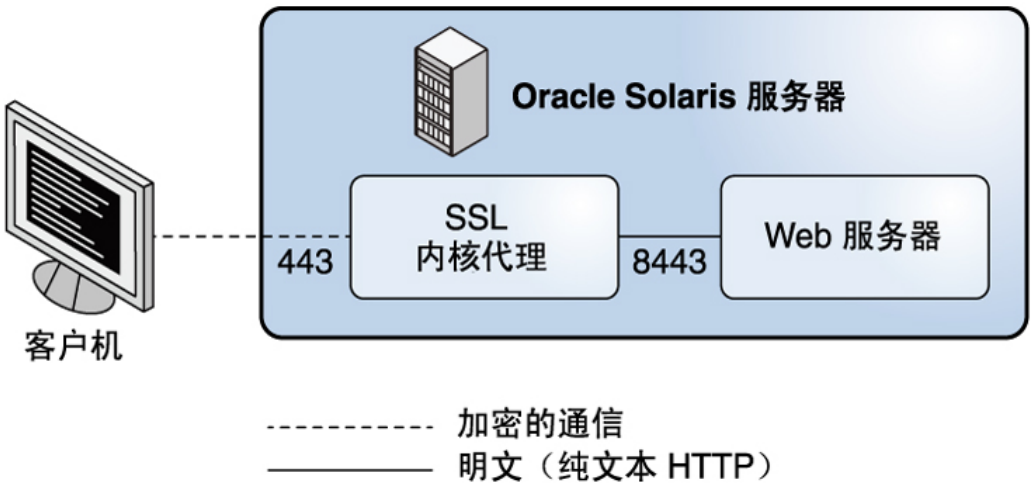
3 Web 服务器和安全套接字层协议

本章说明如何使用安全套接字层 (Secure Sockets Layer, SSL) 协议加密和加速 Oracle Solaris 系统上的 Web 服务器通信。

SSL 内核代理加密 Web 服务器通信

可将在 Oracle Solaris 上运行的任何 Web 服务器配置为在内核级别使用 SSL 协议，即 *SSL* 内核代理。此类 Web 服务器的示例为 Apache 2.2 Web 服务器和 Oracle iPlanet Web Server。SSL 协议可在两个应用程序之间提供保密性、消息完整性和端点身份验证。SSL 内核代理在 Web 服务器上运行时通信将加速。下图显示了基本配置。

图 3-1 内核加密的 Web 服务器通信



SSL 内核代理实现 SSL 协议的服务器端。该代理有以下几个优点。

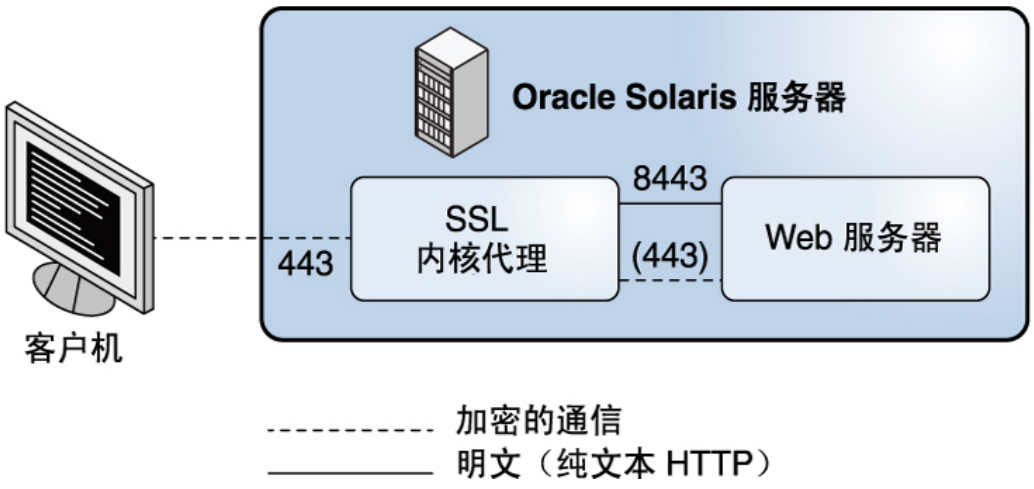
- 该代理加速了服务器应用程序（如 Web 服务器）的 SSL 性能，因此可提供比依赖用户级 SSL 库的应用程序更加优越的性能。性能提高可能超过 35%，这取决于应用程序的工作负荷。
- SSL 内核代理是透明的。它没有指定的 IP 地址。因此，Web 服务器可看到真正的客户机 IP 地址和 TCP 端口。
- SSL 内核代理和 Web 服务器可协同工作。

  图 3-1 “内核加密的 Web 服务器通信” 显示了一个基本方案，其中包含使用 SSL 内核代理的 Web 服务器。在端口 443 上配置 SSL 内核代理，而在端口 8443 上配置 Web 服务器，其中 Web 服务器可收到未加密的 HTTP 通信。
- 如果 SSL 内核代理不支持请求的加密，可将其配置为回退到用户级加密算法。

  图 3-2 “使用用户级回退选项进行内核加密的 Web 服务器通信” 显示了更复杂的方案。将 Web 服务器和 SSL 内核代理配置为回退到用户级 Web 服务器 SSL。

  在端口 443 上配置 SSL 内核代理。在两个端口上配置 Web 服务器。端口 8443 接收未加密的 HTTP 通信，而端口 443 作为回退端口。回退端口接收不受 SSL 内核代理支持的加密套件的加密 SSL 流量。

图 3-2 使用用户级回退选项进行内核加密的 Web 服务器通信



SSL 内核代理 支持 SSL 3.0 和 TLS 1.0 协议，以及最常见的加密套件。有关完整列表，请参见 `ksslcfg(1M)` 手册页。对于不受支持的加密套件，该代理可配置为回退到用户级 SSL 服务器。

使用 SSL 内核代理 保护 Web 服务器

以下过程说明如何配置 Web 服务器以使用 SSL 内核代理：

▼ 如何配置 Apache 2.2 Web 服务器以使用 SSL 内核代理

SSL 内核代理可加速 Apache 2.2 Web 服务器上的 SSL 包处理。此过程可实现图 3-1 “内核加密的 Web 服务器通信” 中所示的简单方案。

开始之前 已配置 Apache 2.2 Web 服务器。Oracle Solaris 中包含该 Web 服务器。

您必须承担 `root` 角色。

1. **停止 Web 服务器。**

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

2. **将服务器私钥和服务器证书放置在一个文件中。**

如果只在 `ssl.conf` 文件中指定了 `SSLCertificateFile` 参数，则指定的文件可直接用于 SSL 内核代理。

如果还指定了 `SSLCertificateKeyFile` 参数，则必须合并证书文件和私钥文件。运行与下面类似的命令以合并文件：

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

3. **确定要用于 `ksslcfg` 命令的参数。**

有关完整的选项列表，请参见 `ksslcfg(1M)` 手册页。必须提供的参数遵循：

- *key-format* – 与 `-f` 选项一起定义证书和密钥格式。对于 SSL 内核代理，支持的格式为 `pkcs11`、`pem` 和 `pkcs12`。
- *key-and-certificate-file* – 与 `-i` 选项一起设置存储 `pem` 和 `pkcs12` *key-format* 选项的服务器密钥和证书的文件位置。
- *password-file* – 与 `-p` 选项一起获取用于加密 `pem` 或 `pkcs12`*key-format* 选项的密钥的口令。对于 `pkcs11`，该口令用于验证 PKCS #11 令牌。必须使用 `0400` 权限保护口令文件。无人参与的重新引导需要该文件。
- *token-label* – 与 `-T` 选项一起指定 PKCS #11 令牌。

- *certificate-label* – 与 -C 选项一起选择 PKCS #11 令牌的证书对象中的标签。
- *proxy-port* – 与 -x 选项一起设置 SSL 代理端口。必须指定标准端口 80 之外的其他端口。Web 服务器在 SSL 代理端口上侦听未加密纯文本流量。通常，此值为 8443。
- *ssl-port* – 为 SSL 内核代理指定侦听端口。通常，此值为 443。

4. 创建 SSL 内核代理 的服务实例。

   使用以下格式之一指定 SSL 代理端口及关联的参数：

   - 指定 PEM 或 PKCS #12 作为密钥格式。

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

   - 指定 PKCS #11 作为密钥格式。

     ```
     # ksslcfg create -f pkcs11 -T PKCS11-token -C certificate-label \
     -p password-file -x proxy-port ssl-port
     ```

5. 验证服务实例是否处于联机状态。

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

   以下输出表明未创建服务实例：

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

6. 配置 Web 服务器以在 SSL 代理端口上侦听。

   编辑 /etc/apache2/2.2/http.conf 文件并添加一行，以定义 SSL 代理端口。如果使用服务器的 IP 地址，Web 服务器将只在该接口上侦听。该行类似于以下内容：

   ```
   Listen proxy-port
   ```

7. 为 Web 服务器设置 SMF 依赖性。

   Web 服务器服务仅在启动 SSL 内核代理实例之后才能启动。以下命令将建立该相关性：

   ```
   # 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. 启用 Web 服务器服务。

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

▼ 如何配置 Oracle iPlanet Web Server 以使用 SSL 内核代理

SSL 内核代理可加速 Oracle iPlanet Web Server 上的 SSL 包处理。此过程可实现图 3-1 “内核加密的 Web 服务器通信” 中所示的简单方案。

开始之前 已安装并配置 Oracle iPlanet Web Server。此服务器可从 Oracle iPlanet Web Server (http://www.oracle.com/technetwork/middleware/iplanetwebserver-098726.html?ssSourceSiteId=ocomen) 下载。有关说明，请参见 Oracle iPLANET WEB SERVER 7.0.15 (http://docs.oracle.com/cd/E18958_01/index.htm)。

您必须成为分配有 "Network Security"（网络安全）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **停止 Web 服务器。**

   使用管理员 Web 界面停止服务器。有关说明，请参见 Oracle iPLANET WEB SERVER 7.0.15 (http://docs.oracle.com/cd/E18958_01/index.htm)。

2. **确定要用于 `ksslcfg` 命令的参数。**

   有关完整的选项列表，请参见 `ksslcfg`(1M) 手册页。有关必须提供的参数列表，请参见如何配置 Apache 2.2 Web 服务器以使用 SSL 内核代理 [33] 中的步骤 3。

3. **创建 SSL 内核代理 的服务实例。**

   使用以下格式之一指定 SSL 代理端口及关联的参数：

   - **指定 PEM 或 PKCS #12 作为密钥格式。**

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

   - **指定 PKCS #11 作为密钥格式。**

     ```
     # ksslcfg create -f pkcs11 -T PKCS11-token -C certificate-label \
     -p password-file -x proxy-port ssl-port
     ```

4. **验证实例是否处于联机状态。**

   ```
   # svcs svc:/network/ssl/proxy
   STATE          STIME    FMRI
   ```

```
online         02:22:22 svc:/network/ssl/proxy:default
```

5. 配置 Web 服务器以在 SSL 代理端口上侦听。
   有关说明，请参见 Oracle iPLANET WEB SERVER 7.0.15 (http://docs.oracle.com/cd/E18958_01/index.htm)。

6. 为 Web 服务器设置 SMF 相关性。
   Web 服务器服务仅在启动 SSL 内核代理实例之后才能启动。以下命令将建立该相关性，假设 Web 服务器服务的 FMRI 为 `svc:/network/http:webserver7`：

```
# 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. 启用 Web 服务器服务。

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

▼ 如何配置 SSL 内核代理以回退到 Apache 2.2 SSL

在此过程中，从头配置 Apache 2.2 Web 服务器并将 SSL 内核代理 配置为主 SSL 会话处理机制。如果客户机提供的 SSL 加密算法集合不包含 SSL 内核代理 提供的加密算法，则 Apache 2.2 Web 服务器将用作回退机制。此过程可实现图 3-2 “使用用户级回退选项进行内核加密的 Web 服务器通信” 中所示的复杂方案。

开始之前 您必须承担 `root` 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 在 Apache 2.2 Web 服务器上，创建要由服务器的 SSL 内核代理使用的密钥证书。

   a. 生成证书签名请求 (Certificate Signing Request, CSR)。
      以下命令为 SSL 内核代理生成 CSR 及关联的私钥：

```
# 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
```

注 - 要符合 FIPS 140 要求，RSA 密钥长度至少为 2048 位。有关更多信息，请参见《Using a FIPS 140 Enabled System in Oracle Solaris 11.2 》。

有关更多信息，请参见 openssl(5) 手册页。

b. 向证书颁发机构 (certificate authority, CA) 发送 CSR。

c. 使用 CA 的签名证书替换 **webcert.pem** 文件。

2. 为 SSL 内核代理配置口令短语和公钥/私钥证书。

a. 创建、保存和保护口令短语。

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

注 - 口令短语不能包含任何空格字符。

b. 将私钥证书和公钥证书合并到一个文件中。

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

c. 为 SSL 内核代理配置公钥/私钥证书和口令短语。

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

3. 将 Web 服务器配置为在端口 **8443** 上侦听未加密的通信。

在 /etc/apache2/2.2/httpd.conf 文件中编辑 Listen 行。

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

4. 将 SSL 模块模板 **ssl.conf** 添加到 Apache 配置目录。

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

该模块为加密连接添加侦听端口 443。

5. 使 Web 服务器可解密 **/root/kssl.pass** 中的口令短语。

a. 创建一个读取 **kssl.pass** 文件的 shell 脚本。

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

b. 使脚本可执行并保护该文件。

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

c. 在 **ssl.conf** 文件中修改 **SSLPassPhraseDialog** 参数以调用 shell 脚本。

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

6. 将 Web 服务器的公钥和私钥证书置于正确位置。

   ssl.conf 文件中的 SSLCertificateFile 和 SSLCertificateKeyFile 参数值包含预期的位置和名称。您可以将证书复制或链接到正确位置。

```
# 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. 启用 Apache 服务。

```
# svcadm enable apache22
```

8. （可选） 验证两个端口是否正在运行。

   使用 openssl s_client 和 kstat 命令查看包。

   a. 使用可供 SSL 内核代理使用的加密算法。

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

   kstat 计数器 kssl_full_handshakes 增加 1 可确认 SSL 内核代理 已对 SSL 会话进行处理。

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

   b. 使用不可供 SSL 内核代理使用的加密算法。

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

   kstat 计数器 kssl_fallback_connections 增加 1 可确认包已到达，但 Apache Web 服务器已对 SSL 会话进行处理。

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

例 3-1 配置 Apache 2.2 Web 服务器以使用 SSL 内核代理

以下命令将为使用 pem 密钥格式的 SSL 内核代理创建一个服务实例：

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

▼ 如何使用区域中的 SSL 内核代理

SSL 内核代理在区域中工作时具有以下限制：

- 所有内核 SSL 管理都必须从全局区域中执行。全局区域管理员需要访问本地区域证书和密钥文件。在全局区域中使用 ksslcfg 命令配置服务实例后，可以在非全局区域中启动 Web 服务器。
- 配置实例时，必须使用 ksslcfg 命令来指定特定的主机名或 IP 地址。特别是，该实例无法为 IP 地址指定 INADDR_ANY。

开始之前 已在非全局区域中配置并启用 Web 服务器服务。

您必须成为分配有 "Network Security"（网络安全）和 "Zone Management"（区域管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **在非全局区域中停止 Web 服务器。**

   例如，要停止 apache-zone 区域中的 Apache Web 服务器，请运行以下命令：

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

2. **在全局区域中，为区域中的 SSL 内核代理创建服务实例。**

   要为 apache-zone 创建服务实例，请使用类似下面的命令：

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

3. **在非全局区域中，启用 Web 服务实例。**

   例如，启用 apache-zone 中的 Web 服务。

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

♦ ♦ ♦ 第 4 章

关于 Oracle Solaris 中的 IP 过滤器

本章概述 Oracle Solaris 的 IP 过滤器功能。有关 IP 过滤器任务，请参见第 5 章 配置 IP 过滤器。

本章包含以下信息：

IP 过滤器介绍

Oracle Solaris 的 IP 过滤器功能是一个防火墙，可提供有状态包过滤和网络地址转换 (network address translation, NAT)。IP 过滤器还包括无状态包过滤以及创建和管理地址池的功能。

包过滤可提供基本的保护以防止基于网络的攻击。IP 过滤器可以按 IP 地址、端口、协议、网络接口和流量方向来进行过滤。IP 过滤器还可以按单个源 IP 地址、目标 IP 地址、IP 地址范围或地址池进行过滤。

IP 过滤器是从开源 IP 过滤器软件派生的。要查看开源 IP 过滤器的许可证条款、所有权和版权声明，缺省路径为 `/usr/lib/ipf/IPFILTER.LICENCE`。如果已经将 Oracle Solaris 安装在其他位置而没有安装在缺省位置，请修改指定的路径，以便在安装位置访问该文件。

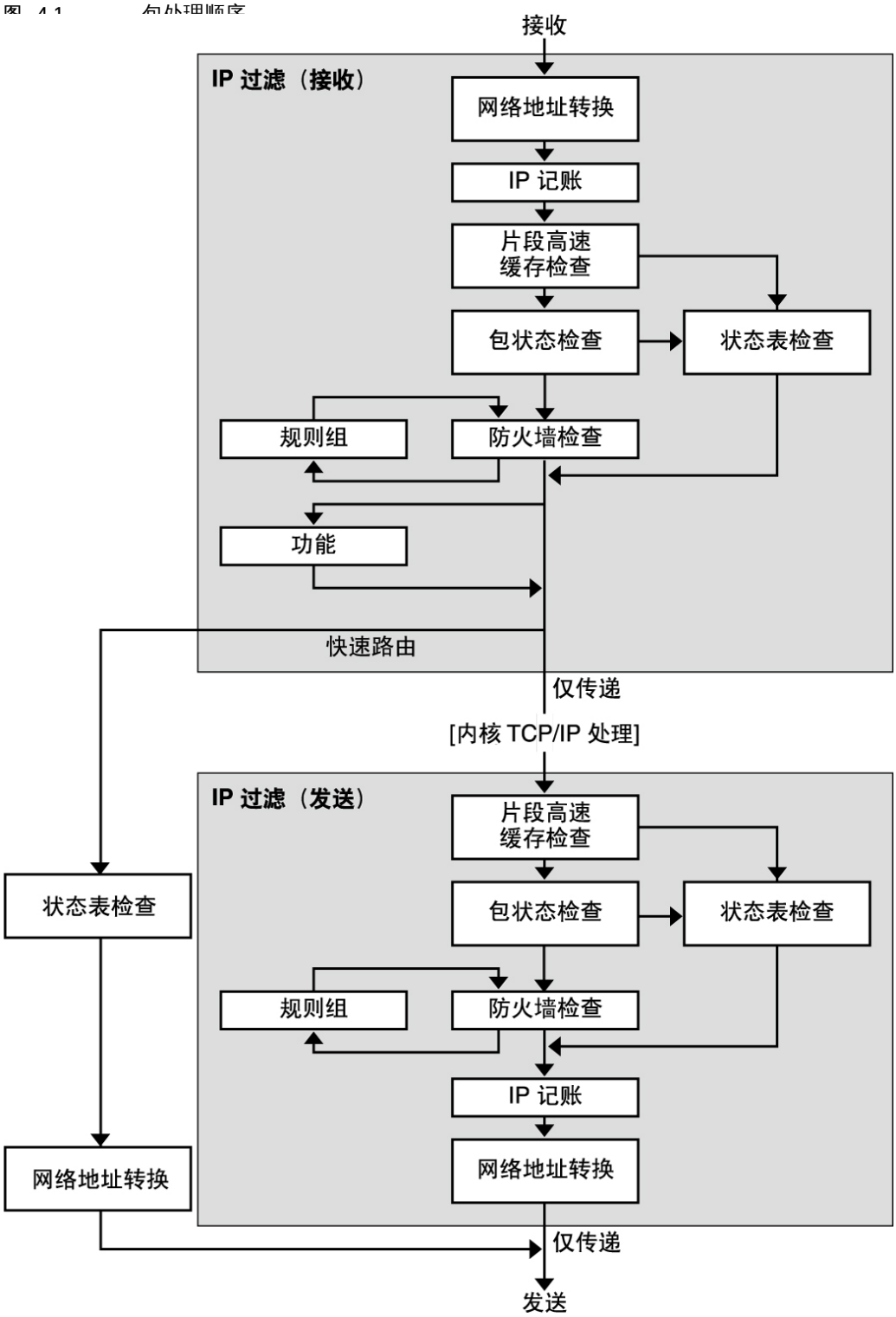
开源 IP 过滤器的信息源

Darren Reed 编写的开源 IP 过滤器软件的主页位于 http://coombs.anu.edu.au/~avalon/ip-filter.html。此站点包括有关开源 IP 过滤器的信息，其中包括指向标题为 "IP Filter Based Firewalls HOWTO"（Brendan Conoboy 和 Erik Fichtner，2002）的教程的链接。此教程提供了在 BSD UNIX 环境中构建防火墙的逐步说明。此教程虽然是针对 BSD UNIX 环境编写的，但也与 Oracle Solaris 上 IP 过滤器的配置相关。

IP 过滤器包处理

在处理包时，IP 过滤器会执行一系列步骤。下图说明处理包的步骤，以及过滤如何与 TCP/IP 协议栈集成在一起。

图 4-1 包处理顺序


接收
IP 过滤（接收）
网络地址转换
IP 记账
片段高速
缓存检查
包状态检查
状态表检查
规则组
防火墙检查
功能
快速路由
仅传递
[内核 TCP/IP 处理]
IP 过滤（发送）
片段高速
缓存检查
状态表检查
包状态检查
状态表检查
规则组
防火墙检查
IP 记账
网络地址转换
网络地址转换
仅传递
发送

包处理顺序包括下列步骤：

- 网络地址转换 (Network Address Translation, NAT)

  将专用 IP 地址转换为不同的公共地址，或者将多个专用地址的别名指定为单个公共地址。当组织具有现有的网络并需要访问 Internet 时，通过 NAT，该组织可解决 IP 地址用尽的问题。
- IP 记帐

  可以分别设置输入规则和输出规则，从而记录所通过的字节数。每次与规则匹配时，都会将包的字节计数添加到该规则中，并允许收集层叠统计信息。
- 片段高速缓存检查

  缺省情况下，分段包会被缓存。特定包的所有段到达时，将应用过滤规则并允许或阻止段。如果规则文件中出现 `set defrag off`，则段未缓存。
- 包状态检查

  如果规则中包括 `keep state`，则会自动传递或阻止指定会话中的所有包，具体取决于规则指明了 `pass` 还是 `block`。
- 防火墙检查

  可以分别设置输入规则和输出规则，确定是否允许包通过 IP 过滤器传入内核的 TCP/IP 例程或者传出到网络上。
- 组

  通过分组可以按树的形式编写规则集合。
- 功能

  功能是指要执行的操作。可能的功能包括 `block`、`pass`、`literal` 和 `send ICMP response`。
- 快速路由

  快速路由指示 IP 过滤器不将包传入 UNIX IP 栈进行路由，从而导致 TTL 递减。
- IP 验证

  已验证的包仅通过防火墙循环一次来防止双重处理。

IP 过滤器使用准则

- IP 过滤器由 SMF 服务 `svc:/network/ipfilter` 管理。有关完整的 SMF 概述，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。有关与 SMF 相关的逐步过程的信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 3 章 “管理服务”。
- IP 过滤器要求直接编辑配置文件。
- IP 过滤器作为 Oracle Solaris 的一部分安装。缺省情况下，将您的系统配置为使用自动联网时将启用 IP 过滤器服务。自动网络配置文件（如 `nwam(5)` 和 `netadm(1M)`

手册页所述）可启用此防火墙。对于自动联网系统上的定制配置，将不启用 IP 过滤器服务。有关与启用服务关联的任务，请参见“配置 IP 过滤器服务” [53]。

- 要管理 IP 过滤器，您必须承担 `root` 角色或分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件。可以将 "IP Filter Management"（IP 过滤器管理）权限配置文件分配给您创建的用户或角色。要创建该角色并将它分配给用户，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“创建角色”。
- Oracle Solaris Cluster 软件对可伸缩服务不支持使用 IP 过滤器进行过滤，但对故障转移服务支持 IP 过滤器。有关在群集中配置 IP 过滤器的准则和限制，请参见《*Oracle Solaris Cluster* 软件安装指南》中的“Oracle Solaris OS 功能限制”。
- 如果在充当系统中其他区域的虚拟路由器的区域中实现 IP 过滤器过则，则支持在各区域之间进行过滤。

使用 IP 过滤器配置文件

IP 过滤器可用于提供防火墙服务或网络地址转换 (network address translation, NAT)。缺省情况下不提供防火墙和 NAT 规则。您必须创建定制配置文件并将这些文件的路径名设置为 IP 过滤器服务属性的值。启用服务后，将在重新引导系统时自动装入这些文件。有关配置文件样例，请参见“IP 过滤器配置文件示例” [76]。有关更多信息，请参见 `svc.ipfd(1M)` 手册页。

使用 IP 过滤器规则集合

要管理防火墙，请使用 IP 过滤器指定用于过滤网络通信流量的规则集合。可以创建以下类型的规则集合：

- 包过滤规则集合
- 网络地址转换 (Network Address Translation, NAT) 规则集合

另外，可以创建地址池以引用 IP 地址组。然后，可以在规则集合中使用这些池。地址池可以加速规则处理。还可使大型地址组更易于管理。

使用 IP 过滤器的包过滤功能

可以使用包过滤规则集合来设置包过滤。使用 `ipf` 命令可以对包过滤规则集合进行处理。有关 `ipf` 命令的更多信息，请参见 `ipf(1M)` 命令。

可以在命令行上使用 `ipf` 命令或在包过滤配置文件中创建包过滤规则。要装入配置文件，您必须创建文件并提供 IP 过滤器服务的路径名。

使用 IP 过滤器可以维护两种包过滤规则集合：活动规则集合和非活动规则集合。大多数情况下，会使用活动规则集合。但是，使用 `ipf -I` 命令可以将命令操作应用于非活动规则列表。除非您选择非活动规则列表，否则 IP 过滤器不会使用该列表。非活动规则列表可提供一个存储规则的位置，而不会影响活动包过滤。

在传递或阻止包之前，IP 过滤器会按照从已配置规则列表开头到其结尾的顺序处理规则列表中的规则。IP 过滤器可维护用于确定它是否将传递包的标志。它会遍历整个规则集合，并基于最后一个匹配规则来确定是传递包还是阻止包。

此过程有两种例外情况。第一种例外情况是当包与包含 `quick` 关键字的规则匹配时。如果规则包括 `quick` 关键字，则会针对该规则执行操作，并且不会检查后续规则。第二种例外情况是当包与包含 `group` 关键字的规则匹配时。如果包与组匹配，则仅会检查标记有该组的规则。

配置包过滤规则

使用以下语法可创建包过滤规则：

action `[in|out]` *option keyword, keyword...*

1. 每个规则都以操作开头。如果包与规则匹配，则 IP 过滤器将操作应用于该包。以下列表包括应用于包的常用操作。

`block`	阻止包通过过滤器。
`pass`	允许包通过过滤器。
`log`	记录包但不确定是阻止包还是传递包。使用 `ipmon` 命令可查看日志。
`count`	将包包括在过滤器统计信息中。使用 `ipfstat` 命令可查看统计信息。
`skip` *number*	使过滤器跳过 *number* 个过滤规则。
`auth`	请求由验证包信息的用户程序执行包验证。该程序会确定是传递包还是阻止包。

2. action 后面的下一个单词必须是 `in` 或 `out`。您的选择将确定是将包过滤规则应用于传入包还是应用于传出包。
3. 接下来，可以从选项列表中进行选择。如果使用多个选项，则这些选项必须采用此处显示的顺序。

`log`	如果规则是最后一个匹配规则，则记录包。使用 `ipmon` 命令可查看日志。

`quick`	如果存在匹配的包，则执行包含 `quick` 选项的规则。所有进一步的规则检查都将停止。
`on` *interface-name*	仅当包移入或移出指定接口时才应用规则。
`dup-to` *interface-name*	复制包并通过 *interface-name* 将副本向外发送到指定时可选的 IP 地址。

注 - 规则中的 `dup-to` 选项允许网络管理员创建一个网络分流器。尽管 Oracle Solaris 仍支持此选项，但其重要性已大幅降低。您可以直接配置现代交换机的端口以执行网络分流，无需在规则中定义此功能。请参阅您的交换机文档，了解如何配置端口以执行网络分流。

`to` *interface-name*	将包移动到 *interface-name* 上的外发队列。

4. 指定选项后，可以从确定包是否与规则匹配的各关键字中进行选择。必须按此处显示的顺序使用以下关键字。

注 - 缺省情况下，所有与配置文件中的任何规则都不匹配的包会通过此过滤器。

`tos`	基于表示为十六进制或十进制整数的服务类型值，对包进行过滤。
`ttl`	基于包的生存时间值与包匹配。在包中存储的生存时间值指明了包在被废弃之前可在网络中存在的时间长度。
`proto`	与特定协议匹配。可以使用在 `/etc/protocols` 文件中指定的任何协议名称，或者使用十进制数来表示协议。关键字 `tcp/udp` 可以用于与 TCP 包或 UDP 包匹配。
`from`/`to`/`all`/`any`	与以下任一项或所有项匹配： 源 IP 地址、目标 IP 地址和端口号。`all` 关键字用于接受来自所有源和发往所有目标的包。
`with`	与和包关联的指定属性匹配。在关键字前面插入 `not` 或 `no` 一词，以便仅当选项不存在时才与包匹配。
`flags`	供 TCP 用来基于已设置的 TCP 标志进行过滤。有关 TCP 标志的更多信息，请参见 `ipf(4)` 手册页。
`icmp-type`	根据 ICMP 类型进行过滤。仅当 `proto` 选项设置为 `icmp` 时才使用此关键字；如果使用 `flags` 选项，则不使用此关键字。

`keep` *keep-options*	确定为包保留的信息。可用的 *keep-options* 包括 `state` 选项。`state` 选项会保留有关会话的信息，并可以保留在 TCP、UDP 和 ICMP 包中。
`head` *number*	为过滤规则创建一个新组，该组由数字 *number* 表示。
`group` *number*	将规则添加到编号为 *number* 的组而不是缺省组。如果未指定其他组，则将所有过滤规则放置在组 0 中。

以下示例说明如何组织包过滤规则语法以创建规则。要阻止从 IP 地址 `192.168.0.0/16` 传入的通信，需要在规则列表中包括以下规则：

```
block in quick from 192.168.0.0/16 to any
```

有关用于编写包过滤规则的完整语法和句法，请参见 `ipf(4)` 手册页。有关与包过滤关联的任务，请参见“管理 IP 过滤器的包过滤规则集合” [59]。有关示例中所示的 IP 地址方案 (`192.168.0.0/16`) 的说明，请参见《在 Oracle Solaris 11.2 中规划网络部署 》中的第 1 章 “规划网络部署”。

使用 IP 过滤器的 NAT 功能

NAT 可设置映射规则，用于将源 IP 地址和目标 IP 地址转换为其他 Internet 或内联网地址。这些规则可修改传入或传出 IP 包的源地址和目标地址并继续发送包。另外，还可以使用 NAT 将流量从一个端口重定向到另一个端口。在对包进行任何修改或重定向的过程中，NAT 将维护包的完整性。

可以在命令行上使用 `ipnat` 命令或在 NAT 配置文件中创建 NAT 规则。必须创建 NAT 配置文件并将其路径名设置为该服务的 `config/ipnat_config_file` 属性的值。缺省值为 `/etc/ipf/ipnat.conf`。有关更多信息，请参见 `ipnat(1M)` 命令。

NAT 规则可以应用到 IPv4 和 IPv6 地址。然而，必须为每种地址类型创建单独的规则。在包含 IPv6 地址的 NAT 规则中，不能同时使用 `mapproxy` 和 `rdrproxy` NAT 命令。

配置 NAT 规则

使用以下语法创建 NAT 规则：

command interface-name parameters

1. 每个规则都以以下命令之一开头：

`map`	在无法控制的循环过程中将一个 IP 地址或网络映射到另一个 IP 地址或网络。

`rdr`	将包从一个 IP 地址和端口对重定向到另一个 IP 地址和端口对。
`bimap`	在外部 IP 地址和内部 IP 地址之间建立双向 NAT。
`map-block`	建立基于静态 IP 地址的转换。此命令基于将地址强制转换为目标范围的算法。

2. 此命令后面的下一个单词是接口名称，如 `bge0`。
3. 接下来，可以从确定 NAT 配置的各种参数中进行选择。其中一些参数包括：

`ipmask`	指定网络掩码。
`dstipmask`	指定 `ipmask` 要转换成的地址。
`mapport`	指定 `tcp`、`udp` 或 `tcp/udp` 协议以及端口号的范围。

以下示例说明如何构造 NAT 规则。要重新编写从源地址为 `192.168.1.0/24` 的 `net2` 设备上传出的包并在外部将该设备的源地址显示为 `10.1.0.0/16`，需要在 NAT 规则集合中包括以下规则：

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

以下规则适用于 IPv6 地址：

```
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
```

有关完整的语法和句法，请参见 `ipnat(4)` 手册页。

使用 IP 过滤器的地址池功能

地址池可为一组地址/网络掩码对建立单个引用。地址池可以减少将 IP 地址与规则相匹配所需的时间。还可使大型地址组更易于管理。

地址池配置规则可驻留在 IP 过滤器服务装入的文件中。必须创建文件，然后将其路径名设置为该服务的 `config/ippool_config_file` 属性的值。缺省值为 `/etc/ipf/ippool.conf`。

配置地址池

使用以下语法可创建地址池：

`table role =` *role-name* `type =` *storage-format* `number =` *reference-number*

`table` 定义对多个地址的引用。

`role` 指定 IP 过滤器中池的角色。可以引用的唯一角色是 `ipf`。

`type` 指定池的存储格式。

`number` 指定过滤规则所用的引用号。

例如，要将地址组 `10.1.1.1` 和 `10.1.1.2` 以及网络 `192.16.1.0` 作为池编号 `13` 引用，需要在地址池配置文件中包括以下规则：

```
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24 };
```

然后，要在过滤规则中引用池编号 `13`，需要构建与以下示例类似的规则：

```
pass in from pool/13 to any
```

请注意，必须在装入包含对池的引用的规则文件之前装入池文件。如果不这样做，则池是未定义的，如以下输出所示：

```
# ipfstat -io
empty list for ipfilter(out)
block in from pool/13(!) to any
```

即使稍后添加池，所添加的池也不会更新内核规则集合。另外，还需要重新装入引用池的规则文件。

有关完整的语法和句法，请参见 `ippool(4)` 手册页。

用于 IP 过滤器的 IPv6

IPv6 包过滤可以基于源/目标 IPv6 地址、包含 IPv6 地址的池和 IPv6 扩展头进行过滤。

IPv6 在许多方面都与 IPv4 类似。但是，这两个版本的 IP 的包头和包大小是不同的，这是 IP 过滤器的重要注意事项。称为 *jumbogram* 的 IPv6 包包含长度超过 65,535 字节的数据报。IP 过滤器不支持 IPv6 jumbogram。

注 - 有关 Jumbograms 的更多信息，请参见 IPv6 Jumbograms, RFC 2675 (http://www.ietf.org/rfc/rfc2675.txt)。

与 IPv6 关联的 IP 过滤器任务和与 IPv4 关联的任务差异不大。最明显的差异是，前者将 `-6` 选项与某些命令一起使用。`ipf` 命令和 `ipfstat` 命令都包括用于 IPv6 包过滤的 `-6`

选项。使用带有 -6 选项的 ipf 命令可以装入和刷新 IPv6 包过滤规则。要显示 IPv6 统计信息，请使用带有 -6 选项的 ipfstat 命令。尽管没有用于 IPv6 支持的关联选项，ipmon 和 ippool 命令仍支持 IPv6。ipmon 命令已增强为包含 IPv6 包的日志记录。ippool 命令支持具有 IPv6 地址的包。您可以为 IPv4 或 IPv6 地址创建单独的池，或创建同时包含 IPv4 地址和 IPv6 地址的池。

要创建可重复使用的 IPv6 包过滤规则，您必须创建特定的 IPv6 文件。然后，将其路径名设置为 IP 过滤器服务的 config/ip6_config_file 属性的值。缺省值为 /etc/ipf/ip6.conf。

有关与 IP 过滤器相关的任务，请参见第 5 章 配置 IP 过滤器。

IP 过滤器手册页

以下手册页涵盖 IP 过滤器。

ipf(1M)	管理 IP 过滤器规则、显示可调参数并执行其他任务。
ipf(4)	包含用于创建 IP 过滤器包过滤规则的语法和句法。
ipfilter(5)	描述 IP 过滤器软件。
ipfs(1M)	在重新引导时保存和恢复 NAT 信息和状态表信息。
ipfstat(1M)	检索和显示有关包处理的统计信息。
ipmon(1M)	打开日志设备并查看包过滤和 NAT 的记录包。
ipnat(1M)	管理 NAT 规则并显示 NAT 统计信息。
ipnat(4)	包含用于创建 NAT 规则的语法和句法。
ippool(1M)	创建和管理地址池。
ippool(4)	包含用于创建 IP 过滤器地址池的语法和句法。
svc.ipfd(1M)	提供有关配置 IP 过滤器服务的信息。

♦ ♦ ♦ 第 5 章

5 配置 IP 过滤器

本章提供 IP 过滤器任务的逐步说明。有关概述信息，请参见第 4 章 关于 Oracle Solaris 中的 IP 过滤器。

本章包含以下主题：

配置 IP 过滤器服务

以下任务列表列出了用于创建 IP 过滤器规则以及启用和禁用服务的过程。

表 5-1 配置 IP 过滤器服务任务列表

任务	参考
查看 IP 过滤器使用的文件以及服务的状态。	如何显示 IP 过滤器服务缺省值 [53]
通过 NAT 和地址池为网络通信和包定制包过滤规则集合。	如何创建 IP 过滤器配置文件 [54]
启用、刷新或禁用 IP 过滤器服务。	如何启用和刷新 IP 过滤器 [56]
修改到达段的包的缺省设置。	如何禁用包重组 [56]
过滤系统上区域之间的流量。	如何启用回送过滤 [57]
停止使用 IP 过滤器。	如何禁用包过滤 [58]

▼ 如何显示 IP 过滤器服务缺省值

开始之前 要运行 `ipfstat` 命令，您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 查看 IP 过滤器服务的配置文件名称和位置。

```
$ 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
```

前三个文件属性具有缺省的文件位置。这些文件不存在，直到您创建它们。如果更改配置文件的位置，必须更改该文件的属性值。有关过程，请参见如何创建 IP 过滤器配置文件 [54]。
在定制自己的包过滤规则时修改第四个文件属性。请参见如何创建 IP 过滤器配置文件 [54] 中的步骤 1 和步骤 2。

2. 确定是否已启用 IP 过滤器服务。

   - 在手动联网系统上，缺省情况下不启用 IP 过滤器。

```
$ 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.
```

   - 在 IPv4 网络的自动联网系统上，运行以下命令来查看 IP 过滤器策略：

```
# ipfstat -io
```

     - 要查看创建策略的文件，请阅读 `/etc/nwam/loc/NoNet/ipf.conf`。此文件仅用于查看。
     - 要修改策略，请参见如何创建 IP 过滤器配置文件 [54]。

注 - 要查看 IPv6 网络的 IP 过滤器策略，请添加 `-6` 选项，与在 `ipfstat -6io` 中一样。有关更多信息，请参见 `ipfstat(1M)` 手册页。

▼ 如何创建 IP 过滤器配置文件

要修改自动配置的网络配置的 IP 过滤器策略，或在手动配置的网络中使用 IP 过滤器，您可以创建配置文件、通知该服务这些文件并启用该服务。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 为 IP 过滤器服务指定策略文件的文件位置。

   该文件包含包过滤规则集合。

   a. 首先将策略文件设置为 **custom**。

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

   b. 然后，指定位置。

   例如，将 /etc/ipf/myorg.ipf.conf 设置为包过滤规则集合的位置。

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

2. 创建包过滤规则集合。

   有关包过滤的信息，请参见“使用 IP 过滤器的包过滤功能” [45]。有关配置文件的示例，请参见“IP 过滤器配置文件示例” [76]和 /etc/nwam/loc/NoNet/ipf.conf 文件。

   注 - 如果您指定的策略文件为空，则不会进行过滤。空的包过滤文件相当于具有以下规则集合：

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

   ---

3. （可选） 为 IP 过滤器创建网络地址转换 (network address translation, NAT) 配置文件。

   要通过 NAT 过滤包，请使用缺省文件名 /etc/ipf/ipnat.conf 为 NAT 规则创建文件。如果使用不同的名称，则必须更改 config/ipnat_config_file 服务属性的值，如：

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

   有关 NAT 的更多信息，请参见“使用 IP 过滤器的 NAT 功能” [48]。

4. （可选） 创建地址池配置文件。

   要将一组地址作为单个地址池引用，请使用缺省名称 /etc/ipf/ippool.conf 为池创建文件。如果使用不同的名称，则必须更改 config/ippool_config_file 服务属性的值，如：

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

   一个地址池可以包含 IPv4 地址和 IPv6 地址的任意组合。有关地址池的更多信息，请参见“使用 IP 过滤器的地址池功能” [49]。

5. （可选） 启用回送流量的过滤。

   如果打算过滤系统中配置的区域之间的流量，则必须启用回送过滤。请参见如何启用回送过滤 [57]。还必须定义应用于这些区域的规则集合。

6. （可选） 禁用分段包重组。

   缺省情况下，在 IP 过滤器中对段进行重组。要修改缺省值，请参见如何禁用包重组 [56]。

▼ 如何启用和刷新 IP 过滤器

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

已完成如何创建 IP 过滤器配置文件 [54]。

1. 启用 IP 过滤器。

   要在最初启用 IP 过滤器，请键入以下命令：

   ```
   # svcadm enable network/ipfilter
   ```

2. 在服务运行时修改 IP 过滤器配置文件后，刷新该服务。

   ```
   # svcadm refresh network/ipfilter
   ```

注 - `refresh` 命令可暂时禁用防火墙。要保留防火墙，请附加规则或添加新的配置文件。有关包含示例的过程，请参见“使用 IP 过滤器规则集合” [59]。

▼ 如何禁用包重组

缺省情况下，在 IP 过滤器中对段进行重组。要禁用该重组，请在策略文件开头插入规则。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件和 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 授权的管理员。`root` 角色具有所有这些权限。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 禁用 IP 过滤器。

```
# svcadm disable network/ipfilter
```

2. 在 IP 过滤器策略文件开头添加以下规则。

```
set defrag off;
```

使用 `pfedit` 命令，如下所示：

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

该规则必须位于文件中的所有 `block` 和 `pass` 规则之前。不过，可以在此行之前插入注释，与以下示例类似：

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

3. 启用 IP 过滤器。

```
# svcadm enable network/ipfilter
```

4. 验证是否未对包进行重组。

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

如果 `current` 的值为 `0`，则段不会进行重组。如果 `current` 为 `1`，则段将进行重组。

▼ 如何启用回送过滤

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件和 `solaris.admin.edit/`*path-to-IPFilter-policy-file* 授权的管理员。`root` 角色具有所有这些权限。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 如果 IP 过滤器正在运行，则将其停止。

```
# svcadm disable network/ipfilter
```

2. 在 IP 过滤器策略文件开头添加以下规则。

```
set intercept_loopback true;
```

使用 `pfedit` 命令，如下所示：

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

此行必须位于文件中定义的所有 `block` 和 `pass` 规则之前。不过，可以在此行之前插入注释，与以下示例类似：

```
...
#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. 启用 IP 过滤器。

```
# svcadm enable network/ipfilter
```

4. 要验证回送过滤的状态，请使用以下命令：

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

如果 `current` 的值为 `0`，则禁用回送过滤。如果 `current` 为 `1`，则启用回送过滤。

▼ 如何禁用包过滤

此过程可从内核中删除所有规则并禁用该服务。如果使用此过程，则必须使用相应的配置文件启用 IP 过滤器以重新启动包过滤和 NAT。有关更多信息，请参见如何启用和刷新 IP 过滤器 [56]。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 要禁用该服务，请使用 **`svcadm`** 命令。

```
# svcadm disable network/ipfilter
```

要测试或调试该服务，您可以在服务运行时删除规则集合。有关更多信息，请参见“使用 IP 过滤器规则集合” [59]。

使用 IP 过滤器规则集合

在以下情况下，可能希望修改或取消激活包过滤和 NAT 规则：

- 要进行测试
- 在认为系统问题是由 IP 过滤器所导致时，对这些问题进行故障排除

以下任务列表列出了与 IP 过滤器规则集合相关的过程。

表 5-2 使用 IP 过滤器规则集合任务列表

任务	参考
查看活动的包过滤规则集合。	如何查看活动的包过滤规则集合 [60]
查看非活动的包过滤规则集合。	如何查看非活动的包过滤规则集合 [60]
激活不同的活动规则集合。	如何激活不同的或更新的包过滤规则集合 [60]
删除规则集合。	如何删除包过滤规则集合 [61]
将规则添加到规则集合。	如何将规则附加到活动的包过滤规则集合 [62] 如何将规则附加到非活动的包过滤规则集合 [63]
在活动和非活动的规则集合之间切换。	如何在活动和非活动的包过滤规则集合之间切换 [64]
从内核中删除非活动规则集合。	如何从内核中删除非活动的包过滤规则集合 [65]
查看活动的 NAT 规则。	如何查看 IP 过滤器中的活动 NAT 规则 [65]
删除 NAT 规则。	如何取消激活 IP 过滤器中的 NAT 规则 [66]
将规则添加到活动的 NAT 规则。	如何将规则附加到 NAT 包过滤规则 [66]
查看活动的地址池。	如何查看活动地址池 [67]
删除地址池。	如何删除地址池 [68]
将规则添加到地址池。	如何将规则附加到地址池 [68]

管理 IP 过滤器的包过滤规则集合

IP 过滤器允许活动和非活动的包过滤规则集合驻留在内核中。活动规则集合确定正在对传入包和传出包执行的过滤。非活动规则集合也存储规则，但不会使用这些规则，除非使非活动规则集合成为活动规则集合。可以管理、查看和修改活动和非活动的包过滤规则集合。

注 - 以下过程提供了 IPv4 网络的示例。对于 IPv6 包，使用 -6 选项，如如何显示 IP 过滤器服务缺省值 [53] 中的步骤 2 所述。

▼ 如何查看活动的包过滤规则集合

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **查看活动的包过滤规则集合。**
  以下示例显示装入到内核中的活动包过滤规则集合的输出。

```
# 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
```

▼ 如何查看非活动的包过滤规则集合

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **查看非活动的包过滤规则集合。**
  以下示例显示非活动的包过滤规则集合的输出。

```
# ipfstat -I -io
pass out quick on net1 all
pass in quick on net1 all
```

▼ 如何激活不同的或更新的包过滤规则集合

如果要执行以下任一任务，请使用以下过程：

- 激活当前 IP 过滤器正在使用的包过滤规则集合之外的另一个包过滤规则集合。
- 重新装入最近已更新的同一过滤规则集合。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **选择以下步骤之一：**
   - 如果要激活完全不同的规则集合，请在单独文件中创建一个新规则集合。
   - 在配置文件中更新当前规则集合。
2. **删除当前的规则集合，并装入新规则集合。**

```
# ipf -Fa -f filename
```

filename 中的规则将替换活动规则集合。

注 - 请勿使用 `ipf -D` 或 `svcadm restart` 之类的命令来装入更新的规则集合。此类命令会公开您的网络，因为它们在装入新规则集合之前禁用防火墙。

例 5-1 激活不同的包过滤规则集合

以下示例说明如何将一个包过滤规则集替换为其他规则集合。

```
# 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
```

例 5-2 重新装入更新的包过滤规则集合

以下示例说明如何重新装入当前处于活动状态且已更新的包过滤规则集合。

也可以列出活动的规则集合。

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

然后，编辑 `/etc/ipf/myorg.ipf.conf` 配置文件，刷新服务，接着再次列出活动的规则集合。

```
# 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
```

▼ 如何删除包过滤规则集合

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 删除规则集合。

```
# ipf -F [a|i|o]
```

`-a`	从规则集合中删除所有过滤规则。
`-i`	删除传入包的过滤规则。
`-o`	删除传出包的过滤规则。

例 5-3 删除包过滤规则集合

以下示例显示如何从活动的过滤规则集合中删除所有过滤规则。

```
# 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)
```

▼ 如何将规则附加到活动的包过滤规则集合

将规则附加到现有规则集合在测试或故障排除时可能非常有用。IP 过滤器服务在添加规则时保持启用状态。但是，这些规则将在刷新、重启或启用服务时丢失，除非它们位于 IP 过滤器服务的属性文件中。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **使用以下方法之一将规则附加到活动规则集合：**
  - 在命令行上使用 `ipf -f -` 命令，将规则附加到规则集合。

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

    在刷新、重启或启用服务时，这些附加的规则不属于 IP 过滤器配置的一部分。
  - 执行以下命令：

    1. 在所选的文件中创建规则集合。
    2. 将已创建的规则添加到活动规则集合。

       ```
       # ipf -f filename
       ```

       filename 中的规则将添加到活动规则集合的结尾。由于 IP 过滤器使用“最后一个匹配规则”算法，因此，除非使用 `quick` 关键字，否则所添加的规则将确定过滤优先级。如果包与包含 `quick` 关键字的规则匹配，则执行该规则的操作，且不检查后续规则。

如果 *filename* 是其中一个 IP 过滤器配置文件属性的值，则将在启用、重启或刷新服务时重新装入这些规则。否则，附加的规则将提供一个临时的规则集合。

例 5-4 将规则附加到活动的包过滤规则集合

以下示例显示如何从命令行将规则添加到活动的包过滤规则集合。

```
# 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
```

▼ 如何将规则附加到非活动的包过滤规则集合

在内核中创建非活动规则集合在测试或故障排除时可能非常有用。该规则集合可与活动规则集合进行切换，而无需停止 IP 过滤器服务。但是，刷新、重启或启用该服务时，必须添加非活动规则集合。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **在所选的文件中创建规则集合。**
2. **将已创建的规则添加到非活动规则集合。**

   ```
   # ipf -I -f filename
   ```

   filename 中的规则将添加到非活动规则集合的结尾。由于 IP 过滤器使用“最后一个匹配规则”算法，因此，除非使用 `quick` 关键字，否则所添加的规则将确定过滤优先级。如果包与包含 `quick` 关键字的规则匹配，则执行该规则的操作，且不检查后续规则。

例 5-5 将规则附加到非活动规则集合

以下示例显示如何将规则从文件添加到非活动规则集合。

```
# 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
```

▼ 如何在活动和非活动的包过滤规则集合之间切换

在内核中切换到其他规则集合在测试或故障排除时可能非常有用。无需停止 IP 过滤器服务即可激活规则集合。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **在活动和非活动的规则集合之间切换。**

```
# ipf -s
```

使用此命令，可以在内核中活动和非活动的规则集合之间切换。请注意，如果非活动规则集合为空，则没有包过滤。

注 - 刷新、重启或启用 IP 过滤器服务时，将会恢复 IP 过滤器服务属性文件中的规则。不会恢复非活动规则集合。

例 5-6 在活动和非活动的包过滤规则集合之间切换

以下示例显示使用 `ipf -s` 命令如何导致非活动规则集合成为活动规则集合，并导致活动规则集合成为非活动规则集合。

- 运行 `ipf -s` 命令之前，`ipfstat -I -io` 命令的输出显示非活动规则集合中的规则。`ipfstat -io` 命令的输出显示活动规则集合中的规则。

```
# 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
```

- 运行 `ipf -s` 命令后，`ipfstat -I -io` 和 `ipfstat -io` 命令的输出显示两个规则集合的内容已切换。

```
# 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
```

▼ 如何从内核中删除非活动的包过滤规则集合

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **在“全部刷新”命令中指定非活动规则集合。**

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

注 - 如果随后运行 `ipf -s`，则空的非活动规则集合将成为活动规则集合。空的活动规则集合意味着不会执行过滤。

例 5-7 从内核中删除非活动的包过滤规则集合

以下示例显示如何刷新非活动的包过滤规则集合以便删除所有规则。

```
# 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)
```

管理 IP 过滤器的 NAT 规则

以下过程可以管理、查看和修改 IP 过滤器的 NAT 规则。

▼ 如何查看 IP 过滤器中的活动 NAT 规则

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **查看活动的 NAT 规则。**

  以下示例显示活动 NAT 规则集合的输出。

```
# ipnat -l
```

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

List of active sessions:
```

▼ 如何取消激活 IP 过滤器中的 NAT 规则

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 从内核中删除 NAT 规则。

  ```
  # ipnat -FC
  ```

  `-C` 选项删除当前 NAT 规则列表中的所有项。`-F` 选项删除当前 NAT 转换表（它显示当前活动的 NAT 映射）中的所有活动项。

例 5-8 删除 NAT 规则

以下示例显示如何删除当前 NAT 规则中的项。

```
# 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:
```

▼ 如何将规则附加到 NAT 包过滤规则

将规则附加到现有规则集合在测试或故障排除时可能非常有用。IP 过滤器服务在添加规则时保持启用状态。但是，NAT 规则将在刷新、重启或启用服务时丢失，除非它们位于 IP 过滤器服务的属性文件中。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 使用以下方法之一将规则附加到活动规则集合：
  - 在命令行上使用 `ipnat -f -` 命令，将规则附加到 NAT 规则集合。

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

在刷新、重启或启用服务时，这些附加的规则不属于 IP 过滤器配置的一部分。

- 执行以下命令：

  1. 在所选的文件中创建其他 NAT 规则。
  2. 将已创建的规则添加到活动的 NAT 规则。

     ```
     # ipnat -f filename
     ```

     filename 中的规则将添加到 NAT 规则的结尾。

     如果 *filename* 是其中一个 IP 过滤器配置文件属性的值，则将在启用、重启或刷新服务时重新装入这些规则。否则，附加的规则将提供一个临时的规则集合。

例 5-9 将规则附加到 NAT 规则集合

以下示例显示如何从命令行将规则添加到 NAT 规则集合。

```
# 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:
```

管理 IP 过滤器的地址池

以下过程可以管理、查看和修改地址池。

▼ 如何查看活动地址池

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **查看活动地址池。**
  以下示例显示如何查看活动地址池的内容。

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

▼ 如何删除地址池

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **删除当前地址池中的项。**

```
# ippool -F
```

例 5-10 删除地址池

以下示例显示如何删除地址池。

```
# 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
```

▼ 如何将规则附加到地址池

将规则附加到现有规则集合在测试或故障排除时可能非常有用。IP 过滤器服务在添加规则时保持启用状态。但是，地址池规则将在刷新、重启或启用服务时丢失，除非它们位于 IP 过滤器服务的属性文件中。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **使用以下方法之一将规则附加到活动规则集合：**

   - 在命令行上使用 `ippool -f -` 命令，将规则附加到规则集合。

     ```
     # 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 -
     ```

     在刷新、重启或启用服务时，这些附加的规则不属于 IP 过滤器配置的一部分。
   - 执行以下命令：

     1. 在所选的文件中创建其他地址池。
     2. 将已创建的规则添加到活动地址池。

        ```
        # ippool -f filename
        ```

        filename 中的规则将添加到活动地址池的结尾。

2. 如果这些规则包含不在原始规则集合中的池，请执行以下步骤：

   a. 将池添加到新的包过滤规则。

   b. 将新的包过滤规则附加到当前规则集合。
   请按照如何将规则附加到活动的包过滤规则集合 [62]中的说明操作。

注 - 不要刷新或重新启动 IP 过滤器服务。您将丢失已添加的地址池规则。

例 5-11 将规则附加到地址池

以下示例显示如何从命令行将地址池添加到地址池规则集合。

```
# 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; };
```

显示 IP 过滤器的统计信息

表 5-3 显示 IP 过滤器统计信息和信息任务列表

任务	参考
查看状态表。	如何查看 IP 过滤器的状态表 [69]
查看包状态统计信息。	如何查看 IP 过滤器的状态统计信息 [70]
列出 IP 过滤器的可调参数。	如何查看 IP 过滤器的可调参数 [71]
查看 NAT 统计信息。	如何查看 IP 过滤器的 NAT 统计信息 [71]
查看地址池统计信息。	如何查看 IP 过滤器的地址池统计信息 [72]

▼ 如何查看 IP 过滤器的状态表

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 查看状态表。

```
# ipfstat
```

注 - 可以使用 -t 选项以 UNIX top 实用程序格式查看状态表。

例 5-12 查看 IP 过滤器的状态表

以下示例显示状态表的输出。

```
# 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
```

▼ 如何查看 IP 过滤器的状态统计信息

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 查看状态统计信息。

```
# ipfstat -s
```

例 5-13 查看 IP 过滤器的状态统计信息

以下示例显示状态统计信息的输出。

```
# 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
```

▼ 如何查看 IP 过滤器的可调参数

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 查看 IP 过滤器的内核可调参数。
  以下输出已截断。

```
# 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
```

▼ 如何查看 IP 过滤器的 NAT 统计信息

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 查看 NAT 统计信息。

```
# ipnat -s
```

例 5-14 查看 IP 过滤器的 NAT 统计信息

以下示例显示 NAT 统计信息。

```
# ipnat -s
mapped  in      0       out     0
added   0       expired 0
no memory       0       bad nat 0
inuse   0
rules   1
wilds   0
```

▼ 如何查看 IP 过滤器的地址池统计信息

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- 查看地址池统计信息。

```
# ippool -s
```

例 5-15 查看 IP 过滤器的地址池统计信息

以下示例显示地址池统计信息。

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

处理 IP 过滤器的日志文件

表 5-4 使用 IP 过滤器的日志文件任务列表

任务	参考
创建单独的 IP 过滤器日志文件。	如何为 IP 过滤器设置日志文件 [73]
查看状态日志文件、NAT 日志文件和常规日志文件。	如何查看 IP 过滤器的日志文件 [74]
刷新包日志缓冲区。	如何刷新包日志缓冲区 [75]
将记录的包保存到文件中，以供日后参考。	如何将记录的包保存到文件中 [75]

▼ 如何为 IP 过滤器设置日志文件

缺省情况下，IP 过滤器的所有日志信息都由 `syslog` 记录。不妨创建一个日志文件来单独记录 IP 过滤器通信信息，将其与可能记录在 `syslog` 日志文件中的其他数据相区分，这不失为一个良好做法。

开始之前 您必须承担 `root` 角色。

1. **确定已启用的 `system-log` 服务实例。**

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

注 - 如果 `rsyslog` 服务实例联机，请修改 `rsyslog.conf` 文件。

2. **通过添加以下两行来编辑 `/etc/syslog.conf` 文件：**

```
# Save IP Filter log output to its own file
local0.debug             /var/log/log-name
```

注 - 在条目中使用 Tab 键而不是空格键来分隔 `local0.debug` 与 `/var/log/`*log-name*。有关更多信息，请参见 `syslog.conf(4)` 和 `syslogd(1M)` 手册页。

3. **创建新日志文件。**

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

4. **刷新 `system-log` 服务的配置信息。**

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

注 - 如果 `rsyslog` 服务已启用，请刷新 `system-log:rsyslog` 服务实例。

例 5-16 创建 IP 过滤器日志

以下示例说明如何创建 `ipmon.log` 以归档 IP 过滤器信息。

编辑 `syslog.conf`。

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

然后，在命令行上创建文件，重新启动服务。

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

▼ 如何查看 IP 过滤器的日志文件

开始之前 已完成如何为 IP 过滤器设置日志文件 [73]。

您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **查看状态日志文件、NAT 日志文件或常规日志文件。**

  要查看日志文件，请键入以下命令，并使用适当的选项：

  # ipmon -o [S|N|I] *filename*

S	显示状态日志文件。
N	显示 NAT 日志文件。
I	显示常规 IP 日志文件。

  - **要查看所有状态日志文件、NAT 日志文件和常规日志文件，请使用所有选项：**

    # ipmon -o SNI *filename*

  - **停止 ipmon 守护进程后，您可以使用 ipmon 命令来显示状态日志文件、NAT 日志文件和 IP 过滤器日志文件：**

    ```
    # pkill ipmon
    # ipmon -a filename
    ```

注 - 如果 ipmon 守护进程仍在运行，请勿使用 ipmon -a 语法。通常，该守护进程会在系统引导期间自动启动。发出 ipmon -a 命令可以打开 ipmon 的另一个副本。这样，两个副本将读取相同的日志信息，但只有一个副本会获得特定日志消息。

有关查看日志文件的更多信息，请参见 ipmon(1M) 手册页。

例 5-17 查看 IP 过滤器的日志文件

以下示例显示了来自 /var/ipmon.log 的输出。

```
# 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
```

或者

```
# 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
```

▼ 如何刷新包日志缓冲区

此过程可清除缓冲区并在屏幕上显示输出。

开始之前 您必须成为分配有 "IP Filter Management"（IP 过滤器管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

- **刷新包日志缓冲区。**

```
# ipmon -F
```

例 5-18 刷新包日志缓冲区

以下示例显示删除日志文件时的输出。即使日志文件为空，系统也将提供一个报告，如此示例所示。

```
# ipmon -F
0 bytes flushed from log buffer
0 bytes flushed from log buffer
0 bytes flushed from log buffer
```

▼ 如何将记录的包保存到文件中

您可以在故障排除过程中或希望手动审计通信时将包保存到文件中。

开始之前 您必须承担 `root` 角色。

- **将记录的包保存到文件中。**

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

继续将包记录到 *filename* 文件中，直到您通过键入 `Ctrl-C` 组合键使命令行提示符重新出现来中断该过程。

例 5-19 将记录的包保存到文件中

以下示例显示将记录的包保存到文件中时所出现的结果。

```
# 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 过滤器配置文件示例

以下示例说明应用到单个主机、服务器和路由器的包过滤规则。

配置文件遵循标准的 UNIX 语法规则：

- 井号 (#) 指示包含注释的行。
- 规则和注释可以共存于同一行上。
- 允许使用额外的空格来增强规则的可读性。
- 规则可以延续多行。行尾的反斜杠 (\) 指示规则在下一行上继续。

有关详细的语法信息，请参见“配置包过滤规则” [46]。

例 5-20 IP 过滤器主机配置

此示例显示带有 `net0` 网络接口的主机系统上的配置。

```
# 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
```

此规则集合以两个无限制规则开始，分别允许将任何内容传入和传出 `net0` 接口。第二个规则集合阻止从专用地址空间 `10.0.0.0` 和 `172.16.0.0` 传入的任何包进入防火墙。下一个规则集合阻止来自主机系统的特定内部地址。最后一个规则集合阻止从端口 6000 和端口 111 上传入的包。

例 5-21 IP 过滤器服务器配置

此示例显示用作 Web 服务器的主机系统的配置。此系统具有 `net0` 网络接口。

```
# 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
```

例 5-22 IP 过滤器路由器配置

此示例说明具有内部接口 `net0` 和外部接口 `net1` 的路由器的配置。

```
# 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
```

♦ ♦ ♦ 第 6 章

关于 IP 安全体系结构

IP 安全体系结构 (IPsec) 为 IPv4 和 IPv6 网络中的 IP 包提供加密保护。

本章包含以下主题：

要在网络上实现 IPsec，请参见第 7 章 配置 IPsec。有关参考信息，请参见第 12 章 IPsec 和密钥管理参考。

IPsec 介绍

IPsec 通过使用加密保护 IP 包内容，通过验证包内容提供完整性检查。因为 IPsec 在网络层执行，所以网络应用程序能够利用 IPsec 的优势，同时又不必将自身配置为使用 IPsec。若使用得当，IPsec 是保证网络通信安全的有效工具。

IPsec 使用以下术语：

- **安全协议** – 应用于 IP 包的保护。authentication header（验证头）(AH) 通过添加完整性检查矢量 (integrity check vector, ICV) 来保护 IP 包，它是包括多个 IP 头的完整包的散列。向接收者保证包未遭到修改。它不通过加密提供保密性。

  encapsulating security payload, ESP（封装安全有效负荷）会保护 IP 包的有效负荷。加密包的有效负荷可以提供保密性，使用 ICV 可以确保数据完整性。

- **安全关联 (security association, SA)** – 加密参数、密钥、IP 安全协议、IP 地址、IP 协议、端口数以及其他用于匹配特定 SA 与特定通信流的参数。
- **安全关联数据库 (security associations database, SADB)** – 存储安全关联的数据库。SA 会被 security parameter index, SPI（安全参数索引）、安全协议以及目标 IP 地址引用。这三个元素可以唯一地标识 IPsec SA。收到含有 IPsec 头（ESP 或 AH）的 IP 包时，系统会在 SADB 中搜索匹配的 SA。如果找到匹配的 SA，则用它来允许 IPsec 执行包解密和验证操作。如果验证失败或未找到匹配的 SA，则放弃包。
- **密钥管理** – 安全地生成和分发加密算法使用的密钥以及生成用来存储密钥的 SA。
- **安全策略数据库 (security policy database, SPD)** – 指定要应用到 IP 通信的安全策略的数据库。SPD 会过滤通信内容，确定应该如何处理包。可以放弃包，也可以以明文形式传递包。或者，包可以由 IPsec 保护，即应用安全策略。

  对于外发包，IPsec 策略会确定是否应将 IPsec 应用到 IP 包。如果应用 IPsec，IP 模块将在 SADB 中搜索匹配的 SA，并使用此 SA 执行策略。

  对于传入包，IPsec 策略会确保收到的包具有适当的保护级别。如果策略要求来自特定 IP 地址的包要受 IPsec 保护，系统将放弃任何未受保护的包。如果传入包受 IPsec 保护，IP 模块将在 SADB 中搜索匹配的 SA，并将此 SA 应用到包。

应用程序也可以调用 IPsec，以便在每个套接字级别将安全机制应用于 IP 包。如果端口上的套接字为连接状态，且随后对此端口应用 IPsec 策略，则使用此套接字的通信不受 IPsec 保护。当然，将 IPsec 策略应用于端口之后，在此端口上打开的套接字将受 IPsec 策略保护。

IPsec 包流

图 6-1 “应用于外发包过程的 IPsec” 显示了当已经在外发包上调用 IPsec 时，IP packet（IP 包）如何继续传送。此流程图说明了可以对包应用验证头 (authentication header, AH) 和封装安全有效负荷 (encapsulating security payload, ESP) 实体的位置。后续各节介绍了如何应用这些实体以及如何选择算法。

图 6-2 “应用于传入包过程的 IPsec” 显示了 IPsec 传入流程。

图 6-1 应用于外发包过程的 IPsec

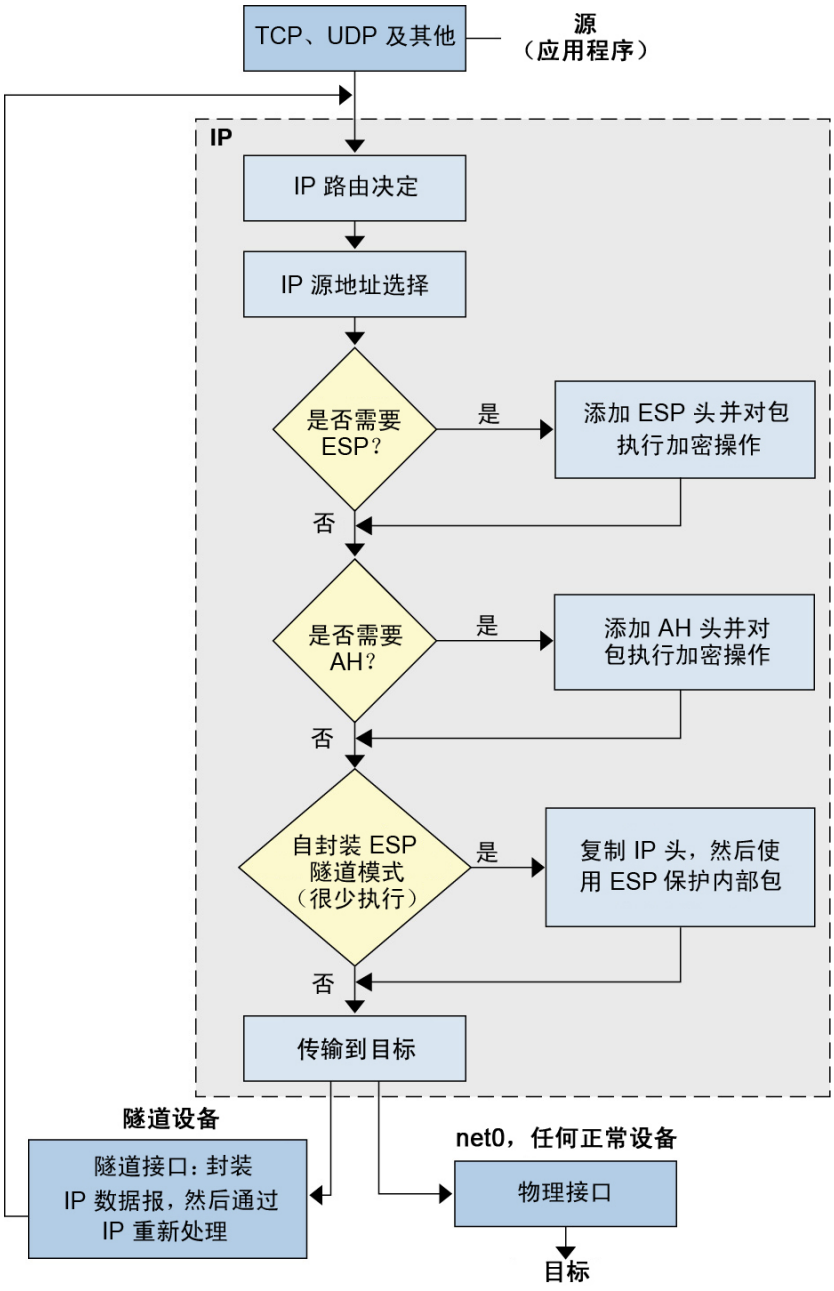
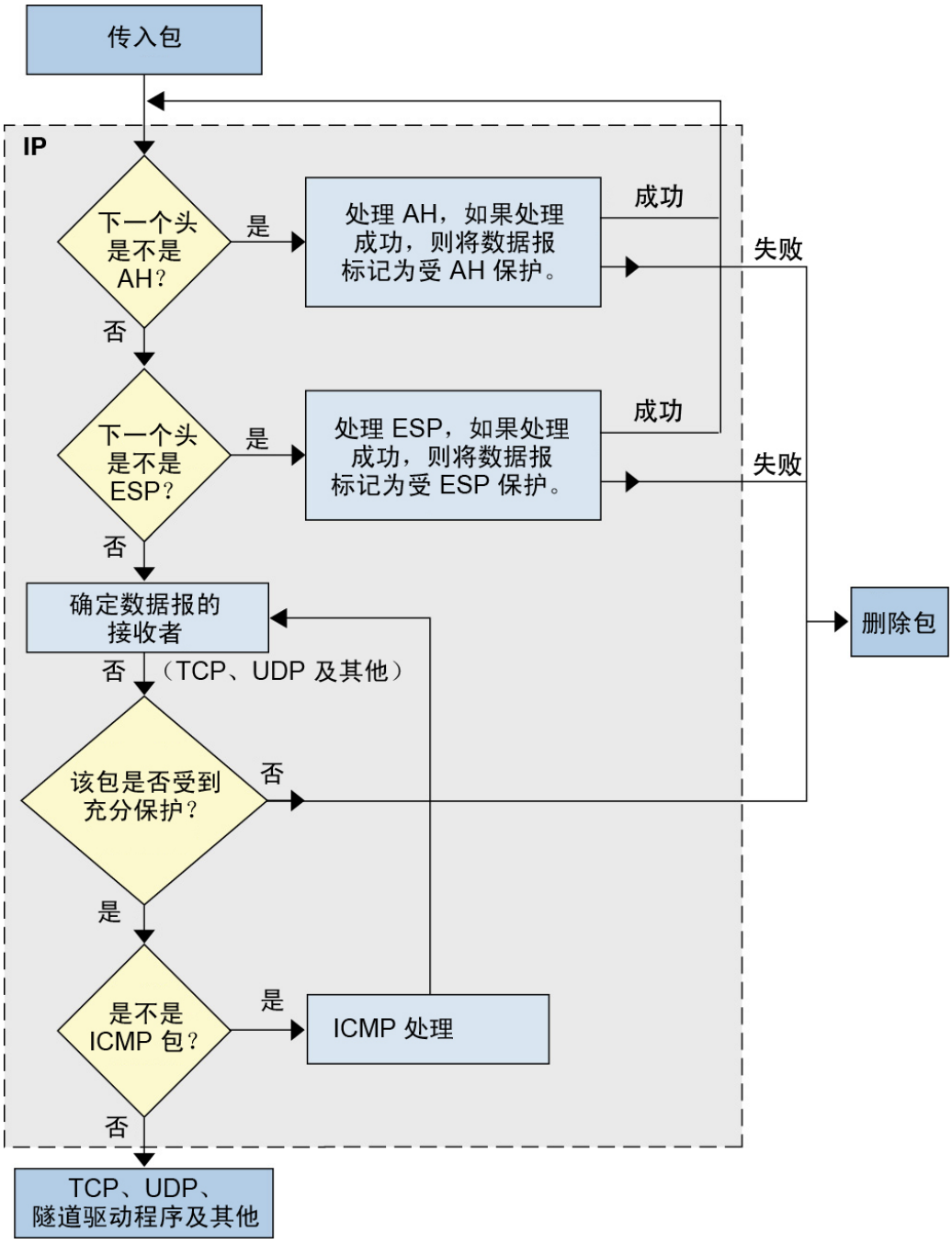


图 6-2 应用于传入包过程的 IPsec



IPsec 安全关联

IPsec 安全关联 SA 会定义安全属性，应用于与 SA 中也存储的 IP 参数匹配的 IP 包。每个 SA 都是单向的。因为大部分通信都是双向的，所以单一连接需要两个 SA。

以下三个元素综合在一起，可以唯一地标识 IPsec SA：

- 安全协议（AH 或 ESP）
- 目标 IP 地址
- security parameter index, SPI（安全参数索引）

SA 的 SPI 可以提供额外保护，并在受 IPsec 保护的包的 AH 或 ESP 头中传送。`ipsecah(7P)` 和 `ipsecesp(7P)` 手册页说明了 AH 和 ESP 提供的保护范围。完整性校验和值用于验证包。如果验证失败，则会丢弃包。

安全关联存储在安全关联数据库 (security associations database, SADB) 中。基于套接字的管理接口 `PF_KEY` 使特权应用程序能以编程方式管理数据库。例如，IKE 守护进程和 `ipseckey` 命令会使用 `PF_KEY` 套接字接口。

有关 IPsec SADB 的更为完整的说明，请参见“IPsec 的安全关联数据库” [201]。

有关如何管理 SADB 的更多信息，请参见 `pf_key(7P)` 和 `ipseckey(1M)` 手册页。

IPsec 安全关联的密钥管理

安全关联 (security association, SA) 需要加密材料来进行验证和加密。加密材料的管理称作密钥管理。Oracle Solaris 可以为 IPsec SA 提供两种密钥管理方法：IKE 和手动密钥管理。

使用 IKE 生成 IPsec SA

Internet 密钥交换 (Internet Key Exchange, IKE) 协议可以自动处理密钥管理。Oracle Solaris 11.2 支持 IKE 版本 2 (IKEv2) 和 IKE 版本 1 (IKEv1)。

建议用户使用 IKE 管理 IPsec SA。这些密钥管理协议可以提供以下优势：

- 配置简单
- 提供强大的对等方验证功能
- 自动使用高质量的随机密钥源生成 SA
- 无需管理员干预就能生成新的 SA

有关更多信息，请参见“IKE 的工作原理” [118]。

要配置 IKE，请参见第 9 章 配置 IKEv2。如果与不支持 IKEv2 协议的系统通信，请遵循第 10 章 配置 IKEv1 中的说明。

使用手动密钥生成 IPsec SA

使用手动密钥比使用 IKE 更加复杂，而且有可能带来风险。系统文件 `/etc/inet/secret/ipseckeys` 包含加密密钥。如果这些密钥泄密，它们可用于解密记录的网络通信。因为 IKE 会频繁更改密钥，所以泄密暴露窗口要小得多。请仅对不支持 IKE 的系统使用 `ipseckeys` 文件或其命令接口 `ipseckey`。

虽然 `ipseckey` 命令只有有限的常规选项，但是此命令支持丰富的命令语言。您可以指定使用专用于手动加密的程序接口发送请求。有关其他信息，请参见 `ipseckey(1M)` 和 `pf_key(7P)` 手册页。

通常，由于某种原因而无法使用 IKE 时，会使用手动 SA 生成。但是，如果 SPI 值是唯一的，可以同时使用手动 SA 生成和 IKE。

IPsec 保护协议

IPsec 提供了两种用于保护数据的安全协议：

- 验证头 (Authentication Header, AH)
- 封装安全有效负荷 (Encapsulating Security Payload, ESP)

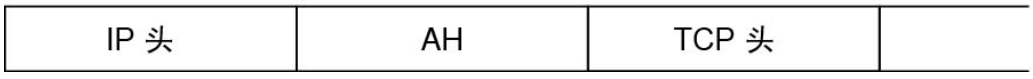
AH 通过使用验证算法提供数据完整性。它不对包进行加密。

ESP 通常通过加密算法保护包，通过验证算法提供数据完整性。有些加密算法可以同时提供加密和验证，例如 AES GCM。

AH 协议不能与网络地址转换 (network address translation, NAT) 一起使用。

验证头

authentication header（验证头）为 IP 包提供了数据验证、高完整性以及重放保护。AH 会保护 IP 包中更为重要的部分。如下图所示，AH 插在 IP 数据包头和传输头之间。



传输头可以是 TCP、UDP、SCTP 或 ICMP。如果使用的是 tunnel（隧道），则传输头可以是另一个 IP 数据包头。

封装安全有效负荷

encapsulating security payload, ESP（封装安全有效负荷）协议为 ESP 所封装的内容提供保密性。ESP 也提供 AH 提供的服务。然而，ESP 不保护外部的 IP 头。ESP 会提供验证服务以确保受保护的包的完整性。因为 ESP 使用启用了加密的技术，因此提供 ESP 的系统可能会受进出口控制法制约。

ESP 头和尾会封装 IP 有效负荷。将加密与 ESP 一起使用时，它仅应用于 IP 有效负荷数据，如下图所示。

加密

在 TCP 包中，ESP 头会接受验证，并且它可以封装 TCP 头及其数据。如果包是 IP-in-IP 包，则 ESP 会保护内部的 IP 包。由于每个套接字的策略允许自封装，因此，ESP 可以在需要时封装 IP 选项。

编写使用 `setsockopt（）` 的程序时，可以使用自封装。如果设置了自封装，则会生成 IP 头的副本，用来构建 IP-in-IP 包。例如，如果未在 TCP 套接字上设置自封装，则会以下列格式发送包：

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

如果在 TCP 套接字上设置了自封装，则会以下列格式发送包：

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

有关进一步介绍，请参见“IPsec 中的传输模式和隧道模式” [89]。

使用 AH 和 ESP 时的安全注意事项

下表比较了 AH 和 ESP 提供的保护。

表 6-1 由 IPsec 中的 AH 和 ESP 提供的保护

协议	包范围	保护	防止的攻击
AH	保护包中从 IP 包头到传输数据结尾的内容	提供高完整性、数据验证：	重放、剪贴

协议	包范围	保护	防止的攻击
		■ 确保接收者接收到的正是发送者发送的内容 ■ 在 AH 没有启用重放保护时容易受到重放攻击影响	
ESP	保护包中从 ESP 头到传输数据结尾的内容	使用加密选项时，对 IP 有效负荷进行加密。保证保密性	窃听
		使用验证选项时，提供与 AH 相同的有效负荷保护	重放、剪贴
		同时使用两个选项时，提供高完整性、数据验证和保密性	重放、剪贴、窃听

IPsec 中的验证算法和加密算法

IPsec 安全协议使用两种类型的算法，即验证和加密。AH 协议使用验证算法。ESP 协议可以使用加密算法以及验证算法。您可以使用 `ipsecalgs` 命令获取系统上的算法及其属性的列表。有关更多信息，请参见 `ipsecalgs(1M)` 手册页。您也可以使用 `getipsecalgbyname(3NSL)` 手册页中介绍的功能来检索算法属性。

IPsec 使用加密框架执行加密和验证。使用加密框架，IPsec 可以利用硬件支持的硬件加速。

有关更多信息，请参见以下内容：

- 《在 Oracle Solaris 11.2 中管理加密和证书 》中的第 1 章 “加密框架”
- 《面向开发者的 Oracle Solaris 11 安全性指南 》中的第 8 章 “Oracle Solaris 加密框架介绍”

IPsec 保护策略

IPsec 保护策略可以在以下级别应用：

- 系统范围级别
- 每个套接字级别

IPsec 将系统范围策略应用到与 IPsec 策略规则匹配的外发包和传入包。此规则可以指定特定算法或允许若干算法中的一个。由于存在系统可识别的其他数据，因此可以将其他规则应用于外发包。

传入包可被接受或丢弃。丢弃或接受传入包的决定根据若干条件做出。如果条件重叠或冲突，则使用首先解析的规则。

您可以为 IPsec 策略指定例外情况，使其在除例外情况以外的其他情况下应用于大多数包。也就是说，您可以绕过 IPsec 策略。绕过操作可以在系统范围级别或每个套接字级别执行。

在包含共享 IP 地址上区域的系统中，该系统内的通信将执行策略，但是不会应用实际的安全机制。相反，应用于系统内部包上的外发策略将转移到应用了那些机制的传入包。对于专用 IP 区域，则会执行策略，应用实际的安全机制。

可以使用 `ipsecinit.conf` 文件和 `ipsecconf` 命令来配置 IPsec 策略。有关详细信息和示例，请参见 `ipsecconf(1M)` 手册页和第 7 章 配置 IPsec。

IPsec 中的传输模式和隧道模式

IPsec 标准定义了 IPsec 操作的两种不同模式：传输模式和隧道模式。传输模式和隧道模式之间的主要差异在于策略的应用位置。在隧道模式中，原始包在另一个 IP 头中封装。另一个头中的地址可以有所不同。

在每种模式下，包可以由 AH、ESP 或同时由这二者提供保护。这些模式在策略应用方面存在差异，具体如下所示：

- 在传输模式中，外部头中的 IP 地址用于确定将应用于包的 IPsec 策略。
- 在隧道模式中，会发送两个 IP 头。内部 IP 包会确定保护其内容的 IPsec 策略。

  隧道模式可以应用于任何端点系统与中间系统的组合，例如安全网关。

在传输模式下，IP 头、下一个头以及下一个头支持的任何端口都可用于确定 IPsec 策略。实际上，IPsec 可在一个端口不同粒度的两个 IP 地址之间强制实行不同的传输模式策略。例如，如果下一个头是 TCP（支持端口），则可为外部 IP 地址的 TCP 端口设置 IPsec 策略。

隧道模式仅对 IP-in-IP 包有效。在隧道模式下，系统在内部 IP 包的内容上强制执行 IPsec 策略。可针对不同的内部 IP 地址强制实施不同的 IPsec 策略。也就是说，内部 IP 头、其下一个头及下一个头支持的端口，可以强制执行策略。与传输模式不同，在隧道模式下，外部 IP 头并不指示其内部 IP 包使用的策略。

因此，在隧道模式下，可为路由器后面的 LAN 的子网和这些子网上的端口指定 IPsec 策略。也可在这些子网上为特定的 IP 地址（即主机）指定 IPsec 策略。这些主机的端口也可以具有特定的 IPsec 策略。但是，如果有动态路由协议在隧道上运行，请勿使用子网选择或地址选择，因为对等网络上的网络拓扑的视图可能会更改。更改可能使静态 IPsec 策略失效。有关包括配置静态路由的隧道设置过程示例，请参见“使用 IPsec 保护 VPN” [101]。

在 Oracle Solaris 中，隧道模式只能在 IP 隧道网络接口上强制执行。有关隧道接口的信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道 》中的第 4 章 “关于 IP 隧道管理”。IPsec 策略会提供 `tunnel` 关键字，选择 IP 隧道网络接口。当规则中出现 `tunnel` 关键字时，在此规则中指定的所有选定器都应用到内部包中。

下图显示了不受保护的 TCP 包的 IP 数据包头。

图 6-3 携带 TCP 信息的不受保护的 IP 包

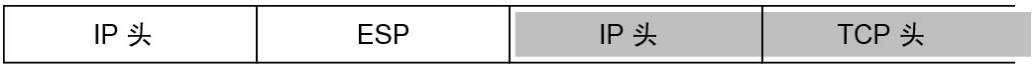
在传输模式下，ESP 按下图所示的方式保护数据。阴影部分表示包的加密部分。

图 6-4 携带 TCP 信息的受保护的 IP 包

加密

在隧道模式中，整个包位于 ESP 头内部。图 6-3 “携带 TCP 信息的不受保护的 IP 包”中的包由外部 IPsec 头以隧道模式保护（在此示例中为 ESP），如下图所示。

图 6-5 以隧道模式保护的 IPsec 包



加密

IPsec 策略为隧道模式和传输模式提供关键字。有关更多信息，请查看以下内容：

- 有关每个套接字策略的详细信息，请参见 `ipsec(7P)` 手册页。
- 有关每个套接字策略的示例，请参见如何使用 IPsec 保护 Web 服务器与其他服务器的通信 [99]。
- 有关隧道的更多信息，请参见 `ipsecconf(1M)` 手册页。
- 有关隧道配置的示例，请参见如何在隧道模式下使用 IPsec 保护两个 LAN 之间的连接 [105]。

虚拟专用网络和 IPsec

术语 virtual private network, VPN（虚拟专用网络）通常用于描述建立在公共网络（例如 Internet）基础上的专用、安全的点对点网络。点对点网络（或 VPN）可用于将专用网络上的系统或专用网络上的系统网络连接在一起。

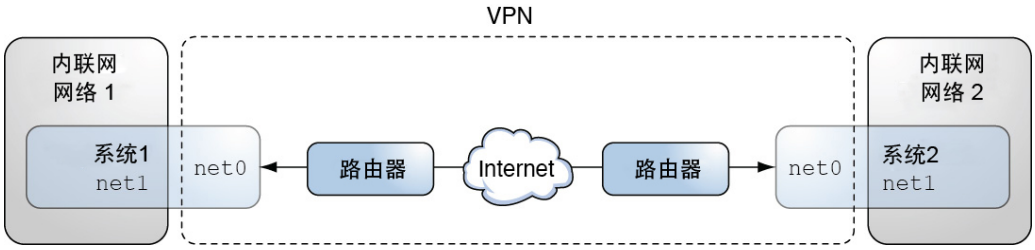
已配置的 tunnel（隧道）是点对点接口。使用隧道，可以将一个 IP 包封装到另一个 IP 包中。正确配置的隧道同时要求隧道源和隧道目标。有关更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道 》中的“如何创建和配置 IP 隧道”。

隧道会创建一个明显的 physical interface（物理接口）连接至 IP。经过 IP 隧道接口的 IP 通信可以使用 IPsec 进行保护。

Oracle Solaris 中的隧道接口可用于封装或通过隧道传输系统间传送的 IP 包。通过隧道传输的包会在原始 IP 头之前添加一个 IP 头。添加的头使用可在公共网络上路由的地址。这些地址由下图中的 `net0` 接口表示。

下图说明了两个站点如何使用 IPsec 在它们之间创建 VPN。`Intranet 1` 和 `Intranet 2` 之间的通信使用 IP-in-ESP 封装并在 Internet 上通过隧道传输。在此示例中，`net0` 地址在外部 IP 头中使用，而内部 IP 地址是来自内联网并通过隧道传输的包的地址。因为内部 IP 地址受 ESP 保护，所以它们在通过 Internet 时会免受检查。

图 6-6 虚拟专用网络



有关设置过程的详细示例，请参见如何在隧道模式下使用 IPsec 保护两个 LAN 之间的连接 [105]。

IPsec 和 FIPS 140

在启用 FIPS 140 的系统中，可以轻松配置 IPsec 以符合 FIPS 140 要求。只能选择 FIPS 140 验证算法用于创建密钥和证书。本指南中的操作过程和示例均使用 FIPS 140 认可的算法，除非明确指定 `any` 算法。

注 - 如果有严格的要求，只能使用 FIPS 140-2 验证的加密，则必须运行 Oracle Solaris 11.1 SRU 5.5 发行版或 Oracle Solaris 11.1 SRU 3 发行版。Oracle 已在这两个特定发行版中针对加密框架完成了 FIPS 140-2 验证。Oracle Solaris 11.2 基于此验证的基础而构建并在性能、功能和可靠性方面进行了软件改进。应当尽可能在 FIPS 140-2 模式下配置 Oracle Solaris 11.2，以利用这些改进。

在 Oracle Solaris 中以 FIPS 140 模式运行时，IPsec 可使用以下得到认可的机制：

- CBC、CCM、GCM 和 GMAC 模式下的 AES，密钥长度为 128 位到 256 位
- 3DES
- SHA1
- SHA2，密钥长度为 256 位和 512 位

有关 Oracle Solaris 通过 FIPS 140 验证的算法的明确列表，请参见 http://www.oracle.com/technetwork/topics/security/140sp2061-2082028.pdf。有关更全面的讨论，请参见《Using a FIPS 140 Enabled System in Oracle Solaris 11.2 》。

IPsec 和 NAT 遍历

IKE 可以通过 NAT 盒 (NAT box) 来协商 IPsec SA。此功能使系统可以从远程网络安全地连接，即使当系统位于 NAT 设备之后也可如此。例如，在家工作或从会议地点登录的雇员可以使用 IPsec 保护其通信。

NAT 盒 (NAT box) 可以将专用内部地址转换为唯一的 Internet 地址。NAT 常见于 Internet 的公共访问点，例如宾馆。

NAT 盒 (NAT box) 位于通信系统之间时使用 IKE 的能力称为“NAT 遍历”，即 NAT-T。NAT-T 具有下列限制：

- 由于 AH 协议取决于未更改的 IP 头，因此 AH 不能与 NAT-T 一起使用。ESP 协议可用于 NAT-T。
- NAT 盒 (NAT box) 不使用特殊的处理规则。使用特殊 IPsec 处理规则的 NAT 盒 (NAT box) 可能会干扰 NAT-T 的实现。
- 仅当 IKE 启动器是位于 NAT 盒 (NAT box) 之后的系统时，NAT-T 才运行。IKE 响应者不能位于 NAT 盒 (NAT box) 之后，除非此盒已经过编程可以将 IKE 包转发到位于盒之后的相应单个系统。

以下 RFC 介绍了 NAT 的功能和 NAT-T 的限制。可以从 http://www.rfc-editor.org 获取 RFC 的副本。

- RFC 3022，"Traditional IP Network Address Translator (Traditional NAT)"，2001 年 1 月
- RFC 3715，"Psec-Network Address Translation (NAT) Compatibility Requirements"，2004 年 3 月

- RFC 3947，"Negotiation of NAT-Traversal in the IKE"，2005 年 1 月
- RFC 3948，"UDP Encapsulation of IPsec Packets"，2005 年 1 月

IPsec 和 SCTP

Oracle Solaris 支持流控制传输协议 (Streams Control Transmission Protocol, SCTP)。支持使用 SCTP 协议和 SCTP 端口号来指定 IPsec 策略，但是这种方法不可靠。RFC 3554 中指定的 SCTP 的 IPsec 扩展尚未实现。这些限制可能会使为 SCTP 创建 IPsec 策略的过程更为复杂。

SCTP 可以在单个 SCTP 关联的上下文中使用多个源地址和目标地址。当 IPsec 策略应用于单个源地址或目标地址时，通信可能会在 SCTP 切换此关联的源地址或目标地址时失败。IPsec 策略仅识别初始地址。有关 SCTP 的信息，请阅读 Stream Control Transmission Protocol (SCTP)（流控制传输协议）RFC。

IPsec 和 Oracle Solaris 区域

区域中支持 IPsec。每个区域都可以有自己的 IPsec 策略和 IKE 配置。区域可被视为单独的主机。

但共享 IP 区域除外，它们没有自己的 IP 栈。对于共享 IP 区域，IPsec 策略和 IKE 配置会在全局区域中执行。共享 IP 区域的 IPsec 策略规则使用分配给该区域的 IP 地址。

有关更多信息，请参见《Oracle Solaris Zones 介绍 》中的第 1 章 “Oracle Solaris Zones 介绍”。

IPsec 和虚拟机

IPsec 可以与虚拟机 (virtual machine, VM) 一起使用。要在 SPARC 系统上创建 VM，请使用 Oracle VM Server。在 x86 系统上，您可以使用 Oracle VM VirtualBox。有关配置的信息，请参见管理指南，了解您的 Oracle VM 的版本。

IPsec 配置命令和文件

表 6-2 “部分 IPsec 配置命令和文件” 介绍了用于配置和管理 IPsec 的文件、命令和服务标识符。为了体现完整性，此表包括密钥管理文件、套接字接口和命令。

有关服务标识符的更多信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。

有关在网络中实现 IPsec 的说明，请参见“使用 IPsec 保护网络通信” [95]。

有关 IPsec 实用程序和文件的更多详细信息，请参见第 12 章 IPsec 和密钥管理参考。

表 6-2 部分 IPsec 配置命令和文件

IPsec 命令、文件或服务	说明	手册页
`svc:/network/ipsec/ipsecalgs`	管理 IPsec 算法的 SMF 服务。	`ipsecalgs(1M)`
`svc:/network/ipsec/manual-key`	手动管理加密 IPsec SA 的 SMF 服务。	`ipseckey(1M)`
`svc:/network/ipsec/policy`	管理 IPsec 策略的 SMF 服务。	`smf(5)`、`ipsecconf(1M)`
`svc:/network/ipsec/ike:ikev2`、`svc:/network/ipsec/ike:default`	使用 IKE 自动管理 IPsec SA 的 SMF 服务实例。	`smf(5)`、`in.ikev2d(1M)`、`in.iked(1M)`
`/etc/inet/ipsecinit.conf` 文件	IPsec 策略文件。 由 SMF `policy` 服务用来在系统引导时配置 IPsec 策略。	`ipsecconf(1M)`
`ipsecconf` 命令	IPsec 策略命令。用于查看和修改当前的 IPsec 策略，以及进行测试。 由 SMF `policy` 服务用来在系统引导时配置 IPsec 策略。	`ipsecconf(1M)`
`PF_KEY` 套接字接口	安全关联数据库 (security associations database, SADB) 的接口。处理手动密钥管理和自动密钥管理。	`pf_key(7P)`
`ipseckey` 命令	IPsec SA 加密命令。`ipseckey` 是 `PF_KEY` 接口的命令行前端。`ipseckey` 可以创建、销毁或修改 SA。	`ipseckey(1M)`
`/etc/inet/secret/ipseckeys` 文件	包含手动加密的 SA。 由 SMF `manual-key` 服务用来在系统引导时手动配置 SA。	
`ipsecalgs` 命令	IPsec 算法命令。可用于查看和修改 IPsec 算法及其属性的列表。 由 SMF `ipsecalgs` 服务用来在系统引导时使已知 IPsec 算法与内核同步。	`ipsecalgs(1M)`
`/etc/inet/ipsecalgs` 文件	包含已配置的 IPsec 机制和算法定义。此文件由 `ipsecalgs` 命令管理，并且决不能手动编辑。	
`/etc/inet/ike/ikev2.config` 文件	IKEv2 配置和策略文件。密钥管理基于此文件中的规则和全局参数。请参见“IKEv2 实用程序和文件” [202]。	`ikev2.config(4)`
`/etc/inet/ike/config` 文件	IKEv1 配置和策略文件。缺省情况下，此文件不存在。密钥管理基于此文件中的规则和全局参数。请参见“IKEv1 实用程序和文件” [205]。 如果此文件存在，`svc:/network/ipsec/ike:default` 服务会启动 IKEv1 守护进程 `in.iked`。	`ike.config(4)`

♦ ♦ ♦ 第 7 章

7 配置 IPsec

本章提供了在网络中实现 IPsec 的过程。这些过程将在以下各节中进行介绍：

- “使用 IPsec 保护网络通信” [95]
- “使用 IPsec 保护 VPN” [101]
- “其他 IPsec 任务” [109]

有关 IPsec 的概述信息，请参见第 6 章 关于 IP 安全体系结构。有关 IPsec 的参考信息，请参见第 12 章 IPsec 和密钥管理参考。

注 - 这些任务假设系统已分配了静态 IP 地址，并且正在运行网络配置文件 `DefaultFixed`。如果 `netadm list` 命令返回 `Automatic`，请参见 `netcfg`(1M) 手册页了解更多信息。

使用 IPsec 保护网络通信

使用本节中的过程可以保护两个系统之间的通信以及保护 Web 服务器。要保护 VPN，请参见“使用 IPsec 保护 VPN” [101]。有关管理 IPsec 以及将 SMF 命令与 IPsec 和 IKE 结合使用的其他过程，请参见“其他 IPsec 任务” [109]。

以下信息适用于所有的 IPsec 配置任务：

- **IPsec 和区域** – 每个系统要么为全局区域，要么为专用 IP 区域。有关更多信息，请参见“IPsec 和 Oracle Solaris 区域” [93]。
- **IPsec 和 FIPS 140 模式** – IPsec 管理员需要负责选择 Oracle Solaris 通过 FIPS 140 验证的算法。本章中的操作过程和示例均使用 FIPS 140 认可的算法，除非明确指定 `any` 算法。
- **IPsec 和 RBAC** – 要使用角色来管理 IPsec，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的第 3 章 “在 Oracle Solaris 中指定权限”。有关示例，请参见如何配置网络安全角色 [111]。
- **IPsec 和 SCTP** – 可以使用 IPsec 来保护流控制传输协议 (Streams Control Transmission Protocol, SCTP) 关联，但使用时必须谨慎。有关更多信息，请参见“IPsec 和 SCTP” [93]。

- IPsec 和 Trusted Extensions 标签 – 在配置有 Oracle Solaris 的 Trusted Extensions 功能的系统上，可以为 IPsec 包添加标签。有关更多信息，请参见《Trusted Extensions 配置和管理 》中的“有标签 IPsec 的管理”。
- IPv4 和 IPv6 地址 – 本指南中的 IPsec 示例使用 IPv4 地址。Oracle Solaris 还支持 IPv6 地址。要为 IPv6 网络配置 IPsec，请将示例中的地址替换为对应的 IPv6 地址。使用 IPsec 保护隧道时，您可以对内部地址和外部地址混用 IPv4 和 IPv6 地址。例如，通过此类配置，可以在 IPv4 网络上以隧道方式传输 IPv6 数据。

以下任务列表列出了在一个或多个系统之间设置 IPsec 的过程。`ipsecconf(1M)`、`ipseckey(1M)` 和 `ipadm(1M)` 手册页也在各自的 "Examples"（示例）部分介绍了有用的过程。

表 7-1 使用 IPsec 保护网络通信任务列表

任务	说明	参考
保证两个系统之间的通信安全。	确保系统间传送的包的安全。	如何使用 IPsec 保护两台服务器之间的网络通信 [96]
使用 IPsec 策略保证 Web 服务器的安全。	要求非 Web 通信使用 IPsec。Web 客户机由特定端口识别，这些端口将绕过 IPsec 检查。	如何使用 IPsec 保护 Web 服务器与其他服务器的通信 [99]
使用 IKE 为 IPsec SA 自动创建加密材料。	推荐的创建 IPsec SA 的方法。	“配置 IKEv2” [127] 和“配置 IKEv1” [151]
设置安全的虚拟专用网络 (virtual private network, VPN)。	在 Internet 中的两个系统之间设置 IPsec。	“使用 IPsec 保护 VPN” [101]
设置手动密钥管理。	在不使用 IKE 的情况下为 IPsec SA 提供原始数据。	如何手动创建 IPsec 密钥 [109]

▼ 如何使用 IPsec 保护两台服务器之间的网络通信

假设此过程具有以下设置：

- 系统已分配了静态 IP 地址，并且正在运行网络配置文件 `DefaultFixed`。如果 `netadm list` 命令返回 `Automatic`，请参见 `netcfg(1M)` 手册页了解更多信息。
- 两个系统的名称为 `enigma` 和 `partym`。
- 每个系统都有 IP 地址。该地址可以是 IPv4 地址、IPv6 地址或这两类地址。此过程使用 IPv4 地址。
- 每个系统均为全局区域或专用 IP 区域。有关更多信息，请参见“IPsec 和 Oracle Solaris 区域” [93]。
- 每个系统都使用 AES 算法加密通信，使用 SHA-2 验证通信。

注 - 某些站点可能要求使用 SHA-2 算法。

- 每个系统都使用共享安全关联。

如果使用共享 SA，则仅需要一对 SA 来保护两个系统。

注 - 要在 Trusted Extensions 系统上使用带标签的 IPsec，请参见《Trusted Extensions 配置和管理 》中的“如何在多级别 Trusted Extensions 网络中应用 IPsec 保护”中此过程的扩展。

开始之前 拥有特定权限的用户可以运行以下命令，不必成为 `root`：

- 要运行配置命令，您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。
- 在此管理角色中，您可以使用 `pfedit` 命令编辑与 IPsec 相关的系统文件，并创建密钥。
- 要编辑 `hosts` 文件，您必须承担 `root` 角色或拥有编辑此文件的显式权限。请参见例 7-7 “使可信用户能够配置和管理 IPsec”。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. **在每个系统上，将主机项添加到 `/etc/inet/hosts` 文件中。**

   此步骤可使本地命名服务将系统名称解析为 IP 地址，而不必依赖网络命名服务。

   a. **在名为 `partym` 的系统上，将以下内容键入到 `hosts` 文件中：**

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

   b. **在名为 `enigma` 的系统上，将以下内容键入到 `hosts` 文件中：**

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

2. **在每个系统上，创建 IPsec 策略文件。**

   该文件名为 `/etc/inet/ipsecinit.conf`。有关示例，请参见 `/etc/inet/ipsecinit.sample` 文件。

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

3. **将 IPsec 策略项添加到 `ipsecinit.conf` 文件。**

   有关 IPsec 策略项的语法以及若干示例，请参见 `ipsecconf(1M)` 手册页。

   a. **在 `enigma` 系统上添加以下策略：**

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

由于未使用 `dir` 关键字，所以策略会同时应用于外发包和传入包。

b. **在 `partym` 系统上添加相同的策略：**

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

4. **在每个系统上，将 IKE 配置为管理 IPsec SA。**

按“配置 IKEv2” [127]中的配置过程之一执行操作。有关 IKE 配置文件的语法，请参见 `ikev2.config(4)` 手册页。如果与仅支持 IKEv1 协议的系统通信，请参阅“配置 IKEv1” [151]和 `ike.config(4)` 手册页。

注 - 如果您必须手动生成并维护密钥，请参见如何手动创建 IPsec 密钥 [109]。

---

5. **检验 IPsec 策略文件的语法。**

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

修复任何错误、检验文件的语法，然后继续。

6. **刷新 IPsec 策略。**

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

IPsec 策略缺省情况下处于启用状态，因此要对其进行刷新。如果已禁用了 IPsec 策略，请将其启用。

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

7. **激活 IPsec 密钥。**

- **如果未启用 `ike` 服务，请将其启用。**

注 - 如果与只能运行 IKEv1 协议的系统通信，请指定 `ike:default` 实例。

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

- **如果已启用 `ike` 服务，请重新启动此服务。**

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

如果在步骤 4中手动配置了密钥，请完成过程如何手动创建 IPsec 密钥 [109]以激活密钥。

8. **验证是否对包进行了保护。**

有关过程，请参见如何检验包是否受 IPsec 保护 [115]。

例 7-1 通过使用 ssh 连接远程配置 IPsec 策略

在此示例中，root 角色的管理员通过使用 ssh 命令访问第二个系统，在两个系统上配置 IPsec 策略和密钥。管理员在两个系统上具有相同的定义。有关更多信息，请参见 ssh(1) 手册页。

1. 管理员通过执行如何使用 IPsec 保护两台服务器之间的网络通信 [96]的步骤 1 至步骤 5 来配置第一个系统。
2. 在不同的终端窗口中，管理员使用定义相同的用户名和 ID 通过 ssh 命令远程登录。

```
local-system % ssh -l jdoe other-system
other-system # su - root
Enter password: xxxxxxxx
other-system #
```

3. 在 ssh 会话的终端窗口中，管理员通过完成步骤 1 至步骤 7 来配置第二个系统的 IPsec 策略和密钥。
4. 管理员结束 ssh 会话。

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

5. 管理员通过完成步骤 6 和步骤 7 在第一个系统上启用 IPsec 策略。

下次这两个系统进行通信（包括使用 ssh 连接）时，此通信将会受 IPsec 保护。

例 7-2 配置 IPsec 策略以在 FIPS 140 模式下运行

在本示例中，管理员在启用了 FIPS 140 的系统上配置了 IPsec 策略，以便遵循要求使用密钥长度至少为 192 位的对称算法的站点安全策略。

管理员指定了两个可能的 IPsec 策略。第一个策略指定 CCM 模式下的 AES 用于加密和验证，第二个策略指定密钥长度为 192 位和 256 位 的 AES 用于加密，SHA384 用于验证。

```
{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}
```

▼ 如何使用 IPsec 保护 Web 服务器与其他服务器的通信

在运行 Web 服务器的系统上可以使用 IPsec 保护除 Web 客户机请求以外的所有通信。受保护的网络通信通常发生在 Web 服务器和其他后端服务器之间。

除了允许 Web 客户机绕过 IPsec 外，此过程中的 IPsec 策略还允许服务器发出 DNS 客户机请求。所有其他通信均受 IPsec 保护。

开始之前 此过程假设您已完成了如何使用 IPsec 保护两台服务器之间的网络通信 [96] 中在两台服务器上配置 IPsec 的步骤，因此以下条件会生效：

- 每个系统均为有固定地址的全局区域或专用 IP 区域。有关更多信息，请参见“IPsec 和 Oracle Solaris 区域” [93]。
- 与 Web 服务器的通信已受 IPsec 保护。
- 生成了加密材料。
- 已检验是否对包进行了保护。

拥有特定权限的用户可以运行这些命令，不必成为 `root`。

- 要运行配置命令，您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。
- 要编辑与 IPsec 相关的系统文件和创建密钥，请使用 `pfedit` 命令。
- 要编辑 `hosts` 文件，您必须承担 `root` 角色或拥有编辑此文件的显式权限。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. **确定哪些服务需要绕过 IPsec 策略检查。**
   对于 Web 服务器，这些服务包括 TCP 端口 80 (HTTP) 和 443 (安全 HTTP)。如果 Web 服务器提供 DNS（域名系统）名称查找，则服务器还可能需要针对 TCP（传输控制协议）和 UDP（用户数据报协议）包括端口 53。

2. **将 Web 服务器策略添加到 IPsec 策略文件。**
   将以下行添加到 `ipsecinit.conf` 文件：

```
# 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}
```

   此配置仅允许安全通信访问系统，绕过检查的例外情况在步骤 1 中进行了介绍。

3. 检验 IPsec 策略文件的语法。

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

4. 刷新 IPsec 策略。

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

5. 刷新 IPsec 的密钥。

   重新启动 `ike` 服务。

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

   注 - 如果与只能运行 IKEv1 协议的系统通信，请指定 `ike:default` 实例。

   如果手动配置了密钥，请按照如何手动创建 IPsec 密钥 [109]中的说明操作。

   您的设置已完成。

6. （可选） 使远程系统与 Web 服务器进行非 Web 通信。

   将以下行添加到远程系统的 `/etc/inet/ipsecinit.conf` 文件：

```
## Communicate with web server about nonweb stuff
##
{raddr webserver} ipsec {encr_algs aes encr_auth_algs sha512 sa shared}
```

   验证语法，然后刷新 IPsec 策略将其激活。

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

   仅当系统的 IPsec 策略匹配时，远程系统才能与 Web 服务器安全地进行非 Web 通信。

7. （可选） 按照匹配项出现的顺序显示 IPsec 策略项，包括每个隧道的项。

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

使用 IPsec 保护 VPN

您可以使用 IPsec 来保护 VPN。有关背景的信息，请参见“IPsec 中的传输模式和隧道模式” [89]。本节中的示例和过程使用 IPv4 地址，但这些示例和过程同样适用于 IPv6 VPN。有关简短介绍，请参见“使用 IPsec 保护网络通信” [95]。

有关适用于隧道模式的 IPsec 策略的示例，请参见“在隧道模式下使用 IPsec 保护 VPN 的示例” [102]。

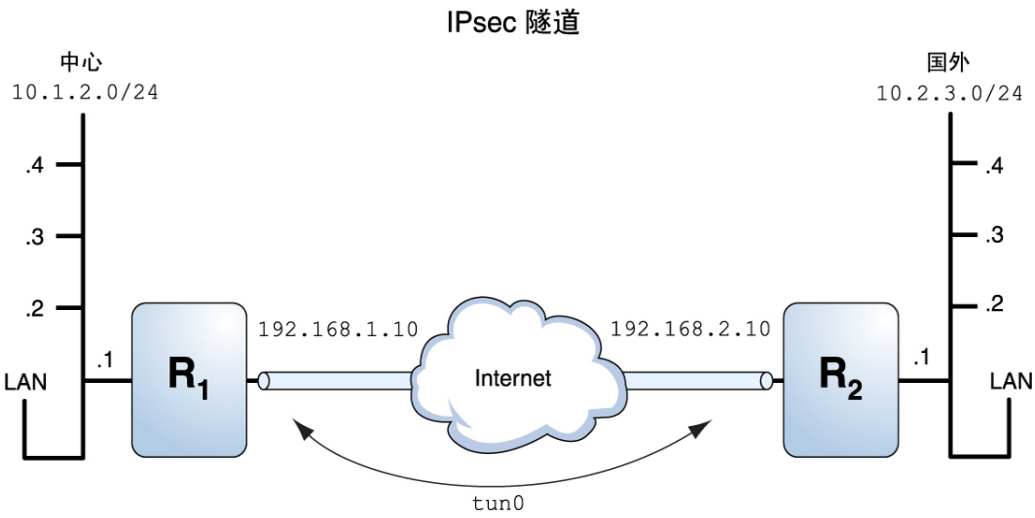
在隧道模式下使用 IPsec 保护 VPN 的示例

LAN 的所有子网都配置了下图中的隧道，如下所示：

```
## 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
```

图 7-1 受 IPsec 保护的隧道



以下示例都基于此图。

例 7-3 创建一个所有子网都可以使用的隧道

在此示例中，来自图 7-1 “受 IPsec 保护的隧道” 中的中心 LAN 的本地 LAN 的所有通信都可以通过隧道从路由器 1 传送到路由器 2，然后再传送到国外 LAN 的所有本地 LAN。通信使用 AES 进行加密。

```
## IPsec policy ##
```

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

例 7-4 创建一个仅连接两个子网的隧道

在此示例中，仅为中心 LAN 的子网 10.1.2.0/24 和国外 LAN 的子网 10.2.3.0/24 之间的通信建立了隧道并对通信进行了加密。在中心 LAN 没有其他 IPsec 策略的情况下，如果中心 LAN 尝试通过此隧道路由其他 LAN 的任何通信，则通信会在路由器 1 处被丢弃。

```
## 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}
```

用于保护 VPN 的 IPsec 任务的网络拓扑说明

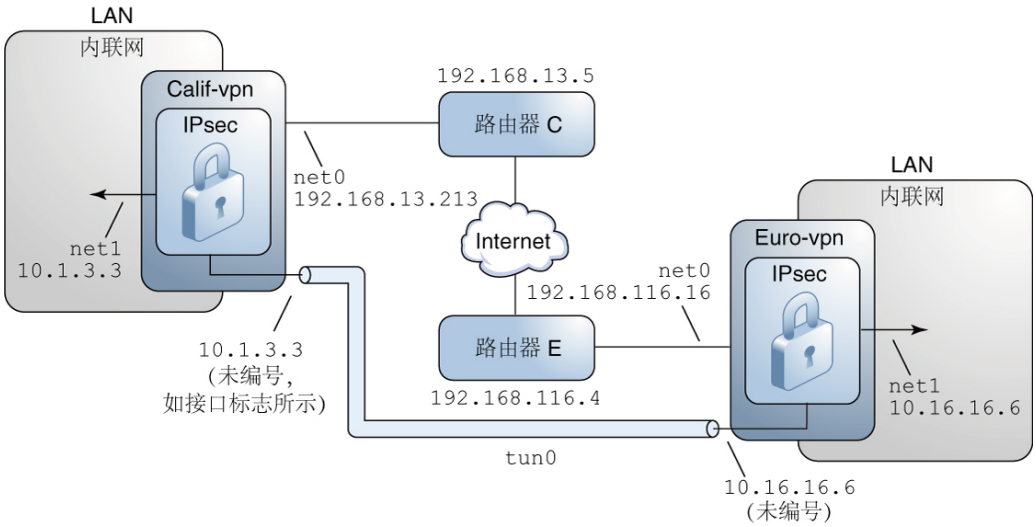
本节中的过程假设您已经进行了以下设置。有关此网络的描述，请参见图 7-2 “通过 Internet 连接的办公室之间的 VPN 样例”。

- 每个系统都使用 IPv4 地址空间。

  这些过程也可以与 IPv6 地址或 IPv4 和 IPv6 地址组合协同使用。
- 每个系统都有两个接口。net0 接口连接到 Internet。在此示例中，Internet IP 地址以 192.168 开始。net1 接口连接到公司的 LAN（即公司的内联网）。在此示例中，内联网 IP 地址以数字 10 开始。
- 每个系统都需要采用 AES 算法的 ESP 加密。AES 算法使用 128 位或 256 位的密钥。
- 每个系统都需要采用 SHA-2 算法的 ESP 验证。在此示例中，SHA-2 算法使用 512 位的密钥。
- 每个系统都可以连接到能直接访问 Internet 的路由器。
- 每个系统都使用共享安全关联。

下图显示了这些过程中使用的配置参数。

图 7-2 通过 Internet 连接的办公室之间的 VPN 样例



下表列出了配置参数。

参数	欧洲	加利福尼亚
系统名	euro-vpn	calif-vpn
系统内联网接口	net1	net1
系统内联网地址，到另一个网络的缺省路由	10.16.16.6	10.1.3.3
系统内联网地址对象	net1/inside	net1/inside
系统 Internet 接口	net0	net0
系统 Internet 地址	192.168.116.16	192.168.13.213
Internet 路由器名称	router-E	router-C
Internet 路由器地址	192.168.116.4	192.168.13.5
隧道名称	tun0	tun0
隧道名称地址对象	tun0/v4tunaddr	tun0/v4tunaddr

有关隧道名称的信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道 》中的“管理 IP 隧道”。有关地址对象的信息，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件 》中的“如何配置 IPv4 接口”和 ipadm(1M) 手册页。

▼ 如何在隧道模式下使用 IPsec 保护两个 LAN 之间的连接

在隧道模式下，内部 IP 包决定保护其内容的 IPsec 策略。

此过程扩展了如何使用 IPsec 保护两台服务器之间的网络通信 [96] 过程。“用于保护 VPN 的 IPsec 任务的网络拓扑说明” [103] 介绍了具体设置。

有关运行特定命令的更详尽的原因说明，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96] 中的相应步骤。

注 - 在两个系统中执行此过程中的步骤。

除了连接两个系统之外，还要连接两个连接到这两个系统的内联网。此过程中的系统作为网关使用。

注 - 要在 Trusted Extensions 系统上在隧道模式下使用带标签的 IPsec，请参见《Trusted Extensions 配置和管理 》中的“如何通过不可信网络配置隧道”中此过程的扩展。

开始之前 每个系统均为全局区域或专用 IP 区域。有关更多信息，请参见“IPsec 和 Oracle Solaris 区域” [93]。

拥有特定权限的用户可以运行这些命令，不必成为 `root`。

- 要运行配置命令，您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。
- 要编辑与 IPsec 相关的系统文件和创建密钥，请使用 `pfedit` 命令。
- 要编辑 `hosts` 文件，您必须承担 `root` 角色或拥有编辑此文件的显式权限。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. 在配置 IPsec 之前控制包流。

   a. 禁用 IP 转发和 IP 动态路由。

```
# routeadm -d ipv4-routing
# ipadm set-prop -p forwarding=off ipv4
# routeadm -u
```

禁用 IP 转发功能可阻止包通过此系统从一个网络转发到另一个网络。有关 `routeadm` 命令的说明，请参见 routeadm(1M) 手册页。

b. 启用 IP 严格多宿主。

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

启用 IP 严格多宿主要求发往系统的目标地址之一的包到达正确的目标地址。

当 `hostmodel` 参数设置为 `strong` 时，到达特定接口的包的地址必须为该接口的本地 IP 地址之一。所有其他包，甚至是传送到系统其他本地地址的包，均被丢弃。

c. 检验是否已禁用大多数网络服务。

验证 `ssh` 服务是否正在运行。

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

2. 将 VPN 的 IPsec 策略添加到 **/etc/inet/ipsecinit.conf** 文件。

有关其他示例，请参见“在隧道模式下使用 IPsec 保护 VPN 的示例” [102]。

在此策略中，本地 LAN 上的系统与网关的内部 IP 地址之间不需要 IPsec 保护，因此将添加 `bypass` 语句。

a. 在 **euro-vpn** 系统上，向 **ipsecinit.conf** 文件添加以下项：

```
# 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. 在 **calif-vpn** 系统上，向 **ipsecinit.conf** 文件添加以下项：

```
# 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. 在每个系统上，配置 IKE 以在两个系统之间添加一对 IPsec SA。

按照“配置 IKEv2” [127] 中的配置过程之一来配置 IKE。有关 IKE 配置文件的语法，请参见 ikev2.config(4) 手册页。如果与仅支持 IKEv1 协议的系统通信，请参阅“配置 IKEv1” [151] 和 ike.config(4) 手册页。

注 - 如果您必须手动生成并维护密钥，请参见如何手动创建 IPsec 密钥 [109]。

4. **检验 IPsec 策略文件的语法。**

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

修复任何错误、检验文件的语法，然后继续。

5. **刷新 IPsec 策略。**

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

IPsec 策略缺省情况下处于启用状态，因此要对其进行刷新。如果已禁用了 IPsec 策略，请将其启用。

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

6. **创建并配置隧道 `tun0`。**

以下命令用于配置内部接口和外部接口、创建 `tun0` 隧道并为该隧道指定 IP 地址。

a. **在 `calif-vpn` 系统上，创建该隧道并进行配置。**

```
# 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
```

第一个命令用于创建 IP 接口 `net1`。第二个命令用于将地址添加到 `net1`。第三个命令用于创建 IP 接口 `tun0`。第四个命令用于添加在隧道链路中封装的 IP 地址。有关更多信息，请参见 `dladm`(1M) 和 `ipadm`(1M) 手册页。

b. **在 `euro-vpn` 系统上，创建该隧道并进行配置。**

```
# 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
```

注 - `ipadm` 命令的 `-T` 选项用于指定要创建的地址类型。`dladm` 命令的 `-T` 选项用于指定该隧道。

有关这些命令的信息，请参见 `dladm`(1M) 和 `ipadm`(1M) 手册页以及《在 Oracle Solaris 11.2 中配置和管理网络组件 》中的“如何配置 IPv4 接口”。有关定制名称

的信息，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件 》中的“Oracle Solaris 中的网络设备和数据链路命名”。

7. 在每个系统上，配置转发。

```
# 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 转发指可以转发来自其他位置的包。IP 转发也指由此接口发出的包可能源于其他位置。要成功转发包，必须启用接收接口和传送接口的 IP 转发功能。

因为 `net1` 接口在内联网内部，所以必须启用 `net1` 的 IP 转发功能。因为 `tun0` 通过 Internet 连接两个系统，所以必须启用 `tun0` 的 IP 转发功能。`net0` 接口已禁用其 IP 转发功能以阻止 Internet 上的外部入侵者向受保护的内联网中注入包。

8. 在每个系统上，禁止公布专用接口。

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

即使 `net0` 禁用 IP 转发功能，路由协议实现仍会通告接口。例如，`in.routed` 协议仍会通告 `net0` 可将包转发到内联网中的对等接口。可以通过设置接口的专用标志，阻止这些通告。

9. 重新启动网络服务。

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

10. 手动添加通过 **net0** 接口实现的缺省路由。

缺省路由必须是可以直接访问 Internet 的路由器。

a. 在 **calif-vpn** 系统上，添加以下路由：

```
# route -p add net default 192.168.13.5
```

b. 在 **euro-vpn** 系统上，添加以下路由：

```
# route -p add net default  192.168.116.4
```

即使 `net0` 接口不是内联网的一部分，`net0` 也需要通过 Internet 访问其同级系统。要找到其同级系统，`net0` 需要有关 Internet 路由的信息。对于 Internet 的其他部分来说，VPN 系统像是一台主机，而不是路由器。因此，您可以使用缺省的路由器或运行路由器搜索协议来查找同级系统。有关更多信息，请参见 `route`(1M) 和 `in.routed`(1M) 手册页。

其他 IPsec 任务

以下任务列表列出了在管理 IPsec 时可能要执行的任务。

表 7-2 其他 IPsec 任务的任务列表

任务	说明	参考
手动创建或替换 IPsec SA。	为 IPsec SA 提供原始数据：	如何手动创建 IPsec 密钥 [109]
创建网络安全角色。	创建可以设置安全网络，但权限级别低于 root 角色的角色。	如何配置网络安全角色 [111]
创建一个可以处理所有网络管理任务的权限配置文件。	创建可以执行网络管理，但权限级别低于 root 角色的角色。	例 7-7 “使可信用户能够配置和管理 IPsec”
检查 IPsec 是否正在保护包。	检查 snoop 输出以了解指示如何保护 IP 包的特定头。	如何检验包是否受 IPsec 保护 [115]
将 IPsec 和加密材料作为一组 SMF 服务来管理。	启用、禁用、刷新和重新启动服务。此外也可以更改服务的属性值。	“查看 IPsec 和手动密钥服务属性” [189]

▼ 如何手动创建 IPsec 密钥

以下过程提供了 IPsec 密钥，以便在不只使用 IKE 进行密钥管理时使用。

使用 ipseckey 命令添加的 IPsec SA 在系统重新引导后不会保留。对于持久性的 IPsec SA，请将项添加到 /etc/inet/secret/ipseckeys 文件。

注意 - 如果必须使用手动加密，请谨慎操作，以确保生成的密钥安全可靠。这些都是用于保证数据安全的实际密钥。

开始之前 必须位于全局区域中才能手动管理共享 IP 区域中的加密材料。对于专用 IP 区域，请在此专用 IP 区域中配置加密材料。

您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **生成 IPsec SA 的密钥。**

   这些密钥必须支持 ipsecinit.conf 文件中的特定策略。例如，可以使用如何使用 IPsec 保护两台服务器之间的网络通信 [96] 中的策略：

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

   此策略使用 AES 和 SHA-2 算法。

   a. **确定所需的密钥。**

您需要生成 aes 的密钥、sha512 的密钥以及 SA 的 security parameter index, SPI（安全参数索引）：

- 两个作为 SPI 值的十六进制随机数。一个数字用于外发通信，一个数字用于传入通信。每个数字的长度最大可以为八个字符。
- 两个用于 SHA-2 验证算法的十六进制随机数。每个数字的长度都必须为 512 个字符。一个数字用于 dst enigma，另一个数字用于 dst partym。
- 两个用于 AES 加密算法的十六进制随机数。每个数字的长度都必须为 128 个字符。一个数字用于 dst enigma，另一个数字用于 dst partym。

注 - ipsecalgs -l 命令会显示算法的密钥大小。使用手动密钥（即，使用 SHA512 和 AES 算法）时，请遵循此过程。请勿将弱算法、组合模式算法或 GMAC 算法用于手动密钥。

---

b. 生成所需的密钥。

- 如果您的站点上有随机数生成器，请使用此生成器。
- 使用 pktool 命令，如《在 Oracle Solaris 11.2 中管理加密和证书 》中的“如何使用 pktool 命令生成对称密钥”和该节中的 IPsec 示例所示。

2. 将密钥添加到 IPsec 的手动密钥文件。

a. 编辑 **enigma** 系统上的 **/etc/inet/secret/ipseckeys** 文件，使其显示与以下内容类似：

```
## 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. 使用只读权限保护该文件。

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

如果使用 `pfedit -s` 命令创建了 `ipseckeys` 文件，则权限设置正确。有关更多信息，请参见 `pfedit(1M)` 手册页。

c. 验证文件的语法。

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

注 - 两个系统上的密钥必须完全相同。

3. 激活 IPsec 密钥。

- 如果未启用 **manual-key** 服务，请将其启用。

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

- 如果已启用 **manual-key** 服务，请刷新此服务。

```
# svcadm refresh ipsec/manual-key
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何配置网络安全角色

如果使用 Oracle Solaris 的权限功能管理系统，请使用此过程提供网络管理角色或网络安全角色。

开始之前 您必须承担 `root` 角色才能创建和指定角色。一般用户可以列出并查看可用权限配置文件的内容。

1. 列出与网络相关的可用权限配置文件。

```
% 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（网络管理）配置文件是 System Administrator（系统管理员）配置文件的补充配置文件。如果您将 System Administrator（系统管理员）权限配置文件纳入角色，则此角色可以执行 Network Management（网络管理）配置文件中的命令。

2. 列出 Network Management（网络管理）权限配置文件中的命令。

```
% 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. **确定网络安全角色在站点中的作用范围。**

   使用步骤 1 中的权限配置文件定义指导您做出决定。

   - 要创建处理所有网络安全的角色，请使用 Network Security（网络安全）权限配置文件。
   - 要创建只处理 IPsec 和 IKE 的角色，请使用 Network IPsec Management（网络 IPsec 管理）权限配置文件。
   - 要创建处理网络管理和安全的角色，除 "Network Management"（网络管理）配置文件以外，请使用 "Network Security"（网络安全）或 "Network IPsec Management"（网络 IPsec 管理）权限配置文件。

4. **创建角色，并将角色分配给一个或多个用户。**

   有关步骤，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“创建角色”和例 7-7 “使可信用户能够配置和管理 IPsec”。

例 7-5 创建并分配网络管理和安全角色

在本示例中，管理员向一个角色分配两个权限配置文件，即 "Network Management"（网络管理）和 "Network Security"（网络安全）。然后，管理员将角色分配给可信用户。

```
# 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
```

在 jdoe 承担 netmgtsec 角色后，配置文件中的权限可供用户 jdoe 使用。

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

例 7-6 在角色之间划分网络安全职责

在此示例中，管理员要在两个角色之间划分网络安全职责。其中一个角色负责管理 Wifi 和链路安全，另一个角色负责管理 IPsec 和 IKE。为每个角色指定三个人，一人一班。

管理员创建的角色如下：

1. 管理员将第一个角色命名为 LinkWifi。
2. 管理员将 Network Wifi（网络 Wifi）、Network Link Security（网络链路安全）和 Network Management（网络管理）权限配置文件指定给该角色。
3. 管理员将 LinkWifi 角色分配给适当的用户。
4. 管理员将第二个角色命名为 IPsec Administrator。
5. 管理员将 Network IPsec Management（网络 IPsec 管理）和 Network Management（网络管理）权限配置文件指定给该角色。
6. 管理员将 "IPsec Administrator"（IPsec 管理员）角色分配给适当的用户。

例 7-7 使可信用户能够配置和管理 IPsec

在本示例中，管理员让一个用户负责配置和管理 IPsec。

除了 "Network Management"（网络管理）和 "IPsec Network Management"（IPsec 网络管理）权限配置文件，管理员还赋予用户编辑 hosts 文件的能力以及读取日志的能力。

1. 管理员创建两个权限配置文件，一个用于编辑文件，另一个用于读取日志。

```
# 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
```

通过权限配置文件，用户可以使用 `more`、`tail` 和 `page` 命令读取日志。`cat` 和 `head` 命令无法使用。

2. 管理员创建权限配置文件，允许用户执行 IPsec 及其加密服务的所有配置和管理任务。

```
# 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. 管理员为配置文件创建一个角色，为角色分配一个口令，然后将角色分配给了解网络和安全的可信用户。

```
# 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. 在带外，管理员为 `jdoe` 提供角色口令。

▼ 如何检验包是否受 IPsec 保护

要检验包是否受到保护，请使用 snoop 命令来测试连接。以下前缀可以在 snoop 输出中显示：

- AH: 前缀指明 AH 正在保护头。如果已使用 auth_alg 来保护通信，则会看到此前缀。
- ESP: 前缀指明正在发送加密数据。如果已使用 encr_auth_alg 或 encr_alg 来保护通信，则会看到此前缀。

开始之前 必须可以同时访问两个系统才能测试连接。

您必须承担 root 角色才能创建 snoop 输出。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **在一个系统上，例如 partym，承担 root 角色。**

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

2. **（可选） 显示 SA 的详细信息。**

```
# ipseckey dump
```

此输出指示哪些 SPI 值与使用的 SA 匹配、使用了哪些算法、密钥等等。

3. **在此系统上，准备搜寻来自远程系统的包。**
在 partym 上的一个终端窗口中，从 enigma 系统搜寻包。

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

4. **从远程系统发送包。**
在另一个终端窗口中，远程登录到 enigma 系统。提供您的口令。然后，承担 root 角色并将包从 enigma 系统发送到 partym 系统。包应该由 snoop -v enigma 命令捕获。

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

5. **检查 snoop 输出。**

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

您也可以将 snoop 输出装入 Wireshark 应用程序。有关更多信息，请参见如何准备 IPsec 和 IKE 系统以便开展故障排除 [181]和“snoop 命令和 IPsec” [200]。

在此文件中，您应该在初始 IP 头信息之后看到包括 AH 和 ESP 信息的输出。类似以下内容的 AH 和 ESP 信息表明包正在受到保护：

```
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 -----
...
```

♦ ♦ ♦ 第 8 章

关于 Internet 密钥交换

Internet 密钥交换 (Internet Key Exchange, IKE) 自动进行 IPsec（Internet 协议安全）的密钥管理。本章包含有关 IKE 的以下信息：

有关实现最新版本 IKE 协议的说明，请参见第 9 章 配置 IKEv2。要继续使用 IKEv1，请参见第 10 章 配置 IKEv1。有关参考信息，请参见第 12 章 IPsec 和密钥管理参考。有关 IPsec 的信息，请参见第 6 章 关于 IP 安全体系结构。

IKE 介绍

对 IPsec 安全关联 (security association, SA) 的加密材料进行的管理称为密钥管理。自动密钥管理需要使用一个安全信道来创建、验证和交换密钥。Oracle Solaris 使用 Internet 密钥交换 (Internet Key Exchange, IKE) 自动管理密钥。IKE 消除了手动分发密钥的管理开销和安全风险。

IKE 可以利用可用的硬件加密加速和密钥存储。硬件加密加速器可以分流系统中的 CPU 密集型密钥操作。硬件上的密钥存储提供了额外的一层保护。

Oracle Solaris 支持两个版本的 IKE 协议。

- IKE 版本 2 (IKEv2)，基于 Internet Key Exchange Protocol Version 2 (IKEv2), RFC 5996（Internet 密钥交换协议版本 2，RFC 5996）
- IKE 版本 1 (IKEv1)，基于 The Internet Key Exchange (IKE), RFC 2409（Internet 密钥交换，RFC 2409）

IKE 概念和术语

以下概念和术语是两个版本的 IKE 通用的。它们可能在两个版本中以不同的方式实现。

- **密钥协商和交换** – 以安全的方式交换加密材料和验证对等方身份。此流程使用非对称加密算法。两种主要方法是 RSA 和 Diffie-Hellman 协议。

  IKE 在运行 IKE 守护进程的系统之间创建和管理 IPsec SA。IKE 协商一个安全信道，保护加密材料的传输。IKE 守护进程使用 `/dev/random` 设备从随机数生成器创建密钥。该守护进程按可配置的速率更改密钥。加密材料可供在 IPsec 策略的配置文件 `ipsecinit.conf` 中指定的算法使用。
- **Diffie-Hellman (DH) 算法** – 一种密钥交换算法，允许两个系统在不安全的信道上安全地生成一个共享密钥。
- **RSA 算法** – 一种非对称密钥算法，通常通过提供 X.509 证书的所有权来验证对等方系统的身份。此算法以其三个创建者（Rivest、Shamir 和 Adleman）命名。

  另外，DSA 或 ECDSA 算法也可以用于此目的。
- **完全正向保密 (perfect forward secrecy, PFS)** – 在 PFS 中，用于保护数据传输的密钥不用于派生其他密钥。此外，也不能使用保护数据传输的密钥的源派生其他密钥。因此，PFS 可防止解密以前记录的通信。
- **Oakley 组** – 用于协商 PFS。请参见 The Internet Key Exchange (IKE)（Internet 密钥交换）RFC 的第 6 节。
- **IKE 策略** – 一套 IKE 规则，用于定义 IKE 守护进程在尝试与对等方系统建立安全密钥交换信道时使用的可接受的参数。这在 IKEv2 中称作 IKE SA，或者在 IKEv1 中称作阶段 1。

  这些参数包括算法、密钥大小、Oakley 组和验证方法。Oracle Solaris IKE 守护进程支持将预共享密钥和证书用作验证方法。

IKE 的工作原理

系统正在运行 IKE 守护进程时，可以协商在它与另一个正在运行 IKE 守护进程的系统之间创建安全关联 (security association, SA) 所需的参数。用于协商此 SA 和后续 IPsec SA 的协议称为 IKE。此版本 Oracle Solaris 支持 IKE 协议的版本 1 (IKEv1) 和版本 2 (IKEv2)。

IKE 安全关联（在 IKEv1 中也称为 ISAKMP 或阶段 1 SA）会保护这两个 IKE 系统之间的进一步协议交换。这些交换会协商加密算法、IPsec 策略以及创建 IPsec SA 所需的其他参数。

正在运行 IKE 守护进程的系统也可以配置为代表其他系统协商 IPsec SA。按这种方式配置时，系统被称为安全网关。如果 IKE 协商成功，IPsec SA 可以用于保护网络包。

注 - 在 Oracle Solaris 11.2 中，IKEv2 使用加密框架中的加密算法，这些算法通过了第 1级 FIPS 140-2 验证，但 IKEv1 未使用这些算法。缺省情况下 FIPS 140 未启用。要比较这两个版本的功能，请参见“比较 IKEv2 和 IKEv1” [121]。要启用 FIPS 140-2 模式，请参见《在 Oracle Solaris 11.2 中管理加密和证书 》中的“如何创建启用了 FIPS 140 的引导环境”。

如果有严格的要求，只能使用 FIPS 140-2 验证的加密，则必须运行 Oracle Solaris 11.1 SRU 5.5 发行版或 Oracle Solaris 11.1 SRU 3 发行版。Oracle 已在这两个特定发行版中针对加密框架完成了 FIPS 140-2 验证。Oracle Solaris 11.2 基于此验证的基础而构建并在性能、功能和可靠性方面进行了软件改进。应当尽可能在 FIPS 140-2 模式下配置 Oracle Solaris 11.2，以利用这些改进。

为创建 IKE SA 而协商的参数包括保护 IKE 交换和某些验证材料的加密算法。验证材料用于确定包含 IKE 协议交换的包是否可信。可信意味着包来自可信的系统，而不是伪装成可信系统的系统。

Oracle Solaris 支持两种类型的 IKE 验证材料，即预先共享的密钥和公钥证书。

使用预先共享的密钥验证的 IKE

预先共享的密钥是十六进制或 ASCII 字符串，只有两个 IKE 系统知悉。这些密钥被称为预先共享的密钥，因为两个端点必须在 IKE 交换之前知道密钥值。此密钥必须是两个系统上的 IKE 配置的一部分。预先共享的密钥用于生成 IKE 有效负荷，而这些有效负荷构成了实现 IKE 协议的包。处理这些 IKE 有效负荷的系统使用同一密钥验证它收到的有效负荷。

预先共享的密钥不能使用 IKE 协议在 IKE 端点之间交换。通常，此密钥通过不同的媒介（例如，打电话）与对等方系统共享。

使用此验证方法的对等方系统上的预先共享密钥必须相同。密钥存储在每个系统上的某个文件中。

IKE，使用公钥证书

公钥证书及其信任链会提供一个以数字方式标识系统的机制，无需手动交换任何机密信息。因此，公钥证书比预先共享的密钥更加安全。

公钥证书是一个对公钥值进行编码的数据 blob，含有关于生成证书的信息，例如名称和签名人、证书的散列或校验和以及散列的数字签名。这些值共同构成了证书。数字签名可以确保证书不被修改。

公钥是以数学方式从另一个称为私钥的值派生出的值。由于采用了从私钥派生公钥的数学算法，这就使得根据公钥检索私钥变得不现实。因此，公钥证书可以自由共享。用于派生公钥的算法示例包括 RSA 和椭圆曲线算法。

数字签名是通过 RSA、DSA 或 ECDSA 等数字签名算法传递证书内容的结果。这些算法使用私有签名密钥（不是证书的一部分）生成数字签名。签名会附加到证书上。同样，根据证书内容和签名计算签名密钥也不现实。更确切的说，证书签名以及证书内容可以使用派生自签名密钥的公钥值轻松验证。

证书可以自签名，在这种情况下，证书签名可由证书的公钥验证，或者它可以由另一个实体签名。当另一个实体对证书签名时，用于验证证书的公钥值也作为公钥证书分发。第二个证书将由可信的 certificate authority, CA（证书颁发机构）或中间方签名。中间方最终为签名实体（即根 CA 或 trust anchor（信任锚））所信任。

这些公钥证书组件以及实现它们的过程和结构通常称为公钥基础结构 (public key infrastructure, PKI)。组织的 PKI 范围可能各不相同。简单的 PKI 可能由对本地使用的几个证书加以签名的 CA 构成。更加广泛的 PKI 可能使用全球公认的信任锚作为权威 CA。

在 IKE 中使用公钥证书

本节介绍在 IKE 中创建和使用公钥证书的总体步骤。有关具体过程，请参见“使用预先共享的密钥配置 IKEv2” [128] 和“使用预先共享的密钥配置 IKEv1” [152]。

1. 要使用自签名证书或证书颁发机构 (certificate authority, CA) 颁发的证书，您首先要生成一个公钥/私钥对。
   - 对于自签名证书，IKE 对等方接着可以交换这些证书，在带外验证证书的真伪，然后将对等方的证书导入本地密钥库。接下来，密钥库包含原始自签名证书和导入的证书。
   - 对于 CA 颁发的证书，您可以执行更多步骤。生成公钥/私钥对时，您也可以生成证书签名请求 (certificate signing request, CSR)。CSR 包含公钥和标识符。典型的标识符是一个 distinguished name, DN（标识名），例如：

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

提示 - 创建一个 DN 或其他尽可能具体的标识符，降低与另一个证书的标识符匹配的可能性。

---

2. 将 CSR 发送到 CA 以获得签名。

   在典型的流程中，您可以将 CSR 粘贴到 Web 表单中，然后将表单提交给 CA。CA 可能会向您发送多个已签名证书。
3. 获得 CA 提供的已签名证书，然后将它们导入 IKEv2 密钥库或 IKEv1 数据库。

   您必须导入 CA 发送的全部证书。这些证书由一个从可信锚或根 CA 到您单独识别的已签名证书的“信任链”构成。
4. 在 IKE 对等方上重复此流程。
5. 使用 IKE 规则中的证书。

您可以通过标识符指定证书，例如 DN。对于 CA 签名的证书，您可以配置 IKE，使其接受由特定 CA 签名的证书。

处理已撤销的证书

已签名证书被视为可信的有效证书，因为签名机构可以保证其有效性。如果证书受已泄密或者被确定为无效，CA 将撤销此证书。

CA 负责维护一个已撤销证书列表，通常称作 certificate revocation list, CRL（证书撤销列表）。您可以使用联机证书状态协议 (Online Certificate Status Protocol, OCSP) 动态查看证书状态。有些公钥证书中嵌入了 URI。它们会标识可以检查 CRL 的 Web 位置或者 OCSP 服务器的 Web 位置。

有关更多信息，请参见 RFC 2459: Certificate and CRL Profile（RFC 2459：证书和 CRL 配置文件）和 RFC 2560: Online Certificate Status Protocol - OCSP（RFC 2560：联机证书状态协议 – OCSP）。

在使用公共证书的系统上协调时间

公钥证书包含证书的颁发日期和时间以及有效期。因此，生成和使用证书的系统上的时钟必须精准。网络时间协议 (Network Time Protocol, NTP) 软件可以用于同步系统上的时钟。Oracle Solaris 发行版中包括了由美国特拉华大学开发的 NTP 公共域软件。相关文档可从 NTP Documentation（NTP 文档）网站获取。您也可以安装 `service/network/ptp` 软件包，配置精确时间协议 (Precision Time Protocol, PTP) 服务。请参见 IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems（IEEE 1588 网络测量和控制系统的精密时钟同步协议标准）。

比较 IKEv2 和 IKEv1

下表比较了 Oracle Solaris 系统上的 IKEv2 和 IKEv1 版本实现。

表 8-1 Oracle Solaris 中的 IKEv2 和 IKEv1 实现

功能	IKEv2	IKEv1
证书 chain of trust（信任链）	隐式，基于密钥库中的对象	`ike/config` 文件中的 `cert_trust` 参数
证书创建	`ikev2cert` 命令	`ikecert certlocal` 命令
证书导入	`ikev2cert import` 命令可以将证书和密钥导入至 PKCS #11 密钥库	`ikecert certdb` 命令可以将独立的证书导入至 IKE 密钥库

功能	IKEv2	IKEv1
证书所有者	`ikeuser`	`root`
证书策略文件	`kmf-policy.xml`	`ike/config` 文件中的一些策略
证书存储	PKCS #11 softtoken 库	本地 IKEv1 数据库
配置文件目录	`/etc/inet/ike/`	`/etc/inet/ike/` 和 `/etc/inet/secret/`
配置所有者	`ikeuser` 帐户	`root` 帐户
守护进程	`in.ikev2d`	`in.iked`
适用于守护进程之间通信的 FIPS 140 算法 †	IKE SA 使用加密框架	不是所有交换都使用加密框架
适用于 IPsec 通信 的 FIPS 140 算法†	使用加密框架	使用加密框架
IKE 策略文件	`ike/ikev2.config`	`ike/config`
IKE 预先共享的密钥	`ike/ikev2.preshared`	`secret/ike.preshared`
NAT 端口	UDP 端口 `4500`	UDP 端口 `4500`
端口	UDP 端口 `500`	UDP 端口 `500`
权限配置文件	网络 IPsec 管理	网络 IPsec 管理
服务名称 (FMRI)	`svc:/ipsec/ike:ikev2`	`svc:/ipsec/ike:default`

†Oracle Solaris 11.1 SRU 5.5 和 SRU 3 的加密框架功能通过了第 1 级 FIPS 140-2 验证。如果启用了 FIPS 140 模式并且正在使用加密框架，则会使用 FIPS 140 验证算法。缺省情况下 FIPS 140 模式未启用。

IKEv2 协议

本节介绍 IKEv2 实现。有关 IKEv1 信息，请参见“IKEv1 协议” [123]。有关比较，请参见“比较 IKEv2 和 IKEv1” [121]。有关适用于两个协议的信息，请参见“IKE 介绍” [117]。Oracle Solaris 同时支持两个版本的 IKE 协议。

IKEv2 守护进程 `in.ikev2d` 可以协商和验证 IPsec SA 的加密材料。请参见 `in.ikev2d(1M)` 手册页。

IKEv2 配置选择

`/etc/inet/ike/ikev2.config` 配置文件包含 `in.ikev2d` 守护进程的配置。配置由若干条规则构成。每项均包含此系统可以与采用类似配置的 IKEv2 对等方一起使用的参数，例如算法和验证数据。

`in.ikev2d` 守护进程支持使用预先共享的密钥 (preshared key, PSK) 和公钥证书来验证身份。

`ikev2.config(4)` 手册页提供了规则样例。每条规则必须都有唯一标签。下表列出了手册页中规则样例的描述性标签：

- 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

注 - 预先共享的密钥可以与许多类型的对等方 ID 一起使用，包括 IP 地址、DN、FQDN 和电子邮件地址。

IKEv2 公共证书策略

`kmf-policy.xml` 文件包含 IKEv2 的证书验证策略。`kmfcfg dbfile=/etc/inet/ike/kmf-policy.xml policy=default` 命令用于修改证书验证策略。典型的修改包括使用 OCSP 和 CRL 以及证书验证期间的网络超时持续时间。另外，使用此策略时，管理员能够修改证书验证的各个方面，例如验证日期执行和密钥使用要求。建议不要放松证书验证的缺省要求。

IKEv1 协议

以下部分对 IKEv1 进行了概述。IKEv1 已经被 IKEv2 取代，后者可以提供更快的安全密钥管理。有关 IKEv2 的信息，请参见“IKEv2 协议” [122]。有关比较，请参见“比较 IKEv2 和 IKEv1” [121]。有关两个协议的通用信息，请参见“IKE 介绍” [117]。IKEv1 和 IKEv2 可以同时运行，与其他系统上的对等方协议进行协商。

IKEv1 密钥协商

IKEv1 守护进程 `in.iked` 能够以安全的方式协商密钥并验证 IPsec SA。IKEv1 提供完全正向保密 (perfect forward secrecy, PFS)。在 PFS 中，不能使用保护数据传输的密钥派生其他密钥。此外，不重新使用用于创建数据传输密钥的种子。请参见 `in.iked(1M)` 手册页。

IKEv1 阶段 1 交换

IKEv1 协议有两个阶段。Oracle Solaris 支持主要模式阶段 1 交换。主要模式交换会协商可接受的参数，以便在两个对等方之间创建 ISAKMP 安全关联 (security association, SA)。此 ISAKMP SA 使用非对称加密交换其加密材料，并使用预先共享的密钥或公钥证书验证其对等方。与 IPsec SA 不同，ISAKMP SA 是双向的，因此只需要一个安全关联。

IKEv1 在阶段 1 交换中协商 ISAKAMP SA 的方式可以配置。IKEv1 会从 `/etc/inet/ike/config` 文件读取配置信息。配置信息包括：

- 全局参数，如公钥证书的名称
- 是否需要使用完全正向保密 (perfect forward secrecy, PFS)
- 此系统的 IKE 对等方
- 保护阶段 1 交换的算法
- 验证方法

  两种验证方法是预先共享的密钥和公钥证书。公钥证书可以自签名，或者由 certificate authority, CA（证书颁发机构）颁发。

有关更多信息，请参见 `ike.config(4)` 手册页。

IKEv1 阶段 2 交换

阶段 2 交换称为快速模式。快速模式交换会协商创建 IPsec SA 所需的 IPsec 算法和加密材料。此交换由在阶段 1 中协商的 ISAKMP SA 保护（加密）。

快速模式交换中的算法和安全协议来自 IPsec 策略文件 `/etc/inet/ipsecinit.conf`。

到期后，IPsec SA 需要重设密钥。创建 IPsec SA 时，SA 的生命周期由 `in.iked` 守护进程设置。此值可配置。

有关更多信息，请参见 `ipsecconf(1M)` 和 `in.iked(1M)` 手册页。

IKEv1 配置选择

`/etc/inet/ike/config` 配置文件包含 `in.iked` 守护进程的配置。配置由若干条规则构成。每项均包含此系统可以与采用类似配置的 IKEv1 对等方一起使用的参数，例如算法和验证数据。`in.iked` 守护进程支持使用预先共享的密钥和公钥证书来验证身份。

项 `auth_method preshared` 指示使用预先共享的密钥。除 `preshared` 之外的 `auth_method` 值指示要使用公钥证书。

在 IKEv1 中，预先共享的密钥与特定的 IP 地址或地址范围关联。这些密钥放置在每个系统上的 /etc/inet/secret/ike.preshared 文件中。

有关更多信息，请参见“IKE 的工作原理” [118]，以及 ike.config(4) 和 ike.preshared(4) 手册页。

♦ ♦ ♦ 第 9 章

9 配置 IKEv2

本章介绍如何为系统配置 Internet 密钥交换版本 2 (IKEv2)。配置和启用 IKEv2 后，它会自动为其指定的 IPsec 端点生成加密材料。本章包含以下信息：

有关 IKE 的概述信息，请参见第 8 章 关于 Internet 密钥交换。有关 IKE 的参考信息，请参见第 12 章 IPsec 和密钥管理参考。有关更多过程，请参见 `ikeadm(1M)`、`pktool(1)`、`ikev2cert(1M)`、`ikev2.config(4)`、`in.ikev2d(1M)` 和 `kmfcfg(1)` 手册页中的示例。

配置 IKEv2

可以使用预先共享的密钥、自签名证书和证书颁发机构 (certificate authority, CA) 颁发的证书来验证 IKE。规则将特定的验证方法与受保护的端点相关联。因此，可以在系统上使用一种或所有验证方法。也可以在 IKEv2 系统上运行 IKEv1。通常，可以运行 IKEv1 以保护与不支持 IKEv2 的系统之间的通信。IKEv2 也可以使用 PKCS #11 硬件令牌来存储密钥和证书。

注 - 这些任务假设系统已分配了静态 IP 地址，并且正在运行网络配置文件 `DefaultFixed`。如果 `netadm list` 命令返回 `Automatic`，请参见 `netcfg(1M)` 手册页了解更多信息。

配置 IKEv2 后，请完成第 7 章 配置 IPsec 中使用这些 IKEv2 规则管理其密钥的 IPsec 过程。以下部分重点介绍特定的 IKEv2 配置。

使用预先共享的密钥配置 IKEv2

如果您要将对等方系统或子网配置为使用 IKEv2，而且您是这些子网的管理员，则使用预先共享的密钥是一个不错的选择。预先共享的密钥也可以在测试时使用。有关更多信息，请参见“使用预先共享的密钥验证的 IKE” [119]。

▼ 如何使用预先共享的密钥配置 IKEv2

在此过程中，请用您的系统名称替换名称 `enigma` 和 `partym`。您可以配置两个 IKE 端点。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. **在每个系统上，编辑 `/etc/inet/ike/ikev2.config` 文件。**

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

2. **在此文件中，创建使用预先共享密钥的规则。**

注 - 您将在步骤 4 中创建密钥。

此文件中的规则和全局参数必须管理系统 `ipsecinit.conf` 文件中 IPsec 策略中的密钥。以下 IKEv2 配置示例会管理如何使用 IPsec 保护两台服务器之间的网络通信 [96] 中 `ipsecinit.conf` 示例的密钥。

a. **例如，在 `enigma` 系统上修改 `ikev2.config` 文件：**

注 - 以下示例显示了全局参数部分的两处修改。您可以使用这两处修改中的任意一处配置对等方。若要使用特定的修改，请将此修改包含在规则中。

```
### 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. 在 **partym** 系统上修改 **ikev2.config** 文件：

```
## 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. 在每个系统上，验证该文件的语法。

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

4. 将预先共享的密钥放置在每个系统上的 **/etc/inet/ike/ikev2.preshared** 文件中。

注意 - 此文件拥有特殊权限，归 ikeuser 所有。切勿删除或替换此文件。相反，请使用 pfedit 命令编辑其内容，确保文件保留其原始属性。

a. 例如，在 **enigma** 系统上，**ikev2.preshared** 文件的显示与以下内容类似：

```
# 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"
}
```

有关 `pfedit` 命令的选项的信息，请参见 pfedit(1M) 手册页。

b. **在 `partym` 系统上，`ikev2.preshared` 文件具有类似的内容，但其唯一标签除外：**

```
## 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. 启用 IKEv2 服务实例。

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

替换预先共享的密钥时，请在对等方系统上编辑预先共享密钥文件，重新启动 `ikev2` 服务。

```
# svcadm restart ikev2
```

例 9-1 使用不同的本地和远程 IKEv2 预先共享密钥

在本示例中，IKEv2 管理员为每个系统创建一个预先共享的密钥，交换密钥，并将每个密钥添加到预先共享密钥文件中。预先共享密钥条目的标签与 `ikev2.config` 文件中某个规则中的标签匹配。然后，他们重新启动 `in.ikev2d` 守护进程。

接收到另一系统的预先共享密钥后，管理员编辑 `ikev2.preshared` 文件。`partym` 上的文件如下所示：

```
# 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"
}
```

因此，`enigma` 上的 `ikev2.preshared` 密钥文件必须如下所示：

```
#…
```

```
{ 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)"
}
```

管理员重新启动每个系统上的 IKEv2 服务实例。

```
# svcadm restart ikev2
```

接下来的步骤

如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

有关更多示例，请参见 `ikev2.config(4)` 和 `ikev2.preshared(4)` 手册页。

▼ 在 IKEv2 中使用预先共享的密钥时如何添加新的对等方

如果将 IPsec 策略项添加到相同对等方之间的工作配置，则需要刷新 IPsec 策略服务。无需重新配置或重新启动 IKE。

如果将新的对等方添加到 IPsec 策略，则除了进行 IPsec 更改之外，还必须修改 IKEv2 配置。

开始之前

您已更新了对等方系统的 `ipsecinit.conf` 文件并刷新了 IPsec 策略。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. **为 IKEv2 创建一个规则以管理使用 IPsec 的新系统的密钥。**

   a. **例如，在 `enigma` 系统上，将以下规则添加到文件 `/etc/inet/ike/ikev2.config`：**

   ```
   # 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
}
```

有关 pfedit 命令的选项的信息，请参见 pfedit(1M) 手册页。

b. 在 **ada** 系统上，添加以下规则：

```
## 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. （可选） 在每个系统上，验证该文件的语法。

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

3. 为对等方系统创建 IKEv2 预先共享的密钥。

a. 在 **enigma** 系统上，将以下信息添加到 **/etc/inet/ike/ikev2.preshared** 文件：

```
# 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"
}
```

有关 pfedit 命令的选项的信息，请参见 pfedit(1M) 手册页。

b. 在 **ada** 系统上，将以下信息添加到 **ikev2.preshared** 文件：

```
# 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. 在每个系统上，将变更读入到内核。

- 如果已启用服务，请刷新此服务。

```
# svcadm refresh ikev2
```

- 如果未启用服务，请启用它。

```
# svcadm enable ikev2
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

初始化密钥库以存储 IKEv2 的公钥证书

要将公共证书与 IKEv2 配合使用，必须创建一个 PKCS #11 密钥库。最常用的密钥库使用由 Oracle Solaris 的加密框架功能提供的 `pkcs11_softtoken`。

IKEv2 的 `pkcs11_softtoken` 密钥库位于归特殊用户 `ikeuser` 所有的目录中。缺省目录为 `/var/user/ikeuser`。用户 ID `ikeuser` 随系统一起提供，但必须创建密钥库。创建密钥库时，请为密钥库创建 PIN。IKEv2 服务需要使用此 PIN 登录到密钥库。

`pkcs11_softtoken` 密钥库会存储 IKEv2 使用的私钥、公钥和公共证书。这些密钥和证书通过 `ikev2cert` 命令管理，此命令是 `pktool` 命令的包装器。此包装器可确保所有密钥和证书操作应用于归 `ikeuser` 所有的 `pkcs11_softtoken` 密钥库。

如果未将 PIN 添加为 `ikev2` 服务的属性值，`/var/log/ikev2/in.ikev2d.log` 文件中将显示以下消息：

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

如果未使用公钥证书，可以忽略此消息。

▼ 如何为 IKEv2 公钥证书创建并使用密钥库

如果计划将公共证书与 IKEv2 一起使用，必须创建一个密钥库。要使用密钥库，必须登录到其中。`in.ikev2d` 守护进程启动时，您或自动流程需要向守护进程提供 PIN。如果站点安全允许自动登录，您必须对其进行配置。缺省设置为通过交互式登录使用密钥库。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 设置 IKEv2 密钥库的 PIN。

   使用 `ikev2cert setpin` 命令创建 IKEv2 密钥库。此命令将 PKCS #11 密钥库所有者设置为 `ikeuser`。

   不能在 PIN 中使用空格。例如，值 `WhatShouldIWrite` 有效，但值 `"What Should"` 无效。

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

注意 - 将此口令短语存储在一个安全的位置。使用密钥库时需要用到它。

2. 自动或以交互方式登录到密钥库。

   首选自动登录。如果站点安全策略不允许自动登录，必须在重新启动 `in.ikev2d` 守护进程时以交互方式登录到密钥库。

   - 将密钥库配置为启用自动登录。

     a. 将 PIN 添加为 **`pkcs11_softtoken/pin`** 服务属性的值。

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

     此时会打开一个临时编辑窗口。

     b. 对 **`setprop pkcs11_token/pin =`** 行取消注释。

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

     c. 用来自步骤 1 的 PIN 替换括号。

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

     在冒号和 PIN 之间留一个空格。

     d. 对文件底部的 **`refresh`** 行取消注释，然后保存更改。

```
# refresh
refresh
```

     e. （可选） 验证 **`pkcs11_token/pin`** 属性的值。

     `pkcs11_token/pin` 属性会存储访问归 `ikeuser` 所有的密钥库时要检查的值。

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

```
pkcs11_token/pin     astring      PIN
```

- **未配置密钥库自动登录时，请以手动方式登录到密钥库。**

  每当 `in.ikev2d` 守护进程启动时运行此命令。

```
# 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. （可选） 验证是否已在密钥库中设置 PIN。

```
# 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` 列中的 `LI` 指示 PIN 已设置。

4. **要手动注销 `pkcs11_softtoken`，请使用 `ikeadm` 命令。**

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

您可以执行注销操作，将两个站点之间的通信限定为一段有限的时间。注销后，私钥会变得不可用，因此无法发起新的 IKEv2 会话。现有的 IKEv2 会话会继续，除非使用 `ikeadm delete ikesa` 命令删除会话密钥。预先共享的密钥规则会继续起作用。请参见 `ikeadm`(1M) 手册页。

使用公钥证书配置 IKEv2

公共证书对于大型部署是一个不错的选择。有关更多信息，请参见“IKE，使用公钥证书” [119]。

公钥证书由加密框架存储在 softtoken 密钥库中。在连接了硬件的系统上，也可以在硬件中生成和存储证书。有关过程，请参见如何在硬件中为 IKEv2 生成和存储公钥证书 [147]。

有关背景信息，请参见“IKE 的工作原理” [118]。

以下任务列表列出了为 IKEv2 创建公钥证书的过程。如果系统连接了 Sun Crypto Accelerator 6000 板，这些过程会包括如何在硬件密钥库中存储证书的内容。

表 9-1 使用公钥证书任务列表配置 IKEv2

任务	说明	参考
为证书创建密钥库。	初始化用来存储 IKEv2 证书的 PKCS #11 密钥库。	“初始化密钥库以存储 IKEv2 的公钥证书” [133]
使用自签名公钥证书配置 IKEv2。	创建一个由您签名的公钥证书。将证书导出至对等方，并导入对等方的证书。	如何使用自签名公钥证书配置 IKEv2 [136]
使用 CA 颁发的证书配置 IKEv2。	需要您创建一个 CSR，然后将所有返回的证书导入到密钥库。然后，验证并导入 IKE 对等方的证书。	如何使用 CA 签名的证书配置 IKEv2 [141]
配置处理已撤销证书的方式。	确定是否使用 CRL 以及是否轮询 OCSP 服务器，包括如何处理网络延迟。	如何在 IKEv2 中设置证书验证策略 [144]
执行配置，将证书存储在连接的硬件的密钥库中。	找到 Sun Crypto Accelerator 6000 板，将 IKEv2 配置为使用此板。	如何在硬件中为 IKEv2 生成和存储公钥证书 [147]

▼ 如何使用自签名公钥证书配置 IKEv2

在此过程中，请创建一个公钥证书并签名。私钥和证书存储在 IKEv2 的 PKCS #11 softtoken 密钥库中。将公钥证书发送给 IKE 对等方，随后对等方会发来它们的公共证书。

在所有使用自签名证书的 IKE 系统上执行此过程。

开始之前 要使用证书，必须完成如何为 IKEv2 公钥证书创建并使用密钥库 [133]。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。必须使用配置文件 shell。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 ssh 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. **在密钥库中创建一个自签名证书。**

   有关 `ikev2cert gencert` 命令参数的说明，请查看 `pktool(1)` 手册页中的 `pktool gencert keystore=pkcs11` 子命令。

   有关 `subject` 参数的格式，请参见“在 IKE 中使用公钥证书” [120]。

   注 - 给证书分配一个标签。此标签用来标识本地密钥库中的证书及其相应的密钥。

   a. **例如，`partym` 系统上命令的显示与以下内容类似：**

```
# 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
```

以下错误消息表明键入了错误的 PIN 或密钥库未初始化：

```
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
```

提示 - 显示的 pktool 命令语法表明部分证书项键入错误。检查命令是否使用了禁用的算法、缺少双引号和等号以及存在其他拼写错误。找到无效参数的策略之一是检索命令，然后每次删除一个参数。

---

b. **enigma** 系统上命令的显示与以下内容类似：

```
# 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. （可选） 列出密钥和证书。

```
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
```

提示 - 缺省的散列算法是 SHA1。要使用更强大的签名算法创建证书，请使用 keytype 选项和不同的散列算法，例如 keytype=rsa hash=sha384。有关选项，请参见 pktool(1) 手册页。

3. 向另一个系统提供证书。

   a. 在每个系统上，仅将证书导出到文件。

      outformat=pem 选项可确保公共证书以适合直接导入的格式放置在文件中。标签会标识密钥库中的证书。

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

   b. 通过电子邮件、**sftp** 或 **ssh** 将证书发送到另一个系统。

      例如，如果您管理两个系统，可使用 sftp 命令将证书从另一个系统带过来。

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

      此时会提示您输入口令。在本示例中，jdoe 必须提供一个口令。

4. 检验证书是否完全相同。

   您需要确保在将证书装入密钥库之前收到了正确的证书。

   a. 在每个系统上创建一个导出文件摘要。

      例如，partym 管理员通过电子邮件将包含 partym 的证书的文件摘要发送给另一个管理员。enigma 管理员通过电子邮件发送 enigma 证书文件摘要。

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

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

   b. 在另一个系统上，对包含来自第一个系统的证书的文件运行 **digest** 命令。

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

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

这些摘要必须匹配。如果不匹配，则将无法将文件导入密钥库。关于另一种检验证书有效性的方法，请参见例 9-3 “根据指纹验证公钥证书”。

5. **检验之后，将另一个系统的证书导入您的密钥库。**

   将证书导入密钥库时，必须为证书分配一个标签，用来唯一标识您系统上的证书。标签会将公钥与其公钥证书相关联。

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

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

6. **（可选）列出密钥库中的对象。**

   比较此列表及步骤 2中的列表。例如，在 `enigma` 密钥库中，添加了 `partym` 证书。

```
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. **在每个系统上，使用 IKEv2 规则中的证书。**

   使用 `pfedit` 命令编辑 `/etc/inet/ike/ikev2.config` 文件。

a. 例如，在 **partym** 系统上，**ikev2.config** 文件中规则的显示与以下内容类似：

```
##  ... 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. 在 **enigma** 系统上，将 **enigma** 证书的 DN 用作 **ikev2.config** 文件中 **local_id** 的值。

对于远程参数，请使用 partym 证书的 DN 作为该值。确保本地系统上 label 关键字的值是唯一的。

```
…
 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. （可选） 在每个系统上，检查 **ikev2.config** 文件的有效性。

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

修复任何拼写错误或差错，然后继续。

9. 在每个系统上，检查 IKEv2 服务实例的状态。

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

10. 在每个系统上，启用 IKEv2 服务实例。

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

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

例 9-2 创建一个生命周期有限的自签名证书

在本示例中，管理员指定证书有效期为两年。

```
# 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
```

例 9-3 根据指纹验证公钥证书

在本示例中，管理员使用证书指纹验证证书。此方法的缺点是，管理员必须在查看指纹之前将对等方的证书导入密钥库。

管理员导入证书，使用 `ikev2cert list objtype=cert` 命令列出证书，然后在输出内容中复制证书指纹，将其发送给另一个系统管理员。

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

如果验证失败，导入证书的管理员必须将证书及其公钥从密钥库删除。

```
# 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
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何使用 CA 签名的证书配置 IKEv2

保护大量通信系统的组织通常使用证书颁发机构 (certificate authority, CA) 颁发的公共证书。有关背景信息，请参见“IKE，使用公钥证书” [119]。

请在所有使用 CA 颁发的证书的 IKE 系统上执行此过程。

开始之前 要使用证书，必须完成如何为 IKEv2 公钥证书创建并使用密钥库 [133]。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 ssh 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. 转到可写入目录。

   以下错误消息可能表明 CSR 文件无法写入磁盘：

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

   例如，使用 /tmp 目录。

   ```
   # cd /tmp
   ```

2. 创建证书签名请求。

   使用 ikev2cert gencsr 命令创建证书签名请求 (certificate signing request, CSR)。有关该命令参数的说明，请查看 pktool(1) 手册页中的 pktool gencsr keystore=pkcs11 子命令。

   例如，以下命令会在 partym 系统上创建一个包含 CSR 的文件：

   ```
   # 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. （可选） 复制 CSR 的内容并粘贴到 CA 的 Web 表单中。

   ```
   # 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. 将 CSR 提交给 certificate authority, CA（证书颁发机构）。

CA 可能会告诉您如何提交 CSR。大多数组织具有包含提交表单的 Web 站点。该表单要求证明提交是合法的。通常，您需要将 CSR 粘贴到表单中。

提示 - 有些 Web 表单有一个 "Advanced"（高级）按钮，您可以在这里粘贴证书。CSR 以 PKCS#10 格式生成。因此，查找 Web 表单中提到 PKCS#10 的部分。

5. 将您从 CA 收到的每个证书导入密钥库。

   `ikev2cert import` 会将证书导入密钥库。

   a. 导入从 CA 收到的公钥和证书。

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

      提示 - 为方便管理，向导入的证书分配相同的标签作为原始 CSR 的标签。

   b. 导入来自 CA 的根证书。

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

   c. 将任何中间 CA 证书导入密钥库。

      提示 - 为方便管理，向导入的中间证书分配相同的标签作为原始 CSR 的标签。

      如果 CA 为每个中间证书发送了单独的文件，则在导入前述证书时导入它们。然而，如果 CA 通过一个 PKCS#7 文件提供其证书链，则必须从此文件提取各个证书，然后在导入前述证书时导入每个证书：

      注 - 您必须承担 `root` 角色才能运行 `openssl` 命令。请参见 `openssl(5)` 手册页。

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

6. 设置证书验证策略。

   如果证书包含适用于 CRL 或 OCSP 的部分，则必须根据站点要求配置证书验证策略。有关说明，请参见如何在 IKEv2 中设置证书验证策略 [144]。

7. 在所有使用您的证书的 IKE 系统上完成此过程后，再在所有系统上启用 **ikev2** 服务。

   对等方系统需要使用 trust anchor（信任锚）证书和已配置的 `ikev2.config` 文件。

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何在 IKEv2 中设置证书验证策略

您可以针对证书在 IKEv2 系统中的处理方式执行多方面的配置。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 ssh 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. **查看缺省证书验证策略。**

   证书策略是在安装时在 `/etc/inet/ike/kmf-policy.xml` 文件中设置的。此文件归 `ikeuser` 所有，可以使用 `kmfcfg` 命令进行修改。缺省证书验证策略是将 CRL 下载到 `/var/user/ikeuser/crls` 目录。缺省情况下允许使用 OCSP。如果站点需要通过代理连接 Internet，您必须配置此代理。请参见如何在 IKEv2 中处理已撤销的证书 [145]。

```
# 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. 查看证书，确认有无表明相关验证选项要加以修改的功能。

   例如，一个包含 CRL 或 OCSP 的证书可以使用验证策略指定 URI，以便用来查看证书撤销状态。您也可以配置超时。

3. 查看 **kmfcfg**(1) 手册页，了解可配置的选项。

4. 配置证书验证策略。

   有关策略样例，请参见如何在 IKEv2 中处理已撤销的证书 [145]。

▼ 如何在 IKEv2 中处理已撤销的证书

已撤销证书是指因为某个原因而泄密的证书。使用已撤销证书会带来安全风险。验证证书是否已经撤销时，您可以使用多种方式。您可以使用静态列表，或者通过 HTTP 协议动态验证证书是否已经撤销。

开始之前 您收到并安装了来自 CA 的证书。

您熟悉检查证书是否已经撤销的 CRL 和 OSCP 方法。有关信息和指示，请参见“IKE，使用公钥证书” [119]。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员，并且使用配置文件 shell。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 在从 CA 接收的证书中找到 CRL 和 OCSP 部分。

   您可以根据 CSR 的标签标识证书。

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

   例如，以下已截断的输出突出显示了证书中的 CRL 和 OCSP URI。

```
# 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
```

在 CRL Distribution Points 项下，URI 值指示此组织的 CRL 可从 Web 上的文件获取。OCSP 项指示各个证书的状态可以从服务器动态确定。

2. 通过指定一个代理，启用对 CRL 或 OCSP 服务器的使用。

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

在代理为可选的站点，无需指定一个代理。

3. 确认证书验证策略是否已更新。

例如，确认 OCSP 是否已更新。

```
# 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. 重新启动 IKEv2 服务。

```
# svcadm restart ikev2
```

5. （可选） 停止使用 CRL 或 OCSP。

- 要停止使用 CRL，请键入：

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

crl-none=true 参数会强制系统使用从本地高速缓存下载的 CRL。

- 要停止使用 OCSP，请键入：

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

例 9-4 更改系统等待 IKEv2 证书验证的时间

在本示例中，管理员将等待证书验证的时间限定为 20 秒。

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

缺省情况下，当响应超时时，表明对等方验证成功。在这里，管理员配置一个策略，规定验证失败时拒绝连接。在此配置中，如果 OCSP 或 CRL 服务器不响应，则证书验证失败。

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

要激活此策略，管理员需要重新启动 IKEv2 服务。

```
# svcadm restart ikev2
```

▼ 如何在硬件中为 IKEv2 生成和存储公钥证书

公钥证书也可以存储在连接的硬件上。Sun Crypto Accelerator 6000 板会提供存储，并允许将公钥操作从系统分流到板上。

在硬件上生成和存储公钥证书，与在系统上生成和存储公钥证书类似。在硬件上，`ikev2cert gencert token=`*hw-keystore* 命令用于标识硬件密钥库。

开始之前 此过程假设 Sun Crypto Accelerator 6000 板已连接到系统。此过程还假定已安装板的软件，而且已配置硬件密钥库。有关说明，请参见 Sun Crypto Accelerator 6000 Board Product Library Documentation (http://docs.oracle.com/cd/E19321-01/index.html)（Sun Crypto Accelerator 6000 板产品库文档）。这些说明包括密钥库设置内容。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. 确认您拥有已连接 Sun Crypto Accelerator 6000 板的令牌 ID。

```
# 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. 生成自签名证书或 CSR，并指定令牌 ID。

注 - 对于 RSA，Sun Crypto Accelerator 6000 板最多支持 2048 位的密钥。对于 DSA，此板最多支持 1024 位的密钥。

选择以下选项之一：

- 对于自签名证书，请使用此语法：

```
# 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
```

- 对于证书签名请求，请使用此语法：

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

有关 `ikev2cert` 命令参数的说明，请参见 `pktool(1)` 手册页。

3. 在系统提示输入 PIN 时，键入 Sun Crypto Accelerator 6000 用户名、冒号和该用户的口令。

注 - 您必须知道密钥库的用户名和口令。

如果 Sun Crypto Accelerator 6000 板由口令为 `inThe%4ov` 的用户 `admin` 配置，应键入以下内容：

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

4. 发送您的证书以供对方使用。

选择以下选项之一：

- **将自签名证书发送到远程系统。**
  可以将证书粘贴到电子邮件中。

- **将证书签名请求发送给 CA。**
  按 CA 提供的说明提交 CSR。有关更详细的论述，请参见“在 IKE 中使用公钥证书” [120]。

5. **将证书导入硬件密钥库。**
   导入从 CA 收到的证书，并提供步骤 3 中的用户和 PIN。

```
# 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. **启用自动使用或以交互方式使用硬件密钥库。**
   首选自动登录。如果站点安全策略不允许自动登录，必须在重新启动 `in.ikev2d` 守护进程时以交互方式登录到密钥库。

   - **配置密钥库自动登录。**

     a. **将 PIN 添加为 `pkcs11_token/uri` 服务属性的值。**
        有关此属性的说明，请参见“IKEv2 服务” [203]。

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

        此时会打开一个临时编辑窗口。

     b. **对 `setprop pkcs11_token/uri =` 行取消注释，用以下格式的令牌名称替换括号：**

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

     c. **对 `setprop pkcs11_token/uri =` 行取消注释，用来自步骤 3 的 *username:PIN* 替换括号。**

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

     d. **对文件底部的 `refresh` 行取消注释，然后保存更改。**

```
# refresh
refresh
```

e. （可选）验证 **pkcs11_token** 属性的值。

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

- 如果未配置自动登录，请手动登录到硬件密钥库。

  每当 `in.ikev2d` 守护进程启动时运行此命令。

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

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

♦ ♦ ♦ 第 10 章

10 配置 IKEv1

本章介绍如何为系统配置 Internet 密钥交换版本 1 (IKEv1)。配置 IKEv1 后，它将自动为网络上的 IPsec 生成加密材料。本章包含以下信息：

注 - 如果您计划仅实现 IKEv2，请转至第 9 章 配置 IKEv2。

有关 IKE 的概述信息，请参见第 8 章 关于 Internet 密钥交换。有关 IKE 的参考信息，请参见第 12 章 IPsec 和密钥管理参考。有关更多过程，请参见 `ikeadm(1M)`、`ikecert(1M)` 和 `ike.config(4)` 手册页的示例部分。

注 - 这些任务假设系统已分配了静态 IP 地址，并且正在运行网络配置文件 `DefaultFixed`。如果 `netadm list` 命令返回 `Automatic`，请参见 `netcfg(1M)` 手册页了解更多信息。

配置 IKEv1

可以使用预先共享的密钥、自签名证书和证书颁发机构 (certificate authority, CA) 颁发的证书来验证 IKE。`ike/config` 文件中的规则将特定的 IKEv1 验证方法与 IKEv1 对等方相关联。因此，可以在系统上使用一种或所有 IKE 验证方法。利用指向 PKCS #11 库的指针，IKEv1 可以使用连接的硬件加速器。

配置 IKEv1 后，请完成第 7 章 配置 IPsec 中使用 IKEv1 配置的 IPsec 任务。

使用预先共享的密钥配置 IKEv1

如果您要将对等方系统或子网配置为使用 IKEv1，而且您是这些子网的管理员，则使用预先共享的密钥是一个不错的选择。预先共享的密钥也可以在测试时使用。有关更多信息，请参见“使用预先共享的密钥验证的 IKE” [119]。

▼ 如何使用预先共享的密钥配置 IKEv1

IKE 实现提供了采用可变密钥长度的算法。所选的密钥长度是由站点安全性确定的。通常，密钥越长，提供的安全性就越高。

在此过程中，将生成 ASCII 格式的密钥。

以下过程使用系统名称 `enigma` 和 `partym`。请用您的系统名称替换名称 `enigma` 和 `partym`。

注 - 要在 Trusted Extensions 系统上使用带标签的 IPsec，请参见《Trusted Extensions 配置和管理 》中的“如何在多级别 Trusted Extensions 网络中应用 IPsec 保护”中此过程的扩展。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. **在每个系统上，创建一个 `/etc/inet/ike/config` 文件。**

   您可以使用 `/etc/inet/ike/config.sample` 作为模板。

2. **在每个系统上的 `ike/config` 文件中输入规则和全局参数。**

   此文件中的规则和全局参数应该允许系统的 `ipsecinit.conf` 文件中的 IPsec 策略可以成功实施。以下 IKEv1 配置示例会与如何使用 IPsec 保护两台服务器之间的网络通信 [96]中的 `ipsecinit.conf` 示例配合使用。

   a. **例如，在 `enigma` 系统上修改 `/etc/inet/ike/config` 文件：**

   ```
   ### 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. 在 **partym** 系统上修改 **/etc/inet/ike/config** 文件：

```
### 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. 在每个系统上，验证该文件的语法。

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

4. 将预先共享的密钥放置在每个系统上的 **/etc/inet/secret/ike.preshared** 文件中。

a. 例如，在 **enigma** 系统上，**ike.preshared** 文件的显示与以下内容类似：

```
## 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. 在 **partym** 系统上，**ike.preshared** 文件的显示与以下内容类似：

```
## 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. 启用 IKEv1 服务。

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

例 10-1 刷新 IKEv1 预先共享的密钥

如果 IKEv1 管理员要刷新预先共享的密钥，他们可以编辑对等方系统上的文件，然后重新启动 `in.iked` 守护进程。

首先，在两个子网中使用预先共享的密钥的每个系统上，管理员更改预先共享的密钥项。

```
# 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)"
 }
```

然后，管理员重新启动每个系统上的 IKEv1 服务。

有关 `pfedit` 命令的选项的信息，请参见 `pfedit(1M)` 手册页。

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

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何针对新的对等方系统更新 IKEv1

如果将 IPsec 策略项添加到相同对等方之间的工作配置，则需要刷新 IPsec 策略服务。无需重新配置或重新启动 IKEv1。

如果将新的对等方添加到 IPsec 策略，则除了进行 IPsec 更改之外，还必须修改 IKEv1 配置。

开始之前 您已更新了对等方系统的 `ipsecinit.conf` 文件并刷新了 IPsec 策略。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. 为 IKEv1 创建一个规则以管理使用 IPsec 的新系统的密钥。

   a. 例如，在 **`enigma`** 系统上，将以下规则添加到文件 **`/etc/inet/ike/config`**：

   ```
   ### 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. 在 **`ada`** 系统上，添加以下规则：

   ```
   ### 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. 为对等方系统创建 IKEv1 预先共享的密钥。

a. 在 **enigma** 系统上，将以下信息添加到 **/etc/inet/secret/ike.preshared** 文件：

```
## 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. 在 **ada** 系统上，将以下信息添加到 **ike.preshared** 文件：

```
## 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. 在每个系统上，刷新 **ike** 服务。

```
# svcadm refresh ike:default
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

使用公钥证书配置 IKEv1

使用公钥证书，通信系统就无需在带外共享秘密的加密材料。证书颁发机构 (certificate authority, CA) 颁发的公共证书通常需要与外部组织进行协商。证书很容易扩展为保护大量通信系统。

公钥证书也可以生成和存储于所连接的硬件中。有关过程，请参见“配置 IKEv1 查找连接的硬件” [179]。

所有证书都有一个采用 X.509 distinguished name, DN（标识名）格式的唯一名称。此外，证书可能有一个或多个主题备用名称，例如电子邮件地址、DNS 名称、IP 地址等。您可以按完整的 DN 或按某个主题备用名称在 IKEv1 配置中标识证书。这些备用名称采用 *tag=value* 格式，其中值的格式与其标记类型相对应。例如，`email` 标记的格式为 *name@domain.suffix*。

以下任务列表列出了为 IKEv1 创建公钥证书的过程。这些过程包括如何在连接的硬件上加速和存储证书。

表 10-1 使用公钥证书任务列表配置 IKEv1

任务	说明	参考
使用自签名公钥证书配置 IKEv1。	在每个系统上创建并放置密钥和两个证书： ■ 自签名证书及其密钥 ■ 来自对等方系统的公钥证书	如何使用自签名公钥证书配置 IKEv1 [157]
通过证书颁发机构配置 IKEv1。	创建证书签名请求，然后在每个系统上放置 CA 颁发的证书。请参见“在 IKE 中使用公钥证书” [120]。	如何使用 CA 签名的证书配置 IKEv1 [162]
在本地硬件中配置公钥证书。	涉及以下操作之一： ■ 在本地硬件中生成自签名证书，然后将公钥从远程系统添加到硬件。 ■ 在本地硬件中生成证书签名请求，然后将来自 CA 的公钥证书添加到硬件。	如何在硬件中为 IKEv1 生成和存储公钥证书 [166]
更新来自 CA 的证书撤销列表 (certificate revocation list, CRL)。	从中心分发点访问 CRL。	如何在 IKEv1 中处理已撤销的证书 [170]

注 - 要在 Trusted Extensions 系统上为包和 IKE 协商贴上标签，请按照《Trusted Extensions 配置和管理 》中的“配置有标签的 IPsec”中的过程操作。

公钥证书在 Trusted Extensions 系统上的全局区域中管理。Trusted Extensions 不会更改管理和存储证书的方式。

▼ 如何使用自签名公钥证书配置 IKEv1

在此过程中，您可以创建一个称作证书对的公钥/私钥和证书。私钥存储于本地证书数据库中的磁盘上，可以使用 `ikecert certlocal` 命令进行引用。公钥和证书存储在公共证书数据库中。可以使用 `ikecert certdb` 命令进行引用。您将与对等方系统交换此公共证书。两个证书用于验证 IKEv1 传输。

自签名证书比 CA 颁发的公共证书所需的开销少，但不太易于扩展。不同于 CA 颁发的证书，自签名证书必须由两位交换证书的管理员进行验证。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. 在每个 IKEv1 系统上，在 **ike.privatekeys** 数据库中创建自签名证书。

   有关 `ikecert certlocal` 命令的参数，请参见 ikecert(1M) 手册页。

   a. 例如，**partym** 系统上命令的显示与以下内容类似：

```
# 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-----
```

   其中，

   `-ks` 创建自签名证书。

   `-m` *keysize* 指定密钥的大小。

   `-t` *keytype* 指定要使用的算法类型。

   `-D` *dname* 指定证书主题的 X.509 标识名 (distinguished name, DN)。有关示例，请参见“在 IKE 中使用公钥证书” [120]。

   `-A` *altname* 指定证书的替代名称或昵称。*altname* 的格式为 *tag=value*。有效标记是 `IP`、`DNS`、`email` 和 `DN`。

注 - `-D` 和 `-A` 选项的值是仅标识证书而非任何系统的名称，例如 `192.168.13.213`。事实上，由于这些值为证书昵称，必须在带外验证对等方系统上是否安装了正确的证书。

---

   b. **enigma** 系统上命令的显示与以下内容类似：

```
# 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. 保存证书并将它发送到远程系统。

输出为证书的公共部分的编码版本。您可以安全地将此证书粘贴到电子邮件中。接收方必须在带外验证其是否安装了正确的证书，如步骤 4 中所述。

a. 例如，将 **partym** 证书的公共部分发送给 **enigma** 管理员。

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

b. **enigma** 管理员将向您发送 **enigma** 证书的公共部分。

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

3. 在每个系统上，将接收到的证书添加到公钥数据库中。

a. 将管理员的电子邮件保存到能以 **root** 身份读取的文件中。

b. 将该文件重定向到 **ikecert** 命令。

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

该命令将导入 BEGIN 与 END 标记之间的文本。

4. 与另一位管理员一起验证证书是否来自该管理员。
例如，可以致电其他管理员，以验证您拥有的其公共证书的散列是否与仅他们拥有的其私钥证书的散列匹配。

a. 列出 **partym** 中存储的证书。
在以下示例中，Note 1 指示了 slot 0 中证书的 distinguished name, DN（标识名）。slot 0 中的私钥具有相同的散列（请参见 Note 3），因此这些证书具有相同的证书对。要使用公共证书，必须具有匹配的证书对。certdb 子命令可列出公共部分，而 certlocal 子命令可列出私密部分。

```
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
```

此检查操作已验证 partym 系统具有有效的证书对。

b. **验证 enigma 系统是否具有 partym 的公共证书。**

您可通过电话读取公钥散列。

将上一步骤中 partym 上 Note 3 的散列与 enigma 上 Note 4 的散列进行比较。

```
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
```

存储于 enigma 的公共证书数据库中的最后一个证书的公钥散列和主题名称与上一步骤中 partym 的私密证书匹配。

5. **在每个系统上，信任这两个证书。**

编辑 /etc/inet/ike/config 文件以识别证书。

远程系统的管理员提供 cert_trust、remote_addr 和 remote_id 参数的值。

a. 例如，在 **partym** 系统上，**ike/config** 文件的显示与以下内容类似：

```
# 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. 在 **enigma** 系统上，在 **ike/config** 文件中添加本地参数的 **enigma** 值。

对于远程参数，请使用 partym 值。确保本地系统上 label 关键字的值是唯一的。

```
…
{
 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. 在对等方系统上，启用 IKEv1。

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

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何使用 CA 签名的证书配置 IKEv1

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. 使用 **`ikecert certlocal -kc`** 命令创建证书签名请求 (certificate signing request, CSR)。

   有关该命令参数的说明，请参见如何使用自签名公钥证书配置 IKEv1 [157] 中的步骤 1。

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

   a. 例如，以下命令会在 **`partym`** 系统上创建 CSR：

```
# 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. 以下命令会在 **enigma** 系统上创建 CSR：

```
# 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. 将 CSR 提交给 CA。

   CA 可能会告诉您如何提交 CSR。大多数组织具有包含提交表单的 Web 站点。该表单要求证明提交是合法的。通常，您需要将 CSR 粘贴到表单中。组织检查您的请求后，将向您颁发已签名证书。有关更多信息，请参见“在 IKE 中使用公钥证书” [120]。

3. 将每个证书添加到系统。

   `ikecert certdb -a` 的 `-a` 选项将已粘贴的对象添加到系统上的适当证书数据库。有关更多信息，请参见“IKE，使用公钥证书” [119]。

   a. 成为管理员。

      有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

   b. 添加从 CA 收到的公钥及其证书。

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

   c. 添加 CA 提供的公共证书。

      您可能还需要添加中间证书。

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

   d. 如果 CA 发送了已撤销证书列表，则将 CRL 添加到 **certrldb** 数据库：

      ```
      # ikecert certrldb -a
      Press the Return key
      Paste the CRL
      -----BEGIN CRL-----
      …
      -----END CRL----
      Press the Return key
      ```

```
Press Control-D
```

4. 使用 **/etc/inet/ike/config** 文件中的 **cert_root** 关键字标识颁发证书的 CA。
使用 CA 证书的标识名 (Distinguished Name, DN)。

   a. 例如，**partym** 系统上 **ike/config** 文件的显示可能与以下内容类似：

```
# 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}
}
```

注 - auth_method 参数的所有变量都必须在同一行上。

   b. 在 **enigma** 系统上，创建一个类似的文件。
   具体而言，enigma ike/config 文件必须执行以下操作：

   - 包括相同的 cert_root 值。
   - 对于本地参数，使用 enigma 值。
   - 对于远程参数，使用 partym 值。
   - 为 label 关键字创建唯一值。此值必须与远程系统的 label 值不同。

```
…
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. 设置 IKEv1 策略以处理已撤销的证书。

   选择适当的选项：

   - 未提供 OCSP

     如果公钥证书提供一个访问 OCSP 服务器的 URI，但您的系统无法连接到 Internet，请将关键字 `ignore_ocsp` 添加到 `ike/config` 文件。

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

     `ignore_ocsp` 关键字会要求 IKEv1 假设此证书有效。

   - 未提供 CRL

     如果 CA 没有提供可靠的 CRL 来源，或者您的系统无法连接到 Internet 检索 CRL，请将关键字 `ignore_crls` 添加到 `ike/config` 文件。

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

   - 提供了 CRL 或 OCSP 的 URI

     如果 CA 为已撤销证书提供了中心分发点，则可以修改 `ike/config` 文件以使用 URI。

     有关示例，请参见如何在 IKEv1 中处理已撤销的证书 [170]。

例 10-2 配置 IKEv1 时使用 `rsa_encrypt`

在 `ike/config` 文件中使用 `auth_method rsa_encrypt` 时，必须将对等方系统的证书添加到 `publickeys` 数据库。

1. 将证书发送给远程系统的管理员。

   可以将证书粘贴到电子邮件中。

   例如，`partym` 管理员可以发送以下消息：

   ```
   To: admin@enigma.ja.example.com
   From: admin@party.us.example.com
   ```

```
Message: -----BEGIN X509 CERTIFICATE-----
MII…
----END X509 CERTIFICATE-----
```

enigma 管理员可以发送以下消息：

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

2. 在每个系统上，将通过电子邮件发送的证书添加到本地 publickeys 数据库。

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

RSA 加密的验证方法可防止窃听者知道 IKE 中的标识。由于 rsa_encrypt 方法会隐藏对等方的身份，这导致 IKEv1 无法检索对等方的证书。因此，rsa_encrypt 方法要求 IKEv1 对等方知道彼此的公钥。

所以，在 /etc/inet/ike/config 文件中使用 rsa_encrypt 的 auth_method 时，必须将对等方的证书添加到 publickeys 数据库。添加证书后，publickeys 数据库至少包含每对通信系统的三个证书：

- 您的公钥证书
- CA 的证书链
- 对等方的公钥证书

故障排除 – IKEv1 有效负荷（至少包括三个证书）可能变得过大而无法由 rsa_encrypt 加密。诸如 "authorization failed"（授权失败）和 "malformed payload"（有效负荷格式错误）之类的错误，可以指明 rsa_encrypt 方法无法对总有效负荷进行加密。使用仅需要两个证书的方法（如 rsa_sig）来减小有效负荷的大小。

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何在硬件中为 IKEv1 生成和存储公钥证书

在硬件上生成和存储公钥证书，与在系统上生成和存储公钥证书类似。在硬件上，ikecert certlocal 和 ikecert certdb 命令必须标识硬件。带有令牌 ID 的 -T 选项向命令标识硬件。

开始之前

- 必须配置硬件。

- 该硬件使用 `/usr/lib/libpkcs11.so` 库，除非 `/etc/inet/ike/config` 文件中的 `pkcs11_path` 关键字指向其他库。该库必须按照以下标准实现：RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)，即 PKCS #11 库。

  有关设置说明，请参见如何配置 IKEv1 来查找 Sun Crypto Accelerator 6000 板 [179]。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 `ssh` 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全远程登录的说明。

1. **生成自签名证书或 CSR，并指定令牌 ID。**

   注 - 对于 RSA，Sun Crypto Accelerator 6000 板最多支持 2048 位的密钥。对于 DSA，此板最多支持 1024 位的密钥。

   选择以下选项之一：

   - **对于自签名证书，请使用此语法：**

     ```
     # 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` 选项的参数是来自已连接 Sun Crypto Accelerator 6000 板的令牌 ID。

   - **对于 CSR，请使用此语法：**

     ```
     # 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
     ```

   有关 `ikecert` 命令参数的说明，请参见 `ikecert(1M)` 手册页。

2. **在系统提示输入 PIN 时，键入 Sun Crypto Accelerator 6000 用户名、冒号和该用户的口令。**

   如果 Sun Crypto Accelerator 6000 板具有口令为 `rgm4tigt` 的用户 `ikemgr`，应键入以下内容：

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

注 - 如果您键入 ikecert 命令及 -p 选项，PKCS #11 令牌将作为明文存储在磁盘上，受 root 权限保护。如果您没有在磁盘上存储 PIN，则必须在运行 in.iked 命令后使用 ikeadm 命令解除令牌锁定。

键入口令后，证书会显示以下输出：

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

3. **将您的证书发送给另一方。**
   选择以下选项之一：

   - **将自签名证书发送到远程系统。**
     可以将证书粘贴到电子邮件中。

   - **将 CSR 发送到 certificate authority, CA（证书颁发机构）。**
     按 CA 的说明提交证书请求。有关更详细的论述，请参见如何使用 CA 签名的证书配置 IKEv1 [162] 的步骤 2。

4. **在系统上，编辑 /etc/inet/ike/config 文件以识别这些证书。**
   选择以下选项之一。

   - **自签名证书**
     使用远程系统管理员为 cert_trust、remote_id 和 remote_addr 参数提供的值。例如，在 enigma 系统上，ike/config 文件的显示与以下内容类似：

```
# 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}
}
```

- **证书请求**

  将 CA 提供的名称作为 `cert_root` 关键字的值键入。例如，`enigma` 系统上 `ike/config` 文件的显示可能与以下内容类似：

```
# 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. **在硬件中存放来自对方的证书。**

   按照在步骤 2 中作出的响应，响应 PIN 请求。

注 - 必须将公钥证书添加到生成私钥的那个连接硬件上。

- **自签名证书。**

  添加远程系统的自签名证书。在此示例中，证书存储在 `DCA.ACCEL.STOR.CERT` 文件中。

```
# ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
Enter PIN for PKCS#11 token:      Type user:password
```

  如果自签名证书将 `rsa_encrypt` 用作 `auth_method` 参数的值，则将对等方的证书添加到硬件存储。

- **来自 CA 的证书。**

  添加 CA 根据您的证书请求生成的证书以及组织的证书。

  您可能还需要添加中间证书。

```
# 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
```

要添加来自 CA 的证书撤销列表 (certificate revocation list, CRL)，请参见如何在 IKEv1 中处理已撤销的证书 [170]。

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

▼ 如何在 IKEv1 中处理已撤销的证书

已撤销证书是指因为某个原因而泄密的证书。使用已撤销证书会带来安全风险。验证证书是否已经撤销时，您可以使用多种方式。您可以使用静态列表，或者通过 HTTP 协议动态验证证书是否已经撤销。您可以通过四种方式处理已撤销证书。

- 您可以指示 IKEv1 忽略在证书中嵌入了其统一资源指示符 (uniform resource indicator, URI) 的 CRL 或 OCSP。步骤 5 展示了此选项。
- 您可以指示 IKEv1 从一个 URI 访问 CRL 或 OCSP，该 URI 的地址嵌入在来自 CA 的公钥证书中。
- 您可以指示 IKEv1 从 LDAP 服务器访问 CRL，该服务器的 DN（directory name，目录名称）项嵌入在来自 CA 的公钥证书中。
- 您可以提供 CRL 作为 `ikecert certrldb` 命令的参数。有关示例，请参见例 10-3 “将 CRL 粘贴到 IKEv1 的本地 `certrldb` 数据库中”。

开始之前 您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. 显示从 CA 收到的证书。

   有关 `ikecert certdb` 命令的参数的信息，请参见 `ikecert`(1M) 手册页。

   例如，以下证书由公司的 PKI 颁发。详细信息已更改。

```
# 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
```

请注意 `CRL Distribution Points` 项。

- URI 项指示此组织的 CRL 在 Web 上是可用的。
- DN 项指示 CRL 在 LDAP 服务器上是可用的。在 IKE 访问 CRL 后，将高速缓存该 CRL 以供将来使用。

要访问 CRL，您需要到达分发点。

2. 选择以下方法之一从中心分发点访问 CRL。

   - 使用 URI。
     将关键字 `use_http` 添加到主机的 `/etc/inet/ike/config` 文件。例如，`ike/config` 文件的显示与以下内容类似：

     ```
     # Use CRL or OCSP from organization's URI
     use_http
     …
     ```

   - 使用 Web 代理。
     将关键字 `proxy` 添加到 `ike/config` 文件。`proxy` 关键字将 URL 用作参数，如下所示：

     ```
     # Use web proxy to reach CRLs or OCSP
     proxy "http://proxy1:8080"
     ```

   - 使用 LDAP 服务器。
     在主机的 `/etc/inet/ike/config` 文件中，将 LDAP 服务器指定为 `ldap-list` 关键字的参数。您的组织提供 LDAP 服务器的名称。`ike/config` 文件中项的显示与以下内容类似：

     ```
     # Use CRL from organization's LDAP
     ```

```
ldap-list "ldap1.example.com:389,ldap2.example.com"
…
```

在证书到期之前，IKE 检索并高速缓存 CRL。

例 10-3 将 CRL 粘贴到 IKEv1 的本地 `certrldb` 数据库中

如果无法从中心分发点获取 CA 的 CRL，则可以将该 CRL 手动添加到本地 `certrldb` 数据库。按照 CA 的说明将 CRL 提取到文件中，然后使用 `ikecert certrldb -a` 命令将此 CRL 添加到数据库。

```
# ikecert certrldb -a < ExampleCo.Cert.CRL
```

为移动系统配置 IKEv1

IPsec 和 IKE 要求用唯一 ID 标识源和目标。对于没有唯一 IP 地址的站点外系统或移动系统，必须使用其他 ID 类型。可以使用诸如 `DNS`、`DN` 或 `email` 之类的 ID 类型唯一地标识系统。

对于具有唯一 IP 地址的站点外系统或移动系统，最好也应使用其他 ID 类型进行配置。例如，如果系统尝试从 NAT 盒 (NAT box) 之后连接到中心站点，则不会使用它们的唯一地址。NAT 盒 (NAT box) 指定一个中心系统无法识别的任意 IP 地址。

预先共享的密钥也不太适合用作移动系统的验证机制，因为预先共享的密钥需要固定的 IP 地址。通过使用自签名证书或来自 CA 的证书，移动系统可以与中心站点通信。

以下任务列表列出了配置 IKEv1 以处理远程登录到中心站点的系统的过程。

表 10-2 为移动系统配置 IKEv1 任务列表

任务	说明	参考
从站点外与中心站点进行通信。	允许站点外系统与中心站点进行通信。站点外系统可能是移动系统。	如何为站点外系统配置 IKEv1 [173]
在接受来自移动系统的通信的中心系统上使用 CA 的公共证书和 IKEv1。	将网关系统配置为接受来自没有固定 IP 地址的系统的 IPsec 流量。	例 10-4 “将中心计算机配置为使用 IKEv1 接受来自移动系统的受保护通信”
在没有固定 IP 地址的系统上使用 CA 的公共证书和 IKEv1。	将移动系统配置为保护它传输到中心站点（如公司总部）的流量。	例 10-5 “使用 IPsec 和 IKEv1 配置 NAT 之后的系统”
在接受来自移动系统的通信的中心系统上使用自签名证书和 IKEv1。	使用自签名证书配置网关系统，以接受来自移动系统的 IPsec 流量。	例 10-6 “接受来自移动系统的自签名证书”
在没有固定 IP 地址的系统上使用自签名证书和 IKEv1。	使用自签名证书配置移动系统，以保护它传输到中心站点的流量。	例 10-7 “使用自签名证书联系中心系统”

▼ 如何为站点外系统配置 IKEv1

开始之前 您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。如果执行远程管理，请参见例 7-1 “通过使用 ssh 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. 将中心系统配置为识别移动系统。

   a. 配置 **ipsecinit.conf** 文件。

      中心系统需要一个允许很宽的 IP 地址范围的策略。随后，IKE 策略中的证书确保进行连接的系统是合法的。

```
# /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. 配置 IKEv1 配置文件。

      DNS 标识中心系统。证书用于验证该系统。

```
## /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. **登录到每个移动系统，然后将该系统配置为查找中心系统。**

   a. **配置 `/etc/hosts` 文件。**

   `/etc/hosts` 文件不需要移动系统的地址，但是可以提供一个地址。该文件必须包含中心系统 *central* 的公共 IP 地址。

   ```
   # /etc/hosts on mobile
   central 192.xxx.xxx.x
   ```

   b. **配置 `ipsecinit.conf` 文件。**

   移动系统需要按照中心系统的公共 IP 地址来查找中心系统。这些系统必须配置相同的 IPsec 策略。

   ```
   # /etc/inet/ipsecinit.conf on mobile
   # Find central
   {raddr 192.xxx.xxx.x} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}
   ```

   c. **配置 IKEv1 配置文件。**

   标识符不能是 IP 地址。以下标识符对移动系统有效：

   - `DN=`*ldap-directory-name*
   - `DNS=`*domain-name-server-address*
   - `email=`*email-address*

   证书用于验证移动系统 *mobile*。

   ```
   ## /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. 启用 **ike:default** 服务。

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

例 10-4 将中心计算机配置为使用 IKEv1 接受来自移动系统的受保护通信

IKE 可以从 NAT 盒 (NAT box) 之后启动协商。但是，IKE 的理想设置是在 NAT 盒 (NAT box) 没有介入的情况下进行的。在以下示例中，CA 的公共证书放置在移动系统和中心系统上。中心系统接受来自 NAT 盒 (NAT box) 之后的系统的 IPsec 协商。`main1` 是可以接受来自站点外系统的连接的公司系统。有关如何设置站点外系统，请参见例 10-5 “使用 IPsec 和 IKEv1 配置 NAT 之后的系统”。

```
## /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}
}
```

例 10-5 使用 IPsec 和 IKEv1 配置 NAT 之后的系统

在以下示例中，CA 的公共证书放置在移动系统和中心系统上。`mobile1` 将从本部连接到公司总部。Internet 服务提供商 (Internet service provider, ISP) 网络使用 NAT 盒 (NAT box) ，以允许 ISP 为 `mobile1` 指定专用地址。然后，NAT 盒 (NAT box) 将专用地址转换为与其他 ISP 网络节点共享的公共 IP 地址。公司总部不在 NAT 之后。有关如何在公司总部设置计算机，请参见例 10-4 “将中心计算机配置为使用 IKEv1 接受来自移动系统的受保护通信”。

```
## /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 }
}
```

例 10-6 接受来自移动系统的自签名证书

在以下示例中，自签名证书已经颁发，并存放在移动系统和中心系统上。main1 是可以接受来自站点外系统的连接的公司系统。有关如何设置站点外系统，请参见例 10-7 “使用自签名证书联系中心系统”。

```
## /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 }
}
```

例 10-7 使用自签名证书联系中心系统

在以下示例中，mobile1 将从本部连接到公司总部。证书已经颁发，并放置在移动系统和中心系统上。ISP 网络使用 NAT 盒 (NAT box) ，以允许 ISP 为 mobile1 指定专用地址。然后，NAT 盒 (NAT box) 将专用地址转换为与其他 ISP 网络节点共享的公共 IP 地址。公司总部不在 NAT 之后。有关如何在公司总部设置计算机，请参见例 10-6 “接受来自移动系统的自签名证书”。

```
## /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 }
}
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

配置 IKEv1 查找连接的硬件

公钥证书也可以存储在连接的硬件上。Sun Crypto Accelerator 6000 板提供存储，并允许将公钥操作从系统转移到板上。

▼ 如何配置 IKEv1 来查找 Sun Crypto Accelerator 6000 板

开始之前 以下过程假定 Sun Crypto Accelerator 6000 板已连接到系统。此过程还假定已安装板的软件，而且已配置该软件。有关说明，请参见 Sun Crypto Accelerator 6000 Board Product Library Documentation (http://docs.oracle.com/cd/E19321-01/index.html)（Sun Crypto Accelerator 6000 板产品库文档）。

您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

如果执行远程管理，请参见例 7-1 “通过使用 ssh 连接远程配置 IPsec 策略” 和《在 Oracle Solaris 11.2 中管理安全 Shell 访问 》中的“如何使用安全 Shell 远程管理 ZFS”，了解进行安全的远程登录的说明。

1. **验证 PKCS #11 库是否已链接。**

   IKEv1 使用该库的例程在 Sun Crypto Accelerator 6000 板上处理密钥生成和密钥存储操作。

   ```
   $ ikeadm get stats
   …
   PKCS#11 library linked in from /usr/lib/libpkcs11.so
   $
   ```

2. **查找已连接的 Sun Crypto Accelerator 6000 板的令牌 ID。**

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

该库返回一个包含 32 个字符的令牌 ID（也称为 keystore name（密钥库名称））。在此示例中，可以将 `Sun Metaslot` 令牌与 `ikecert` 命令一起使用来存储和加速 IKEv1 密钥。

有关如何使用令牌的说明，请参见如何在硬件中为 IKEv1 生成和存储公钥证书 [166]。

结尾空格是由 `ikecert` 命令自动填充的。

例 10-8 查找和使用 metaslot 令牌

令牌可以存储在磁盘上、连接的板上或加密框架提供的 softtoken 密钥库中。softtoken 密钥库令牌 ID 可能与以下信息类似。

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                  "
```

有关如何为 softtoken 密钥库创建口令短语，请参见 `pktool(1)` 手册页。

如下所示的命令可向 softtoken 密钥库添加证书。`Sun.Metaslot.cert` 是一个包含 CA 证书的文件。

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:      Type user:passphrase
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。有关保护 VPN 的 IPsec 策略的示例，请参见“使用 IPsec 保护 VPN” [101]。有关 IPsec 策略的其他示例，请参见如何使用 IPsec 保护两台服务器之间的网络通信 [96]。

♦ ♦ ♦ 第 11 章

11 对 IPsec 及其密钥管理服务进行故障排除

本章介绍如何对 IPsec 及其密钥进行故障排除，如何查看配置信息，以及如何查看有关活动 IPsec、IKE 和手动密钥服务的信息。

本章包含以下信息：

对 IPsec 及其密钥管理配置进行故障排除

在发生需要排除的问题之前或期间，您可以设置系统以便开展故障排除。

开展故障排除时，您能够以拥有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员身份，在配置文件 shell 中运行许多命令。然而，要读取日志，您必须承担 `root` 角色。

故障排除部分的提示符会指示您是否必须拥有运行命令的权限。

- `#` 提示符 – 具备相应管理权限的用户或具备这些权限的角色可以运行此命令。
- `%` 提示符 – 普通用户可以运行命令。

▼ 如何准备 IPsec 和 IKE 系统以便开展故障排除

启用 IPsec 及其密钥管理服务之前，可以使用有助于故障排除的日志和工具设置系统。

1. 找到 IPsec 和 IKEv2 服务的日志。

-L 选项提供了完整的日志路径。这些日志包含信息消息以及错误消息。

```
% svcs -L policy
/var/svc/log/network-ipsec-policy:default.log

% svcs -L ikev2
/var/svc/log/network-ipsec-ike:ikev2.log
```

2. 针对 IKEv2 配置调试日志文件。

   root 角色可以读取这些日志。

```
% svccfg -s ikev2 listprop | grep debug
config/debug_level            astring     op
config/debug_logfile          astring     /var/log/ikev2/in.ikev2d.log
```

   ikeadm(1M) 手册页中介绍了调试级别。值 verbose 和 all 在故障排除时非常有用。

3. （可选） 配置调试级别。

   以下命令可以永久设置调试级别。要临时设置调试级别，请参见例 11-3 “在正在运行的 IKE 守护进程上设置新的调试级别”。

```
# svccfg -s ikev2 setprop config/debug_level = all
```

   如果已启用 ikev2 服务，必须刷新此服务以使用新的调试级别。

```
# svcadm refresh ikev2
```

4. （可选） 安装 **wireshark** 软件包。

   Wireshark 应用程序可以读取 snoop 输出。

```
% 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
```

▼ 如何在 IPsec 和 IKE 运行之前对系统进行故障排除

您可以在运行这些服务之前，检查 IPsec 配置文件的语法、IPsec 密钥文件的语法以及密钥库中证书的有效性。

1. 检验 IPsec 配置文件的语法。

```
# 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.
```

如果输出显示错误，请修复错误，运行命令，直到验证成功。

2. **检查 `ipseckeys` 文件的语法是否正确。**

```
# ipseckey -c /etc/inet/secret/ipseckeys
Config file /etc/inet/secret/ipseckeys has insecure permissions,
will be rejected in permanent config.
```

如果输出显示错误，请修复错误，然后刷新此服务。

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

注 - 通过运行 IKE 守护进程来验证 IKE 配置文件和 IKE 预先共享的密钥文件。

3. **检验证书的有效性。**

   - 要在 IKEv2 中检验自签名证书的有效性，请执行如何使用自签名公钥证书配置 IKEv2 [136] 中的步骤 4。
   - 要在 IKEv2 中检验公钥证书是否未被撤销，请执行过程如何在 IKEv2 中设置证书验证策略 [144]。
   - 要在 IKEv1 中检验自签名证书的有效性，请执行如何使用自签名公钥证书配置 IKEv1 [157] 中的步骤 4。
   - 要在 IKEv1 中检验公钥证书是否未被撤销，请执行过程如何在 IKEv1 中处理已撤销的证书 [170]。

接下来的步骤 如果您的配置在启用 IPsec 及其加密服务时不工作，必须在服务运行时进行故障排除。

▼ 如何在 IPsec 运行时对系统进行故障排除

在正在运行且使用 IKE 交换或尝试交换包的系统上，可以使用 `ikeadm` 命令查看统计信息、规则、预先共享的密钥和其他项。此外，也可以使用日志文件和部分工具，例如 Wireshark 应用程序。

1. **检查以下项：**

   - **检验 `policy` 和相应的的密钥管理服务是否已启用。**

在以下测试系统上，使用 manual-key 服务进行密钥管理：

```
% 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
```

如果已禁用此服务，请启用它。

可以同时使用两个 IKE 服务。也可以同时使用手动密钥和 IKE，但此配置会导致难以进行故障排除的异常情况。

- 查看 IKEv2 服务的日志文件的结尾。

```
# 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.
```

- （可选） 可以为正在运行的守护进程的调试级别设置临时值。

```
# 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. 检验 **ipsecconf** 命令的输出是否与策略文件的内容相匹配。

```
# 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 }
```

注 - 通配符地址可能会让匹配变得不易辨识，因此请检验 `ipsecinit.conf` 文件中的任何特定地址是否在 `ipsecconf` 输出的通配符地址范围内。

如果无法为 `ipsecconf` 命令显示任何输出，请检验策略服务是否已启用，并刷新服务。

```
% svcs policy
STATE          STIME    FMRI
online         Apr_10   svc:/network/ipsec/policy:default
```

如果输出显示错误，请编辑 `ipsecinit.conf` 文件以修复错误，然后刷新服务。

3. 验证 IKEv2 配置。

   有关可能需要修复的配置输出，请参见例 11-1 “修复无效的 IKEv2 配置” 和例 11-2 “修复无匹配规则消息”。以下示例中的输出指示配置有效。

```
# /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.
```

注 - 有关调试输出的警告不会改变，即便在指定调试日志文件之后也是如此。如果指定 `debug_logfile` 服务属性的值，警告意味着正在向该文件提供调试输出。否则，调试输出会提供给控制台。

- 在 `Checking IKE rule` 行中，检验 IKE 规则是否连接适当的 IP 地址。例如，以下项匹配。来自 `ipsecinit.conf` 文件的 `laddr` 值与来自 `ikev2.config` 文件的 `local_addr` 值相匹配，并且远程地址相匹配。

```
{  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 **/
```

  如果这些项不对应，请修复配置，标识正确的 IP 地址。

注 - 规则可以使用通配符地址，例如涵盖地址范围的 10.134.0.0/16。针对特定地址检验范围。

- 如果 Pre-shared key file 行指示文件无效，请修复此文件。

  查找拼写错误。另外，在 IKEv2 中，查看 ikev2.config 中规则的标签值是否与 ikev2.preshared 文件中的标签值相匹配。接下来，如果您使用两个密钥，请检验一个系统上的本地预先共享密钥是否与其对等方上的远程预先共享密钥相匹配，以及远程密钥是否与对等方上的本地密钥相匹配。

  如果配置依然不工作，请参见“对 IPsec 和 IKE 语义错误进行故障排除” [187]。

例 11-1 修复无效的 IKEv2 配置

在以下输出中，IKE SA 的生命周期太短。

```
# /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"
...
```

已在 ikev2.config 文件中显式设置了该值。要删除警告，请将生命周期值更改为至少 100，然后刷新服务。

```
# pfedit /etc/inet/ike/ikev2.config
...
## childsa_lifetime_secs   60
childsa_lifetime_secs   100
...
# /usr/lib/inet/in.ikev2d -c
...
# svcadm refresh ikev2
```

例 11-2 修复无匹配规则消息

在以下输出中，定义了预先共享的密钥，但该密钥未在规则中使用。

```
# /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.
```

此输出指示仅存在一条规则。

- 如果规则要求预先共享的密钥，那么预先共享的密钥的标签与规则的标签不匹配。修复 `ikev2.config` 规则标签和 `ikev2.preshared` 密钥标签，使得它们相匹配。
- 如果规则使用证书，则可以删除或注释掉在 `ikev2.preshared` 文件第 12 行结束的预先共享的密钥，阻止 `No matching` 消息。

例 11-3 在正在运行的 IKE 守护进程上设置新的调试级别

在以下输出中，调试输出在 `ikev2` 服务中设置为 `all`。

```
# /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"
...
```

如果完成了如何在 IPsec 和 IKE 运行之前对系统进行故障排除 [182]中的步骤 2，并且调试输出依然为 `op` 而非 `all`，请使用 `ikeadm` 命令在正在运行的 IKE 守护进程上设置调试级别。

```
# ikeadm set debug_level all
```

对 IPsec 和 IKE 语义错误进行故障排除

如果在如何在 IPsec 运行时对系统进行故障排除 [183] 中开展的调查无法解决问题，那么问题很可能出在配置的语义而非文件或服务配置的语法上。

- 如果 `ike:default` 和 `ike:ikev2` 服务实例已启用，请确保 IKEv2 和 IKEv1 规则不重叠。应用于相同网络端点的规则会导致冗余 IPsec SA，并且有可能在特定情况下造成连接不足。

  如果更改 IKE 规则，请将规则读取到内核。

  ```
  # ikeadm -v[1|2] read rule
  ```

- 如果运行 IKEv1，请确保规则中的算法机制在您连接到的 IKEv1 系统上可用。要查看可用的算法，请在不支持 IKEv2 的系统上运行 `ikeadm dump` *algorithms* 命令：

  ```
  # ikeadm dump groups      Available Diffie-Hellman groups
  # ikeadm dump encralgs    All IKE encryption algorithms
  # ikeadm dump authalgs    All IKE authentication algorithms
  ```

  纠正 IPsec 和 IKEv1 策略文件，使用在两个系统上均可用的算法。然后，重新启动 IKEv1 服务，并刷新 IPsec 服务。

  ```
  # svcadm restart ike:default; svcadm refresh ipsec/policy
  ```

- 如果通过 IKEv1 使用预先共享的密钥并且远程 IKEv1 系统已重新引导，请在本地系统上运行 `ipseckey flush` 命令。
- 如果使用自签名证书，请与另一位管理员一起验证并未重新创建 DN 相同的证书，而且证书的散列值匹配。有关验证步骤，请参见如何使用自签名公钥证书配置 IKEv2 [136] 中的步骤 4。

  如果证书已更新，请导入新的证书，然后刷新并重新启动 IKEv2 服务。
- 使用 `ikeadm -v2 dump | get` 命令查看当前的 IKEv2 配置。有关使用摘要，请参见“查看 IKE 信息” [189]。
- 使用 `kstat` 命令显示与 IPsec 相关的统计信息。有关更多信息，请参见 `kstat(1M)` 手册页。

  ```
  # kstat -m ipsecesp
  # kstat -m ipsecah
  # kstat -m ip
  ```

  以下示例中的 `kstat` 输出指示 `ipsecesp` 模块中没有问题。

  ```
  # 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
  ```
- 使用 `snoop` 命令查看未受保护的通信。Wireshark 应用程序可以读取 `snoop` 输出。有关 `snoop` 输出的示例，请参见如何检验包是否受 IPsec 保护 [115]。

查看有关 IPsec 及其加密服务的信息

注 - 对于大多数命令，您必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。您必须在配置文件 shell 中键入信息。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

查看 IPsec 和手动密钥服务属性

可以查看 IPsec 策略文件以及保存手动密钥的文件的名称。

- 显示 IPsec 配置文件的名称：

```
% svccfg -s policy listprop config/config_file
config/config_file         astring     /etc/inet/ipsecinit.conf
```

- 显示保存 IPsec 手动密钥的文件的名称：

```
% svccfg -s manual-key listprop config/config_file
config/config_file         astring     /etc/inet/secret/ipseckeys
```

查看 IKE 信息

可以查看 IKE 服务的属性、IKE 状态和 IKE 守护进程对象的各方面情况以及证书验证策略。如果运行两个 IKE 服务，可以显示每个服务或两个服务的信息。这些命令在测试、故障排除和监视过程中很有帮助。

- 查看 IKE 服务实例的属性 – 此输出可以显示 IKEv2 服务的可配置属性，包括配置文件的名称。

注 - 查看 ipsecconf(1M)、in.ikev2d(1M) 和 in.iked(1M) 手册页，确保您可以或应该在 IPsec、IKEv2 或 IKEv1 服务的 config 组中修改属性。例如，IKEv2 配置文件使用特殊权限创建，归 ikeuser 所有。这些权限和文件所有者不得更改。

```
% 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
```

以下示例中的输出显示了 IKEv1 服务的可配置属性。请勿指定 `:default` 服务实例。

```
% 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
```

- 查看 IKE 守护进程的当前状态 – 以下示例中的输出显示了 `ikeadm` 命令的参数。这些参数可以显示守护进程的当前状态。

注 - 要使用 `ikeadm` 命令，必须运行 IKE 守护进程。

```
% 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]
```

- 显示 `ikeadm` 命令的特定参数的语法 – 使用 `help` 子命令显示命令参数语法。例如：

```
% 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.
```

- 查看预先共享的密钥 – 可以查看 IKEv1 和 IKEv2 的预先共享的密钥。

注 - 如果仅运行一个 IKE 版本，可以忽略 -v 选项。

对于 IKEv2：

```
# ikeadm -v2 dump preshared
```

对于 IKEv1：

```
# 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
```

- 查看 IKE SA – 此输出包括有关 SA、转换、本地和远程系统以及其他细节的信息。如果未请求通信，则 SA 不存在，也就不存在可显示的信息。

```
# 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
```

- 查看活动的 IKE 规则 – 已列出的 IKE 规则可能未被使用，但它可以使用。

```
# 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
```

- 查看 IKEv2 中的证书验证策略 – 必须指定 `dbfile` 值和 `policy` 值。
  - 动态下载的 CRL 可能要求管理员干预，对响应者超时进行调整。

    在以下示例的输出中，先从证书中嵌入的 URI 下载 CRL，然后缓存列表。当高速缓存包含过期的 CRL 时，将下载新的 CRL 替换旧的 CRL。

    ```
    # 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
    …
    ```

  - 静态下载的 CRL 会频繁要求管理员予以注意。

当管理员将 CRL 项设定为以下值时，管理员负责手动下载 CRL、填充目录以及维护当前的 CRL：

```
…
        Directory: /var/user/ikeuser/crls
        Download and cache CRL: false
        Proxy: [not set]
…
```

管理 IPsec 及其加密服务

缺省情况下会启用 IPsec 策略，但它缺少配置信息。

缺省情况下不会启用密钥管理。您可以配置 IKE 或手动密钥管理或同时配置这两者。每个 IKE 规则都会指示使用哪个密钥管理服务。`ikeadm` 命令可以修改正在运行的 IKE 守护进程。

配置和管理 IPsec 及其加密服务

- 配置和刷新 IPsec，然后查看策略：

  ```
  # pfedit /etc/inet/ipsecinit.conf
  # ipsecconf -c /etc/inet/ipsecinit.conf
  # svcadm refresh ipsec/policy
  # ipsecconf -Ln
  ```

- 配置和启用 IPsec 的手动密钥：

  ```
  # pfedit -s /etc/inet/secret/ipseckeys
  # svcadm enable ipsec/manual-key
  ```

- 配置和启用 IKEv2：

  ```
  # pfedit /etc/inet/ike/ikev2.config
  # /usr/lib/inet/in.ikev2d -c
  # svcadm enable ipsec/ike:ikev2
  ```

- 配置和启用 IKEv1：

  ```
  # pfedit /etc/inet/ike/config
  # /usr/lib/inet/in.iked -c
  # svcadm enable ipsec/ike:default
  ```

- 检验是否在已启用服务的系统上配置了 IPsec 和 IKE：

  ```
  # ipsecconf -Ln
  ```

```
# ikeadm -v2 dump rule
# ikeadm set priv keymat
# ikeadm -v1 dump rule
```

- 修改密钥管理：

  对于 IKEv2：

  ```
  # pfedit /etc/inet/ike/ikev2.config
  # /usr/lib/inet/in.ikev2d -c
  # svcadm restart ipsec/ike:ikev2
  ```

  对于 IKEv1：

  ```
  # pfedit /etc/inet/ike/config
  # /usr/lib/inet/in.iked -c
  # svcadm restart ipsec/ike:default
  ```

  对于手动密钥管理：

  ```
  # pfedit -s /etc/inet/secret/ipseckeys
  # ipseckey -c /etc/inet/secret/ipseckeys
  # svcadm refresh ipsec/manual-key
  ```

- 修改 IPsec 和 IKE 可配置属性：

  IPsec 服务：

  ```
  # svccfg -s ipsec/policy setprop config/property = value
  # svcadm refresh ipsec/policy; svcadm restart ipsec/policy
  ```

  IKEv2 服务：

  ```
  # svccfg -s ike:ikev2 editprop
  # svcadm refresh ipsec/ike:ikev2; svcadm restart ipsec/ike:ikev2
  ```

  IKEv1 服务：

  ```
  # svccfg -s ipsec/ike setprop config/property = value
  # svcadm refresh ipsec/ike:ikev2; svcadm restart ipsec/ike:ikev2
  ```

  手动密钥服务：

  ```
  # svccfg -s ipsec/manual-key setprop config/property = value
  # svcadm refresh ipsec/manual-key; svcadm restart ipsec/manual-key
  ```

- 为 IKEv2 配置预先共享的密钥：

  ```
  # pfedit -s /etc/inet/ike/ikev2.preshared
  # /usr/lib/inet/in.ikev2d -c
  # svcadm restart ikev2
  ```

- 为 IKEv1 配置预先共享的密钥：

```
# pfedit -s /etc/inet/secret/ike.preshared
# svcadm restart ike
```

管理正在运行的 IKE 守护进程

有关更多信息，请查看 ikeadm(1M) 手册页。本节中的命令仅在 IKEv2 或 IKEv1 守护进程运行时可用。

- 修改正在运行的 IKE 守护进程：

  以下输出显示了 ikeadm 命令的参数，此命令可以修改守护进程的当前状态。有些参数特定于 IKEv2 或 IKEv1 守护进程。

```
% 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
```

- 显示 ikeadm 命令的特定参数的语法：

```
% 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.
```

- 使用 ikeadm 命令修改 IKEv2 守护进程：

```
# 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
```

- 使用 `ikeadm` 命令修改 IKEv1 守护进程：

```
# 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
```

♦ ♦ ♦ 第 12 章

12 IPsec 和密钥管理参考

本章包含 IPsec、IKEv2 和 IKEv1 的参考信息。

有关如何在网络中实现 IPsec 的说明，请参见第 7 章 配置 IPsec。有关 IPsec 的概述，请参见第 6 章 关于 IP 安全体系结构。

有关实现 IKE 的说明，请参见第 9 章 配置 IKEv2。有关概述信息，请参见第 8 章 关于 Internet 密钥交换。

IPsec 参考

IPsec 服务、文件和命令

本节列出了 IPsec 服务、部分 IPsec RFC 以及与 IPsec 相关的文件和命令。

IPsec 服务

服务管理工具 (Service Management Facility, SMF) 为 IPsec 提供以下服务：

- `svc:/network/ipsec/policy` 服务 – 管理 IPsec 策略。缺省情况下，此服务处于启用状态。`config_file` 属性的值确定了 `ipsecinit.conf` 文件的位置。在运行 DefaultFixed 网络配置文件的系统上，初始值为 `/etc/inet/ipsecinit.conf`。在未运行此配置文件的系统上，属性值为空。
- `svc:/network/ipsec/ipsecalgs` 服务 – 管理可用于 IPsec 的算法。缺省情况下，此服务处于启用状态。
- `svc:/network/ipsec/manual-key` 服务 – 激活手动密钥管理。缺省情况下，此服务处于禁用状态。`config_file` 属性的值确定了 `ipseckeys` 配置文件的位置。初始值为 `/etc/inet/secret/ipseckeys`。

- svc:/network/ipsec/ike 服务 – 管理 IKE。缺省情况下，此服务处于禁用状态。有关可配置的属性，请参见“IKEv2 服务” [203] 和“IKEv1 服务” [206]。

有关 SMF 的信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。另请参见 smf(5)、svcadm(1M) 和 svccfg(1M) 手册页。

ipsecconf 命令

您可以使用 ipsecconf 命令为主机配置 IPsec 策略。当运行此命令来配置策略时，系统会在内核中创建 IPsec 策略项。系统使用这些项来检查所有传入和外发 IP 包的策略。未经过隧道传输和转发的包不受使用此命令添加的策略检查的约束。ipsecconf 命令还管理 security policy database, SPD（安全策略数据库）中的 IPsec 项。有关 IPsec 策略选项，请参见 ipsecconf(1M) 手册页。

您必须承担 root 角色才能调用 ipsecconf 命令。此命令可以配置保护双向通信的项，同时也可以配置仅保护单向通信的项。

具有本地地址和远程地址格式的策略项可以借助单个策略项保护双向通信。例如，如果没有为指定的主机指定方向，则包含模式 laddr host1 和 raddr host2 的项会保护双向通信。因此，对于每台主机，只需一个策略项。

由 ipsecconf 命令添加的策略项在系统重新引导后不会保留。要确保在系统引导时 IPsec 策略处于活动状态，请向 /etc/inet/ipsecinit.conf 文件中添加相应策略项，然后刷新或启用 policy 服务。有关示例，请参见“使用 IPsec 保护网络通信” [95]。

ipsecinit.conf 配置文件

要在启动 Oracle Solaris 时启用 IPsec 安全策略，请创建一个配置文件以通过特定的 IPsec 策略项来初始化 IPsec。此文件的缺省名称为 /etc/inet/ipsecinit.conf。有关策略项及其格式的详细信息，请参见 ipsecconf(1M) 手册页。在配置策略后，可以使用 svcadm refresh ipsec/policy 命令刷新该策略。

ipsecinit.conf 文件样例

Oracle Solaris 软件中包括样例 IPsec 策略文件 ipsecinit.sample。您可以使用此文件作为模板来创建自己的 ipsecinit.conf 文件。ipsecinit.sample 文件包含以下示例：

```
...
# 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` 和 `ipsecconf` 的安全注意事项

无法更改已建立连接的 IPsec 策略。其策略不能更改的套接字称为锁定的套接字。新策略项不保护已锁定的套接字。有关更多信息，请参见 `connect`(3SOCKET) 和 `accept`(3SOCKET) 手册页。如果有疑虑，请重新启动连接。有关更多信息，请参见 `ipsecconf`(1M) 手册页的“安全”部分。

`ipsecalgs` 命令

加密框架为 IPsec 提供了验证和加密算法。`ipsecalgs` 命令可以列出每个 IPsec 协议支持的算法。`ipsecalgs` 配置存储在 `/etc/inet/ipsecalgs` 文件中。通常，此文件无需修改，并且绝不能直接编辑。然而，如果您需要修改此文件，请使用 `ipsecalgs` 命令。系统引导时会通过 `svc:/network/ipsec/ipsecalgs:default` 服务使支持的算法与内核同步。

有效的 IPsec 协议和算法由 RFC 2407 中介绍的 ISAKMP domain of interpretation, DOI（系统解释域）进行说明。具体而言，ISAKMP DOI 为有效的 IPsec 算法及其协议

（PROTO_IPSEC_AH 和 PROTO_IPSEC_ESP）定义命名约定和编号约定。每个算法都仅与一项协议相关联。这些 ISAKMP DOI 定义位于 /etc/inet/ipsecalgs 文件中。算法和协议编号由 Internet 编号分配机构 (Internet Assigned Numbers Authority, IANA) 定义。使用 ipsecalgs 命令，可以针对 IPsec 扩展算法列表。

有关算法的更多信息，请参阅 ipsecalgs(1M) 手册页。有关加密框架的更多信息，请参见《在 Oracle Solaris 11.2 中管理加密和证书 》中的第 1 章 “加密框架”。

ipseckey 命令

带有各种选项的 ipseckey 命令可用来手动管理 IPsec 密钥。有关 ipseckey 命令的说明，请参见 ipseckey(1M) 手册页。

ipseckey 的安全注意事项

拥有 Network Security（网络安全）或 Network IPsec Management（网络 IPsec 管理）权限配置文件的角色可以使用 ipseckey 命令输入敏感的密钥加密信息。如果入侵者获得对此文件的访问权，便会威胁 IPsec 通信的安全。

注 - 如果可以，请使用 IKE 而非手动加密。

有关更多信息，请参见 ipseckey(1M) 手册页的“安全”部分。

kstat 命令

kstat 命令可以显示关于 ESP、AH 和其他 IPsec 数据的统计信息。“对 IPsec 和 IKE 语义错误进行故障排除” [187] 中列出了与 IPsec 相关的选项。另请参见 kstat(1M) 手册页。

snoop 命令和 IPsec

snoop 命令可以解析 AH 头和 ESP 头。由于 ESP 对自己的数据进行加密，因此，snoop 命令不能查看受 ESP 保护的加密头。AH 不对数据进行加密，因此，可以使用 snoop 命令来检查受 AH 保护的通信。此命令的 -v 选项显示何时对包使用 AH。有关更多信息，请参见 snoop(1M) 手册页。

有关受保护包中的详细 snoop 输出样例，请参见如何检验包是否受 IPsec 保护 [115]。

也可以使用第三方网络分析器，例如免费的开源软件 Wireshark (http://www.wireshark.org/about.html)，此发行版中已捆绑该软件。

IPsec RFC

Internet 工程任务组 (Internet Engineering Task Force, IETF) 已经发布了许多介绍 IP 层安全体系结构的请求注解文档 (Requests for Comment, RFC)。有关指向 RFC 的链接，请参见 http://www.ietf.org/。以下 RFC 列表包含更为常见的 IP 安全参考信息：

- RFC 2411，"IP Security Document Roadmap"，1998 年 11 月
- RFC 2401，"Security Architecture for the Internet Protocol"，1998 年 11 月
- RFC 2402，"IP Authentication Header"，1998 年 11 月
- RFC 2406，"IP Encapsulating Security Payload (ESP)"，1998 年 11 月
- RFC 2408，"Internet Security Association and Key Management Protocol (ISAKMP)"，1998 年 11 月
- RFC 2407，"The Internet IP Security Domain of Interpretation for ISAKMP"，1998 年 11 月
- RFC 2409，"The Internet Key Exchange (IKEv1)"，1998 年 11 月
- RFC 5996，"Internet Key Exchange Protocol Version 2 (IKEv2)"，2010 年 9 月
- RFC 3554，"On the Use of Stream Control Transmission Protocol (SCTP) with IPsec"，2003 年 7 月

IPsec 的安全关联数据库

有关 IPsec 安全服务加密材料的信息保留在安全关联数据库 (SADB) 中。安全关联 (security association, SA) 保护传入包和外发包。

in.iked 守护进程和 ipseckey 命令使用 PF_KEY 套接字接口维护 SADB。有关 SADB 如何处理请求和消息的更多信息，请参见 pf_key(7P) 手册页。

IPsec 中的密钥管理

Internet 密钥交换 (Internet Key Exchange, IKE) 协议会自动管理 IPsec 密钥。也可以使用 ipseckey 命令手动管理 IPsec SA，但推荐使用 IKE。有关更多信息，请参见“IPsec 安全关联的密钥管理” [85]。

Oracle Solaris 的服务管理工具 (Service Management Facility, SMF) 功能为 IPsec 提供以下密钥管理服务：

- `svc:/network/ipsec/ike` 服务 – 用于自动管理密钥的 SMF 服务。`ike` 服务有两个实例。`ike:ikev2` 服务实例运行 `in.ikev2d` 守护进程 (IKEv2) 以提供自动密钥管理。`ike:default` 服务运行 `in.iked` 守护进程 (IKEv1)。有关 IKE 的说明，请参见第 8 章 关于 Internet 密钥交换。有关这些守护进程的更多信息，请参见 `in.ikev2d(1M)` 和 `in.iked(1M)` 手册页。
- `svc:/network/ipsec/manual-key:default` 服务 – 用于手动管理密钥的 SMF 服务。`manual-key` 服务运行带有各种选项的 `ipseckey` 命令来手动管理密钥。有关 `ipseckey` 命令的说明，请参见 `ipseckey(1M)` 手册页。

IKEv2 参考

IKEv2 取代了 IKEv1。有关比较，请参见“比较 IKEv2 和 IKEv1” [121]。

IKEv2 实用程序和文件

下表概述了 IKEv2 策略的配置文件、IKEv2 密钥的存储位置以及实现 IKEv2 的各种命令和服务。有关服务的更多信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。

表 12-1 IKEv2 服务名称、命令、配置和密钥存储位置以及硬件设备

文件、位置、命令或服务	说明	手册页
`svc:/network/ipsec/ike:ikev2`	管理 IKEv2 的 SMF 服务。	`smf(5)`
`/usr/lib/inet/in.ikev2d`	Internet 密钥交换 (Internet Key Exchange, IKE) 守护进程。启用 `ike:ikev2` 服务时将激活自动密钥管理。	`in.ikev2d(1M)`
`/usr/sbin/ikeadm [-v 2]`	用于查看和临时修改 IKEv2 策略的 IKE 管理命令。用于查看 IKEv2 管理对象，例如可用的 Diffie-Hellman 组。	`ikeadm(1M)`
`/usr/sbin/ikev2cert`	用于以配置所有者 (`ikeuser`) 的身份创建和存储公钥证书的证书数据库管理命令。调用 `pktool` 命令。	`ikev2cert(1M)` `pktool(1)`
`/etc/inet/ike/ikev2.config`	IKEv2 策略的缺省配置文件。包含用于匹配传入 IKEv2 请求和准备外发 IKEv2 请求的站点规则。 如果此文件存在，在启用 `ike:ikev2` 服务时，`in.ikev2d` 守护进程会启动。可以使用 `svccfg` 命令更改此文件的位置。	`ikev2.config(4)`
`/etc/inet/ike/ikev2.preshared`	包含密钥，供两个未使用基于证书的验证的 IKEv2 实例进行彼此验证。	`ikev2.preshared(4)`
softtoken 密钥库	包含 IKEv2 的私钥和公钥证书，归 `ikeuser` 所有。	`pkcs11_softtoken (5)`

IKEv2 服务

服务管理工具 (Service Management Facility, SMF) 提供 svc:/network/ipsec/ike:ikev2 服务实例以管理 IKEv2。缺省情况下，此服务处于禁用状态。启用此服务之前，必须在 /etc/inet/ike/ikev2.config 文件中创建有效的 IKEv2 配置。

以下 ike:ikev2 服务属性是可配置的：

- config_file **属性** – 指定 IKEv2 配置文件的位置。初始值为 /etc/inet/ike/ikev2.config。此文件拥有特殊权限，必须归 ikeuser 所有。请不要使用其他文件。
- debug_level **属性** – 设置 in.ikev2d 守护进程的调试级别。初始值为 op 或 operational。有关可能的值，请参见 ikeadm(1M) 手册页中 "Object Types"（对象类型）下有关调试级别的表。
- debug_logfile **属性** – 指定用于调试 IKEv2 的日志文件的位置。初始值为 /var/log/ikev2/in.ikev2d.log。
- kmf_policy **属性** – 设置证书策略的日志文件的位置。缺省值为 /etc/inet/ike/kmf-policy.xml。此文件拥有特殊权限，必须归 ikeuser 所有。请不要使用其他文件。
- pkcs11_token/pin **属性** – 设置在 IKEv2 守护进程启动时用来登录到密钥库的 PIN。该值必须与使用 ikev2cert setpin 命令为令牌设置的值匹配。
- pkcs11_token/uri **属性** – 设置密钥库的 PKCS #11 URI。要使用加密加速卡上的硬件存储，必须提供该值。

有关 SMF 的信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。另请参见 smf(5)、svcadm(1M) 和 svccfg(1M) 手册页。

IKEv2 守护进程

in.ikev2d 守护进程自动管理 Oracle Solaris 系统上 IPsec 的加密密钥。该守护进程与运行相同协议的远程系统协商，以便以受保护方式为安全关联 (security association, SA) 提供经过验证的加密材料。此守护进程必须在所有计划使用 IPsec 并通过 IKEv2 协议保护通信的系统上运行。

缺省情况下，svc:/network/ipsec/ike:ikev2 服务未启用。配置了 /etc/inet/ike/ikev2.config 文件并启用 ike:ikev2 服务实例后，SMF 会在系统引导时启动 in.ikev2d 守护进程。

当 IKEv2 守护进程运行时，系统会针对其对等方 IKEv2 实体进行自我验证，建立会话密钥。按配置文件中指定的时间间隔，自动替换 IKE 密钥。in.ikev2d 守护进程通过 PF_KEY 套接字侦听从网络传入的 IKE 请求，并侦听外发通信请求。有关更多信息，请参见 pf_key(7P) 手册页。

有两个命令支持 IKEv2 守护进程。ikeadm 命令可用于查看 IKE 策略。有关更多信息，请参见“IKEv2 的 ikeadm 命令” [204]。使用 ikev2cert 命令可以查看和管理公钥与私钥证书。有关更多信息，请参见“IKEv2 ikev2cert 命令” [205]。

IKEv2 配置文件

IKEv2 配置文件 /etc/inet/ike/ikev2.config 负责管理规则，这些规则用来为 IPsec 策略文件 /etc/inet/ipsecinit.conf 中受保护的指定网络端点协商密钥。

使用 IKE 的密钥管理包括规则和全局参数。IKE 规则标识加密材料保护的系统或网络。该规则还指定验证方法。全局参数包括多个项，例如重设 IKEv2 SA 密钥之前的缺省时间 ikesa_lifetime_secs。有关 IKEv2 配置文件的示例，请参见“使用预先共享的密钥配置 IKEv2” [128]。有关 IKEv2 策略项的示例和说明，请参见 ikev2.config(4) 手册页。

IKEv2 支持的 IPsec SA 会根据 IPsec 配置文件 /etc/inet/ipsecinit.conf 中的策略来保护 IP 包。

ike/ikev2.config 文件的安全注意事项与 ipsecinit.conf 文件的安全注意事项类似。有关详细信息，请参见“ipsecinit.conf 和 ipsecconf 的安全注意事项” [199]。

IKEv2 的 ikeadm 命令

当 in.ikev2d 守护进程运行时，可以使用 ikeadm [-v2] 命令执行以下操作：

- 查看 IKEv2 状态的各个方面。
- 显示 IKEv2 守护进程对象，例如策略规则、预先共享的密钥、可用的 Diffie-Hellman 组、加密和验证算法以及现有的活动 IKEv2 SA。

有关此命令的选项的示例和完整说明，请参见 ikeadm(1M) 手册页。

ikeadm 命令的安全注意事项与 ipseckey 命令的安全注意事项类似。有关详细信息，请参见“ipseckey 的安全注意事项” [200]。

IKEv2 预先共享的密钥文件

/etc/inet/ike/ikev2.preshared 文件包含 IKEv2 服务使用的预先共享的密钥。此文件归 ikeuser 所有，按 0600 保护。

在需要预先共享的密钥的 ike/ikev2.config 文件中配置规则时，必须定制缺省的 ikev2.preshared 文件。由于 IKEv2 使用这些预先共享的密钥验证 IKEv2 对等方，此文件必须在 in.ikev2d 守护进程读取任何需要预先共享的密钥的规则之前已有效。

IKEv2 ikev2cert 命令

ikev2cert 命令用于生成、存储和管理公钥、私钥及证书。在 ike/ikev2.config 文件需要公钥证书时，可以使用此命令。由于 IKEv2 使用这些证书验证 IKEv2 对等方，证书必须在 in.ikev2d 守护进程读取任何需要证书的规则之前已到位。

ikev2cert 命令会以 ikeuser 身份调用 pktool 命令。

以下 ikev2cert 命令可以管理 IKEv2 的证书。这些命令必须由 ikeuser 帐户运行。结果存储在 PKCS #11 softtoken 密钥库中。

- ikev2cert setpin – 为 ikeuser 用户生成一个 PIN。使用证书时需要用到此 PIN。
- ikev2cert gencert – 生成自签名证书。
- ikev2cert gencsr – 生成证书签名请求 (certificate signing request, CSR)。
- ikev2cert list – 列出密钥库中的证书。
- ikev2cert export – 将证书导出至导出文件。
- ikev2cert import – 导入证书或 CRL。

有关 ikev2cert 子命令的语法的信息，请参见 pktool(1) 手册页。有关示例，请参见 ikev2cert(1M) 手册页。有关 softtoken 密钥库的信息，请参见 cryptoadm(1M) 手册页。

IKEv1 参考

以下部分提供有关 IKEv1 的参考信息。IKEv1 已经被 IKEv2 取代，后者可以提供更快的自动密钥管理。有关 IKEv2 的更多信息，请参见“IKEv2 参考” [202]。有关比较，请参见“比较 IKEv2 和 IKEv1” [121]。

IKEv1 实用程序和文件

下表概述了 IKEv1 策略的配置文件、IKEv1 密钥的存储位置以及实现 IKEv1 的各种命令和服务。有关服务的更多信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。

表 12-2 IKEv1 服务名称、命令、配置和密钥存储位置以及硬件设备

服务、命令、文件或设备	说明	手册页
`svc:/network/ipsec/ike:default`	管理 IKEv1 的 SMF 服务。	smf(5)
`/usr/lib/inet/in.iked`	Internet 密钥交换 (IKEv1) 守护进程。启用 `ike` 服务时将激活自动密钥管理。	in.iked(1M)
`/usr/sbin/ikeadm [-v1]`	用于查看和临时修改 IKE 策略的 IKE 管理命令。允许您查看 IKE 管理对象，例如阶段 1 算法和可用的 Diffie-Hellman 组。	ikeadm(1M)
`/usr/sbin/ikecert`	用于处理包含公钥证书的本地数据库的证书数据库管理命令。这些数据库也可以存储在连接的硬件上。	ikecert(1M)
`/etc/inet/ike/config`	IKEv1 策略的缺省配置文件。包含用于匹配传入 IKEv1 请求和准备外发 IKEv1 请求的站点规则。 如果此文件存在，在启用 `ike` 服务时，`in.iked` 守护进程会启动。可以使用 `svccfg` 命令更改此文件的位置。	ike.config(4)
`ike.preshared`	`/etc/inet/secret` 目录中的预先共享密钥文件。包含用于在阶段 1 交换中验证的密钥。在用预先共享的密钥配置 IKEv1 时使用。	ike.preshared(4)
`ike.privatekeys`	`/etc/inet/secret` 目录中的私钥目录。包含公钥/私钥对的私钥部分。	ikecert(1M)
`publickeys` 目录	`/etc/inet/ike` 目录中包含公钥和证书文件的目录。包含公钥/私钥对的公钥部分。	ikecert(1M)
`crls` 目录	`/etc/inet/ike` 目录中包含公钥和证书文件的撤销列表的目录。	ikecert(1M)
Sun Crypto Accelerator 6000 板	通过分流操作系统中的操作来加快公钥操作的硬件。该板还存储公钥、私钥和公钥证书。Sun Crypto Accelerator 6000 板是第 3 级的 FIPS 140-2 认证设备。	ikecert(1M)

IKEv1 服务

服务管理工具 (Service Management Facility, SMF) 提供 `svc:/network/ipsec/ike:default` 服务以管理 IKEv1。缺省情况下，此服务处于禁用状态。启用此服务之前，必须创建 IKEv1 配置文件 `/etc/inet/ike/config`。

以下 `ike` 服务属性是可配置的：

- `config_file` **属性** – 设置 IKEv1 配置文件的位置。初始值为 `/etc/inet/ike/config`。
- `debug_level` **属性** – 设置 `in.iked` 守护进程的调试级别。初始值为 `op` 或 operational。有关可能的值，请参见 ikeadm(1M) 手册页中 "Object Types"（对象类型）下有关调试级别的表。
- `admin_privilege` **属性** – 设置 `in.iked` 守护进程的特权级别。初始值为 `base`。其他值为 `modkeys` 和 `keymat`。有关详细信息，请参见“IKEv1 `ikeadm` 命令” [208]。

有关 SMF 的信息，请参见《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”。另请参见 smf(5)、svcadm(1M) 和 svccfg(1M) 手册页。

IKEv1 守护进程

in.iked 守护进程可以自动管理 IPsec SA，包括保护使用 IPsec 的包的加密密钥。此守护进程与运行 IKEv1 协议的对等方系统安全地协商 ISAKMP SA 和 IPsec SA。

缺省情况下，svc:/network/ipsec/ike:default 服务未启用。配置 /etc/inet/ike/config 文件并启用 ike:default 服务后，SMF 会在系统引导时启动 in.iked 守护进程。除了 /etc/inet/ike/config 文件外，其他配置存储在其他文件和数据库中，或者作为 SMF 属性。有关更多信息，请参见“IKEv1 实用程序和文件” [205] 和 ike.preshared(4)、ikecert(1M) 以及 in.iked(1M) 手册页。

启用 ike:default 服务后，in.iked 守护进程可以读取配置文件，侦听来自 IKE 对等方的外部请求以及来自 IPsec 的内部 SA 请求。

对于来自 IKEv1 对等方的外部请求，ike:default 服务的配置可以确定守护进程的响应方式。内部请求通过 PF_KEY 接口路由。此接口可以处理 IPsec 内核部分（存储 IPsec SA 并执行包加密和解密）与密钥管理守护进程 in.iked（在其用户级中运行）之间的通信。当内核需要 SA 保护包时，它会通过 PF_KEY 接口向 in.iked 守护进程发送一条消息。有关更多信息，请参见 pf_key(7P) 手册页。

有两个命令支持 IKEv1 守护进程。ikeadm 命令可以向正在运行的守护进程提供命令行接口。ikecert 命令管理磁盘和硬件上的证书数据库、ike.privatekeys 和 publickeys。

有关这些命令的更多信息，请参见 in.iked(1M)、ikeadm(1M) 和 ikecert(1M) 手册页。

IKEv1 配置文件

IKEv1 配置文件 /etc/inet/ike/config 根据 IPsec 配置文件 /etc/inet/ipsecinit.conf 中的策略为需要 IPsec 保护的网络包管理 SA。

使用 IKE 的密钥管理包括规则和全局参数。IKEv1 规则标识正在运行另一个 IKEv1 守护进程的系统。该规则还指定验证方法。全局参数包括诸如已连接硬件加速器路径之类的项。有关 IKEv1 策略文件的示例，请参见“使用预先共享的密钥配置 IKEv2” [128]。有关 IKEv1 策略项的示例和说明，请参见 ike.config(4) 手册页。

/etc/inet/ike/config 文件可以包含根据以下标准实现的库路径：RSA Security Inc. 的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)。IKEv1 使用此 PKCS #11 库访问硬件以进行密钥加速和密钥存储。

ike/config 文件的安全注意事项与 ipsecinit.conf 文件的安全注意事项类似。有关详细信息，请参见“ipsecinit.conf 和 ipsecconf 的安全注意事项” [199]。

IKEv1 ikeadm 命令

可以使用 ikeadm 命令执行以下操作：

- 查看 IKE 状态的各个方面
- 更改 IKE 守护进程的属性
- 显示在阶段 1 交换期间有关 SA 创建的统计信息
- 调试 IKE 协议交换
- 显示 IKE 守护进程对象，例如所有阶段 1 SA、策略规则、预先共享的密钥、可用的 Diffie-Hellman 组、阶段 1 加密和验证算法，以及证书高速缓存

有关此命令的选项的示例和完整说明，请参见 ikeadm(1M) 手册页。

正在运行的 IKE 守护进程的特权级别决定了可以查看和修改 IKE 守护进程的哪些方面。可以有三种特权级别：

base 级别	不能查看或修改密钥。base 级别是缺省特权级别。
keymat 级别	可以使用 ikeadm 命令查看实际的密钥。
modkeys 级别	可以删除、更改和添加预先共享的密钥。

如果要临时更改特权，可使用 ikeadm 命令。如果要进行永久更改，请更改 ike 服务的 admin_privilege 属性。有关临时特权更改，请参见“管理正在运行的 IKE 守护进程” [195]。

ikeadm 命令的安全注意事项与 ipseckey 命令的安全注意事项类似。请参见“ipseckey 的安全注意事项” [200]。有关特定于 ikeadm 命令的详细信息，请参见 ikeadm(1M) 手册页。

IKEv1 预先共享的密钥文件

如果手动创建预先共享的密钥，这些密钥将存储在 /etc/inet/secret 目录下的文件中。配置 ike/config 中的规则以使用预先共享的密钥时，ike.preshared 文件包含用于阶段 1 交换的预先共享的密钥。ipseckeys 文件包含用于保护 IP 包的预先共享的密钥。按 0600 保护这些文件。按 0700 保护 secret 目录。

由于预先共享的密钥用于验证阶段 1 交换，因此在 in.iked 守护进程启动之前，该文件必须有效。

有关手动管理 IPsec 密钥的示例，请参见如何手动创建 IPsec 密钥 [109]。

IKEv1 公钥数据库和命令

ikecert 命令管理本地系统的公钥/私钥、公共证书和静态 CRL 数据库。在 IKEv1 配置文件需要公钥证书时，可以使用此命令。由于 IKEv1 使用这些数据库验证阶段 1 交换，因此必须在激活 in.iked 守护进程之前填充这些数据库。以下三个子命令可分别处理三种数据库的其中一种：certlocal、certdb 和 certrldb。

如果系统连接了一个 Sun Crypto Accelerator 6000 板，ikecert 命令会使用 PKCS #11 库访问硬件密钥和证书存储。

有关更多信息，请参见 ikecert(1M) 手册页。有关 metaslot 以及 softtoken 密钥库的信息，请参见 cryptoadm(1M) 手册页。

IKEv1 ikecert tokens 命令

tokens 参数会列出可用的令牌 ID。使用令牌 ID 时，ikecert certlocal 和 ikecert certdb 命令可以生成公钥证书和 CSR。这些密钥和证书也可以存储在连接的 Sun Crypto Accelerator 6000 板上。ikecert 命令使用 PKCS #11 库访问硬件密钥库。

IKEv1 ikecert certlocal 命令

certlocal 子命令管理私钥数据库。使用此子命令的选项，可以添加、查看和删除私钥。此子命令还用于创建自签名的证书或 CSR。-ks 选项用于创建自签名的证书。-kc 选项会创建 CSR。密钥存储在系统的 /etc/inet/secret/ike.privatekeys 目录中，或者通过 -T 选项存储在连接的硬件上。

创建私钥时，ikecert certlocal 命令的选项必须在 ike/config 文件中具有相关项。ikecert 选项和 ike/config 项之间的对应关系如下表所示。

表 12-3 IKEv1 中的 ikecert 选项和 ike/config 项之间的对应关系

ikecert 选项	ike/config 项	说明
-A *subject-alternate-name*	cert_trust *subject-alternate-name*	唯一标识证书的别名。可能的值是 IP 地址、电子邮件地址或域名。
-D *X.509-distinguished-name*	*X.509-distinguished-name*	证书颁发机构的完整名称，包括国家/地区 (C)、组织名称 (ON)、组织单元 (OU) 和公用名称 (CN)。
-t dsa-sha1 | dsa-sha256	auth_method dsa_sig	一种速度比 RSA 稍慢的验证方法。
-t rsa-md5 和 -t rsa-sha1 | rsa-sha256 | rsa-sha384 | rsa-sha512	auth_method rsa_sig	一种速度比 DSA 稍快的验证方法。

ikecert 选项	ike/config 项	说明
		RSA 公钥必须大到足以加密最大的 payload（有效负荷）。通常，标识有效负荷（如 X.509 标识名）是最大的有效负荷。
-t rsa-md5 和 -t rsa-sha1 | rsa-sha256 | rsa-sha384 | rsa-sha512	auth_method rsa_encrypt	RSA 加密可防止窃听者知道 IKE 中的身份，但是要求 IKE 对等方知道彼此的公钥。

如果使用 ikecert certlocal -kc 命令发出 CSR，则会将该命令的输出发送到证书颁发机构 (certificate authority, CA)。如果您的公司运行自己的公钥基础结构 (public key infrastructure, PKI)，则会将输出发送给 PKI 管理员。接下来，CA 或 PKI 管理员会创建证书。返回给您的证书是 certdb 子命令的输入。CA 返回给您的证书撤销列表 (Certificate Revocation List, CRL) 是 certrldb 子命令的输入。

IKEv1 ikecert certdb 命令

certdb 子命令管理公钥数据库。使用此子命令的选项，可以添加、查看以及删除证书和公钥。该命令将 ikecert certlocal -ks 命令在远程系统上生成的证书作为输入接受。有关过程，请参见如何使用自签名公钥证书配置 IKEv1 [157]。此命令还将您从 CA 接收的证书接受为输入。有关过程，请参见如何使用 CA 签名的证书配置 IKEv1 [162]。

证书和公钥存储在系统的 /etc/inet/ike/publickeys 目录中。-T 选项在连接的硬件上存储证书、私钥和公钥。

IKEv1 ikecert certrldb 命令

certrldb 子命令会管理证书撤销列表 (Certificate Revocation List, CRL) 数据库 /etc/inet/ike/crls。CRL 数据库维护公钥的撤销列表。不再有效的证书包含在此列表中。当 CA 为您提供 CRL 时，您可以使用 ikecert certrldb 命令在 CRL 数据库中安装 CRL。有关过程，请参见如何在 IKEv1 中处理已撤销的证书 [170]。

IKEv1 /etc/inet/ike/publickeys 目录

/etc/inet/ike/publickeys 目录将公钥/私钥对的公钥部分及其证书包含在文件或插槽中。按 0755 保护该目录。ikecert certdb 命令填充该目录。-T 选项将密钥存储在 Sun Crypto Accelerator 6000 板上，而不是存储在 publickeys 目录中。

这些插槽以编码的格式包含在另一个系统上生成的证书的 X.509 标识名。如果使用自签名的证书，则将从远程系统管理员处接收的证书用作该命令的输入。如果使用来自 CA 的证书，则将两个签名证书从 CA 安装到此数据库中。将安装一个基于发送到 CA 的 CSR 的证书。也安装 CA 的证书。

IKEv1 /etc/inet/secret/ike.privatekeys 目录

/etc/inet/secret/ike.privatekeys 目录中存储属于公钥/私钥对一部分的私钥文件。按 0700 保护该目录。ikecert certlocal 命令填充 ike.privatekeys 目录。在安装其对应公钥、自签名的证书或 CA 后，私钥才生效。对应公钥存储在 /etc/inet/ike/publickeys 目录中，或存储在支持的硬件上。

IKEv1 /etc/inet/ike/crls 目录

/etc/inet/ike/crls 目录包含证书撤销列表 (Certificate Revocation List, CRL) 文件。每个文件都对应于 /etc/inet/ike/publickeys 目录中的公共证书文件。CA 为其证书提供 CRL。可以使用 ikecert certrldb 命令填充数据库。

网络安全词汇表

3DES	请参见 Triple-DES（三重 DES）。
AES	Advanced Encryption Standard（高级加密标准）。一种对称的块数据加密技术。美国政府在 2000 年 10 月采用该种算法的 Rijndael 变体作为其加密标准。AES 从而取代了 DES 成为政府的加密标准。
asymmetric key cryptography（非对称密钥密码学）	一种加密系统，消息的发送者和接收者使用不同的密钥对消息进行加密和解密。非对称密钥用于为对称密钥加密建立一个安全信道。Diffie-Hellman algorithm（Diffie-Hellman 算法）就是一种非对称密钥协议。该加密系统与 symmetric key cryptography（对称密钥密码学）相对。
authentication header（验证头）	为 IP 包提供验证和完整性而不提供保密性的扩展头。
bidirectional tunnel（双向隧道）	可以双向传输包的隧道。
Blowfish	一种对称块加密算法，它采用 32 位到 448 位的可变长度密钥。其作者 Bruce Schneier 声称 Blowfish 已针对密钥不经常更改的应用程序进行优化。
broadcast address（广播地址）	IPv4 网络地址，其主机部分的所有位全为 0 (10.50.0.0) 或全为 1 (10.50.255.255)。从本地网络上的计算机发送到广播地址的包将被传送到该网络中的所有计算机。
certificate authority, CA（证书颁发机构）	可信任的第三方组织或公司，可以颁发用于创建数字签名和公钥/私钥对的数字证书。CA 保证被授予唯一证书的个人的身份。
certificate revocation list, CRL（证书撤销列表）	已由 CA 撤销的公钥证书的列表。CRL 存储在 CRL 数据库中，该数据库通过 IKE 进行维护。
chain of trust（信任链）	在 X.509 证书中，证书颁发机构保证从 trust anchor（信任锚）到用户证书在内的证书可以提供不间断的验证链。
DES	Data Encryption Standard（数据加密标准）。一种对称密钥加密方法，开发于 1975 年，1981 年由 ANSI 标准化为 ANSI X.3.92。DES 使用 56 位密钥。

Diffie-Hellman algorithm（Diffie-Hellman 算法）	也称为“公钥”密码学。Diffie 和 Hellman 于 1976 年开发的非对称密钥一致性协议。使用该协议，两个用户可以在以前没有任何密钥的情况下通过不安全的介质交换密钥。IKE 协议需要使用 Diffie-Hellman。
digital signature（数字签名）	附加到以电子方式传输的消息的数字代码，可唯一地标识发送者。
distinguished name, DN（标识名）	一种使用普通字符串表示共享信息的标准化方法。标识名用在 LDAP 和 X.509 证书以及其他技术中。有关更多信息，请参见 A String Representation of Distinguished Names (http://www.ietf.org/rfc/rfc1779.txt)（标识名的字符串表示）。
domain of interpretation, DOI（系统解释域）	DOI 定义数据格式、网络通信流量交换类型和安全相关信息的命名约定。安全策略、加密算法和加密模式都属于安全相关信息。
DSA	Digital Signature Algorithm（数字签名算法）。一种公钥算法，采用大小可变（512 位到 4096 位）的密钥。美国政府标准 DSS 可达 1024 位。DSA 的输入依赖于 SHA-1。
dynamic packet filter（动态包过滤器）	请参见 stateful packet filter（有状态包过滤器）。
ECDSA	Elliptic Curve Digital Signature Algorithm（椭圆曲线数字签名算法）。一种基于椭圆曲线数学运算的公钥算法。在生成相同长度的签名时，所需的 ECDSA 密钥大小明显小于 DSA 公钥大小。
encapsulating security payload, ESP（封装安全有效负荷）	为包提供完整性和保密性的扩展头。ESP 是 IP 安全体系结构 (IPsec) 的五个组件之一。
encapsulation（封装）	在第一个包中放置头和有效负荷的过程，随后将第一个包放置在第二个包的有效负荷中。
firewall（防火墙）	将组织的专用网络或内联网与 Internet 隔离，从而防止它受到外部侵入的任何设备或软件。防火墙可以包括包过滤、代理服务器和 NAT（network address translation，网络地址转换）。
hash value（散列值）	一个从文本字符串生成的数字。使用散列函数可以确保已传输的消息未被篡改。MD5 和 SHA-1 都属于单向散列函数。
HMAC	用于进行消息验证的加密散列方法。HMAC 是密钥验证算法。HMAC 与重复加密散列函数（例如 MD5 或 SHA-1）以及机密共享密钥配合使用。HMAC 的加密能力取决于底层散列函数的特性。
ICMP echo request packet（ICMP 回显请求包）	发送到 Internet 上的计算机以要求响应的包。此类包通常称为 "ping" 包。
IKE	Internet Key Exchange（Internet 密钥交换）。IKE 用于自动为 IPsec 安全关联 (Security Association, SA) 提供经过验证的加密材料。

Internet Protocol, IP（Internet 协议）	在 Internet 上将数据从一台计算机发送到另一台计算机所用的方法或协议。
IP	请参见 Internet Protocol, IP（Internet 协议）、IPv4 和 IPv6。
IP header（IP 头）	唯一标识 Internet 包的二十字节数据。该头包括包的源地址和目标地址。头中存在一个选项，该选项允许添加更多字节。
IP in IP encapsulation（IP-in-IP 封装）	封装在 IP 包中的 IP 包的隧道传送机制。
IP link（IP 链路）	通信工具或介质，节点可以通过它在链路层上进行通信。链路层是紧邻 IPv4/IPv6 层的下一层。例如以太网（简单或桥接）或 ATM 网络。可以将一个或多个 IPv4 子网号或前缀指定给一个 IP 链路。不能将一个子网号或前缀指定给多个 IP 链路。在 ATM LANE 中，一个 IP 链路便是一个仿真 LAN。在使用 ARP 时，ARP 协议的范围是单个 IP 链路。
IP packet（IP 包）	通过 IP 传输的信息包。IP 包包含头和数据。头包括包的源地址和目标地址。头中的其他字段有助于标识和重新组合目标中包附带的数据。
IP stack（IP 栈）	TCP/IP 经常被称为“栈”。这是指数据交换的客户机端和服务器端的所有数据传送时所经过的各层（TCP 层、IP 层，有时还经过其他层）。
IPsec	IP security（IP 安全性）。为 IP 包提供保护的安全体系结构。
IPv4	Internet 协议版本 4IPv4 有时称为 IP。此版本支持 32 位地址空间。
IPv6	Internet 协议版本 6IPv6 支持 128 位地址空间。
key management（密钥管理）	管理安全关联 (security association, SA) 的方式。
keystore name（密钥库名称）	管理员为 network interface card, NIC（网络接口卡）上的存储区域（或密钥库）指定的名称。密钥库名称也称为令牌或令牌 ID。
label（标签）	1. IKEv2 规则的关键字，如果 `auth_method` 为 `preshared`，其值必须与预先共享密钥文件中 `label` 关键字的值匹配。 2. 创建 IKEv2 证书时使用的关键字。此值便于在密钥库中找到证书的所有部分（私钥、公钥和公钥证书）。 3. 对象或流程的敏感级别的强制访问控制 (mandatory access control, MAC) 指示。Confidential（机密）和 Top Secret（绝密）是标签样例。有标签的网络传输包含 MAC 标签。 4. IKEv1 规则的关键字，其值用于获取此规则。
link layer（链路层）	紧邻 IPv4/IPv6 的下一层。

link-local address（链路本地地址）	在 IPv6 中，用于在单个链路上寻址以实现诸如自动配置地址目的的标识。缺省情况下，链路本地地址是从系统的 MAC 地址创建的。
marker（标记器）	1. diffserv 体系结构和 IPQoS 中的一个模块，它使用指示包转发方式的值标记 IP 包的 DS 字段。在 IPQoS 实现中，标记器模块是 `dscpmk`。 2. IPQoS 实现中的一个模块，它使用用户优先级值标记以太网包的虚拟 LAN 标记。用户优先级值指示使用 VLAN 设备在网络中转发包的方式。此模块称为 `dlcosmk`。
MD5	一种重复加密散列函数，用于进行消息验证（包含数字签名）。该函数于 1991 年由 Rivest 开发。
message authentication code, MAC（消息验证代码）	MAC 可确保数据的完整性，并验证数据的来源。MAC 不能防止窃听。
multicast address（多播地址）	以特定方式标识一组接口的 IPv6 地址。发送到多播地址的包将被传送到组中的所有接口。IPv6 多播地址与 IPv4 广播地址具有类似的功能。
multihomed host（多宿主主机）	具有多个物理接口且不执行包转发的系统。多宿主主机可以运行路由协议。
NAT	请参见 network address translation, NAT（网络地址转换）。
network address translation, NAT（网络地址转换）	将一个网络中使用的 IP 地址转换为另一个网络中已知的不同 IP 地址的过程。用于限制所需的全局 IP 地址的数目。
network interface card, NIC（网络接口卡）	作为网络接口的网络适配卡。一些 NIC 可以具有多个物理接口，如 `igb` 卡。
packet filter（包过滤器）	一种防火墙功能，可以配置为允许或禁止指定的包通过防火墙。
packet header（包头）	请参见 IP header（IP 头）。
packet（包）	请参见 IP packet（IP 包）。
packet（包）	通过通信线路作为一个单位传输的一组信息。包含 IP header（IP 头）以及 payload（有效负荷）。
payload（有效负荷）	通过包传输的数据。有效负荷不包括将包传输到其目标所需的头信息。
perfect forward secrecy, PFS（完全正向保密）	在 PFS 中，不能使用保护数据传输的密钥派生其他密钥。此外，也不能使用保护数据传输的密钥的源派生其他密钥。因此，PFS 可防止解密以前记录的通信。 PFS 仅适用于经过验证的密钥交换。另请参见 Diffie-Hellman algorithm（Diffie-Hellman 算法）。

physical interface（物理接口）	系统与链路的连接。此连接通常作为设备驱动程序以及网络接口卡 (network interface card, NIC) 实现。一些 NIC 可以具有多个连接点，例如 `igb`。
PKI	Public Key Infrastructure（公钥基础结构）。由数字证书、证书颁发机构和其他注册机构组成的系统，用于检验和验证 Internet 事务中涉及的各方的有效性。
proxy server（代理服务器）	位于客户机应用程序（如 Web 浏览器）和另一个服务器之间的服务器。用于过滤请求 – 例如，阻止对某些 Web 站点的访问。
public key cryptography（公钥密码学）	一种加密系统，它使用两种不同的密钥。公钥对所有用户公开。私钥只对消息接收者公开。IKE 为 IPsec 提供公钥。
replay attack（重放攻击）	在 IPsec 中，侵入者捕获了包的攻击。存储的包稍后将替换或重复原先的包。为了避免遭到此类攻击，可以在包中包含一个字段，并使该字段在包的保护密钥的生命周期内递增。
router advertisement（路由器通告）	路由器通告其存在以及各种链路和 Internet 参数的过程，要么是定期进行通告，要么是作为对路由器请求消息的响应进行通告。
router discovery（路由器搜索）	主机查找驻留在已连接链路上的路由器的过程。
router solicitation（路由器请求）	主机请求路由器以立即（而非下一个预定时间）生成路由器通告的过程。
router（路由器）	通常具有多个接口、运行路由协议并转发包的系统。如果只有一个接口的系统是 PPP 链路的端点，则可以将该系统配置为路由器。
RSA	获取数字签名和公钥密码系统的方法。该方法于 1978 年首次由其开发者 Rivest、Shamir 和 Adleman 介绍。
SADB	Security Associations Database（安全关联数据库）。指定密钥和加密算法的表。在数据的安全传输中会使用这些密钥和算法。
security association, SA（安全关联）	指定从一个主机到另一个主机的安全属性的关联。
security parameter index, SPI（安全参数索引）	指定安全关联数据库 (security associations database, SADB) 中接收者应该用来对收到的包进行解密的行的一个整数。
security policy database, SPD（安全策略数据库）	指定应用于包的保护级别的数据库。SPD 对 IP 通信流量进行过滤，以确定一个包是应该被废弃、应该以明文方式进行传递还是应该用 IPsec 进行保护。
SHA-1	Secure Hashing Algorithm（安全散列算法）。该算法可以针对长度小于 2^{64} 的任何输入进行运算，以生成消息摘要。SHA-1 算法是 DSA 的输入。

smurf attack（smurf 攻击）	使用从远程位置定向到一个 IP broadcast address（广播地址）或多个广播地址的 ICMP 回显请求包以造成严重的网络拥塞或故障。
sniff（探查）	在计算机网络中窃听 – 通常作为自动化程序的一部分，以便从线路中筛选出信息，如明文口令。
spoof（电子欺骗）	使用一个 IP 地址（该地址指示消息来自受信任主机）向计算机发送消息，以获取对该计算机的未经授权的访问。要进行 IP 电子欺骗，黑客必须先使用各种方法查找受信任主机的 IP 地址，然后修改包头以便使这些包看起来像是来自该主机。
stateful packet filter（有状态包过滤器）	可以监视活动连接的状态和使用获取的信息确定允许哪些网络包通过 packet filter（包过滤器）的 firewall（防火墙）。通过跟踪和匹配请求与回复，有状态包过滤器可以筛选出与请求不匹配的回复。
stream control transport protocol, SCTP（流控制传输协议）	以与 TCP 类似的方式提供面向连接的通信的传输层协议。此外，SCTP 还支持连接多宿主，即连接的端点之一可以具有多个 IP 地址。
symmetric key cryptography（对称密钥密码学）	一种加密系统，其中消息的发送者和接收者共享一个公用密钥。此公用密钥用于对消息进行加密和解密。对称密钥用于对在 IPsec 中大量传输的数据进行加密。AES 是一个对称密钥的示例。
TCP/IP	TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/Internet 协议）是 Internet 的基本通信语言或协议。它还可以在专用网络（内联网或外联网）中用作通信协议。
Triple-DES（三重 DES）	Triple-Data Encryption Standard（三重数据加密标准）。一种对称密钥加密方法。三重 DES 要求密钥长度为 168 位。三重 DES 也写作 3DES。
trust anchor（信任锚）	在 X.509 证书中，来自证书颁发机构的根证书。从根证书到最终证书在内的证书可以建立一个信任链。
tunnel（隧道）	packet（包）在封装期间采用的路径。请参见 encapsulation（封装）。 在 IPsec 中，已配置的隧道是点对点接口。使用隧道，可以将一个 IP 包封装到另一个 IP 包中。
virtual LAN (VLAN) device（虚拟 LAN 设备）	在 IP 协议栈的以太网（数据链路）级别上提供通信流量转发的网络接口。
virtual network interface, VNIC（虚拟网络接口）	提供虚拟网络连通性（不论是否是在物理网络接口上配置的）的伪接口。容器（如专用 IP 区域）在 VNIC 上配置以形成虚拟网络。
virtual network（虚拟网络）	软件和硬件网络资源以及作为单个软件项同时管理的功能组合。内部虚拟网络将网络资源整合到单个系统，有时称为“网络集成 (network in a box)”。

virtual private network, VPN（虚拟专用网络）	单个安全逻辑网络，使用跨公共网络（如 Internet）的隧道进行传输。

索引

B

G

H

I

Q

R

S

T

Z