在 Oracle® Solaris 11.1 中保护网络安全

文件号码 E36648–02
2013 年 2 月

ORACLE

130326@25097

目录

前言

该指南假设已安装 Oracle Solaris 操作系统 (Oracle Solaris OS) 并且您已为保护网络做好准备。

注 – 此 Oracle Solaris 发行版支持使用 SPARC 和 x86 系列处理器体系结构的系统。支持的系统可以在 Oracle Solaris OS: Hardware Compatibility Lists（Oracle Solaris OS：硬件兼容性列表）中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

目标读者

本书适用于所有负责管理运行 Oracle Solaris 的联网系统的人员。要使用本书，您应当至少具备两年的 UNIX 系统管理经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info`，或访问 `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs`（如果您听力受损）。

印刷约定

下表介绍了本书中的印刷约定。

表 P–1 印刷约定

字体或符号	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出	编辑 `.login` 文件。 使用 `ls -a` 列出所有文件。 `machine_name% you have mail.`

表 P–1 印刷约定 （续）

字体或符号	含义	示例
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	`machine_name% su` `Password:`
aabbcc123	要使用实名或值替换的命令行占位符	删除文件的命令为 `rm` *filename*。
AaBbCc123	保留未译的新词或术语以及要强调的词	这些称为 *Class* 选项。 **注意**：有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	**高速缓存**是存储在本地的副本。 请**勿**保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P–2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	`$`
Bash shell、Korn shell 和 Bourne shell 超级用户	`#`
C shell	`machine_name%`
C shell 超级用户	`machine_name#`

◆ ◆ ◆ 第 1 章

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：

链路保护概述

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 machines, 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)` 手册页中所指定的那样对 `allowed-dhcp-cids` 列表中的项进行格式化。

`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 – 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，请参见第 15 页中的“如何查看链路保护配置和统计信息”。

配置链路保护（任务列表）

要使用链路保护，请设置链路的 `protection` 属性。如果保护类型适用于其他配置文件，例如 `ip-nospoof` 和 `allowed-ips` 或 `dhcp-nospoof` 和 `allowed-dhcp-cids`，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。

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。	第 13 页中的“如何启用链路保护”
禁用链路保护。	删除链路保护。	第 13 页中的“如何禁用链路保护”
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。	第 14 页中的“如何指定 IP 地址以防止受到 IP 欺骗”

任务	说明	参考
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。	第 14 页中的 “如何指定 DHCP 客户机以防止受到 DHCP 欺骗”
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。	第 15 页中的 “如何查看链路保护配置和统计信息”

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1 查看可用链路保护类型。

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

有关可能的类型的说明，请参见第 11 页中的 “链路保护类型”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，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
```

3 验证是否已启用链路保护。

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

`VALUE` 下的链路保护类型表明已启用保护。

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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    DEFAULT    POSSIBLE
   vnic0     protection    rw   --       --         mac-nospoof,
                                                     restricted,
                                                     ip-nospoof,
                                                     dhcp-nospoof
   ```

   VALUE 下没有列出链路保护类型表明已禁用链路保护。

▼ 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，如第 13 页中的 “如何启用链路保护”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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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          DEFAULT      POSSIBLE
   link     protection     rw      ...
                                   ip-nospoof                  ip-nospoof
   ```

   VALUE 下的 ip-nospoof 列表表明已启用该保护类型。

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，如第 13 页中的 “如何启用链路保护”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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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          DEFAULT       POSSIBLE
link     protection      rw      ...
                                 dhcp-nospoof                 dhcp-nospoof
```

VALUE 下的 dhcp-nospoof 列表表明已启用该保护类型。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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           DEFAULT     POSSIBLE
vnic0    protection          rw       mac-nospoof     --          mac-nospoof,
                                      restricted                  restricted,
                                      ip-nospoof                  ip-nospoof,
                                      dhcp-nospoof                dhcp-nospoof
vnic0    allowed-ips         rw       10.0.0.1,       --          --
                                      10.0.0.2
vnic0    allowed-dhcp-cids   rw       hello           --          --
```

注 – 仅当启用 ip-nospoof 时才使用 allowed-ips 属性，其列在 VALUE 下。仅当启用 dhcp-nospoof 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（任务）

本章说明如何调优影响 Oracle Solaris 安全的网络参数。

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。	第 18 页中的 “如何禁用网络路由选择守护进程”
防止散播有关网络拓扑的信息。	防止广播包。	第 18 页中的 “如何禁用广播包转发”
	阻止对广播回显请求和多播回显请求的响应。	第 19 页中的 “如何禁用回显请求的响应”
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。	第 20 页中的 “如何设置严格多宿主”
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。	第 20 页中的 “如何设置不完整 TCP 连接的最大数目”
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。	第 21 页中的 “如何设置暂挂 TCP 连接的最大数目”
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。	第 21 页中的 “如何为初始 TCP 连接指定强随机数”
防止 ICMP 重定向。	删除网络拓扑的指示符。	第 21 页中的 “如何禁止 ICMP 重定向”
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。	第 22 页中的 “如何将网络参数重置为安全值”

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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, 2011 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, 2011 10:10:10 AM PDT
Reason: Disabled by an administrator.
   See: http://support.oracle.com/msg/SMF-8000-05
   See: in.routed(1M)
Impact: This service is not running.
```

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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 11.1 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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 11.1 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 VPN。

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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 11.1 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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 11.1 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

以下过程设置 TCP 初始序列号生成参数以遵守 RFC 6528 (`http://www.ietf.org/rfc/rfc6528.txt`)。

开始之前 您必须是分配有 `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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1. **更改 `TCP_STRONG_ISS` 变量的缺省值。**

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

2. **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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

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

有关更多信息，请参见《Oracle Solaris 11.1 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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 11.1 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 11.1 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

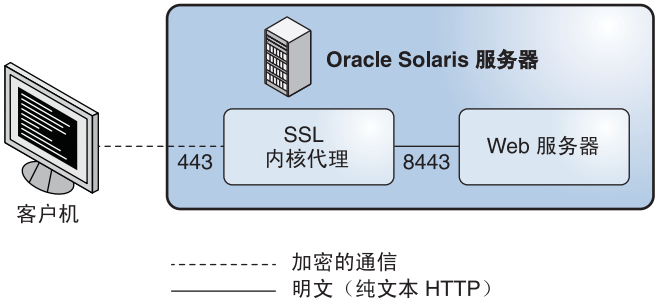
3 Web 服务器和安全套接字层协议

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。

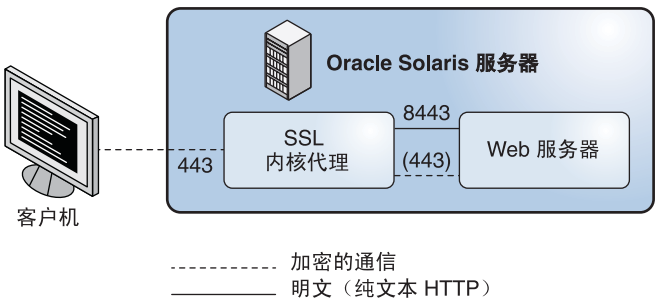
- 该代理加速了服务器应用程序（如 Web 服务器）的 SSL 性能，因此可提供比依赖用户级 SSL 库的应用程序更加优越的性能。性能提高可能超过 35%，这取决于应用程序的工作负荷。
- SSL 内核代理是透明的。它没有指定的 IP 地址。因此，Web 服务器可看到真正的客户机 IP 地址和 TCP 端口。
- SSL 内核代理和 Web 服务器可协同工作。

  图 3–1 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 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（任务）

以下过程说明如何配置 Web 服务器以使用 SSL 内核代理：

- 第 27 页中的 “如何配置 Apache 2.2 Web 服务器以使用 SSL 内核代理”
- 第 28 页中的 “如何配置 Oracle iPlanet Web Server 以使用 SSL 内核代理”
- 第 30 页中的 “如何配置 SSL 内核代理以回退到 Apache 2.2 SSL”

- 第 32 页中的“如何使用区域中的 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 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 PKCS#11-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 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：

```
# 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 中所示的简单方案。

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

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《Oracle Solaris 11.1 管理：安全服务》中的"如何使用指定给您的管理权限"。

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，请参见第 27 页中的 "如何配置 Apache 2.2 Web 服务器以使用 SSL 内核代理"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 PKCS#11-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 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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
```

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 – 口令短语不能包含空格。

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 上侦听纯文本。**

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 kernel proxy 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
```

示例 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。使用 `ksslcfg` 命令在全局区域中配置服务实例后，便可以启动本地区域 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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

Oracle Solaris 中的 IP 过滤器（概述）

本章概述 Oracle Solaris 中的 IP 过滤器功能。有关 IP 过滤器的任务，请参见第 5 章，IP 过滤器（任务）。

本章包含以下信息：

IP 过滤器简介

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。

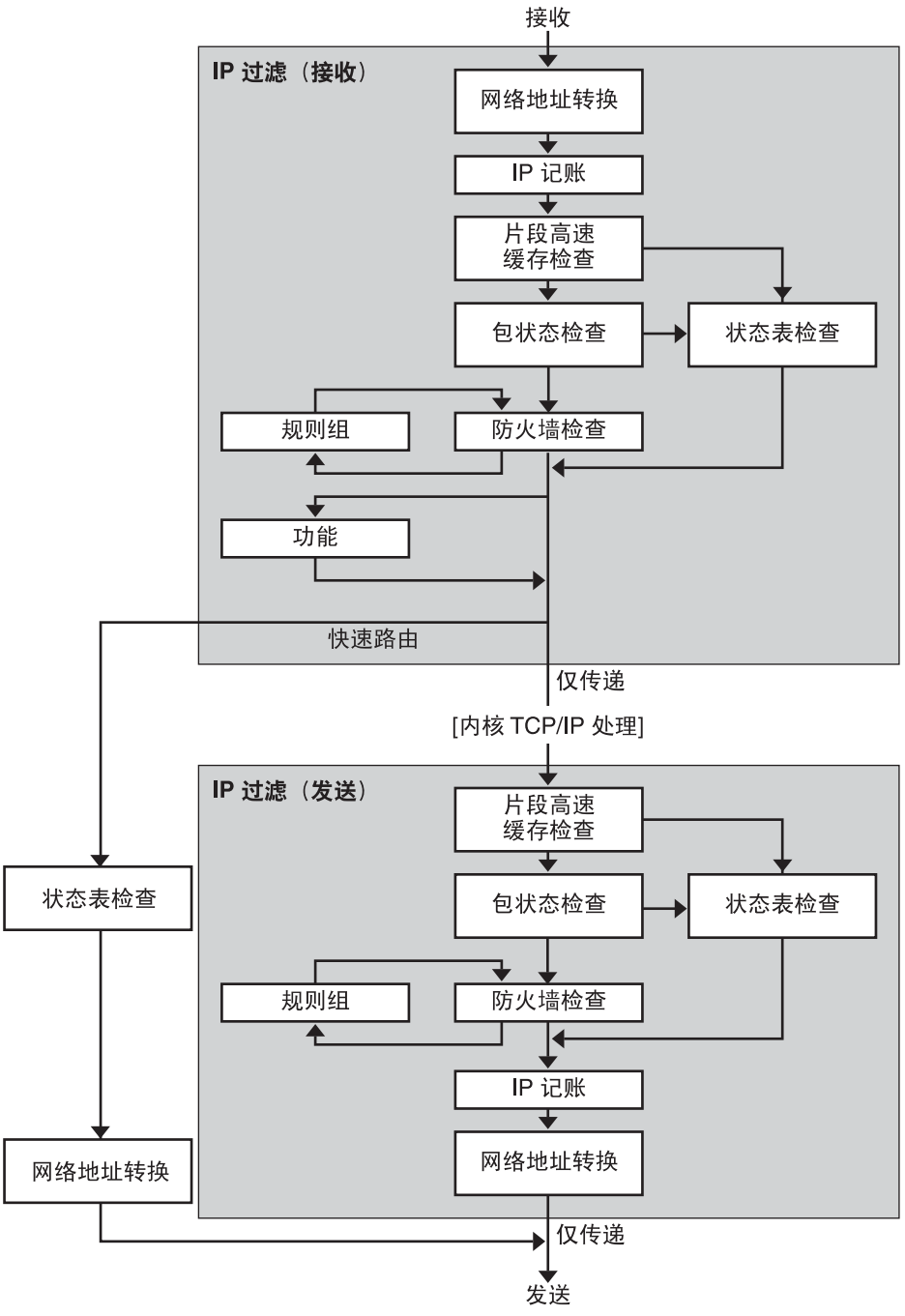
开源 IP 过滤器的信息源

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
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
状态表检查
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 SMF 的完整概述，请参见《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 章 “管理服务（概述）”。有关与 SMF 关联的逐步过程的信息，请参见《在 Oracle Solaris 11.1 中管理服务和故障》中的第 2 章 “管理服务（任务）”。
- 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，请参见第 47 页中的 “配置 IP 过滤器”。
- 要管理 IP 过滤器，必须能够承担 root 角色或拥有 IP Filter Management（IP 过滤器管理）权限配置文件的角色。可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《Oracle Solaris 11.1 管理：安全服务》中的“初次配置 RBAC（任务列表）”。
- 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，请参见第 70 页中的 “IP 过滤器配置文件示例”。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

此外，还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。地址池有助于加快规则处理速度，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. 操作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*　复制包并将 *interface-name* 上的副本向外发送到随意指定的 IP 地址。

`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` 传入的通信流量，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，请参见第 53 页中的 “管理 IP 过滤器的包过滤规则集合”。有关示例中所示的 IP 地址方案 (192.168.0.0/16) 的说明，请参见《配置和管理 Oracle Solaris 11.1 网络》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。但是，不能在单个规则中指定两种类型的地址。相反，必须为每种地址类型设置单独的规则。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

地址池可建立用于命名一组地址/网络掩码对的单个引用。地址池提供可减少将 IP 地址与规则相匹配的时间的进程，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。必须创建文件并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。此时，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。要了解有关其他 IPv6 功能的更多信息，请参见《System Administration Guide: IP Services》中的"Major Features of IPv6"。

注 – 有关 jumbogram 的更多信息，请参阅 Internet 工程任务组 (Internet Engineering Task Force, IETF) 的文档 "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 IPv6 的更多信息，请参见《System Administration Guide: IP Services》中的第 3 章 "Introducing IPv6 (Overview)"。有关与 IP 过滤器相关的任务，请参见第 5 章，IP 过滤器（任务）。

IP 过滤器手册页

下表介绍与 IP 过滤器相关的手册页文档。

手册页	说明
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

IP 过滤器（任务）

本章提供有关任务的逐步说明。有关 IP 过滤器的概述信息，请参见第 4 章，Oracle Solaris 中的 IP 过滤器（概述）。

本章包含以下信息：

配置 IP 过滤器

以下任务列表提供了用于创建 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（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。	第 48 页中的“如何显示 IP 过滤器服务缺省值”
通过 NAT 和地址池为网络通信和包定制包过滤规则集合。	第 48 页中的“如何创建 IP 过滤器配置文件”
启用、刷新或禁用 IP 过滤器服务。	第 50 页中的“如何启用和刷新 IP 过滤器”
修改到达段的包的缺省设置。	第 50 页中的“如何禁用包重组”
过滤系统上区域之间的流量。	第 51 页中的“如何启用回送过滤”
停止使用 IP 过滤器。	第 52 页中的“如何禁用包过滤”

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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建议的文件位置。这些文件不存在，直到您创建它们。您可以通过更改配置文件属性值来更改该文件的位置。有关过程，请参见第 48 页中的“如何创建 IP 过滤器配置文件”。

在定制自己的包过滤规则时修改第四个文件属性。请参见第 48 页中的“如何创建 IP 过滤器配置文件”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，请参见第 48 页中的“如何创建 IP 过滤器配置文件”。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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，请参见第 39 页中的 “使用 IP 过滤器的包过滤功能”。有关配置文件的示例，请参见第 70 页中的 “IP 过滤器配置文件示例”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，请使用相应的名称（如 /etc/ipf/ipnat.conf）为 NAT 规则创建文件。要更改该名称，请更改 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，请参见第 42 页中的 “使用 IP 过滤器的 NAT 功能”。

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，请使用相应的名称（如 /etc/ipf/ippool.conf）为池创建文件。要更改该名称，请更改 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，请参见第 43 页中的 “使用 IP 过滤器的地址池功能”。

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。请参见第 51 页中的 “如何启用回送过滤”。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，请参见第 50 页中的 “如何禁用包重组”。

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

已完成第 48 页中的 “如何创建 IP 过滤器配置文件”。

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，请参见第 52 页中的 “使用 IP 过滤器规则集合”。

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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：

```
# 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 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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` 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，请参见第 50 页中的“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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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，请参见第 52 页中的“使用 IP 过滤器规则集合”。

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

以下任务列表提供了与 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。	第 53 页中的 “如何查看活动的包过滤规则集合”
查看非活动的包过滤规则集合。	第 54 页中的 “如何查看非活动的包过滤规则集合”
激活不同的活动规则集合。	第 54 页中的 “如何激活不同的或更新的包过滤规则集合”
删除规则集合。	第 55 页中的 “如何删除包过滤规则集合”
将规则添加到规则集合。	第 56 页中的 “如何将规则附加到活动的包过滤规则集合” 第 57 页中的 “如何将规则附加到非活动的包过滤规则集合”
在活动和非活动的规则集合之间切换。	第 57 页中的 “如何在活动和非活动的包过滤规则集合之间切换”
从内核中删除非活动规则集合。	第 58 页中的 “如何从内核中删除非活动的包过滤规则集合”
查看活动的 NAT 规则。	第 59 页中的 “如何查看 IP 过滤器中的活动 NAT 规则”
删除 NAT 规则。	第 59 页中的 “如何取消激活 IP 过滤器中的 NAT 规则”
将规则添加到活动的 NAT 规则。	第 60 页中的 “如何将规则附加到 NAT 包过滤规则”
查看活动的地址池。	第 61 页中的 “如何查看活动地址池”
删除地址池。	第 61 页中的 “如何删除地址池”
将规则添加到地址池。	第 62 页中的 “如何将规则附加到地址池”

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，如第 48 页中的 “如何显示 IP 过滤器服务缺省值”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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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。

示例 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
```

示例 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。

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

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

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

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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。

示例 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调试时可能非常有用。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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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：
    a. 在所选的文件中创建规则集合。
    b. 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。

示例 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调试时可能非常有用。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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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。

示例 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调试时可能非常有用。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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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。

示例 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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。

示例 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

使用以下过程可以管理、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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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。

示例 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调试时可能非常有用。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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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：
    a. 在所选的文件中创建其他 NAT 规则。
    b. 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。

示例 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

使用以下过程可以管理、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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **删除当前地址池中的项。**

```
$ ippool -F
```

示例 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调试时可能非常有用。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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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：

  a. 在所选的文件中创建其他地址池。

  b. 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。

请按照第 56 页中的 “如何将规则附加到活动的包过滤规则集合”中的说明操作。

注 – 请不要刷新或重启 IP 过滤器服务，否则将丢失已添加的地址池规则。

示例 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 过滤器的统计信息（任务列表）

任务	参考
查看状态表。	第 63 页中的 “如何查看 IP 过滤器的状态表”
查看包状态统计信息。	第 64 页中的 “如何查看 IP 过滤器的状态统计信息”
列出 IP 过滤器的可调参数。	第 65 页中的 “如何查看 IP 过滤器的可调参数”
查看 NAT 统计信息。	第 65 页中的 “如何查看 IP 过滤器的 NAT 统计信息”
查看地址池统计信息。	第 65 页中的 “如何查看 IP 过滤器的地址池统计信息”

▼ 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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。

示例 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **查看状态统计信息。**

```
$ ipfstat -s
```

示例 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **查看 NAT 统计信息。**

```
$ ipnat -s
```

示例 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **查看地址池统计信息。**

```
$ ippool -s
```

示例 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。	第 66 页中的 “如何为 IP 过滤器设置日志文件”
查看状态日志文件、NAT 日志文件和常规日志文件。	第 67 页中的 “如何查看 IP 过滤器的日志文件”
刷新包日志缓冲区。	第 68 页中的 “如何刷新包日志缓冲区”
将记录的包保存到文件中，以供日后参考。	第 69 页中的 “如何将记录的包保存到文件中”

▼ 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记录在 `syslogd` 文件中。创建日志文件来单独记录 IP 过滤器流量信息以将其与可能记录在缺省日志文件中的其他数据相区分不失为一个良好做法。

开始之前 您必须承担 `root` 角色。

1 确定联机的 `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` 服务联机，请刷新 `system-log:rsyslog` 服务实例。

示例 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。

在 `/etc/syslog.conf` 中：

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

在命令行中：

```
# 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 已完成第 66 页中的“如何为 IP 过滤器设置日志文件”。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **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 命令还会打开 ipmon 的另一个副本。在此情况下，两个副本将读取相同的日志信息，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。

示例 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

```
# 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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **刷新包日志缓冲区。**

```
# ipmon -F
```

示例 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。即使未在日志文件中存储任何内容，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

您可以在调试过程中或希望手动审计流量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。

示例 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。在行尾使用反斜杠 (\) 以指示规则在下一行上继续。

有关详细的语法信息，请参见第 40 页中的 “配置包过滤规则”。

示例 5–20 IP 过滤器主机配置

此示例说明具有 `net0` 网络接口的主机上的配置。

```
# 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。

示例 5–21 IP 过滤器服务器配置

此示例显示用作 Web 服务器的主机的配置。此计算机具有 `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 ***
```

示例 5–21 IP 过滤器服务器配置 （续）

```
# 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
```

示例 5–21 IP 过滤器服务器配置 （续）

```
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
```

示例 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 aren't on the loopback
# interface should not exist.
block in log quick on net0 from 127.0.0.0/8 to any
block in log quick on net0 from any to 127.0.0.0/8
block in log quick on net1 from 127.0.0.0/8 to any
block in log quick on net1 from any to 127.0.0.0/8


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

示例 5–22 IP 过滤器路由器配置 （续）

```
# 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
```

示例 5–22 IP 过滤器路由器配置 （续）

```
# 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

IP 安全体系结构（概述）

IP 安全体系结构 (IPsec) 为 IPv4 和 IPv6 网络包中的 IP 数据报提供加密保护。

本章包含以下信息：

有关如何在网络中实现 IPsec 的信息，请参见第 7 章，配置 IPsec（任务）。有关参考信息，请参见第 8 章，IP 安全体系结构（参考信息）。

IPsec 介绍

IPsec 通过验证包、加密包或同时执行这两种操作来保护 IP 包。IPsec 在 IP 模块内执行。因此，Internet 应用程序可以直接利用 IPsec，而不必配置自身以使用 IPsec。若使用得当，IPsec 是保证网络通信安全的有效工具。

IPsec 保护涉及以下主要组件：

- **安全协议**－IP 数据报保护机制。authentication header（验证头）(AH) 包括 IP 包的散列并确保完整性。数据报的内容没有加密，但是可以向接收者保证包的内容尚未更改，还可以向接收者保证包已由发送者发送。encapsulating security payload, ESP（封装安全有效负荷）对 IP 数据进行加密，因此在包传输过程中会遮蔽内容。ESP 还可以通过验证算法选项来确保数据的完整性。
- **安全关联 (security association, SA)**－应用于特定网络通信流的加密参数和 IP 安全协议。每个 SA 都有一个称为安全参数索引 (Security Parameters Index, SPI) 的唯一引用。
- **安全关联数据库 (Security associations database, SADB)**－将安全协议与 IP 目标地址和索引号进行关联的数据库。索引号称为 security parameter index, SPI（安全参数索引）。这三个元素（安全协议、目标地址和 SPI）会唯一标识合法的 IPsec 包。此数据库确保到达包目的地的受保护包可由接收者识别。接收者还可使用数据库中的信息解密通信、检验包未曾受到更改、重新组装包并将包发送到其最终目的地。
- **密钥管理**－针对加密算法和 SPI 生成和分发密钥。
- **安全机制**－用于保护 IP 数据报中的数据的验证和加密算法。
- **安全策略数据库 (Security policy database, SPD)**－用于指定要应用到包的保护级别的数据库。SPD 过滤 IP 通信来确定应该如何处理包。包可能被废弃，可以毫无阻碍地进行传送，或者也可以受到 IPsec 的保护。对于外发包，SPD 和 SADB 确定要应用的保护级别。对于传入包，SPD 帮助确定包的保护级别是否可接受。如果包受 IPsec 保护，将在对包进行解密和验证之后参考 SPD。

IPsec 将安全机制应用于发往 IP 目标地址的 IP 数据报。接收者使用其 SADB 中的信息来检验到达的包是否合法并对其进行解密。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 数据报。

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 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 (Requests for Comment, RFC)。所有 RFC 均受 Internet 协会版权保护。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 (IKE)"，1998 年 11 月
- RFC 3554，"On the Use of Stream Control Transmission Protocol (SCTP) with IPsec"，2003 年 7 月

IPsec 术语

IPsec RFC 定义许多用于识别何时在系统上实现 IPsec 的术语。下表列出了 IPsec 术语，提供了常用的首字母缩略词并定义了每个术语。有关在密钥协商中使用的术语的列表，请参见表 9–1。

表 6–1 IPsec 术语、首字母缩略词和用法

IPsec 术语	首字母缩略词	定义
Security association（安全关联）	SA	应用于特定网络通信流的加密参数和 IP 安全协议。SA 由三个元素定义：安全协议、唯一安全参数索引 (security parameter index, SPI) 和 IP 目标。
Security associations database（安全关联数据库）	SADB	包含所有活动的安全关联的数据库。
Security parameter index（安全参数索引）	SPI	安全关联的索引值。SPI 是可以将具有相同 IP 目标和安全协议的 SA 区分开来的 32 位的值。
Security policy database（安全策略数据库）	SPD	确定外发包和传入包是否具有指定的保护级别的数据库。
Key exchange（密钥交换）		使用非对称加密算法生成密钥的过程。两种主要方法是 RSA 和 Diffie-Hellman。
Diffie-Hellman	DH	用于密钥生成和密钥验证的密钥交换算法。通常称为**经过验证的密钥交换**。
RSA	RSA	用于密钥生成和密钥分发的密钥交换算法。此协议以其三个创建者 Rivest、Shamir 和 Adleman 命名。

表 6–1 IPsec 术语、首字母缩略词和用法 （续）

IPsec 术语	首字母缩略词	定义
Internet Security Association and Key Management Protocol（Internet 安全关联和密钥管理协议）	ISAKMP	用于建立 SA 属性格式以及协商、修改和删除 SA 的通用框架。ISAKMP 是处理 IKE 交换的 IETF 标准。

IPsec 包流

图 6–1显示了当已经在外发包上调用 IPsec 时，作为IP datagram（IP 数据报）一部分的带有 IP 地址的包如何继续传送。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。如何应用这些实体以及如何选择算法将在后续各节中进行介绍。

图 6–2 显示了 IPsec 传入过程。

图 6–1 应用于外发包过程的 IPsec

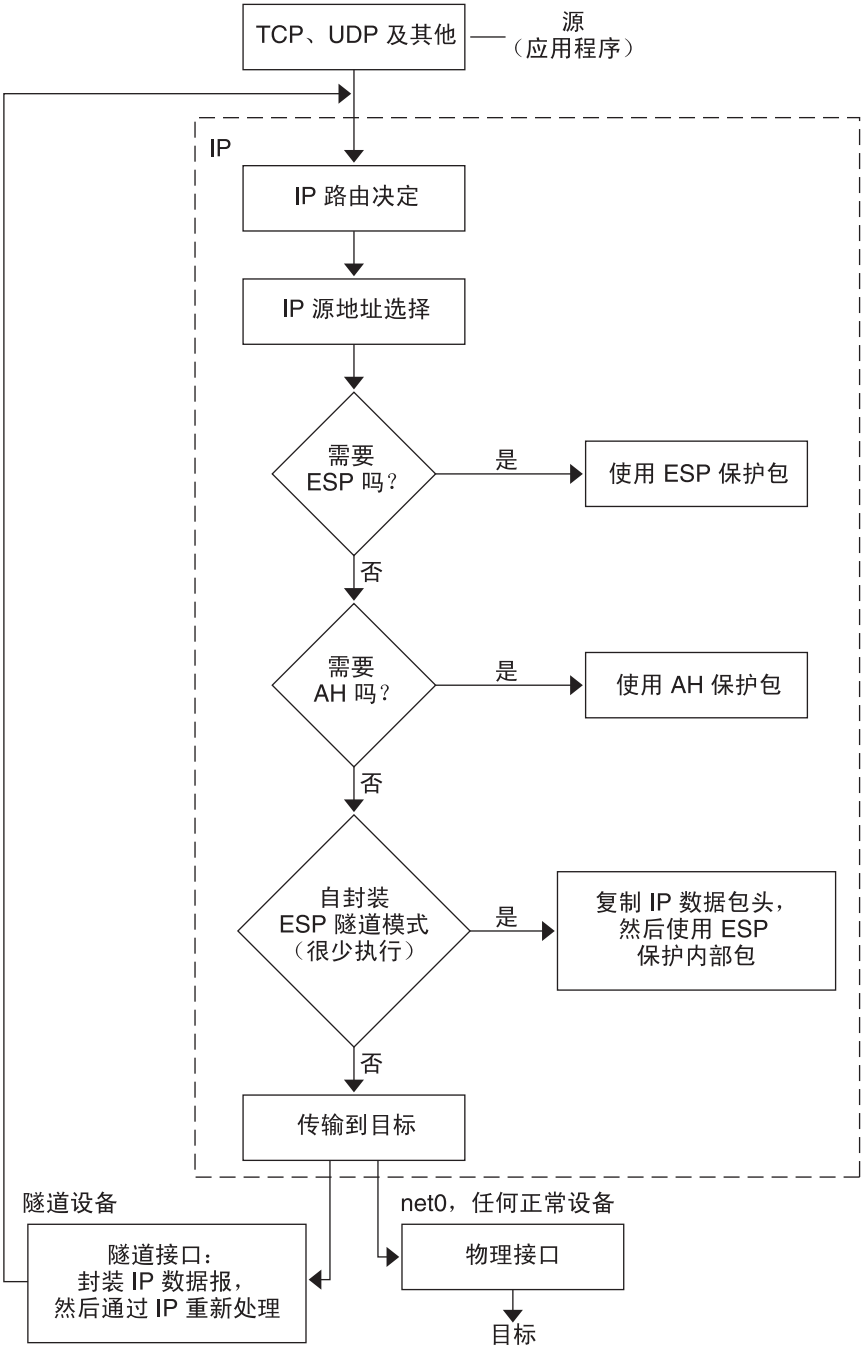
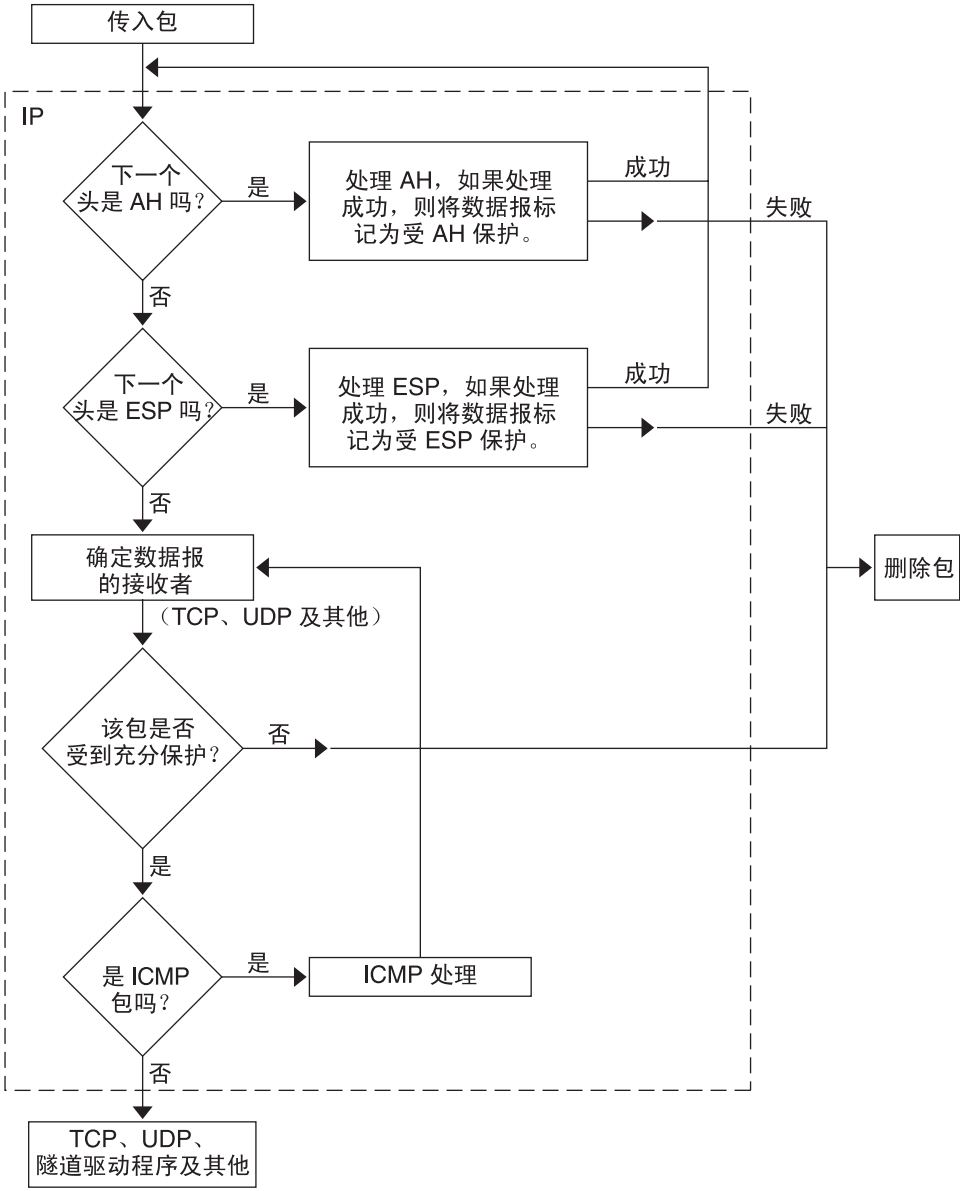


图 6–2 应用于传入包过程的 IPsec



IPsec 安全关联

IPsec **安全关联** (security association, SA) 指定由通信主机识别的安全属性。单个 SA 保护单一方向的数据，此保护针对单个主机或一组（多播）地址。由于多数通信为对等通信或客户机/服务器通信，因此，必须存在两个 SA 来保证两个方向的通信安全。

以下三个元素唯一地标识 IPsec SA :

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

SPI 是任意 32 位的值，与 AH 或 ESP 包一起传输。`ipsecah(7P)` 和 `ipsecesp(7P)` 手册页说明了由 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` 使特权应用程序可以管理数据库。例如，IKE 应用程序和 `ipseckeys` 命令会使用 `PF_KEY` 套接字接口。

- 有关 IPsec SADB 的更为完整的说明，请参见第 114 页中的 “IPsec 的安全关联数据库”。
- 有关如何管理 SADB 的更多信息，请参见 `pf_key(7P)` 手册页。

IPsec 中的密钥管理

安全关联 (Security association, SA) 需要加密材料来进行验证和加密。对此加密材料的管理称为**密钥管理**。Internet 密钥交换 (Internet Key Exchange, IKE) 协议自动处理密钥管理。您还可以使用 `ipseckey` 命令手动管理密钥。

IPv4 和 IPv6 包上的 SA 可以使用任一密钥管理方法。除非您有充分的理由使用手动密钥管理，否则，请首选使用 IKE。

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:default` 服务－为用于进行自动密钥管理的 SMF 服务。`ike` 服务运行 `in.iked` 守护进程以提供自动密钥管理。有关 IKE 的说明，请参见第 9 章，Internet 密钥交换（概述）。有关 `in.iked` 守护进程的更多信息，请参见 `in.iked(1M)` 手册页。有关 `ike` 服务的信息，请参见第 155 页中的 “IKE 服务”。
- `svc:/network/ipsec/manual-key:default` 服务－为用于进行手动密钥管理的 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，请参见第 114 页中的 “IPsec 中用于生成 SA 的实用程序”。有关 `ipseckey` 命令选项的详细说明，请参见 `ipseckey(1M)` 手册页。

IPsec 保护机制

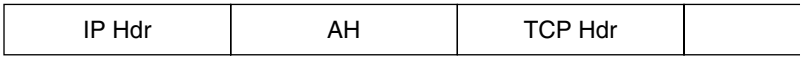
IPsec 提供了两种用于保护数据的安全协议：

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

AH 使用验证算法来保护数据。ESP 使用加密算法来保护数据。ESP 应只与验证机制一起使用。如果不将遍历 NAT，可以将 ESP 与 AH 结合使用。或者，可以将验证算法和加密机制与 ESP 一起使用。组合的模式算法（例如 AES-GCM）在单一算法中提供加密和验证。

验证头

authentication header（验证头）为数据报提供了数据验证、高完整性以及重放保护。AH 保护 IP 数据报的更为重要的部分。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）模块为 ESP 所封装的内容提供了保密性。ESP 也提供 AH 提供的服务。但是，ESP 仅为 ESP 所封装的数据报部分提供保护。ESP 提供可选的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。

由于 ESP 封装其数据，因此 ESP 仅保护数据报中跟在其后的数据，如下图所示。

加密的

在 TCP 包中，ESP 仅封装 TCP 数据包头及其数据。如果包是 IP-in-IP 数据报，则 ESP 会保护内部的 IP 数据报。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。

如果设置了自封装，会生成 IP 数据包头的副本来构建 IP-in-IP 数据报。例如，如果未在 TCP 套接字上设置自封装，会以下列格式发送数据报：

```
[ 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数据报：

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

有关进一步介绍，请参见第 84 页中的 “IPsec 中的传输模式和隧道模式”。

使用 AH 和 ESP 时的安全注意事项

下表比较了由 AH 和 ESP 提供的保护。

表 6–2 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 数据包头到传输层头的内容	提供高完整性、数据验证： ■ 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	重放、剪贴
ESP	保护数据报中紧跟在 ESP 之后的包。	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 模块使用验证算法。ESP 模块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 使用加密框架访问算法。加密框架为算法提供了一个中心系统信息库，同时还提供了其他服务。使用此框架，IPsec 可以利用高性能的加密硬件加速器。

有关更多信息，请参见以下内容：

- 《Oracle Solaris 11.1 管理：安全服务》中的第 11 章 “加密框架（概述）”
- 《Oracle Solaris 11 开发者安全性指南》中的第 8 章 “Oracle Solaris 加密框架介绍”

IPsec 中的验证算法

验证算法将生成完整性校验和值或基于数据和密钥的**摘要**。AH 模块使用验证算法。ESP 模块也可以使用验证算法。

IPsec 中的加密算法

加密算法使用密钥来加密数据。IPsec 中的 ESP 模块使用加密算法。算法以**块大小**为单位对数据进行操作。

IPsec 保护策略

IPsec 保护策略可以使用任何安全机制。IPsec 策略可以在以下级别应用：

- 在系统范围级别
- 在每个套接字级别

IPsec 会将系统范围的策略应用于外发数据报和传入数据报。外发数据报既可以在受保护的情况下发送，也可以在不受保护的情况下发送。如果应用了保护，则算法可能是特定的，也可能是非特定的。由于存在系统可识别的其他数据，因此可以将其他一些规则应用于外发数据报。传入数据报可以被接受或丢弃。确定丢弃还是接受传入数据报时取决于若干个条件，这些条件有时会重叠或冲突。可以通过确定首先要解析的规则来解决冲突。将自动接受通信，但是当策略项表明通信应绕过所有其他策略时除外。

可以绕过通常保护数据报的策略。您既可以在系统范围策略内指定例外，也可以在每个套接字策略中请求绕过。对于系统内的通信，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。

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。

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**。模式不影响包的编码。在每种模式下，包受 AH、ESP，或二者的保护。如果内部包是 IP 包，这两种模式在策略应用程序方面有所不同，如下所示：

- 在传输模式下，外部头决定保护内部 IP 包的 IPsec 策略。
- 在隧道模式下，内部 IP 包决定保护其内容的 IPsec 策略。

在传输模式下，外部头、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。类似地，如果下一个头是 IP 数据包头，外部头和内部 IP 数据包头可用于决定 IPsec 策略。

隧道模式仅适用于 IP-in-IP 数据报。如果在家中的计算机用户要连接到中心计算机位置，以隧道模式进行隧道连接将会很有用。在隧道模式下，IPsec 策略强制实施于内部 IP 数据报的内容中。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 数据包头、其下一个头及下一个头支持的端口，可以强制实施策略。与传输模式不同，在隧道模式下，外部 IP 数据包头不指示其内部 IP 数据报的策略。

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，请参见第 97 页中的 “使用 IPsec 保护 VPN”。

在 Oracle Solaris 中，只能在 IP 隧道连接网络接口上强制执行隧道模式。有关隧道连接接口的更多信息，请参见《配置和管理 Oracle Solaris 11.1 网络》中的第 6 章 “配置 IP 隧道”。`ipsecconf` 命令提供 `tunnel` 关键字来选择 IP 隧道连接网络接口。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。

在传输模式下，ESP、AH、或二者可以保护该数据报。

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

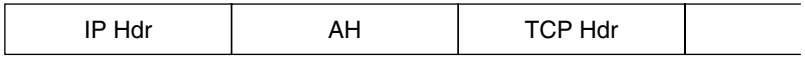
在传输模式下，ESP 按下图所示的方式保护数据。阴影部分表示包的加密部分。

图 6–4 携带 TCP 信息的受保护的 IP 包

加密的

在传输模式下，AH 按下图所示的方式保护数据。

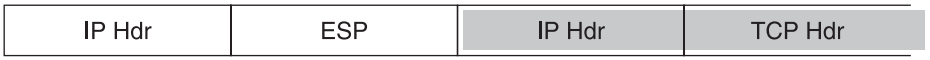
图 6–5 由验证头保护的包



甚至在传输模式下，AH 保护也会涵盖大多数 IP 数据包头。

在隧道模式下，整个数据报处于 IPsec 数据包头的保护之内。图 6–3 中的数据报由外部 IPsec 数据包头（在此示例中为 ESP）以隧道模式保护，如下图所示。

图 6–6 以隧道模式保护的 IPsec 包



加密的

`ipsecconf` 命令包括用于将隧道设置为隧道模式或传输模式的关键字。

- 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，请参见第 95 页中的 “如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响”。
- 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，请参见第 100 页中的 “如何在隧道模式下使用 IPsec 保护 VPN”。

虚拟专用网络和 IPsec

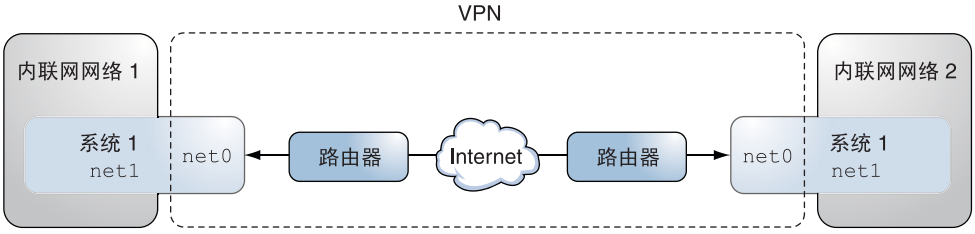
已配置的隧道是点对点接口。使用隧道，可以将一个 IP 包封装到另一个 IP 包中。正确配置的隧道同时要求隧道源和隧道目标。有关更多信息，请参见《配置和管理 Oracle Solaris 11.1 网络》中的“如何创建和配置 IP 隧道”。

隧道可创建明显的 IP physical interface（物理接口）。物理链接的完整性取决于底层安全协议。如果您安全地设置了安全关联 (security association, SA)，则可以信任隧道。退出隧道的包必须源于隧道目标中指定的对等设备。如果此信任存在，则可以使用按接口 IP 转发来创建 virtual private network, VPN（虚拟专用网络）。

您可以对 VPN 添加 IPsec 保护。IPsec 保证连接安全。例如，使用 VPN 技术将办公室与独立网络连接的组织可以添加 IPsec 来保证两个办公室之间的通信安全。

下图说明了两个办公室如何使用其网络系统上部署的 IPsec 来形成 VPN。

图 6–7 虚拟专用网络



有关设置过程的详细示例，请参见第 100 页中的 “如何在隧道模式下使用 IPsec 保护 VPN”。

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 表示网络地址转换。NAT 盒 (NAT box) 用于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。有关更全面的论述，请参见第 42 页中的 “使用 IP 过滤器的 NAT 功能”。

当 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 数据包头，因此 AH 不能用于 NAT-T。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` 检索 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

有关如何通过 NAT 使用 IPsec 的信息，请参见第 145 页中的 “为移动系统配置 IKE（任务列表）”。

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 RFC 和《System Administration Guide: IP Services》中的"SCTP Protocol"。

IPsec 和 Oracle Solaris Zones

对于共享 IP 区域，IPsec 是在全局区域中配置的。IPsec 策略配置文件 `ipsecinit.conf` 仅存在于全局区域中。该文件既可具有应用到非全局区域的项，又可具有应用到全局区域的项。

对于专用 IP 区域，在每个非全局区域中配置 IPsec。

有关如何在区域上使用 IPsec 的信息，请参见第 91 页中的 "使用 IPsec 保护通信"。有关区域的信息，请参见《Oracle Solaris 11.1 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 15 章 "Oracle Solaris Zones 介绍"。

IPsec 和逻辑域

IPsec 与逻辑域一同工作。逻辑域必须运行包含 IPsec 的 Oracle Solaris 版本，如 Oracle Solaris 10 发行版。

要创建逻辑域，必须使用 SPARC 的 Oracle VM Server，之前它被称为 Logical Domains。有关如何配置逻辑域的信息，请参见《Oracle VM Server for SPARC 2.2 管理指南》。

IPsec 实用程序和文件

表 6–3 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 11.1 中管理服务和故障》中的第 1 章 "管理服务（概述）"。

- 有关在网络中实现 IPsec 的说明，请参见第 91 页中的 "使用 IPsec 保护通信"。

- 有关 IPsec 实用程序和文件的更多详细信息，请参见第 8 章，IP 安全体系结构（参考信息）。

表 6–3 所选 IPsec 实用程序和文件的列表

IPsec 实用程序、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`	使用 IKE 自动管理 IPsec SA 的 SMF 服务。	`smf(5)`、`in.iked(1M)`
`/etc/inet/ipsecinit.conf` 文件	IPsec 策略文件。	`ipsecconf(1M)`
	SMF `policy` 服务在系统引导时使用此文件配置 IPsec 策略。	
`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。	`ipsecconf(1M)`
	由 SMF `policy` 服务使用以在系统引导时配置 IPsec 策略。	
`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` 服务使用以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。	`ipsecalgs(1M)`
	由 SMF `ipsecalgs` 服务使用以在系统引导时使已知 IPsec 算法与内核同步。	
`/etc/inet/ipsecalgs` 文件	包含已配置的 IPsec 协议和算法定义。此文件由 `ipsecalgs` 命令管理，并且决不能手动编辑。	
`/etc/inet/ike/config` 文件	IKE 配置和策略文件。缺省情况下，此文件不存在。密钥管理基于 `/etc/inet/ike/config` 文件中的规则和全局参数。请参见第 120 页中的 “IKE 实用程序和文件”。	`ike.config(4)`
	如果此文件存在，`svc:/network/ipsec/ike` 服务会启动 IKE 守护进程 `in.iked` 来提供自动密钥管理。	

◆ ◆ ◆ 第 7 章

7 配置 IPsec（任务）

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：

- 第 91 页中的 “使用 IPsec 保护通信”
- 第 97 页中的 “使用 IPsec 保护 VPN”
- 第 103 页中的 “管理 IPsec 和 IKE”

有关 IPsec 的概述信息，请参见第 6 章，IP 安全体系结构（概述）。有关 IPsec 的参考信息，请参见第 8 章，IP 安全体系结构（参考信息）。

使用 IPsec 保护通信

本节提供保证两个系统之间的通信安全以及保证 Web 服务器的安全的过程。要保护 VPN，请参见第 97 页中的 “使用 IPsec 保护 VPN”。有关管理 IPsec 以及将 SMF 命令与 IPsec 和 IKE 结合使用的其他过程，请参见第 103 页中的 “管理 IPsec 和 IKE”。

以下信息适用于所有的 IPsec 配置任务：

- **IPsec 和区域** – 要管理共享 IP 非全局区域的 IPsec 策略和密钥，请在全局区域中创建 IPsec 策略文件，然后从全局区域运行 IPsec 配置命令。请使用对应于要配置的非全局区域的源地址。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。
- **IPsec 和 RBAC** – 要使用角色来管理 IPsec，请参见《Oracle Solaris 11.1 管理：安全服务》中的第 9 章 “使用基于角色的访问控制（任务）”。有关示例，请参见第 105 页中的 “如何配置网络安全角色”。
- **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，请参见第 88 页中的 “IPsec 和 SCTP”。
- **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。

以下任务列表提供了指向在一个或多个系统之间设置 IPsec 的过程的链接。`ipsecconf(1M)`、`ipseckey(1M)` 和 `ipadm(1M)` 手册页也在各自的“示例”部分中介绍了有用的过程。

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。	第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”
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。	第 95 页中的 “如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响”
显示 IPsec 策略。	按照执行的顺序显示当前正在执行的 IPsec 策略。	第 96 页中的 “如何显示 IPsec 策略”
使用 IKE 为 IPsec SA 自动创建加密材料。	为安全关联提供原始数据。	第 125 页中的 “配置 IKE（任务列表）”
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。	第 97 页中的 “使用 IPsec 保护 VPN”

▼ 如何使用 IPsec 保证两个系统之间的通信安全

假设此过程具有以下设置：

- 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。
- 每个系统都需要采用 AES 算法的 ESP 加密（此算法需要 128 位的密钥）和采用 SHA-2 消息摘要算法的 ESP 验证（此算法需要 512 位的密钥）。
- 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 可以在全局区域或专用 IP 栈区域中配置 IPsec 策略。共享 IP 栈区域的策略必须在全局区域中配置。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

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。要编辑系统文件并创建密钥，您必须承担 root 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 ssh 命令实现安全的远程登录。有关示例，请参见示例 7–1。

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

此步骤使服务管理工具 (Service Management Facility, SMF) 能够使用系统名称，而不需要依赖于不存在的命名服务。有关更多信息，请参见 smf(5) 手册页。

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。

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

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}
```

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}
```

有关 IPsec 策略项的语法，请参见 ipsecconf(1M) 手册页。

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

按照第 125 页中的“配置 IKE（任务列表）”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 ike.config(4) 手册页。

注 – 如果您必须手动生成并维护密钥，请参见第 103 页中的“如何手动创建 IPsec 密钥”。

---

5 **检验 IPsec 策略文件的语法。**

```
# ipsecconf -f -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 svc:/network/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**。如果您已禁用了 IPsec 策略，请将其启用。

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

7 **激活 IPsec 密钥。**

- **如果未启用 ike 服务，请将其启用。**

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

- **如果已启用 ike 服务，请重新启动此服务。**

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

如果在步骤 4 中手动配置了密钥，请完成第 103 页中的 “如何手动创建 IPsec 密钥”以激活密钥。

8 **验证是否对包进行了保护。**

有关过程，请参见第 108 页中的 “如何检验包是否受 IPsec 保护”。

示例 7–1 使用 ssh 连接时添加 IPsc 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。

- 首先，管理员通过执行上述过程的步骤 1 至步骤 5 来配置第一个系统。
- 接着，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:
  other-system #
  ```

- 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。
- 然后，管理员结束 ssh 会话。

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

- 最后，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。

▼ 如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响

安全的 Web 服务器允许 Web 客户机与 Web 服务对话。在安全的 Web 服务器上，不属于 Web 通信的通信**必须**通过安全检查。以下过程会绕过 Web 通信。此外，此 Web 服务器可以发出不安全的 DNS 客户机请求。所有其他通信都需要使用 AES 和 SHA-2 算法的 ESP。

开始之前 必须位于全局区域中才能配置 IPsec 策略。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

您已完成了第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”，因此实际环境符合以下状况：

- 两个系统之间的通信受 IPsec 保护。
- 生成了加密材料。
- 已检验是否对包进行了保护。

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。要编辑系统文件，您必须承担 `root` 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1. **确定哪些服务需要绕过安全策略检查。**

   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 `/etc/inet/ipsecinit.conf` 文件：

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

4 **刷新 IPsec 策略。**

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

5 **刷新 IPsec 的密钥。**

重新启动 `ike` 服务。

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

如果手动配置了密钥，请按照第 103 页中的 “如何手动创建 IPsec 密钥”中的说明操作。

您的设置已完成。（可选）您可以执行步骤 6。

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
#
{laddr 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 -f -c /etc/inet/ipsecinit.conf
remote-system # svcadm refresh svc:/network/ipsec/policy:default
```

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。

▼ 如何显示 IPsec 策略

当您发出不带任何参数的 `ipsecconf` 命令时，便可以查看在系统中配置的策略。

开始之前 必须在全局区域中运行 `ipsecconf` 命令。对于专用 IP 区域，请在非全局区域中运行 `ipsecconf` 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。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- **显示 IPsec 策略。**
  - **按照全局 IPsec 策略项的添加顺序显示这些项。**

    ```
    $ ipsecconf
    ```

    此命令将每项显示为后面跟有一个数字的**索引**。
  - **按照匹配项出现的顺序显示 IPsec 策略项。**

    ```
    $ ipsecconf -l -n
    ```

- **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
```

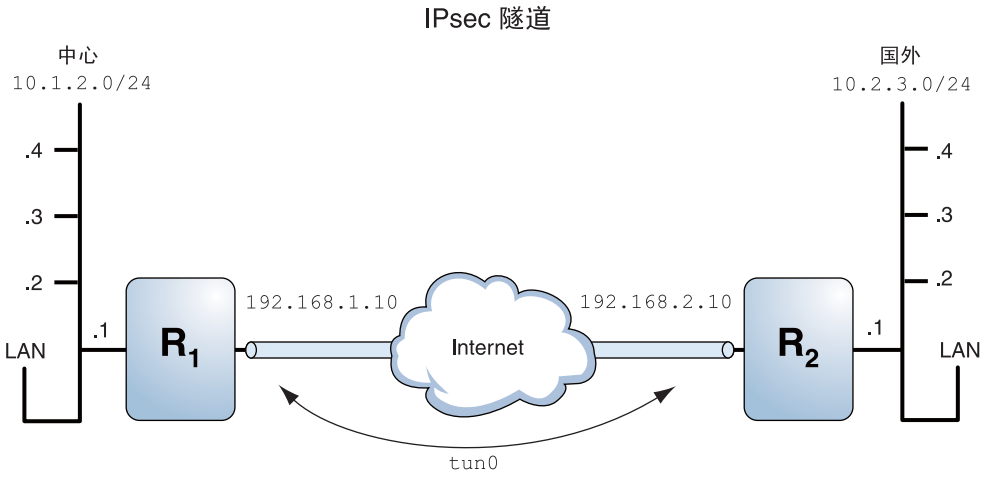
使用 IPsec 保护 VPN

Oracle Solaris 可以配置受 IPsec 保护的 VPN。可在**隧道模式**或**传输模式**下创建隧道。有关介绍，请参见第 84 页中的 “IPsec 中的传输模式和隧道模式”。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，请参见第 91 页中的 “使用 IPsec 保护通信”。

有关处于隧道模式的隧道的 IPsec 策略的示例，请参见第 97 页中的 “在隧道模式下使用 IPsec 保护 VPN 的示例”。

在隧道模式下使用 IPsec 保护 VPN 的示例

图 7–1　受 IPsec 保护的隧道



以下示例假设这些 LAN 的所有子网都配置了隧道：

```
## Tunnel configuration ##
# Tunnel name is tun0
# Intranet point for the source is 10.1.2.1
# Intranet point for the destination is 10.2.3.1
# Tunnel source is 192.168.1.10
# Tunnel destination is 192.168.2.10

# Tunnel name address object is tun0/to-central
# Tunnel name address object is tun0/to-overseas
```

示例 7–2 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 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}
```

示例 7–3 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。

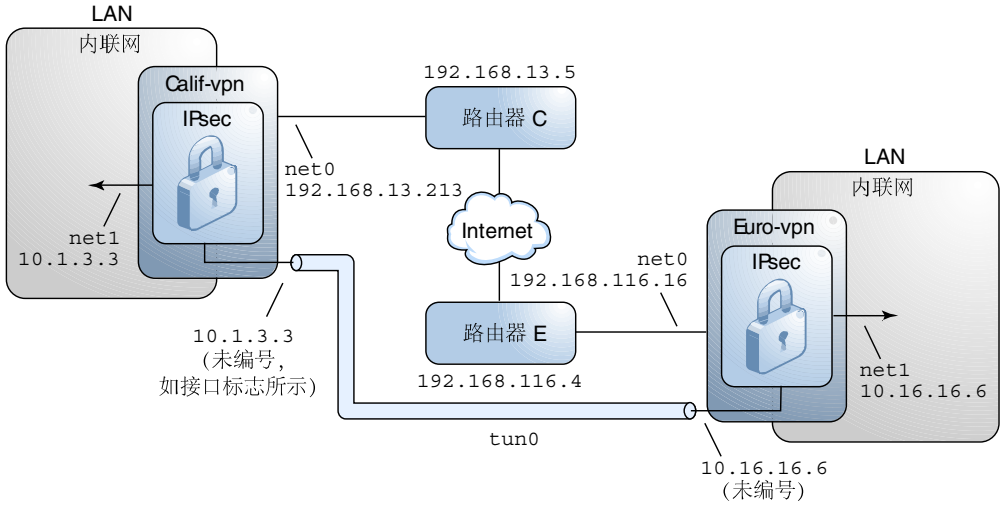
```
## IPsec policy ##
{tunnel tun0 negotiate tunnel laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs aes encr_auth_algs sha512 shared}
```

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。

- 每个系统都使用 IPv4 地址空间。
- 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 SHA-2 算法的 ESP 验证。在此示例中，SHA-2 算法需要 512 位的密钥。
- 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。

图 7–2 通过 Internet 连接的办公室之间的 VPN 样例



如上图所示，这些过程使用以下配置参数。

参数	欧洲	加利福尼亚
系统名称	euro-vpn	calif-vpn
系统内联网接口	net1	net1
系统内联网地址，也是步骤 6 中的 *-point* 地址	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，也是步骤 6 中的 *tsrc* 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《配置和管理 Oracle Solaris 11.1 网络》中的“使用 dladm 命令进行隧道配置和管理”。有关地址对象的信息，请参见《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”和 ipadm(1M) 手册页。

▼ 如何在隧道模式下使用 IPsec 保护 VPN

在隧道模式下，内部 IP 包决定保护其内容的 IPsec 策略。

此过程扩展了第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”过程。第 98 页中的 “用于保护 VPN 的 IPsec 任务的网络拓扑说明”介绍了具体设置。

有关运行特定命令的更详尽的原因说明，请参见第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”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 必须位于全局区域中才能为系统或共享 IP 区域配置 IPsec 策略。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

要运行配置命令，您必须成为分配有 "Network Management"（网络管理）和 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。要编辑系统文件，您必须承担 `root` 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

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
```

关闭 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. 打开 IP 严格多宿主。

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

打开 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。

检验回送挂载和 ssh 服务是否正在运行。

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

2 添加 IPsec 策略。

编辑 /etc/inet/ipsecinit.conf 文件来为 VPN 添加 IPsec 策略。有关其他示例，请参见第 97 页中的 “在隧道模式下使用 IPsec 保护 VPN 的示例”。

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，将以下项键入到 ipsecinit.conf 文件中：

```
# 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，将以下项键入到 ipsecinit.conf 文件中：

```
# 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。

按照第 125 页中的 “配置 IKE（任务列表）”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 ike.config(4) 手册页。

注 – 如果您必须手动生成并维护密钥，请参见第 103 页中的 “如何手动创建 IPsec 密钥”。

---

4 验证 IPsec 策略文件的语法。

```
# ipsecconf -f -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 svc:/network/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**。如果您已禁用了 IPsec 策略，请将其启用。

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

6 **创建并配置隧道** *tunnel-name*。

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。**

如果接口 `net1` 不存在，第一个命令将会创建该接口。

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

b. **在 `euro-vpn` 系统上，创建该隧道并进行配置。**

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

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 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。有关定制名称的信息，请参见《Oracle Solaris 管理：网络接口和网络虚拟化》中的“网络设备和数据链路名称”。

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=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，必须打开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**，所以必须打开 `net1` 的 IP 转发功能。由于 `tun0` 通过 Internet 连接两个系统，因此 IP 转发必须对 `tun0` 保持打开状态。`net0` 接口已关闭其 IP 转发功能以阻止**外部**入侵者向受保护的内联网中注入包。**外部**是指 Internet。

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` 关闭 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。

管理 IPsec 和 IKE

以下任务列表列出了在管理 IPsec 时可能要执行的任务。

任务	说明	参考
手动创建或替换安全关联。	为安全关联提供原始数据： ■ IPsec 算法名称和加密材料 ■ 安全参数索引 (security parameter index, SPI) ■ IP 源地址和目标地址及其他参数	第 103 页中的 “如何手动创建 IPsec 密钥”
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。	第 105 页中的 “如何配置网络安全角色”
将 IPsec 和加密材料作为一组 SMF 服务来管理。	介绍何时以及如何使用相应命令来启用、禁用、刷新和重新启动服务。此外，还介绍了用于更改服务的属性值的命令。	第 107 页中的 “如何管理 IPsec 和 IKE 服务”
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 数据报的特定头。	第 108 页中的 “如何检验包是否受 IPsec 保护”

▼ 如何手动创建 IPsec 密钥

以下过程为第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”中的步骤 4 提供了加密材料。您要为两个系统（partym 和 enigma）生成密钥。您在一个系统上生成密钥，然后在两个系统中使用在第一个系统中生成的密钥。

开始之前 您必须位于全局区域中，才能手动管理非全局区域的加密材料。

您必须承担 root 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1 为 SA 生成加密材料。

a. 确定所需的密钥。

您需要将三个十六进制随机数用于外发通信，三个十六进制随机数用于传入通信。因此，一个系统需要生成以下数字：

- 两个作为 spi 关键字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 AH 的 SHA-2 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 ESP 的 3DES 算法的十六进制随机数。每个数字的长度都必须为 168 个字符。一个数字用于 dst enigma，另一个数字用于 dst partym。

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《Oracle Solaris 11.1 管理：安全服务》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 在每个系统的 root 角色中，将密钥添加到 IPsec 的手动密钥文件中。

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

```
# 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 3des \
   auth_alg sha512  \
   encrkey  d41fb74470271826a8e7a80d343cc5aa... \
   authkey  e896f8df7f78d6cab36c94ccf293f031...
#
# for inbound packets
add esp spi 0x122a43e4 \
   src 192.168.13.213 dst 192.168.116.16 \
   encr_alg 3des \
   auth_alg sha512  \
   encrkey dd325c5c137fb4739a55c9b3a1747baa... \
   authkey ad9ced7ad5f255c9a8605fba5eb4d2fd...
```

b. **使用只读权限保护该文件。**

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

c. **验证文件的语法。**

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

注 – 两个系统上的加密材料**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。**

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

- **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。

▼ 如何配置网络安全角色

如果您要使用 Oracle Solaris 的基于角色的访问控制 (role-based access control, RBAC) 功能来管理系统，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
Console User:RO::Manage System as the Console User...
Network Management:RO::Manage the host and network configuration...
Network Autoconf Admin:RO::Manage Network Auto-Magic configuration via nwamd...
Network Autoconf User:RO::Network Auto-Magic User...
Network ILB:RO::Manage ILB configuration via ilbadm...
Network LLDP:RO::Manage LLDP agents via lldpadm...
Network VRRP:RO::Manage VRRP instances...
Network Observability:RO::Allow access to observability devices...
Network Security:RO::Manage network and host security...:profiles=Network Wifi
Security,Network Link Security,Network IPsec Management...
Network Wifi Management:RO::Manage wifi network configuration...
Network Wifi Security:RO::Manage wifi network security...
Network Link Security:RO::Manage network link security...
Network IPsec Management:RO::Manage IPsec and IKE...
System Administrator:RO::Can perform most non-security administrative tasks:
profiles=...Network Management...
Information Security:RO::Maintains MAC and DAC security policies:
profiles=...Network Security...
```

Network Management（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。**

```
% getent exec_attr | grep "Network Management"
...
Network Management:solaris:cmd:::/sbin/dlstat:euid=dladm;egid=sys
...
Network Management:solaris:cmd:::/usr/sbin/snoop:privs=net_observability
Network Management:solaris:cmd:::/usr/sbin/spray:euid=0 ...
```

3 **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。

4 **创建拥有 Network Management（网络管理）权限配置文件的网络安全角色。**

对于拥有除 Network Management（网络管理）配置文件外还拥有 Network Security（网络安全）或 Network IPsec Management（网络 IPsec 管理）权限配置文件的角色，除了可执行其他命令外，还可按相应的特权执行 `ipadm`、`ipseckey` 和 `snoop` 命令。

要创建该角色、将该角色指定给用户以及使用命名服务注册更改，请参见《Oracle Solaris 11.1 管理：安全服务》中的“初次配置 RBAC（任务列表）”。

示例 7–4 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：

- 管理员将第一个角色命名为 LinkWifi。
  - 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。
  - 然后，管理员将 LinkWifi 角色指定给适当的用户。
- 管理员将第二个角色命名为 IPsec Administrator。
  - 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。
  - 然后，管理员将 IPsec Administrator（IPsec 管理员）角色指定给适当的用户。

▼ 如何管理 IPsec 和 IKE 服务

以下步骤提供了最有可能针对 IPsec、IKE 和手动密钥管理使用 SMF 服务的情况。缺省情况下，`policy` 和 `ipsecalgs` 服务处于启用状态，而 `ike` 和 `manual-key` 服务处于禁用状态。

开始之前 您必须承担 `root` 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1. **要管理 IPsec 策略，请执行以下操作之一：**

   - **将新策略添加到 `ipsecinit.conf` 文件后，刷新 `policy` 服务。**

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

   - **更改服务属性的值后，查看属性值，然后刷新并重新启动 `policy` 服务。**

     ```
     # svccfg -s policy setprop config/config_file=/etc/inet/MyIpsecinit.conf
     # svccfg -s policy listprop config/config_file
     config/config_file  astring  /etc/inet/MyIpsecinit.conf
     # svcadm refresh svc:/network/ipsec/policy
     # svcadm restart svc:/network/ipsec/policy
     ```

2. **要自动管理密钥，请执行以下操作之一：**

   - **将项添加到 `/etc/inet/ike/config` 文件后，启用 `ike` 服务。**

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

   - **更改 `/etc/inet/ike/config` 文件中的项后，重新启动 `ike` 服务。**

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

   - **更改服务属性的值后，查看属性值，然后刷新并重新启动服务。**

     ```
     # svccfg -s ike setprop config/admin_privilege = astring: "modkeys"
     # svccfg -s ike listprop config/admin_privilege
     config/admin_privilege  astring  modkeys
     # svcadm refresh svc:/network/ipsec/ike
     # svcadm restart svc:/network/ipsec/ike
     ```

   - **要停止 `ike` 服务，请将其禁用。**

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

3. **要手动管理密钥，请执行以下操作之一：**

   - **将项添加到 `/etc/inet/secret/ipseckeys` 文件后，启用 `manual-key` 服务。**

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

   - **更改 `ipseckeys` 文件后，刷新服务。**

     ```
     # svcadm refresh manual-key
     ```

- **更改服务属性的值后，查看属性值，然后刷新并重新启动服务。**

```
# svccfg -s manual-key setprop config/config_file=/etc/inet/secret/MyIpseckeyfile
# svccfg -s manual-key listprop config/config_file
config/config_file  astring  /etc/inet/secret/MyIpseckeyfile
# svcadm refresh svc:/network/ipsec/manual-key
# svcadm restart svc:/network/ipsec/manual-key
```

- **要阻止手动密钥管理，请禁用 manual-key 服务。**

```
# svcadm disable svc:/network/ipsec/manual-key
```

4. **如果修改 IPsec 协议和算法表，请刷新 ipsecalgs 服务。**

```
# svcadm refresh svc:/network/ipsec/ipsecalgs
```

故障排除 使用 `svcs` *service* 命令找到服务的状态。如果服务处于 `maintenance` 模式，请遵循 `svcs -x` *service* 命令输出的调试建议。

▼ 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 `AH:`。
- `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 `ESP:`。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1. **在一个系统（如 partym）上，承担 root 角色。**

```
% su -
Password:       Type root password
#
```

2. **在 partym 系统上，准备从远程系统搜寻包。**

在 `partym` 上的一个终端窗口中，从 `enigma` 系统搜寻包。

```
# snoop -d net0 -v enigma
Using device /dev/bge (promiscuous mode)
```

3 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。

```
% ssh enigma
Password:       Type your password
% su -
Password:       Type root password
# ping partym
```

4 检查 snoop 输出。

在 partym 系统上，您应该看到在初始 IP 头信息之后显示 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

IP 安全体系结构（参考信息）

本章包含以下参考信息：

有关如何在网络中实现 IPsec 的说明，请参见第 7 章，配置 IPsec（任务）。有关 IPsec 的概述，请参见第 6 章，IP 安全体系结构（概述）。

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。初始值为 /etc/inet/ipsecinit.conf。
- 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，请参见第 155 页中的 “IKE 服务”。

有关 SMF 的信息，请参见《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 章 “管理服务（概述）”。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 数据报的策略。转发的数据报不受使用此命令添加的策略检查的约束。ipsecconf 命令也可配置安全策略数据库 (security policy database, SPD)。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。此命令接受保护双向通信的项，同时也接受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。例如，请参见第 91 页中的 “使用 IPsec 保护通信”。

ipsecinit.conf 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 does 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。

保护您的名称系统。如果发生以下两种情况，则您的主机名不再值得信任：

- 您的源地址是可以在网络中查找到的主机。
- 您的名称系统受到威胁。

安全漏洞通常是由工具使用不当造成的，而并非由工具本身引起。应慎用 ipsecconf 命令。请将 ssh、控制台或其他硬连接的 TTY 用作最安全的操作模式。

ipsecalgs 命令

加密框架为 IPsec 提供验证和加密算法。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，不需要修改此文件。但是，如果需要修改此文件，请使用 ipsecalgs 命令。决不能直接编辑此文件。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。通常，DOI（解释域）定义数据格式、网络通信交换类型，以及命名安全相关信息的约定。安全策略、加密算法和加密模式都属于安全相关信息。

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《Oracle Solaris 11.1 管理：安全服务》中的第 11 章 “加密框架（概述）”。

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。SADB 可能由某个用户进程维护，也可能由多个协作进程（以特定类的套接字发送消息）维护。这种维护 SADB 的方法类似于 route(7P) 手册页中介绍的方法。只有 root 角色才能访问该数据库。

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 中用于生成 SA 的实用程序

IKE（Internet 密钥交换）协议提供自动执行的 IPv4 和 IPv6 地址密钥管理。有关如何设置 IKE 的说明，请参见第 10 章，配置 IKE（任务）。手动加密实用程序是 ipseckey 命令，在 ipseckey(1M) 手册页中对其进行了介绍。

可使用 ipseckey 命令手动填充安全关联数据库 (security associations database, SADB)。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。

ipseckey 命令可用于查看系统识别的所有 SA，无论密钥是手动添加的还是由 IKE 添加的。借助 -c 选项，ipseckey 命令可检查作为参数提供的密钥文件的语法。

由 ipseckey 命令添加的 IPsec SA 在系统重新引导后不会保留。要在系统引导时启用手动添加的 SA，请将相应的项添加到 /etc/inet/secret/ipseckeys 文件，然后启用 svc:/network/ipsec/manual-key:default 服务。有关过程，请参见第 103 页中的 “如何手动创建 IPsec 密钥”。

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 pf_key(7P) 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 – 如果可能，使用 IKE 而不是通过 `ipseckey` 进行密钥管理。

当处理加密材料和使用 `ipseckey` 命令时，您应该考虑以下问题：

- 是否已更新加密材料？定期更新密钥是一项基本的安全措施。密钥更新可以防止遭到潜在的算法和密钥漏洞攻击，并且限制对已公开的密钥的破坏。
- TTY 是否联网？`ipseckey` 命令是否处于交互模式？
  - 在交互模式下，加密材料的安全即为此 TTY 通信的网络路径的安全。应该避免在明文 telnet 或 rlogin 会话中使用 `ipseckey` 命令。
  - 甚至本地窗口也可能受到读取窗口事件的隐藏程序的攻击。
- 是否已使用 `-f` 选项？是否通过网络访问文件？此文件是否完全公开？
  - 入侵者可能会在系统读取网络挂载的文件时也读取此文件。您应该避免使用包含加密材料的完全公开文件。
  - 保护您的名称系统。如果发生以下两种情况，则您的主机名不再值得信任：
    - 您的源地址是可以在网络中查找到的主机。
    - 您的名称系统受到威胁。

安全漏洞通常是由工具使用不当造成的，而并非由工具本身引起。应慎用 `ipseckey` 命令。请将 `ssh`、控制台或其他硬连接的 TTY 用作最安全的操作模式。

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 不会对数据进行加密。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。有关更多详细信息，请参见 `snoop(1M)` 手册页。

有关受保护包中的详细 `snoop` 输出样例，请参见第 108 页中的 “如何检验包是否受 IPsec 保护”。

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。

◆ ◆ ◆ 第 9 章

Internet 密钥交换（概述）

Internet 密钥交换 (Internet Key Exchange, IKE) 自动进行 IPsec 的密钥管理。Oracle Solaris 实现 IKEv1。本章包含有关 IKE 的以下信息：

有关实现 IKE 的说明，请参见第 10 章，配置 IKE（任务）。有关参考信息，请参见第 11 章，Internet 密钥交换（参考信息）。有关 IPsec 的概述信息，请参见第 6 章，IP 安全体系结构（概述）。

使用 IKE 进行密钥管理

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**。自动密钥管理需要用于创建、验证和交换密钥的安全信道。Oracle Solaris 使用 Internet 密钥交换 (Internet Key Exchange, IKE) 版本 1 自动进行密钥管理。IKE 可轻松扩展以便为大量通信提供安全信道。IPv4 和 IPv6 包中的 IPsec SA 可以利用 IKE。

IKE 可以利用可用的硬件加速和硬件存储。通过硬件加速器，可以将密集的密钥操作转移到系统外处理。硬件上的密钥存储提供了额外的一层保护。

IKE 密钥协商

IKE 守护进程 `in.iked` 以安全方式为 IPsec SA 协商和验证加密材料。该守护进程使用来自 OS 提供的内部函数的随机密钥种子。IKE 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。

IKE 密钥术语

下表列出在密钥协商中使用的术语，提供其常用的首字母缩略词，并给出每个术语的定义和用法。

表 9–1 密钥协商术语及其首字母缩略词和用法

密钥协商术语	首字母缩略词	定义和用法
Key exchange（密钥交换）		生成非对称加密算法的密钥的过程。两种主要方法是 RSA 和 Diffie-Hellman 协议。
Diffie-Hellman algorithm（Diffie-Hellman 算法）	DH	提供密钥生成和密钥验证的密钥交换算法。通常称为**经过验证的密钥交换**。
RSA algorithm（RSA 算法）	RSA	提供密钥生成和密钥传输的密钥交换算法。此协议以其三个创建者 Rivest、Shamir 和 Adleman 命名。
Perfect forward secrecy（完全正向保密）	PFS	仅适用于经过验证的密钥交换。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。
Oakley group（Oakley 组）		一种以安全方式为阶段 2 建立密钥的方法。Oakley 组用于协商 PFS。请参见 The Internet Key Exchange (IKE) (`http://www.faqs.org/rfcs/rfc2409.html`)（Internet 密钥交换 (Internet Key Exchange, IKE)）的第 6 节。

IKE 阶段 1 交换

阶段 1 交换称为**主模式**。在阶段 1 交换中，IKE 使用公钥加密方法向对等 IKE 实体进行自我验证。结果是 Internet 安全关联和密钥管理协议 (Internet Security Association and Key Management Protocol, ISAKMP) 安全关联 (security association, SA)。ISAKMP SA 是 IKE 用于协商 IP 数据报的加密材料的安全信道。与 IPsec SA 不同，ISAKMP SA 是双向的，因此只需要一个安全关联。

IKE 在阶段 1 交换中协商加密材料的方式是可配置的。IKE 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
- 是否使用完全正向保密 (perfect forward secrecy, PFS)
- 受影响的接口
- 安全协议及其算法
- 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，证书可以由来自公钥基础结构 (public key infrastructure, certificate authority, CA（证书颁发机构）) 组织的 PKI颁发。

IKE 阶段 2 交换

阶段 2 交换称为**快速模式**。在阶段 2 交换中，IKE 在运行 IKE 守护进程的系统之间创建和管理 IPsec SA。IKE 使用在阶段 1 交换中创建的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刷新密钥。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。

IKE 配置选择

`/etc/inet/ike/config` 配置文件包含 IKE 策略项。为了使两个 IKE 守护进程相互验证，这些项必须是有效的。此外，加密材料必须可用。配置文件中的项确定使用加密材料验证阶段 1 交换的方法。可以选择预先共享的密钥或公钥证书。

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。公钥证书可以自签名，也可以从 PKI 组织安装。有关更多信息，请参见 `ike.config(4)` 手册页。

使用预先共享的密钥验证的 IKE

预先共享的密钥用于验证两个或多个对等方系统。预先共享的密钥为由一个系统上的管理员创建的十六位数字或 ASCII 字符串。然后，以安全方式与对等方系统的管理员在带外共享该密钥。如果预先共享的密钥被入侵者拦截，则该入侵者可以假扮某个对等方系统。

使用此验证方法的对等方中预先共享的密钥必须相同。这些密钥与特定的 IP 地址或地址范围相关联。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.preshared(4)` 手册页。

IKE，使用公钥证书

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。公钥使用 Diffie-Hellman algorithm（Diffie-Hellman 算法）(DH) 来验证和协商密钥。公钥证书有两种类型。这些证书可以自签名，也可以由 certificate authority, CA（证书颁发机构）认证。

自签名的公钥证书由管理员创建。`ikecert certlocal -ks` 命令为系统创建公钥/私钥对的私钥部分。然后，从远程系统获取 X.509 格式的自签名证书输出。远程系统的证书是用于创建密钥对的公钥部分的 `ikecert certdb` 命令的输入。在通信系统上，自签名的证书驻留在 `/etc/inet/ike/publickeys` 目录中。使用 `-T` 选项时，证书驻留在连接的硬件上。

自签名的证书介于预先共享的密钥和 CA 中间。与预先共享的密钥不同，自签名的证书可以在移动计算机或可能重新编号的系统上使用。要为没有固定编号的系统对证书自行签名，请使用 `DNS (www.example.org)` 或 `email (root@domain.org)` 替换名称。

公钥可以由 PKI 或 CA 组织提供。在 `/etc/inet/ike/publickeys` 目录中安装公钥及其相应的 CA。使用 `-T` 选项时，证书驻留在连接的硬件上。供应商还发布证书撤销列表 (certificate revocation list, CRL)。除安装密钥和 CA 以外，您还负责在 `/etc/inet/ike/crls` 目录中安装 CRL。

CA 的优势在于由外部组织而不是由站点管理员认证。在某种意义上，CA 是经过确认的证书。与自签名的证书一样，CA 可以在移动计算机或可能重新编号的系统上使用。与自签名的证书不同的是，CA 可以非常容易地扩展以保护大量的通信系统。

IKE 实用程序和文件

下表概述了 IKE 策略的配置文件、IKE 密钥的存储位置以及实现 IKE 的各种命令和服务。有关服务的更多信息，请参见《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 章 “管理服务（概述）”。

表 9–2 IKE 配置文件、密钥存储位置、命令和服务

文件、位置、命令或服务	说明	手册页
`svc:/network/ipsec/ike`	管理 IKE 的 SMF 服务。	`smf(5)`
`/usr/lib/inet/in.iked`	Internet 密钥交换 (Internet Key Exchange, IKE) 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`	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`	IKE 策略的缺省配置文件。包含用于匹配传入 IKE 请求和准备外发 IKE 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)`

表 9–2 IKE 配置文件、密钥存储位置、命令和服务 （续）

文件、位置、命令或服务	说明	手册页
ike.preshared	/etc/inet/secret 目录中的预先共享密钥文件。包含用于阶段 1 交换中验证的秘密加密材料。在用预先共享的密钥配置 IKE 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	通过从操作系统转移操作来加速公钥操作的硬件。该板还存储公钥、私钥和公钥证书。Sun Crypto Accelerator 6000 板是第 3 级别的 FIPS 140-2 认证设备。	ikecert(1M)

◆ ◆ ◆ 第 10 章

10 配置 IKE（任务）

本章介绍如何为系统配置 Internet 密钥交换 (Internet Key Exchange, IKE)。配置 IKE 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：

有关 IKE 的概述信息，请参见第 9 章，Internet 密钥交换（概述）。有关 IKE 的参考信息，请参见第 11 章，Internet 密钥交换（参考信息）。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。

显示 IKE 信息

您可以查看阶段 1 IKE 协商中使用的算法和组。

▼ 如何显示阶段 1 IKE 交换的可用组和算法

在此过程中，将确定哪些 Diffie-Hellman 组可用于阶段 1 IKE 交换。此外，还将查看可用于 IKE 阶段 1 交换的加密和验证算法。这些数值与 Internet 号码分配机构 (Internet Assigned Numbers Authority, IANA) 为这些算法指定的值匹配。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1. **显示 IKE 可在阶段 1 中使用的 Diffie-Hellman 组的列表。**

   Diffie-Hellman 组设置 IKE SA。

```
# ikeadm dump groups
Value Strength Description
1     66       ietf-ike-grp-modp-768
2     77       ietf-ike-grp-modp-1024
5     91       ietf-ike-grp-modp-1536
14    110      ietf-ike-grp-modp-2048
15    130      ietf-ike-grp-modp-3072
16    150      ietf-ike-grp-modp-4096
17    170      ietf-ike-grp-modp-6144
18    190      ietf-ike-grp-modp-8192

Completed dump of groups
```

   将在 IKE 阶段 1 转换中使用这些值之一作为 `oakley_group` 参数的变量，如以下内容中所示：

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

2. **显示 IKE 可在阶段 1 中使用的验证算法的列表。**

```
# ikeadm dump authalgs
Value Name
1     md5
2     sha1
4     sha256
5     sha384
6     sha512

Completed dump of authalgs
```

   将在 IKE 阶段 1 转换中使用这些名称之一作为 `auth_alg` 参数的变量，如以下内容中所示：

```
p1_xform
  { auth_method preshared oakley_group 15 auth_alg sha256 encr_alg 3des }
```

3. **显示 IKE 可在阶段 1 中使用的加密算法的列表。**

```
# ikeadm dump encralgs
Value Name
3     blowfish-cbc
5     3des-cbc
1     des-cbc
7     aes-cbc

Completed dump of encralgs
```

   将在 IKE 阶段 1 转换中使用这些名称之一作为 `encr_alg` 参数的变量，如以下内容中所示：

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

另请参见 有关配置需要这些值的 IKE 规则的任务，请参见第 125 页中的“配置 IKE（任务列表）”。

配置 IKE（任务列表）

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) 所颁发的证书来验证 IKE。规则将特定的 IKE 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 IKE 验证方法。利用指向 PKCS #11 库的指针，IKE 可以使用连接的硬件加速器。

配置 IKE 后，完成使用 IKE 配置的 IPsec 任务。下表提供了着重说明特定 IKE 配置的任务列表。

任务	说明	参考
使用预先共享的密钥配置 IKE。	通过使两个系统共享一个密钥来保护它们之间的通信。	第 125 页中的“使用预先共享的密钥配置 IKE（任务列表）”
使用公钥证书配置 IKE。	使用公钥证书保护通信。这些证书可以是自签名的，也可以由 PKI 组织认证。	第 130 页中的“使用公钥证书配置 IKE（任务列表）”
跨 NAT 边界。	将 IPsec 和 IKE 配置为与移动系统进行通信	第 145 页中的“为移动系统配置 IKE（任务列表）”
配置 IKE 以使用硬件密钥库生成证书对。	启用 Sun Crypto Accelerator 6000 板以加速 IKE 操作和存储公钥证书。	第 152 页中的“将 IKE 配置为查找连接的硬件”

使用预先共享的密钥配置 IKE（任务列表）

下表包含使用预先共享的密钥配置和维护 IKE 的过程的链接。

任务	说明	参考
使用预先共享的密钥配置 IKE。	创建 IKE 配置文件和要共享的一个密钥。	第 126 页中的“如何使用预先共享的密钥配置 IKE”
将预先共享的密钥添加到正运行的 IKE 系统。	将新的 IKE 策略项和新的加密材料添加到当前实施 IKE 策略的系统。	第 128 页中的“如何为新的对等方系统更新 IKE”

使用预先共享的密钥配置 IKE

使用预先共享的密钥是验证 IKE 的最简单方法。如果要将对等方系统配置为使用 IKE，而且您是这两个系统的管理员，则使用预先共享的密钥是一个良好的选择。但是，与公钥证书不同，预先共享的密钥与 IP 地址相关联。可以将预先共享的密钥与特定的 IP 地址或 IP 地址范围关联。预先共享的密钥不能用于移动系统或可能重新编号的系统，除非重新编号处于指定的 IP 地址范围内。

▼ 如何使用预先共享的密钥配置 IKE

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 除了 `solaris.admin.edit/etc/inet/ike/config` 授权以外，您还必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。`root` 角色具有所有这些权限。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

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。以下是与第 92 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”中的 `ipsecinit.conf` 示例配合使用的 IKE 配置示例。

   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 在每个系统上创建文件 **/etc/inet/secret/ike.preshared**。

在每个文件中放置预先共享的密钥。

a. 例如，在 **enigma** 系统上，**ike.preshared** 文件的显示与以下信息类似：

```
# 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 文件的显示与以下信息类似：**

```
# 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 IKE 服务。**

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

示例 10–1 刷新 IKE 预先共享的密钥

如果 IKE 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，管理员添加一个预先共享的密钥项，该项对 `192.168.13.0/24` 子网中的任意主机都有效。

```
#...
{ localidtype IP
    localid 192.168.116.0/24
    remoteidtype IP
    remoteid 192.168.13.0/24
    # enigma and partym's shared passphrase for keying material
key "LOooong key Th@t m^st Be Ch*angEd \'reguLarLy)"
    }
```

然后，管理员重新启动每个系统上的 IKE 服务。

```
# 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。

▼ 如何为新的对等方系统更新 IKE

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 IKE 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。

除了 solaris.admin.edit/etc/inet/ike/config 授权以外，您还必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。root 角色具有所有这些权限。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 ssh 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1 **为 IKE 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 IKE 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
```

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。

使用公钥证书配置 IKE（任务列表）

下表提供了为 IKE 创建公钥证书的过程的链接。这些过程包括如何在连接的硬件上加速和存储证书。

公共证书必须是唯一的，因此公钥证书的创建者应为该证书生成任意的唯一名称。通常，将使用 X.509 标识名。此外，也可使用替代名称进行标识。这些名称的格式为 *tag=value*。这些值是任意的，但值的格式必须与其标记类型对应。例如，`email` 标记的格式为 *name@domain.suffix*。

任务	说明	参考
使用自签名的公钥证书配置 IKE。	在每个系统上创建并放置两个证书： ■ 自签名证书 ■ 来自对等方系统的公钥证书	第 131 页中的“如何使用自签名的公钥证书配置 IKE”
通过 PKI 证书颁发机构配置 IKE。	创建证书请求，然后在每个系统上放置三个证书： ■ 证书颁发机构 (Certificate Authority, CA) 根据您的请求创建的证书 ■ 来自 CA 的公钥证书 ■ 来自 CA 的 CRL	第 136 页中的“如何使用 CA 签名的证书配置 IKE”
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请求，然后将公钥证书从 CA 添加到硬件。	第 140 页中的“如何在硬件中生成和存储公钥证书”
更新来自 PKI 的证书撤销列表 (certificate revocation list, CRL)。	从中心分发点访问 CRL。	第 143 页中的“如何处理证书撤销列表”

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 IPsec（任务列表）”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。

使用公钥证书配置 IKE

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。与预先共享的密钥不同，公钥证书可以在移动计算机或可能重新编号的系统上使用。

公钥证书也可以生成和存储于所连接的硬件中。有关过程，请参见第 152 页中的 “将 IKE 配置为查找连接的硬件”。

▼ 如何使用自签名的公钥证书配置 IKE

在此过程中，将创建证书对。私钥存储于本地证书数据库中的磁盘上，可以使用 `certlocal` 子命令进行引用。证书对的公共部分存储于公共证书数据库中，可以使用 `certdb` 子命令进行引用。您将与对等方系统交换该公共部分。两个证书的组合用于验证 IKE 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。与 CA 颁发的证书不同，自签名证书必须在带外进行验证。

开始之前 除了 `solaris.admin.edit/etc/inet/ike/config` 授权以外，您还必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。`root` 角色具有所有这些权限。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1 在 `ike.privatekeys` 数据库中创建自签名证书。

```
# ikecert certlocal -ks -m keysize -t keytype \
-D dname -A altname \
[-S validity-start-time] [-F validity-end-time] [-T token-ID]
```

`-ks`	创建自签名证书。
`-m` *keysize*	是密钥的大小。*keysize* 可以是 512、1024、2048、3072 或 4096。
`-t` *keytype*	指定要使用的算法类型。*keytype* 可以是 `rsa-sha1`、`rsa-md5` 或 `dsa-sha1`。

-D *dname*	是证书主题的 X.509 标识名。*dname* 通常具有以下格式：C=*country*，O=*organization*，OU=*organizational unit*，CN=*common name*。有效标记是 C、O、OU 和 CN。
-A *altname*	是证书的替代名称。*altname* 的形式为 tag=value。有效标记是 IP、DNS、email 和 DN。
-S *validity-start-time*	为证书提供绝对或相对有效开始时间。
-F *validity-end-time*	为证书提供绝对或相对有效结束时间。
-T *token-ID*	启用 PKCS #11 硬件标记来生成密钥。然后证书将被存储在硬件中。

a. 例如，partym 系统上命令的显示与以下信息类似：

```
# ikecert certlocal -ks -m 2048 -t rsa-sha1 \
-D "O=exampleco, OU=IT, C=US, CN=partym" \
-A IP=192.168.13.213
Creating private key.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEfdZgKjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
a...+
zBGi4QkNdI3f
-----END X509 CERTIFICATE-----
```

注 – -D 和 -A 选项的值是任意的。这些值仅用于标识证书。它们不用于标识系统，例如 192.168.13.213。事实上，由于这些值具有特异性，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信息类似：

```
# ikecert certlocal -ks -m 2048 -t rsa-sha1 \
-D "O=exampleco, OU=IT, C=US, CN=enigma" \
-A IP=192.168.116.16
Creating private key.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIEBl5JnjANBgkqhkiG9w0BAQUFADAaMRgwFgYDVQQDEw9T
...
y85m6LHJYtC6
-----END X509 CERTIFICATE-----
```

2 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@ja.enigmaexample.com
From: admin@us.partyexample.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@us.partyexample.com
From: admin@ja.enigmaexample.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 向其他管理员核实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 中证书的标识名 (distinguished name, DN)。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，必须具有匹对的证书对。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-sha1
        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，同时信任这两个证书。

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 文件的显示与以下信息类似：

```
# 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-enigmax"
 local_id_type dn
 local_id "O=exampleco, OU=IT, C=US, CN=partym"
 remote_id "O=exampleco, OU=IT, C=US, CN=enigma"

 local_addr  192.168.13.213
# We could explicitly enter the peer's IP address here, but we don't need
# to do this with certificates, so use a wildcard address. The wildcard
# allows the remote device to be mobile or behind a NAT box
 remote_addr 0.0.0.0/0


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

b. 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-enigmax to US-partym"
 local_id_type dn
 local_id "O=exampleco, OU=IT, C=US, CN=enigma"
 remote_id "O=exampleco, OU=IT, C=US, CN=partym"

 local_addr  192.168.116.16
 remote_addr 0.0.0.0/0
...
```

6 在对等方系统上，启用 IKE。

```
partym # svcadm enable ipsec/ike

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。

▼ 如何使用 CA 签名的证书配置 IKE

证书颁发机构 (Certificate Authority, CA) 颁发的公共证书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。

开始之前 除了 `solaris.admin.edit/etc/inet/ike/config` 授权以外，您还必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。`root` 角色具有所有这些权限。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1 使用 `ikecert certlocal -kc` 命令创建证书请求。

有关该命令的参数的说明，请参见步骤 1中的第 131 页中的 “如何使用自签名的公钥证书配置 IKE”。

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

a. 例如，以下命令在 `partym` 系统上创建证书请求：

```
# ikecert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany\, Inc., OU=US-Partym, CN=Partym" \
> -A "DN=C=US, O=PartyCompany\, Inc., OU=US-Partym"
Creating software private keys.
  Writing private key to file /etc/inet/secret/ike.privatekeys/2.
Enabling external key providers - done.
Certificate Request:
  Proceeding with the signing operation.
  Certificate request generated successfully (.../publickeys/0)
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwUzELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zckO80mO9X
-----END CERTIFICATE REQUEST-----
```

b. 以下命令在 `enigma` 系统上创建证书请求：

```
# ikecert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax, CN=Enigmax" \
> -A "DN=C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax"
Creating software private keys.
...
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
8qlqdjaStLGfhDOO
-----END CERTIFICATE REQUEST-----
```

2 **将证书请求提交到 PKI 组织。**

PKI 组织可以告诉您如何提交证书请求。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，将证书请求粘贴到表单中。组织在检查您的请求后，将向您发出以下两个证书对象和已撤销证书的列表：

- 公钥证书－此证书基于您提交给组织的请求。所提交的请求是此公钥证书的一部分。证书可对您进行唯一标识。
- 证书颁发机构－组织的签名。CA 检验公钥证书是否合法。
- 证书撤销列表 (Certificate Revocation List, CRL)－组织已撤销的证书的最新列表。如果在公钥证书中嵌入对 CRL 的访问，则不会将 CRL 作为证书对象单独发送。

  在公钥证书中嵌入 CRL 的 URI 时，IKE 可以自动检索 CRL。同样，在公钥证书中嵌入 DN（LDAP 服务器上的目录名称）项时，IKE 可以从指定的 LDAP 服务器检索并高速缓存 CRL。

  有关公钥证书中的嵌入式 URI 和嵌入式 DN 项的示例，请参见第 143 页中的 “如何处理证书撤销列表”。

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，请参见第 119 页中的 “IKE，使用公钥证书”。

a. **成为管理员。**

有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

b. **添加从 PKI 组织收到的公钥证书。**

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

c. **添加来自 PKI 组织的 CA。**

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

d. **如果 PKI 组织已发送撤销证书列表，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
<Control>-D
```

4 **在 /etc/inet/ike/config 文件中使用 cert_root 关键字标识 PKI 组织。**
使用 PKI 组织提供的名称。

a. 例如，partym 系统上 ike/config 文件的显示可能与以下信息类似：

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform
 { auth_method rsa_sig oakley_group 1 auth_alg sha384 encr_alg aes}
p2_pfs 2

{
 label "US-partym to JA-enigmax - Example PKI"
 local_id_type dn
 local_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
 remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

 local_addr  192.168.13.213
 remote_addr 192.168.116.16

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

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 的 ike/config 文件应该满足以下要求：

- 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=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"
...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id   "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213
...
```

5 **通知 IKE 如何处理 CRL。**

选择适当的选项：

- **未提供 CRL**

  如果 PKI 组织未提供 CRL，则将关键字 `ignore_crls` 添加到 `ike/config` 文件。

  ```
  # Trusted root cert
  ...
  cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example,...
  ignore_crls
  ...
  ```

  `ignore_crls` 关键字指示 IKE 不搜索 CRL。

- **提供了 CRL**

  如果 PKI 组织提供了 CRL 的中心分发点，则可以修改 `ike/config` 文件以指向该位置。

  有关示例，请参见第 143 页中的 “如何处理证书撤销列表”。

示例 10–2 配置 IKE 时使用 rsa_encrypt

在 `ike/config` 文件中使用 `auth_method rsa_encrypt` 时，必须将对等方的证书添加到 `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` 管理员将发送以下电子邮件：

   ```
   To: admin@ja.enigmaexample.com
   From: admin@us.partyexample.com
   Message: -----BEGIN X509 CERTIFICATE-----
   MII...
   ----END X509 CERTIFICATE-----
   ```

   `enigma` 管理员将发送以下电子邮件：

   ```
   To: admin@us.partyexample.com
   From: admin@ja.enigmaexample.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` 方法隐藏对等方的标识，IKE 无法检索对等方的证书。因此，`rsa_encrypt` 方法要求 IKE 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 数据库包含每对通信系统的三个证书：

- 您的公钥证书
- CA 证书
- 对等方的公钥证书

故障排除－IKE 有效负荷（它包括这三个证书）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。

▼ 如何在硬件中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。带有标记 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，请参见第 152 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 6000 板”。

除了 solaris.admin.edit/etc/inet/ike/config 授权以外，您还必须成为分配有 "Network IPsec Management"（网络 IPsec 管理）权限配置文件的管理员。root 角色具有所有这些权限。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 ssh 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1 生成自签名证书或证书请求，并指定标记 ID。

选择以下选项之一：

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。

- **对于自签名证书，请使用此语法。**

```
# ikecert certlocal -ks -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

  -T 选项的参数是来自已连接 Sun Crypto Accelerator 6000 板的标记 ID。

- **对于证书请求，请使用此语法。**

```
# ikecert certlocal -kc -m 2048 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

有关 ikecert 命令的参数的说明，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、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 – PIN 响应**以明文形式**存储在磁盘上。

键入口令后，将输出证书内容：

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

3 **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。

- **将证书请求发送到处理 PKI 的组织。**

  按照 PKI 组织的说明提交证书请求。有关更详细的论述，请参见第 136 页中的 “如何使用 CA 签名的证书配置 IKE”中的步骤 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 文件的显示与以下信息类似：

```
# 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-enigmax to US-partym"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

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

- **证书请求**

  将 PKI 组织提供的名称作为 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信息类似：

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

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

5 **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 PKI 组织的证书。**

  添加组织从证书请求生成的证书，然后添加证书颁发机构 (certificate authority, CA)。

  ```
  # 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 PKI 组织的证书撤销列表 (certificate revocation list, CRL)，请参见第 143 页中的 “如何处理证书撤销列表”。

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。

▼ 如何处理证书撤销列表

证书撤销列表 (certificate revocation list, CRL) 包含来自证书颁发机构的过时证书或已损坏证书。CRL 的处理方式有四种。

- 如果 CA 组织未发出 CRL，则您必须指示 IKE 忽略 CRL。第 136 页中的 “如何使用 CA 签名的证书配置 IKE”的步骤 5 展示了此选项。
- 可以指示 IKE 从一个 URI（uniform resource indicator，统一资源指示符）访问 CRL，该 URI 的地址嵌入到来自 CA 的公钥证书中。
- 可以指示 IKE 从 LDAP 服务器访问 CRL，该服务器的 DN（directory name，目录名称）项嵌入到来自 CA 的公钥证书中。
- 可以将 CRL 作为 `ikecert certrldb` 命令的参数提供。有关示例，请参见示例 10–3。

以下过程介绍如何指示 IKE 从中心分发点使用 CRL。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

1 显示从 CA 收到的证书。

```
# ikecert certdb -lv certspec
```

`-l` 列出 IKE 证书数据库中的证书。

`-v` 以详细模式列出证书。应谨慎使用此选项。

certspec 是一种与 IKE 证书数据库中的证书匹配的模式。

例如，以下证书由 Oracle 颁发。详细信息已更改。

```
# ikecert certdb -lv example-protect.oracle.com
Certificate Slot Name: 0   Type: dsa-sha1
   (Private key in certlocal slot 0)
 Subject Name: <O=Oracle, CN=example-protect.oracle.com>
 Issuer Name: <CN=Oracle CA (Cl B), O=Oracle>
 SerialNumber: 14000D93
   Validity:
      Not Valid Before: 2011 Sep 19th, 21:11:11 GMT
      Not Valid After:  2015 Sep 18th, 21:11:11 GMT
   Public Key Info:
      Public Modulus  (n) (2048 bits): C575A...A5
      Public Exponent (e) (  24 bits): 010001
   Extensions:
      Subject Alternative Names:
              DNS = example-protect.oracle.com
      Key Usage: DigitalSignature KeyEncipherment
      [CRITICAL]
   CRL Distribution Points:
      Full Name:
         URI = #Ihttp://www.oracle.com/pki/pkismica.crl#i
         DN = <CN=Oracle CA (Cl B), O=Oracle>
      CRL Issuer:
      Authority Key ID:
      Key ID:              4F ... 6B
      SubjectKeyID:        A5 ... FD
      Certificate Policies
      Authority Information Access
```

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` 文件的显示与以下信息类似：

  ```
  # Use CRL 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 own web proxy
  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信息类似：

  ```
  # Use CRL from organization's LDAP
  ldap-list "ldap1.oracle.com:389,ldap2.oracle.com"
  ...
  ```

在证书到期之前，IKE 检索并高速缓存 CRL。

示例 10–3 将 CRL 粘贴到本地 certrldb 数据库中

如果无法从中心分发点获取 PKI 组织的 CRL，则可以将该 CRL 手动添加到本地 `certrldb` 数据库。按照 PKI 组织的说明将 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 < Oracle.Cert.CRL
```

为移动系统配置 IKE（任务列表）

下表包含将 IKE 配置为处理远程登录到中心站点的系统的过程的链接。

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。	第 146 页中的 “如何为站点外系统配置 IKE”

任务	说明	参考
在接受来自移动系统的通信流量的中心系统上使用 CA 的公共证书和 IKE。	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。	示例 10–4
在没有固定 IP 地址的系统上使用 CA 的公共证书和 IKE。	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。	示例 10–5
在接受来自移动系统的通信流量的中心系统上使用自签名证书和 IKE。	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。	示例 10–6
在没有固定 IP 地址的系统上使用自签名证书和 IKE。	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。	示例 10–7

为移动系统配置 IKE

在进行适当配置后，家庭办公室和手提电脑可以使用 IPsec 和 IKE 与其公司的中央计算机进行通信。利用与公钥证书验证方法组合的综合 IPsec 策略，离站系统可以保护它们传输到中心系统的流量。

▼ 如何为站点外系统配置 IKE

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。使用自签名证书或来自 PKI 的证书，移动系统可以与中心站点进行通信。

开始之前 您必须承担 `root` 角色。有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

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 IKE 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 IKE 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。

```
## /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 服务。

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

示例 10–4 将中心计算机配置为接受来自移动系统的 IPsec 流量

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，请参见示例 10–5。

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site system with root certificate"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# CA's public certificate ensures trust,
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256}
}
```

示例 10–5 使用 IPsec 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，请参见示例 10–4。

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha256 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site mobile1 with root certificate"
  local_id_type DNS
  local_id "mobile1.domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the root certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha256 }
}
```

示例 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，请参见示例 10–7。

```
## /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 }
}
```

示例 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，请参见示例 10–6。

```
## /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。

将 IKE 配置为查找连接的硬件

公钥证书也可存储于连接的硬件上。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。

▼ 如何将 IKE 配置为查找 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 Version 1.1 User's Guide》 (`http://download.oracle.com/docs/cd/E19321-01/820-4144-12/820-4144-12.pdf`)（《Sun Crypto Accelerator 6000 板 1.1 版用户指南》）。

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《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

如果您以远程方式登录，请使用 `ssh` 命令实现安全的远程登录。有关示例，请参见示例 7–1。

1 检查是否已链接 PKCS #11 库。

IKE 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。键入以下命令，以确定 PKCS #11 库是否已链接：

```
$ ikeadm get stats
...
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

2 **查找已连接的 Sun Crypto Accelerator 6000 板的标记 ID。**

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                          "
```

该库返回一个包含 32 个字符的标记 ID（也称为 keystore name（密钥库名称））。在此示例中，可以将 Sun Metaslot 标记与 ikecert 命令一起使用来存储和加速 IKE 密钥。

有关如何使用标记的说明，请参见第 140 页中的 “如何在硬件中生成和存储公钥证书”。

结尾空格是由 ikecert 命令自动填充的。

示例 10–8 查找和使用 metaslot 标记

标记可以存储在磁盘上、连接的板上或加密框架提供的 softtoken 密钥库中。softtoken 密钥库标记 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。

◆ ◆ ◆ 第 11 章

11 Internet 密钥交换（参考信息）

本章包含有关 IKE 的以下参考信息：

有关实现 IKE 的说明，请参见第 10 章，配置 IKE（任务）。有关概述信息，请参见第 9 章，Internet 密钥交换（概述）。

IKE 服务

`svc:/network/ipsec/ike:default` **服务** – 服务管理工具 (Service Management Facility, SMF) 提供 `ike` 服务以管理 IKE。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 IKE 配置文件 `/etc/inet/ike/config`。

以下 `ike` 服务属性是可配置的：

- `config_file` **属性** – 为 IKE 配置文件的位置。初始值为 `/etc/inet/ike/config`。
- `debug_level` **属性** – 为 `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**对象类型**下有关调试级别的表。
- `admin_privilege` **属性** – 为 `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，请参见第 157 页中的 “ikeadm 命令”。

有关 SMF 的信息，请参见《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 章 “管理服务（概述）”。另请参见 `smf`(5)、`svcadm`(1M) 和 `svccfg`(1M) 手册页。

IKE 守护进程

in.iked 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。必须在计划以安全方式通信的所有系统上运行该守护进程。

缺省情况下，未启用 svc:/network/ipsec/ike:default 服务。配置了 /etc/inet/ike/config 文件并启用 ike 服务后，in.iked 守护进程会在系统引导时运行。

在 IKE 守护进程运行时，系统在阶段 1 交换中向其对等 IKE 实体进行自我验证。与验证方法一样，对等实体也是在 IKE 策略文件中定义的。然后守护进程建立阶段 2 交换的密钥。按照在策略文件中指定的时间间隔，自动刷新 IKE 密钥。in.iked 守护进程通过 PF_KEY 套接字侦听从网络传入的 IKE 请求，并侦听外发通信流量请求。有关更多信息，请参见 pf_key(7P) 手册页。

有两个命令支持 IKE 守护进程。ikeadm 命令可用于查看并临时修改 IKE 策略。要永久修改 IKE 策略，请修改 ike 服务的属性。要修改 IKE 服务的属性，请参见第 107 页中的 “如何管理 IPsec 和 IKE 服务”。ikeadm 命令也可用于查看阶段 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。

使用 ikecert 命令可以查看和管理公钥数据库。此命令管理本地数据库 ike.privatekeys 和 publickeys。它还管理公钥操作和公钥在硬件上的存储。

IKE 配置文件

IKE 配置文件 /etc/inet/ike/config 管理 IPsec 策略文件 /etc/inet/ipsecinit.conf 中受保护的接口的密钥。

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。全局参数包括诸如已连接硬件加速器路径之类的项。有关 IKE 策略文件的示例，请参见第 125 页中的 “使用预先共享的密钥配置 IKE（任务列表）”。有关 IKE 策略项的示例和说明，请参见 ike.config(4) 手册页。

IKE 支持的 IPsec SA 根据 IPsec 配置文件 /etc/inet/ipsecinit.conf 中的策略来保护 IP 数据报。IKE 策略文件确定是否在创建 IPsec SA 时使用完全转发保密 (perfect forward security, PFS)。

/etc/inet/ike/config 文件可以包括按照以下标准实现的库的路径：RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)。IKE 使用此 PKCS #11 库访问用于密钥加速和密钥存储的硬件。

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，请参见第 113 页中的 “ipsecinit.conf 和 ipsecconf 的安全注意事项”。

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　　不能查看或修改加密材料。base 级别是缺省特权级别。

modkeys 级别　　可以删除、更改和添加预先共享的密钥。

keymat 级别　　可以使用 ikeadm 命令查看实际的加密材料。

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。有关过程，请参见第 107 页中的 “如何管理 IPsec 和 IKE 服务”。

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，请参见第 114 页中的 “ipseckey 的安全注意事项”。

IKE 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。ike.preshared 文件包含用于 Internet 安全关联和密钥管理协议 (Internet Security Association and Key Management Protocol, ISAKMP) SA 的预先共享的密钥。ipseckeys 文件包含用于 IPsec SA 的预先共享的密钥。按 0600 保护这些文件。按 0700 保护 secret 目录。

- 将 ike/config 文件配置为需要预先共享的密钥时，您需要创建 ike.preshared 文件。在 ike.preshared 文件中输入 ISAKMP SA 的加密材料（即用于 IKE 验证）。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。
- ipseckeys 文件包含 IPsec SA 的加密材料。有关手动管理该文件的示例，请参见第 103 页中的 “如何手动创建 IPsec 密钥”。IKE 守护进程不使用此文件。IKE 为 IPsec SA 生成的加密材料存储在内核中。

IKE 公钥数据库和命令

ikecert 命令处理本地系统的公钥数据库。在 ike/config 文件需要公钥证书时，可以使用此命令。由于 IKE 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中的其中一种：certlocal、certdb 和 certrldb。

ikecert 命令还处理密钥存储。密钥可以存储在磁盘上、连接的 Sun Crypto Accelerator 6000 板上或 softtoken 密钥库中。当加密框架中的 metaslot 用于和硬件设备进行通信时，softtoken 密钥库是可用的。ikecert 命令使用 PKCS #11 库定位密钥存储。

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。

ikecert tokens 命令

tokens 参数列出可用的标记 ID。使用标记 ID，ikecert certlocal 和 ikecert certdb 命令可以生成公钥证书和证书请求。证书和证书请求也可以由加密框架存储在 softtoken 密钥库中或存储在连接的 Sun Crypto Accelerator 6000 板上。ikecert 命令使用 PKCS #11 库定位证书存储。

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证书请求。-ks 选项用于创建自签名的证书。-kc 选项用于创建证书请求。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 11–1 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	auth_method dsa_sig	一种速度比 RSA 稍慢的验证方法。

表 11–1 ikecert 选项和 ike/config 项之间的对应关系 （续）

ikecert 选项	ike/config 项	说明
-t rsa-md5 和 -t rsa-sha1	auth_method rsa_sig	一种速度比 DSA 稍快的验证方法。 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	auth_method rsa_encrypt	RSA 加密防止窃听者知道 IKE 中的标识，但是要求 IKE 对等方知道彼此的公钥。

如果使用 ikecert certlocal -kc 命令发出证书请求，则会将该命令的输出发送到 PKI 组织或证书颁发机构 (certificate authority, CA)。如果您的公司运行自己的 PKI，则会将输出发送到 PKI 管理员。然后，PKI 组织、CA 或 PKI 管理员将创建证书。PKI 或 CA 返回给您的证书是 certdb 子命令的输入。PKI 返回给您的证书撤销列表 (certificate revocation list, CRL) 是 certrldb 子命令的输入。

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，请参见第 131 页中的 “如何使用自签名的公钥证书配置 IKE”。此命令还将您从 PKI 或 CA 接收的证书接受为输入。有关过程，请参见第 136 页中的 “如何使用 CA 签名的证书配置 IKE”。

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。

ikecert certrldb 命令

certrldb 子命令管理证书撤销列表 (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 PKI 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，请参见第 143 页中的 “如何处理证书撤销列表”。

/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。

插槽以编码形式包含在其他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 的证书签名请求的证书。也安装 CA 的证书。

/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。

/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。PKI 组织为其证书提供 CRL。可以使用 ikecert certrldb 命令填充数据库。

词汇表

3DES 请参见 Triple-DES（三重 DES）。

AES 高级加密标准 (Advanced Encryption Standard)。一种对称的 128 位块数据加密技术。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。

anycast address（任播地址） 为一组接口（通常属于不同的节点）指定的 IPv6 地址。发送到任播地址的包将被路由到最近的具有该地址的接口。包的路由符合路由协议的距离度量原则。

anycast group（任播组） 一组具有相同任播 IPv6 地址的接口。Oracle Solaris 实现的 IPv6 不支持创建任播地址和任播组。不过，Oracle Solaris IPv6 节点可以将通信流量发送到任播组。

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 数据报提供验证和完整性而不提供保密性的扩展头。

autoconfiguration（自动配置） 主机根据站点前缀和本地 MAC 地址自动配置其 IPv6 地址的过程。

bidirectional tunnel（双向隧道） 可以双向传输数据报的隧道。

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。

CA 请参见 certificate authority, CA（证书颁发机构）。

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。

classless inter-domain routing (CIDR) address（无类域间路由地址） 一种不基于网络类（A、B 和 C 类）的 IPv4 地址格式。CIDR 地址的长度为 32 位。它们使用标准的 IPv4 点分十进制表示法格式，并添加网络前缀。此前缀定义网络号和网络掩码。

class（类） 在 IPQoS 中，具有类似特征的一组网络流。可以在 IPQoS 配置文件中定义类。

datagram（数据报） 请参见 IP datagram（IP 数据报）。

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。

diffserv model（diffserv 模型） Internet 工程任务组体系结构标准，用于在 IP 网络上实现区分服务。主要模块是分类器、计量器、标记器、调度程序和丢包器。IPQoS 实现分类器、计量器和标记器模块。diffserv 模型在 RFC 2475 *An Architecture for Differentiated Services* 中进行了介绍。

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。

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。

DS codepoint, DSCP（DS 代码点） 一个 6 位值，包含在 IP 数据包头的 DS 字段中时指示必须转发包的方式。

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。

dual stack（双栈） 一种 TCP/IP 协议栈，IPv4 和 IPv6 均位于网络层，栈的其余部分是完全相同的。如果在安装 Oracle Solaris 的过程中启用 IPv6，则主机将收到 TCP/IP 的双栈版本。

dynamic packet filter（动态包过滤器） 请参见 stateful packet filter（有状态包过滤器）。

dynamic reconfiguration, DR（动态重新配置） 一种功能，允许您在系统运行的同时重新配置系统，而对正在进行的操作影响很小或者没有任何影响。并非所有 Oracle Sun 平台都支持 DR。有些 Oracle Sun 平台可能仅支持某些类型硬件（例如 NIC）的 DR。

encapsulating security payload, ESP（封装安全有效负荷） 为数据报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。

filter（过滤器） IPQoS 配置文件中定义类特性的规则集合。IPQoS 系统选择符合其 IPQoS 配置文件中过滤器的任何通信流以进行处理。请参见 packet filter（包过滤器）。

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）。

flow accounting（流记帐） 在 IPQoS 中，累积和记录有关通信流的信息的过程。通过在 IPQoS 配置文件中定义 `flowacct` 模块的参数，可以建立流记帐。

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。

header（头） 请参见 IP header（IP 数据包头）。

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。

hop（跃点） 用于标识分隔两个主机的路由器数量的度量。如果源主机和目标主机之间有三个路由器，则这两个主机之间有四个跃点。

host（主机） 不执行包转发的系统。在安装 Oracle Solaris 时，系统在缺省情况下成为主机，即系统无法转发包。一个主机通常具有一个物理接口，尽管它可以具有多个接口。

ICMP Internet Control Message Protocol（Internet 控制消息协议）。用于处理错误和交换控制消息。

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 datagram（IP 数据报） 通过 IP 传输的信息包。IP 数据报包含头和数据。头包括数据报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数据报附带的数据。

IP header（IP 数据包头） 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 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）。

IPQoS 一种软件功能，提供 diffserv model（diffserv 模型）标准的实现以及虚拟 LAN 的流记帐和 802.1 D 标记。使用 IPQoS，可以为用户和应用程序提供不同级别的网络服务（如 IPQoS 配置文件中所定义）。

IPsec IP security（IP 安全性）。为 IP 数据报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。密钥库名称也称为标记或标记 ID。

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。

load spreading（负荷分配） 在一组接口中分配传入或传出通信的过程。通过负荷分配，可以获得较高的吞吐量。仅当网络通信流向使用多个连接的多个目标时，才会发生负荷分配。负荷分配有两种类型：传入负荷分配（对于传入通信）和传出负荷分配（对于外发通信）。

local-use address（本地使用地址） 只能在本地范围内（在子网内或在用户网络内）路由的单播地址。此地址还可以具有本地或全局唯一性范围。

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数据报的虚拟 LAN 标记。用户优先级值指示使用 VLAN 设备在网络中转发数据报的方式。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。

meter（计量器） diffserv 体系结构中的一个模块，用于度量特定类的通信流速率。IPQoS 实现包括以下两个计量器：`tokenmt` 和 `tswtclmt`。

minimal encapsulation（最小封装） 家乡代理、外地代理和移动节点支持的可选 IPv4 嵌套隧道传送形式。最小封装的系统开销比 IP-in-IP 封装少 8 或 12 个字节。

MTU Maximum Transmission Unit（最大传输单元）。可以通过链路传输的大小，以八位字节表示。例如，以太网的 MTU 是 1500 个八位字节。

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（网络地址转换）。

neighbor advertisement（相邻节点通告） 对相邻节点的请求消息的响应，或一个节点发送未经请求的相邻节点通告以通告链路层地址更改的过程。

neighbor discovery（相邻节点搜索） 一种 IP 机制，使主机可以查找驻留在已连接链路上的其他主机。

neighbor solicitation（相邻节点请求） 由一个节点发送的请求，用于确定相邻节点的链路层地址。相邻节点请求还通过高速缓存的链路层地址验证相邻节点是否仍然可以访问。

network address translation（网络地址转换） NAT。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。

node（节点） 在 IPv6 中，启用了 IPv6 的任何系统，而不管是主机还是路由器。

outcome（结果） 作为计量通信流量的结果而执行的操作。IPQoS 计量器具有三种结果：红色、黄色和绿色，如在 IPQoS 配置文件中所定义。

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 数据包头）。

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 数据包头）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。

per-hop behavior, PHB（单跳行为） 为通信类指定的优先级。PHB 指示该类的流相对其他通信类的优先顺序。

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。

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。

private address（专用地址） 无法通过 Internet 进行路由的 IP 地址。无需 Internet 连通性的主机上的家乡网络可以使用专用地址。这些地址在 Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`) 中进行了定义，通常称为 "1918" 地址。

protocol stack（协议栈） 请参见 IP stack（IP 栈）。

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。

redirect（重定向） 在路由器中，通告主机有一个更好的第一跃点节点可以到达特定目标。

repair detection（修复检测） 检测 NIC 或从 NIC 到某个第 3 层设备的路径在出现故障后何时开始正常工作的过程。

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。

reverse tunnel（反向隧道） 开始于移动节点的转交地址并结束于家乡代理的隧道。

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。

SA 请参见 security association, SA（安全关联）。

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。

SCTP 请参见 streams control transport protocol（流控制传输协议）。

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。

selector（选定器） 一个元素，专门用于定义应用于特定类的包的条件，以便从网络流中选择该类通信流量。可以在 IPQoS 配置文件的过滤子句中定义选定器。

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。

site-local-use address（站点本地使用的地址） 用于在单个站点上寻址的标识。

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。

SPD 请参见 security policy database, SPD（安全策略数据库）。

SPI 请参见 security parameter index, SPI（安全参数索引）。

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。

stack（栈） 请参见 IP stack（IP 栈）。

standby（待机接口） 不用来传输数据通信流量的物理接口，除非某个其他物理接口出现故障。

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。

stateless autoconfiguration（无状态自动配置） 主机通过组合其 MAC 地址和 IPv6 前缀（由本地 IPv6 路由器通告）生成自己的 IPv6 地址的过程。

stream control transport protocol（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。DES 就是一个对称密钥系统。

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。

tunnel（隧道） datagram（数据报）在被封装时跟踪的路径。请参见 encapsulation（封装）。

unicast address（单播地址） 标识启用了 IPv6 的节点的单个接口的 IPv6 地址。单播地址包括以下几部分：站点前缀、子网 ID 和接口 ID。

user-priority（用户优先级） 一个实现服务类标记的 3 位值，它定义如何在 VLAN 设备网络中转发以太网数据报。

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

E

F

H

I

K

L

M

N

命

目

内

配

欺

请