

## 第8章

# HTTP 服务 (208)

本主题的权重为 11 分,包含以下目标:

目标 208.1; Apache 基本配置 (4 分) 考生应该能够安装和配置 Web 服务器。此目标包括监视服务器的负载和性能、限制客户端用户访问、将脚本语言支持配置为模块以及设置客户端用户身份验证。还包括配置服务器选项以限制资源的使用。考生应该能够配置 Web 服务器以使用虚拟主机和自定义文件

使用权。

目标 208.2; HTTPS 的 Apache 配置 (3 分) 考生应该能够配置 Web 服务器以提供 HTTPS。

目标 208.3; Implementing Squid as a caching proxy (2分)考生应该能够安装和配置代理服务器,包括访问策略、身份验证和资源使用。

目标 208.4;将 Nginx 实现为 Web 服务器和反向代理 (2 分) 考生应该能够安装和配置反向代理服务器 Nginx。包括 Nginx 作为 HTTP 服务器的基本配置。

## 基本 Apache 配置 (208.1)

考生应该能够安装和配置 Web 服务器。该目标包括监控服务器的负载和性能、限制客户端用户访问、将脚本语言支持配置为模块以及设置客户端用户身份验证。还包括配置服务器选项以限制资源的使用。考生应该能够配置 Web 服务器以使用虚拟主机和自定义文件访问。

### 关键知识领域

Apache 2.x 包括 2.4 配置文件、术语和实用程序

Apache 日志文件配置和内容

访问限制方法和文件

mod\_perl 和 PHP 配置

客户端用户身份验证模块、文件和实用程序

配置最大请求、最小和最大服务器和客户端

Apache 2.x 虚拟主机实现 (有和没有专用 IP 地址)

在 Apache 的配置文件中使用重定向语句来自定义文件访问

## 条款和实用程序

- access.log 或 access\_log
- error.log 或 error\_log
- .htaccess
- httpd.conf
- mod\_auth
- mod\_authn\_file
- mod\_access\_compat
- 密码
- AuthUserFile,AuthGroupFile
- apache2ctl
- httpd

资源: [LinuxRef06](#); [LPIC2sybex2nd](#);粗00;诗人99;威尔逊00; [Engelschall00](#); [PerlRef01](#);克劳斯01;阿帕奇博士; [apache24upgra digochanges](#); [apache24升级](#); [apache24升级](#);各种命令的手册页。

## 安装 Apache 网络服务器

当 Apache 出现时,从源代码构建 Apache 是例行公事。如今,大多数现代 (2005 年后)Linux 发行版都可以使用二进制格式的 Apache。从源安装程序已包含在 206.1 中。因此,在本章中,我们将专注于使用 rpm 和 apt 包管理器和工具。不要低估从源代码构建 Apache 的重要性。根据要求和 (缺乏) 可用性,可能仍然需要从源代码编译 Apache 和 Apache 模块。可以使用影响服务器行为的某些命令行选项调用 Apache 二进制 httpd。但一般来说,Apache 是由其他充当 httpd 包装器的脚本启动的。这些脚本应该负责将所需的标志传递给 httpd。服务器的行为是通过设置称为指令的各种选项来配置的。这些指令在配置文件中声明。配置文件的位置及其组织方式各不相同。Red Hat 和类似的发行版在 /etc/httpd/conf 目录中有它们的配置文件。

其他正在使用或已经使用的位置是 /etc/apache/config、/etc/httpd/config 和 /etc/apache2。

根据您的 Linux 发行版和启用的存储库,您的发行版可能附带 Apache 2.2 或 Apache 2.4 或两者。Apache 2.0 因其名称而符合 LPIC-2 Apache 2.x 范围,但不再维护 Apache 2.0。

因此不推荐使用 Apache 2.0。相反,Apache 基金会建议使用 Apache 2.4。然而,作为 Linux 管理员,您可能仍会在服务器上遇到 Apache 2.0。因此建议熟悉不同版本之间的配置差异。Apache 基金会确实提供了指导:可以通过<https://httpd.apache.org/docs/>访问升级文档,其中介绍了从 Apache 2.0 升级到 2.2 以及从 Apache 2.2 升级到 2.4 时的必要步骤。

重要的是要区分影响 Apache 服务器进程的 (全局) 指令和影响 Apache 服务器特定组件的选项,即仅影响特定网站的选项。配置文件的布局方式通常可以作为配置设置位置的线索。尽管这假定是显而易见的,但不要做出假设也很重要。始终熟悉所有已配置的选项。当对特定选项有疑问时,请使用文档或网络搜索来了解更多信息并考虑是否正确配置了该选项。

在许多 (基于 Red Hat 的) 发行版中,主要的 Apache 配置文件是 httpd.conf,其他 (基于 Debian 的) 发行版更喜欢 apache2.conf 文件名。根据您的分发和安装,它可能是一个大文件或一个通过 Include 和/或 IncludeOptional 语句引用其他配置文件的通用小文件。这两个指令的区别在于可选部分。如果 Apache 配置为包含某个目录中的所有 \*.conf 文件,则必须至少有一个文件与要包含的模式匹配。否则,Apache 服务器将无法启动。作为替代方案,IncludeOptional 指令可用于包含存在且可访问的配置文件。Apache 主配置文件可以配置通用设置,例如服务器名、侦听端口以及这些端口应该是哪个 IP 地址。

势必。虽然也可能有一个单独的 ports.conf 配置文件,因此请始终遵循 Include 指令并熟悉所有配置文件的内容。Apache 应该运行的用户和组也可以从主配置文件中配置。这些帐户可以设置为启动后切换。这样,Apache 软件可以以 root 用户身份启动,然后切换到专用的“www”、“httpd”或“apache”用户以遵守最小权限原则。还有各种指令可以影响 Apache 从其文档树中提供文件的方式。例如,有 Directory 指令控制是否允许执行位于其中的 PHP 文件。默认配置文件是不言自明的,包含很多有价值的信息。关于 LPIC-2 考试,您需要熟悉最常见的 Apache 指令。我们将在接下来的部分中介紹其中的一些内容。在撰写本文时,Apache 2.4 是最新的稳定版本,也是根据其发行商 Apache Foundation 推荐使用的版本。在适用的情况下,本书将尝试指出各个版本之间的差异。

为文档树的细分设置选项的另一种方法是使用 .htaccess 文件。出于安全原因,您还需要通过为该目录上下文设置 AllowOverride 指令来在主配置文件中启用 .htaccess 文件。.htaccess 文件中的所有选项都会影响目录中的文件及其下方的文件,除非它们被另一个 .htaccess 文件或主配置文件中的指令覆盖。

## 模块化

Apache 具有模块化的源代码架构。您可以自定义构建仅包含您真正需要的模块的服务器。Internet 上有许多模块,您也可以编写自己的模块。

模块是用 C 语言编写的编译对象。如果您对 Apache 模块的开发有疑问,请加入位于<http://httpd.apache.org/lists.html> 的 Apache 模块邮件列表。请记住先做功课:在发布问题之前研究过去的消息并检查 Apache 站点上的所有文档。

存在用于解释语言(如 Perl 和 Tcl)的特殊模块。它们允许 Apache 在本机运行解释脚本,而不必在每次脚本运行时重新加载解释器(例如 mod\_perl 和 mod\_tcl)。这些模块包括一个 API,允许以解释(脚本)语言编写的模块。

Apache 源代码的模块化结构不应与 Apache 模块的运行时加载功能相混淆。运行时模块在 Apache 的核心功能启动后加载,是一个相对较新的功能。在旧版本中,要使用模块的功能,需要在构建阶段进行编译。Apache 的当前实现能够进行运行时模块加载。**DSO**部分有更多详细信息。

## 模块的运行时加载 (DSO)

大多数现代 Unix 衍生产品都有一种机制,用于按需链接和加载所谓的动态共享对象(DSO)。这是一种在运行时将特殊程序加载到可执行文件的地址空间的方法。这通常可以通过两种方式完成:在可执行文件启动时通过名为 ld.so 的系统程序自动完成,或者通过系统调用 dlopen() 和 dlsym() 从正在执行的程序中手动完成。

在后一种方法中,DSO 通常称为共享对象或 DSO 文件,并且可以使用任意扩展名来命名。按照惯例,使用扩展名 .so。这些文件通常安装在特定于程序的目录中。可执行程序在运行时通过 dlopen() 手动将 DSO 加载到其地址空间中。

### 小费

如何将 Apache-SSL 作为可共享 (DSO) 模块运行：安装适当的包：

```
packagemanager installcommand 模块名称
```

根据您的发行版，配置文件可能会或可能不会相应地进行调整。始终检查其中一个配置文件中是否存在 LoadModule 行：

```
LoadModule apache_ssl_module 模块/libssl.so
```

此行可能属于 Apache 主配置文件或包含的配置文件之一。最近得到很多支持的一种结构是使用单独的模块可用目录和模块启用目录。这些目录是 Apache 配置目录中的子目录。模块安装在 modules-available 目录中，并且对 modules-enabled 目录中的符号链接进行包含引用。这个符号链接然后指向模块。Include 引用可能是一个通配符，包括某个目录中的所有文件。

另一个结构类似，但在 Apache 配置目录中包含一个 conf.modules.d 目录。这个文件实际上是一个符号链接，指向文件系统上其他地方的 Apache 程序目录中的一个目录。来自基于 Red Hat 的主机的示例：

```
包含 conf.modules.d/*.conf
```

同样，您可能遇到的实现可能彼此有很大不同。使用的 Linux 发行版、安装的 Apache 版本或 Apache 是否从包或源安装等各个方面都可能影响 Apache 的实现方式。更不用说值班管理员了。重要的是要记住，Apache 经常使用可能嵌套的配置文件。但是在层次结构的顶部总会有一个主要的 Apache 配置文件。

### 提

示要查看您的 Apache 版本是否支持 DSO，请执行命令 httpd -l，它会列出已编译到 Apache 中的模块。如果 mod\_so.c 出现在模块列表中，那么您的 Apache 服务器可以使用动态模块。

## APache 扩展 (APXS) 支持工具

APXS 是 Apache 1.3 及更高版本的新支持工具，可用于将 Apache 模块构建为 Apache 源代码树之外的 DSO。它知道用于制作 DSO 文件的依赖于平台的构建参数，并提供了一种使用它们运行构建命令的简单方法。

## 监控 Apache 负载和性能

可用于定期加载测试 Web 服务器页面的开源系统是 Cricket。Cricket 可以很容易地设置为记录页面加载时间，并且它有一个基于网络的绘图器，可以生成图表以多种格式显示数据。它基于 RRDtool，其祖先是 MRTG（“Multi-Router Traffic Grapher”的缩写）。RRDtool（Round Robin Data Tool）是一个在“round robin”数据库中收集数据的包；每个数据文件的大小都是固定的，因此运行 Cricket 不会慢慢填满您的磁盘。数据库表在创建时就确定了大小，并且不会随着时间的推移而变大。随着数据老化，它被平均化。

## 增强 Apache 性能

可用 RAM 不足可能会导致内存交换。交换网络服务器的性能会很差，尤其是在磁盘子系统达不到标准的情况下。导致用户停止并重新加载，进一步增加负载。您可以使用 MaxClients 设置来限制您的服务器可能生成的子节点数量，从而减少内存占用。建议通过 Apache 主配置文件对所有以 Min 或 Max 开头的指令进行 grep。这些设置为每个受影响的设置定义了 MINimal 和 MAXimum 边界。默认值应该在一方面空闲的服务器负载和另一方面处理重负载的可能性之间提供平衡。由于每条链的强度取决于它最薄弱的环节，因此应充分配置底层系统以处理预期的负载。LPIC-2 考试更侧重于检测 200.1 章中的这些性能瓶颈。

## Apache访问日志文件

access\_log 包含对您的网络服务器的页面请求的一般概述。访问日志的格式是高度可配置的。格式是使用看起来很像 C 风格 printf 格式字符串的格式字符串来指定的。访问日志的典型配置可能如下所示：

```
LogFormat "%h %l %u %t \ %r\% >s %b" 常见  
CustomLog 日志/access_log 通用
```

这定义了昵称 common 并将其与特定的日志格式字符串相关联。所示格式称为通用日志格式 (CLF)。它是许多网络服务器产生的标准格式，可以被大多数日志分析程序读取。CLF 中生成的日志文件条目将类似于此行：

```
127.0.0.1 - 鲍勃 [10/Oct/2000:13:55:36 -0100] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

CLF 包含以下字段：

1. 客户端IP地址(%h)
2. identd (%l) 确定的 RFC 1413 身份
3. 请求者的用户名 (%u)
4. 时间服务器完成服务请求 (%t)
5. 用户请求行(%r)
6. 服务器发送给客户端的状态码 (%s)
7. 返回对象的大小 (%b)。

## Apache error\_log 文件

服务器错误日志，其名称和位置由 ErrorLog 指令设置，是一个非常重要的日志文件。这是 Apache httpd 将向其发送诊断信息并记录它在处理请求时遇到的任何错误的文件。当启动服务器或操作服务器时出现问题时，这是查看的好地方。它通常会包含出错的细节以及如何修复它。

错误日志通常写入文件（在 Unix 系统上通常是 error\_log，在 Windows 上通常是 error.log）。在 Unix 系统上，也可以让服务器将错误发送到系统日志或通过管道将它们发送到程序。

错误日志的格式相对自由和描述性。但是大多数错误日志条目中都包含某些信息。例如，这是一条典型的消息：

```
[2000 年 10 月 11 日星期三 14:32:52] [错误] [客户端 127.0.0.1] 客户端被服务器 \ 配置拒绝：/export/home/live/ap/htdocs/test
```

日志条目中的第一项是消息的日期和时间。第二项列出了所报告错误的严重性。

LogLevel 指令用于通过限制严重级别来控制发送到错误日志的错误类型。第三项给出了产生错误的客户端的 IP 地址。除此之外是消息本身，在这种情况下，它表示服务器已配置为拒绝客户端访问。服务器报告所请求文档的文件系统路径（与 Web 路径相对）。

错误日志中可能会出现各种各样的不同消息。大多数看起来与上面的示例相似。错误日志还将包含 CGI 脚本的调试输出。由 CGI 脚本写入 stderr 的任何信息都将直接复制到错误日志中。

无法通过添加或删除信息来自定义错误日志。但是，处理特定请求的错误日志条目在访问日志中有相应的条目。例如，上面的示例条目对应于状态代码为 403 的访问日志条目。由于可以自定义访问日志，因此您可以使用该日志文件获取有关错误情况的更多信息。

在测试期间，连续监视错误日志以查找任何问题通常很有用。在 Unix 系统上，您可以使用以下方法完成此操作：

```
tail -f 错误日志
```

知道如何自定义 Apache 日志记录可能被证明是一项非常有用的技能。手动查看 Apache 日志不适合胆小的人。对于低流量服务器,这可能仍然可行。否则,通过在为多个网站提供服务的繁忙服务器上筛选日志来查找信息,可能会变成一项非常紧张的文本文件操作练习。这就产生了一个悖论:几乎没有日志记录,在查找问题原因时几乎没有任何输入可用。使用非常精细的日志记录,信息可能会非常庞大。因此,Apache 日志通常由外部设施解释。日志被发送到具有可视化统计数据和识别模式能力的系统或由系统读取。为确保提供的日志记录足够,可能需要先自定义 Apache 日志记录。

Apache 2.3.6 及更高版本提供了在每个模块或每个目录的基础上启用不同类型的 LogLevel 配置的可能性。关于 Loglevel 指令的 Apache 文档非常出色,我们可以添加的内容不多。

## 限制客户端用户访问

许多系统使用 DAC 或 MAC 来控制对对象的访问:

自主访问控制 (DAC) 采用 DAC 的系统允许用户自己设置对象权限。他们能自行决定更改这些。

强制访问控制 (MAC) 使用 MAC 的系统将其所有对象 (例如,文件) 置于系统的严格控制之下行政人员。不允许用户自己设置任何权限。

Apache 采取自由主义立场,将自由控制定义为基于用户名和密码的控制,将强制控制定义为基于静态或准静态数据 (如请求客户端的 IP 地址) 的控制。

Apache 使用模块来验证和授权用户。首先要明确认证和授权的区别。身份验证是用户验证其身份的过程。这是谁的部分,授权是决定允许谁做什么的过程。授权允许或拒绝向 Apache 服务器发出的请求。

授权依赖于身份验证来做出这些决定。

用于身份验证目的的 Apache 模块遵循 mod\_authn\_\* 的命名约定。用于授权目的的模块遵循 mod\_authz\_\* 的约定。此规则的一个例外是 mod\_authnz\_ldap 模块。正如您可能已经猜到的那样,由于 LDAP 的性质,此模块可以帮助进行身份验证和授权。

这些模块在文件系统上的位置可能不同。大多数发行版在 Apache 配置目录中创建一个 modules、modules.d 或 modules-available 目录。这个目录很可能是指向文件系统上其他地方目录的符号链接。这可以通过从模块目录中调用 pwd -P 或 ls -ld 来确定,如以下示例所示:

```
[user@redhatbased /etc/httpd]$ pwd -P /usr/lib64/httpd/  
modules
```

在上面的示例中,符号链接 /etc/httpd/modules 提供了从 Apache 配置文件中轻松引用模块的方法。Apache 模块使用 LoadModule 指令加载。该指令期望模块的路径相对于 ServerRoot 指令声明的 Apache 配置目录。

通常,模块将使用某种形式的数据库来存储和检索凭证数据。例如,mod\_authn\_file 模块使用文本文件,而 mod\_auth\_dbm 使用 Unix DBM 数据库。

下面是作为标准 Apache 发行版的一部分包含的一些模块的列表。

mod\_auth\_file (DAC) 这是大多数 Apache 安全模块的基础;它使用普通文本文件进行身份验证  
数据库。

mod\_access (MAC) 这曾经是标准 Apache 发行版中唯一应用 Apache 定义为强制控制的模块。它过去允许您列出允许或拒绝访问文档的主机、域和/或 IP 地址或网络。从 Apache 2.4 开始，不再使用此模块。Apache 2.4 及更新版本使用更新的身份验证和授权模型。这个新模型还带有新模块、新指令和新语法。

mod\_access 模块仍然是 LPIC-2 考试目标，因此您应该仍然熟悉 2.4 之前的语法。为了帮助向 Apache 2.4 迁移，Apache 2.4 附带了一个名为 mod\_access\_compat 的模块。该模块的目的是在 Apache 2.4 服务器上仍然接受 pre-2.4 语法。如果您在从以前的版本升级到 Apache 2.4 后遇到与 mod\_access 相关的错误，请确保 Apache 2.4 配置使用类似于以下的行加载此兼容性模块：

```
LoadModule mod_access_compat module/mod_access_compat.so
```

mod\_authn\_anon (DAC) 该模块模仿匿名 FTP 的行为。它不是拥有有效凭证的数据库，而是识别有效用户名列表（即 FTP 服务器识别“ftp”和“匿名”的方式）并授予对任何具有几乎任何密码的用户的访问权限。与实际访问控制相比，此模块对于记录对资源的访问和防止机器人进入更有用。

mod\_authn\_dbm (DAC) 与 mod\_auth\_db 类似，只是凭据存储在 Unix DBM 文件中。

mod\_auth\_digest (DAC) 此模块实现 HTTP 摘要身份验证 (RFC2617)，用于提供比 mod\_auth\_basic 功能更安全的替代方案。下面的解释很高兴知道但已经过时了。摘要式身份验证的全部意义在于防止用户凭据通过未加密的 HTTP 在线传输。然而，摘要模块使用的哈希算法已经严重过时了。使用摘要式身份验证而不是基本的 HTTP 身份验证在安全性方面不如使用 HTTPS 提供那么多优势。以下文档页面提供了更多详细信息：[http://httpd.apache.org/docs/2.4/mod/mod\\_auth\\_digest.html](http://httpd.apache.org/docs/2.4/mod/mod_auth_digest.html)。

在收到请求和用户名后，服务器将通过发送随机数来挑战客户端。nonce 的内容可以是任意（最好是 base 64 编码的）字符串，服务器可以使用 nonce 来防止重放攻击。例如，可以使用一分钟分辨率内的加密时间戳来构造随机数，即“201611291619”。

时间戳（可能还有识别请求的 URI 的其他静态数据）可能使用只有服务器知道的私钥进行加密。

收到随机数后，客户端计算收到的随机数、用户名、密码、HTTP 方法和请求的 URI 的哈希值（默认情况下为 MD5 校验和），并将结果发送回服务器。服务器将从本地摘要数据库检索的会话数据和密码数据中收集相同的数据。为了重建随机数，服务器将尝试两次：第一次尝试将使用当前时钟时间，第二次尝试（如有必要）将使用当前时钟时间减去一分钟。其中一次尝试应该给出与客户端计算的完全相同的哈希值。如果是这样，将授予对该页面的访问权限。

这将挑战的有效性限制为一分钟并防止重放攻击。

请注意，随机数的内容可以由服务器随意选择。提供的示例是多种可能性之一。与 mod\_auth 一样，凭证存储在文本文件（摘要数据库）中。摘要数据库文件使用 htdigest 工具进行管理。有关详细信息，请参阅模块文档。

mod\_authz\_host mod\_authz\_host 模块可用于向 Apache 请求特定的请求源。

mod\_authz\_host 模块对提供的参数非常灵活。由于模块的名称，提供主机名似乎是合乎逻辑的。虽然这确实有效，但它可能不是首选。该模块不仅需要对提供的主机名执行正向 DNS 查找以将其解析为数字 IP，该模块还配置为在执行正向查找后对解析的数字 IP 执行反向 DNS 查找。

因此，提供主机名会导致每个受影响的网络服务器请求至少进行两次 DNS 查找。如果反向 DNS 结果与提供的主机名不同，则无论配置如何允许，请求都将被拒绝。为了规避有关正向和反向 DNS 记录匹配的要求，可以在提供主机名时使用 forward-dns 选项。幸运的是，mod\_authz\_host 不仅接受主机名作为参数。它还可以处理（部分）IP 地址，包括 IPv4 和 IPv6，以及 CIDR 风格的符号。还有一个名为 local 的参数可用。这将转换为 127.0.0.0/8 或 ::1 环回地址以及服务器的已配置 IP 地址。当限制与本地主机相关的连接时，此设置可能会派上用场。由于解释 IP 地址的自由方式，建议在使用此模块时尽可能明确。例如，以下所有内容都被视为有效输入，并将由适用的规则进行解释：

```
要求主机:snow.nl 要求 ip:10.6.6 要求  
ip:172
```

需要ip:10.9.9.9/32 需要forward-dns:  
cloudhost.snow.nl 需要本地

Apache 2.2 和 2.4 之间值得注意的差异之一在于用于授权的指令。授权功能由 Apache mod\_authz\_\* 模块提供。以前版本的 Apache 使用 Order、Allow、Deny from 和 Satisfy 指令，Apache 2.4 使用名为 all、env、host 和 ip 的新指令。这些新的 from 指令对配置文件的语法有重大影响。为了帮助向后兼容性，从 Apache 2.4 过渡，mod\_access\_compat 模块仍然可以解释以前使用的授权指令。但是必须显式启用此模块。这样做可以保持对以前授权配置的支持。

当前的授权指令提供了关于谁有权做什么的更精细配置的可能性。这种增加的粒度主要来自 Require 指令的可用性。该指令在 Apache 2.4 之前已经可以用于身份验证目的。但是从 Apache 2.4 开始，该指令也可以由授权模块解释。

下面的示例将旧的和新的语法进行比较，同时提供相同的功能。

一、pre-2.4风格：

```
<Directory /lpic2bookdev> Order  
deny,allow Deny from all allow from  
10.6.6.0/24
```

需要集团员工  
满足任何  
</目录>

现在是相同的代码块，但使用 Apache 2.4 风格的语法：

```
<Directory /lpic2bookdev> <RequireAny>  
需要ip 10.6.6.0/24 需要群员工</RequireAny>  
</Directory>
```

新语法的好处在于效率。通过用更少的线路完成相同的功能，这些线路的处理将由人和计算机更有效地处理。计算机在完成相同结果的同时受益于花费更少的处理周期。人类受益于一个简短的配置部分。长配置更容易包含可能被忽略的错误。通过使用 RequireAll、RequireAny 和 RequireNone 指令在配置文件中创建部分，这些配置可以包含细化规则，同时保持其可读性。

另一个值得一提的 2.4 变化与关于 mod\_auth 模块的 LPIC-2 考试目标有关。

从 Apache 2.1 开始，mod\_auth 模块的功能已被更具体的模块取代。这些模块之一，mod\_authn\_file 现在提供了以前由 mod\_auth 提供的功能。mod\_authn 文件允许使用保存用户名和密码的文件作为授权过程的一部分。可以创建此文件，其内容可以由 htpasswd 实用程序维护。当使用 mod\_auth\_digest 而不是 mod\_auth\_basic 时，应该使用 htdigest 实用程序。本书将重点介绍 mod\_auth\_basic 选项。htpasswd -c 选项将在创建用户名和密码对期间创建一个文件，其中提供的参数作为文件名。htpasswd 允许创建 crypt、MD5 或 SHA1 密码算法。从 Apache 2.4.4 开始，也可以使用 bcrypt 作为密码加密算法。明文密码也可以使用 htpasswd -p 选项生成，但仅当 Apache 2.4 托管在 Netware 和 MS Windows 平台上时才有效。crypt 算法曾经是 htpasswd 默认算法，直到 Apache 版本 2.2.17，但被认为是不安全的。Crypt 会将提供的密码限制为前八个字符。从第 9 个字符开始的密码字符串的每一部分都将被忽略。地穴密码字符串容易被快速暴力破解，因此会带来相当大的安全风险。应尽可能避免使用 crypt 算法。相反，应在可用时考虑 bcrypt 算法。在装有 Apache 2.4.4 或更高版本的系统上，可以使用以下语法创建新的密码文件 htpasswdfile，为它提供用户“bob”并使用 bcrypt 算法为用户帐户设置密码：

```
htpasswd -cB /path/outside/document/root/htpasswdfile 鲍勃
```

系统将要求输入新密码两次。要在以后随时通过添加用户“alice”来更新此文件，可以省略-c选项以防止文件被重写：

```
htpasswd -B /path/outside/document/root/htpasswdfile 爱丽丝
```

将 brypt 算法与 htpasswd 一起使用还可以使用 -C 选项。使用此选项，用于计算密码哈希的计算时间可能会受到影响。默认情况下，系统使用设置 5。可以提供 4 到 31 之间的值。根据可用资源，在提高安全性的同时生成最大 18 的值应该是可以接受的。要将用户 eve 添加到现有的 htpasswd 文件中，同时将计算时间的值增加到 18，可以使用以下语法：

```
htpasswd -B -C18 /path/outside/document/root/htpasswdfile 前夕
```

在上面的示例中，建议在网络服务器文档树之外创建密码文件。否则，客户端可能会下载密码文件。

要使用生成的密码文件进行身份验证，Apache 必须知道 htpasswdfile 文件。这可以通过定义 AuthUserFile 指令来完成。该指令可以在 Apache 配置文件或单独的 .htaccess 文件中定义。该 .htaccess 文件应该位于它应该代表的文档根目录中。负责该文档根目录的 Apache 配置应该指定 AllowOverride 指令。这样，Apache 将覆盖其配置中包含 .htaccess 文档的目录的指令。.htaccess 文档的语法与 Apache 配置文件的语法相同。用于用户身份验证的代码块可能如下所示：

```
<目录/web/document/root>
AuthName “需要身份验证”
授权类型基本
AuthUserFile /path/outside/document/root/htpasswdfile
需要有效用户
文档根目录 /web/document/root
</目录>
```

查看 Apache 模块目录的内容以了解是否存在 mod\_auth\* 文件。有多个身份验证和授权模块可用。每个人都有自己的目的，有些人相互依赖。每个模块都在 Apache 中添加功能。此功能可以通过使用特定于模块的特定指令来解决。有关 Apache 2.4 可用模块的详细使用选项，请参阅 Apache 文档网站<https://httpd.apache.org/docs/2.4/mod/>。

## 配置身份验证模块

Apache 安全模块由配置指令配置。这些是从集中式配置文件（主要位于 /etc/ 目录下或中）或分散式 .htaccess 文件中读取的。后者主要用于限制对目录的访问，并放置在它们帮助保护的树的顶级目录中。例如，身份验证模块将使用 AuthUserFile 或 AuthDBMGroupFile 指令读取其数据库的位置。

集中配置 这是一个配置示例，因为它可能出现在集中配置文件中：

```
<目录/home/johnson/public_html> <文件 foo.bar>

AuthName Foo for Thought
AuthType Basic
AuthUserFile /home/johnson/foo.htpasswd 需要有效用户 </Files> </
Directory>
```

受保护的资源是 /home/johnson/public\_html 目录或任何底层子目录中的“任何名为 foo.bar 的文件”。同样，该文件指定谁有权访问 foo.bar：在 /home/johnson/foo.htpasswd 文件中具有凭据的任何用户。

分散配置 另一种方法是将 .htaccess 文件放在需要访问保护的任何文档树的顶级目录中。请注意，您必须在中央配置中设置指令 AllowOverride 才能启用此功能。

.htaccess 的第一部分确定应使用哪种身份验证类型。它可以包含要使用的密码或组文件的名称，例如：

```
AuthUserFile {密码文件的路径}
AuthGroupFile {组文件的路径}
AuthName {对话框的标题}
授权类型基本
```

.htaccess 的第二部分确保只有用户 {username} 可以访问 (GET) 当前目录：

```
<限制获取>
需要用户{用户名}
</极限>
```

限制部分可以包含其他指令以限制对某些 IP 地址或一组用户的访问。

以下将允许本地网络上的任何客户端（IP 地址 10.\*.\*.\*）访问 foo.html 页面并要求其他任何人提供用户名和密码：

```
<文件 foo.html>
Order Deny,Allow Deny
from All Allow from
10.0.0.0/8 AuthName “仅限内部人员”

AuthType Basic AuthUserFile /
/usr/local/web/apache/.htpasswd-foo Require valid-user 满足任意 </Files>
```

## 用户档案

mod\_auth 模块使用包含有效用户列表的纯文本文件。 htpasswd 命令可用于创建和更新这些文件。生成的文件是纯文本文件，任何编辑器都可以读取。它们包含“用户名：密码”形式的条目，其中密码是加密的。允许附加字段，但会被软件忽略。

htpasswd 使用为 Apache 修改的 MD5 版本或旧的 crypt() 例程来加密密码。您可以混合搭配。

概要  
htpasswd [-c] 密码文件用户名

以下是使用 htpasswd 创建 Apache 密码文件的两个示例。第一个用于在添加用户时创建新密码文件，第二个用于更改现有用户的密码。

```
$ htpasswd -c /home/joe/public/.htpasswd joe $ htpasswd /home/joe/public/.htpasswd
斯蒂芬
```

## 笔记

使用-c选项，如果指定的密码文件已经存在，将被覆盖！

## 组文件

Apache 可以使用组文件。群组文件包含群组名称,后跟群组中人员的姓名。通过授权一个组,该组中的所有用户都可以访问。组文件被称为 .htgroup 文件,按照惯例使用该名称 - 尽管您可以使用任何您想要的名称。组文件可以位于目录树中的任何位置,但通常放置在它们帮助保护的树的顶级目录中。要允许使用组文件,您需要在 Apache 主配置文件中包含一些指令。这通常在正确的目录定义中。在 AuthUserFile 可以指定绝对路径或相对路径的情况下,AuthGroupFile 指令将始终将提供的参数视为相对于 ServerRoot。AuthGroupFile 文件作为 AuthUserFile 的补充。该文件应在每一行包含一个组,后跟一个冒号。一个例子:

Apache 主配置文件:

```
...
AuthType Basic
AuthUserFile /var/www/.htpasswd AuthGroupFile /
var/www/.htgroup 需要组管理
...
...
```

关联的 .htgroup 文件可能具有以下语法:

```
管理:鲍勃·爱丽丝
会计:乔
```

现在帐户 “bob”和 “alice”可以访问该资源,但帐户 “joe”不会由于主配置文件中的 “Require group Management”语句而无法访问,因为 “joe”不是所需 “管理”的成员 “团体。为此, .htgroup 文件中指定的用户必须在 .htpasswd 文件中也有一个条目。

---

### 笔记

用户名可以在多个组条目中。这仅仅意味着用户是这两个组的成员。

---

要使用 DBM 数据库 (由 mod\_auth\_db 使用),您可以使用 dbmmanage。对于其他类型的用户文件/数据库,请参阅所选模块附带的文档。

---

### 注意确

保网络服务器可读取各种文件。

---

## 配置 mod\_perl

mod\_perl 是 Apache 的另一个模块,它将 Perl 解释器加载到您的 Apache 网络服务器中,减少了子进程的产生,从而减少了内存占用和对处理器能力的需求。另一个好处是代码缓存:模块和脚本只加载和编译一次,并将在网络服务器的剩余生命周期中从缓存中提供服务。

使用 mod\_perl 允许将 Perl 语句包含到您的网页中,如果请求页面,这些语句将动态执行。一个非常基本的页面可能如下所示:

```
打印“内容类型:文本/纯文本\r\n\r\n”;打印“你好,你这个东西!\n”;
```

mod\_perl 还允许您在 Perl 中编写新模块。您可以完全访问 Web 服务器的内部工作,并可以在请求处理的任何阶段进行干预。这允许定制处理 (仅举几个阶段)

URI->文件名翻译、身份验证、响应生成和日志记录。运行时开销非常小。

Apache 中的标准通用网关接口 (CGI) 可以完全替换为处理请求处理的响应生成阶段的 Perl 代码。mod\_perl 包括两个用于此目的的通用模块。第一个是 Apache:::

注册表,它可以透明地运行编写良好的现有 perl CGI 脚本。如果你的脚本写得不好,你应该重写它们。如果您缺少资源,您可以选择使用第二个模块 Apache::PerlRun,因为它不使用缓存并且比 Apache::Registry 更宽松。

您可以使用 PerlSetVar 和 <Perl> 部分在 Perl 中配置您的 httpd 服务器和处理程序。您还可以定义自己的配置指令,以供您自己的模块读取。

有许多方法可以安装 mod\_perl,例如作为 DSO,使用或不使用 APXS,从源代码或从 RPM。大多数可能的场景都可以在 Mod\_perl Guide PerlRef01 中找到。

#### 从源代码构建 Apache

要从源代码构建 Apache,您应该下载 Apache 源代码和 mod\_perl 的源代码,并将它们解压缩到同一目录中。1. 您需要在系统上安装最新版本的 perl。要构建模块,在大多数情况下,这些命令就足够了:

```
$ cd ${带有模块源代码的目录名称} $ perl Makefile.PL APACHE_SRC=../apache_x.xx/src \ DO_HTTPD=1 USE_APACI=1  
EVERYTHING=1 $制作&&制作测试&&制作安装
```

构建模块后,您还应该构建 Apache 服务器。这可以使用以下命令完成:

```
$ cd ${Apache 的目录名称} $ make install
```

剩下的就是向 httpd.conf (Apache 配置文件)添加几行配置并启动服务器。您应该添加哪些行取决于特定的安装类型,但通常一些 LoadModule 和 AddModule 行就足够了。

例如,这些都是您需要添加到 httpd.conf 以将 mod\_perl 用作 DSO 的行:

```
LoadModule perl_module 模块/libperl.so AddModule mod_perl.c  
PerlModule Apache::Registry
```

```
别名 /perl/ /home/httpd/perl/  
<位置/perl>  
SetHandler perl 脚本  
PerlHandler Apache::注册表  
选项 +ExecCGI  
PerlSendHeader 打开  
</位置>
```

前两行将在 Apache 启动时添加 mod\_perl 模块。在启动期间,PerlModule 指令确保指定的 Perl 模块也被读入。这通常是以 .pm 结尾的 Perl 包文件。Alias 关键字将对形式为 http://www.example.com/perl/file.pl 的 URI 的请求重新路由到目录 /home/httpd/perl。接下来,我们定义该位置的设置。通过设置 SetHandler,所有对目录 /home/httpd/perl 中的 Perl 文件的请求现在将被重定向到 perl 脚本处理程序,它是 Apache::Registry 模块的一部分。下一行只是允许在指定位置执行 CGI 脚本,而不是显示此文件。http://www.example 形式的任何 URI。com/perl/file.pl 现在将被编译一次并缓存在内存中。每当其源在磁盘上更新时,将通过重新编译 Perl 例程来刷新内存映像。将 PerlSendHeader 设置为 on 告诉服务器在每次脚本调用时向浏览器发送 HTTP 标头,但大多数时候最好使用 Apache Perl API 的 \$r->send\_http\_header 方法或使用 \$q->header 来自 CGI.pm 模块的方法。

#### 配置 mod\_php 支持

PHP 是一种服务器端、跨平台、HTML 嵌入式脚本语言。PHP 最初是由 Rasmus Lerdorf 在 1994 年底编写的快速 Perl hack。后来他用 C 重写了他的代码,因此“个人主页/表单解释器”(PHP/FI) 诞生了。

mod\_perl 模块可以在 [perl.apache.org](http://perl.apache.org) 获得, Apache 的源代码在 [www.apache.org](http://www.apache.org)

在接下来的两到三年里,它演变成 PHP/FI 2.0。 Zeev Suraski 和 Andi Gutmans 在 1997 年夏天编写了一个新的解析器,这导致了 PHP 3.0 的推出。 PHP 3.0 定义了版本 3 和版本 4 中使用的语法和语义。PHP 成为数百万 Web 开发人员事实上的编程语言。另一个版本的 (Zend) 解析器和更好的面向对象编程支持导致 2004 年 7 月推出了 5.0 版。随后出现了几个版本,并且第 6 版开始包含本机 Unicode 支持。然而这个版本被放弃了。2015 年计划启动 7.0 版。

可以从 CGI 接口调用 PHP,但常见的方法是在 Apache Web 服务器中将 PHP 配置为 (动态) DSO 模块。为此,您可以使用从 RPM 中提取的预构建模块,也可以从源代码中滚动您自己的模块<sup>2</sup>。您需要先配置 make 过程。要告诉配置将模块构建为 DSO,您需要告诉它使用 APXS:

```
./configure-with-apxs
```

.. 或者,如果您想指定 apxs 二进制文件的位置:

```
./configure -with-apxs={path-to-apxs}/apxs
```

接下来,您可以通过运行 make 命令来编译 PHP。成功编译所有源文件后,使用 make install 命令安装 PHP。

在 Apache 可以使用 PHP 之前,它必须了解 PHP 模块以及何时使用它。apxs 程序负责将 PHP 模块告知 Apache,因此剩下要做的就是将 .php 文件告知 Apache。文件类型在 httpd.conf 文件,它通常包含注释掉的有关 PHP 的行。您可能想要搜索这些行并取消注释它们:

添加类型 application/x-htpd-php .php

然后通过发出 apachectl restart 命令重新启动 Apache。apachectl 命令是另一种将命令传递到 Apache 服务器而不是使用 /etc/init.d/httpd 的方法。请查阅 apachectl(8) 联机帮助页以获取更多信息。

要测试它是否真的有效,请创建以下页面:

```
<HTML>
<HEAD><TITLE>PHP 测试</TITLE></HEAD>
<身体>
<?php phpinfo() ?> </BODY> </HTML>
```

将文件另存为 Apache 的 htdocs 目录中的 test.php,并将浏览器指向 http://localhost/test.php。应显示一个页面,其中包含 PHP 徽标和有关 PHP 配置的其他信息。请注意,PHP 命令包含在 <? 和 ?> 标签。

## httpd 二进制文件

httpd 二进制文件是 Apache 的实际 HTTP 服务器组件。在正常运行期间,建议使用 apachectl 或 apache2ctl 命令来控制 httpd 守护进程。在某些发行版中,httpd 二进制文件被命名为 apache2。

Apache 曾经是一个守护进程,仅在需要时分叉子进程。为了获得更好的响应时间,现在 Apache 也可以在预分叉模式下运行。这意味着服务器将提前生成许多子进程,准备好为任何通信请求提供服务。在大多数发行版中,预分叉模式默认运行。

## 配置 Apache 服务器选项

httpd.conf 文件包含许多允许您配置 Apache 服务器行为的部分。下面列出了一些关键字/部分。

<sup>2</sup> PHP4 的源代码可以在 [www.php.net](http://www.php.net) 获得

MaxKeepAliveRequests持久连接期间允许的最大请求数。设置为 0 允许无限量。

StartServers最初启动的服务器数量。

MinSpareServers、 MaxSpareServers用于服务器池大小调节。 Apache 不会让您猜测需要多少个服务器进程，而是动态适应它所看到的负载。也就是说，它会尝试维护足够的服务器进程来处理当前负载，以及一些备用服务器来处理瞬态负载峰值（例如，来自单个浏览器的多个同时请求）。它通过定期检查有多少服务器正在等待请求来做到这一点。如果少于 MinSpareServers，它会创建一个新的备用服务器。如果多于 MaxSpareServers，多余的备用将被杀死。

MaxClients限制运行的服务器总数，即限制可以同时连接的客户端数量。如果达到此限制，客户端将被锁定，因此不应设置得太低。它主要用作制动器，以防止失控的服务器在系统螺旋下降时带走系统。

---

#### 注意

在大多数 Red Hat 派生版本中，Apache 配置被分成两个子目录。主要配置文件 httpd.conf 位于 /etc/httpd/conf。Apache 模块的配置位于 /etc/httpd/conf.d 中。该目录中后缀为 .conf 的文件会在 Apache 启动期间添加到 Apache 配置中。

---

## Apache 虚拟主机

虚拟主机是一种提供在一台物理主机上托管多个域的能力的技术。有两种实现虚拟主机的方法：

\* 基于名称的虚拟主机 对于基于名称的虚拟主机，HTTP 服务器依赖客户端（例如浏览器）将主机名报告为 HTTP 请求标头的一部分。通过使用基于名称的虚拟主机，一个 IP 地址可以为不同 Web 域的多个网站提供服务。换句话说：基于名称的虚拟主机使用 URL 中的网站地址来确定要提供服务的正确虚拟主机。

\* 基于 IP 的虚拟主机 使用基于 IP 的虚拟主机，每个配置的 Web 域都至少有一个 IP 地址。

由于大多数主机系统都可以配置多个 IP 地址，因此一台主机可以为多个 Web 域提供服务。每个 Web 域都配置为使用特定的 IP 地址或 IP 地址范围。换句话说：基于 IP 的虚拟主机使用 TCP 连接的 IP 地址来确定要服务的正确虚拟主机。

### 基于名称的虚拟主机

基于名称的虚拟主机是一种相当简单的技术。您需要先配置您的 DNS 服务器以将每个域名映射到正确的 IP 地址。然后，配置 Apache HTTP 服务器以识别不同的域名并为相应的网站提供服务。

---

#### 提

示基于名称的虚拟主机缓解了对稀缺 IPv4 地址的需求。因此，您可以（或应该？）使用基于名称的虚拟主机，除非有特定原因选择基于 IP 的虚拟主机，请参阅 [基于 IP 的虚拟主机](#)。

---

要使用基于名称的虚拟主机，您必须在将接受主机请求的服务器上指定 IP 地址（可能还有端口）。在 Apache 2.x 到 2.4 上，这是使用 NameVirtualHost 指令配置的。自 Apache 2.4 以来，此 NameVirtual Host 指令已被弃用。每个 VirtualHost 还暗示一个 NameVirtualHost，因此从 Apache 2.4 开始定义一个 VirtualHost 就足够了。可以使用任何可用的 IP 地址。一方面应该在易于配置、使用和管理与另一方面安全性之间取得平衡。使用通配符作为 NameVirtualHost 或 VirtualHost 段内的侦听 IP 地址将在 Apache 主配置文件的 Listen 指令指定的所有 IP 地址上启用该特定配置的功能。如果主配置文件还对 Listen 选项使用通配符，这将导致 Apache HTTPD 服务器在服务器的所有已配置 IP 地址上可用。

因此,前面提到的功能在所有这些 IP 地址上的可用性也是如此。这是否可取或带来风险取决于具体情况。如果服务器使用多个网络接口和/或 IP 地址,则在配置服务时应特别小心。每个向网络公开服务的守护进程都可能包含基于代码或配置的错误。这些错误可能会被怀有恶意的人滥用。通过最小化所谓的服务器网络足迹,可用的攻击面也被最小化。防止通配符的额外配置开销是否值得付出努力,始终是一种权衡。

- Listen 可用于指定 Apache 侦听器应打开的 IP 地址和端口,以便为配置的内容。

<VirtualHost> 指令是为您想要服务的每个不同的 web 域创建的下一步。<VirtualHost> 指令的参数应该与 (Apache 2.4 之前的)NameVirtualHost 指令的参数相同 (即 IP 地址或 \* 表示所有地址)。在每个 <VirtualHost> 块中,您至少需要一个 ServerName 指令来指定为哪个主机提供服务,以及一个 DocumentRoot 指令来指出文件系统中可以找到该 web 域的内容的位置。

假设 www.domain.tld 和 www.otherdomain.tld 都指向 IP 地址 111.22.33.44。然后,您可以将以下内容添加到 httpd.conf 或等效 (包含的) 配置文件中:

```
名称虚拟主机 111.22.33.44  
  
<虚拟主机 111.22.33.44>  
服务器名称 www.domain.tld  
DocumentRoot /www/域  
</虚拟主机>  
  
<虚拟主机 111.22.33.44>  
服务器名称 www.otherdomain.tld  
DocumentRoot /www/otherdomain  
</虚拟主机>
```

IP 地址 111.22.44.33 可以替换为 \* 以匹配该服务器的所有 IP 地址。上面已经解决了以这种方式使用通配符的含义。

许多网站应该可以通过多个名称访问。例如, domain.tld 背后的组织想要促进 blog.domain.tld。有多种方法可以实现此功能,但其中一种使用 ServerAlias 指令。ServerAlias 指令在 <VirtualHost> 部分内声明。

例如,如果您将以下内容添加到上面的第一个 <VirtualHost> 块

```
ServerAlias domain.tld *.domain.tld
```

那么对 domain.tld 域中所有主机的请求将由 www.domain.tld 虚拟主机提供服务。通配符 \* 和 ? 可用于匹配名称。

#### 提示

当然,您不能只是编造名称并将它们放在 ServerName 或 ServerAlias 中。必须正确配置 DNS 系统以将这些名称映射到 NameVirtualHost 指令中声明的 IP 地址。

最后,您可以通过在 <VirtualHost> 容器中放置其他指令来微调虚拟主机的配置。大多数指令都可以放在这些容器中,然后只会更改相关虚拟主机的配置。

在主服务器上下文中设置的配置指令 (在任何 <VirtualHost> 容器之外) 只有在它们没有被虚拟主机设置覆盖时才会被使用。

现在当请求到达时,服务器将首先检查它是否正在请求与 NameVirtualHost 匹配的 IP 地址。如果是,那么它将查看每个具有匹配 IP 地址的 <VirtualHost> 部分,并尝试找到 ServerName 或 ServerAlias 与请求的主机名匹配的部分。如果找到一个,它就会使用该服务器的相应配置。

如果未找到匹配的虚拟主机,则将使用第一个列出的与 IP 地址匹配的虚拟主机。

因此,第一个列出的虚拟主机是默认虚拟主机。当 IP 地址与 NameVirtualHost 指令匹配时,将永远不会使用来自主服务器的 DocumentRoot。如果您希望对不匹配任何特定虚拟主机的请求进行特殊配置,请将该配置放在 <VirtualHost> 容器中,并将其放在 Apache 配置中任何其他 <VirtualHost> 容器规范之前。

### 基于 IP 的虚拟主机

尽管基于名称的虚拟主机具有优势,但出于某些原因您可能会考虑改用基于 IP 的虚拟主机。不过,这些都是利基场景:

- 一些较旧的或异国情调的 Web 客户端与基于名称的 HTTP 或 HTTPS 虚拟主机不兼容。基于 HTTPS 名称的虚拟主机是使用称为服务器名称指示器 (SNI) 的 TLS 协议扩展实现的。在撰写本文时,现代操作系统上的大多数现代浏览器都应该支持 SNI。
- 某些操作系统和网络设备实施带宽管理技术,除非它们位于不同的 IP 地址,否则无法区分主机。

正如术语“基于 IP”所表明的,服务器必须为每个基于 IP 的虚拟主机提供不同的 IP 地址。这可以通过为机器配备多个物理网络连接或使用虚拟接口来实现。大多数现代操作系统都支持虚拟接口(有关 IP 别名和 ifconfig 或 ip 命令的详细信息,请参阅系统文档)。

有两种运行 Apache HTTP 服务器以支持多主机的方法:

- 通过为每个主机名运行一个单独的 httpd 守护进程;
- 通过运行支持所有虚拟主机的单个守护程序。

在以下情况下使用多个守护进程:

- 存在安全问题,例如,如果您想为不同的客户保持网页之间的严格分离。在这种情况下,每个客户需要一个守护进程,每个守护进程运行不同的用户、组、监听和服务器根设置;
- 您可以承受在机器上侦听每个 IP 别名的内存和文件描述符要求。只能收听“通配符”地址或特定 IP 地址。因此,如果您需要将一个网络域限制为特定的 IP 地址,则所有其他网络域也需要配置为使用特定的 IP 地址。

在以下情况下使用单个守护进程:

- 在虚拟主机之间共享 httpd 配置是可以接受的。
- 机器处理大量请求,因此运行单独守护进程的性能损失可能很大。

### 设置多个守护进程

为每个虚拟主机创建一个单独的 httpd 安装。对于每个安装,使用配置文件中的 Listen 指令来选择守护进程服务的 IP 地址(或虚拟主机):

收听 123.45.67.89:80

Listen 指令可以定义为一个 IP:PORT 组合,由上面的冒号分隔。另一种选择是仅指定端口号。通过这样做,Apache 服务器将默认激活指定端口上所有已配置 IP 地址的侦听器:

听80  
听443

上面的 Listen 配置也可以使用 0.0.0.0 作为 IP 地址来定义,再次使用冒号作为分隔符。

Listen 指令的另一个选项启用协议的确切规范。在前面的示例中,使用了端口 80 和 443。默认情况下,在 Apache 中为 HTTP 配置端口 80,为 HTTPS 配置端口 443。此配置可以通过端口 8443 上的另一个 HTTPS 网站进行扩展:

```
听80  
听443  
收听 8443 https
```

配置一个或多个 Apache 守护程序时,可以使用 Listen 指令指定一个或多个 1024 以上的端口。

如果没有指定低于 1025 的其他端口,这将避免该守护程序需要 root 权限。除非配置中包含某些只能使用 root 权限访问的密钥或证书文件。您将在本书的下一页阅读更多相关内容。

从 Apache 2.4 开始,Listen 指令是强制性的,应该指定。如果未指定 Listen 指令,以前版本的 Apache 将默认认为所有可用 IP 地址上的 HTTP 端口 80 和 HTTPS 端口 443。从 Apache 2.4 开始,如果没有指定有效的 Listen 指令,Apache 服务器将无法启动。

### 设置单个守护进程

对于这种情况,单个 httpd 将为主服务器和所有虚拟主机的请求提供服务。配置文件中的 VirtualHost 指令用于将 ServerAdmin、ServerName、DocumentRoot、ErrorLog 和 TransferLog 或 CustomLog 配置指令的值设置为每个虚拟主机的不同值。

```
<虚拟主机 www.snow.nl>  
ServerAdmin webmaster@mail.snow.nl DocumentRoot /  
groups/snow/www ServerName www.snow.nl ErrorLog /  
groups/snow/logs/error_log TransferLog /groups/  
snow/logs/access_log </VirtualHost>
```

```
<虚拟主机 www.unix.nl>  
ServerAdmin webmaster@mail.unix.nl DocumentRoot /  
groups/unix_nl/www ServerName www.unix.nl  
  
错误日志 /groups/unix_nl/logs/error_log  
传输日志 /groups/unix_nl/logs/access_log  
</虚拟主机>
```

### 自定义文件访问

Redirect 允许您告诉客户端有关曾经存在于您服务器的命名空间中但现在不存在的文档。这使您可以告诉客户在哪里寻找重新定位的文档。

```
重定向 {old-URI} {new-URI}
```

### HTTPS (208.2) 的 Apache 配置

考生应该能够配置 Web 服务器以提供 HTTPS。

## 关键知识领域

### SSL 配置文件、工具和实用程序

能够为商业 CA 生成服务器私钥和 CSR

能够从私有 CA 生成自签名证书

能够安装密钥和证书

意识到虚拟主机的问题和 SSL 的使用

### SSL 使用中的安全问题

虚拟主机和通过服务器名称指示符 (SNI) 使用 SSL

禁用不安全的协议和密码

## 条款和实用程序：

- Apache2 配置文件
- /etc/ssl/, /etc/pki/
- openssl,CA.pl
- SSLEngine,SSLCertificateKeyFile,SSLCertificateFile
- SSLCACertificateFile,SSLCACertificatePath
- SSLProtocol,SSLCipherSuite,ServerTokens,ServerSignature,TraceEnable

资源：[LinuxRef08](#); [Engelschall00](#); [克劳斯01](#); [SSL01](#); [SSL02](#); [SSL03](#); [SSL04](#); [wikipedia\\_apache 模块](#); [mozsslconf](#); [raymii.org](#); [密码列表](#); [apachesslhowto](#); [维基犯罪](#); 各种命令的手册页。

## Apache2 配置文件

根据所使用的 Linux 发行版,当从软件包安装 Apache 时,以下文件和目录可用于 Apache 2.x 的配置:

```
httpd.conf  
apache.conf  
apache2.conf /etc/  
httpd/ /etc/httpd/  
conf /etc/httpd/conf.d /  
etc/apache2/
```

配置文件应该包含预定义的指令。如果未明确定义指令,Apache 将使用默认设置。此默认值可能因 Linux 发行版而异,因此请查阅您的发行版的 Apache 文档。 /usr/share/doc 是一个很好的起点。可以使用以下任一命令检查配置文件的语法错误:

```
$ sudo apachectl 配置测试 $ sudo 服务 httpd  
配置测试
```

因为 Apache 通常提供一个监听 1024 以下端口的守护进程,所以应该使用 sudo 或 root shell 来调用所有与 Apache 相关的命令。请参阅您的系统文档以检查 apachectl 或 apache2ctl 命令的可用性。

如果两者都存在,则它们可能是符号链接的。此命令的命名差异具有历史原因。 apachectl 用于 Apache 1.x,当 Apache2 发布时,该命令因此重命名为匹配新名称。现在 Apache2.x 已成为标准,或者 apache2ctl 已重命名为 apachectl,或者出于兼容性原因这两个命令都可用。如果可用,服务设施可能指向基于 Red Hat 的系统上的 httpd,或指向基于 Debian 的系统上的 apache2: apachectl 命令有许多有用的选项。它实际上是一个 shell 脚本,用作 httpd 二进制文件的包装器。请查阅手册页以获取所有可用的参数和选项。再举两个例子来帮助您入门:要显示所有已配置的虚拟主机,请使用:

```
$ sudo apachectl -t -D DUMP_VHOSTS
```

要显示所有当前正在运行的网站,请使用:

```
$ sudo apachectl -S
```

仔细解释上述命令的输出。该输出显示了当前正在运行的网站的配置。不能保证磁盘上的网站配置在这些网站上线后发生了变化。换句话说,运行进程的输出不必(不再)与配置文件的内容相匹配。

关于 Apache 配置文件,了解 Apache 的不同安装和配置方式很重要。根据所使用的 Linux 发行版和 Apache2.x 版本,配置文件在其他类似系统中的位置甚至命名可能不同。正如我们将在本章后面看到的,Apache 通常使用一个主配置文件。在此文件中,可以使用 INCLUDE /path/to/other/config 指令包含其他配置文件。可以通过调用 apachectl 脚本来检查配置文件语法是否有错误,如前所示。将检查正在使用的主配置文件中包含的每个配置文件的一致性和语法。这里的一致性意味着如果用户无法正确访问依赖的配置文件、证书文件或密钥文件,httpd 二进制文件运行时,将显示警告。如果 apachectl 没有出现在您的 \$PATH 中,请使用带有 apachectl 或 apache2ctl 作为参数的 sudo find 命令。根据存储卷的大小,将此搜索缩小到特定目录可能更明智。你被警告了。如果服务命令在您的系统上不可用,Apache 守护程序可能会由 SysV 脚本启动、检查和停止。在 /etc/init.d/ 目录中查找名为 httpd、apache2 或等效脚本的脚本。然后可以按如下方式调用此脚本,以显示可用参数:

```
$ 须藤/etc/init.d/apache2
```

## 加密网络服务器:SSL

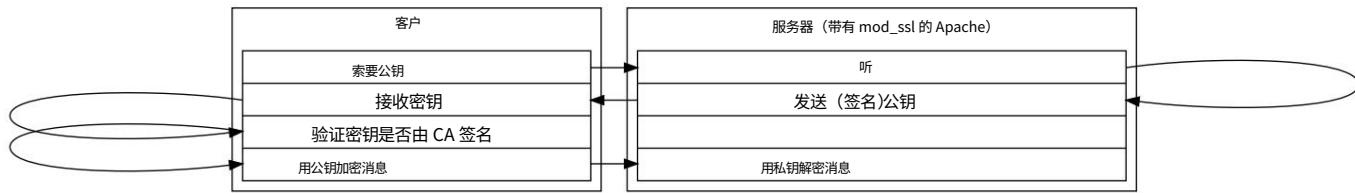
Apache 可以支持 SSL/TLS 以实现(相当)安全的在线通信。虽然 1.2 版中的 TLS 实际上是当前有利的选项,但 TLS 加密的 HTTPS 会话仍被称为“SSL”加密会话。TLS 实际上可以被视为 SSL (v3.0) 的继承者。因此,就像 Apache 与 Apache2 一样,本章中每当提到 Apache/SSL 时,也暗示了 TLS。除非另有规定。我们将在本章后面介绍这两种协议的优点和缺点。

安全套接字层协议(SSL)是可以置于可靠的面向连接的网络层协议(例如,TCP/IP)和应用层协议(例如,HTTP)之间的协议。SSL 通过允许相互身份验证和使用数字签名来确保完整性和加密来保护隐私,从而在客户端和服务器之间提供安全通信。目前有两个版本的 SSL 仍在使用:版本 2 和版本 3。此外,SSL 的后继版本 TLS (基于 SSL 的版本 1.0、1.1 和 1.2)是由 IETF 组织设计的。

### 公钥加密

SSL/TLS 使用公钥密码术(PKC),也称为非对称密码术。公钥加密用于发送方和接收方不共享公共秘密的情况,例如,在浏览器和网络服务器之间,但希望为他们的通信建立一个可信的通道。

PKC 定义了一种使用两个密钥的算法,每个密钥都可用于加密消息。如果一个密钥用于加密消息,则必须使用另一个密钥对其进行解密。这使得通过简单地发布一个密钥(公钥)并保密另一个密钥(私钥)来接收安全消息成为可能。任何人都可以使用公钥加密消息,但只有私钥的所有者才能读取它。例如,Alice 可以通过使用您的服务器发布的公钥加密消息来向密钥对的所有者(例如,您的 Web 服务器)发送私人消息。只有服务器才能使用相应的私钥对其进行解密。



安全的 Web 服务器（例如 Apache/SSL）使用基于 SSL/TLS 的 HTTP，默认使用端口 443。可以通过在主配置文件中定义 Listen 指令来配置 SSL/TLS 端口。应该已经为端口 80 (HTTP) 配置了一个侦听器。在基于 Debian 的系统上，有一个专门用于定义活动侦听器的文件。该文件称为 ports.conf，包含在主配置文件中。除此文件外，个别网站应在 ServerName 或 NameVirtualHost 声明的末尾指定侦听主机。从 Apache v2.4 开始，NameVirtualHost 已被弃用，取而代之的是 VirtualHost。这样的声明可能如下所示：<VirtualHost \*:443>。在浏览器中，HTTPS 的使用通过在 URL 中使用 https:// 方案来表示。公钥在服务器和客户端（浏览器）之间的通信建立期间交换。该公钥应由所谓的有效 CA（证书颁发机构）签名（它包含数字签名，例如消息摘要）。每个浏览器都包含许多所谓的根证书；这些证书可用于确定签署密钥的 CA 的有效性。并非每个证书都由这些有效 CA 之一签名。特别是出于测试目的，在没有有效 CA 干预的情况下签署证书是很常见的。这样做是为了节省（验证）时间和（注册费）金钱。截至 2015 年，维护有效的 CA 签名证书变得更加容易。只要您遵守规则，一个名为 Let's Encrypt 的组织愿意免费签署证书。阅读本章后，使用您最喜欢的网络搜索引擎查找有关 Let's Encrypt 的更多信息。

#### 带有 mod\_ssl 的 Apache

Apache 软件基金会提供了有关使用 mod\_ssl 的出色文档。我们敦促您花时间阅读通过以下 URL 收集的资源：<https://httpd.apache.org/docs/current/ssl/>

加密的主题是如此广泛和复杂，以至于围绕它写了整本书。只有在正确实施加密时，增加的机密性和完整性才能提供它们的价值。关于加密的所谓“最佳实践”可能会在一夜之间改变。除了上述 URL 列出的资源集合外，我们还想添加以下 URL：[http://httpd.apache.org/docs/trunk/ssl/ssl\\_howto.html](http://httpd.apache.org/docs/trunk/ssl/ssl_howto.html)

如您所见，此 URL 并未指向文档的当前版本。相反，它指向主干版本。

在撰写本文时，这对应于 Apache 2.5 文档。主干文档将始终指向开发中的最新 Apache 版本。虽然可能不建议使用主干 Apache 代码，但文档可能比其他地方更新得最近。关于 SSL/TLS 的主题，这导致了比 2.4 文档提供的更新的最佳实践。

Apache 软件基金会提供的文档是供应商中立的。因此，当 Apache 文档声明以下指令应出现在 Apache 主配置文件中时：

```
LoadModule mod_ssl module/mod_ssl.so 听 443
<虚拟主机>
</虚拟主机>
```

这些指令很可能是在多个配置文件中配置的。这取决于您的 Linux 发行版。除了 Apache 软件基金会提供的文档之外，我们还将尝试指出基于 Red Hat 和 Debian 的发行版之间的配置差异。

要使用 mod\_ssl，您需要安装 Apache 和 mod\_ssl 包。在基于 Red Hat 的系统上，这是使用以下命令完成的：

```
$ sudo yum 安装 httpd mod_ssl
```

在基于 Debian 的系统上，这是使用以下命令完成的：

```
$ sudo apt-get install apache2 openssl
```

安装后,确保在 Apache 中启用了 OpenSSL 模块。该模块应该可供 Apache 守护进程使用,并包含在守护进程启动期间加载。同样,有几种方法可以实现这一点。一种常见的方式类似于网站可用和网站启用策略。但是,现在我们正在处理模块可用和模块启用的目录。另外,基于 Debian 的系统带有一个名为 a2enmod 的实用程序。通过如下调用此命令:

```
$ sudo a2enmod 启用 ssl
```

a2enmod 将在 mods-enabled 目录中创建符号链接,分别指向 mods-available/ssl.conf 和 mods-available/ssl.load。重新加载 Apache 时,这些符号链接将确保 SSL 模块也将被加载。

基于 Red Hat 的系统改为使用 LoadModule 指令。应该声明该指令,以便在 Apache 守护程序启动期间读取它。在基于 Red Hat 的系统上,这可以通过包含以下 INCLUDE 指令的 /etc/httpd/conf/httpd.conf 来实现:

```
包含 conf.d/*conf
```

默认文件 /etc/httpd/conf.d/ssl.conf 然后可以包含以下 LoadModule 和 Listen 语句:

```
LoadModule ssl_module 模块/mod_ssl.so  
听443
```

重新加载 Apache 后,SSL 模块应该与 Apache 守护进程一起加载。在重新启动 Apache 守护程序之前检查配置错误始终是一个好习惯。这可以使用 apachectl configtest 命令完成,前面已经介绍过。输出应该清楚地解释 Apache 是否会遇到错误,以及为什么(会)。

然后,生成密钥和证书签名请求 (CSR)。要么自己签署 csr 文件,从而创建“自签名”证书,要么由有效的证书颁发机构 (CA) 签署。根据您的身份,自签证书在通过 HTTPS 呈现时可能会导致浏览器警告。让有效的 CA 签署 csr 可能会阻止这种情况的发生。

应该使用一些额外的指令来配置安全服务器 例如密钥文件的位置。记录所有这些指令超出了本书的范围。但是,您应该熟悉大多数 mod\_ssl 指令。

您可以通过搜索网络找到最佳实践,还应该参考您的发行版的特定 mod\_ssl 文档。

可以在 mod\_ssl [mod\\_ssl 网站](#) 上找到通用的 mod\_ssl 文档。

mod\_ssl 也可用于使用客户端证书对客户端进行身份验证。这些客户端证书可以由您自己的 CA 签名,mod\_ssl 将针对该 CA 验证证书。要启用此功能,请将 SSLVerifyClient 设置为 require。使用值 none 将其关闭。

作为 Linux 发行版的一部分安装的证书通常安装在基于 Debian 的系统上的 /etc/ssl/certs 中,以及基于 Red Hat 的系统上的 /etc/pki/tls/certs 中。基于 Red Hat 的系统可能有一个符号链接,将 /etc/ssl/certs 指向 /etc/pki/tls/certs 以方便和兼容。

作为 Linux 发行版的一部分安装的密钥或密钥文件通常依次安装在基于 Debian 的系统上的 /etc/ssl/private 和基于 Red Hat 的系统上的 /etc/pki/tls/private 中。/etc/ssl 和 /etc/pki 中的其他目录也可能包含特定的密钥文件。

在使用特定密钥和/或证书时,创建子目录通常被认为是最佳做法。特别是因为特定的加密密钥和证书属于彼此。通过为每个密钥对分配一个专用的子目录,结构将在文件系统和指向这些文件的配置文件中得到维护。这些子目录可以创建为 /etc/ssl 或 /etc/pki 层次结构的一部分。但是也可以在 /etc/apache2 或 /etc/httpd 下面创建子目录。

## 目录 /etc/ssl/

```
/etc/ssl$ ls -l 总共 32  
  
drwxr-xr-x 3 根根 16384 2011-03-06 15:31 证书  
-rw-r--r-- 1 根根 9374 2010-09-24 22:05 openssl.cnf drwxr-x--- 2 root ssl-cert  
4096 2011-03-06 13:19 私有
```

openssl 程序是 OpenSSL 加密库的命令行界面。您可以使用它来生成证书、加密和解密文件、创建哈希等等。它通常被视为密码学的“瑞士军刀”。一种更常见的用法是生成（自签名）证书以在安全的网络服务器上使用（以支持 https 协议）。/etc/ssl/openssl.cnf 是其配置文件的标准位置，您可以在其中设置组织名称、地址等的默认值。

---

#### 注意如

果您为网络服务器生成证书，您首先要创建证书签名请求 (.csr)。openssl 工具将提示您输入创建请求所需的信息，使用它从配置文件中获取的默认值。当您生成这样的签名请求时，请确保在 openssl 提示您输入“通用名称”或 CN（它是“专有名称”的一部分）时输入服务器的 FQDN（“完全限定域名”）。例如，当您为网站 https://www.foo.example/ 生成 CSR 时，输入 www.foo.example 作为 CN。请注意，提供 foo.example 的证书对于通过 https://www.foo.example 访问的网站无效。该证书对于 URL https://webmail.foo.example 后面的网站也无效。应为每个域设置单独的证书。为了应对这种必要性，许多组织选择使用通配符证书。特别是对于内部托管网站。当可用于任何 foo.example 网站的证书颁发 CSR 时，应针对 CN 值 \*.foo.example 完成请求。浏览器将在呈现时理解此通配符证书，并做出相应决定。

www.foo.example 和 webmail.foo.example 可以配置为使用此证书。另一方面，https://foo.example 将使用此证书发出浏览器警告。

---

## 如何创建 SSL 服务器证书

安装 OpenSSL 时，程序 openssl 会安装在您的系统上。此命令可用于创建实现（自签名）服务器证书的必要文件。

进一步来说：

- 首先生成 RSA 密钥文件。它包含一对相关的密钥，用于加密和解密发送给您的消息和来自您的消息。密钥对的一半将用于加密将使用公钥发送给您的消息。另一半用于使用私钥解密这些收到的消息。公钥将成为您的数字证书的一部分。这允许客户端系统将加密消息发送到您的网络服务器，只有该网络服务器可以解密，因为它拥有相关的私钥；
- 接下来您将创建证书签名请求 (CSR)。这是一个包含公钥和识别信息的文件  
公司名称、地址等；
- CSR 被发送到证书颁发机构 (CA)，该机构应验证您提供的信息的正确性并生成证书。此证书包含一个数字签名，可以验证 CA 是否已批准证书的内容。该证书将包含您提供的数据（包括您的公钥），并由 CA 使用其私钥签名。证书包含您的 RSA 公钥、您的姓名、CA 的名称并由您的 CA 进行数字签名。知道 CA 的浏览器可以验证该证书上的签名，从而获得您的 RSA 公钥。这使他们能够发送只有您可以解密的消息。

---

#### 笔记

您可以创建签名请求，然后自己签名。事实上，这就是证书颁发机构在创建根证书时所做的。根证书只是一个证书，表明他们说他们是他们所说的人。因此，任何人都可以创建一个根证书，并随心所欲地在上面放置任何凭据。根证书本身不能证明任何事情。您需要确保它确实是您信任的一方签发的。您要么访问他们并直接从他们那里获取副本，要么使用您信任的其他方法获取它，或者您依靠您信任的其他人为您完成此操作。您暗中“信任”大量 CA 的方法之一是依赖它们作为浏览器一部分的根证书。

---

举个例子：要创建一个密钥大小为 2048 位的 RSA 私钥，并且将进行三次 des (3DES) 加密，以默认格式（称为 PEM）存储在名为 server.key 的文件中，类型：

```
$ openssl genrsa -des3 -out server.key 2048
```

小于 1024 位的 RSA 密钥大小被认为是过时的。 1024 位似乎是当今的最佳实践,如果所有涉及的组件都能够处理这些密钥大小而不会超过阈值,则 2048、3072、4096 及以后是有效选项。

openssl 将要求一个密码短语,它将用作加密私钥的密钥。请将此文件存储在安全的备份位置并记住密码。如果您丢失了密码短语,您将无法恢复密钥。

出于测试目的,最好从密钥文件中删除密码。这可以通过读取密钥并将其导出来完成,如下所示:

```
$ openssl rsa -in server.key -out stripped.key
```

server.key 文件中仍然保存着加密后的密文私钥信息。 stripped.key 文件是一个纯文本文件,其内容是未加密的私钥信息。小心轻放。

要使用服务器 RSA 私钥创建证书签名请求 (CSR) (输出将采用 PEM 格式),请执行以下命令:

```
$ openssl req -new -key server.key -out server.csr
```

现在可以将签名请求发送到真实的 CA,它会签署请求并创建数字证书,或者您可以创建自己的 CA 并自己完成。请注意,如果您自己做,您还需要将您的 CA 的根证书安装到您的客户端 (例如浏览器) 中,以向他们表明由您自己的 CA 签署的证书是可信的。如果省略此步骤,您将收到很多关于失去信任和不安全感的令人不安的警告。

您可以自己提供 openssl 参数,但这对于经验不足的用户来说可能是一项艰巨的任务。因此,为了方便起见,OpenSSL 软件套件提供了一个 perl 脚本 (CA.pl) 来更轻松地处理大多数与 CA 相关的任务。它具有简化的语法,并为底层 openssl 命令提供更复杂的命令行参数。

CA.pl 将默认使用它从标准 OpenSSL 配置文件 /etc/ssl/openssl.cnf 中读取的值。要创建您自己的 CA,请找到应该包含在 OpenSSL 包中的 CA shellscript 或 CA.pl perlscript。在基于 Red Hat 的系统上,此脚本位于 /etc/pki/tls/misc 目录中。根据您的发行版,脚本可能不会解释参数的文件名。然后脚本会查找密钥文件和 csr 文件的预定义值。使用 less 或 more 之类的命令对脚本源进行分页并寻找线索。 STDERR 输出也可能显示一些有价值的指针。在以下示例中,newkey.pem 和 newreq.pem 被 CA.pl 脚本用作文件名:

```
# /usr/lib/ssl/misc/CA.pl -newca CA证书文件名 (或回车创建)
制作CA证书...
生成 2048 位 RSA 私钥
.....+++
.....+++
将新私钥写入 "./demoCA/private/cakey.pem" 输入 PEM 密码:***** 验证 - 输入 PEM 密码:*****
您将被询问输入将合并到您的证书请求中的信息。
```

您要输入的是所谓的专有名称或 DN。

有很多字段,但您可以保留一些空白 对于某些字段,将有一个默认值,如果您输入 “.” ,该字段将保留为空白。

```
-----  
国家名称 (2 个字母代码)[NL]:  
州或省名称 (全名)[无]:  
地点名称 (例如城市)[]:组织名称 (例如公司)[Snow  
BV]组织单位名称 (例如部分)[]:通用名称 (例如服务器 FQDN 或您的名称)  
[:ssltest.snow.荷兰电子邮件地址 [:]
```

请输入以下“额外”属性,与您的证书请求一起发送 质询密码 []:可选的公司名称 []:使用来自 /usr/lib/ssl/openssl.cnf 的配置 输入 ./demoCA/private 的密码/cakey.pem:检查请求是否匹配签名 Signature ok Certificate Details:

序列号:  
ca:d8:22:43:94:6d:ca:6c

有效性  
不早于:格林威治标准时间 2013 年 7 月 9 日 13:49:38  
不晚于:格林威治标准时间 2016 年 7 月 8 日 13:49:38

主题:  
countryName = 荷兰  
stateOrProvinceName = 无  
机构名称 = 雪 BV  
常用名 = ssltest.snow.nl

X509v3 扩展:  
X509v3 主题密钥标识符: 83:F3:99:4B:98:E0:F1:37:78:67:DC:04:AC:04:65:03:48:BB:31:FB  
X509v3 权限密钥标识符: keyid:83:F3:99:4B:98:E0:F1:37:78:67:DC:04:AC:04:65:03:48:BB:31:FB

X509v3 基本约束:  
加州:真  
证书将被认证到 Jul 8 13:49:38 2016 GMT (1095 天)

用 1 个新条目写出数据库  
数据库已更新

## 接下来创建一个签名请求:

```
# /usr/lib/ssl/misc/CA.pl -newreq 生成 2048 位 RSA 私钥
```

```
.....+++  
.....+++
```

将新私钥写入“newkey.pem” 输入 PEM 密码:验证 - 输入 PEM 密码:

-----

您将被要求输入将合并到您的证书请求中的信息。

您要输入的是所谓的专有名称或 DN。  
有很多字段,但您可以保留一些空白 对于某些字段,将有一个默认值,如果您输入“.”,该字段将保留为空白。

-----

国家名称 (2 个字母代码)[NL]:州或省名称 (全名)[]:无地点名称 (例如城市)[]:  
组织名称 (例如公司)[]:Snow BV

组织单位名称 (例如,部分)[]:通用名称 (例如服务器 FQDN 或您的名称)[]:ssltest.snow.nl 电子  
邮件地址 []:

请输入以下“额外”属性,以与您的证书请求一起发送 质询密码 []:可选公司名称 []:请求在 newreq.pem 中,私钥在 newkey.pem 中

然后,我们签署请求:

```
# /usr/lib/ssl/misc/CA.pl -signreq 使用来自 /usr/lib/ssl/openssl.cnf 的配  
置 输入 ./demoCA/private/cakey.pem 的密码短语 :检查请求是否与签名 Signature 匹配好的证书详细信息:
```

```
序列号:  
ca:d8:22:43:94:6d:ca:6d  
有效性  
不早于:格林威治标准时间 2013 年 7 月 9 日 13:53:53  
不晚于:格林威治标准时间 2014 年 7 月 9 日 13:53:53  
主题:  
countryName 州名或省 =荷兰  
名 =无  
机构名称 = 雪 BV  
常用名 = ssltest.snow.nl  
X509v3 扩展:  
X509v3 基本约束:  
加州:假  
网景评论:  
OpenSSL 生成的证书  
X509v3 主题密钥标识符:  
21:A4:61:83:B4:E7:C3:E9:2B:2C:0A:DD:36:FA:82:D0:77:3A:E2:01  
X509v3 权限密钥标识符:  
keyid:83:F3:99:4B:98:E0:F1:37:78:67:DC:04:AC:04:65:03:48:BB:31:FB
```

证书将被认证到 Jul 9 13:53:53 2014 GMT (365 天)

签署证书? [是/否]:是的

1 个证书请求中有 1 个已认证,提交? [y/n]y 用 1 个新条目写出数据库

数据库更新签名证书在 newcert.pem  
中

您现在创建了一个由您自己的 CA (newcert.pem) 签名的证书。您可能希望将文件重命名为更易于区分的名称,例如 Certificate:ssltest.snow.nl。同时,也重命名服务器密钥文件,例如 PrivateKey:ssltest.snow.nl。特别是如果您在许多服务器上维护大量密钥和证书,那么能够从文件名中了解其中的内容真的很有帮助。

证书签名请求 (CSR) 可能已发送到外部证书颁发机构 (CA)。您通常必须将 CSR 发布到 Web 表单中,支付签名费用并等待签名证书。有一些非盈利的 CA 可以免费执行类似的任务,例如 CACert。但是,他们的根证书尚未包含在大多数浏览器中,因此如果您要使用他们的服务,您将需要自己这样做。

不再需要 server.csr 文件。现在您有两个文件:server.key 和 newcert.pem。在 Apache 的 httpd.conf 文件中,您应该使用如下行来引用它们:

```
SSL证书文件 /path/to/Certificate:ssltest.snow.nl SSLCertificateKeyFile /  
path/to/PrivateKey:ssltest.snow.nl
```

在管理密钥和证书文件时,遵循“最小特权”原则被认为是最佳实践。这些文件最好以只有运行 Web 服务器的用户帐户才能访问的方式存储。

## Apache SSL 指令

您应该熟悉以下 Apache SSL 配置指令:

**SSLEngine**该指令切换 SSL/TLS 协议引擎的使用。这应该在 <VirtualHost> 部分中使用,以便为该虚拟主机启用 SSL/TLS。默认情况下,主服务器和所有配置的虚拟主机都禁用 SSL/TLS 协议引擎。

**SSLCertificateKeyFile**该指令指向服务器的 PEM 编码私钥文件。如果包含的私钥已加密,则在启动时强制使用密码短语对话框。当并行使用 RSA、DSA 和基于 ECC 的私钥时,此指令最多可使用三次 (引用不同的文件名)。对于每个 SSLCertificateKey File 指令,必须有一个匹配的 SSLCertificateFile 指令。

**SSLCertificateFile**该指令指向一个包含 PEM 格式证书数据的文件。该文件至少必须包含一个终端实体 (叶)证书。当并行使用基于 RSA、DSA 和 ECC 的服务器证书时,该指令最多可使用三次 (引用不同的文件名)。

## 为 Apache 创建和安装自签名证书

有时,使用 Apache 的自签名 SSL 证书可能是可以接受的。以下步骤解释了如何在基于 Debian 的系统上完成此操作。首先,创建一个目录来保存 SSL 密钥。在我们用作示例的系统上,所有系统范围的 SSL 证书都存储在目录 /etc/ssl/certs 中。为了我们的目的,我们创建一个名为 /etc/ssl/webserver 的新目录,并用它来存储我们的新密钥对:

```
# mkdir /etc/ssl/webserver # openssl req -new
-x509 -days 365 -nodes \ >-out /etc/ssl/webserver/apache.pem -keyout /etc/ssl/
webserver/apache.key 生成一个 2048 位RSA私钥

.....+++  
....+++
将新私钥写入 “/etc/ssl/webserver/apache.key”# ls /etc/ssl/webserver/  
  
apache.key apache.pem
```

### 笔记

在创建过程中,openssl 会使用 /etc/ssl/openssl.cnf 的内容来填充一些变量。交互式脚本将询问其他值。请务必在此处使用正确的 FQDN,以便稍后将此证书与具有其他用途的证书区分开来。

为了能够在 Apache 中使用 SSL,必须加载一个名为 mod\_ssl 的模块。在这个系统上,我们可以通过列出 /etc/apache2/mods-enabled 目录的内容来检查启用的模块。可以通过列出 /etc/apache2/mods-available 目录的内容来检查所有当前可用的模块:

```
# ls /etc/apache2/mods-enabled/
别名.conf          自动索引.conf mime.conf           reqtimeout.load setenvif.conf
别名.load          自动索引.load mime.load
auth_basic.load authn_file.load cgi.load             negotiation.conf setenvif.load negotiation.load status.conf
authz_default.load authz_groupfile.load dir.conf authz_host.load dir.load deflate.conf文件 perl.load php5.conf php5.load reqtimeout.conf
authz_user.load    环境负载                         放气.load          状态.load
                                                              
# ls /etc/apache2/mods-available/ actions.conf cgid.conf
cgid.load cgi.load charset_lite.load ldap.conf dav.conf      包含.load          proxy_ftp.conf
动作.load          log_forensic.load               信息.conf          proxy_ftp.load
别名.conf          dav.load                      信息加载          proxy_http.load proxy.load
别名.load          mem_cache.conf                reqtimeout.conf   proxy_scgi.load
asis.load
auth_basic.load
auth_digest.load
```

authn_alias.load	dav_lock.load dbd.load	mem_cache.load	重写.load
authn_anon.load	deflate.conf文件	mime.conf	setenvif 配置文件
authn_dbd.load authn_dbm.load	放气.load	mime.load	setenvif.load
authn_default.load	目录配置文件	mime_magic.conf	speling.load ssl.conf
authn_file.load	目录加载	mime_magic.load mod-dnssd.conf	ssl.load
authnz_ldap.load	disk_cache.conf	mod-dnssd.load	状态文件
authz_dbm.load	磁盘缓存.load	negotiation.conf negotiation.load	状态.load
authz_default.load	dump_io.load	perl.load php5.conf php5.load	替代.load
authz_groupfile.load env.load authz_host.load expires.load		proxy_ajp.load	suexec.load
ext_filter.load file_cache.load filter.load		proxy_balancer.conf	unique_id.load userdir.conf
authz_owner.load		usertrack.load	用户目录.load
authz_user.load autoindex.conf		proxy_balancer.load	vhost_alias.load proxy.conf proxy_connect.load
自动索引.load	headers.load		
缓存.load	识别.load		
cern_meta.load	图像映射.load		

ssl 似乎可用但尚未启用,因为 ssl 文件 ssl.load 和 ssl.conf 仍然存在于 /etc/apache2/mods-available/ 目录中,而不是 /etc/apache2/mods-启用/目录。我们可以自己创建一个符号链接来激活对 ssl 的支持,但是 Debian 提供了一个用 perl 编写的名为 a2enmod 的实用程序来处理这个问题。有关详细信息,请参阅 A2ENMOD(8) 联机帮助页。它的对应物,通常称为 a2dismod,做相反的事情并通过从 /etc/apache2/mods-enabled/ 中删除符号链接来禁用 Apache 模块。

让我们启用 SSL:

```
#a2enmod ssl
启用模块 ssl。
请参阅 /usr/share/doc/apache2.2-common/README.Debian.gz 了解如何配置 SSL 和创建自签名证书。

要激活新配置,您需要运行:service apache2 restart # service apache2 restart [ok] Restarting web
server: apache2 ... waiting. # apachectl status |grep -i ssl 服务器版本:Apache/2.2.22 (Debian) PHP/
5.4.4-15.1 mod_ssl/2.2.22 OpenSSL/
```

现在已在 Apache HTTP 服务器上启用 SSL。为了让站点真正使用 SSL,必须正确配置它的配置。HTTPS 默认使用 tcp 端口 443,所以我们要在 Debian 的 apache 配置中指定这个。将以下行添加到您的 /etc/apache2/ports.conf 文件中:

听443

现在,所有想要使用 SSL 的站点都需要重新配置它们的配置文件。需要将以下行添加到每个应通过 HTTPS 提供其内容的“已启用”站点:

```
SSLEngine 上的
SSLCertificateFile /etc/ssl/webserver/apache.pem SSLCertificateKeyFile /etc/ssl/webserver/
apache.key
```

启用 HTTP 和 HTTPS 的站点的示例站点配置文件可能如下所示:

```
名称虚拟主机 *:80
名称虚拟主机 *:443

<虚拟主机 *:80>
服务器名 webserver.intranet
DocumentRoot /srv/http ErrorLog /var/log/
apache2/error.log </VirtualHost>

<虚拟主机 *:443>
```

```
SSLEngine On  
SSLCertificateFile /etc/ssl/webserver/apache.pem SSLCertificateKeyFile /etc/  
ssl/webserver/apache.key Servername webserver.intranet DocumentRoot /srv/  
http ErrorLog /var/log/apache2/error.log </VirtualHost>
```

现在,使用 `apachectl configtest` 来测试您的站点配置,如果没有出现错误,请重新启动 Apache HTTP 服务器。现在应该可以使用 https URL 而不是 http 访问启用 SSL 的站点。

## 其他 Apache 指令

除了上面使用的指令外,您应该熟悉以下 Apache 配置指令:

**SSLCACertificateFile** 该指令设置一体化文件,您可以在其中组装与您打交道的客户的认证机构 (CA) 的证书。这些用于客户端身份验证。这样的文件只是按优先顺序排列的各种 PEM 编码证书文件的串联。

**SSLCACertificatePath** 设置保存证书颁发机构 (CA) 证书的目录,其客户端处理。这些用于在客户端身份验证上验证客户端证书。

**SSLCipherSuite** 该复杂指令使用由 OpenSSL 密码规范组成的以冒号分隔的密码规范字符串来配置允许客户端在 SSL 握手阶段协商的密码套件。请注意,此指令既可以在每个服务器上下文中使用,也可以在每个目录上下文中使用。在每个服务器上下文中,它适用于建立连接时的标准 SSL 握手。在每个目录上下文中,它会在读取 HTTP 请求之后但在发送 HTTP 响应之前强制与重新配置的密码套件进行 SSL 重新协商。

**SSLProtocol** 该指令可用于控制 mod\_ssl 在建立其服务器环境时应使用的 SSL 协议风格。然后客户端只能使用提供的协议之一进行连接。

**ServerSignature** **ServerSignature** 指令允许在服务器生成的文档 (错误消息、mod\_proxy ftp 目录列表、mod\_info 输出...) 下配置尾随页脚行。您想要启用这样一个页脚行的原因是,在代理链中,用户通常无法分辨哪个链式服务器实际产生了返回的错误消息。

**ServerTokens** 该指令控制发送回客户端的服务器响应标头字段是否包含最少的信息、所有值得一提的信息或介于两者之间的信息。默认情况下,ServerTokens 指令设置为 Full。通过声明此 (全局) 指令并将其设置为 Prod,所提供的信息将减少到最低限度。在本主题的第一章中提到了从源代码编译 Apache 的必要性。修改 Apache 服务器响应标头字段值可能是需要修改源代码的场景。这很可能是服务器强化过程的一部分。因此,Apache 服务器可以提供不同的值作为响应标头字段。

**TraceEnable** 该指令覆盖核心服务器和 mod\_proxy 的 TRACE 行为。根据 RFC 2616,默认的 TraceEnable on 允许 TRACE 请求,这不允许任何请求主体伴随请求。

TraceEnable 关闭导致核心服务器和 mod\_proxy 向客户端返回 405 (方法不允许) 错误。

还有一个扩展的不合规设置,它将允许消息正文伴随跟踪请求。

此设置应仅用于调试目的。尽管安全扫描可能会说什么,但 TRACE 方法是 HTTP/1.1 RFC 2616 规范的一部分,因此不应在没有特定原因的情况下被禁用。

## SSL 与 Apache 虚拟主机

正如我们之前看到的,FQDN 在 SSL 中起着重要作用。它必须与证书的 CN 值匹配。当启动与 HTTPS 服务器的连接时,此证书将提供给浏览器。只有当证书有效、由已知的、受信任的注册方颁发并且与主机名匹配时,浏览器才会在没有警告的情况下启动连接。否则,浏览器应该显示有关无效或过期证书、未知颁发者或无效主机名的警告。

对于基于 IP 的虚拟主机,我们为每个虚拟主机提供不同的 IP/端口组合,这意味着我们可以为每个虚拟主机配置 SSL 证书。毕竟 HTTPS 连接将在专用 IP 地址上启动。

然而,当使用基于名称的虚拟主机时,除了主机名之外,我们没有被请求资源的唯一标识符。因此,Apache HTTP 服务器接收对它在同一 IP/端口组合上服务的虚拟主机的所有请求。

直到建立了 SSL 连接后,HTTP 服务器才知道根据 URL 实际请求的是哪个虚拟主机。该 URL 公开了虚拟主机的主机名。但是,到那时可能为时已晚;由于 SSL/TLS 事务中 HTTP 的性质,客户端可能已获得具有不同 CN 值的证书。

目前,可以使用名为 SNI (服务器名称指示)的扩展来规避这种基于名称的问题。使用此扩展,浏览器将请求的主机名包含在其 SSL 握手的第一条消息中,作为 UTF-8 编码的字节字符串值,表示客户端问候消息的 server\_name 属性。此值应仅包含主机名和/或域名。不应使用 IPv4 或 IPv6 IP 地址。浏览器和 Apache 都需要支持 SNI 才能使 SNI 机制发挥作用。如果在服务器上使用 SNI 而浏览器不支持 SNI,则浏览器可能会显示“不受信任的证书”警告。这取决于为 HTTP 服务器的“默认”网站提供的证书。在撰写本文时,大多数浏览器都支持 SNI。例外情况是 Android 2.x 默认浏览器、MS Windows XP SP3 之前的 MS Internet Explorer 和任何操作系统上 1.7 之前的 Oracle Java 版本。

要在 Apache 服务器上使用 SNI 并防止由于不支持 SNI 的浏览器而出现“不受信任的证书”警告,可以使用多域证书。该证书应包含所有必要的域名,并应在单独的虚拟主机中的 Apache 配置中使用。在此虚拟主机中,不应配置任何服务器名,因此它将匹配所有没有主机名的请求,因此为所有不支持 SNI 的浏览器提供服务。由于从 URL 中提取请求的网站,Apache 将显示正确请求(SNI)站点的内容。此提取发生在使用多域证书创建加密会话之后。这个解决方案可能永远不会因为它的外观而获奖,但它背后的科学至少是有效的。

如果没有 SNI,Web 服务器可以通过 HTTPS 为多个基于名称的虚拟主机提供服务,而不会出现浏览器警告。要求(或限制)所有虚拟主机使用的 SSL 证书都相同。虚拟主机也必须属于同一域,例如 virtual01.example.com 和 virtual02.example.com。必须以 CN(通用名称)指向所用域的通配符的方式配置 SSL 证书,例如 \*.example.

com。

## SSL 安全问题

SSL 的加密安全方面完全基于信任。2013 年年中,大约有 650 个证书颁发机构。

这些机构中的每一个都可能成为“最薄弱的环节”,因此您的 SSL 证书的安全性取决于您对其 CA 的信任程度。

由于;原因(2.4.8),SSLCertificateChainFile 指令已从 LPIC-2 目标中删除。尽管如此,这个指令还是值得了解的。它是对 SSLCertificateFile 和 SSLCertificateKeyFile 指令的补充。前面已经指出了自签名证书对 HTTPS 会话的影响。但是,即使您将 CSR 发送到已知的 CA 并收到经过验证的证书作为回报,使用此证书和用于生成 CSR 的正确密钥文件正确设置您的服务器,这并不意味着每个浏览器都会使用此验证会话证书有效。这通常是因为所涉及的 CA 使用您的浏览器不知道的证书对 CSR 进行了签名。例如,如果签名证书比您的浏览器更新,则可能会出现这种情况。还因为许多 CA 从其产品组合中提供不同类型的证书。浏览器根据所谓的根证书的可用性来识别 HTTPS 证书的有效性。这些可以被视为证书链中的顶级。另一方面,CSR 通常使用所谓的中间证书进行签名。这些中间证书与根证书相关,但存在于证书链的较低级别。为了限制浏览器软件随附的证书数量,默认情况下并非所有用于签署 CSR 的证书都包括在内。这可能会导致浏览器警告证书链不完整。

要补救这些警告(实际上是错误),可能需要一个或多个中间证书来重新组装证书链以确保完整性。为了促进这一点,使用中间证书的 CA 通常会提供这些中间证书供下载。CA 的网站和/或持有签名证书信息的电子邮件应指向适当的中间证书。不同级别通常被称为 Gn 级证书,其中 n 代表一定的数字。

这些中间证书填补了您的签名证书与所有主要浏览器已知的根证书之间的空白。

通过使用 SSLCertificateChainFile 指令,您可以将 Apache 指向一个包含两个或多个串联证书的文件。通过以正确的顺序连接丢失的证书,证书链间隙将被关闭,证书链将再次完整。

## 禁用不安全的协议和密码

当 Apache 配置为安全服务器以通过 HTTPS 协议提供内容时,客户端和服务器会在加密启动时协商哪些密码可用于保护连接。Apache 服务器和浏览器都会向对方提供可用的加密算法列表。根据强制执行的设置,服务器和浏览器然后使用可用密码列表中的重叠加密算法启动安全通道。通过将 SSLHonorCipherOrder 指令设置为值 on,服务器将遵循 SSLCipherSuite 指令指定的密码顺序。否则,客户端在决定使用哪些密码时将占上风。这个安全通道然后用于传输加密密钥,这些密钥将用于保护从这里开始的通信。这些加密密钥还必须采用服务器和客户端都可以处理的密码格式。通常情况下,必须在兼容性和安全性之间做出权衡。最大化提供的密码数量(以及浏览器兼容性)也增加了使用中的一种或多种密码可能易受攻击向量影响的可能性。“弱密码”一词通常用于指代这些易受攻击的加密算法。(不)推荐的密码列表在很大程度上取决于滥用这些密码的已知漏洞的发布。这导致意见可能会随着时间而改变。因此,建议及时了解已知漏洞。目前,不推荐使用所谓的密码块代码密码。这些密码可以通过其名称中的“CBC”部分来识别。Arcfour(或简称 RC4)也不是推荐使用的密码。

就协议而言,已知 SSL v2 和 SSL v3 容易受到大量攻击媒介的攻击。TLS v1.0 和 v1.1 也有其弱点。如果担心安全问题,目前推荐使用 TLS v1.2 协议。随着 TLS v1.3 即将出现。Apache 允许配置提供给客户端的密码。通过使用以下指令,可以限制 Apache 提供给客户端软件的密码列表:

SSL密码套件  
SSL协议

以下部分显示了有关如何使用这些指令的示例配置。SSLCipherSuite 配置为仅使用强密码套件,同时明确禁用 RC4。SSLProtocol 指令禁用对所有协议的支持,同时显式启用 TLSv1.2 支持。服务器和客户端应评估密码以相互支持的顺序由服务器通过使用 SSLHonorCipherOrder 指令确定。最后,SSLCompression 指令被配置为禁用 SSL/TLS 压缩。禁用此压缩可减轻已知的 CRIME(压缩比信息泄漏变得容易)攻击向量。

```
SSLCipherSuite ECDH+AESGCM:EDH+AESGCM:AES256+ECDH: AES256+EDH:!RC4 SSLProtocol -所有+TLSv1.2  
SSLHonorCipherOrder On  
SSLCompression 关闭
```

从 Apache 2.4 开始,mod\_ssl 放弃了对 SSLv2 的支持。仅支持 SSLv3 和 TLS。请注意,对最新 TLS 版本的支持取决于最新 OpenSSL 库的可用性。

Github 上的 Mozilla TLS Server 项目是配置安全 Apache 服务器之前的一个很好的参考。这个网站有各种主要 HTTP 服务器的示例配置字符串,包括 Apache。项目团队成员应根据最新建议使示例配置保持最新。另一个很好的参考是<https://cipherli.st>。该网站还提供示例配置。

对于每个示例配置,在未验证内容的情况下不要复制/粘贴配置始终很重要。如前所述,在大多数情况下必须进行权衡。配置 Apache 以严格服务于像 TLS 1.2 这样的现代密码将减轻与 SSL 和 TLS 连接有关的大多数已知攻击向量。但并非每个操作系统上的每个浏览器都能够遵守此要求。在客户端软件中采用 TLS v1.2 需要最新的 SSL 库可用。并非所有的软件供应商都在跟踪这些库。因此,这里的权衡是安全性有利于兼容性。

使用 SSL v2 和 SSL v3 等较旧的密码可能会增加加密连接易受已知攻击的可能性。但与此同时,这将最大限度地提高能够建立这些连接的客户端的兼容性。

另一个权衡。

除了明确指定可以使用哪些协议和密码外,使用特定协议或密码的偏好可能会有所不同。指定指令的顺序有影响,但不提供任何保证。服务器和客户端通常使用一种称为机会加密的技术来决定将使用哪些协议和密码。同时,客户端软件可以准确指定应使用的协议和密码。根据服务器配置,服务器将尊重客户端的这些要求。正是这种功能是称为降级攻击的已知攻击类别的基础。

设置服务器后,最好定期扫描系统以查找已知漏洞。这可以通过多种方式完成。一种简单的方法是使用公共 Web 服务,例如 Qualys SSLabs: <https://ssllabs.com>。输出将详细显示是否以及如果是,则检测到哪些弱协议和/或密码。

## 将 Squid 实现为缓存代理 (208.3)

考生应该能够安装和配置代理服务器,包括访问策略、身份验证和资源使用。

### 关键知识领域

Squid 3.x 配置文件、术语和实用程序

访问限制方法

客户端用户认证方式

Squid 配置文件中 ACL 的布局和内容

以下是所用文件、术语和实用程序的部分列表:

· squid.conf

· 访问控制列表

· http\_access

资源: [Kiracofe01](#); [布罗克迈尔01](#); [威塞尔01](#); [培生00](#); 各种命令的手册页。

### 网络缓存

Web 缓存,也称为 http 代理,用于减少带宽需求,并且通常允许更细粒度的访问控制。使用代理,在客户端软件中必须指定代理的主机名和端口号。当浏览器尝试连接到 Web 服务器时,请求将发送到指定的代理服务器,但对用户来说,请求似乎已直接发送到请求的 Web 服务器。代理服务器现在与指定的 Web 服务器建立连接,等待应答并将其发送回客户端。代理像解释器一样工作:客户端与代理交谈并收听代理,代理与客户想要交谈的 Web 服务器交谈并收听。代理还将使用本地缓存的网页版本(如果它们尚未过期)并且还将验证客户端请求。

此外,还有透明代理。通常这是常规代理和重定向路由器的串联。在这些情况下,代理可以透明地拦截 Web 请求。在这种情况下,无需在客户端软件的设置中设置代理。就客户端软件所知,它直接与目标服务器对话,而实际上是与代理对话。

### 乌贼

squid 是用于 Web 客户端的高性能代理缓存服务器。squid 不仅支持 HTTP 数据对象:它还支持 FTP 和 gopher 对象。squid 在一个单一的、非阻塞的、I/O 驱动的进程中处理所有请求。squid 保留元数据,特别是缓存在 RAM 中的热对象,它缓存 DNS 查找,支持非阻塞 DNS 查找并实现失败请求的负缓存。squid 还支持 SSL、广泛的访问控制和完整的请求日志记录。通过使用轻量级 Internet 缓存协议,可以将 squid 缓存安排在层次结构或网格中,以进一步节省带宽。

squid 可用于多种用途,包括节省带宽、处理流量高峰和缓存偶尔不可用的站点。squid 也可以用于负载均衡。本质上,squid 第一次从浏览器收到请求时,它充当中介并将请求传递给服务器。squid 然后保存该对象的副本。如果没有其他客户请求相同的对象,则不会获得任何好处。但是,如果多个客户端在对象从缓存中过期之前请求该对象,squid

可以加快交易速度并节省带宽。如果您曾经需要来自慢速站点的文档,比如位于另一个国家或托管在慢速连接上的文档,或者两者兼而有之,您会注意到缓存文档的好处。第一个请求可能比 molasses 慢,但下一个对同一个文档的请求会快得多,并且原始服务器的负载会减轻。

squid 由一个主服务器程序squid、一个域名系统查找程序dnsserver、一些用于重写请求和执行身份验证的可选程序以及一些管理和客户端工具组成。当 squid 启动时,它会生成数量可配置的 dnsserver 进程,每个进程都可以执行单个阻塞域名系统 (DNS) 查找。

这减少了缓存等待 DNS 查找的时间。

squid 通常以源代码格式获得。在大多数系统上,一个简单的 make install 就足够了。之后,您还将拥有一组配置文件。在大多数发行版中,所有 squid 配置文件默认保存在目录 /usr/local/squid/etc 中。但是,位置可能会有所不同,具体取决于您的发行版的风格和习惯。例如,Debian 软件包将配置文件放在 /etc 中,这是 .conf 文件的正常主目录。

虽然这个目录中有多个文件,但只有一个文件对大多数管理员来说是重要的,即 squid.conf 文件。这个文件中只有大约 125 个选项标签,但实际上只需要八个选项就可以启动和运行 squid。其他选项只是给你额外的灵活性。

如果在 squid.conf 文件中没有出现标记,squid 假定您希望使用默认值。理论上,您甚至可以使用零长度配置文件运行 squid。但是,您至少需要更改配置文件的一部分,即默认的 squid.conf 拒绝所有浏览器的访问。您将需要编辑访问控制列表以允许您的客户端使用 squid 代理。执行访问控制的最基本方法是使用 http\_access 选项 (见下文)。

#### SQUID.CONF文件中的部分

http\_port这个选项决定了 squid 将在哪个端口上监听请求。默认情况下,这是端口 3128。另一个常用端口是端口 8080。

cache\_dir用于配置具体的存储区域。如果为缓存数据使用多个磁盘,则可能需要多个挂载点 (例如, /usr/local/squid/cache1 用于第一个磁盘, /usr/local/squid/cache2 用于第二个磁盘)。 squid 允许您在配置文件中有多个 cache\_dir 选项。该选项可以有四个参数:

```
cache_dir /usr/local/squid/cache/ 100 16 256
```

第一个选项确定应在哪个目录中维护缓存。下一个选项是以兆字节为单位的大小值,默认值为 100 兆字节。 squid 将在指定目录中存储最多该数量的数据。接下来的两个选项将设置要在此目录中创建的子目录 (第一层和第二层) 的数量。 squid 创建大量目录并在每个目录中仅存储几个文件以尝试加快磁盘访问速度 (在包含一百万个文件的目录中找到正确的条目效率不高:最好将文件分开分成许多较小的文件集)。

http\_access, acl选项的基本语法是http\_access allow|deny [!]aclname。如果您想提供对内部网络的访问,并拒绝其他任何人访问,您的选项可能如下所示:

```
acl home src 10.0.0.0/255.0.0.0 http_access allow  
home
```

第一行设置了一个名为 “home”的访问控制列表类,属于内部网络 IP 地址范围。第二行允许访问该范围的 ip 地址。假设它是访问列表中的最后一行,所有其他客户端都将被拒绝。

另请参阅有关[acl 的部分](#)。

#### 提示

请注意,如果squid找不到匹配的条目,它的默认行为是执行与上次访问行相反的操作。例如,如果最后一行设置为“允许”访问一组特定的网络地址,则squid将拒绝任何不符合其任何规则的客户端。另一方面,如果最后一行设置为“拒绝”访问,则squid将允许访问任何不符合其规则的客户端。

auth\_param此选项用于指定启动哪个程序作为验证器。您可以指定程序的名称和所需的任何参数。

redirect\_program, redirect\_children redirect\_program 用于指定启动哪个程序作为重定向器。选项 redirect\_children 用于指定启动多少个进程来进行重定向。

在对配置进行更改后,发出 squid -k reconfigure 以便 squid 将使用更改。

### 重定向器

squid 可以配置为通过重定向器进程传递每个传入的 URL,该进程返回一个新的 URL 或一个空行以指示没有变化。重定向器是一个外部程序,例如您自己编写的脚本。因此,重定向器程序不是 squid 包的标准部分。但是,源代码分发的 contrib/ 目录中提供了一些示例。由于每个人都有不同的需求,因此由各个管理员编写自己的实现。

重定向器允许管理员控制他的用户可以访问的网站。它可以与透明代理一起使用,以拒绝您网络的用户访问某些网站,例如色情网站等。

重定向器程序必须在标准输入上读取 URL (每行一个),并在标准输出上写入重写的 URL 或空行。此外,squid 在 URL 之后写入附加信息,重定向器可以使用这些信息做出决定。输入行由四个字段组成:

URL ip-address/fqdn 识别方法

- 最初请求的URL。
- 发出请求的客户端的IP 地址和域名 (如果已经被squid 缓存)。
- 为该客户端完成的任何 IDENT / AUTH 查找的结果 (如果启用)。
- 请求中使用的HTTP 方法,例如GET。

未知或未指定的参数用破折号代替。

示例重定向器输入行:

ftp://ftp.gnome.org/pub/GNOME/stable/releases/gnome-1.0.53/README 192.168.12.34/- - 获取

示例响应:

ftp://ftp.net.lboro.ac.uk/gnome/stable/releases/gnome-1.0.53/README 192.168.12.34/- - 获取

可以将指向新 URL 的 HTTP 重定向直接发送到客户端,而不是让 squid 静默地获取替代 URL。为此,重定向器应以 301: 或 302: 开始其响应,具体取决于重定向的类型。

一个叫做 squirm 的简单的非常快速的重定向器是一个很好的起点,它使用正则表达式库来允许模式匹配。

以下 Perl 脚本也可以用作编写您自己的重定向器的模板:

```
#!/usr/local/bin/perl $|=1; # 取消缓冲
输出 while (<>) { s@http://fromhost.com@http://
tohost.org@;打印; }
```

此 Perl 脚本将对 “http://fromhost.com”的请求替换为 “http://tohost.org”。

## 认证器

鱿鱼可以使用身份验证。身份验证可以在各种级别上完成，例如网络或用户。

浏览器能够使用特殊的“授权请求标头”发送用户的身份验证凭据。这工作如下。如果 squid 收到请求，假设有指向 proxy\_auth ACL 的 http\_access 规则列表，squid 会查找授权标头。如果标头存在，squid 对其进行解码并提取用户名和密码。如果标头丢失，squid 将返回状态为 407（需要代理身份验证）的 HTTP 回复。用户代理（浏览器）收到 407 回复，然后提示用户输入名称和密码。名称和密码经过编码，并在授权标头中发送，以供后续向代理请求。

身份验证实际上是在主 squid 进程之外执行的。当 squid 启动时，它会生成许多身份验证子进程。这些进程读取 stdin 上的用户名和密码，并在 stdout 上回复 OK 或 ERR。此技术允许您使用多种不同的身份验证方案。当前支持的方案有：basic、digest、ntlm 和 negotiate。

Squid 有一些基本的身份验证后端。这些包括：

- LDAP：使用轻型目录访问协议
- NCSA：使用 NCSA 风格的用户名和密码文件
- MSNT：使用 Windows NT 身份验证域
- PAM：使用 Unix Pluggable Authentication Modules 方案
- SMB：使用 SMB 服务器，如 Windows NT 或 Samba
- getpwam：使用老式的 Unix 密码文件
- SASL：使用 SASL 库（简单身份验证和安全层）
- ms-win-sspi：Windows 原生验证器
- YP：使用 NIS 数据库

ntlm、negotiate 和 digest 身份验证方案提供更安全的身份验证方法，因为密码不会以明文形式通过有线或空中交换。

每个方案的配置都是通过配置文件中的 auth\_param 目录完成的。每个方案都有一些全局和特定于方案的配置选项。身份验证方案呈现给客户端的顺序取决于方案首次出现在配置文件中的顺序。

具有多个控制器的示例配置文件：

```
#Recommended minimum configuration per scheme: #auth_param negotiate
program <取消注释并完成此行以激活> #auth_param negotiate children 20
startup=0 idle=1 #auth_param negotiate keep_alive on #

#auth_param ntlm program <取消注释并完成此行以激活> #auth_param ntlm children 20 startup=0 idle=1 #auth_param ntlm
keep_alive on #

#auth_param digest program <取消注释并完成这一行> #auth_param digest children 20 startup=0 idle=1
#auth_param digest realm Squid proxy-caching web server #auth_param digest nonce_garbage_interval
5 minutes #auth_param digest nonce_max_duration 30 minutes #auth_param digest nonce_max_count
50 #

#auth_param basic program <取消注释并完成这一行> #auth_param basic children 5 startup=5 idle=1
#auth_param basic realm Squid proxy-caching web server #auth_param basic credentialsttl 2 hours
```

## 访问策略

许多 squid.conf 选项需要使用访问控制列表 (ACL)。每个 ACL 都包含名称、类型和值（字符串或文件名）。 ACL 通常被认为是 squid 缓存配置中最困难的部分，即布局和概念对大多数人来说并不是显而易见的。此外，外部身份验证器和默认 ACL 的使用增加了混乱。

ACL 可以看作是资源的定义，这些资源可能会或可能不会访问 Web 缓存中的某些功能。允许使用代理服务器是这些功能之一。

要规范对某些功能的访问，您必须首先定义一个 ACL，然后添加一行以拒绝或允许访问缓存的功能，从而使用该 ACL 作为参考。在大多数情况下，允许或拒绝的功能是 http\_access，它允许或拒绝 Web 浏览器访问 Web 缓存。相同的原则适用于其他选项，例如 icp\_access（Internet 缓存协议）。

为了确定资源（例如用户）是否可以访问网络缓存，squid 从上到下遍历 http\_access 列表。它将匹配规则，直到找到与用户匹配并拒绝或允许访问的规则。因此，如果您只想允许那些机器在特定 IP 范围内的用户访问代理，您可以使用以下命令：

```
acl ourallowedhosts src 192.168.1.0/255.255.255.0 acl all src 0.0.0.0/0.0.0.0
```

http\_access 允许我们允许的主机 http\_access 拒绝所有

如果来自 192.168.1.2 的用户使用 TCP 连接并请求 URL，squid 将通过 http\_access 行列表工作。它从上到下遍历此列表，在第一个匹配项后停止以决定它们在哪个列表中。在这种情况下，squid 将在第一个 http\_access 行匹配。由于匹配的策略是允许的，squid 将继续允许该请求。

第一行的 src 选项是您可以用来决定请求用户所在域的选项之一。您可以根据源或目标 IP 地址、域或域正则表达式、小时、天、URL 来调节访问、端口、协议、方法、用户名或浏览器类型。 ACL 还可能需要用户身份验证、指定 SNMP 读取社区字符串或设置 TCP 连接限制。

例如，这些行将使所有内部 IP 地址远离 Web，午餐时间除外：

```
acl allowed_hosts src 192.168.1.0/255.255.255.0  
acl 午餐时间 MTWTF 12:00-13:00  
http_access allow allowed_hosts 午餐时间
```

MTWTF 字符串表示星期几，其中 M 指定星期一，T 指定星期二，依此类推。 WHFAS 表示周三至周日。有关更多选项，请查看 squid 在您的系统上安装的默认配置文件。

另一个例子是根据域名阻止某些网站：

```
acl 成人 dstdomain playboy.com sex.com acl ourallowedhosts src  
192.168.1.0/255.255.255.0  
acl 所有 src 0.0.0.0/0.0.0.0  
  
http_access deny adults http_access allow  
ourallowedhosts http_access deny all
```

这些行阻止请求成人 ACL 中列出的站点的用户访问网络缓存 (http\_access)。如果请求另一个站点，如果用户在 ACL ourallowedhosts 指定的范围内，则下一行允许访问。如果用户不在该范围内，第三行将拒绝访问网络缓存。

要使用 **验证器**，您必须告诉 squid 它应该使用哪个程序来验证用户（使用 squid.conf 文件中的 authenticate\_program 选项）。接下来，您需要设置一个类型为 proxy\_auth 的 ACL，并添加一行以使用该 ACL 来规范对 Web 缓存的访问。这是一个例子：

```
authenticate_program /sbin/my_auth -f /etc/my_auth.db acl name proxy_auth REQUIRED  
http_access 允许名称
```

ACL 指向外部验证器 /sbin/my\_auth。如果用户想要访问网络缓存（http\_access 函数），您会期望（像往常一样）如果 ACL 名称匹配则请求被授予。然而，这种情况并非如此！

### 验证器行为

身份验证器允许规则充当拒绝规则！

如果外部验证器允许访问，则允许规则实际上就像拒绝规则一样！随后也会检查任何后续规则，直到找到另一个匹配的 ACL。换句话说：规则 “http\_access allow name” 应该读作 “http\_access deny !name”。感叹号表示否定，因此规则 “http\_access deny !name”的意思是：“拒绝不匹配 ‘name’ 规则的用户访问”。

! squid 总是添加一个默认的 ACL！ squid 自动将

最终规则添加到 ACL 部分，以反转前面的（最后一个）规则：如果最后一个规则是“允许”规则，则会添加“全部拒绝”规则，反之亦然；如果最后一个规则是“拒绝”规则，将自动添加“允许所有”规则。

这两个警告都暗示，如果按原样实施上述示例，最后一行 “http\_access allow name” 隐式添加了最终规则 “http\_access deny all”。如果外部验证器授予访问权限，则不会授予访问权限，但会检查下一条规则 - 如果您自己未指定，则下一条规则是默认拒绝规则！这意味着适当授权的人将被拒绝访问。 squid 的这种异常行为经常被误解，并使许多新手 squid 管理员感到困惑。一个常见的解决方案是添加额外的一行，如下所示：

```
http_access 允许名称 http_access 允许  
所有
```

### 利用内存使用

鱿鱼使用大量内存。出于性能原因，这是有道理的，因为与直接从内存中读取相比，从磁盘读取内容需要花费更长的时间。每个缓存对象的少量元数据保存在内存中，即所谓的 StoreEntry。对于 squid 版本 2，在“小”指针架构（Intel、Sparc、MIPS 等）上为 56 字节，在“大”指针架构（Alpha）上为 88 字节。此外，每个 StoreEntry 都有一个 16 字节的缓存键（MD5 校验和）。

这意味着缓存中的每个对象在内存中都有 72 或 104 字节的元数据。因此，具有 1,000,000 个对象的缓存仅需要 72 MB 的元数据内存。

实际上，鱿鱼需要的远不止这些。 squid 对内存的其他使用包括：

- 用于读取和写入的磁盘缓冲区
- 网络 I/O 缓冲区
- IP 缓存内容
- FQDN 缓存内容
- Netdb ICMP 测量数据库
- 每个请求的状态信息，包括完整的请求和回复标头
- 杂项统计收集

- 完全保存在内存中的热对象

您可以使用 squid.conf 中的许多参数来确定 squid 的内存利用率：

- cache\_mem 参数指定用于缓存热（非常流行）请求的内存量。 squid 的实际内存使用量在很大程度上取决于您的磁盘空间（缓存空间）和您的传入请求负载。减少 cache\_mem 通常也会减少 squid 的进程大小，但不一定。
- squid.conf 中的 maximum\_object\_size 选项指定将被缓存的最大文件大小。大于此大小的对象将不会保存在磁盘上。该值以千字节为单位指定，默认值为 4MB。如果速度比节省带宽更重要，则应保持低值。
- minimum\_object\_size 选项指定小于此大小的对象将不会保存在磁盘上。该值以千字节为单位指定，默认值为 0 KB，这意味着没有最小值（所有内容都会保存到磁盘）。
- cache\_swap 选项告诉 squid 它可能使用多少磁盘空间。如果你有一个大的磁盘缓存，你可能会发现你有足够的内存来有效地运行 squid。如果性能不佳，请考虑增加 RAM 量或减少 cache\_swap。

## 将 Nginx 实现为 Web 服务器和反向代理 (208.4)

考生应该能够安装和配置反向代理服务器 Nginx，包括 Nginx 作为 HTTP 服务器的基本配置。

### 关键知识领域

Nginx

反向代理

基本网络服务器

### 条款和实用程序

- /etc/nginx/
- nginx

资源：[NginX01](#)；[NginX02](#)；各种命令的手册页。

## NGINX

Nginx 可用作网络服务器、HTTP 反向代理或 IMAP/POP3 代理。它发音为 engine-x。Nginx 的表现非常出色，以至于 Netflix、Wordpress 和 GitHub 等大型网站都依赖 Nginx。它不像大多数其他网络服务器软件那样使用线程工作，但它使用事件驱动（异步）架构。它占地面积小，并且在可预测的资源使用情况下在负载下表现非常好。这种可预测的性能和较小的内存占用使得 Nginx 在小型和大型环境中都很有趣。Nginx 作为开源软件分发。还有 Nginx Plus，这是商业版。本书将重点关注开源版本。

## 反向代理

代理服务器是中间服务器或中间服务器,它将来自多个客户端的内容请求转发到 Internet 上的不同服务器。反向代理服务器是一种代理服务器,通常位于专用网络的防火墙后面,并将客户端请求定向到适当的后端服务器。反向代理提供额外的抽象和控制级别,以确保客户端和服务器之间的网络流量顺畅流动。

反向代理服务器的常见用途包括:

- 负载平衡 反向代理服务器可以充当“交通警察”,位于您的后端服务器前面,以最大化速度和容量利用率的方式在一组服务器之间分配客户端请求,同时确保没有服务器过载,这会降低性能。如果服务器出现故障,负载均衡器会将流量重定向到剩余的在线服务器
- Web 加速 反向代理可以压缩入站和出站数据,以及缓存通常请求的内容,这两者都可以加速客户端和服务器之间的流量。它们还可以执行额外的任务,例如 SSL 加密,以减轻您的 Web 服务器的负载,从而提高它们的性能。
- 安全性和匿名性 通过拦截发往您的后端服务器的请求,反向代理服务器可以保护他们的身份,并充当额外的防御措施来抵御安全攻击。它还确保可以从单个记录定位器或 URL 访问多个服务器,而不管您的局域网结构如何。

使用 Nginx 作为反向 HTTP 代理并不难配置。一个非常基本的反向代理设置可能如下所示:

```
地点 / {
    proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-
    For $remote_addr; proxy_set_header 主机
        $主机;
    代理通行证
        http://localhost:8000;
}
```

在这个例子中,Nginx 收到的所有请求,根据 /etc/nginx/nginx.conf 中服务器参数的配置,被转发到在本地主机上运行并侦听端口 8000 的 HTTP 服务器。Nginx 配置文件如下所示:

```
服务器 { 听 80;

    根 /var/www/; index
        index.php index.html index.htm;

    服务器名称 example.com www.example.com;

    地点 / {
        try_files $uri $uri/ /index.php;
    }

    位置 ~ \.php$ {
        proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For
            $remote_addr; proxy_set_header 主机 $host; proxy_pass http://localhost:8080;

    }

    位置 ~ /\.ht { 全部拒绝;
    }
}
```

添加以 location ~ /\.ht 开头的行是为了防止 Nginx 显示 Apache 的 .htaccess 文件的内容。  
try\_files 行用于尝试为访问者请求的任何页面提供服务。如果 nginx 不可用,则将文件传递给代理。

## 基本网络服务器

Nginx 中配置文件的默认位置是 /etc/nginx/nginx.conf。

能够提供 html 文件的基本 Nginx 配置如下所示：

```
server { # 这将
    监听所有接口,你可以选择一个特定的 IP # 例如 listen xxxx:80;听80;

    # 在这里你可以设置一个服务器名称,你可以使用通配符,例如 *.example.com # 但是记住如果你使用 server_name *.example.com;您将只匹配子域 # 以匹配两个子域和主域同时使用 example.com 和 *.example.com server_name example.com www.example.com;

    # 最好将服务器块的根放在服务器级别,而不是 ←
    # 位置级别
    # 任何位置块路径都将相对于此根。根 /usr/local/www/example.com;

    # 访问和错误记录。注意:无法关闭错误记录。 access_log /var/log/nginx/example.access.log; error_log /var/log/nginx/
    example.error.log;

    地点 / {
        # 重写规则和其他标准可以在这里 # 记住尽可能避免使用 if() (http://wiki.nginx.org/IfIsEvil)
    }
}
```

对于 PHP 支持,Nginx 依赖于 PHP fast-cgi spawner。最好是 php-fpm,可以在<http://php-fpm.org> 找到。

它具有一些独特的功能,如自适应进程生成和统计,并且能够使用不同的 uid/gid/chroot/environment 和不同的 php.ini 启动 worker。可以使用此功能替换 safe\_mode。

您可以将以下内容添加到 nginx.conf。更好的做法是将内容放在一个文件中,并将这个文件包含到 Nginx 的主配置文件中。创建一个文件,例如 php.conf 并在 Nginx 主配置文件末尾包含下一行:

包括 php.conf;

php.conf 的内容:

```
位置 ~ \.php {
    # 出于安全原因,强烈建议使用下一行 try_files $uri =404;

    fastcgi_param QUERY_STRING fastcgi_param           $query_string; $request_method;
    REQUEST_METHOD fastcgi_param CONTENT_TYPE          $content_type; $content_length;
    fastcgi_param CONTENT_LENGTH

    fastcgi_param SCRIPT_NAME                         $fastcgi_script_name;

    # 如果你的下一行仍然包含 $document_root # 考虑切换到 $request_filename 提供 # 更好的指令支持,例如 alias
    $request_filename; fastcgi_param SCRIPT_FILENAME

    fastcgi_param REQUEST_URI fastcgi_param           $request_uri; $document_uri;
    DOCUMENT_URI fastcgi_param                         $document_root; $server_protocol;
    DOCUMENT_ROOT fastcgi_param
    SERVER_PROTOCOL
```

```
fastcgi_param GATEWAY_INTERFACE CGI/1.1; fastcgi_param 服务器软件
nginx;

fastcgi_param REMOTE_ADDR fastcgi_param
REMOTE_PORT fastcgi_param SERVER_ADDR
fastcgi_param SERVER_PORT fastcgi_param
SERVER_NAME $远程地址; $远程端
口; $服务器地址;
$服务器端口; $服务器
名称;

# 如果使用 unix socket... # fastcgi_pass unix:/tmp/php5-fpm.sock;
# 如果使用 TCP 连接... fastcgi_pass 127.0.0.1:9000;
}
```

## 问题和解答

### 网页服务

1. 包含可以在每个目录基础上设置的 Apache 配置项的文件的名称是什么？

.htaccess。 .htaccess [210]

2. Apache Web 服务器支持哪种协议,可以实现客户端和服务器之间的安全在线通信?

SSL。 SSL [227]

3. PKC (Public Key Cryptography)背后的原理是什么?

PKC 基于公钥和私钥,其中消息的发送方使用接收方的公钥加密数据。只有相应私钥的所有者才能解密此数据。公钥加密[227]

4. Covalent 的 Raven SSL 模块和 mod\_ssl 有什么区别?

区别在于许可证; mod\_ssl 是开源的,而 Covalent 的 Raven SSL 模块不是。

5. 哪个硬件组件直接影响 Apache Web 服务器的性能?

内存。内存[212]

6. 哪个设置控制 Apache Web 服务器的并发连接数?

最大客户。最大客户端[222]

7. Apache 网络服务器访问日志文件中使用哪种标准格式写入条目?

通用日志格式。通用日志格式[213]

8. 说出 Apache Web 服务器使用的两种访问控制方法。

自主访问控制 (DAC)和强制访问控制 (MAC) 。访问控制[214]

9. 描述配置认证模块的两种方法。

一种方法是使用 Apache 配置文件中的指令,另一种方法是使用 .htaccess 文件。认证模块[217]

10. 哪个命令用于重启 Apache Web 服务器?

apachectl 重新启动。阿帕奇重启[221]

11. 可以通过哪些方式检查Apache Web 服务器配置文件的语法错误?

apachectl 配置测试。 apachectl 配置测试[226]

12. 描述 Apache 虚拟主机。

虚拟主机是一种技术,它提供了使用单个 Web 服务器在物理主机上托管多个域的能力。[虚拟主机](#)

13. 说出实现网络服务器虚拟主机的两种方法。

基于名称和基于 IP 的虚拟主机。[虚拟主机方法\[222\]](#)

14. 首选两种虚拟主机方法 (名称或基于 ip)中的哪一种?为什么?

基于名称的虚拟主机,因为它减轻了对稀缺 IP 地址的需求。[首选虚拟主机\[222\]](#)

## 第9章

# 文件共享 (209)

本主题的权重为 8 分,包含以下目标:

目标 209.1; SAMBA 服务器配置 (5 分) 考生应该能够为各种客户端设置 SAMBA 服务器。此目标包括将 Samba 设置为 Windows Active Directory 域的独立服务器和成员服务器。两种设置都应配置为与客户端共享目录和打印机。

目标 209.2; NFS 服务器配置 (3 分) 考生应该能够使用 NFS 导出文件系统。这个对象包括访问限制、在客户端上安装 NFS 文件系统和保护 NFS。

## SAMBA 服务器配置 (209.1)

资源: [Sharpe01](#);各种命令的手册页。

目标 209.1;配置 Samba 服务器 (5 分) 考生应该能够为各种客户端设置 SAMBA 服务器。此目标包括将 Samba 设置为 Windows Active Directory 域的独立服务器和成员服务器。两种设置都应配置为与客户端共享目录和打印机。

### 关键知识领域

Samba 4 文档

桑巴配置文件

Samba 工具和实用程序

在 Linux 上挂载 Samba 共享

桑巴守护进程

将 Windows 用户名映射到 Linux 用户名

用户级和共享级安全

### 条款和实用程序

- smbd, nmbd
- smbstatus, testparm, smbpasswd, nmblookup
- smbclient

- 桑巴工具
- 网
- /etc/smb/
- /var/日志/samba/

### 什么是桑巴？

Samba 实现服务器消息块 (SMB) 协议。这是微软用来实现文件和打印机共享的协议。通过在 Linux 机器上安装 Samba，运行 Windows 操作系统和 SMB 客户端可用的其他平台的机器可以连接到 Linux 机器，从而使用 Linux 机器提供的文件和打印机。共享资源也称为“共享”或“服务”。

Samba 可用于许多平台，包括 Linux、AIX、HP-UX、Solaris、FreeBSD、OS/2、AmigaOS。有关支持 Samba 的平台以及为您的平台下载二进制或源代码分发的更多信息，请参阅 [Samba：打开 Windows 走向更广阔的世界](#)。

### 安装 Samba 组件

根据您的分布，您可以

- 获取源代码并自行编译
- 使用 yum 或 rpm (Red Hat、SuSE 等) 安装包
- 使用 apt-get 或 aptitude 安装软件包 (Debian、Ubuntu)

Samba 可以从 inetc 运行，也可以作为守护进程运行。当通过 inetc 运行时，您可以节省一些内存并使用 tcpwrappers 来获得额外的安全性。当作为守护进程运行时，服务器总是准备就绪并且会话速度更快。如果您希望使用加密密码，您将需要一个单独的 /etc/samba/smbpasswd 文件，因为布局与 /etc/passwd 不同。在安装期间，您可以选择从 /etc/passwd 文件生成 /etc/samba/smbpasswd。如果您选择不这样做，请使用 smbpasswd 为用户设置单独的密码。

Samba 由两个守护进程组成：

- nmbd：处理 NetBIOS 名称查找和 WINS 请求的 NetBIOS 名称服务守护程序。如果您已将 Samba 设置为 WINS 服务器，则将运行 nmbd 的额外副本。此外，如果 DNS 用于转换 NetBIOS 名称，则将运行 nmbd 的另一个副本。
- smbd：处理文件和打印机访问的服务器消息块守护进程。对于连接到服务器的每个客户端，smbd 运行的额外副本。

Samba 同时使用 UDP 和 TCP 协议。TCP/139 用于文件和打印机共享。UDP 用于 NetBIOS 名称的注册和翻译，以及浏览网络。UDP/137 用于名称服务请求和响应。

UDP/138 用于数据报服务以传输少量数据，例如服务器公告。

### Samba 命令

Samba 核心命令

SMB状态

报告当前的 Samba 连接：

\$ SMB状态			
Samba 版本 4.1.12			
PID	用户名	团体	机器
23632	没有人	没有人	10.20.24.186 (ipv4:10.20.24.186:49394)
服务			
进程号	机器	连接于	
上市	23632	10.20.24.186	2015 年 10 月 10 日星期六 10:15:11
没有锁定的文件			

## 测试参数

检查 smb.conf 配置文件的内部正确性。如果 testparm 在 smb.conf 文件中发现错误,它会向调用程序返回退出代码 1,否则返回退出代码 0。这允许 shell 脚本测试 testparm 的输出。

有用的命令行选项：

-s 打印服务定义而不提示回车

-v 列出所有选项;默认只列出 smb.conf 中指定的那些

## 密码

更改用户的 SMB 密码。默认情况下（当不带参数运行时）smbpasswd 将尝试更改本地计算机上当前用户的 SMB 密码。这类似于 passwd(1) 程序的工作方式。当由 root 运行时,它可用于在配置的密码后端管理用户帐户。请注意,即使此实用程序称为 smbpasswd,它也不一定会将更改写入 smbpasswd 文件。smbpasswd 在 smb.conf 中配置的 passdb 后端工作。另请参阅[帐户信息数据库](#)。

命令行用法：

作为 root:smbpasswd [选项] [用户名]

作为普通用户:smbpasswd [选项]

有用的命令行选项：

-a 将新用户添加到密码数据库。

-x 从数据库中删除用户

## nmblookup

用于查询 NetBIOS 名称并将它们映射到网络中使用 NetBIOS over TCP/IP 查询的 IP 地址。此命令的选项允许将名称查询定向到特定的 IP 广播区域或特定的机器。所有查询都通过 UDP 完成。

有用的命令行选项：

-M 搜索主浏览器。

-R 递归。当使用 nmblookup 直接查询带有 UNICAST 命令行选项的 WINS 服务器时,需要递归以使 WINS 服务器响应与其自己的 netbios 名称或 IP 地址无关的查询。如果没有设置递归,WINS 服务器将只响应它自己的 netbios 名称。

-U <unicast address> 将查询发送到给定的 UNICAST 地址 (WINS 服务器的)而不是广播查询。

示例：“nmblookup -R -U 10.10.10.2 客户端名称”

## 客户端

是可以连接到 SMB/CIFS 服务器的客户端。它提供了一个类似于 ftp 程序的界面（请参阅 `ftp(1)`）。操作包括从服务器获取文件到本地机器、将文件从本地机器放到服务器、从服务器检索目录信息等操作。

有用的命令行选项：

`-L <netbios 名称/IP>` 列出响应给定 netbios 名称的服务器上可用的服务。

`-I <IP address>` 直接连接到给定的 IP 地址，而不是向网络查询给定 netbios 的 IP 地址  
姓名。

`-c <command>` 在服务器上运行给定的 SMB 命令。一种实现是使用 `smbclient` 进行打印。

`-U` 以给定用户身份连接。

## 桑巴工具

`Samba-tool` 是 `samba4` 可用的主要管理工具。当它配置为 Active Directory 域控制器 (AD DC) 时，它可用于配置和管理 `samba` 服务器的所有方面。尽管联机帮助页当前另有说明，但不支持使用 `samba-tool` 将服务器配置为域成员或独立服务器。这些选项将在 `samba-tool` 的未来版本中删除。请注意，使用软件包安装时，此工具并非在所有系统上都可用。例如，在 RHEL7 和 CentOS 7 上，它只有在从源代码安装 Samba4 时才可用。

命令的简短列表及其用途如下所示。有关命令选项的完整列表，您可以查看联机帮助页或[在线联机帮助页](#)

`dbcheck` 检查本地 AD 数据库是否有错误。

`委托` 管理委托。

`dns` 管理 DNS 记录。

`domain` 管理域选项，例如创建 AD DC。

`drs` 管理目录复制服务 (DRS)。

`dsacl` 管理 DS ACL。

`fsmo` 用于管理灵活的单主机操作 (FSMO)。

`gpo` 管理组策略对象 (GPO)。

`group` 管理或创建组。

`ldapcmp` 比较两个 LDAP 数据库。

`ntacl` 管理 NT ACL。

`rodc` 管理只读域控制器 (RODC)

`站点` 管理站点。

`spn` 管理服务主体名称 (SPN)。

`testparm` 来检查配置文件。

`time` 检索服务器上的时间。

`user` 管理或创建用户。

网

用于管理 Samba 和远程 CIFS 服务器的工具。Samba 网络实用程序旨在像可用于 Windows 和 DOS 的网络实用程序一样工作。第一个参数应用于指定执行特定命令时使用的协议。ADS 用于 ActiveDirectory, RAP 用于旧的 (Win9x/NT3) 客户端, RPC 可用于 NT4 和 Windows。如果省略此参数,net 将尝试自动确定它。并非所有命令都适用于所有协议。

网络的功能过于广泛,无法在本节中涵盖。查看 man net 或 net help 以显示可用命令和命令行选项的列表。net help <command> 将给出命令的具体信息:

```
$ 净帮助用户

net [<方法>] 用户 [杂项。选项] [目标]
      列出用户

net [<方法>] 用户删除 <名称> [杂项。选项] [目标]
      删除指定用户

net [<方法>] 用户信息 <名称> [杂项。选项] [目标]
      列出指定用户的域组

net [<method>] user ADD <name> [password] [-c container] [-F user flags] [misc.选项] ←
      [目标]
      添加指定用户

net [<method>] user RENAME <oldusername> <newusername> [targets]
      重命名指定用户

有效方法: (如果未指定则自动检测)
  广告          活动目录 (LDAP/Kerberos)
  rpc           DCE-RPC
  说唱          RAP ((旧系统))

有效目标:选择一个 (无默认为本地主机)
  -S 或 --server=<server> 服务器名称
  -I 或 --ipaddress=<ipaddr> 目标服务器的地址 -w 或 --workgroup=<wg> 目标工作组或域

有效的杂项选项是:
  -p 或 --port=<port> 目标上的连接端口
  -W 或 --myworkgroup=<wg> 客户端工作组 -d 或 --debuglevel=<level> 调试级别 (0-10) -n 或
  --myname=<name> 客户端名称

  -U 或 --user=<name> 用户名
  -s 或 --configfile=<path> smb.conf 文件的路径名 -l 或 --long -V 或 --version
      显示完整信息
      打印samba版本信息

  -P 或 --machine-pass          验证为机器帐户
  -e 或 --encrypt-k 或 --kerberos 加密 SMB 传输 (仅限 UNIX 扩展服务器)
      使用 kerberos (活动目录)身份验证
  -C 或 --comment=<comment> 描述性注释 (仅用于添加) -c 或 --container=<container> LDAP 容器,默认为 cn=Users (仅用于在
  ADS 中添加 ←。
      )
```

使用 net 从服务器 “sambaserver”获取共享列表:

```
$ net -S sambaserver -U 爱丽丝分享
输入alice的密码:public share1

分享2
打印机_1
```

IPC\$  
爱丽丝  
打印机\_2

使用 net 获取服务器 “sambaserver”的当前时间：

```
$ net -S samba 服务器时间
2015 年 10 月 10 日星期六 10:10:04
```

命令不是 Samba 核心的一部分

挂载

注意：尽管 smbmount 已被大多数主要 Linux 发行版放弃，取而代之的是 mount.cifs，但您仍然可以在 LPIC2 考试期间遇到有关 smbmount 的问题。

即使 smbmount 是由 Samba 社区维护的，它也不是核心 samba-client 包的一部分。“smbfs”包包含 smbmount 命令，必须安装才能使用 smbmount。

smbmount 用于挂载通过 SMB 共享的文件系统。这些文件系统很可能在 Windows 系统上找到，并与安装了 SMB 客户端软件的 Linux 系统共享。smbmount 是用于安装 SMB 文件系统的命令行实用程序。对于更永久的实现，可以在 /etc/fstab 中使用 smbfs。

挂载 SMB 文件系统的两种方法都接受确定文件系统挂载方式的选项。此处列出了最常见的选项：

username 定义用于 SMB 会话身份验证的用户名。

password 定义用于验证 SMB 会话的密码。

credentials 此选项指向包含用户名和密码的文件。使用此选项优于在命令行选项或 /etc/fstab 中使用用户名和密码。此文件必须有适当的保护，以便只有用户和/或 root 可以读取它。

用户名=值  
密码=值

uid 定义 UID，用于挂载文件系统上文件的本地表示。

gid 定义 GID，用于挂载文件系统上文件的本地表示。

fmask 在已安装文件系统的本地表示中定义远程文件的权限。这不会影响远程服务器上的实际权限。

**重要提示：**该选项的名称具有欺骗性。它不是掩码，而是定义的实际权限。

dmask 在已安装文件系统的本地表示中定义远程目录的权限。这不影响远程服务器上的实际权限。

**重要提示：**该选项的名称具有欺骗性。它不是掩码，而是定义的实际权限。

rw/ro 以读写或只读方式挂载文件系统。

命令行用法示例：

```
smbmount //windows/winshare2 /opt/winshare2 -o \
username=alice.jones,password=Alice,uid=nobody,gid=nobody,fmask=775,dmask=775,rw,hard
```

/etc/fstab 用法示例：

```
//windows/winshare2 /opt/winshare2 smbfs \
username=alice.jones,password=Alice,uid=nobody,gid=nobody,fmask=775,dmask=775,rw,hard <-- :://windows/winshare2 0 0
```

## 桑巴记录

Samba 默认情况下将日志记录写入 `/var/log/samba/` 目录中的文件：

`log.nmbd` 从 Netbios 名称查找守护进程中记录。

`log.smbd` 从 SMB 守护进程记录。

可以在 Samba 配置中使用全局参数配置日志记录。有关一些最有用的参数，请参阅[配置参数](#)。

## 帐户信息数据库

Samba 可以配置为使用不同的后端来存储或检索帐户信息。此处描述了最重要的内容。 `Smb.conf` 配置选项：“`passwd backend`”。

### 密码

使用 `smbpasswd` 方法，纯文本文件包含所有帐户信息。密码已加密。

使用 `smbpasswd` 的缺点：

- 无法扩展。
- 无复制。
- 缺少 Windows 信息（RID 或 NT 组）的存储。

不建议使用 `smbpasswd`，因为它不能很好地扩展或保存任何 Windows 信息。

### tdbsam

`Tdbsam` 还缺乏可扩展性，因为它只是一个不支持复制的本地数据库（Trivial 数据库）。 `tdbsam` 优于 `smbpasswd` 的优势之一是它还可以将 Windows 信息与帐户一起存储。

不建议在企业环境中使用 `tdbsam`，因为它不能很好地扩展并且保存任何 Windows 信息。`Tdbsam` 可用于建议最多 250 个用户的独立 Samba 服务器。

### ldapsam

在企业环境中，建议使用 `ldapsam`。`Ldapsam` 使用 LDAP 作为后端，并且 LDAP 具有高度可扩展性。

## 桑巴配置

Samba 配置目录 `/etc/smb` 或 `/etc/samba`。

LPI 目标要求了解有关 `/etc/smb/` 的信息。在某些发行版中，使用 `/etc/samba/` 代替。`/etc/smb/` 或 `/etc/samba/` 中存在的文件和文件夹是：

- `lmhosts` - Samba NetBIOS 主机文件；
- `smb.conf` - Samba 套件的配置文件；
- `netlogon` - 用户登录的登录目录。

## 配置文件

Samba 通过 /etc/samba/smbd.conf 配置。该文件由包含配置选项的部分组成。该部分的名称是共享资源的名称。

## 特别栏目

samba 配置文件中有三个特殊部分：

[global] 全局 Samba 配置

[homes] 特殊部分：主目录的定义

可以使用用户名作为服务访问主目录，或者直接访问“homes”服务：

```
smbclient //sambaserver/alice -U alice smbclient //sambaserver/  
homes -U alice
```

这两个命令都会导致访问用户“alice”的主目录。

[打印机] 特殊资源或服务：全局配置以启用对所有打印机的访问

注意：单个打印机也可以作为具有可打印参数的服务提供。

## 配置参数

在本节中，将解释最重要的配置选项，按 Samba 配置部分（类型）分组。大多数也可以在后面的示例部分找到。

smb.conf 手册页将参数分为两组：

global 只能在 Samba 配置的 [global] 部分中使用的参数。

服务 在 Samba 配置的服务部分中使用的参数。

其中一些参数也可以全局使用。

### [全球的]

[global] 部分包含全局参数，但它也用于在全局上下文中设置服务参数（如果未为特定服务设置参数，则提供默认值）。

netbios name 此选项设置 Samba 服务器已知的 NetBIOS 名称。这个名字将是服务的名字

在下面做广告。默认情况下，它与系统的主机名相同。

netbios aliases 此选项设置一个别名，通过该别名可以知道 Samba 服务器。

日志文件 此选项指示写入日志记录的文件。文件名接受启用例如写入日志的宏

每个客户端的文件： /var/log/samba/log.%m。

工作组 服务器和客户端必须是同一个工作组的成员。

领域 此选项指定要使用的 kerberos 领域。该领域用作 NT4 域的 ADS 等效项。

服务器字符串 您想要在列表上下文中显示的任何字符串。

加密密码 Windows 加密密码。还需要为 Samba 打开此选项。

security 这个选项决定了使用什么安全模式。最常用的是独立文件服务器或同时用作 DC 的 Samba 服务器的用户。如果 Samba 服务器连接到 Windows 域，则必须将此选项设置为广告或域。

unix password sync [global]部分中的这个布尔参数控制当 smbpasswd 文件中的加密 SMB 密码更改时,Samba 是否尝试将 UNIX 密码与 SMB 密码同步。如果此项设置为 yes (unix password sync = yes) ,在 passwd 程序参数中指定的程序称为 AS ROOT - 允许设置新的 UNIX 密码而无需访问旧的 UNIX 密码（作为 SMB 密码更改代码无法访问旧的明文密码）。

passdb backend 这个选项决定了使用什么账户/密码后端。另请参阅[帐户信息数据库](#)

最常用的选项是：

smbpasswd[:argument] |旧明文 passdb 后端。可选择将 smbpasswd 文件的路径作为参数。

示例:passdb 后端 =smbpasswd:/etc/samba/smbpasswd

tdbsam[:argument] 基于 TDB 的密码存储后端。可选择将 TDB 文件的路径作为参数。

示例:passdb 后端 =tdbsam:/etc/samba/private/passdb.tdb

ldapsam[:argument] LDAP 后端。可选择将 LDAP URL 作为参数。 (默认为 “ldap://localhost” )

示例:passdb 后端 =ldapsam:ldap://localhost

用户名映射[global]部分中的此选项允许您将客户端提供的用户名映射到服务器上的另一个用户名。最常见的用法是将 DOS 或 Windows 机器上使用的用户名映射到 UNIX 系统上使用的用户名。另一种用法是将多个用户映射到一个用户名,以便他们可以更轻松地共享文件。用户名映射是一个文件,其中每行应在左侧包含一个 UNIX 用户名,然后在右侧包含一个“=”,后跟以空格分隔的用户名列表。必须使用引号来指定包含空格的用户名。右侧的用户名列表可能包含 @group 形式的名称,在这种情况下,它们将匹配该组中的任何 UNIX 用户名。特殊客户端名称“\*”是一个通配符,可用于将未知名称与已知用户匹配。映射文件的每一行最长可达 1023 个字符。如果一行以“!”开头如果它与名称匹配,则处理将在该点停止。这对于在通配符之前使用的行很有用,否则名称仍将映射到使用通配符的行。

这是一个例子：

```
用户名映射 = /usr/local/samba/private/usermap.txt
```

usermap.txt 的示例内容：

```
root = administraor 管理员
nobody = guest pcguest smbguest alice.jones = alice
readonly = glen fred terry sarah lachlan = Lachlan Smith
```

```
用户 = @sales
```

guest ok 此参数配置服务的访客访问。

map to guest 是否启用来宾访问此选项确定将哪些会话映射到来宾访问。可用值是：

- 绝不
- 不良用户
- 密码错误
- Bad Uid (仅在 ADS 或 DOMAIN 安全模式下可用)

服务部门

以下参数用于服务定义（与普通服务一样特殊）。

path 使用此参数的上下文决定了它的解释方式：

- 在[homes]部分,它指定必须作为用户主目录的目录路径。如果省略,主目录默认为系统的主目录。如果使用此参数,则必须包含 "%S"宏,扩展为用户名。

- 在设置为可打印的部分中,此参数指向打印机假脱机文件在发送到打印队列之前写入的目录。如果打印机配置为访客访问,则该目录必须是全局可写的并且设置了粘滞位。

- 在共享定义中,此参数指向共享必须授予访问权限的目录。

**comment** 显示在服务列表中的文本字段。

**printer** 打印机名称 在配置单个打印机时指向本地打印队列。

**printable** 将服务声明为打印机。

**browsable** 使服务可浏览。客户端可以浏览到服务,而不必知道服务的完整路径。

**guest ok** 已为此服务 (或全局)启用访客访问。

**(in)valid users** 提供允许访问此服务 (有效用户)或拒绝访问 (无效用户)的用户列表。以“@”开头的名称被解释为 NIS 网络组或 Unix 组。当名称以“+”开头时, nsswitch 机制用于查找组。带有“&”的组只会在 NIS 中被查找。有关详细信息,请参阅手册。

**hosts allow|deny** 提供可以授予 (hosts allow)或拒绝 (hosts deny)访问权限的客户端列表。名字可以是 IP 地址、网络或主机名。以“@”开头的名称是 NIS 网络组。

**writable** 确定是否允许用户写入此服务。默认为“否”。

## 安全级别和模式

Samba 知道两个安全级别:“用户级别”和“共享级别”。服务器将通知客户端安全级别,客户端将根据选择的级别进行响应。安全级别由设置安全模式决定。

安全模式是全局设置。

### 用户级安全

用户级安全意味着每个连接都通过用户名和密码组合进行身份验证,该组合必须与所请求服务的授权相匹配。对于“用户级”安全,服务器可以设置为三种模式:

用户安全=用户

Samba 作为独立服务器运行,并将使用本地密码数据库。

广告安全=广告

Samba 将充当 ADS 领域中的 Active Directory 域成员。

域安全=域

Samba 将向 Windows NT 主域控制器或备用域控制器验证用户名/密码。

### 共享级安全

安全=分享

对于共享级别的安全性,客户端希望密码与每个共享相关联,而与用户无关。使用共享级别的安全性,客户端将只传递远程用户提供的密码,并对每个单独的共享执行此操作。然后 Samba 服务器将尝试将密码与配置的用户列表相匹配 (如果为受影响的共享提供),或者将使用系统调用 (查看 nsswitch.conf 或 /etc/passwd) 来查找与提供的密码。

共享级服务参数:

仅用户仅用户 - 是

只有用户名中列出的用户才能访问此服务。如果没有给出与提供的密码匹配的任何用户使用权。

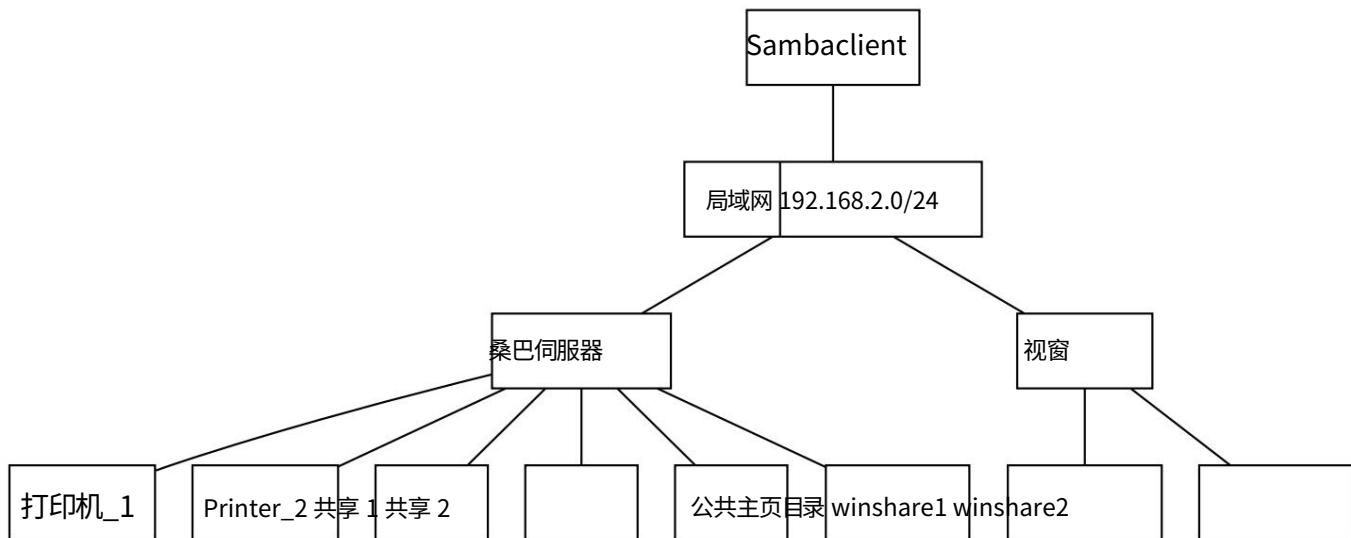
用户名 用户名 =fred, 爱丽丝

确定哪些用户有权访问此服务。

注意:因为共享级安全性,访问共享的密码不是只有一个人知道,而是所有需要访问共享级安全性的人都知道,共享级安全性被认为是不安全的,因此已从 Samba 版本中删除了对共享级安全性的支持4.

## 例子

下图描述了用于实现下述示例的环境。



我们通过网络连接了三台机器,我们希望在其上完成以下任务:

- 机器“sambaserver”和“windows”包含其他机器必须能够操作的文件。
- 所有机器必须能够使用连接到“sambaserver”的打印机。
- “sambaserver”正在运行 Linux 并安装了 Samba。
- “windows”正在运行 Microsoft Windows。
- “sambaclient”正在运行 Linux 并安装了 smbclient 以能够作为 Samba 客户端运行。
- 我们只想共享“sambaserver”和“windows”上的部分资源。

-向所有人公开分享

-使 share share1 可供 alice 使用

-使经过身份验证的用户可以使用 share share2

-使主目录对其各自的拥有者可用-将远程用户 alice.jones 映射到用户 alice

-使 sambaclient 上的用户可以使用 windows 上的共享

-允许所有人在 sambaserver 上的所有打印机上打印

-禁止从 sambaclient 在 Printer\_1 上打印

-列出 sambaserver 上的可用服务

支持示例的基本 [global] 部分

支持以下示例所需的基本全局部分：

```
[global] 工作
组 = OURGROUP 服务器字符串 = Linux
Samba 服务器 %L for LPIC2 examples encrypt passwords = yes security = user netbios name = sambaserver

netbios 别名 = ss2 日志文件 = /var/log/
samba/log.%m map to guest = bad user hosts allow =

有效用户 =
客人好 = 否
```

示例：向所有人提供“公开”共享

插入或修改的配置部分以实现此示例：

```
[public]
comment = Public Storage on %L path = /export/public
Browsable = yes writeable = yes guest ok = yes # valid
users =
```

添加了[public]部分。

- 此服务可访问的路径是/export/public。
- 该服务是可浏览的，因此客户端可以通过直接连接到 Samba 服务器来浏览该服务。
- 服务可写。
- 访客访问已启用，因此无需身份验证。

· 未设置有效用户并使用全局值（默认为“所有经过身份验证的用户”）：所有经过身份验证的用户都可以访问。

所有经过身份验证的用户都可以访问，无法通过身份验证的用户将以“访客”身份获得访问权限。

创建一个测试文件，连接到无法通过身份验证的帐户“jack”（实际上是“guest”），检查活动共享并将测试文件复制到共享。

```
$ touch jack.txt $ smbclient //  
SAMBASERVER/public -U jack -N Domain=[OURGROUP] OS=[Unix] Server=[Samba  
4.1.12] smb:\> volume Volume: [public] 序列号 0x2b5c2e91 smb:\> put jack.txt 将文件 jack.txt  
作为 \jack.txt (0.0 kb/s) (平均 0.0 kb/s) smb:\> ls  
  
. ..  
public.txt 杰克.txt
```

	日期
丁	2015 年 10 月 21 日星期三 09:16:57
丁	2015 年 10 月 21 日星期三 07:44:56
否	2015 年 10 月 21 日星期三 07:45:07
~	2015 年 10 月 21 日星期三 09:16:57

54864 个大小为 131072 的块。可用 47234 个块  
某人:\>

smbstatus 的输出显示来自用户 “nobody”的会话,这是我们 (默认)为 “guest”配置的 Linux 帐户,并检查 “public”共享上的测试文件:

```
$ SMB状态
Samba 版本 4.1.12
PID      用户名      团体      机器
24265    没有人     没有人    10.20.27.158 (ipv4:10.20.27.158:49009)
服务      进程号      机器      连接于
上市      24265 10.20.27.158 2015 年 10 月 21 日星期三 08:58:48
没有锁定的文件
$ pwd /
/export/public $ ls -l
总计 0
-rw-r--r--。 1 没有人没有人 0 10 月 21 日 09:16 jack.txt -rw-r--r--。 1 root root 0 10 月 21 日 07:45 public.txt
```

示例:使 “share1”共享可供 alice 使用

插入或修改的配置部分以实现此示例:

```
[分享 1] 评论 =
在 %L 上分享 1
path = /export/share1 # guest ok = no
browsable = yes writeable = yes valid users
= alice
```

·添加了[share1]部分。

- 此服务可访问的路径是/export/share2。
- 该服务是可浏览的,因此用户可以通过直接连接到 Samba 服务器来浏览该服务。
- 服务可写。
- guest ok未设置且使用全局值 (默认值= “no” ) :不允许来宾访问。
- 有效用户设置为 “alice”以允许Linux 用户 “alice”访问。

以 fred 身份访问 share1 失败:

```
$ smbclient //SAMBASERVER/share1 输入 fred 的密码:
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] 树连接失
败:NT_STATUS_ACCESS_DENIED
```

成功尝试以 alice 身份访问 share1:

```
$ smbclient //SAMBASERVER/share1 输入 alice 的密码:
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] smb: \>
volume
卷: |share1|序列号 0xd62d5fc5 smb: \>
```

示例：使“share2”共享对经过身份验证的用户可用

插入或修改的配置部分以实现此示例：

```
[share2]
comment = %S 上 %L
path = /export/share2 browsable = yes
writeable = no

# guest ok = no # 有效用户 =
```

·添加了[share2]部分。

- 此服务可访问的路径是/export/share2。
- 该服务是可浏览的，因此用户可以通过直接连接到 Samba 服务器来浏览该服务。
- 该服务不可写。
- guest ok未设置且使用全局值（默认值=“no”）：不允许来宾访问。

·未设置有效用户并使用全局值（默认=“空”）：所有经过身份验证的用户都可以访问。

因为guest ok默认为全局值“no”，而空的有效用户默认为全局值“authenticated user”所有（且只有）经过身份验证的用户可以访问“share2”

任何

尝试以访客身份访问 share2 失败：

```
$ smbclient //SAMBASERVER/share1 输入 jack 的密码：
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] tree
connect failed: NT_STATUS_ACCESS_DENIED
```

成功尝试以经过身份验证的用户身份访问 share2：

```
$ smbclient //SAMBASERVER/share2 输入 alice 的密码：
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] smb:
\> volume Volume: |share2|序列号 0xb954cdf0 smb: \>
```

示例：使主目录对其各自的所有者可用

插入或修改的配置部分以实现此示例：

```
[homes]
comment = %U  s homedirectory on %L from %m # path = browsable = no

writeable = yes # guest ok =
no # 有效用户 =
```

·添加了[public]部分。

- 此服务可访问的路径是/export/public。
- 该服务是可浏览的，因此用户可以通过直接连接到 Samba 服务器来浏览该服务。

- 服务可写。
  - 访客访问已启用，因此无需身份验证。
- 未设置有效用户并使用全局值（默认值 = “空”）：所有经过身份验证的用户都可以访问此特殊服务。

作为“fred”访问“sambaserver”上的主目录：

```
$ smbclient //SAMBASERVER/fred 输入 fred 的密码:
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] smb:
\> volume

卷: |弗雷德|序列号 0xce0909dd smb: \>
```

smbstatus 的输出显示来自用户“fred”的会话：

\$ SMB状态 Samba 版本 4.1.12			
PID	用户名	团体	机器
24457	弗雷德	弗雷德	10.20.27.158 (ipv4:10.20.27.158:49017)
服务	进程号	机器	连接于
弗雷德	24457	10.20.27.158	2015 年 10 月 21 日星期三 09:36:34
没有锁定的文件			

示例：将远程用户“alice.jones”映射到 Linux 用户“alice”

添加到全局部分的参数：

```
[全局的]
...
用户名映射= /etc/samba/usermap.txt
...
```

/usr/local/samba/private/usermap.txt 的示例内容：

```
root = administraor 管理员
nobody = guest pcguest smbguest alice = alice.jones readonly =
glen fred terry sarah lachlan = Lachlan Smith

用户 = @sales
```

- 用户映射是全局设置。登录名（很可能是 Windows 帐户名）映射到本地（Linux）用户。

如果“alice.jones”尝试连接到相关的主目录，“alice.jones”将被映射到“alice”，用户将有权访问为“alice”启用的所有服务，“alice”的主目录将是服务而不是“alice.jones”。

从“sambaclient”连接到“sambaserver”作为“alice.jones”：

```
$ smbclient //SAMBASERVER/alice.jones 输入 alice.jones 的密码:
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12] smb: \> volume

卷: |爱丽丝|序列号 0x37da1047 smb: \>
```

“sambaserver”上显示活动连接的 smbstatus 输出不显示 “alice.jones” ,而只显示 “alice” :

```
$ SMB状态
Samba 版本 4.1.12
PID      用户名      团体      机器
23788    爱丽丝    爱丽丝    10.20.27.158 (ipv4:10.20.27.158:48988)

服务      进程号      机器      连接于
爱丽丝    23788 10.20.27.158 2015 年 10 月 21 日星期三 07:29:39

没有锁定的文件
```

示例:使 “windows”上的共享对 “sambaclient”上的用户可用

使用 smbclient 将文件复制到 “windows”上的 winshare1:

```
[fred@sambaclient ~]$ echo file from Fred > fred.txt [fred@sambaclient ~]$ smbclient //windows/winshare1
输入 fred 的密码:Domain=[WINDOWS] OS=[Windows 7 Professional 7601 Service Pack 1]服务器=[Windows 7 ←

专业版 6.1] smb: \> dir

.

..
desktop.ini 模式          DR          0 2015 年 10 月 27 日星期二 07:21:20
                           DR          0 2015 年 10 月 27 日星期二 07:21:20
                           AHS         46 2015 年 10 月 27 日星期二 07:16:15
                           ↑          0 2015 年 10 月 27 日星期二 07:21:20
                           ↑          0 2015 年 10 月 26 日星期一 09:22:17
                           ↑          1055 2015 年 10 月 26 日星期一 09:25:34

新建文本文件.txt
密码

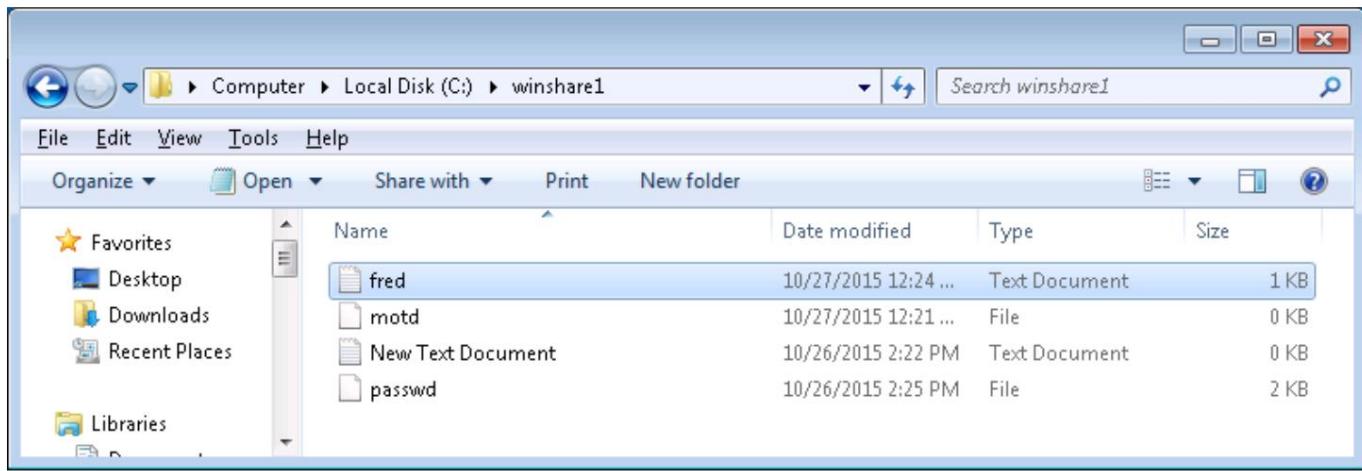
40551 个大小为 262144 的块。可用 3397 个块
smb: \> put fred.txt 将文件 fred.txt 作为
\fred.txt (14.6 kb/s) (平均 14.6 kb/s) smb: \> dir

.

..
桌面.ini          DR          0 2015 年 10 月 27 日星期二 07:23:58
弗雷德.txt        DR          0 2015 年 10 月 27 日星期二 07:23:58
模式              AHS         46 2015 年 10 月 27 日星期二 07:16:15
                  ↑          15 星期二 10 月 27 日 07:23:58 2015
                  ↑          0 2015 年 10 月 27 日星期二 07:21:20
                  ↑          0 2015 年 10 月 26 日星期一 09:22:17
                  ↑          1055 2015 年 10 月 26 日星期一 09:25:34

40551 个大小为 262144 的块。可用 3397 个块
```

在 “windows”上查看结果:



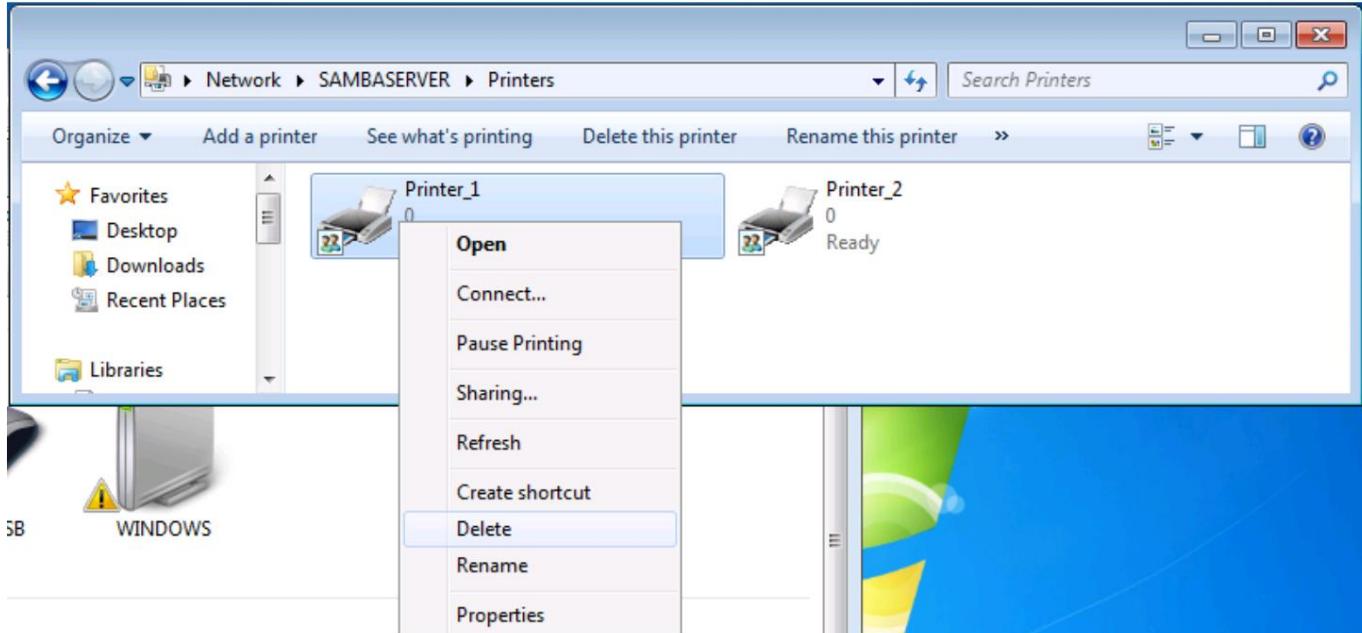
示例：允许所有人在“sambaserver”上的所有打印机上打印

插入或修改的配置部分以实现此示例：

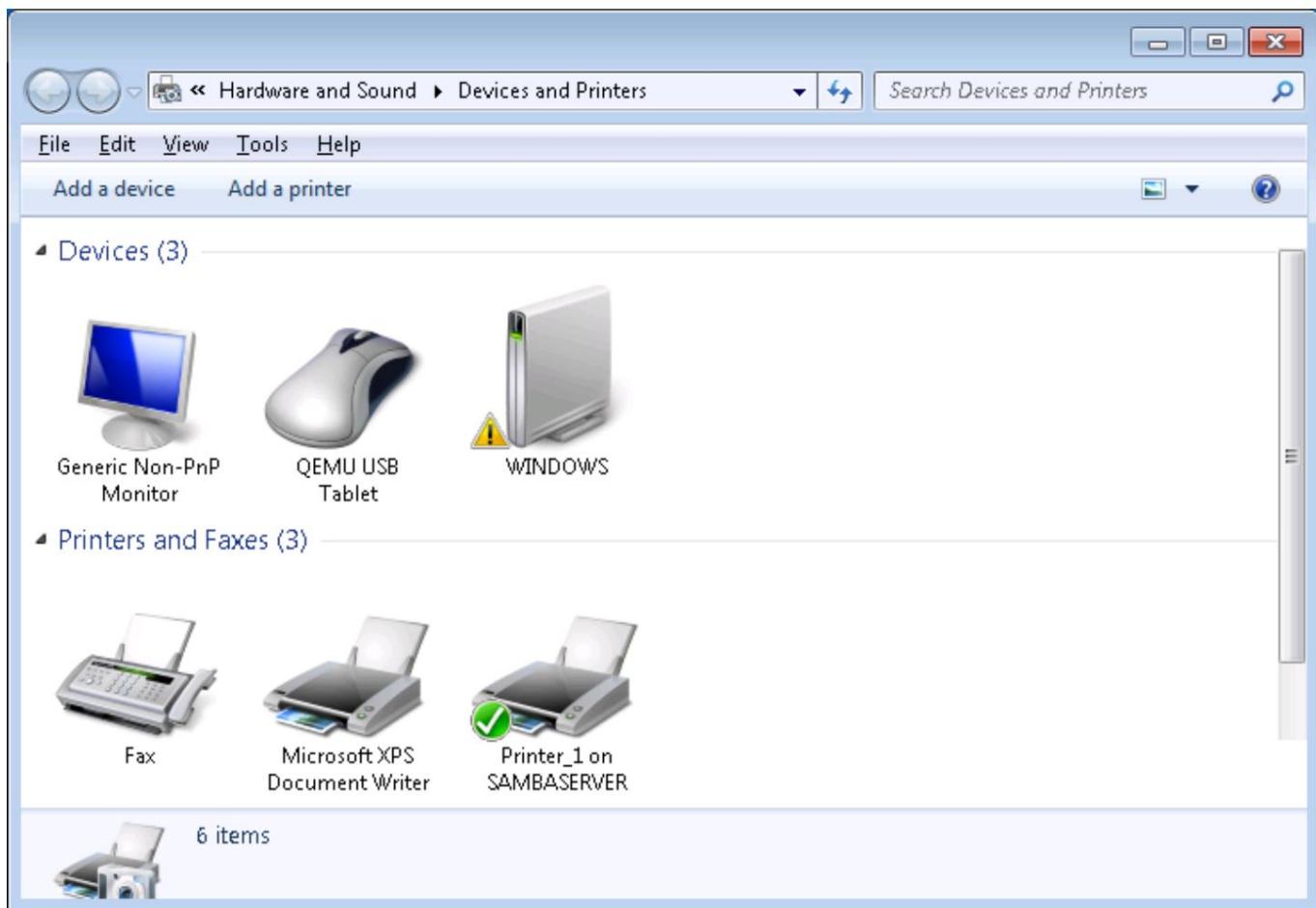
```
[printers]
comment = %L 路径上的打印机 %p = /var/
spool/samba printable = yes browseable =
yes guest ok = yes # valid users = #
```

- 添加了特殊部分[打印机]。
- 假脱机文件写入/var/spool/samba。
- 匹配此部分的服务（所有打印机）可打印。
- 该服务是可浏览的，因此可以通过连接到服务器来查找。
- 访客访问已启用，因此无需身份验证。

在 Windows 上使用 Windows 资源管理器浏览并连接到（启用）Printer\_1。



右键单击可以连接到打印机并将其添加为基于通用文本的打印机。



右键单击 Printer\_1 并打印测试页。在“sambaserver”上检查打印机的假脱机文件：

```

视窗
打印机测试页

恭喜！
如果您可以阅读此信息，则您已在 WINDOWS 上正确安装了 Generic / Text Only。

以下信息描述了您的打印机驱动程序和端口设置。
提交时间:11:45:31 AM 10/26/2015
计算机名称:WINDOWS
打印机名称:\\SAMBASERVER\Printer_1
打印机型号:通用/纯文本
颜色支持:否
端口名称:\\SAMBASERVER\Printer_1
数据格式: 生的
司机姓名: UNIDRV.DLL文件
数据文件: TTY.GPD
配置文件: 帮助文件: UNIDRVUI.DLL
驱动程序版本: 6.00 UNIDRV.HLP 文件

环境: Windows NT x86

此驱动程序使用的其他文件:C:
\\Windows\system32\spool\DRIVERS\W32X86\3\TTYRES.DLL
(6.1.7600.16385 (win7_rtm.090713-1255))
C:\\Windows\system32\spool\DRIVERS\W32X86\3\TTY.INI C:
\\Windows\system32\spool\DRIVERS\W32X86\3\TTY.DLL (6.1.7600.16385
(win7_rtm.090713-1255))
C:\\Windows\system32\spool\DRIVERS\W32X86\3\TTYUI.DLL

```

```
(6.1.7600.16385 (win7_rtm.090713-1255))
C:\Windows\system32\spool\DRIVERS\W32X86\3\TTYUI.HLP C:
\Windows\system32\spool\DRIVERS\W32X86\3\UNIRES.DLL (6.1.7600.16385 (win7_rtm.090713-1255))

C:\Windows\system32\spool\DRIVERS\W32X86\3\STDNAMES.GPD C:
\Windows\system32\spool\DRIVERS\W32X86\3\STDDTYPE.GDL C:
\Windows\system32\spool\DRIVERS\W32X86\3\STDSCHEM.GDL C:
\Windows\system32\spool\DRIVERS\W32X86\3\STDSCHMX.GDL
这是打印机测试页的结尾。
```

示例:禁止从“sambaclient”在“Printer\_1”上打印。

插入或修改的配置部分以实现此示例:

```
[打印机_1]
注释 = %L 上的打印机 1
路径 = /var/spool/samba 打印机名称 =
Printer_1 可打印 = 是可浏览 = 是访客正常 = 是主
机拒绝 = sambaclient
```

- 创建一个部分以明确匹配“Printer\_1”
- 使服务可打印将服务标识为打印机
- 假脱机文件写入/var/spool/samba
- 打印作业发送到本地打印机队列“Printer\_1”
- 该服务不可浏览,因此无法通过连接到服务器进行查找
- 访客访问已启用,因此无需身份验证。
- 明确拒绝“sambaclient”的访问。

在尝试匹配特殊部分[printers]之前,Samba 将首先将请求的服务与明确匹配服务名称的部分进行匹配。除“Printer\_1”之外的任何服务都不会匹配任何显式部分,并且会落入特殊部分[printers]。服务“Printer\_1”的请求将首先匹配[Printer\_1]部分,因此永远不会匹配特殊部分[printers]。

请注意,在 Samba 中,不可能配置类似“sambaclient 可以访问除 Printer\_1 之外的所有打印机”之类的内容。在这种情况下,我们需要将所有打印机配置为可供所有人访问,并为任何例外添加配置。

**使用 smbclient 测试来自 sambaclient 的打印**

```
$ smbclient //sambaserver/Printer_1 -c print /etc/hosts
输入 alice 的密码:Domain=[OURGROUP]
OS=[Unix] Server=[Samba 4.1.12] tree connect failed: NT_STATUS_ACCESS_DENIED
```

示例:列出“sambaserver”上的可用服务。

这个例子不需要额外的配置。

使用 smbclient 创建“sambaserver”列表。注意评论。

```
$ smbclient -L //SAMBASERVER 输入 fred 的密码:  
Domain=[OURGROUP] OS=[Unix] Server=[Samba 4.1.12]
```

共享名	类型	评论
公开分享1	磁盘	sambaserver 上的公共存储
	磁盘	在 samba 服务器上共享 1
分享2	磁盘	在 Samba 服务器上共享 2
打印机_1	Samba 服务器上的打印机 Printer 1	
IPC\$ 示 例)fred	工控机	IPC 服务 (用于 LPIC2 的 Linux Samba 服务器 sambaserver ← 来自 sambaclient 的 sambaserver 上 fred 的主目录
打印机_2	打印机	打印机 杯子打印机 Printer_2
域=[OURGROUP] 操作系统=[Unix] 服务器=[Samba 4.1.12]		
服务器		评论
桑巴客户端		桑巴 4.1.12
桑巴服务器		用于 LPIC2 示例的 Linux Samba 服务器 sambaserver
SS2		用于 LPIC2 示例的 Linux Samba 服务器 sambaserver
工作组	掌握	
我们的组		桑巴服务器

## 设置 nmbd WINS 服务器

什么是 WINS 服务器?

WINS 代表 Windows Internet 名称服务。这是一种名称服务,用于通过使用 TCP/IP 查询上的 NetBIOS 将 NetBIOS 名称转换为 IP 地址。这是使用 UDP 数据包完成的。

使用 Samba 作为 WINS 服务器

要告诉 Samba 它也应该扮演 WINS 服务器的角色,请将以下行添加到 Samba 配置文件 /etc/samba/smb.conf 的[global]部分:

```
[全局] 赢得支持  
= 是
```

请注意,网络上不应有多个 WINS 服务器,并且在启用“wins 支持”时,您不应设置任何其他 WINS 参数,例如“wins server”。

重新启动 smb 和 nmb 服务以获取更改后的配置

```
# 服务 smb 重启 # 服务 nmb 重启
```

## 为客户端创建登录脚本

登录脚本非常方便。例如,如果每个用户都需要将他的主目录自动映射到驱动器 H:,则登录脚本可以解决这个问题。然后,用户会看到一个额外的硬盘驱动器,这使您作为管理员可以在需要时自由地将主目录移动到另一台服务器。对于用户来说,它仍然是驱动器 H:,您所要做的就是更改登录脚本中的一行。

这同样适用于在特定用户登录或特定机器登录时应可访问或运行的打印机和进程。

批处理文件必须是 Windows 风格的批处理文件,因此在每行的末尾应该有一个回车符和一个换行符。

首先要做的是启用登录支持。这是通过将以下行添加到 Samba 配置文件 /etc/samba/smb.conf 的[global]部分来完成的：

```
[全局] 登录服  
务器 = 是
```

要做的第二件事是创建一个名为[netlogon]的共享,登录脚本将驻留在其中并且所有用户都可以读取：

```
[网络登录]  
注释 = Windows 客户端的 Netlogon 路径 = /home/netlogon 可浏览 = 否  
  
来宾 ok = 不可写 = 否
```

登录脚本的定义取决于您是希望每个用户还是每个客户端都有一个脚本。

基于用户名

将以下行添加到[netlogon]部分：

```
登录脚本 = %U.bat
```

并且,假设用户是 “fred” ,创建一个名为 /home/netlogon/fred.bat 的文件。

基于客户的名字

将以下行添加到[netlogon]部分：

```
登录脚本 = %m.bat
```

并且,假设机器名为 “workstation1” ,创建一个名为 /home/netlogon/workstation1.bat 的文件。

## 将 Samba 配置为域成员

要将 Samba4 配置为域成员,您需要确保在开始之前系统上没有任何配置。

加入域有两种选择。服务器可以是 Active Directory 域或较旧的 NT4 域的成员。因为 Active Directory 域使用 Kerberos 和 DNS,所以在加入域之前正确配置服务器很重要。

配置 DNS

要让服务器定位域,正确配置 DNS 设置很重要。一个 AD DC 有一个内置的 DNS 服务器,我们要连接的系统应该使用它。手动配置 ip 设置时,您应该将 AD 域控制器配置为 DNS 服务器。如何执行此操作取决于您使用的发行版。

如果配置正确,当 AD 域控制器的 ip 地址为 192.168.1.2 且域为 example.com 时, /etc/resolv.conf 文件应如下所示:

```
域名服务器 192.168.1.2  
搜索 example.com
```

当您将主机加入域时,Samba 会尝试在 AD DNS 区域中注册主机。为此,net 实用程序会尝试使用 DNS 或 /etc/hosts 中的正确条目来解析主机名。

使用 /etc/hosts 时,主机名或 FQDN 不解析为 127.0.0.1 很重要。因此,正确配置的主机文件如下所示,其中 server2.example.com 是我们添加为域成员的服务器的主机名:

```
127.0.0.1 本地主机 localhost.localdomain  
192.168.1.3 server2.example.com server2
```

要检查分辨率是否正确,您可以使用 getent 命令,如下所示:

```
$ getent 主机 server2 192.168.1.3  
server2.example.com server2
```

## 配置 Kerberos

目前 Samba 使用 Heimdal Kerberos。这意味着 Kerberos 文件 /etc krb5.conf 只需要包含以下信息:

```
[libdefaults]  
default_realm = EXAMPLE.COM  
dns_lookup_realm = 假 dns_lookup_kdc = 真
```

使用上述以外的任何内容都可能导致错误。

您需要将 EXAMPLE.COM 替换为您的 Kerberos 领域。

Kerberos 需要所有域成员的同步时间。建议设置 NTP 客户端。

## 配置桑巴

只有在加入 Active Directory 域时才需要执行前面的步骤。Active Directory 域和 NT4 域都需要执行以下步骤。

### 设置 smb.conf 文件

下一步是配置域成员 smb.conf 文件。该文件通常位于 /etc/smb/smb.conf 或 /etc/samba/smb.conf。如果没有,您可以使用以下命令找到该文件:

```
$ smbd -b | grep 配置文件配置文件: /usr/local/  
samba/etc/smb.conf
```

现在我们知道文件所在的位置,我们可以添加以下配置:

```
[全局] 安全 =  
ADS 工作组 = EXAMPLE 领  
域 = EXAMPLE.COM  
  
日志文件 = /var/log/samba/%m.log 日志级别 = 1  
  
# 本地内置帐户的默认 ID 映射配置 # 和域成员上的组。默认 (*) 域:# - 不得与任何域 ID 映射配置重叠! # - 必须使用支持读写的  
后端,例如 tdb。 idmap 配置 *:后端 = tdb idmap 配置 *:范围 = 3000-7999
```

## 加入域

现在我们已经配置了 samba, 是时候加入域了。如上所述, 不支持使用 samba-tool 实用程序来执行此操作。

要加入域, 您可以使用以下命令。输出将取决于您加入的域的类型。

加入 Active Directory 域时:

```
$ net ads join -U administrator 输入管理员密码 : 使用短域名 - 示  
例加入 “server2” 到 dns 域 “example.com”
```

加入 NT4 域时:

```
$ net ads join -U administrator 输入管理员密码 : 加入域示例。
```

## 配置名称服务开关 (NSS)

为了使域用户和组对本地系统可用, 我们必须将 winbind 条目附加到 /etc/nsswitch.conf 中的以下数据库:

```
密码:文件 winbind 组:文件 winbind
```

## 启动服务

现在我们可以启动服务了。如果您只需要 Samba 查找域用户和组, 您只需启动 winbind 服务。如果您还设置了文件和打印机共享, 您还需要启动 smbd 和 nmbd 服务。

```
$ systemctl 启动 winbind smbd nmbd
```

您不应该启动 samba 服务。此服务仅在 Active Directory 域控制器上需要,

## 测试 winbind 连接

要验证 winbind 服务是否能够连接到 Active Directory 域控制器或 NT4 域控制器, 您可以使用 wbinfo 命令:

```
$ wbinfo --ping-dc 检查 NETLOGON  
域 [EXAMPLE] dc 连接到 “server1.example.com”←。  
成功了
```

## 配置 NFS 服务器 (209.2)

资源和进一步阅读: [NFS](#)、[NFSv4](#)、[NFSv4.2](#)、[Zadok01](#)。

考生应该能够使用 NFS 导出文件系统。此目标包括访问限制、在客户端上安装 NFS 文件系统和保护 NFS。

## 关键知识领域

- NFS 版本 3 配置文件
- NFS 工具和实用程序
- 对某些主机和/或子网的访问限制
- 服务器和客户端上的挂载选项
- TCP 包装器
- 对 NFSv4 的认识

## 条款和实用程序

- /etc/出口
- exportfs
- showmount
- nfsstat
- /proc/mounts
- /etc/fstab
- rpc 信息
- 安装

· 端口映射器

## NFS 网络文件系统

缩写 NFS 扩展为网络文件系统。使用 NFS，您可以使远程磁盘（或其中的一部分）成为本地文件系统的一部分。

NFS 协议正在调整，到目前为止，这个过程已经花费了数年时间。这对那些使用 NFS 的人有影响。

现代 NFS 守护程序当前将在内核空间（运行内核的一部分）中运行，并支持 NFS 协议的版本 3（仍将支持版本 2 以与旧客户端兼容）。运行在用户空间（几乎独立于内核）并且只接受协议版本 2 NFS 请求的较旧的 NFS 守护进程仍然存在。本节将主要描述支持协议版本 3 和兼容客户端的内核空间 NFS 服务器。将在适当的时候指出与旧版本的差异。

---

### 注意有

关于 NFS 正在进行的工作的详细信息，请参见下面的第 9.2.8 节。

---

客户端、服务器或两者？

使文件系统对其他系统可用的系统称为服务器。连接到服务器的系统称为客户端。每个系统都可以配置为服务器、客户端或两者。

## 设置 NFS

本节介绍 NFS 相关软件及其配置。

## NFS 的要求

要运行 NFS,需要以下内容:

- 必须将对 NFS 的支持 (多个选项)内置于内核中
- 端口映射器必须正在运行
- 在支持 NFS 服务器的系统上,NFS 守护进程和安装守护进程必须处于活动状态
- 可能需要支持守护进程

下面将详细讨论每一点。

## 为 NFS 配置内核

为 NFS 配置内核时,必须决定系统是客户端还是服务器。具有包含 NFS 服务器支持的内核的系统也可以用作 NFS 客户端。

---

### 笔记

此处描述的情况适用于 2.4.x 内核系列。此处描述的规格将来可能会发生变化。

---

NFS 相关的内核选项 请参阅 NFS 的内核选项。

NFS 文件系统支持 (CONFIG\_NFS\_FS) :如果你想使用NFS作为客户端,选择这个。如果这是唯一选择的 NFS 选项,系统将仅支持 NFS 协议版本 2。要使用协议版本 3,您还需要选择 CONFIG\_NFS\_V3。选择 CONFIG\_NFS\_FS 时,还支持老式的用户空间 NFS 服务器 (协议版本 2)。当系统只是一个内核空间 NFS 服务器 (即,既不是客户端也不是用户空间 NFS 服务器)时,您可以不使用此选项。

提供 NFSv3 客户端支持(CONFIG\_NFS\_V3):如果客户端系统应该能够与 NFS 版本 3 服务器建立 NFS 连接,请选择此项。仅当还选择了 NFS 支持 (CONFIG\_NFS\_FS) 时才能选择此项。

NFS 服务器支持(CONFIG\_NFSD):仅内核空间。选择此选项时,您将获得支持 NFS 协议版本 2 的内核空间 NFS 服务器。需要额外的软件来控制内核空间 NFS 服务器 (稍后将显示)。

要运行老式的用户空间 NFS 服务器,不需要此选项。请改为选择 CONFIG\_NFS。

提供 NFSv3 服务器支持(CONFIG\_NFSD\_V3):此选项将对 NFS 协议版本 3 的支持添加到内核空间 NFS 服务器。内核空间 NFS 服务器将支持 NFS 协议的第 2 版和第 3 版。如果您还选择了 NFS 服务器支持 (CONFIG\_NFSD),则只能选择此项。

在编译器构建期间进行配置时 (即 make menuconfig、make xconfig 等),上面列出的选项可以在文件系统部分的网络文件系统小节中找到。

NFS 的[内核选项](#)概述了内核中的 NFS 支持。

选择至少一个 NFS 内核选项会自动打开 Sun RPC (远程过程调用)支持。这导致内核空间 RPC 输入/输出守护进程。它可以在进程列表中被识别为 [rpciod]。

## 端口映射器

端口映射器用于将 TCP/IP 连接连接到适当的 RPC 调用。所有 NFS 流量都需要它,它不仅将 (传入的)TCP 连接映射到 NFS (RPC) 调用,还可以用于将不同的 NFS 版本映射到运行 NFS 的不同端口。如果安装了 NFS 软件 (内核除外),大多数发行版都会安装端口映射器。<sup>1</sup>

不需要配置端口映射器本身。然而,端口映射器的安全性是一个问题:强烈建议您限制对端口映射器的访问。这可以使用 tcp 包装器来完成。

<sup>1</sup> 严格来说,您可以在客户端系统上运行没有端口映射器的 NFS,但是,连接会很慢并且 (甚至更)不可靠。

描述	选项)	允许/提供 允许 NFS
NFS 文件系统支持	配置_NFS_FS	(v2) 客户端和用户空间 NFS (v2) 服务器 允许 NFS (v2 + v3) 客户端
NFSv3 客户端支持	CONFIG_NFS_FS 和 配置_NFS_V3	
NFS 服务器支持	配置_NFSD	提供 NFS (v2) 内核服务器 提供 NFS (v2
NFSv3 服务器支持	CONFIG_NFSD 和 配置_NFSD_V3	+ v3) 内核服务器

表 9.1:NFS 的内核选项

### 保护端口映射器

首先,确保 portmapper 支持内置的 tcp wrapper (因为它不是通过 inetc 启动的, portmapper 需要它自己的内置 tcpwrapper 支持)。您可以通过运行 ldd /sbin/portmap 2 来测试它。结果可能类似于

```
libwrap.so.0 => /lib/libwrap.so.0 (0x40018000) libnsl.so.1 => /lib/libnsl.so.1
(0x40020000) libc.so.6 => /lib/libc.so.6 (0x40036000) /lib/ld-linux.so.2 => /lib/ld-
linux.so.2 (0x40000000)
```

带有 libwrap.so.0 的行 (libwrap 属于 tcp 包装器) 表明此端口映射器是使用 tcp-wrapper 支持编译的。如果缺少该行,请获得更好的端口映射器或自己编译一个。

一个常见的安全策略是默认阻止传入的端口映射器请求,但允许特定主机连接。该策略将在此处进行描述。

首先编辑文件 /etc/hosts.deny 并添加以下行:

端口映射:全部

这会拒绝每个系统访问端口映射器。它可以用命令扩展:

```
portmap: ALL: (echo illegal rpc request from %h | mail root) &
```

现在所有端口映射器请求都被拒绝了。在第二个示例中,请求被拒绝并报告给 root。

下一步是只允许那些被允许这样做的系统访问。这是通过在 /etc/hosts 中放置一行来完成的。允许:

端口映射:121.122.123。

这允许每个具有以显示的数字开头的 IP 地址的主机连接到端口映射器,因此使用 NFS。

另一种可能性是指定主机名的一部分:

端口映射: .example.com

这允许 example.com 域内的所有主机进行连接。允许 NIS 工作组的所有主机:

端口映射:@workstations

允许子网中具有 IP 地址的主机:

端口映射:192.168.24.16/255.255.255.248

这允许从 192.168.24.16 到 192.168.24.23 的所有主机进行连接 (来自 Zadok01 的示例)。

2端口映射器的位置可能不同。

## 端口映射和rpcbind

一些 Linux 发行版使用端口映射。其他 linux 发行版使用 rpcbind。

portmap 守护进程被 rpcbind 取代。 Rpcbind 具有更多功能,如 ipv6 支持和 nfs4 支持。

## 通用 NFS 守护进程

### nfs -utils包

NFS 是作为一组守护进程实现的。这些可以通过它们的名字来识别:它们都以 rpc 开头。前缀后跟守护进程的名称。其中包括:rpc.nfsd (仅是具有内核 NFS 服务器的系统中的支持程序) .rpc.mountd、.rpc.lockd 和 rpc.statd。

这些守护进程的源代码可以在 nfs-utils 包中找到 (有关 nfs-utils 的更多信息,请参阅[NFS](#))。它还将包含其他支持程序的源代码,例如 exportfs、showmount 和 nfsstat。这些将在稍后的[导出 NFS 和测试 NFS 中讨论](#)。

发行版可能会提供 nfs-utils 作为即用型软件包,有时名称不同。例如,Debian 在一个特殊的 nfs-common 包中提供锁定和状态守护进程,在 nfs-\*server 包 (有用户空间和内核空间版本)中提供 NFS 和挂载守护进程。

这里提到的每个守护进程也可以使用 tcp 包装器来保护。详情见第[9.2.6 节](#)。

## NFS 服务器软件

### NFS 守护进程

实现 NFS 服务器时,您可以安装对内核空间或用户空间 NFS 服务器的支持,具体取决于内核配置。 rpc.nfsd 命令 (有时称为 nfsd)是用户空间中完整的 NFS 服务器。而在内核空间中,它只是一个支持程序,可以在内核中启动NFS服务器。

### 内核空间或用户空间NFS服务器

内核空间 NFS 服务器 内核空间 NFS 服务器是运行内核的一部分。内核 NFS 服务器显示为 [nfsd] 在进程列表中。

内核中支持NFS服务器的rpc.nfsd版本只是一个控制NFS内核服务器的支持程序。

用户空间 NFS 守护进程 rpc.nfsd 程序还可以包含老式的用户空间 NFS 服务器 (仅限版本 2)。

用户空间 NFS 服务器是一个完整的 NFS 服务器。在进程列表中可以识别为rpc.nfsd。

### 挂载守护进程

mountd (或 rpc.mountd)挂载守护进程处理传入的 NFS (挂载)请求。它在提供 NFS 服务器支持的系统上是必需的。

挂载守护进程的配置包括将文件系统导出 (使其可用)到某些主机并指定它们如何使用该文件系统。

使用 /etc/exports 文件和 exportfs 命令导出文件系统将在[导出 NFS 中讨论](#)。

### 锁守护进程

NFS 的锁守护进程在 rpc.lockd 中实现。

使用现代 (2.4.x) 内核时,您不需要锁定守护程序支持。这些内核内部提供了一个,可以在进程列表中识别为[lockd]。由于内部内核锁守护进程优先,意外启动 rpc.lockd 不会有任何坏处。

rpc.lockd 没有配置。

### 状态守护进程

根据手册页,状态守护进程 rpc.statd 仅实现重启通知服务。它是一个用户空间守护进程 即使在支持内核空间 NFS 版本 3 的系统上也是如此。在进程列表中可以识别为rpc.statd。

它用于支持 NFS 客户端和 NFS 服务器的系统。

rpc.statd 没有配置。

### 导出文件系统

导出文件系统或其中的一部分,使其可供另一个系统使用。文件系统可以导出到单个主机、一组主机或每个人。

导出定义在文件 /etc/exports 中配置,并将由 exportfs 命令激活。可以使用命令 showmount --exports 查询当前的导出列表。

---

### 笔记

在下面的示例中,名为nfsshop的系统将成为 NFS 服务器,而名为clientN的系统将成为客户端之一。

---

### 文件/etc/exports

文件 /etc/exports 包含要导出的文件系统的定义、允许访问它的主机的名称以及主机访问它的方式。

/etc/exports 中的每一行都具有以下格式:

```
1 /dir 2 hostname( 3 options) 4 ...
```

#### 要导出的目录名称

允许访问 /dir (导出目录) 的系统 (主机) 的名称。如果省略系统名称,则所有主机都可以连接。指定系统名称有五种可能性:

单个主机名 允许连接的主机的名称。可以是名称 (clientN) 或 IP 地址。通配符 允许一组系统。 \*.exam 将允许 example.com 域中的所有系统  
ple.com 通配符。

IP 网络 IP 编号或地址/子网掩码组合的范围。 nothing 将系统部分留空主要是意外造成的 (请参阅下面的注意事项)。它允许所有主机连接。

为防止此错误,请确保系统名称和启动选项的左大括号之间没有空格。

@NISgroup NIS 工作组可以指定为以 @ 开头的名称。

大括号之间的选项。将进一步讨论。

可以列出一个以上的带选项的系统:

```
/home/ftp/pub clientN(rw) *.example.com(ro)
```

解释:允许系统clientN在/home/ftp/pub中读写。 example.com 中的系统允许连接,但只能读取。

**警告**

确保主机名和大括号之间的规范之间没有空格（甚至不是白色）。之间有很大的区别



/home/ftp/pub clientN(rw)

和

/home/ftp/pub clientN (rw)

第一个允许clientN读写访问。第二个允许使用默认选项的clientN访问（参见man 5 exports）和所有系统读写访问！

**导出选项**

/etc(exports 中可以使用几个导出选项。这里只讨论最重要的。有关完整列表,请参阅 exports(5) 手册页。此处将列出两种类型的选项:常规选项和用户/组 ID 选项。

**常规选项**

ro (默认)客户端只有读取权限。

rw客户端具有读写权限。当然,客户端也可以选择以只读方式挂载。

同样相关的是 NFS 跨系统处理用户和组权限的方式。 NFS 软件将具有相同 UID 和相同用户名的用户视为同一用户。 GID 也是如此。

用户根是不同的。因为 root 可以读取 (和写入)所有内容<sup>3</sup>,所以对 NFS 的 root 权限被认为是危险的。

对此的解决方案称为压缩:所有请求都以用户 nobody (实际上 UID 65534,通常称为 -2)和组 nobody (GID 65534)完成。

至少有四个选项与压缩相关:root\_squash、no\_root\_squash、all\_squash 和 no\_all\_squash。  
每一个都会被详细讨论。

**用户/组 ID 压缩**

**root\_squash** (默认)用户 root 在 clientN (客户端)上的所有请求都将作为 nfsshop (服务器)上的用户 nobody 完成。这意味着,例如,客户端上的用户 root 只能读取服务器上全局可读的文件。

**no\_root\_squash**客户端上的所有请求都将以服务器上的根用户身份完成。

例如,当要通过 NFS 进行备份时,这是必需的。

这意味着 nfsshop 上的 root 完全信任 clientN 上的用户 root。

**all\_squash** clientN 上除 root 以外的任何用户的请求都以 nfsshop 上的用户 nobody 执行。

如果您无法轻松映射用户名和 UID,请使用此选项。

**no\_all\_squash** (默认) clientN 上非 root 用户的所有请求都作为 nfsshop 上的同一用户尝试。

系统 nfsshop (服务器系统)上 /etc(exports 中的示例条目:

```
/      client5(ro,no_root_squash) *.example.com(ro)
```

系统 nfsshop 允许系统 client5 对所有内容进行只读访问,用户 root 的读取是在 nfsshop 上以 root 身份完成的。

来自 example.com 域的系统被允许只读访问,但是来自 root 的请求是以用户 nobody 完成的,因为默认情况下 root\_squash 为真。

这是一个示例文件:

<sup>3</sup> 嗯,在大多数情况下

```
# /etc/exports on nfsshop # 可以导出的文件
系统的访问控制列表 # 到 NFS 客户端。请参阅导出 (5)。
```

```
/ client2.exworks(ro,root_squash) /
client3.exworks(ro,root_squash) /
client4.exworks(ro,root_squash) /home
client9.exworks(ro,root_squash)
```

解释:client2、client3 和 client4 被允许挂载完整的文件系统 (/:root)。但是他们具有只读访问权限,并且请求是作为用户 nobody 完成的。主机 client9 只允许以与其他三台主机相同的权限挂载 /home 目录。

### exportfs命令

一旦 /etc/exports 配置完成,就可以使用 exportfs 命令激活其中的导出列表。它还可用于在更改后重新加载列表或停用导出列表。[Exportfs 和 fstab](#)展示了 exportfs 的一些功能。

命令 exportfs	说明重新导出所
-r exportfs -a	有目录导出或取消导出所有目
exportfs -ua	录取消激活导出列表 (全部取消导出)

表 9.2:exportfs 概览

#### 注意

较旧的 (用户空间)NFS 系统可能没有exportfs命令。在这些系统上,导出列表将在安装守护程序启动时自动安装。通过向正在运行的 mount-daemon 进程发送SIGHUP信号来完成更改后的重新加载。

### 激活导出列表

使用以下命令激活 (或重新激活)导出列表:

```
exportfs -r
```

r 源自再出口一词。

在发出 exportfs -r 之前,没有文件系统被导出,也没有其他系统可以连接。

当导出列表被激活时,内核导出表将被填充。以下命令将显示内核导出表:

```
cat /proc/fs/nfs(exports
```

输出将类似于:

```
# Version 1.1 # Path
Client(Flags) # IPs /
client4.exworks(ro,root_squash,async,wdelay) # 192.168.72.4 /home client9.exworks(ro,root_squash,async,wdelay)
# 192.168.72.9 / client2.exworks (ro,root_squash,async,wdelay) # 192.168.72.2 /
client3.exworks(ro,root_squash,async,wdelay) # 192.168.72.3
```

说明:允许所有命名主机使用列出的选项挂载本机的根目录 (client9: /home)。列出 IP 地址是为了方便起见。

在对正在运行的系统上的 /etc/exports 进行更改后,还可以使用 exportfs -r。



## 警告当运行

exportfs -r 时,一些事情将在目录/var/lib/nfs 中完成。那里的文件很容易被人为干预破坏,并产生深远的影响。

## 停用导出列表

使用以下命令取消导出所有活动的导出条目：

```
exportfs -ua
```

字母 ua 是 unexport all 的缩写。

在 exportfs -ua 之后,没有出口活动了。

## showmount命令

showmount 显示有关导出的文件系统和活动挂载到主机的信息。表9.3显示了如何使用 show mount。

命令	说明
showmount --exports	显示活动
showmount showmount --	显示具有活动安装的客户端的名称
directories showmount --all	显示远程客户端安装的目录
	显示客户
	端名称和目录

表 9.3:showmount 概览

showmount 接受一个主机名作为它的最后一个参数。如果存在,showmount 将查询该主机上的 NFS 服务器。如果省略,将查询当前主机（如以下示例中,当前主机称为 nfsshop）。

使用 --exports 选项 showmount 列出当前活动的导出列表：

```
# showmount --exports nfsshop 导出
列表：/
client2.exworks,client3.exworks,client4.exworks /home client9.exworks
```

与前面显示的 cat /proc/fs/nfs/exports 的输出相比,信息更加稀疏。

如果没有选项,showmount 将显示当前连接到系统的主机的名称：

```
# showmount
Hosts on nfsshop:
client9.exworks
```

使用 --directories 选项 showmount 将显示当前由远程主机挂载的目录的名称：

```
# showmount --目录
nfsshop 上的目录：/home
```

使用 --all 选项,showmount 命令会列出远程客户端（主机）和挂载的目录：

```
# showmount --all
nfsshop 上的所有挂载点：client9.exworks:/home
```

## NFS 客户端:软件和配置

NFS 客户端系统是使用 `mount` 命令进行挂载尝试的系统。挂载需要支持内置的 NFS。通常是这样的。

NFS 客户端系统需要在内核中具有适当的 NFS 支持,如前所示 (请参阅[配置 NFS](#))。接下来,它需要一个正在运行的端口映射器。最后,需要软件来执行远程安装尝试:`mount` 命令。

### 笔记

本段假设您熟悉`mount`命令和文件/`/etc/fstab`。如有疑问,请查阅相应的手册页。

通常用于通过 NFS 挂载远程文件系统的 `mount` 命令:

```
mount -t nfs remote:/那里 /这里
```

这指定了远程服务器remote上的文件系统/there。

像往常一样在客户端上的挂载点/here。

示例:要将服务器系统 nfsshop 上的 /usr 文件系统挂载到本地挂载点 /usr,请使用:

```
mount -t nfs nfsshop:/usr /usr
```

挂载请求的微调是通过选项完成的。

```
mount -t nfs  -o opts remote:/there /here
```

在 -o 选项选择器之后可能有几个选项。这些选项会影响挂载尝试或活动 NFS 连接。

### NFS的挂载选项

`ro`与`rw`如果指定了 `ro`,则远程 NFS 文件系统将以只读方式挂载。使用 `rw` 选项远程文件系统 `tem` 将可用于读取和写入 (如果 NFS 服务器同意)。

#### 提

示NFS 服务器端(`/etc/exports`)的默认值是`ro`,但客户端(`mount -t nfs`)的默认值是`rw`。服务器设置优先,因此挂载将以只读方式完成。

#### 提

示-o `ro`也可以写成`-r`; -o `rw`也可以写成`-w`。

`rsize=nnn`和`wsize=nnn` `rsize` 选项指定读取传输 (从服务器到客户端)的大小。`wsize` 选项指定相反的方向。数字越大,数据在可靠网络上的传输速度就越快。在需要多次重试的网络上,传输可能会变慢。

默认值为 1024 或 4096,具体取决于您的内核版本。当前内核最多接受 8192。基于 tcp 的 NFS 版本 3,在您阅读本文时可能已准备好生产,允许最大大小为 32768。此大小在文件 `include/linux/nfsd` 中用 `NFSSVC_MAXBLKSIZE` 定义/`const.h` 在内核源代码存档中找到。

udp和tcp指定 NFS 连接的传输层协议。大多数 NFS 版本 2 实现仅支持 udp,但确实存在 tcp 实现。 NFS 版本 3 将允许 udp 和 tcp (后者正在积极开发中)。未来版本 4 将只允许 tcp。请参阅第9.2.8 节。

nfsvers=n指定用于传输的 NFS 版本 (参见第9.2.8 节)。默认情况下,现代版本的 mount 将使用版本 3。仍然使用版本 2 的旧实现可能很多。

retry=n重试选项指定在放弃之前继续重试挂载尝试的分钟数。默认值为 10000 分钟。

timeo=n timeo 选项指定挂载尝试超时的时间。超时值以分秒 (十分之一秒)为单位指定。默认值为 7 分秒 (0.7 秒)。

hard (默认)与soft这些选项控制系统尝试的难度。

hard系统会无限期地尝试。 soft系统将尝试直到发生 RPC (端口映射器)超时。

intr与nointr (默认)使用这些选项可以控制是否允许用户中断挂载试图。

intr如果指定了 intr,则用户可以中断挂载尝试。 nointr如果设置了 nointr,则用户不能中断挂载尝试。如果重试具有默认值 (10000 分钟),装载请求似乎会挂起数天。

fg (默认)和bg这些选项控制后台安装工具。它默认关闭。

bg这会打开后台挂载:客户端首先尝试在前台挂载。所有重试都发生在后面地面。

fg所有尝试都发生在前台。

背景安装也受其他选项的影响。指定 intr 时,挂载尝试将被 RPC 超时中断。例如,当远程主机关闭或端口映射器未运行时,就会发生这种情况。在测试设置中,后台仅在发生“连接被拒绝”时才完成。

可以使用逗号组合选项:

```
mount -t nfs -o ro,rsize=8192 nfsshop:/usr/share /usr/local/share
```

首选的选项组合可能是 :hard,intr 和 bg。挂载将无限期地尝试,在后台重试,但仍然可以被启动挂载的用户中断。

其他要考虑的挂载选项是 noatime,noauto、nosuid 甚至 noexec。请参见 man 1 mount 和 man 5 nfs。

当然,所有这些选项也可以在 /etc/fstab 中指定。如果文件系统不应在引导时自动挂载,请务必指定 noauto 选项。用户选项将允许非根用户执行挂载。这不是默认值。/etc/fstab 中的示例条目:

```
nfsshop:/home /homesOnShop nfs ro,noauto,user 0
```

0

现在每个用户都可以做

```
挂载/homesOnShop
```

您还可以使用自动挂载程序来挂载和卸载远程文件系统。然而,这些都超出了这个目标的范围。

## 测试 NFS

NFS 设置完成后,就可以对其进行测试了。以下工具可以提供帮助:showmount、rpcinfo 和 nfsstat。

`showmount --exports`命令

如`showmount`中所示，`showmount --exports`命令列出服务器系统的当前导出。这可以用作创建的 NFS 系统健康状况的快速指示。尽管如此，还有更复杂的方法可以做到这一点。

/proc/mounts 文件

要查看挂载了哪些文件系统,请检查 /proc/mounts。它还将显示 nfs 挂载的文件系统。

```
$ cat /proc/挂载
```

## rpc信息

`rpcinfo` 命令报告 RPC 信息。这可用于探测本地或远程系统上的端口映射器或发送伪请求。

## rpcinfo:探测系统

rpcinfo -p命令列出端口映射器知道的所有已注册服务。每个 rpc... 程序在启动时向端口映射器注册自己,因此显示的名称对应于真正的守护进程（或内核等效项,如 NFS 版本 3 的情况）。

可以在服务器系统 nfsshop 上使用它来查看端口映射器是否正常运行：

程序与原型端口 100003  
3 udp 2049 nfs

此输出选择表明此端口映射器将接受 udp 上的 nfs 版本 3 的连接。

服务器系统上rpcinfo -p>的完整示例输出：

从上面的清单中可以看出，端口映射器将接受 NFS 协议第 2 版和第 3 版的 RPC 请求，两者都在 udp 上。

注意

可以看出,每个 RPC 服务都有自己的版本号。例如, mountd 服务支持 udp 和 tcp 上 mountd 版本 1.2 或 3 的传入连接。

通过在 -p 之后指定服务器系统的名称,也可以从客户端系统探测 nfsshop (服务器系统) :

```
rpcinfo -p nfsshop
```

当然,如果一切顺利,输出将是相同的。

rpcinfo:发出空请求

可以在不做任何实际工作的情况下测试连接:

```
rpcinfo -u远程主机程序
```

这类似于测试网络连接的 ping 命令。然而, rpcinfo -u 就像一个真正的 rpc/nfs 连接一样工作,发送一个所谓的空伪请求。 -u 选项强制 rpcinfo 使用 udp 传输。在 nfsshop 上测试的结果:

```
rpcinfo -u nfsshop nfs program 100003 version
2 ready and waiting program 100003 version 3 ready and waiting
```

-t 选项将对 tcp 传输执行相同的操作:

```
rpcinfo -t nfsshop nfs rpcinfo: RPC: Program
not registered 程序 100003 不可用
```

这个系统显然在 udp 上支持 nfs,但在 tcp 上不支持。

#### 注意在

示例输出中,数字 100003 代替名称 nfs 或与名称 nfs 一起使用。名称或号码可以在其他地方使用。也就是说,我们也可以这样写:

```
rpcinfo -u nfsshop 100003
```

#### nfsstat 命令

nfsstat 列出有关 nfs 连接的统计信息 (即计数器)。这可以用来查看是否发生了某些事情,也可以确保没有发生任何事情。

表9.4概述了 nfsstat 的相关选项。

	rpc	网络文件系统	两个都
服务器	-sr	-sn	-s
客户双	-cr	-cn	-c
方	-r	-n	-nr

表 9.4:nfsstat 程序的一些选项

服务器主机 nfsshop 上 nfsstat -sn 的示例输出:

```
服务器 nfs v2:
无效的          获取属性      设定值      根      查找 0% 41      阅读链接
          0% 3          0% 0      0% 0      0% 0          0% 0          0%
读            缓存          写          创造      消除      改名
          5595 99% 0      0% 0      0% 1      0% 0          0% 0          0%
关联          符号链接      目录          目录      读目录      统计数据
```

0	0% 0	0% 0	0% 0	0% 7	0% 2	0%
<b>服务器 nfs v3:</b>						
无效的	获取属性	设定值	查找 0% 0	使用权	阅读链接	
100% 0	0% 0		0% 0	0% 0	0% 0	0%
读	写	创造	目录	符号链接	点头	
0	0% 0	0% 0	0% 0	0% 0	0% 0	0%
消除	目录	改名	关联	读目录	读读加号	
0	0% 0	0% 0	0% 0	0% 0	0% 0	0%
统计数据	信息系统	路径配置提交				
0	0% 0	0% 0	0% 0	0%		

两个空标题下的 1 是前面显示的 `rpcinfo -u nfsshop nfs` 命令的结果。

## 保护 NFS

NFS 安全有几个不相关的问题。首先,NFS 协议和实现有一些已知的弱点。NFS 文件句柄是应该随机的数字,但实际上不是。这打开了通过猜测文件句柄建立连接的可能性。另一个问题是所有 NFS 数据传输都是按原样完成的。这意味着任何能够监听连接的人都可以窃听信息(这称为嗅探)。错误的挂载点名称加上人为错误可能会带来完全不同的安全风险。

### 限制访问

可以通过将对每个 NFS 服务器的访问限制为一组包含受信任用户的已知受信任主机来防止嗅探和不需要的连接请求:例如,在一个小型工作组中。TCP-Wrapper 支持或防火墙软件可用于限制对 NFS 服务器的访问。

tcp 包装器早些时候(参见[安全端口映射器](#))展示了如何限制从特定主机到端口映射器的连接。对于与 NFS 相关的守护进程,即 `rpc.mountd` 和 `rpc.statd`,也可以这样做。如果您的系统运行老式的用户空间 NFS 服务器(即,在进程列表中有 `rpc.nfsd`),请考虑保护 `rpc.nfsd` 和可能的 `rpc.lockd`。

另一方面,如果您的系统正在运行现代的基于内核的 NFS 实现(即,在进程列表中有 `[nfsd]`),您不能这样做,因为 `rpc.nfsd` 程序不是接受连接的程序。确保 TCP-Wrapper 支持内置到您希望保护的每个守护进程中。

防火墙软件 TCP-Wrapper 支持的问题是在连接请求被拒绝时主机内部已经存在连接。如果 TCP-Wrapper 库(不太可能)或包含支持的守护程序中存在与安全相关的错误,则可能会授予不需要的访问权限。或者更糟。防火墙软件(例如 iptables)可以在连接进入主机之前使内核阻止连接。您可以考虑在每个 NFS 服务器主机或网络入口点阻止不需要的 NFS 连接到所有可接受的主机。至少阻止端口映射器端口(111/udp 和 111/tcp)。

此外,考虑阻止 2049/udp 和 2049/tcp(NFS 连接)。您可能还想阻止其他端口,例如 `rpcinfo -p` 命令显示的端口:例如,挂载守护程序端口 32771/udp 和 32768/tcp。[第 12 章](#) 详细介绍了如何设置防火墙。

### 防止人为错误

简单的人为错误加上错误的命名也可能导致安全风险。您不会是第一个删除远程目录树的人,因为挂载点不易被识别,而且远程系统挂载时具有读写权限。

以只读方式挂载 以只读方式挂载远程文件系统可以防止意外删除。因此,尽可能以只读方式挂载。如果确实需要挂载一个可读写的部分,那么可写(擦除)的部分越小越好。

设计好您的安装点 此外,命名一个安装点,以便它可以很容易地被识别为一个安装点。一种可能性是使用特殊名称:

/挂载点/nfsshop

## 最佳 NFS 版本

NFS 软件已取得进展。尽管没有软件可以防止人为错误,但可以使用更好的软件来防止其他风险 (例如,可猜测的文件句柄和嗅探)。

### 注意

NFS 版本 4 是 NFS 协议的新版本,旨在修复 NFS 中的所有现有问题。在撰写本文时 (2014 年 5 月)4.0 和 4.1 版本已经发布; 4.2 版正在开发中。有关 NFS 版本 4 的更多信息以及早期版本之间的差异,请参阅第9.2.8 节。

可猜测的文件句柄 破解 NFS 服务器的方法之一是猜测所谓的文件句柄。旧的 (32 位)文件句柄 (在 NFS 版本 2 中使用)很容易猜到。NFS 协议的第 3 版通过使用更难猜测的 64 位文件句柄提高了安全性。

第 4 版安全增强 NFS 协议的第 4 版定义了加密连接。当连接被加密时,通过嗅探获取信息变得更加困难甚至不可能。

## NFS 组件概述

表9.5概述了与 NFS 相关的最重要的文件和软件。

程序或文件 内核 端	说明提供 NFS
口映射器 rpc.nfsd	支持处理 RPC 请求
rpc.mountd 文件 /	
etc/exports exports	NFS 服务器控制 (内核空间)或软件 (用户空间)处理传入的 (卸载)挂载请求 定义导出哪些文件系统远程文件系统卸载所有远程文件系统
命令 showmount --	
exports rpcinfo 命令 nfsstat 命	
令 showmount --all mount -t	
nfs remote:/there /here	
umount -t nfs -一个	

表 9.5:NFS 相关程序和文件概览

## NFS 协议版本

目前,NFS 协议中有很多变化会影响系统的设置方式。表9.6提供了一个概述。

协议版本	当前状态从未发布	内核或用户空间	udp 或 tcp 传输
1 <sup>st</sup>	成为过时的新标准		
2 <sup>nd</sup>		用户 ,内核内核	udp,一些tcp实现。存在udp、
3 <sup>rd</sup>			tcp :正在开发中的tcp
4 <sup>th</sup>	新标准	核心	

表 9.6:NFS 协议版本概述

表9.6中可以看出的趋势是:内核空间而不是用户空间,tcp 而不是 udp。

关于传输协议的注释 tcp 连接 (NFS v3,v4、某些 v2)被认为优于 udp 连接 (NFS v2,v3) 。 udp 选项可能是小型、快速网络上的最佳选择。但是 tcp 允许设置相当大的数据包大小 (rsize、wsize) 。据报道,对于 64k 的大小,tcp 连接比通过 udp 的连接快 10%,这不允许那么大的大小。此外,与 udp 相比,tcp 在设计上是一种更可靠的协议。有关此的讨论,请参见[Zadok01](#)。

## NFSv4

NFS 版本 4 (NFSv4) 与其前身相比提供了一些新功能。 NFSv4 不是导出多个文件系统,而是为每个客户端导出一个伪文件系统。这个伪文件系统的来源可能来自不同的文件系统,但这对客户端来说是透明的。

NFSv4 提供一组扩展属性,包括对 MS Windows ACL 的支持。尽管 NFSv4 与以前版本的 NFS 相比提供了增强的安全功能,并且自 2003 年以来一直存在,但它从未被广泛采用。鼓励用户实施已于 2010 年 1 月批准的NFSv4.1。有关不同 NFS 版本及其特性的详细信息,请参阅[NFSv4.2](#)。

## 问题和解答

### 文件共享

1. 通过什么方式查看smb配置?

这可以通过使用 `testparm` 来完成。[测试参数](#)

2. `root_squash` 有什么作用?

所有在客户端root用户发出的请求都将在服务器端以nobody用户身份执行。[根南瓜\[278\]](#)

3.为什么一定要确保/etc/中主机名和大括号之间的用户权限之间没有空格  
出口?

因为它允许所有系统访问大括号之间的用户权限。[空间\[278\]](#)

4.samba配置在哪个文件?

Samba 通过 `/etc/samba/smbd.conf` [Samba 配置文件](#)进行配置

5.挂载选项soft有什么作用?

系统将尝试执行挂载,直到发生 RPC (端口映射器)超时。[柔软的](#)

6. NFS 版本 2 中的安全问题是什么?

猜测文件句柄和嗅探数据是可能的。[保护 NFS \[285\]](#)

## 第10章

### 网络客户端管理 (210)

本主题总权重为11分,包含以下4个目标:

目标 210.1; DHCP 配置 (2 分) 考生应该能够配置 DHCP 服务器。这个目标包括设置默认和每个客户端选项,添加静态主机和 BOOTP 主机。还包括配置 DHCP 中继代理和维护 DHCP 服务器。

目标 210.2; PAM 身份验证 (3 分) 考生应该能够配置 PAM 以支持身份验证  
使用各种可用的方法。

目标 210.3; LDAP 客户端使用 (2 分) 考生应该能够对 LDAP 服务器执行查询和更新。  
还包括导入和添加项目,以及添加和管理用户。

目标 210.4; 配置 OpenLDAP 服务器 (4 分) 考生应该能够配置基本的 OpenLDAP 服务器,包括 LDIF 格式和基本访问控制的知识。包括对 SSSD 在身份验证和身份管理中的作用的理解。

#### DHCP 配置 (210.1)

考生应该能够配置 DHCP 服务器。此目标包括设置默认值和每个客户端选项,添加静态主机和 BOOTP 主机。还包括配置 DHCP 中继代理和维护 DHCP 服务器。

##### 关键知识领域

DHCP 配置文件、术语和实用程序

子网和动态分配的范围设置

##### 条款和实用程序

- dhcpcd.conf
- dhcpcd.leases
- /var/log/daemon.log 和 /var/log/messages
- arp
- DHCPCD
- 收音机
- radvd.conf

## 什么是 DHCP?

“动态主机配置协议”(DHCP)是一种允许计算机从网络中获取网络配置信息的协议。地址在一段时间内从服务器“租用”给客户端。有一个用于分配 IPv6 地址的单独协议,称为 DHCPv6,尽管“邻居发现协议”(NDP)更适合此目的。

请求和分配地址的过程如下:

- 当计算机启动时,它向网络发送请求。
- 任何接收此请求的 DHCP 服务器决定分配给客户端的地址和其他配置选项。这通常基于以下内容:请求到达哪个网络,或发送请求的接口的 MAC 地址。
- 每个服务器发送一个数据包,提供给客户端分配地址。
- 客户端决定接受哪个提议,并向服务器发送消息以确认选择。
- 服务器确认它已经记录了这个地址。

最常用的配置项包括:IP 地址、主机名、域名、子网掩码、广播地址、路由器和域名服务器。

该信息由 DHCP 客户端请求并由 DHCP 服务器提供。默认情况下,服务器在 udp 端口 67 上侦听请求并通过 udp 端口 68 进行应答,但可以使用 -p 选项告诉它侦听另一个端口。然后 DHCP 服务器将通过一个比它侦听的端口大一的 udp 端口进行应答。

使用 IPv6 的主机实际上能够使用无状态自动配置为自己分配链路本地 IP 地址。DHCPv6 或 NDP 可用于分配额外的全局唯一地址和其他配置参数。NDP 在 radvd 部分有更详细的描述

DHCP 的网站[资源](#)包含许多关于 DHCP 协议的(指向)信息,包括 RFC。

## 服务器是如何配置的?

DHCP 服务器 dhcpcd 的配置是通过其配置文件 /etc/dhcpcd.conf 完成的。

可以在配置文件中使用的元素有:(全局)参数、共享网络、子网、组和主机。

### 什么是(全局)参数?

参数可以看作是被赋值并从服务器传递到客户端的变量。有些参数以 option 关键字开头,有些则没有。不以 option 关键字开头的参数要么是控制 DHCP 服务器行为的参数,要么是 DHCP 协议中的可选参数。

“普通”参数和“全局”参数之间的区别纯粹在于参数的范围。例如,如果 DNS 始终相同,则向每个网络定义语句添加域名服务器参数定义语句是没有意义的。通过在配置文件的开头为 domain-name-servers 参数分配一个值,该参数成为全局参数,其值成为该参数的默认值。

可以通过在后续部分中为其分配另一个值来覆盖全局参数的值。

### 什么是共享网络声明?

如果同一物理网络上有多个子网,则使用共享网络声明。对于属于共享网络的所有子网相同的参数可以在包含那些子网声明的共享网络声明中的子网声明之上定义一次。

什么是子网声明?

子网声明用于定义网段。仅适用于相关子网的参数在子网声明中定义。

子网声明必须包含一个范围声明,该声明定义了 DHCP 服务器可以为子网上的客户端提供的 IP 地址。

什么是团体声明?

组声明用于将其他声明 (包括组声明) 分组,这些声明具有许多共同的属性,因此只需指定一次共同属性,而不是为每个声明指定一次。

什么是主机声明?

主机声明用于为特定客户端设置属性。客户端通过其唯一属性之一 (例如其 NIC 地址或客户端标识符) 向 DHCP 服务器标识自己。

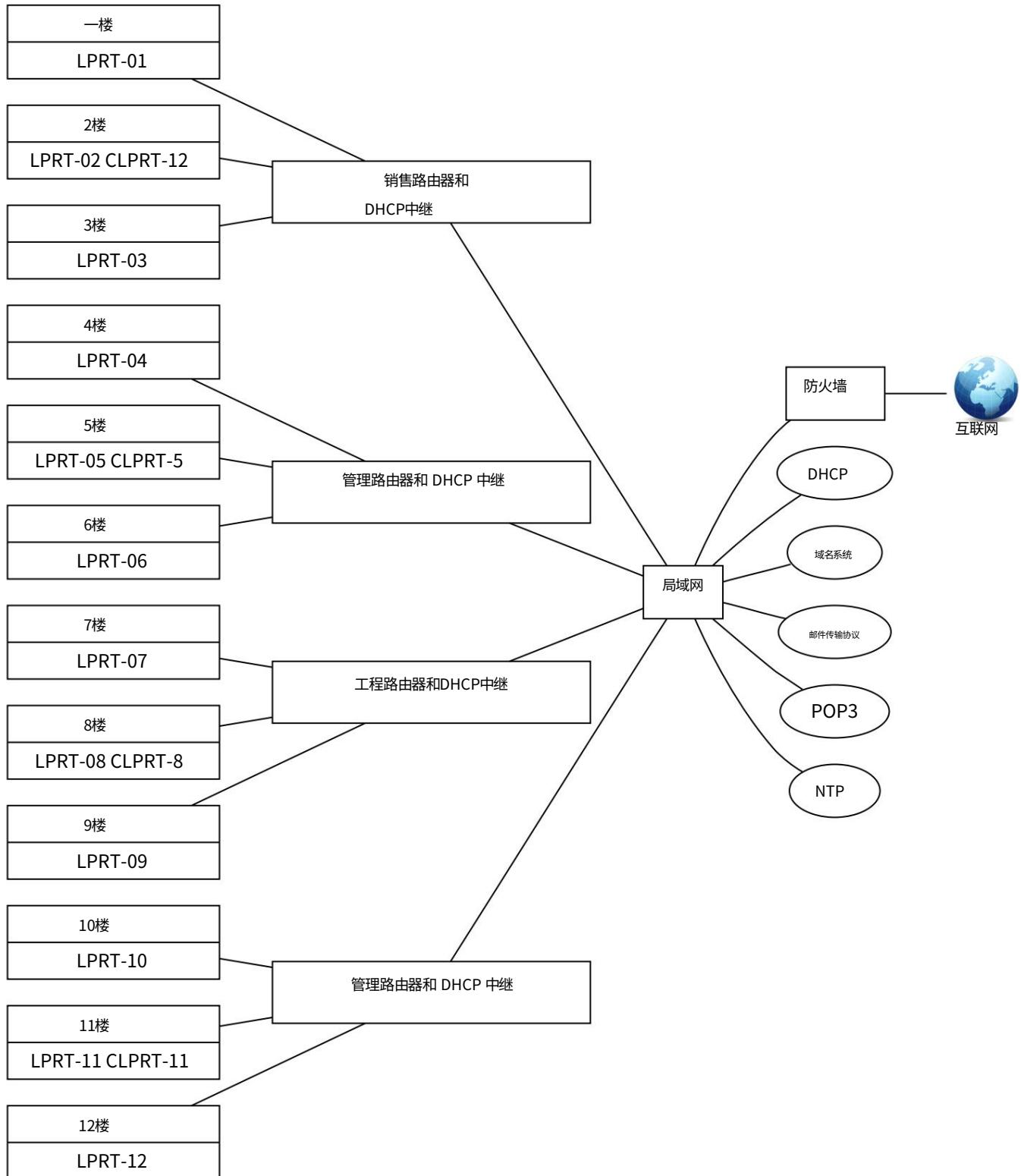
## 一个例子

考虑一家有四个部门的公司:销售、行政、工程和管理。所有部门都位于同一栋楼,每个部门都有三层楼可供使用。

每层楼有多达 200 个工作站和一台激光打印机 (LPRT-xx)。此外,每个部门在中间楼层都有自己的彩色激光打印机 (CLPRT-xx)。打印机只能由打印机所属部门的用户使用。

所有用户都从公司的 DHCP 服务器获得 IP 地址,并且必须能够访问公司的 DNS 服务器和 NTP 服务器。所有用户都使用 POP3 协议接收邮件,使用 SMTP 协议发送邮件,使用 NNTP 协议阅读新闻。

公司网络的图形表示如下所示:



公司网络的网络布局。

#### 网络架构

假设 IP 范围 21.31.xx 已分配给公司并且每个部门都有自己的网络（由第三个八位字节的最高四位决定；换句话说：使用的网络掩码是 /20 或 255.255.240.0），可以设置子网。

如下：

部门	楼层	IP range	路由器	描述
0001	0001	21.31.17.0 - 21.31.17.255	21.31.17.1	销售楼层#1
0001	0010	21.31.18.0 - 21.31.18.255		销售楼层 #2
0001	0011	21.31.19.0 - 21.31.19.255		销售楼层 #3
0010	0100	21.31.36.0 - 21.31.36.255	21.31.36.1	管理 #4
0010	0101	21.31.37.0 - 21.31.37.255		管理 #5
0010	0110	21.31.38.0 - 21.31.38.255		管理#6
0011	0111	21.31.55.0 - 21.31.55.255	21.31.55.1	工程楼层 #7
0011	1000	21.31.56.0 - 21.31.56.255		工程楼层 #8
0011	1001	21.31.57.0 - 21.31.57.255		工程楼层#9
0100	1010	21.31.74.0 - 21.31.74.255	21.31.74.1	管理楼层#10
0100	1011	21.31.75.0 - 21.31.75.255		管理楼层#11
0100	1100	21.31.76.0 - 26.351.7		管理楼层#12

表 10.1:前两个八位字节是 21.31

#### 工作站可用的网络服务

公司网络上的工作站通过部门的 DHCP 中继（也用作路由器）从公司的 DHCP 服务器获取其 IP 地址和可用网络服务的 IP 地址。

#### 独立于子网的服务

与子网无关的服务是公司网络上所有工作站都可用的服务，无论它们位于哪个子网上。下表显示了这些服务及其固定 IP 地址。

服务	描述	IP 地址	主机名
DHCP	公司的DHCP服务器	21.31.0.1	dhcp.company.com
域名系统	公司的DNS	21.31.0.2	dns.company.com
邮件传输协议	公司的SMTP服务器	21.31.0.3	smtp.company.com
POP3	公司的POP3服务器	21.31.0.4	pop3.company.com
消息	公司的NNTP-server	21.31.0.5	news.company.com
NTP	公司的NTP服务器	21.31.0.6	ntp.company.com

表 10.2:公司范围内的服务

#### 子网相关服务

依赖于子网的服务是仅对与该服务位于同一子网上的工作站可用的服务。下表显示了这些服务及其固定 IP 地址。

#### 构建 DHCP 服务器的配置文件

在设计网络拓扑时，已经在前面的部分中收集了能够构建配置文件所需的信息。

在此部分中，实际配置文件 /etc/dhcpd.conf 将填充必要的信息。

部门	服务	描述	IP地址	名称
销售量	路由器	销售路由器楼层 #2	21.31.17.1	rtr-02.company.com
	打印机	激光打印机楼层 #1	21.31.17.2	lppt-01.company.com
	打印机	激光打印机楼层 #2	21.31.18.2	lppt-02.company.com
	打印机	激光打印机楼层 #3	21.31.19.2	lppt-03.company.com
	打印机	彩色激光打印机楼层 #2	21.31.18.3	clppt-02.company.com
行政	路由器	管理路由器层 #5	21.31.36.1	rtr-05.company.com
	打印机	4 楼激光打印机	21.31.36.2	lppt-04.company.com
	打印机	5 楼激光打印机	21.31.37.2	lppt-05.company.com
	打印机	6 楼激光打印机	21.31.38.2	lppt-06.company.com
	打印机	彩色激光打印机楼层 #5	21.31.31.37.3	clppt-05.company.com
工程	路由器	工程路由器楼层 #8	21.31.31.55.1	rtr-08.company.com
	打印机	7 楼激光打印机		lppt-07.company.com
	打印机	8 楼激光打印机		lppt-08.company.com
	打印机	激光打印机楼层 #9		lppt-09.company.com
	打印机	彩色激光打印机楼层 #8		clppt-08.company.com
管理	路由器	管理路由器楼层 #11		rtr-11.company.com
	打印机	10 楼激光打印机		lppt-10.company.com
	打印机	11 楼激光打印机		lppt-11.company.com
	打印机	激光打印机楼层 #12		lppt-12.company.com
	打印机	彩色激光打印机楼层 #11	21.31.31.31.31.31.31.31.31.31.31.31.31.57.2	clppt-11.company.com

表 10.3:子网相关服务

服务的全局参数

全局参数放在配置文件的顶部：

```
# 域名解析  
选项域名服务器 21.31.0.2; # 邮件传输协议  
  
选项 smtp 服务器 21.31.0.3; #POP3  
  
选项弹出服务器 21.31.0.4; # 消息  
  
选项 nntp 服务器 21.31.0.5; #NTP  
  
选项时间服务器 21.31.0.6;
```

另一种方法是使用域名。单个域名必须解析为单个 IP 地址。使用域名，您可以将以下条目放入配置文件中：

```
# DNS 选  
项域名服务器 dns.company.com; # SMTP 选项 smtp-server  
smtp_company_com; # POP3 选项 pop_server.pop3_company_com; # 消息
```

选项 nntp-server news.company.com; #NTP  
选项时间服务器 ntp.company.com;

#### 公司的共享网络和子网

如前几节所述,有四种不同的网络,每个部门一个,还有十二个不同的IP地址范围,每个楼层一个。此外,每个网络都有自己的路由器和打印机。

这转化为四个共享网络，每个共享网络都有自己的网络掩码和广播地址，并包含三个 IP 地址范围。

网络掩码是一个 IP 地址，用于确定工作站或其他使用 IP 地址的网络设备所在的网络。网络掩码在该网络中所有网络设备都相同的位位置为 1，在其他位置为 0。由于一个部门的共享网络上的所有子网都在同一个物理网络上，所以区别是在共享网络级别上进行的，而不是在楼层级别上。楼层已编码为 IP 地址（第三个八位字节的低半字节），以准备明年计划安装的每层一台路由器，而不是每个部门一台路由器。网络掩码计算如下：

21.31.16.0 - :	0001 0101   0001 1111   0001 0000   0000 0000   销售量
21.31.31.255 :	0001 0101   0001 1111   0001 1111   1111 1111   网络
21.31.32.0 - :	0001 0101   0001 1111   0010 0000   0000 0000   行政
21.31.47.255 :	0001 0101   0001 1111   0010 1111   1111 1111   网络
21.31.48.0 - :	0001 0101   0001 1111   0011 0000   0000 0000   工程
21.31.63.255 :	0001 0101   0001 1111   0011 1111   1111 1111   网络
21.31.64.0 - :	0001 0101   0001 1111   0100 0000   0000 0000   管理
21.31.79.255 :	0001 0101   0001 1111   0100 1111   1111 1111   网络
固定位：	1111 1111   1111 1111   1111 0000   0000 0000   网络掩码 0
	255                    255                    240

使用网络掩码 255.255.240.0，可以确定 IP 地址所在的网络。这是通过将 IP 地址与网络掩码进行 AND 运算来完成的。要确定 IP 地址为 21.31.57.105 的工作站位于四个网络中的哪一个，执行以下计算：

21.31.57.105 :	0001 0101   0001 1111   0011 1001   0110 1001   IP地址
255.255.240.0:	1111 1111   1111 1111   1111 0000   0000 0000   网络掩码
21.31.48.0:	0001 0101   0001 1111   0011 0000   0000 0000   给予网络

IP 地址 21.31.57.105 在 21.31.48.0 网络上，这是工程网络。

广播地址用于将数据包发送到网络上的每个工作站。广播地址因网络而异，可以通过将所有为主机地址保留/使用的位（如子网掩码所示）替换为 1 来确定。

确定广播地址的另一种方法是取网络掩码的倒数，在本例中为 0.0.15.255，然后将结果与网络地址进行或运算：

21.31.16.0 - :	0001 0101   0001 1111   0001 0000   0000 0000   销售量
0.0.15.255 :	0000 0000   0000 0000   0000 1111   1111 1111   或 INV 网络掩码
21.31.31.255 :	0001 0101   0001 1111   0001 1111   1111 1111   给予BCAST
21.31.32.0 - :	0001 0101   0001 1111   0010 0000   0000 0000   行政
0.0.15.255 :	0000 0000   0000 0000   0000 1111   1111 1111   或 INV 网络掩码
21.31.47.255 :	0001 0101   0001 1111   0010 1111   1111 1111   给予BCAST
21.31.48.0 - :	0001 0101   0001 1111   0011 0000   0000 0000   工程
0.0.15.255 :	0000 0000   0000 0000   0000 1111   1111 1111   或 INV 网络掩码
21.31.63.255 :	0001 0101   0001 1111   0011 1111   1111 1111   给予BCAST
21.31.64.0 - :	0001 0101   0001 1111   0100 0000   0000 0000   管理
0.0.15.255 :	0000 0000   0000 0000   0000 1111   1111 1111   或 INV 网络掩码
21.31.79.255 :	0001 0101   0001 1111   0100 1111   1111 1111   给予BCAST

IP 地址所在网络的广播地址可以通过将 IP 地址与反向网络掩码进行 OR 运算来确定。

对于 IP 地址为 21.31.57.105 的工作站，广播地址可以计算如下：

21.31.57.105 :	0001 0101   0001 1111   0011 1001   0110 1001   IP地址
0.0.15.255 :	0000 0000   0000 0000   0000 1111   1111 1111   或 INV 网络掩码
21.31.63.255 :	0001 0101   0001 1111   0011 1111   1111 1111   给予BCAST

IP 地址 21.31.57.105 属于广播地址为 21.31.63.255 的网络,这是正确的,因为 IP 地址在工程网络上。

要告诉 DHCP 服务器要为每个子网分配哪些 IP 地址,必须向子网添加范围语句。在这个例子中,每一层的 IP 地址 21.31.x.0 到 21.31.x.10 和 21.31.x.211 到 21.31.x.255 是为打印机和路由器保留的。这意味着对于每个子网,范围语句是:

范围 21.31.x.11 21.31.x.210

其中“x”取决于部门和楼层。

为实现此结构,将以下行添加到配置文件中:

```
# The Sales network, floors 1-3 shared-network sales-
net { # Sales-net 特定参数 option routers 21.31.17.1;选项
    lpr 服务器 21.31.17.2.21.31.18.2.21.31.19.2.21.31.18.3;选
    项广播地址 21.31.31.255;子网 21.31.17.0 网络掩码
    255.255.240.0 {

        #Floor#1具体参数范围21.31.17.11 21.31.17.210;

        } 子网 21.31.18.0 网络掩码 255.255.240.0 {
            # Floor #2 具体参数范围 21.31.18.11 21.31.18.210;
        }

        子网 21.31.19.0 网络掩码 255.255.240.0 {
            # Floor #3 具体参数范围 21.31.19.11 21.31.19.210;
        }

    }

# 管理网络,4-6 层 shared-network administration-net {

    # Administration-net 特定参数 option routers 21.31.36.1;选项 lpr 服务器
    21.31.36.2.21.31.37.2.21.31.38.2.21.31.37.3;选项广播地址 21.31.47.255;子
    网 21.31.36.0 网络掩码 255.255.240.0 {

        #Floor#4具体参数范围21.31.36.11 21.31.36.210;

        } 子网 21.31.37.0 网络掩码 255.255.240.0 {
            #Floor#5具体参数范围21.31.37.11 21.31.37.210;
        }

        } 子网 21.31.38.0 网络掩码 255.255.240.0 {
            #Floor#6具体参数范围21.31.38.11 21.31.38.210;
        }

    }

# The Engineering 网络,7-9 层 shared-network engineering-net {

    # Engineering-net 特定参数 option routers 21.31.55.1;选项 lpr 服务
    器 21.31.55.2.21.31.56.2.21.31.57.2.21.31.56.3;选项广播地址
    21.31.63.255;子网 21.31.55.0 网络掩码 255.255.240.0 {

        #7楼具体参数范围21.31.55.11 21.31.55.210;

    }

}
```

```
子网 21.31.56.0 网络掩码 255.255.240.0 {  
    #Floor#8具体参数范围21.31.56.11 21.31.56.210;  
  
    } 子网 21.31.57.0 网络掩码 255.255.240.0 {  
        #Floor#9具体参数范围21.31.57.11 21.31.57.210;  
    }  
  
    }  
  
    # 管理网络,10-12 层 shared-network management-net {  
  
        # management-net 特定参数 option routers 21.31.74.1;选项 lpr 服  
        务器 21.31.74.2,21.31.75.2,21.31.76.2,21.31.75.3;选项广播地址  
        21.31.79.255; subnet 21.31.74.0 netmask 255.255.240.0 { # Floor #10 具体参数范围 21.31.74.11 21.31.74.210;  
  
    } subnet 21.31.75.0 netmask 255.255.240.0 { # Floor #11 具体参数范围  
        21.31.75.11 21.31.75.210;  
  
    } subnet 21.31.76.0 netmask 255.255.240.0 { # Floor #12 具体参数范围  
        21.31.76.11 21.31.76.210;  
    }  
}
```

## 静态主机

静态主机是始终从 DHCP 服务器获取相同 IP 地址的主机,与动态主机相反,动态主机从一系列 IP 地址获取 IP 地址。

显然,DHCP 服务器必须能够识别主机才能断定主机已在 DHCP 服务器的配置文件中定义为静态主机。这可以通过使用 dhcp-client-identifier 选项或使用硬件以太网选项来完成。

dhcp-client-identifier 由客户端 (主机)发送到服务器,并且必须唯一标识该客户端。这是不安全的,因为无法确定没有第二个客户端使用相同的标识符。

硬件以太网选项导致匹配在全球唯一的客户端 NIC 地址上完成。

如果客户端不发送 dhcp-client-identifier,则使用 NIC-address 来标识客户端。

有两位设计师在工程部门工作,他们有时会来办公室拿一份他们设计的彩色硬拷贝。这些设计师被称为“luke”和“leah”,他们带着笔记本电脑并将它们连接到工程网络。他们机器的主机名将是“luke”和“leah”。

为此,管理员已将以下行添加到 DHCP 服务器的配置文件中:

```
group { # 适  
    用于所有静态主机的选项 option routers 21.31.55.1;选项 lpr 服务器 21.31.56.3;选项  
    广播地址 21.31.63.255;网络掩码 255.255.240.0; host luke { # 特定于 luke 硬件以  
    太网 AA:88:54:72:7F:92;固定地址 21.31.55.211;
```

```

        选项主机名 “luke” ;
    }

host leah { # 特定
    于 leah 硬件以太网
    CC:88:54:72:84:4F;固定地址 21.31.55.212;选项主机名 “leah” ;

}

```

### 静态 BOOTP 主机

这是一个特殊的静态主机。如果 luke 和 leah 的笔记本电脑是 BOOTP 客户端，则管理员可以将以下行添加到 DHCP 服务器的配置文件中：

```

group { # 适
    用于所有静态主机的选项 option routers 21.31.55.1;选项 lpr 服务器 21.31.56.3;
    选项广播地址 21.31.63.255;网络掩码 255.255.240.0;

host luke { # 特定
    于 luke 文件名 lukes-boot-file ;
    服务器名称 “发送给卢克的服务器名称” ; next-
    server <加载启动文件的服务器地址>;硬件以太网 AA:88:54:72:7F:92;固定地址
    21.31.55.211;选项主机名 “luke” ;

}

host leah { # 特定
    于 leah 文件名 leahs-boot-
    file ;server-name “要发送给 leah 的服务器名
    称” ; next-server <加载启动文件的服务器地址>;硬件以太网 CC:88:54:72:84:4F;
    固定地址 21.31.55.212;选项主机名 “leah” ;

}

```

filename 选项说明要从 next-server 选项中定义的服务器获取的文件的名称。如果省略下一个服务器，则从中获取文件的服务器是 DHCP 服务器。server-name 可用于将客户端从中引导的服务器的名称发送到客户端。

有关有效选项的信息，请参阅 [dhcp-options 手册页 \(man dhcp-options\)](#) 和 [dhcpcd.conf 手册页 \(man dhcpcd.conf\)](#)。

## 控制 DHCP 服务器的行为

### 租约

租约是客户端可以使用它从 DHCP 服务器获得的 IP 地址的时间量。客户端必须定期刷新租约，因为如果租约到期，IP 地址可以提供给另一个客户端。通常，如果租约在到期前刷新，客户端将获得相同的 IP 地址。

选项 max-lease-time 用于指定在客户端请求特定到期时间时将分配给租约的最长时间（以秒为单位）。

选项 default-lease-time 用于指定在客户端不要求特定到期时间的情况下将分配给租约的时间量（以秒为单位）。

DHCP 服务器在文件 /var/dhcp/dhcpd.leases 中保存它发布的租约的数据库。如果这个文件是空的，这可能是由于您在 DHCP 服务器的配置文件中只定义了静态主机而没有使用任何范围语句。在 DHCP 客户端上，租用的 IP 地址保存在文件 dhclient.leases 中。

#### DHCP 服务器侦听的接口

除非您另外指定，否则 dhcpd 将在所有接口上侦听 dhcp 请求。例如，如果您只想为 eth0 上的请求提供服务，您可以通过在启动守护程序的命令行中包含参数来将此告诉守护程序。

#### 更改后重新加载 DHCP 服务器

这是按如下方式完成的：

```
# /etc/init.d/dhcp 重启
```

这将停止正在运行的守护进程，等待两秒钟，然后启动一个新的守护进程，这会导致再次读取 /etc/dhcpd.conf。

#### 记录

默认情况下，DHCP 服务器使用 syslogd 记录日志，尽管许多 Linux 发行版已经将 syslogd 替换为 Systemd 的日志。使用 log-facility 关键字在 dhcpd.conf 文件中配置日志记录。一旦读取了 dhcpd.conf 文件，此语句会导致 DHCP 服务器在指定的日志工具上执行所有日志记录。默认情况下，DHCP 服务器记录到守护程序设施。可能的日志工具包括 auth、authpriv、cron、daemon、ftp、kern、lpr、mail、mark、news、ntp、security、syslog、user、uucp 和 local0 到 local7。并非所有这些设施在所有系统上都可用，在其他系统上可能还有其他设施可用。除了设置 log-facility 值之外，您可能需要修改 syslog.conf 文件以配置 DHCP 服务器的日志记录。例如，您可以添加这样一行：

```
local7.debug /var/log/dhcpd.log
```

syslog.conf 文件的语法在某些操作系统上可能有所不同 - 请参阅 syslog.conf 手册页以确保确定。

要让 syslog 开始记录到新文件，您必须首先创建具有正确所有权和权限的文件（通常，与 /var/log/messages 或 /usr/adm/messages 文件相同的所有者和权限应该没问题）并向 syslogd 发送 SIGHUP。

请注意，journalctl 不会记录到纯文本文件；它改用二进制格式。要查看特定于 dhcpcd 的消息，您必须使用 journalctl 命令过滤掉这些消息。

## DHCP 中继

### 什么是 DHCP 中继？

在我们前面的示例中，整个网络有一个 DHCP 服务器，并且在客户端和该服务器之间有路由器。

如果客户端能够通过路由器连接到 DHCP 服务器，则 DHCP 服务器将看不到客户端的 NIC 地址，而是路由器的 NIC 地址。这意味着静态主机无法通过其硬件地址识别。

诸如 dhcrelay 之类的 DHCP 中继代理提供了一种将 DHCP 和 BOOTP 请求从其中一个子网中继到公司的 DHCP 服务器的方法。

如果您需要更多信息，请参阅[互联网联盟 DHCP 主页](#)。

DHCP 中继代理侦听 DHCP 和 BOOTP 查询和响应。当从客户端收到查询时,dhcrelay 将其转发到命令行中指定的 DHCP 服务器列表。当从服务器收到回复时,它会在原始请求来自的网络上进行广播或单播（根据中继代理的能力或客户端的请求）。如果在命令行上没有指定接口名称,dhcrelay 将识别所有网络接口,如果可能,消除非广播接口,并尝试配置每个接口。

请查阅手册页 (`man dhcrelay`) 了解更多详细信息。

### 在 IPv6 网络中分配地址

前面的段落主要关注 IPv4 网络中的自动 IP 分配。使用 IPv6 的网络通常使用“邻居发现协议”来获取对网络有效的 IP 地址。在 Linux 中,此协议由“路由器通告”(radvd) 守护程序处理。

IPv6 地址分配过程与 IPv4 网络略有不同,因为 IPv6 主机总是自动为启用 IPv6 的接口分配链路本地地址,而无需外部主机的任何帮助。NDP 通过分发“前缀”而不是地址来建立在这种无状态自动配置过程的基础上。使用通过 NDP 获得的前缀（基本上是 IP 地址的网络部分）,主机可以为自己分配有效的 IPv6 地址。因此,与 IPv4 网络上的 DHCP 守护程序相比,NDP 没有 IP 池和租约的概念。

客户端不请求地址,而是发送“路由器请求”请求以获得有效的 IPv6 前缀。radvd 守护进程用路由器通告消息响应这些请求。这些消息包含链路上使用的路由前缀、最大传输单元 (MTU) 和负责默认路由器的地址。

radvd 守护进程由 `/etc/radvd.conf` 配置,它必须至少包含守护进程应该侦听的接口和它必须服务的前缀。此外,radvd 可以定期将其前缀重新通告给同一网络上的主机。如果您愿意,您还可以配置主机为自己配置的 IPv6 地址的生命周期。

典型的 radvd.conf 类似于以下内容:

```
接口 eth0 {
    AdvSendAdvert;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;前缀
    2001:0db8:0100:f101::/64 {
        AdvOnLink 开启;
        Adv自主;
        AdvRouterAddr 上;
    };
}
```

## PAM 身份验证 (210.2)

考生应该能够配置 PAM 以支持使用各种可用方法的身份验证。

### 关键知识领域

PAM 配置文件、术语和实用程序

passwd 和 shadow 密码

用于 LDAP 身份验证的基本 SSSD 功能

### 条款和实用程序

· `/etc/pam.d`

- pam.conf
- nsswitch.conf
- pam\_unix,pam\_cracklib,pam\_limits,pam\_listfile

## 资源

资源: **dracut;**

## 什么是 PAM?

PAM 是 Pluggable Authentication Modules 的缩写。 PAM 由一组库和一个 API (应用程序编程接口)组成,可用于执行身份验证任务。登录和 su 等特权授予程序使用 API 来执行标准身份验证任务。

## 它是如何工作的?

身份验证任务可以分为四个不同的功能组:

account 提供账户验证类服务:用户密码是否过期?该用户是否被允许访问  
请求服务?

身份验证 确定用户是否真的是他声称的人。例如,这可以通过询问密码来完成,或者,  
通过读取芯片卡或进行视网膜或指纹扫描,获得正确的模块。

密码 该组的职责是更新身份验证机制。通常,此类服务与身份验证组的服务紧密耦合。一些身份验证机制很适合更新。

可能会向用户提出“请输入新密码”之类的问题。

会话 这组任务涵盖了在提供服务之前和取消服务之后应该完成的事情。此类任务包括审计跟踪的维护和用户主目录的安装。会话管理组很重要,因为它为影响用户可用服务的模块提供了打开和关闭挂钩。

可以使用具有以下格式的文件 /etc/pam.conf 配置 PAM:

### 服务类型控制模块路径模块参数

这五个字段的含义是:

service 这是涉及的应用程序的名称,例如:login,ssh 或 passwd。

type 这是要执行的任务所属的组类型:account、auth (身份验证组)、password 或  
会议。

control 此字段指示 PAM-API 在任何模块的身份验证失败的情况下应该做什么。

控制字段有四个有效值:

requisite 失败时,认证过程将立即终止。 required 这将在调用此服务和类型的其余模块后返回失败。

sufficient 成功后,身份验证过程将得到满足,除非先前所需的模块未能通过 au

期待。

可选 如果这是与此服务关联的唯一模块,则此模块的成功或失败才重要  
和这种类型。

模块路径 这是应用程序要使用的 PAM 的文件名,包括完整路径。

模块参数 这些是要传递给模块的模块特定参数,由空格分隔。有关详细信息,请参阅特定模块的文档。

也可以使用单独的配置文件进行配置,这是推荐的。这些文件都应该位于 /etc/pam.d 目录中。如果此目录存在,文件 /etc/pam.conf 将被忽略。文件名应全部小写并与服务名称相同,例如登录。这些文件的格式与 /etc/pam.conf 相同,只是没有服务字段。

## 模块

### pam\_unix

该模块通过 /etc/passwd 和 /etc/shadow 配置身份验证。

PA M\_U NI XS O 模块支持以下管理组:

account 类型 “account”不对用户进行身份验证,但会检查其他内容,例如密码的到期日期,并且可能会强制用户根据文件 /etc/passwd 和 /etc/shadow 的内容更改其密码。

支持以下选项:

使用 syslog 调试日志信息。 audit 还记录信息,  
甚至比 debug 记录更多。

auth “auth”类型根据密码数据库检查用户密码。该组件在文件中配置  
/etc/nsswitch.conf。请查阅手册页 (man nsswitch.conf) 了解更多详细信息。

支持以下选项:

使用 syslog 审计日志信息。 debug 还使用  
syslog 记录信息,但少于审计。 nodelay 此参数将默认为一秒的失败延迟设置  
为 nodelay。 nullok 允许空密码。如果密码为空,通常身份验证会失败。 try\_first\_pass 使用之前堆叠的 auth 模块中的  
密码,如果检索到则提示输入新密码

密码为空或不正确。

use\_first\_pass 使用先前堆叠的身份验证模块的结果,从不提示用户输入密码,如果出现以下情况则失败  
结果失败了。

password “password”类型更改用户的密码。

支持以下选项:

使用 syslog 审计日志信息。 bigcrypt 使用  
DEC “C2”扩展到 crypt()。 debug 还使用 syslog 记录信息,但  
少于审计。 md5 使用 md5 加密代替 crypt()。 nis 使用 NIS (网络信息服务)密  
码。 not\_set\_pass 不要使用其他堆叠模块的密码,也不要将新密码提供给其他  
堆叠模块

模块。

nullok 允许空密码。如果密码为空,通常身份验证会失败。 remember 记住最后 n 个密码,以防止用户再次使用最后  
n 个密码之一。 try\_first\_pass 使用之前堆叠的 auth 模块的密码,如果检索到则提示输入新密码

密码为空或不正确。

use\_authok 将新密码设置为前一个模块提供的密码。

use\_first\_pass 使用先前堆叠的身份验证模块的结果,从不提示用户输入密码,如果出现以下情况则失败  
结果失败了。

session “session”类型使用 syslog 在会话开始和结束时记录用户名和会话类型。

“会话”类型不支持任何选项。

对于每项需要身份验证的服务,必须在 /etc/pam.d 中创建一个具有该服务名称的文件。这些服务的例子有:login、ssh、ppp、su。

出于示例目的,将使用文件 /etc/pam.d/login:

```
# 执行密码验证并允许没有密码的帐户 auth required pam_unix.so nullok

# 检查密码有效性并继续处理其他 PAM,即使 # 此测试失败。仅当 “足够”的 PAM (紧随其后的 “必需”PAM)成功时,才会授予访问权限。需要 pam_unix.so

帐户

# 在会话开始和结束时将用户名和会话类型记录到系统日志中。

会议      需要 pam_unix.so

# 允许用户更改空密码 (nullok),在接受密码更改之前执行一些额外的 # 检查 (模糊)并强制 # 密码的最小 (min=4)长度为 4,最大 (max=8) #8 个字符的长度。需要密码
pam_unix.so nullok obscure min=4 max=8
```

#### pam\_nis

该模块通过 NIS 配置身份验证。为了能够通过 NIS 进行身份验证,需要模块 pam\_nis.so。该模块可以在[PAM NIS 授权模块中找到。](#)

要以 NIS 身份验证就足够的方式进行设置 (如果不是这种情况,请尝试 pam\_unix.so) , /etc/pam.d/login 中的行是:

```
授权      足够的 pam_nis.so item=user\sense=allow
map=users.byname value=compsci
授权      需要 pam_unix.so try_first_pass

帐户足够 pam_ldap.so \ item=user sense=deny map=cancelled.byname
error=expired account required pam_unix.so
```

#### pam\_ldap

此模块通过 LDAP 配置身份验证。为了能够通过 LDAP 进行身份验证,需要模块 pam\_ldap.so。

该模块可在[PADL Software Pty Ltd](#)找到。

要以 LDAP 身份验证就足够的方式进行设置 (如果不是这种情况,请尝试 pam\_unix.so) , /etc/pam.d/login 中的行是:

```
授权      足够的 pam_ldap.so 需要 pam_unix.so
授权      try_first_pass

帐户足够 pam_ldap.so 帐户需要 pam_unix.so
```

### pam\_cracklib

此插件提供密码强度检查。这是通过执行大量检查以确保密码不会太弱来完成的。它根据字典、以前的密码和数字、大小写和其他字符的使用规则检查密码。

```
#%PAM-1.0
#
# 这些行允许 md5 系统支持至少 14 # 字节的密码,额外的数字为 2,其他为 2 新的 # 密码必须至少有三个字节,这在 # 旧密码中不存在 # password
required pam_cracklib.so所以 \
difok=3 minlen=15 dcredit=2 ocredit=2
需要密码 pam_unix.so use_authok nullok md5
```

### pam\_limits

pam\_limits PAM 模块对可在用户会话中获取的系统资源设置限制。uid=0 的用户也受此限制的影响。默认情况下,限制取自 /etc/security/limits.conf 配置文件。然后读取 /etc/security/limits.d/ 目录中的各个文件。这些文件按照“C”语言环境的顺序依次解析。单个文件的效果与所有文件按解析顺序连接在一起的效果相同。如果使用模块选项明确指定配置文件,则不会解析上述目录中的文件。该模块不得由多线程应用程序调用。

### pam\_list文件

该模块允许或拒绝基于项目在列表文件中存在的操作。列表文件是包含用户名列表的文本文件,每行一个用户名。item 的类型可以通过配置参数 item 来设置,其值可以是 user、tty、rhost、ruser、group 或 shell。sense 配置参数确定是否允许列表中的条目。

可能的值是允许和拒绝。

### 固态硬盘

为 LDAP 身份验证配置 SSSD

以下步骤描述了 SSSD 的配置以使用 LDAP 进行身份验证:

1. 需要安装以下包:

```
SSSD客户端
sssd-通用
sssd-common-pac sssd-ldap
sssd-代理 python-sssdconfig
authconfig authconfig-gtk
```

使用您的包管理器来安装这些包。

2. 检查 sssd 的当前设置 (如果有) :

```
# 授权配置测试
```

这将向您显示已经存在的当前设置。还要检查现有的 /etc/sssd/sssd.conf 文件。  
在全新安装中,您可能会禁用所有设置,并且不会出现 sssd.conf 文件。

3. 现在配置sssd:

```
# authconfig \--enablesssd \--enablesssdauth \--enablelecauthorize \--enableldap \--enableldapauth \--ldapserver=ldap://ldap.example.com:389 \--disableldaptls \--ldapbasedn=dc=example,dc=com \--enablerfc2307bis \--enablemkhomedir \--enablecachecreds \
```

- 更新

#### 4. 检查 /etc/sssd/sssd.conf 中的配置。

如果您使用的是 TLS,请确保正确配置了 `ldap_tls_cacertdir` 和 `ldap_tls_cacert` 参数并指向您的证书。同时将 `ldap_id_use_start_tls` 更改为 “True”。

要使更改生效,请运行:

```
# systemctl 重启 sssd
```

通过运行以下命令验证所有更改是否有效:

```
# 授权配置测试
```

#### 5. 更新 /etc/openldap.conf 以使用相同的 ldap 设置。您的 ldap.conf 文件将如下所示:

```
URI 上的 SASL_NOCANON  
ldaps://ldap.example.com:389 BASE  
dc=example,dc=com TLS_REQUIRE 从不  
TLS_CACERTDIR /etc/pki/tls/cacerts TLS_CACERT /  
etc/pki/tls/certs/mybundle.pem
```

请注意 `TLS_REQUIRE` 设置为从不。这样做是为了避免 PHP 等应用程序堆栈出现 LDAPS 和 TLS 问题。

6. 确保 sssd 已启动并正在运行,并且它将在系统重启后启动。运行 `systemctl status sssd` 来检查这一点。要启动 sssd,请运行 `systemctl start sssd` 并使 sssd 在重新启动后保持不变,请运行 `systemctl enable sssd`。

## LDAP 客户端使用 (210.3)

考生应该能够对 LDAP 服务器执行查询和更新。还包括导入和添加项目,以及添加和管理用户。

### 关键知识领域

用于数据管理和查询的 LDAP 实用程序

更改用户密码

查询 LDAP 目录

## 条款和实用程序

- ldapsearch
- ldappasswd
- ldapadd
- ldapdelete

## LDAP

LDAP 代表轻量级目录访问协议。顾名思义,它是 DAP 的轻量级版本,代表 X.500 标准定义的目录访问协议。有关 X.500 的更多信息,请阅读[RFC 2116](#)。

使用轻量级版本的原因是 DAP 的处理器负载相当重,因此要求的处理器比当时的处理器所能提供的要多。 LDAP 在[RFC 2251 中描述](#)。

LDAP 项目开始于[密歇根大学](#),但是,正如可以在他们的网站上看到的那样,那里不再维护。对于当前信息,密歇根大学网站会将访问者指向[OpenLDAP](#)网站。

最适合存储在目录中的信息类型是变异等级较低的信息。这样做的原因是目录无法与 RDBM 系统竞争,因为它们只针对读取访问进行了优化。那么,我们在目录中存储什么?通常,LDAP 目录包含雇员数据,例如姓氏、教名、地址、电话号码、部门、社会安全号码、电子邮件地址。或者,目录可以存储供所有人阅读的时事通讯、公司政策和程序的描述、支持文档内部风格的模板。

LDAP 是一个客户端/服务器系统。服务器可以使用各种数据库来存储目录,每个数据库都针对快速和大量的读取操作进行了优化。当 LDAP 客户端应用程序连接到 LDAP 服务器时,它可以查询目录或尝试修改目录。在查询的情况下,服务器要么在本地回答查询,要么可以将查询者引向确实有答案的 LDAP 服务器。如果客户端应用程序试图修改 LDAP 目录中的信息,服务器会验证用户是否有权进行更改,然后添加或更新信息。

## LDAP术语

条目 LDAP 目录中的单个单元。每个条目都由其唯一的可分辨名称 (DN) 标识。

attributes 与条目直接关联的信息。例如,一个组织可以表示为一个 LDAP 条目。

与组织关联的属性可能是其传真号码、地址等。人也可以表示为 LDAP 目录中的条目。人们的共同属性包括此人的电话号码和电子邮件地址。

一些属性是必需的,而其他属性是可选的。对象类定义为每个条目设置哪些属性是必需的,哪些不是。对象类定义位于 /etc/openldap/schema/ 目录中的各种模式文件中。

LDIF LDAP 数据交换格式 (LDIF) 是 LDAP 条目的 ASCII 文本表示。用于向 LDAP 服务器导入数据的文件必须是 LDIF 格式。 LDIF 条目类似于以下示例:

```
[<id>]  
dn: <可分辨名称> <attrtype>: <attrvalue>  
<attrtype>: <attrvalue> <attrtype>:  
<attrvalue>
```

每个条目可以根据需要包含尽可能多的 <attrtype>: <attrvalue> 对。空行表示条目的结尾。

### 笔记

所有 <attrtype> 和 <attrvalue> 对必须在相应的模式文件中定义才能使用此信息。

包含在 “<” 和 “>” 中的任何值都是一个变量,并且可以在创建新的 LDAP 条目时进行设置。但是,此规则不适用于 <id>。 <id> 是由用于编辑条目的应用程序确定的数字。

## ldap搜索

ldapsearch 是 ldap\_search(3) 库调用的 shell 可访问接口。 ldapsearch 打开与 LDAP 服务器的连接,绑定并使用指定参数执行搜索。过滤器应符合[RFC 2254 中定义的搜索过滤器的字符串表示形式](#)。

### LDAP 过滤器

平等	=	创建一个过滤器,要求字段具有给定值。例如, cn=Eric Johnson。
在场	=*	表示字段可以等于除 NULL 之外的任何值的通配符。因此它将返回具有一个或多个值的条目。例如, cn=* manager=* 返回包含包含指定子字符串的属性的条目。例如,cn=Bob* cn=*John*
子串	=字符串*字符串	cn=E*John。星号 (*) 表示零 (0) 个或更多字符。
近似	~=	返回包含指定属性的条目 ,其值大约等于搜索过滤器中指定的值。例如,cn~=suret l~=san franciso 可以返回 cn=surette l=san francisco 返回包含大于或等于指定值的属性的条目。
大于或等于	>=	
小于或等于	<=	返回包含小于或等于指定值的属性的条目。
括号	()	分隔过滤器以允许其他逻辑运算符起作用。
和	&	布尔运算符。将过滤器连接在一起。系列中的所有条件都必须为真。例如,(&(过滤器)(过滤器)...).
或者		布尔运算符。将过滤器连接在一起。系列中至少有一个条件必须为真。例如, (  (过滤器) (过滤器) ... )。
不是	!	布尔运算符。排除与过滤器匹配的所有对象。只有一个过滤器受 NOT 运算符的影响。例如,(!过滤器))

表 10.4:LDAP 字段运算符

布尔表达式按以下顺序求值:

- 从最内层到最外层的括号表达式优先。
- 所有表达式从左到右。

例子：

```
ldapsearch -h myhost -p 389 -s base -b "ou=people,dc=example,dc=com" objectclass=*
```

此命令搜索目录服务器 myhost,位于端口 389。搜索范围 (-s) 是 base, 搜索的目录部分是指定的基本 DN (-b)。搜索过滤器 “objectclass=\*>”意味着返回所有条目的对象类的值。未返回任何属性,因为尚未请求它们。该示例假定匿名身份验证,因为未指定身份验证选项。

```
ldapsearch -x "(|(cn=marie)(!(telephoneNumber=9*)))"
```

此示例说明如何搜索 cn 为 marie 或电话号码不以 9 开头的条目。

## ldap密码

**ldappasswd** - 更改 LDAP 条目的密码 **ldappasswd** 是用于设

置 LDAP 用户密码的工具。**ldappasswd** 使用 LDAPv3 密码修改([RFC 3062](#))扩展操作。**ldappasswd** 设置与用户 (或可选的指定用户) 关联的密码。如果未在命令行指定新密码且用户未启用提示,则将请求服务器为用户生成密码。

例子：

```
ldappasswd -x -h localhost -D "cn=root,dc=example,dc=com" \
-s secretpassword -W uid=admin,ou=users,ou=horde,dc=example,dc=com
```

在本地主机上为 “uid=admin,ou=users,ou=horde,dc=example,dc=com”设置密码。

## ldap地址

**ldapadd** - LDAP 添加条目工具

**ldapadd** 是作为到 **ldapmodify** 工具的链接实现的。当作为 **ldapadd** 调用时, -a (添加新条目) 标志会自动打开。

选项:**ldapmodify -a**

-a 添加新条目。**ldapmodify** 的默认设置是修改现有条目。如果作为 **ldapadd** 调用,则始终设置此选项。

例子：

```
ldapadd -h myhost -p 389 -D "cn=orcladmin" -w 欢迎 -f jhay.ldif
```

使用此命令,用户 orcladmin 向位于端口 389 的目录 myhost 进行身份验证。该命令然后打开文件 jhay.ldif 并将其内容添加到目录中。例如,该文件可能会添加条目 “uid=jhay,cn=Human Resources,cn=example,dc=com”及其对象类和属性。

## ldap删除

**ldapdelete** - LDAP 删除条目工具

**ldapdelete** 是 **ldap\_delete\_ext(3)** 库调用的 shell 可访问接口。

**ldapdelete** 打开与 LDAP 服务器的连接,绑定并删除一个或多个条目。如果提供了一个或多个 DN 参数,则删除具有这些专有名称的条目。

例子：

```
ldapdelete -h myhost -p 389 -D "cn=orcladmin" -w welcome \
uid=hricard,ou=sales,ou=people,dc=example,dc=com
```

此命令使用密码 welcome 对目录 myhost 的用户 orcladmin 进行身份验证。然后它删除条目 “uid=hricard,ou=sales,ou=people,dc=example,dc=com”。

## 更多关于 LDAP

如果您想阅读有关 LDAP 的更多信息,本部分将向您指出一些信息来源:

- [OpenLDAP 站点](#)
- [OpenLDAP 软件 2.4 管理员指南](#)
- [LDAP Linux HOWTO](#)
- [Internet FAQ 存档](#),可以在其中搜索和找到 RFC 和其他文档。

## 配置 OpenLDAP 服务器 (210.4)

考生应该能够配置一个基本的 OpenLDAP 服务器,包括 LDIF 格式和基本访问控制的知识。

### 关键知识领域

#### 开放式LDAP

基于目录的配置

访问控制

尊贵的名字

变更类型操作

模式和白页

目录

对象 ID、属性和类

#### 条款和实用程序

- 打耳光
- slapd.conf
- LDIF
- slapadd
- 打屁股
- slapindex
- /var/lib/ldap/
- 日志级别

资源:各种命令的手册页。

## 开放式LDAP

OpenLDAP 使用 slapd,它是独立的 LDAP 守护进程。它在任意数量的端口（默认为 389）上侦听 LDAP 连接，响应通过这些连接接收到的 LDAP 操作。OpenLDAP 通常在引导时启动。

它可以从源代码安装，从[OpenLDAP 软件](#)获取它，但大多数 linux 发行版通过它们的包管理系统（如 yum 或 apt）提供它。

在 Openldap 中，目录条目以分层树状结构排列。传统上，这种结构反映了地理和/或组织边界。代表国家的条目出现在树的顶部。在它们下面是代表州和国家组织的条目。在它们下面可能是代表组织单位、人员、打印机、文档或任何其他内容的条目。

### 访问控制

当 OpenLDAP 目录被填满时，数据变得更加重要。某些数据可能受法律保护或以其他方式保密。因此需要控制对目录的访问。默认策略允许对所有客户端进行读取访问。

无论定义什么访问控制策略，olcRootDN 始终被允许对所有内容拥有全部权限（即授权、搜索、比较、读取和写入）。

对 slapd 条目和属性的访问由 olcAccess 属性控制，其值是一系列访问规则。它们以访问指令开头，后跟条件列表：

```
olcAccess: 到<什么>
    由 <谁> <访问类型> 由 <谁> <访问类型>
```

例如：

```
olcAccess: 属性=用户密码
    通过匿名验证 通过自己编写 通过 * 无
```

此访问规范用于保护用户的密码。它允许匿名用户以登录为目的对密码进行身份验证比较。此外，它还授予用户更改其密码的权限。底线拒绝其他任何人访问密码。

或者，我们可以授予用户使用访问规范更新所有数据的权限，如下所示：

```
olcAccess: 属性=用户密码
    通过匿名身份验证 通过 * 无
    olcAccess: 到 *
        自己写 * 无
```

尊贵的名字

可分辨名称 (DN) 是唯一标识 OpenLDAP 目录中条目的名称（属性集），并且对应于到达该条目必须经过的路径。DN 包含由分隔的属性和值对逗号。

例如：

```
cn=John Doe,ou=编辑,o=巴黎,c=F cn=Jane Doe,ou=编辑,o=伦敦,c=英国 cn=Tom Jones,ou=报道,o=阿姆斯特丹,c=荷兰
```

目录模式中定义的任何属性都可用于构成 DN。组件属性值对的顺序很重要。DN 包含一个组件，用于从根向下到条目所在级别的目录层次结构的每一级别。LDAP DN 从最具体的属性开始，然后逐渐扩展为更广泛的属性。

DN 的第一个组成部分称为相对专有名称 (RDN)。它将一个条目与具有相同父项的任何其他条目区分开来。

为个人创建条目的示例：

```
dn: cn=John Doe,o=bmi,c=us objectclass: top  
objectclass: person cn: John Doe
```

```
SN:母鹿  
电话号码:555-111-5555
```

某些字符在 DN 中具有特殊含义。例如，=（等于）分隔属性名称和值，逗号分隔属性=值对。特殊字符有：逗号、等号、加号、小于号、大于号、分号、反斜杠、引号。

可以在属性值中转义特殊字符以去除特殊含义。要转义 DN 字符串中属性值中的这些特殊字符或其他字符，请使用以下方法：

如果要转义的字符是特殊字符之一，则在其前面加上反斜杠（“\” ASCII 92）。此示例显示了一种转义组织名称中的逗号的方法：

```
CN=L。Eagle,O=Sue\,Grabbit 和 Runn,C=GB
```

slapd配置

OpenLDAP 2.3 及更高版本已过渡到使用动态运行时配置引擎 slapd-config。仍然支持旧样式的 slapd.conf 文件，但不推荐使用它，并且在未来的 OpenLDAP 版本中将取消对它的支持。

---

#### 笔记

虽然 slapd-config 系统将其配置存储为（基于文本的）LDIF 文件，但您不应该直接编辑任何 LDIF 文件。配置更改应通过 LDAP 操作执行，例如 ldapadd、ldapdelete 或 ldapmodify。

---

根据 linux 发行版，slapd-config 配置树 slapd.d 可能位于 /etc/openldap 或 /usr/local/etc/openldap。

一个示例可能如下所示：

```
/etc/openldap/slapd.d |-- cn=config |--  
cn=module{0}.ldif  
  
|~ |-- cn=schema  
|~ |~ |-- cn={0}core.ldif  
|~ |~ |-- cn={1}余弦.ldif  
|~ |~ |-- cn={2}inetorgperson.ldif |~ |-- cn=schema.ldif  
  
|~ |-- olcDatabase={0}config.ldif |~ |-- olcDatabase={-1}frontend.ldif  
|~ |-- olcDatabase={1}hdb.ldif |-- cn=config.ldif
```

slapd.d 树有一个非常特殊的结构。树的根名为 cn=config 并包含全局配置设置。其他设置包含在单独的子条目中。

这些可能是以下内容：

- cn=module{0}.ldif 中的动态加载模块
- cn=schema 目录中的模式定义（更多关于模式的主题将在下文中介绍）
- cn=Database={1}hdb.ldif 中的后端特定配置
- cn=Database={0}config.ldif 中特定于数据库的配置

用于创建配置树的 LDIF 的总体布局（有关 LDIF 的更多信息，请参阅下面的部分）如下所示：

```
# 全局配置设置 dn: cn=config objectClass: olcGlobal cn:
config <global config settings>

# 模式定义
dn: cn=schema,cn=config objectClass:
olcSchemaConfig cn: schema

<系统架构>

dn: cn={X}core,cn=schema,cn=config objectClass:
olcSchemaConfig cn: {X}core

<核心模式>

# 额外的用户指定模式
...

# 后端定义 dn:
olcBackend=<typeA>,cn=config objectClass:
olcBackendConfig olcBackend: <typeA> <backend-
specific settings>

# 数据库定义
dn: olcDatabase={X}<typeA>,cn=config objectClass: olcDatabaseConfig
olcDatabase: {X}<typeA> <数据库特定设置>

# 后续的定义和设置
...
```

对于域 example.com，配置文件可能如下所示：

```
dn:olcDatabase=hdb,cn=config objectClass:
olcDatabaseConfig objectClass:olcHdbConfig
olcDatabase:hdb

olcSuffix: dc=example,dc=com olcRootDN:
cn=Manager,dc=example,dc=com olcRootPW: secret

olcDb 目录: /var/lib/ldap
```

使用 slappasswd 而不是上面示例中的纯文本密码 secret 生成密码哈希更安全。  
在这种情况下，olcRootPW 行将更改为如下所示：

```
olcRootPW:{SSHA}xEIqHqbSyi2FkmObnQ5m4fReBrjwGb
```

`olcLogLevel` 指令指定应在哪个调试级别语句和操作统计信息进行系统记录。日志级别可以指定为整数或关键字。可以使用多个日志级别，并且级别是累加的。

可用级别是：

```

1       (0x1 trace) 跟踪函数调用 (0x2 packets) debug packet handling
2       (0x4 args) heavy trace debugging (function args) (0x8 conns)
4       connection management (0x10 BER) print out packets send and received (0x20 filter)
8       search filter processing (0x40 config) 配置文件处理 (0x80 ACL) 访问控制列表处理 (0x100 stats)
16      统计日志连接/操作/结果 (0x200 stats2) 发送的统计日志条目 (0x400 shell) 打印与 shell 后端的通信 (0x800 parse) 条目解析 16384 (0x4000) sync) LDAPS Sync 复制 32768 (0x8000 无) 仅记录无论日志级别设置的消息
32
64
128
256
512
1024
2048

```

例如：

```
olc日志级别: -1
```

这将导致记录大量调试信息。

```
olcLogLevel: conns 过滤器
```

这只会记录连接和搜索过滤器处理。

```
olcLogLevel:统计
```

基本统计日志是默认配置的。但是，如果未定义 `olcLogLevel`，则不会发生日志记录（相当于 0 级别）。

请注意，保存用户数据的实际 OpenLDAP 数据库并不位于 `slapd.d` 配置目录树中。它的位置可能随 `olcDbDirectory` 目录而改变（参见上面的示例），但按照惯例它通常是 `/var/lib/ldap`。

它的内容通常如下所示：

```
$ ls -l /var/lib/ldap 总计 1168
-rw-r--r--。 1 ldap ldap -rw-----。 1 ldap          4096 12 月 12 日 14:29 锁定
ldap -rw-----。 1 ldap ldap 2351104 12 月        8192 Dec 2 21:31 cn.bdb
12 月 14:29 __db.001 -rw-----。 1 ldap ldap 819200 12 月 12 日 14:29 __db.002 -rw-----。 1 ldap ldap
163840 12 月 12 日 14:29 __db.003 -rw-rw-r--。 1 ldap ldap -rw-----。 1 ldap ldap -rw-----。 1 ldap
ldap 32768 12 月 2 日 21:31 id2entry.bdb -rw-----。 1个LDAP
                           104 12 月 2 日 21:12 DB_CONFIG
                           8192 12 月 2 日 21:31 dn2id.bdb

8192 12 月 2 日 21:31 objectClass.bdb
```

LDIF

对 OpenLDAP 数据库的所有修改都采用 LDIF 格式。LDIF 代表 LDAP 数据交换格式。

OpenLDAP 的工具（如 `slapadd`）使用它向数据库添加数据。一个 LDIF 文件的例子：

```
cat 添加用户.ldif
# John Doe 的条目 dn: cn=John
Doe,dc=example,dc=com
```

```
cn:李四  
cn: 杜琪峰  
objectClass: 人 SN: Doe
```

多个条目使用空行分隔。 Slapcat 可用于以 LDIF 格式从 LDAP 数据库中导出信息。

例如：

```
slapcat -l all.ldif
```

这将生成一个名为 all.ldif 的文件 ,其中包含 LDAP 数据库的完整转储。

slapadd 可以使用生成的输出将数据导入 LDAP 数据库。

例如：

```
slapadd -l all.ldif
```

有时可能需要重新生成 LDAP 的数据库索引。这可以使用 slapindex 工具来完成。它还可以用于为特定属性（如 UID）重新生成索引：

```
slapindex uid
```

请注意 ,运行该命令时 ,slapd 不应处于运行状态（至少 ,不应处于读写模式） ,以确保数据库的一致性。

## 目录

可以将目录设计为分层组织的数据集合。最著名的例子可能是电话目录 ,但文件系统目录是另一个例子。一般来说 ,目录是为阅读、浏览和搜索而优化的数据仓库。 OpenLDAP 目录包含描述性的、基于属性的信息。它们不支持关系数据库管理系统 (RDBMS) 中的回滚机制或复杂事务。如果允许的话 ,更新通常是简单的全有或全无更改。这种类型的目录旨在快速响应大量查找或搜索操作。可以复制 OpenLDAP 目录以提高可用性和可靠性。复制的数据库可能会暂时不同步 ,但最终会同步。

## 模式和白页

模式是描述可能存储在目录中的对象结构的标准方式。白页模式是一种数据模型 ,用于组织地址簿或 LDAP 等目录服务条目中包含的数据。在白页目录中 ,每个条目通常代表一个使用网络资源的个人 ,例如通过接收具有帐户登录系统的电子邮件。 LDAP 模式用于正式定义属性、对象类和构建目录信息树的各种规则。通常使用 include 指令在 slapd-config LDIF 中配置模式。

例如：

```
包括:file:///etc/openldap/schema/core.ldif 包括:file:///etc/openldap/schema/cosine.ldif  
包括:file:///etc/openldap/schema/inetorgperson.ldif
```

第一行导入核心模式 ,其中包含标准 LDAP 使用所需的属性和对象类模式。 cosine.schema 导入了许多常用的对象类和属性 ,包括用于存储文档信息和 DNS 记录的对象类和属性。第三个提供了 inetOrgPerson 对象类定义及其关联的属性定义。其他模式可用于 OpenLDAP (在 /etc/openldap/schema 中) ;有关详细信息 ,请参阅 OpenLDAP 软件 2.4 管理员指南。

## 参考

- OpenLDAP 软件 2.4 管理员指南
- 目录服务
- 轻型目录访问协议
- 系统安全服务守护进程
- 白页架构

## 问题和解答

### 网络客户端管理

1. LDAP 代表什么？

LDAP 代表轻量级目录访问协议 [LDAP \[305\]](#)

2. 什么是dn？

distinguishedName 或 dn 由一组唯一标识 LDAP 条目的属性组成。这组属性对应于到达该条目必须经过的路径。 [DN \[307\]](#)

3. 什么是共享网络声明？

如果同一物理网络上有多个子网，则使用共享网络声明。[共享网络\[289\]](#)

4. DHCP 服务器在哪个文件中保存租约数据库？

/var/dhcp/dhcpd.leases [DHCP 租约\[298\]](#)

5. 您将如何重新加载 DHCP 服务器？

/etc/init.d/dhcp 重新启动 [DHCP 重新启动\[298\]](#)

6. 什么是静态主机？

静态主机是始终从 DHCP 服务器获取相同 IP 地址的主机。[静态主机\[296\]](#)

7. 什么是租约？

租约是客户端可以使用它从 DHCP 服务器收到的 IP 地址的时间量。[租赁\[297\]](#)

8. DHCP 是什么意思？

DHCP 代表动态主机配置协议。 [DHCP \[289\]](#)

9. type auth 的作用是什么？

类型 auth 根据密码数据库检查用户密码。[授权\[301\]](#)

## 第11章

### 电子邮件服务 (211)

本主题的权重为 8 分,包含以下目标:

目标 211.1;使用电子邮件服务器 (4 分) 考生应该能够管理电子邮件服务器,包括电子邮件别名、电子邮件配额和虚拟电子邮件域的配置。此目标包括配置内部电子邮件中继和监视电子邮件服务器。

目标 211.2;管理本地电子邮件发送 (2 分) 考生应该能够实施客户端电子邮件管理  
过滤、分类和监控收到的用户电子邮件的软件。

目标 211.3 管理远程电子邮件传送 (2 分) 考生应该能够安装和配置 POP 和 IMAP  
守护进程。

#### 使用电子邮件服务器 (211.1)

考生应该能够管理电子邮件服务器,包括配置电子邮件别名、电子邮件配额和虚拟电子邮件域。此目标包括配置内部电子邮件中继和监视电子邮件服务器。

##### 关键知识领域

postfix 的配置文件

SMTP 协议基础知识

sendmail 和 exim 的认识

##### 条款和实用程序

- postfix 的配置文件和命令
- /etc/别名
- /etc/后缀/
- /var/spool/postfix/
- sendmail 仿真层命令
- /var/log 中与邮件相关的日志

资源: [PostfixTLSreadme](#)、[PostfixTFSreadme](#)、[SMTPauthHowto](#)、[RFC2487](#)、[raymii.org](#)、[PostfixGuide](#)、各种命令的手册页。

## SMTP 协议基础知识

自 2008 年 10 月起,RFC 5321 用于描述

当有消息要传输时,它会建立双向传输通道以将消息传输到一个或多个 SMTP 服务器,或者报告传输失败。

· 客户的责任是

将消息呈现给 SMTP 客户端的方式,以及如何确定将消息传输到的域名,是本地问题,此处不予讨论。

详细信息可参见:[RFC5321](#)。

出于演示和测试目的,可以使用 telnet、nc 或 ncat 启动纯文本 SMTP 会话:

```
telnet mx1.unix.nl 25
连接到 mx1.unix.nl (213.154.248.146)。
转义符是 ^] 。 220 mx1.unix.nl ESMTP Exim
4.63 Thu, 12 Aug 2010 08:50:30 +0200 ehlo snow.nl

250-mx1.unix.nl 你好 mx1.unix.nl [81.23.226.83]
250-尺寸 52428800
250-流水线
250-AUTH 普通登录
250-STARTTLS
250帮助
```

在执行上述命令时使用 nc 或 ncat 时,可能需要额外的 Enter 键击。这是由于 telnet 和其他类似工具之间可能存在的返回字符差异。

使用 telnet <smtp servername> 25 建立与 SMTP 服务器的初始联系。在上面的示例中,mx1.unix.nl 被用作服务器。服务器以最大消息大小、流水线和服务器的身份验证功能进行响应,在本例中为 AUTH PLAIN LOGIN。服务器还可以使用 STARTTLS 和 HELP。

- SIZE: 当客户端希望发送的消息大小超过设置的限制时,SMTP 事务将被中止,并带有一个错误代码流水线。此外,启用后,SMTP 客户端可以传输一组 SMTP 命令(例如,RSET、MAIL FROM:、RCPT TO:) ,而无需等待服务器的响应。
- AUTH PLAIN LOGIN: SMTP 服务器能够处理普通密码/用户名身份验证。这可以派上用场,适用于使用 SMTP 服务器同时来自不同 IP 地址的移动设备。
- STARTTLS: SMTP 协议本身不包含任何形式的加密。使用 STARTTLS 可以进行通信,使用证书加密。这在 RFC 3207 中有完整描述。

由于我们已经建立了初始连接,我们现在可以继续:

```
邮件来自:nonexistent@snow.nl 250 OK

接收到:nonexistent@unix.nl 250 已接受
```

服务器以“250 OK”响应,但是当“MAIL FROM:”命令后跟格式错误或不完整的电子邮件地址时,服务器以“501:发件人地址必须包含域”或类似错误响应。使用“RCPT TO: <email address>”给出消息的目的地。如果收件人的地址被接受,服务器将响应:“250 已接受”。

现在他们知道我们是谁以及我们希望向谁发送消息。他们还知道 SMTP 服务器的功能。

然后我们继续传输:

```
数据
354 输入消息,以“.”结尾单独一行 邮件来自:aiu <nonexistent@snow.nl> 来自:<nonexistent@snow.nl>

至:信息 <nonexistent@unix.nl>
```

主题:演示消息  
日期:12-02-2010 08:53:00  
MessageID:31231233@snow.nl 这是一条演示消息。

250 好 id=1OjReS-0005kT-Jj 退出

使用 DATA 命令,我们继续处理消息的内容。服务器响应:“354 输入消息,以“.”结尾。自己在一条线上”。

然后输入消息内容,从消息标题开始。这些标头由电子邮件客户端使用。在示例中,使用了命令 Mail From:、To:、Subject:、Date: 和 MessageID:。MessageID:是SMTP客户端生成的唯一标识符。如 RFC 5322 中所述,这些标头是必需的。

输入标题后,空行表示消息的实际文本开始。正如 DATA 命令响应所指示的那样,消息以“.”结尾。(不带引号)单独一行。服务器响应:“250 OK id=1OjReS-0005kT-Jj”。该 id 是唯一的 SMTP 服务器标识符,可用于故障排除。

#### 注意打

架,一些 SMTP 服务器可以检查邮件是否符合 RFC 5322 标准。常规 SMTP 客户端符合 RFC 5322,但垃圾邮件发送者通常使用不那么常规的 SMTP 客户端,因此可能会发送格式错误的消息。

## 发邮件

基本的 sendmail 配置步骤是:

1. 下载Sendmail
2. 设置Sendmail
3. 配置发送邮件
4. 建立Sendmail用户表
5. 将您的域名添加到Sendmail
6. 测试你的配置文件

sendmail 的基本配置描述于: [Sendmail 基本安装](#)。

sendmail 8.12 的安装和操作指南可以在: [Sendmail安装和操作指南中找到](#)。

Sendmail 可以从源代码构建,也可以作为 sendmail 二进制文件安装。对于 sendmail,您必须构建一个运行时配置文件。这是 sendmail 启动时读取的一个文件,描述了它知道的邮件程序,如何解析地址,如何重写邮件头,以及各种选项的设置。尽管它可能相当复杂,但通常可以使用基于 m4 的配置语言(是一种通用宏处理器)来构建。假设您有标准的 sendmail 分发版,请参阅 cf/README 以获取更多信息。

sendmail 配置文件经过 m4 处理,方便本地定制;分发目录中的目录 cf 包含源文件。它包含几个子目录:

比照

主机的站点相关和站点无关的描述。当这些文件是网关时,这些文件可以以主机(例如 ucbvax.mc)命名,或者以类别(例如 generic-solaris2.mc)命名,作为对运行 Solaris 2.x 的 SMTP 连接主机的一般描述。以 .mc (“M4 Configuration”) 结尾的文件是输入定义;输出在相应的 .cf 文件中。文件的一般结构如下所述。

## 领域

站点相关的子域描述。这些与您的组织想要进行寻址的方式相关联。例如, domain/CS.Berkeley.EDU.m4 是我们对 CS.Berkeley.EDU 子域中主机的描述。这些是使用 .mc 文件中的 DOMAIN m4 宏引用的。

## 特征

您站点的特定主机可能需要的特定功能的定义。这些是使用 FEATURE m4 宏引用的。

一个例子是 use\_cw\_file, 它告诉 sendmail 在启动时读取 /etc/mail/local-host-names 文件以确定本地主机集。

## 破解

本地黑客, 使用 HACK m4 宏引用。尽量避免这些。把它们放在这里的目的是让人们清楚地知道它们有气味。

## 立方米

与站点无关的 m4(1) 包含的文件具有所有配置文件共有的信息。这可以被认为是一个#include 目录。

## 邮递员

邮件程序的定义, 使用 MAILER m4 宏引用。此发行版中已知的类型是传真、本地、smtp、uucp 和 usenet。例如, 要包含对基于 UUCP 的邮件程序的支持, 请使用 MAILER(uucp)。

## 操作系统类型

描述各种操作系统环境的定义 (例如支持文件的位置)。这些是使用 OSTYPE m4 宏引用的。

## 嘘

m4 构建过程使用的 Shell 文件。您可能不必更改这些。

## 站点配置

本地 UUCP 连接信息。该目录已被 mailertable 功能所取代;任何新配置都应该使用该功能来执行 UUCP (和其他)路由。不推荐使用 siteconfig 目录。

确保 m4 实用程序在服务器上可用。使用它可以将 .mc 宏文件转换为 sendmail.cf 文件。配置 sendmail 的推荐方法是编辑宏文件而不是 sendmail.cf 文件本身。

sendmail 安装文件详见: [Sendmail 安装与操作指南](#)。

## 重要的发送邮件文件

/etc/mail/sendmail.cf - 主要的 sendmail 配置文件。

/etc/mail/access - 可以创建 sendmail 访问数据库文件来接受或拒绝来自选定域、系统和用户的邮件。由于它是一个数据库, 编辑文本文件后, 您必须使用 makemap 来创建或更新它。

要创建新的访问映射:

```
# makemap 哈希 /etc/mail/access.db < /etc/mail/access
```

在访问文件中可以定义几个动作：

确定 接受邮件,即使运行规则集中的其他规则会拒绝它,例如,如果域名无法解析。

“接受”不是“接力”的意思,最多是对当地收件人的接受。因此,OK 允许小于 RELAY。

RELAY 接受发往指定域或从指定域接收的邮件以通过您的 SMTP 中继服务器。 RELAY 还充当其他检查的隐式 OK。

REJECT 使用通用消息拒绝发件人或收件人。

DISCARD 使用 \$#discard 邮件程序完全丢弃邮件。如果它用在 check\_compat 中,它只影响指定的收件人,而不像在所有其他情况下那样影响整个邮件。只有在确实需要时才应使用它。

SKIP 这只能用于主机/域名和 IP 地址/网络。它将中止当前对该条目的搜索而不接受或拒绝它但会导致默认操作。

/etc/mail/local-host-names - 可以在 local-host-names 文件中定义本地主机名。将每个应视为本地的域添加到 /etc/mail/local-host-names 中。

/etc/mail/virtusertable - 用于将收到的电子邮件映射到本地帐户。使用 virtusertable,发送到一个帐户的消息可以分发给两个用户。还可以设置一个“catch all”电子邮件地址,将所有输入错误的电子邮件路由到一个特定用户,以创建一个新的 virtusertable。

要生成 virtusertable 数据库映射：

```
# makemap hash /etc/mail/virtusertable < 源文件
```

/etc/mail/genericstable - 用于出站邮件。可用于重写本地用户名,使它们看起来来自不同的主机或域。

/etc/mail/genericsdomain - 使用 hostname --long 命令填充此文件。

```
# 主机名 --long > genericsdomain
```

/etc/mail/mailertable - 用于路由来自远程系统的电子邮件。

/etc/mail/domaintable - 可用于从旧域名转换为新域名。

/etc/mail/aliases - 用于为本地收件人重定向邮件。 /etc/aliases 的每一行都有 alias: user 的格式。必须始终存在两个系统别名 :mailer\_daemon: postmaster 和 postmaster: root。您可以为所有类型的守护进程使用别名,例如使用 ntp: root。现在您可以添加一行将所有邮件重定向到 root 到特定用户或管理员组,例如 root: marc。 newaliases 需要在对该文件进行任何更改后运行。

```
# 新别名
```

如果对其中一个配置文件进行了任何更新,sendmail 需要重新加载其配置：

```
# killall -HUP 发送邮件
```

## 抗中继

从 8.9 版本开始,sendmail 默认不中继。使用较旧的 sendmail 版本时,请更改 sendmail.cf 或访问文件以确保它不会中继。反中继提示在以下位置进行了描述:[控制 SMTP 中继提示 - Sendmail.org](#)。

## Sendmail 测试选项

Sendmail 可以在测试模式下运行。使用 -b 和 -t 选项来执行此操作。您需要以 root 身份运行它。

```
# 发送邮件 -bt
```

## 发送邮件和 DNS

确保 MTA 的 MX 记录在 DNS 中可用。这可以通过以下方式检查：

```
$ 挖 MX somedomain.com
```

## sendmail.cf 中的手动条目

如前所述，这不是推荐的方式。改为更改 m4 文件并使用 m4 生成 sendmail.cf。

进出口银行

Exim 是剑桥大学开发的 (MTA)，用于连接到 Internet 的 Unix 系统。可以安装 Exim 而不是 Sendmail，尽管 Exim 的配置有很大不同。有关详细信息，请参阅 [exim.org](#)。

[exim4-config\\_files](#) 手册页包含对每个文件的描述。Exim 带有一个 exim.conf 模板。只需为您的环境编辑此配置文件。[GitHub 上的 Exim Wiki](#) 提供了额外的配置帮助、FAQ 等。

## 后缀

Postfix 是另一种广泛采用的 MTA。它侧重于速度、易于管理和安全性。该软件的作者 Wietse Venema 在学习期间依赖于长时间的计算机计算。正是在那段时间里，他养成了创建尽可能无故障和抗故障软件的习惯。只有当他相信该软件可以连续运行两周而没有错误时，才没有必要在这两周里睡在地板上照看它。在接下来的几年里，这种编码习惯一直伴随着作者。首次编写 Postfix 程序时，它是按照与早期学习作业相同的习惯完成的。这种习惯后来成为一种编码哲学。通过以结构化方式编写代码，仅添加必要的位并确保执行流程尽可能高效，最终的软件变得快速、易于管理且安全。今天，Postfix 仍在积极开发中。许多人为此做出了贡献，现在仍有许多人这样做。

尽管有这种共同的努力，原作者仍然保证只添加必要的代码。而且它的添加方式不会干扰从一开始就属于 Postfix 的理念。这种方法与 Postfix 由多个小程序而不是一个大程序组成的方式相结合，使 Postfix 成为其他 MTA 解决方案的一个非常有吸引力的替代方案。

在撰写本文时，Postfix 版本 3.1 是最新的稳定版本。Postfix 版本 3.2（仍然）被认为是实验性的。

实验性 Postfix 版本可以通过它们的名称来识别。这些将带有日期。实验版本的示例是以下 tarball：postfix-3.2-20161204.tar.gz。稳定的 Postfix 版本仅显示（主要和次要）版本号，末尾没有日期。由于其广泛采用，大多数 Linux 发行版都提供 Postfix。但是，部署的版本种类繁多。根据其网站 <http://www.postfix.org>，Postfix 2.10 之前的版本已终止支持 (EOS)。CentOS 6.8 仍然附带 Postfix 2.6。Debian 7 没有超越 Postfix 2.9。较新的 Linux 发行版本，如 CentOS 7 和 Debian 8 附带 Postfix >2.10，而一些前沿的 Linux 发行版本提供 Postfix >3.1。绝对不能确定较新版本的 Postfix 会比旧版本的 Postfix 有更少的错误。了解您正在向 Internet 公开哪个版本的 Postfix，同时了解已知的 Postfix 漏洞的最新信息被视为“最佳实践”。但是，没有理由将此最佳实践仅限于 Postfix。

为了解决为 LPIC-2 考试定义的 Postfix 目标，以下将提供一些示例。这些假设您正在使用从基于 Debian 或基于 Red Hat 的 Linux 发行版上的软件包安装的 Postfix。可能仍然存在需要从源代码构建 Postfix 的情况。该主题超出了本章的范围，但 LPIC 目标 206.1 涵盖了一般主题。

### Postfix main.cf 文件格式

postfix 配置文件的默认位置是 /etc/postfix。其中有两个主要决定了我的 Postfix 行为：main.cf 和 . 默认情况下，main.cf 配置文件指定控制 Postfix 邮件系统操作的所有参数的一个子集。未明确指定的参数保留其默认值或通过所有源的完整列表，请参阅 postconf(5) 手册页。参见 /etc/postfix/main.cf.dist 可能是精简版。/usr/share/postfix 目录应该包含一个更详细的 main.cf.dist 文件。以下行计数操作给出了基于 Debian 的系统上不同 main.cf 文件之间差异的估计：

```
user@debian:/usr/share/postfix$ wc -l /etc/postfix/main.cf 43 /etc/postfix/main.cf user@debian:/
usr/share/postfix$ wc -l /usr/share/后缀/main.cf* 18 /usr/share/postfix/main.cf.debian 665 /usr/
share/postfix/main.cf.dist 11 /usr/share/postfix/main.cf.tls
```

上面的示例演示了 /usr/share/postfix/main.cf.dist 文件包含 600 多行，而默认的 /etc/postfix/main.cf 文件仅包含 43 行。保持 main.cf 文件干净整洁是个好习惯。在基于 Debian 的系统上使用寻呼机阅读更详细的 main.cf.dist 文件也是一个好习惯。基于 Red Hat 的系统在 /etc/postfix 目录中保存了详细记录的配置文件。

默认情况下，postfix 只会从 /etc/postfix 目录中读取配置文件。可以使用 /etc/postfix/main.cf 中的 alternate\_config\_directories 指令指定其他配置目录。基于 Red Hat 的发行版可能有 /etc/postfix/dynamicmaps.cf.d 和 /etc/postfix/postfix-files.d 目录。这些目录的存在是为了减轻 Postfix 包创建者的生活，除非您真的确定自己在做什么，否则应该单独放置。

---

#### 注意

postfix 配置目录 /etc/postfix 在编译中是“硬编码”的。以下文档的第 4.6 章解释了从源代码构建 Postfix 时如何更改此值：[Postfix Installation From Source Code](#)

---

main.cf 文件的一般格式如下：

- 每条逻辑行都采用 parameter = value 的形式。“=”周围的空格被忽略，就像 a 末尾的空格一样逻辑线。
- 空行和纯空白行被忽略，第一个非空白字符为“#”的行也是如此。
- 逻辑行以非空白文本开始。以空格开头的行继续逻辑行。
- 一个参数值可以引用其他参数。
- 当多次定义同一个参数时，只记住最后一个实例。
- 否则，main.cf 参数定义的顺序无关紧要。

另请参阅 [Postfix 配置参数](#) 作为 postconf 手册页的替代方法。该文档的其余部分是对所有 Postfix 配置参数的描述。默认值显示在参数名称后的括号中。也可以使用 sudo /usr/sbin/postconf -d 命令查找这些默认值。但是请注意，因为此命令将在 Postfix 3.1 实例上产生 877 行输出。因为它是一个管理命令并且通常位于 /usr/sbin 中，所以应该使用 sudo 或具有 root 权限的 shell 来调用此命令。

与 Postfix 一起使用的“最佳实践”之一是在可公开访问的邮件服务器上禁用 SMTP VRFY 命令。

攻击者可以滥用此 VeriFY 命令来枚举有效的用户帐户或电子邮件地址，如下所示：

```
user@mailserver:~$ telnet 邮件服务器 25 正在尝试 ::1...
连接到邮件服务器。
转义符是 ^] 。 220 本地主机 ESMTP 后缀 (Linux/
GNU)
EHLO 本地主机
250-本地主机
250-流水线
250-尺寸 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-增强状态代码
250-8比特MIME
250 深海网
VRFY根
252 2.0.0 根
VRFY沃丹
550 5.1.1 <wodan> :收件人地址被拒绝 :本地收件人表中的用户未知 VRFY postfix 252 2.0.0 postfix
```

200 类响应代码公开有效的收件人。就像 HTTP 服务器一样,SMTP 服务器可以使用状态代码进行响应。建议熟悉最常见的响应代码。这将有助于调试系统性能或问题。通常不需要一眼就知道220和250的区别。但是 2xx 和 5xx 状态码之间的区别应该很明显并且很熟悉。默认情况下,VRFY 命令在 Postfix 上启用。通过在 main.cf 中启用或添加以下行,VRFY 功能被禁用：

```
disable_vrfy_command = 是
```

在禁用 VRFY 命令并重新加载 Postfix 之后,让我们从上面再次尝试 VRFY root 命令：

```
user@mailserver:~$ telnet 邮件服务器 25 正在尝试 ::1...
连接到本地主机。
转义符是 ^] 。 220 本地主机 ESMTP 后缀 (Linux/
GNU)
EHLO 本地主机
250-本地主机
250-流水线
250-尺寸 10240000
250-ETRN
250-STARTTLS
250-增强状态代码
250-8比特MIME
250 深海网
VRFY根
502 5.5.1 VRFY 命令被禁用
VRFY沃丹
502 5.5.1 VRFY 命令被禁用
VRFY 后缀
502 5.5.1 VRFY 命令被禁用
```

通过禁用区分有效 (2xx) 和无效 (5xx) 电子邮件收件人的可能性,Internet 已成为一个更安全的地方。

Postfix 开发人员建议一次更改不超过 2-3 个参数,并在每次更改后测试 Postfix 是否仍然有效。对 main.cf 进行更改后,您需要重新加载 postfix。根据所使用的 Linux 发行版,这可以使用 sudo 或具有 root 权限的 shell 使用以下命令之一来完成:postfix reload 或 systemctl restart postfix 甚至 /etc/init.d/postfix reload。

### Postfix master.cf 文件格式

Postfix 邮件系统遵循模块化方法,通过实现少量 (大部分)由用户调用的客户端命令,以及大量在后台运行的服务。 Postfix 服务由守护进程实现。它们在后台运行并由主进程控制。 master.cf 配置文件定义了客户端程序如何连接到服务,以及在请求服务时运行的守护程序。

master.cf文件的一般格式如下:

- 空行和纯空白行被忽略,第一个非空白字符为 “#” 的行也是如此。
- 逻辑行以非空白文本开始。以空格开头的行继续逻辑行。
- 每个逻辑行定义一个单独的Postfix 服务。每个服务都由其名称和类型标识,如下所述。当多行指定相同的服务名称和类型时,只记住最后一个。否则, master.cf 服务定义的顺序无关紧要。

每条逻辑行由八个由空格分隔的字段组成。下面按照它们在 master.cf 文件中出现的顺序对它们进行了描述。在适用的情况下,“-”字段要求使用内置默认值。对于布尔字段,指定 “y” 或 “n” 以覆盖默认值。

Chroot 选项 (默认值:Postfix >= 3.0: n,Postfix <3.0: y) - 服务是否以 chroot 模式运行到邮件队列目录 (路径名由 de main.cf 文件中的 queue\_directory 配置变量控制) 。

### 后缀准备

在可以使用 postfix 之前,它需要知道:

- 接收邮件的域 (第11.1.11.3.1 节)
- 用于出站邮件的域名 (第11.1.11.3.2 节)
- 允许为哪个域后缀中继邮件 (第11.1.11.3.3 节)
- 使用何种交付方式 (第11.1.11.3.4 节)

### 我的起源

myorigin 参数指定在外发电子邮件中显示为 “发件人:” 域的域。 myorigin 选项受制于其他选项,这些选项定义 “From:” 域是附加还是替换为 myorigin 值。有关详细信息,请参阅 main.cf 文件或 postconf 手册页。默认情况下,myorigin 值定义为 \$myhostname。可以按如下方式将 myorigin 值更改为配置的域名:

```
myorigin = $我的域
```

\$myhostname 或 \$mydomain 被 postfix 替换为运行它的服务器的主机名或域。

### 我的目的地

Postfix 需要知道它将接收哪个域的邮件。为此使用参数 mydestination。可以指定一个以上的域。域名可以使用空格或逗号分隔。此外,模式可用于指向查找表 (散列、btree、nis、ldap 或 mysql) 。

```
mydestination = $mydomain, localhost.$mydomain, hash:/etc/postfix/moredomains
```

### 注意

当服务器用作整个域的邮件服务器时,您必须包括\$mydomain。

## 中继域

postfix 的默认配置将尝试仅将传入邮件传递到授权目的地。 relay\_domains 参数控制 postfix 将接受和转发电子邮件的域。

```
中继域 = (安全:切勿转发陌生人的邮件)
```

```
relay_domains = $mydomain (将邮件转发到我的域和子域)
```

## 中继主机

默认情况下,postfix 尝试根据邮件消息中目标地址的域名直接投递到 Internet。使用 relayhost 参数,我们可以指定使用另一个 SMTP 服务器作为中继:

```
中继主机 =
```

这是默认的,直接传送到互联网,或使用另一个 ISP SMTP 服务器:

```
relayhost = mail.example.com
```

## 记录

Postfix 使用 syslog 守护进程进行日志记录。当 /etc/syslog.conf 配置如下例所示时,Postfix 日志事件将写入 /var/log/maillog。在下面的示例中,错误消息被重定向到控制台。

```
邮件错误          /dev/console  
mail.debug /var/log/maillog
```

### 提示

使用`egrep <reject|warning|error|fatal|panic> /var/log/maillog`将帮助您找到 postfix 遇到的任何问题。  
还有第三方实用程序,如`plogsumm`,可以从 Postfix 日志中生成统计信息。

## 虚拟域

通常,后缀服务器是有限数量域的最终目的地。但是 postfix 也可以配置为处理其他域的邮件,这些域不同于 postfix 服务器所在的域。这些目的地称为虚拟主机。使用 virtual\_alias\_domains 参数,我们可以指定我们希望接收邮件的虚拟主机。参数的格式与上面示例中的相同。使用空格或逗号分隔多个虚拟主机。也可以链接到磁盘上的 (散列)文件:

```
virtual_alias_domains = example.com, snow.nl, unix.nl
```

或者在使用散列文件时 (使用 postmap 实用程序) :

```
virtual_alias_domains = hash:/etc/postfix/virtual
```

/etc/postfix/virtual 的内容可以是:

```
postmaster@example.com peter info@snow.nl  
gerda sales@example.com 姜娜@example.com
```

在上面的示例中,peter 收到了 postmaster@example.com 电子邮件。Gerda 收到 info@snow.nl 电子邮件,sales@example.com 转到 petra。最后一行是“捕获所有”规则,所有没有有效目的地 example.com 的电子邮件都会发送给 jim。

---

#### 注意使

用postmap /etc/postfix/virtual创建散列文件并在之后发出postfix 重新加载。

---

## Sendmail 仿真层命令

由于 Sendmail 多年来一直是类 Unix 系统上事实上的邮件传递标准,因此像 Postfix 这样的替代品不得不实现 sendmail 仿真层命令,以保持与外部程序的兼容性。也就是说,最初包含在 sendmail 包中的某些命令也可用于 Postfix。

Mailq mailq 在大多数系统上都可用,用于检查邮件队列。它相当于 sendmail -bp,与 Postfix 一起使用也。它的本机替代项是 postqueue -p。

Newaliases 需要 newaliases 来生成带有 sendmail 和 postfix 的二进制别名文件。两者都使用 /etc/ 别名作为默认输入文件。

在 Postfix 系统上,man sendmail 将提供更多关于“Postfix to Sendmail compatibility interface”的细节。

## /var/假脱机/邮件

指定默认邮件投递目录。默认情况下,所有邮件都传送到 /var/spool/mail/<username> 文件。

LPIC 目标 208.2 涵盖基于 SSL/TLS 的 HTTP。SMTP 协议类似于 HTTP 协议,因为这两种协议都使用纯文本网络命令。这在调试时会派上用场,因为它允许通过 telnet 或类似 netcat 的软件执行命令。但在跨一系列系统传输敏感信息时,它并不派上用场。由于 SMTP 协议的性质,将电子邮件消息从源传输到最终目标可能涉及多台机器。并且无法排除消息在传输过程中被更改的可能性。这就是加密的用武之地。通过使用加密通信,至少可以将某种形式的机密性和真实性添加到 SMTP 事务中。就像 Apache 需要为 HTTPS 事务正确设置一样,Postfix 也需要配置为与 TLS 一起使用。与 Apache 不同的是,我们省略了 SSL 首字母缩写词,仅提及 TLS for Postfix。以下段落将详细解释 TLS 特定的 Postfix 指令以及配置 Postfix 以与 TLS 一起使用。

在撰写本文时,TLSv1.2 是最新的 TLS 版本。TLSv1.3 已达到 DRAFT 状态,它走上街头只是时间问题。TLSv1.2 和 TLSv1.3 之间会有一些值得注意的变化。直到 TLSv1.2,密码套件由身份验证算法、加密算法、消息摘要算法和密钥交换算法组成。

这四种算法一起工作,并由它们的首字母缩写词定义。从 TLSv1.3 开始,密码套件将只包含加密和消息认证算法。

以下段落旨在帮助您启动和运行带有 TLS 的 Postfix。加密是一个不断扩展的主题,建议花时间阅读和理解有关 Postfix 和 (正确) 使用 TLS 的可用文档。Postfix 附带大量文档,可以在 /usr/share/doc/postfix 目录中找到(在基于 Debian 的系统上,您可能必须安装 postfix-doc 包以确保完整性,如本页底部所述)。TLS\_README 文档绝对值得一读。如果文件被压缩,可以使用 less 之类的寻呼机查看它,或者首先使用 zcat 之类的实用程序展开它。TLS\_README.txt 文档也可以在 Postfix 网站<http://www.postfix.org> 上找到。

创建用于 Apache 的密钥对时,最好使用加密的私钥。加密密钥需要使用密码解密才能使用。默认情况下,OpenSSL 会在创建过程中对私钥进行加密,除非另有说明。Postfix 要求用于 TLS 加密通信的私钥是未加密的,换句话说就是没有密码。通过使用 OpenSSL 读取、导入和导出密钥,可以从密码中剥离私钥。

但在以下示例中, -nodes 选项在密钥创建期间与 OpenSSL 一起使用。这将首先阻止 OpenSSL 加密私钥。

```
sudo openssl req -nodes -x509 -newkey rsa:2048 \ -keyout postfixkey.pem -out
postfixcert.pem \ -days 356
```

使用上面一行创建的自签名证书和私钥可以用于Postfix。Postfix要求私钥文件由root拥有，并且只能由root读取。证书文件可以是世界可读的。因为上面的示例使用了RSA算法，所以使用以下指令来引用这些文件：

```
smtpd_tls_cert_file=postfixcert.pem
smtpd_tls_key_file=postfixkey.pem
```

#### 注意声

明这些指令时，请特别注意区分smtpd\_tls\_指令和smtp\_tls指令。  
后者用于客户端身份验证。

证书文件必须是PEM格式。这是OpenSSL标准导出格式，因此不需要额外的标志。在上面的示例中，使用了RSA算法。代替RSA，DSA也可以用作加密算法。

使用DSA算法时，需要使用-t dsa选项生成密钥，并且证书文件和密钥的指令与对应的RSA略有不同：

```
smtpd_tls_dcert_file=postfixcert.pem
smtpd_tls_dkey_file=postfixkey.pem
```

ECDSA是另一种与Postfix一起使用的有效加密算法。但是由于许多邮件服务器仍然使用缺乏正确ECDSA实现的OpenSSL 0.9.8版本，因此目前不建议将ECDSA用于带有TLS的Postfix。

目前，RSA是推荐用于Postfix的TLS的加密算法。但是，如果您确实想使用ECDSA密钥，则必须使用-t ecdsa选项生成密钥，并且必须使用以下指令指定证书和密钥文件：

```
smtpd_tls_eccert_file=postfixcert.pem
smtpd_tls_eckey_file=postfixkey.pem
```

上面的OpenSSL示例创建了一个自签名证书。这是一种让Postfix STARTTLS功能“快速而肮脏”地工作的方法。但就真实性而言，许多公共SMTP服务器可能无法验证自签名证书。另一种方法是创建密钥和CSR，并使用您自己的CA签署CSR。这在Apache章节中进行了演示。通过使用您自己的CA颁发的证书，您可以让您的邮件服务器相互验证。另一种选择是创建密钥和CSR，并让CSR由公共CA签名。这样，公共SMTP服务器应该能够验证您的Postfix服务器提供的证书。走这条路时，可能需要添加额外的根CA引用来完成证书链。可以使用smtpd\_tls\_CAfile或smtpd\_tls\_CApth指令配置其他根CA文件。文件指令应该指向一个（你猜对了）文件，其中包含必要的根CA信息。

将多个证书连接到一个文件中时，请注意文件是“从下到上”读取的。smtpd\_tls\_CApth指令应指向包含必要文件的目录。

Postfix默认使用机会加密。这意味着连接系统将首先尝试创建加密连接。但是，如果由于任何原因（例如，由于证书无效）而失败，则系统将继续执行所有SMTP命令而不加密。此行为由smtpd\_tls\_security\_level指令的值决定。

通过将此指令的值从may更改为encrypt，未加密的会话将被阻止。不过要注意！如果无法启动TLS会话，此指令可能会阻止Postfix服务器再执行任何SMTP事务。

---

#### 注意

尽管来自各种公共证书颁发机构的所有良好意图,证书颁发机构系统在设计上还是存在缺陷。毕竟,任何证书颁发机构都可以为任何主机上的任何服务颁发证书。DNS 章节提到了 DANE TLSA 记录来解决这个问题。当 DNSSEC 已正确配置并支持 TLSA 记录时,建议为给定的支持 TLS 的 Postfix 服务器创建 301 或 311 TLSA 记录。通过将信任从拥有 1300 多个 CA 根证书的 CA 信任库转移到 DNSSEC 的真实性和完整性,依赖自签名证书的问题变得不那么重要了。如果在同一个 Postfix 服务器上提供多个域,建议在 DNS 中为每个单独的域 (以及证书)发布摘要。可以将 smtpd\_tls\_security\_level 值设置为 dane 或 dane-only 以处理 TLSA 记录。有关详细信息,请参阅 TLS\_README 文件。

---

指令和选项 Postfix 在编写时考虑了许多“如果……会怎样”的回退场景。结果是一个非常有弹性的软件。作为良好编码实践的结果,Postfix 使用多种机制来确定环境是否有利。如果系统资源变得稀缺,Postfix 系统将根据情况进行调整。在配置 Postfix 以使用 TLS 加密时,有一些指令值得额外注意。正确配置后,指令应该相互补充功能。当配置不一致或不正确时,指令可能会与其他功能发生冲突。建议额外注意配置允许的协议和密码套件时涉及的指令。现代 Postfix 版本不使用 SSLv2。因为版本升级可能会更改默认协议和允许的密码套件,所以自己配置这些没有坏处。通过在 main.cf 中声明以下指令,这些选项将保持不变,尽管 Postfix 版本随着时间的推移而升级。

```
smtpd_tls_protocols
smtpd_tls_mandatory_protocols
smtpd_tls_ciphers smtpd_tls_mandatory_ciphers
smtpd_tls_exclude_ciphers
```

通过禁用 aNULL 密码,可以防止使用匿名密码。使用匿名密码是帮助 Postfix 以牺牲责任为代价提供可用性的后备方案之一。

如前所述,可以使用 smtpd\_tls\_security\_level 指令配置 Postfix 中的 TLS 功能。

根据 RFC 2487,该指令应设置为 may 值。这将导致 Postfix 向连接的客户端宣布 STARTTLS 的可用性,而不要求使用它。如果要强制使用 STARTTLS,则应将 smtpd\_tls\_security\_level 的值设置为加密。同样,请注意这可能会导致服务器与其他系统的兼容性下降。通过将指令值设置为 none,Postfix 服务器将完全避免宣布 STARTTLS 方法。当连接系统尝试启动 STARTTLS 时,Postfix 服务器将回复 502 5。

#### 5.1 错误:命令未执行消息。

为 Postfix 启用 TLS 时,可以使用各种附加配置指令。大多数以 smtpd\_tls\_ 或 smtp\_tls\_ 开头的指令对 Postfix 的工作并不重要,除非启用了 TLS。这些指令之一是 smtpd\_tls\_loglevel 指令。可以通过设置 0-4 之间的值来配置该指令。不建议使用高于 2 的设置,除非用于调试目的。值为 0 将禁用记录 TLS 事件。值为 1 将在 TLS 握手完成时记录一条摘要消息。值 2 还将在 TLS 协商期间记录级别。值 3 将记录上述所有内容以及 TLS 事务的十六进制和文本转储。值 4 最终将记录以上所有内容,但在处理 TLS 事务后不会停止转储。它将在 STARTTLS 命令后继续转储整个 SMTP 传输。谨慎使用!

在大多数 Linux 发行版中,可以在 /usr/share/doc/postfix 目录中找到有关 Postfix 的附加文档。在基于 Debian 的系统上,这需要安装 postfix-doc 包。如果不加这个包,那个目录的内容会很浅。

---

#### 注意

Postfix 3.x 通过合并 postfix-tls 使 TLS 更容易。在阅读时,您可能还想查看 tlsmgr 和 tlsproxy 的作用。

---

## 管理电子邮件传送 (211.2)

考生应该能够使用客户端电子邮件管理软件来过滤、分类和监控收到的用户电子邮件。

### 关键知识领域

了解 Sieve 功能、语法和运算符

使用 Sieve 根据发件人、收件人、标题和大小过滤和排序邮件

对procmail的认识

### 条款和实用程序

- 条件和比较运算符

- 保留、归档、重定向、拒绝、丢弃、停止

- 鸽舍假期延长

### 邮箱

Procmail 是一种电子邮件过滤实用程序,可用于对传入邮件进行预处理和分类。它还可用于从邮件列表中整理电子邮件、过滤垃圾邮件和发送自动回复。 Procmail 配置基于放置在用户主目录中的文件。它很少从命令行运行 (测试目的除外),但它是一个自主程序,通常由 MTA (邮件传输代理)如 Sendmail 或 Postfix 调用。

Procmail 遵循以下方案来读取其配置 (它读取两者) :

```
/etc/procmailrc
~/.procmailrc
```

使用系统范围的 /etc/procmailrc 时要小心。它通常以 root 身份读取和处理。这个事实意味着该文件中设计不当的配方可能会造成严重损害。例如,拼写错误可能会导致 Procmail 覆盖重要的系统二进制文件,而不是使用该二进制文件来处理消息。出于这个原因,您应该将系统范围的 Procmail 处理保持在最低限度,而是专注于使用 ~/.procmailrc 来处理使用个人帐户的电子邮件。

### 筛选

Sieve 是一种脚本语言,可用于预处理和分类收到的电子邮件。它还可用于从邮件列表中整理电子邮件、过滤垃圾邮件和发送自动回复。要使用 sieve,首先应该在电子邮件服务器上配置它。在此设置中,postfix 用于将电子邮件传递到 Dovecot 本地传递代理。在文件 main.cf 中,mailbox\_command 选项应配置为使用本地传递代理。

```
mailbox_command = /usr/lib/dovecot/dovecot-lda -a $RECIPIENT
```

下一步是在 dovecot 中启用筛选支持。在配置文件 15-lda.conf 中应配置以下选项:

```
lda_mailbox_autocreate = 是
lda_mailbox_autosubscribe = 是
协议 lda {
    mail_plugins
    = $mail_plugins 筛选
}
```

配置选项:lda\_mailbox\_autocreate 使 dovecot 在 sieve 中的规则启动时创建邮箱。选项:lda\_mailbox\_autosubscribe 如果这是由针对特定邮箱的自动创建操作启动的,则将用户订阅到特定邮箱。选项:mail\_plugins 在 dovecot 中启用 sieve 脚本模块。配置文件90-sieve.conf需要做如下调整:

```
插件 { sieve
    = ~/.dovecot.sieve sieve_dir = ~/sieve

    sieve_default = /var/lib/dovecot/sieve/default.sieve sieve_global_dir = /var/lib/
        dovecot/sieve
}
```

筛选配置选项是用户可以保存筛选规则的位置。sieve\_dir 配置选项是可以容纳多个筛选脚本的用户目录。配置选项 sieve\_default 用于执行默认筛选脚本。如果用户在其主目录中有个人 sieve 配置文件,则全局配置文件将被否决。

选项 sieve\_global\_dir 标识一个全局目录以包含多个全局筛选脚本。配置完成后,服务应该通过 service postfix restart 和 service dovecot restart 重新启动。

### 筛选语法

筛选语法很容易理解。它由三个基本部分组成:控制、测试和动作命令。控制命令控制代码的流程。它们会影响命令的执行方式。测试命令将与控制命令结合使用,以指定可能导致操作的条件。条件评估为真后将执行操作命令。您还可以使用以“#”开头的单行注释和以“/\*”开头并以“\*/”结尾的多行注释来阐明筛选规则。例子:

```
# 评论
需要[ “扩展名” ];

/* 这是多个
行注释 */

# if -> 控制命令
如果 <条件> {
    动作1;动作2;

    ...
停止; #结束处理
```

Sieve 还通过使用 vacation> 扩展提供自动回复功能。下面是一个使用休假扩展的示例。

```
需要 [ fileinto , 假期 ];
假期
# 每天最多回复同一个发件人一次 :days 1

:subject 外出回复
# 自动回复中包含的其他收件人地址列表。
# 如果邮件的收件人不是信封收件人并且不在此列表中,则不会为其发送休假回复。:地址[ “j.doe@company.dom” , “john.doe@company.dom” ]

“我不在办公室,请联系 Joan Doe。
此致
约翰·多伊” ;
```

“假期”扩展提供了几个选项,即:

- :days number - 用于指定保留地址且不回复的时间段 (以天为单位) 。

- :subject string - 指定附加到任何休假响应的主题行。
- :from string - 指定在休假消息的“发件人”字段中使用的替代方式。
- :addresses string-list - 为收件人指定额外的电子邮件地址。
- :mime - 指定任意的mime 内容。例如,以不同语言指定多个假期消息。
- :handle string - 告诉 sieve 将具有不同参数的两个休假操作视为相同的响应命令  
追踪
- reason: string - 实际消息

例子：

```
需要 “假期” ; if header :contains
  subject  lunch { vacation :handle ran-away 我出去了,不
    能见面吃午饭 ; } else { vacation :handle 逃跑 我出去了 ;
  }
```

控制指令

sieve 中的控制命令是基本的 if、else 和 elseif 控制语句。如果测试条件被评估为真,那么相关的动作将被执行。在控制语句中,可以使用以下标记参数: :contains、:is、:matches、:over 和 :under,如即将到来的筛选示例所示。 require 控制命令用于在脚本的开头声明可选扩展(例如 fileinto) , stop 命令结束脚本的所有处理。例子：

```
需要 “归档” ;
如果标头:包含 “来自” “彩票”{丢弃; } elseif header :contains
[ subject ][ $$$ ]{丢弃;

} else { fileinto
  INBOX ;
}
```

例子：

```
如果标头:包含 “主题” “金钱”{
  丢弃;
  停止;
}
```

测试命令

上节所述的控制命令可以支持不同的测试命令,即:address、allof、anyof、exists、false、header、not、size、true。

使用 address 命令,您只能测试标题中是否包含电子邮件地址。如果 to 标头包含 “John Doe <john@doe.com>” ,那么测试将评估为 false。如果收件人地址包含 “john@doe.com” ,那么测试将为真,因为只评估地址。例子：

```
需要 “归档” ;
如果地址:是 “to” “john@doe.com”{
  归档到 “约翰” ;
}
```

`allof` 命令是一个逻辑“AND”，意味着所有条件都应评估为 true 以进行进一步操作。例子：

```
如果所有 (标题:包含 “来自” “Bofh”,标题:包含 “到” “滥用” ){

    归档为 “垃圾邮件” ;
}
```

`anyof` 命令是一个逻辑“或”，意味着任何条件都应该被评估为 true 以进行进一步的操作。例子：

```
if anyof (size :over 1M, header :contains subject big file attached ){

    拒绝 “我不想要声称有大文件的邮件。” ;
}
```

`exists` 命令测试标头是否随消息一起退出。对于正在执行的任何操作，所有标头都必须返回 true。例子：

```
如果存在 “x-custom-header”

{
    重定向 “admin@example.com” ;
}
```

`false` 命令只返回 false。

`header` 命令测试标头是否与参数设置的条件匹配并评估为真。例子：

```
if header :is [ subject ] 快速赚钱 { discard;

    停止;
}
```

`not` 命令应该与另一个测试一起使用。此命令否定要采取的操作的其他测试。下面的示例意味着如果消息不包含“发件人”和“日期”，则将采取丢弃操作。例子：

```
如果不存在 [ from , date ]{

    丢弃;
}
```

`size` 命令用于指定消息大小高于或低于指定值，以便将条件评估为真。该命令接受标记参数`:over` 和`:under` 并且您可以在指定的值之后使用 M 表示兆字节，K 表示千字节，没有字母表示字节。例子：

```
如果大小:超过 500K {

    丢弃;
}
```

`true` 命令只返回 true。

#### 动作指令

在测试命令被评估为 true 或自行运行后，将执行操作命令。操作命令是`:keep`、`fileinto`、`redirect` 和`discard`。`keep` 操作命令使消息保存在默认位置。`fileinto` 操作命令是一个可选命令，可以通过在脚本开头使用`require fileinto` 控制命令来使用。如果测试命令被评估为真，则消息被移入定义的邮箱。例子：

```
如果附件:匹配 [ *.vbs , *.exe ] { fileinto INBOX.suspicious ;

}
```

`redirect` 命令将消息重定向到参数中指定的地址，而不篡改消息。

例子：

```
如果存在 "x-virus-found"{  
    重定向 "admin@example.com" ;  
}
```

discard 命令导致消息在不发送任何通知或任何其他消息的情况下被静默删除。例子：

```
如果大小:超过 2M { 丢弃;  
}
```

## Mbox 和 maildir 存储格式

Mbox 和 maildir 是电子邮件存储格式。Postfix 和 Dovecot 支持两种电子邮件存储格式，其中 maildir 是推荐格式。

### Mbox 格式

Mbox 是传统的电子邮件存储格式。在这种格式中，只有一个常规文本文件用作用户的邮箱。通常，此文件的名称是 `/var/spool/mail/<user name>`。当对邮箱执行操作时，Mbox 会锁定邮箱。操作后邮箱解锁。

优点：

- 普遍支持 Mbox 格式
- 添加新电子邮件的速度很快
- 单个邮箱内搜索速度快

缺点：

- Mbox 因锁定问题而闻名
- mbox 格式容易损坏

### 邮件目录格式

Maildir 是较新的电子邮件存储格式。为每个电子邮件用户创建一个目录 maildir，通常在用户的主目录中。默认情况下，在此 maildir 目录下还存在三个目录：new、cur 和 tmp。

优点：

- 查找、检索和删除特定电子邮件的速度很快，尤其是当电子邮件文件夹包含数百封邮件时
- 几乎不需要文件锁定
- 可用于网络文件系统
- 不受邮箱损坏的影响（假设硬件不会出现故障）

缺点：

- 某些文件系统可能无法有效处理大量小文件
- 搜索文本需要打开所有电子邮件文件，速度很慢

### procmailrc 的 mbox 和 maildir 之间的配方差异

在将本页的食谱复制到您的 procmailrc 文件之前,请记住将它们调整为您特定的 maildir/mbox 格式,同时考虑到 maildir 文件夹的名称以 “/” 结尾。使用 maildir 格式 (:0 而不是 :0:) 时不需要锁定文件。

在 mbox 中,格式为:

```
:0:  
配方目录  
名称
```

在 maildir 中,它将是:

```
:0  
食谱目录  
名称/
```

## 管理邮箱访问 (211.3)

考生应了解 Courier 电子邮件服务器,并能够在 Dovecot 上安装和配置 POP 和 IMAP 守护进程服务器。

### 导游

Courier 邮件传输代理 (MTA) 是一个基于开放商品协议 (如 ESMTP、IMAP、POP3、LDAP、SSL 和 HTTP) 的集成邮件/群件服务器。Courier 在单一、一致的框架内提供 ESMTP、IMAP、POP3、网络邮件和邮件列表服务。可以随意启用或禁用各个组件。Courier 邮件服务器现在实现了集成在 webmail 模块中的基本的基于 web 的日历和调度服务。

Courier 邮件服务器使用 maildirs 作为其本地邮件存储格式,但它也可以将邮件传递到旧邮箱文件。

默认情况下 /etc/courier 是 sysconfdir。所有快递配置文件都存储在这里。邮件队列可以在 /var/spool/mqueue 中找到。

有关 Courier 配置的信息可在以下位置找到: [Courier 安装](#)。

### 鸽舍

Dovecot 是用于 Linux/UNIX 类系统的开源 IMAP 和 POP3 电子邮件服务器,编写时主要考虑安全性。

Dovecot 声称它是小型和大型安装的绝佳选择。

Dovecot 的配置可以在 /etc/dovecot.conf 中找到,我们需要配置几个参数:身份验证、邮箱位置、SSL 设置和作为 POP3 服务器的配置。

### 验证

Dovecot 能够使用多种密码数据库后端,如:PAM、BDSAuth、LDAP、passwd 和 SQL 数据库,如 MySQL、PostgreSQL 和 SQLite。最常见的方法是 PAM 身份验证。PAM 配置通常位于 /etc/pam.d 中。默认情况下,Dovecot 使用 dovecot 作为 PAM 服务名称。

这是 /etc/pam.d/dovecot 的示例:

```
授权      需要帐户 需要          pam_unix.so nullok  
                           pam_unix.so
```

客户端用于将登录凭据发送到服务器的方法是通过机制参数配置的。最简单的身份验证机制是 PLAIN。客户端只需将未加密的密码发送给 Dovecot。所有客户端都支持 PLAIN 机制，但显然存在任何在网络上监听的人都可以窃取密码的问题。出于这个原因（以及其他一些原因），实施了其他机制。

SSL/TLS 加密可用于保护 PLAIN 身份验证机制，因为密码是通过加密流发送的。非明文机制被设计为即使没有 SSL/TLS 加密也可以安全使用。由于它们的设计方式，它们需要访问明文密码或它们自己的特殊哈希版本。这意味着不可能将非明文机制与常用的 DES 或 MD5 密码哈希一起使用。对于成功/失败密码数据库（例如 PAM），根本不可能使用非明文机制，因为它们只支持验证已知的明文密码。

Dovecot 支持以下非明文机制：CRAM-MD5、DIGEST-MD5、APOP、NTLM、GSS-SPNEGO、GSS API、RPA、ANONYMOUS、OTP 和 SKEY、EXTERNAL。默认情况下仅启用 PLAIN 机制。您可以通过修改 /etc/dovecot.conf 来更改它：

```
授权默认{
    机制 = 普通登录 cram-md5 # ..

}
```

#### 邮箱位置

使用 /etc/dovecot.conf 中的 mail\_location 参数，我们可以配置我们要使用的邮箱位置：

```
mail_location = maildir:~/Maildir
```

或者

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

在这种情况下，电子邮件存储在 /var/mail/%u 中，其中“%u”被转换为用户名。

#### SSL

在 Dovecot 可以使用 SSL 之前，需要创建 SSL 证书并且 Dovecot 必须配置为使用它们。

Dovecot 包含一个脚本 mkcert.sh 来创建自签名 SSL 证书：

```
#!/bin/sh

# 生成自签名证书。
# 在运行之前编辑 dovecot-openssl.cnf。

OPENSSL=${OPENSSL-openssl}
SSLDIR=${SSLDIR-/etc/ssl}
OPENSSLCONFIG=${OPENSSLCONFIG-dovecot-openssl.cnf}

CERTDIR=$SSLDIR/证书
KEYDIR=$SSLDIR/私有

CERTFILE=$CERTDIR/dovecot.pem KEYFILE=$KEYDIR/
dovecot.pem

if [ echo ! -d $CERTDIR ];然后
$SSLDIR/certs 目录不存在 exit 1

菲

如果 [ ! -d $KEYDIR ];然后
```

```
echo $SSLDIR/private 目录不存在 exit 1

菲

如果 [-f $CERTFILE ];然后
echo $CERTFILE 已经存在,不会覆盖
出口 1

菲

如果 [-f $KEYFILE ];然后
echo $KEYFILE 已经存在,不会覆盖
出口 1

菲

$OPENSSL req -new -x509 -nodes -config $OPENSSLCONFIG -out $CERTFILE -keyout $KEYFILE -< days 365 ||出口 2

chmod 0600 $KEYFILE 回显

$OPENSSL x509 -subject -fingerprint -noout -in $CERTFILE ||出口 2
```

重要的 SSL 配置选项可以在文件中找到:conf.d/10-ssl.conf。要启用客户端和 Dovecot 服务器之间传输的数据加密,应进行以下更改。

SSL = 必需

此配置选项要求客户端使用 SSL/TLS 作为传输层机制。没有 SSL/TLS 的身份验证尝试将导致身份验证失败。启用 SSL/TLS 的另一个重要配置选项是 SSL/TLS 密钥和 SSL/TLS 证书的配置。本示例中的证书是安装 Dovecot 时自动生成的。

```
ssl_cert = </etc/dovecot/dovecot.pem ssl_key = </etc/dovecot/
private/dovecot.pem
```

证书的首选权限是 0440（全球可读）。证书提供给客户。密钥的权限应该是 0400 和 uid/gid 0。它应该只能由 root 用户读取。如果密钥文件受密码保护,则可以通过更改 ssl\_key\_password 选项在配置文件中配置密码。由于 SSL 和 TLSv1 协议容易受到多种攻击,例如 POODLE (Padding Oracle On Downgraded Legacy Encryption),这些协议应该被禁用。

```
ssl_protocols = !SSLv2 !SSLv3 !TLSv1
```

配置加密的另一个关键特性是确定 Dovecot 应使用的密码套件。密码套件通过启动与客户端的安全连接来定义服务器提供的允许密码。您应该记住,邮件用户代理应该支持在服务器上配置的密码套件,否则无法建立安全连接。下面显示了一个密码套件的示例:

```
ssl_cipher_list = AES256+EECDH:AES256+EDH
```

密码 AES256+EECDH 表示密码使用经过身份验证的 Ephemeral Elliptic Curve Diffie Hellman 密钥协议。该协议用于通过不安全的通道共享秘密。此密钥可用于通过使用对称加密协议 (在此配置中为 AES256 位) 来加密和解密通信。密码 AES256+EDH 和 AES256+EECDH 几乎一样。该密码不是使用椭圆曲线而是 RSA 算法。另一个应该配置的选项是:

```
ssl_prefer_server_ciphers =
```

此选项优先使用服务器上配置的密码,而不是来自客户端的密码。此配置选项避免了所谓的降级攻击。此攻击由中间人攻击执行,并删除强加密套件以仅从客户端启动弱密码。攻击者可以攻击弱密码,主要目的是解密加密的流量。另一个重要的配置选项是:

```
ssl_dh_parameters_length = 2048
```

此选项将 Diffie-Hellman 密钥交换配置为 2048 位密钥。最近发布了 Logjam 漏洞。此攻击与密码套件降级攻击有关。攻击者可以降级 TLS 连接以使用 512 位 DH 加密。

在 Linux 客户端上,支持的密码套件首先列出共享库 (例如 openssl 或 gnutls) ,然后使用命令 openssl ciphers 或 gnutls-cli -l 列出支持的密码。如果邮件用户代理不支持密码,例如 mutt,它将显示错误,例如

```
gnutls_handshake:已收到 TLS 致命警报。(握手失败)
```

- 在 SSL/TLS 配置之后,应该配置 imaps 和 pop3s 侦听器。监听器可以在文件中配置: 10-master.conf。

```
服务 imap 登录 {
    inet_listener imap { port = 0
        #port = 143
    }
    inet_listener imaps {
        端口= 993
    }
}

服务 pop3-登录 {
    inet_listener pop3 { port = 0
        #port = 110
    }
    inet_listener pop3s { 端口 = 995
    }
}
```

如果侦听器端口设置为 0,则 pop3 和 imap 服务不会在服务器上运行。仅启用协议的安全版本。在配置 dovecot 之后,dovecot 服务器应该重新启动。这可以通过命令启动: service dovecot restart。使用以下命令验证 pop3s 和 imaps 服务是否在适当的端口上侦听:

```
# netstat -anp |egrep "993|995|0 0.0.0.0:|TCP|TCP6"
TCP        0 0.0.0.0:993          0.0.0.0:*          听      3515/鸽舍
TCP        0 0.0.0.0:995          0.0.0.0:*          听      3515/鸽舍
TCP TCP6   0::993               ::*:*              听      3515/鸽舍 3515/鸽舍
TCP6 0    0::995               ::*:*              听
```

如您所见,pop3s 和 imaps 正在监听它们配置的端口并准备好使用。

#### POP3 服务器

尽管 Dovecot 主要设计为 IMAP 服务器,但它作为 POP3 服务器也能正常工作,但并未为此进行优化。POP3 规范要求准确报告大小,并使用 Maildir 将换行符存储为纯 LF 字符。

因此,简单地获取文件大小会返回错误的 POP3 邮件大小。

当使用 mbox 而不是 Maildir 时,索引文件会在 POP3 启动时更新并包含所有邮件。用户删除所有邮件后,索引文件再次更新为包含零封邮件。将 Dovecot 用作 POP3 服务器时,您可能需要考虑使用 mbox\_min\_index\_size 设置禁用或限制索引文件的使用。

## 问题和解答

### 电子邮件服务

1. 如何检查是否可以到达某个 SMTP 服务器？

使用 telnet SMTP-server 25 并等待连接响应。[使用 telnet 连接的 SMTP 会话\[316\]](#)

2. Postfix 使用两个配置文件 :postfix.cf 和 master.postfix。正确的？

不，这些文件是 :main.cf 和 master.cf。[后缀\[320\]](#)

3. Postfix 使用哪个参数来了解哪些域可以接收邮件？

要使用的参数是 mydestination。可以使用空格、逗号或两者来分隔多个域名。

[配置后缀\[323\]](#)

4. 使用虚拟域的目的是什么？

为了配置 Postfix 来处理其他域的邮件，除了它自己的虚拟主机之外，还可以添加。[虚拟域\[324\]](#)

5. 什么是 procmail？

Procmail 是一种邮件过滤实用程序，用于对传入邮件进行预处理和分类。[Procmail \[328\]](#)

6. 您如何获得 procmail 识别的所有标志的概览？

`procmail-h`。

7. 菜谱最后一行开头的爆字符（!）是什么意思？

这意味着必须将符合先前条件的邮件转发到另一个邮箱。

8. 为什么 Courier 需要创建系统别名？

Courier 不会将邮件传递给 root（出于安全原因），因此需要创建系统别名（至少对于 root 而言）。

9. Dovecot 最常用的认证方式是什么？

最常见的方法是使用 PAM 身份验证。[鸽舍认证\[333\]](#)

10. 为什么在使用 Dovecot 作为 POP3 时使用 mbox\_min\_index\_size 配置和限制索引文件的使用服务器？

这是一种解决方法，以便在使用 mbox 而不是 maildir 时更新已删除的邮件消息。[Dovecot POP3 服务器\[336\]](#)

## 第12章

# 系统安全 (212)

本主题总权重为 14 分,包含以下目标:

目标 212.1;配置路由器 (3 分) 考生应该能够配置一个系统来执行网络地址转换 (NAT、IP 伪装)并说明它在保护网络方面的重要性。此目标包括配置端口重定向、管理过滤规则和避免攻击。

目标 212.2;保护 FTP 服务器 (2 分) 考生应该能够配置 FTP 服务器以进行匿名下载和上传。此目标包括在允许匿名上传和配置时要采取的预防措施

用户访问。

目标 212.3;安全外壳 (SSH) (4 分) 考生应该能够配置和保护 SSH 守护进程。此目标包括为用户管理密钥和配置 SSH。考生还应该能够通过 SSH 转发应用程序协议并管理 SSH 登录。

目标 212.4;安全任务 (3 分) 考生应该能够从各种来源接收安全警报、安装、配置和运行入侵检测系统并应用安全补丁和错误修复。

目标 212.5; OpenVPN (2 分) 考生应该能够配置 VPN (虚拟专用网络)并创建安全的点对点或站点到站点连接。

信息来源: <https://openvpn.net/IPTables/OpenSSH,FTP>

### 配置路由器 (212.1)

考生应该能够配置系统来执行网络地址转换 (NAT、IP 伪装)并说明其在保护网络方面的重要性。此目标包括配置端口重定向、管理过滤规则和避免攻击。

### 关键知识领域

网络地址转换 (NAT)

iptables 配置文件、工具和实用程序

用于管理路由表的工具、命令和实用程序

私有地址范围

端口重定向和IP转发

根据源或目标协议、端口和地址列出和编写接受或阻止数据报的过滤和规则

保存并重新加载过滤配置

ip6tables 和过滤的认识

## 条款和实用程序

- /proc/sys/net/ipv4
- /etc/服务
- iptables

## 私有网络地址

为什么会有私有网络地址?为所有使用 IP 地址的主机分配全球唯一的地址已成为一种常见的做法。为了延长 IPv4 地址空间的寿命,地址注册机构比以往任何时候都需要更多关于额外地址空间需求的理由,这使得组织更难获得额外的地址空间。

企业内部使用IP的主机可以分为三类:

类别 1 这些主机不需要访问其他企业的主机或 Internet 本身;此类别中的主机可以使用在企业内明确的 IP 地址,但在企业之间可能不明确。

类别 2 这些主机需要访问一组有限的外部服务(例如,电子邮件、FTP、网络新闻、远程登录),这些服务可以由中介网关(例如,应用层网关)处理。对于此类中的许多主机,不受限制的外部访问(通过 IP 连接提供)可能是不必要的,甚至是不受欢迎的(出于隐私/安全原因)。

这些主机与第 1 类主机一样,可以使用在企业内部明确的 IP 地址,但在企业之间可能不明确。

类别 3 这些主机需要在企业外部进行网络层访问(通过 IP 连接提供);最后一类中的主机需要全局明确的 IP 地址。

我们将第一类和第二类主机称为“私有”主机,将第三类主机称为“公共”主机。

许多应用程序只需要一个企业内部的连接,而大多数内部主机不需要外部(企业外部)连接。在大型企业中,通常很容易识别大量使用 TCP/IP 的主机,这些主机不需要企业外部的网络层连接。

互联网号码分配机构 (IANA) 为私人互联网保留了以下三个 IP 地址空间块:

10.0.0.0	- 10.255.255.255 (10/8 前缀)	- 172.31.255.255
172.16.0.0	(172.16/12 前缀)	- 192.168.255.255 (192.168/16 前缀)
192.168.0.0		

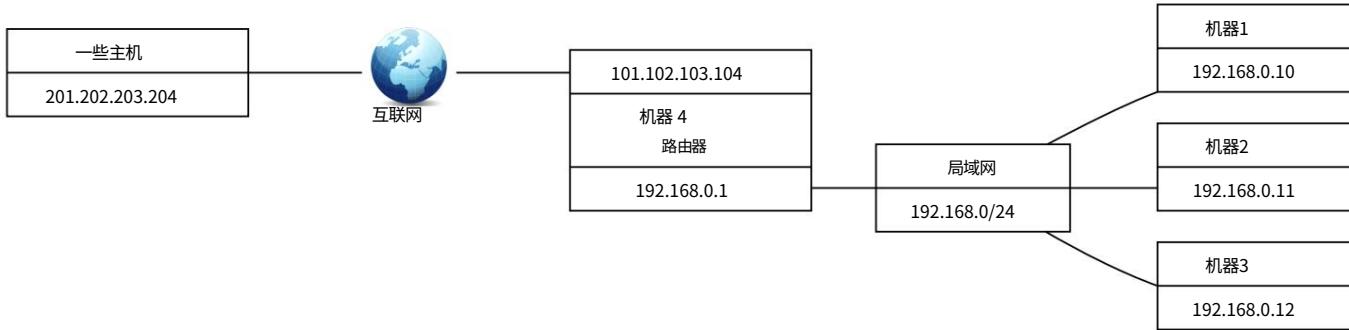
我们将第一个块称为“24 位块”,第二个称为“20 位块”,第三个称为“16 位”块。请注意,当不使用子网划分时(即在预无类域间路由(CIDR)表示法中),第一个块不过是单个 A 类网络号,而第二个块是一组 16 个连续的 B 类网络号,并且第三块是一组 256 个连续的 C 类网络号。

尽管 IPv6 地址在可预见的未来不太可能用完,但人们已经认识到分配私有地址的必要性。RFC4193 描述了地址块 fc00::/7,它是上述 IPv4 私有地址的近似对应物。

除了私有IP地址外,IPv6重新引入链路本地地址的概念,仅对主机所连接的网段(链路)或广播域内的通信有效。路由器不转发具有链路本地地址的数据包,因为不能保证它们在其网段之外是唯一的。在 IPv4 中,网络范围 169.254.0.0/16 是为接口保留的,可以自动为自己分配 IP 地址。实际上,在接口上找到此范围内的 IP 地址通常意味着 DHCP 分配失败,因为 IPv4 网络中通常不使用链路本地寻址。

在 IPv6 网络中,除了可能配置或分配的其他 IPv6 地址之外,接口总是分配链路本地地址。因此,IPv6 接口通常有多个地址。链路本地地址是 IPv6 协议标准的组成部分,用于促进邻居发现(NDP)和使用 DHCP6 分配全球唯一的 IP 地址。为 IPv6 配置的接口使用其 MAC 地址的一部分作为在 fe80::/64 范围内创建(希望如此)唯一链路本地地址的一种方式。

## 网络地址转换 (NAT)



上图显示了一个假设情况,将在以下说明中作为示例。

本节介绍网络地址转换 (NAT),这是一种重写某些 IP 流量的源地址或目标地址 (有时两者)的技术。它可用于使使用私有 IP 地址的主机能够与使用全球唯一 IP 地址的主机进行通信。NAT 主要是一个 IPv4 概念。IPv6 不鼓励使用 NAT,因为它的创建者认为它造成的问题多于它解决的问题。然而,在超出本书范围的某些情况下,可能会出现重写 IPv6 地址的需要。

“The Firm”有四台机器,1 到 4,它们通过网络交换机连接,并具有 192.168.xx 范围内的私有 IP 地址。机器 4 作为互联网的路由器,有两个网络接口。一个通过网络交换机将路由器连接到公司的内部网络,并具有私有 IP 地址 192.168.0.1,而另一个将路由器连接到 Internet,并具有有效 (动态)IP 地址 101.102.103.104。

假设机器 2 的用户希望查看 IP 地址为 201.202.203.204 的某个主机 (<http://SomeHost.example>) 上的网页。为了能够看到网页,机器 2 必须能够从 Internet 获取信息,因此必须以某种方式连接到 Internet。事实上,机器 2 通过机器 4 间接连接到 Internet,但这怎么可能呢?机器 2 的私有 IP 地址在 Internet 上不受支持 (路由) !

这就是 NAT 发挥作用的地方。路由器 Machine 4 在将请求发送到某个主机之前,用自己的 IP 地址替换了 Machine 2 (以及 Machine 1 和 3,如果需要的话)的私有 IP 地址。某些主机认为 IP 地址为 101.102.103.104 的机器请求网页并通过将网页发送给机器 4 来响应。

机器 4 知道在将请求发送到某个主机之前,它已经用自己的 IP 地址替换了机器 2 的 IP 地址,因此它也知道它得到的请求的答案必须转到机器 2。机器 4 通过替换自己的 IP 地址来完成此操作应答中的 IP 地址为机器 2 的 IP 地址。

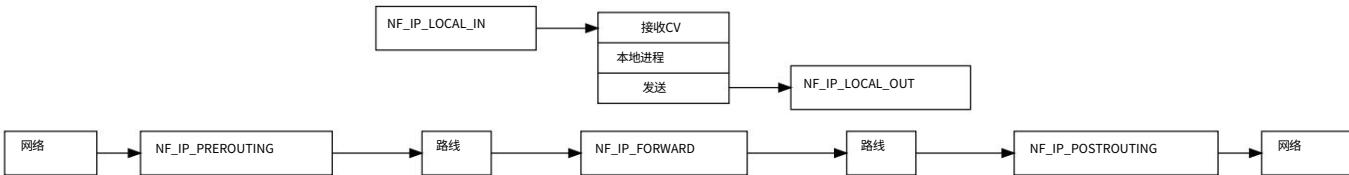
简而言之,这就是 NAT 的工作原理。有关更多详细信息,请参阅 RFC1631。

## Linux 防火墙,概述

### 执行

Linux 防火墙在内核中实现 (自版本 2.3.15 起)。NETFILTER 模块实现数据包过滤规则。用户空间应用程序 iptables 用于配置这些规则。

### Netfilter “钩子”



如上图所示,netfilter 在协议栈中支持五种不同的钩子。这些钩子使我们能够检查和修改 (如果需要)通过内核的每个数据包。

有五个可能的行动:

NF\_ACCEPT 继续正常遍历。

NF\_DROP 丢弃数据包,不继续遍历。

NF\_QUEUE 将数据包排队以供用户空间处理。

NF\_REPEAT 再次调用这个钩子。

NF\_STOLEN 接管 (吸收)数据包但不继续遍历。

## 表和链

默认情况下支持五个链 (netfilter 挂钩) 和三个表。如下图所示,某些链只对某些表有效。

		链			
		预路由输入		向前	输出后路由
#	转发	V			V
	网络地址转换	V			V
	筛选		V	V	V

表 12.1:每个表的有效链

## 过滤表

FILTER 表用于过滤数据包。过滤表包含三个链。 INPUT 链用于所有要发往防火墙本身的数据包。 FORWARD 链用于所有来自防火墙外并发往防火墙外另一台机器的数据包。这些数据包必须流经防火墙。 OUTPUT 链用于防火墙生成的所有数据包。

## NAT 表

NAT 表用于网络地址转换。 NAT 表包含三个链。在任何路由决策发生之前,PREROUTING 链是第一个用于更改传入数据包的链。 OUTPUT 链用于更改防火墙生成的数据包。 POSTROUTING 链是数据包离开防火墙时可以更改的最后一条链。

请注意,流经防火墙的流量通过 PREROUTING 和 POSTROUTING 链,而源自防火墙的流量通过 OUTPUT 和 POSTROUTING 链。

## MANGLE 表

MANGLE 表用于处理数据包。我们可以改变几件事,但我们不能在这里进行伪装或网络地址转换。 mangle 表包含两个链。在任何路由决策发生之前,PREROUTING 链是第一个改变传入数据包的链。 OUTPUT 链用于更改防火墙生成的数据包。

## 连接跟踪:状态防火墙

能够进行连接跟踪的防火墙称为状态防火墙。这些防火墙通过记住源地址和目标地址以及端口号 (所谓的 5 元组) 来跟踪已建立的连接,主要是为了确定有效的返回流量。对于不使用端口号的协议 (例如 ICMP),会保留其他属性。使用状态防火墙时,必须为仅通过单向的流量配置防火墙规则,因为有效的返回流量会自动通过捕获所有规则。

用于连接跟踪的 iptables 选项是 --state 选项。

国家选项

state 此模块与连接跟踪结合使用时,允许访问此数据包的连接跟踪状态。

--state state 其中 state 是要匹配的连接状态的逗号分隔列表。可能的状态有:NEW、ESTABLISHED、RELATED 和 INVALID。

连接跟踪模块

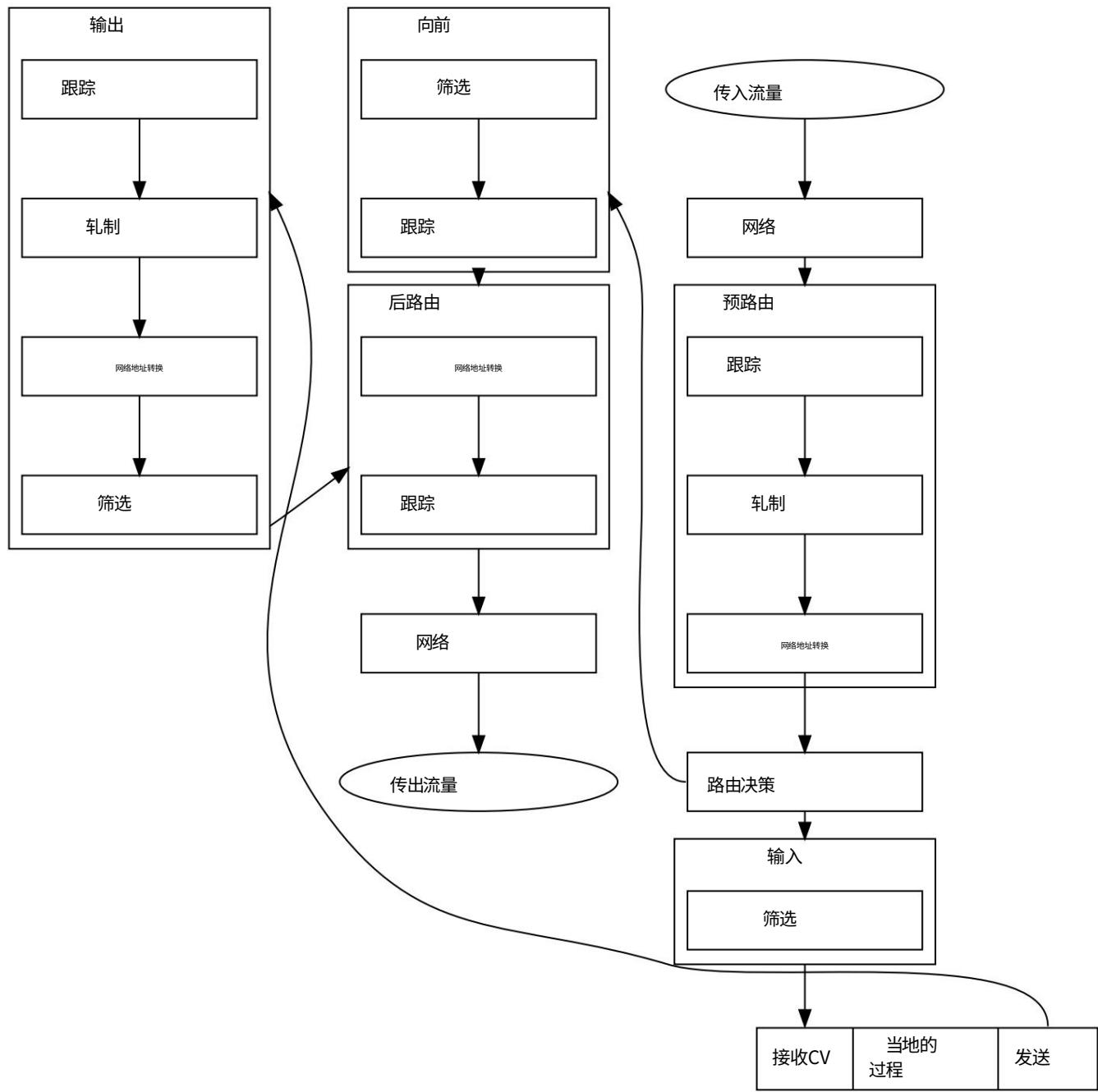
ip\_conntrack 主要的连接跟踪代码。

ip\_conntrack\_ftp 跟踪 ftp 连接所需的附加代码,包括主动和被动。

连接跟踪模块与 PREROUTING、FORWARD、OUTPUT 和 POSTROUTING 挂钩。

钩子、桌子和链条放在一起

将我们到目前为止所讨论的内容放在一张图片中:



添加额外功能

添加目标

每个规则指定如何处理匹配规则的数据包。“做什么”部分称为目标。不匹配链中任何规则的数据包受定义为链策略的隐式目标的约束。标准目标始终存在

是：

ACCEPT 让数据包通过。

DROP 吸收数据包并忘记它。

QUEUE 将数据包传递给用户空间。

RETURN 停止遍历此链并从上一个调用链中的下一条规则继续。如果到达内置链的末尾或内置链中带有目标 RETURN 的规则与数据包匹配,则链策略中指定的目标将决定数据包的命运。

标准发行版中包含许多目标扩展,如果您希望使用它们,必须在内核中启用对这些扩展的支持。有关详细信息,请参阅 iptables 的手册页。这些目标中的大多数都有选项。例如,扩展 LOG 有以下五个选项:“--log-level”、“--log-prefix”、“--log-tcp-sequence”、“--log-tcp-options”、“--log-ip 选项”。有关每个目标的选项的详细信息,请查阅手册页。

大多数Linux发行版中包含的扩展目标模块

LOG 打开匹配数据包的内核日志记录。当为规则设置该选项时,Linux 内核会打印一些信息通过 printk() 在所有匹配的数据包 (例如大多数 IP 标头字段)上。

MARK 这用于设置与数据包关联的 netfilter 标记值。它仅在 mangle 表中有效。

REJECT 用于返回一个错误数据包以响应匹配的数据包;否则,它等同于 DROP。此目标仅在 INPUT、FORWARD 和 OUTPUT 链以及仅由这些链调用的用户定义链中有效。

TOS 这用于设置 IP 标头中的 8 位服务类型字段。它仅在 mangle 表中有效。

MIRROR 这是一个实验性演示目标,它反转 IP 标头中的源和目标字段并重新传输数据包。它仅在 INPUT、FORWARD 和 OUTPUT 链以及仅由这些链调用的用户定义链中有效。

SNAT 该目标仅在 nat 表的 POSTROUTING 链中有效。它指定应修改数据包的源地址 (并且此连接中的所有未来数据包也将被破坏),并且应停止检查规则。

DNAT 该目标仅在 nat 表的 PREROUTING、OUTPUT 和用户定义链 (仅被这些链调用)中有效。它指定应该修改数据包的目标地址 (并且此连接中的所有未来数据包也将被破坏),并且应该停止检查规则。

MASQUERADE 该目标仅在 nat 表的 POSTROUTING 链中有效。它应该只用于动态分配的 IP (拨号)连接:如果你有一个静态 IP 地址,你应该使用 SNAT 目标。伪装相当于指定一个映射到数据包输出接口的 IP 地址,但也具有在接口关闭时忘记连接的效果。当下一次拨号不太可能具有相同的接口地址 (因此任何已建立的连接无论如何都会丢失)时,这是正确的行为。

REDIRECT 此目标仅在 nat 表的 PREROUTING、OUTPUT 和用户定义链 (仅由这些链调用)中有效。它更改目标 IP 地址以将数据包发送到机器本身 (本地生成的数据包映射到 127.0.0.1 地址)。

大多数Linux发行版中包含以下数据包匹配模块

每个规则指定如何处理匹配该规则的数据包。“匹配”部分由数据包匹配模块实现。这些模块中的大多数都支持选项。有关选项的详细信息,请查阅手册页。

tcp 如果指定了 “--protocol tcp”并且没有指定其他匹配项,则会加载这些扩展。

udp 如果指定了 “--protocol udp”并且没有指定其他匹配项,则会加载这些扩展。

icmp 如果指定了 “--protocol icmp”并且没有指定其他匹配项,则加载此扩展。

mac 匹配源MAC地址。它的格式必须为 XX:XX:XX:XX:XX:XX。请注意,这仅对进入来自以太网设备的数据包的 PREROUTING、FORWARD 或 INPUT 链的数据包有意义。

limit 该模块使用令牌桶过滤器以有限的速率匹配:它可以与 LOG 目标结合使用以提供有限的日志记录。使用此扩展名的规则将匹配,直到达到此限制 (除非使用 “!” 标志)。

multiport 此模块匹配一组源或目标端口。最多可以指定 15 个端口。它只能与 -p tcp 或 -p udp 一起使用。

mark 此模块匹配与数据包关联的 netfilter 标记字段（可以使用 MARK 目标设置）。

owner 该模块尝试为本地生成的数据包匹配数据包创建者的各种特征。它仅在 OUTPUT 链中有效，即使如此，某些数据包（例如 ICMP 响应）也可能没有所有者，因此永远不会匹配。

state 此模块与连接跟踪结合使用时，允许访问此数据包的连接跟踪状态。

unclean 该模块不带任何选项，但会尝试匹配格式不正确或异常的数据包。这被视为实验性的。

tos 该模块匹配 IP 报头中的 8 位服务类型字段（即包括优先级）。

#### iptables 选项

-t, --table table 要操作的表（默认值：“filter”）。表格如下：

filter 这是默认表（如果没有传递 -t 选项）。它包含内置链 INPUT（用于发往本地套接字的数据包）、FORWARD（用于通过盒子路由的数据包）和 OUTPUT（用于本地生成的数据包）。nat 当遇到创建新连接的数据包时，会查询此表。它由三个内置函数组成：PREROUTING（用于在数据包进入时立即更改）、OUTPUT（用于在路由之前更改本地生成的数据包）和 POSTROUTING（用于在数据包即将离开时更改）。

mangle 此表用于专门的数据包更改。在内核 2.4.17 之前，它有两个内置链：PREROUTING（用于在路由之前更改传入的数据包）和 OUTPUT（用于在路由之前更改本地生成的数据包）。

从内核 2.4.18 开始，还支持其他三个内置链：INPUT（用于进入盒子本身的数据包）、FORWARD（用于改变通过盒子路由的数据包）和 POSTROUTING（用于改变数据包，因为它们即将到达/出去）。

raw 此表主要用于配置免除连接跟踪和 NOTRACK 目标。它在具有更高优先级的 netfilter 挂钩上注册，因此在 ip\_conntrack 或任何其他 IP 表之前被调用。它提供以下内置链：PREROUTING（用于通过任何网络接口到达的数据包）

OUTPUT（用于本地进程生成的数据包）。

-A, --append chain rule-specification 将一个或多个规则附加到所选链的末尾。

-D, --delete chain rule-specification, -D, --delete chain rulenumber 从所选链中删除一个或多个规则。你可以使用规则规范或规则编号。

-I, --insert chain [rulenumber] rule-specification 在所选链中插入一个或多个规则作为给定的规则编号。

-R, --replace chain rulenumber rule-specification 替换所选链中的规则。

-L, --list [chain] 列出所选链中的所有规则。此选项通常与 -n 选项一起用于数字输出，而不是显示包含主机名、网络名和服务名的输出。此选项还显示每个链的默认策略。

-F, --flush [chain] 刷新选定的链（如果没有给出，则表中的所有链）。

-P, --policy chain target 将与链中任何规则都不匹配的数据包的策略设置为给定目标。此目标可以是用户定义的链或特殊值 ACCEPT、DROP 或 REJECT 之一。

-v, --verbose 详细输出。

--line-numbers 列出规则时，在每个规则的开头添加行号，对应规则在规则中的位置链。

-N, --new-chain chain 通过给定名称创建一个新的用户定义链。必须已经没有该名称的目标。

-X, --delete-chain chain 删除指定的可选用户定义链。必须没有对链的引用。如果有，您必须先删除或替换引用规则，然后才能删除链。如果没有给出参数，它将尝试删除表中的每个非内置链！

每个链都有一个默认策略。这只能在链为空时更改。您可以使用 -L 选项查看默认策略。然后使用 -F 表选项刷新所有规则的链。然后使用 -P 选项设置默认策略。

```
iptables -t filter -L iptables -t filter
-F INPUT iptables -t filter -P INPUT DROP
```

REJECT 不可能作为默认策略。如果您仍然希望 REJECT 作为 (隐含的) 最后一条规则,那么您自己添加一条 REJECT 规则作为链中的最后一条规则。

#### iptables参数

以下参数构成规则规范 (用于添加、删除、插入、替换和追加命令)。

[!] -p, --protocol protocol 要检查的规则或数据包的协议。指定的协议可以是tcp, udp, udplite, icmp, esp, ah, sctp 或特殊关键字 “all”之一,也可以是代表这些协议之一的数值。来自 /etc/protocols 的协议名称也是允许的。一个 “!” 协议反转测试之前的参数。数字零相当于所有。

[!] -s, --source address [/mask][,...] 源规范。地址可以是网络名称、主机名、网络 IP 地址 (带 /mask) 或普通 IP 地址。在将规则提交给内核之前,主机名只会被解析一次。请注意,指定要通过远程查询 (如 DNS) 解析的任何名称是一个非常糟糕的主意。掩码可以是网络掩码或纯数字,指定网络掩码左侧 1 的数量。

因此,掩码 24 相当于 255.255.255.0。可以指定多个地址,但这将扩展到多个规则 (使用 -A 添加时),或者将导致删除多个规则 (使用 -D)。

[!] -d, --destination address [/mask][,...] 目标规范。另见上面的“源地址”参数。

-j, --jump target 这指定了规则的目标;即,如果数据包匹配它该怎么办。

[!] -i, --in-interface name 接收数据包的接口名称。

[!] -o, --out-interface name 将要发送数据包的接口名称。

#### iptables 匹配扩展

iptables 可以使用扩展包匹配模块。它们以两种方式加载:隐式地,当指定 -p 或 --protocol 时,或使用 -m 或 -match 选项,后跟匹配的模块名称。可能有:addrtype, ah, cluster, comment, connbytes, connlimit, connmark, conntrack, dccp, dscp, ecn, esp, hashlimit, helper, icmp, iprange, length, limit, mac, mark, multiport, owner, physdev, pkttype、策略、配额、速率测试、领域、最近、sctp、设置、套接字、状态、统计、字符串、tcp、tcpmss、时间、tos、ttl、u32、udp 和 unclean。其中一些是她解释的:

-m state 此模块与连接跟踪结合使用时,允许访问此数据包的连接跟踪状态。

[!] --state state 这里的 state 是要匹配的连接状态的逗号分隔列表。可能的状态是 INVALID (意味着由于某种原因无法识别数据包), ESTABLISHED (意味着数据包与一个在两个方向上都看到数据包的连接相关联), NEW (意味着数据包已经开始一个新连接,或者与在两个方向上都没有看到数据包的连接相关联) 和 RELATED (意味着数据包正在启动一个新连接,但与现有连接相关联,例如 FTP 数据传输或 ICMP 错误)。

-m time 如果数据包到达时间/日期在给定范围内,则匹配。所有选项都是可选的,但在指定的。它提供以下选项:

-m time --datestart YYYY [-MM [-DD [Thh [:mm [:ss]]]]] 只匹配给定时间,必须是 ISO 8601 “T” 符号。可能的时间范围是 1970-01-01T00:00:00 到 2038-01-19T04:17:07。

-m time --datestop YYYY [-MM [-DD [Thh [:mm [:ss]]]]] 只匹配给定时间,必须是 ISO 8601 “T” 符号。可能的时间范围是 1970-01-01T00:00:00 到 2038-01-19T04:17:07。

-p udp, --protocol udp 如果指定了“--protocol udp”，则可以使用这些扩展。它提供以下选项：

[!] --source-port,--sport port [:port] 源端口或端口范围规范。[!] --destination-port,--dport port [:port] 目标端口或端口范围规范。

-p tcp, --protocol tcp 如果指定了“--protocol tcp”，则加载这些扩展。它提供了以下内容选项：

--source-port, --sport [!] port[:port] 源端口或端口范围规范。这可以是服务名称或端口号。也可以使用格式 port:port 指定包含范围。如果省略第一个端口，则假定为“0”；如果省略最后一个，则假定为“65535”。如果第二个端口大于第一个端口，它们将被交换。

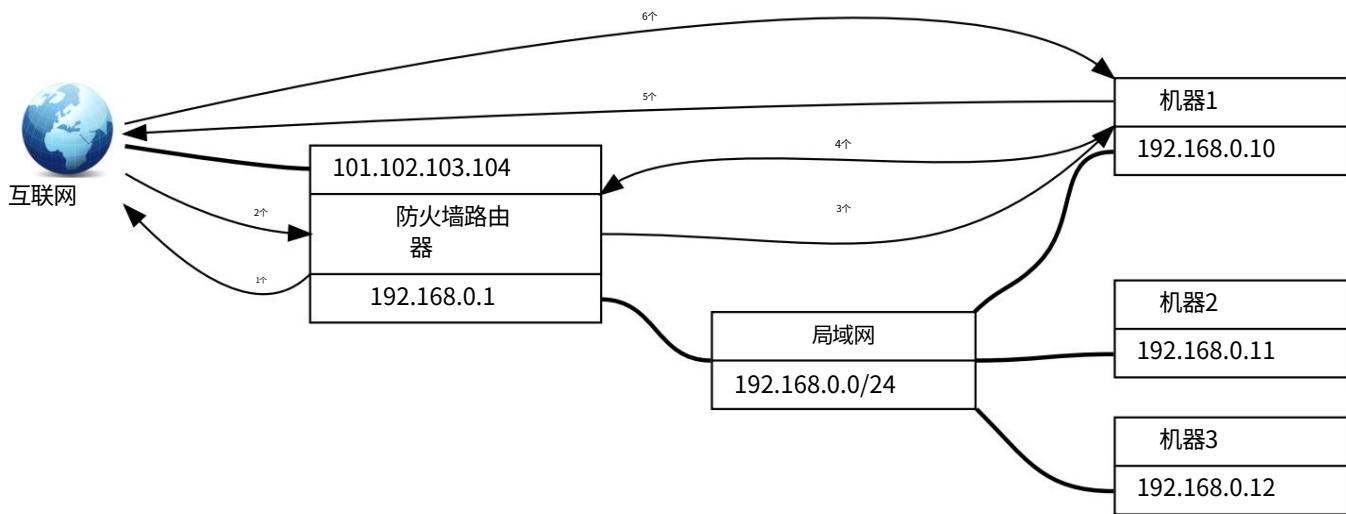
标志 --sport 是此选项的方便别名。--destination-port,--dport [!] port[:port] 目标端口或端口范围规范。标志 --dport 是一个方便的此选项的别名。

--tcp-flags [!] mask comp 当指定 TCP 标志时匹配。第一个参数是我们应该检查的标志，写成逗号分隔的列表，第二个参数是必须设置的逗号分隔的标志列表。标志是：SYN ACK FIN RST URG PSH ALL NONE。因此，命令 iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST SYN 将只匹配设置了 SYN 标志且未设置 ACK、FIN 和 RST 标志的数据包。

[!] --syn 仅匹配设置了 SYN 位且清除了 ACK 和 RST 位的 TCP 数据包。此类数据包用于请求 TCP 连接发起；例如，阻止此类数据包进入接口将阻止传入的 TCP 连接，但传出的 TCP 连接将不受影响。它等同于 --tcp-flags SYN,RST,ACK SYN。如果“！”标志在“--syn”之前，选项的意义是相反的。

#### 公司网络与 IPTABLES

下图显示了公司的网络和流量发起/目的地的可能组合。我们将经历六种可能的情景。



### (1) 防火墙发起的去往Internet的流量

我们在防火墙上运行一个 DNS,它需要能够查询 Internet 上的其他 DNS 服务器（使用 UDP 协议并侦听端口 53）。我们还希望能够使用 ssh（它使用 TCP 协议和端口 22）连接到 Internet 上的其他系统。我们正在参与分布式.net 项目 RC564（RC5 64 位破解），并在防火墙上运行代理服务器（使用 TCP 协议和端口 2064 与密钥服务器通信）。我们希望能够 ping Internet 上的主机（ping 命令使用 ICMP 协议和消息类型 8（echo-request））。防火墙通过接口 eth1 与 Internet 通信。考虑到所有这些，需要的 iptables 命令是：

```
iptables -t filter -A 输出 -o eth1 -p udp --destination-port dns -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port 2064 -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p icmp -m state --state NEW -j echo-request
```

这四个 iptables 命令告诉防火墙允许传出 DNS、SSH、TCP/2064 和 ping 的连接初始化数据包。

### (2) 从 Internet 发起并以防火墙为目标的流量

我们希望能够使用 ssh,它使用 TCP 协议和端口 22,从 Internet 上的其他系统连接到我们的防火墙。需要的 iptables 命令是：

```
iptables -t filter -A 输入 -i eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
```

这个 iptables 命令告诉防火墙允许传入的 SSH 连接初始化数据包。

### (3) 防火墙发起的去往内网的流量

我们希望能够使用 ssh,它使用 TCP 协议和端口 22,从我们的防火墙连接到我们的一台内部机器。需要的 iptables 命令是：

```
iptables -t filter -A 输出 -o eth0 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
```

这个 iptables 命令告诉防火墙允许传出的 SSH 连接初始化数据包发往内部网络上的机器。

### (4) 内网发起的去往防火墙的流量

内部网络上的机器,使用防火墙的 DNS,必须能够使用 SSH 连接到防火墙,正在处理 RC564 密钥,必须能够使用端口 2064 与防火墙上的代理对话并且必须能够 ping 用于系统管理目的的防火墙。需要的 iptables 命令是：

```
iptables -t filter -A 输入 -i eth0 -p udp --destination-port dns -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p tcp --destination-port 2064 -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p icmp -m state --state NEW -j echo-request
```

这四个 iptables 命令告诉防火墙允许传入连接初始化数据包,用于 DNS、SSH、RC564 破解和 ping。

## (5) 内部网络发起的去往Internet的流量

允许从内部网络上的机器连接到 Internet 上的机器。需要的 iptables 命令是：

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state NEW -j ACCEPT
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to-source
10.10.10.104
```

这个 iptables 命令告诉防火墙允许所有传出连接初始化数据包。

## (6) Internet发起的去往内网的流量

这不会发生，因为我们的本地网络使用不能在 Internet 上使用的私有 IP 地址。我们的本地机器在互联网上是不可见的。

为了让我们的一台机器可以从 Internet 访问，我们可以做的是让人们连接到防火墙上的某个端口，并使用 NAT 将他们重定向到内部网络上其中一台机器上的端口。

假设机器 2 有一个 Web 服务器（或其他程序）正在运行，它侦听端口 2345，并且来自 Internet 的人必须能够连接到该程序。由于公司为其内部网络使用私有 IP 地址，因此机器 2 在 Internet 上不可见。这里的解决方案是告诉机器 4 所有来自 Internet 的针对端口 80 的数据都应该路由到机器 2 上的端口 2345。需要的 iptables 命令是：

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --destination-port 80 -j DNAT --to-destination 192.168.0.11:2345
iptables -t filter -A FORWARD -i eth1 -p tcp --destination-port 2345 -m state --state new -j ACCEPT
```

第一行更改目标地址和端口。由于这随后成为针对另一台机器的流量，因此流量必须通过 FORWARD 过滤器。第二行确保允许此流量。

## (!) 流量作为启动流量的结果

到目前为止，我们只指定允许连接发起流量，但这还不够。我们还必须允许 ESTABLISHED 和 RELATED 流量。

让我们告诉防火墙允许所有 ESTABLISHED 和 RELATED 流量，无论类型、接口等如何。我们还必须允许在防火墙 lo 接口上启动流量，否则某些服务（例如缓存 DNS 服务器）将无法工作。我们需要以下 iptables 命令来实现这一点：

```
iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -m state --state NEW -i lo -j ACCEPT
```

请记住，这些最后的规则不会导致安全问题，因为允许的数据包是我们首先接受连接启动这一事实的结果。

## 所有 iptables 命令放在一起

添加内容以从一张白纸开始，并告诉防火墙伪装来自内部网络的数据包，目的是上网可以通过以下命令完成：

```
#####
# 清除 MANGLE,NAT 和 FILTER 表中的所有规则 #####
##### iptables -t mangle -F
##### iptables -t nat iptables -t 过滤器 -F
-F
```

```
#####
# 删除表中所有用户定义的 (非内置的)链 #####
#####
iptables -t mangle -X iptables -t nat
-X
iptables -t 过滤器 -X

#####
# 将所有内置链的所有策略设置为 DROP #####
#####
iptables -P input drop iptables -P forward drop iptables -P output drop
```

```
#####
# (1) 由防火墙发起并发往 Internet 的流量 #
DNS,SSH,RC564,PING
#####
# 允许通过防火墙启动
iptables -t filter -A OUTPUT -o eth1 -p udp --destination-port dns -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p icmp --icmp-type echo-request -m state --state NEW -j ACCEPT
```

```
# 允许传入响应
iptables -t filter -A INPUT -i eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#####
# (2) 从 Internet 发起并发往防火墙的流量 #####
#####
# $SSH#
```

```
# 允许启动 iptables -t filter -A INPUT -i eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
# 允许响应 iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port ssh -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#####
# (3) 由防火墙发起并发往内部网络的流量 #
SSH
#####
# 允许启动
iptables -t filter -A OUTPUT -o eth0 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
```

```
# 允许回应
iptables -t filter -A INPUT -i eth0 -p tcp --destination-port ssh -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#####
# (4) 由内部网络发起并发往防火墙的流量 #####
#####
# DNS,SSH,RC564,PING
```

```
# 允许启动 iptables -t filter -A INPUT -i eth0 -p udp --destination-port dns -m state --state NEW -j ACCEPT
INPUT -i eth0 -p udp --destination-port dns -m state --state NEW -j ACCEPT iptables -t filter -A INPUT -i eth0 -p tcp --
destination-port ssh
```

```

-m state --state NEW -j ACCEPT iptables -t filter -A 输入 -i eth0
-p tcp --destination-port 2064 -m state --state NEW -j ACCEPT iptables -t filter -A 输入 -i eth0 -p icmp --icmp-type echo-request -m state --
state NEW -j 接受 \\
\\

# 允许回应
iptables -t filter -A 输出 -o eth0 -m state --state ESTABLISHED,RELATED -j
ACCEPT \\
\\

#####
# (5) 由内部网络发起并发往外部的流量 #####
##### 我们可以发起的一切都是允许的 #####
\\

# 允许启动一切 iptables -t filter -A FORWARD -i eth0 -o eth1 -m
state --state NEW -j ACCEPT \\
\\

# 允许接收
iptables -t filter -A FORWARD -i eth1 -o eth0 -m 状态 --state ESTABLISHED,RELATED -j ACCEPT \\
\\

#####
# (6) 从互联网发起并发往内部网络的流量 #
##### 所有禁止,除了网络服务器转发到内部机器 #####
\\

# 允许来自防火墙的目标 NAT :80 到内部机器 :2345
iptables -t nat -A PREROUTING -i eth1 -p tcp --destination-port 80 -j DNAT --to-destination 192.168.0.11:2345 iptables -t filter -A FORWARD \\
-i eth1 -p tcp --destination-port 2345 -m 状态 --state 新 -j 接受 \\
\\

#####
##### (! )交通作为发起交通的结果 #####
##### 允许所有来自允许连接的数据包 iptables -t filter -A INPUT -m state --state
ESTABLISHED -j ACCEPT iptables -t filter -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT iptables -t filter -A OUTPUT -m state --state
ESTABLISHED,RELATED -j ACCEPT iptables -t filter -A INPUT -m state --状态新 -i lo -j 接受 \\
\\

#####
# 来自我们内部网络的 MASQUERADE 包,用于互联网
# 这是 SNAT (源 NAT)
#####
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to-source 101.102.103.104 \\
\\

#####
# 启用转发
#####
echo 1 > /proc/sys/net/ipv4/ip_forward

```

## 保存和恢复防火墙规则

使用命令 `iptables-save` (写入 `stdout`) 和 `iptables restore` (从 `stdin` 读取) 可以轻松保存和恢复防火墙规则。假设我们使用文件 `fwrules.saved` 来保存和/或恢复规则,这两个命令是:

```
iptables-save > fwrules.saved iptables-restore < fwrules.saved
```

这些命令可用于在启动时通过将它们放入 SysV 启动脚本来初始化防火墙/路由机器。

### 端口和/或 IP 转发

在本章的第五个例子中,客户端认为它直接连接到外部服务器。中间的路由器透明地路由所有流量并执行 SOURCE NAT 以伪装内部 (私有) 地址。此示例中的关键字是透明度。如果您需要 NAT 将传出流量传输到 Internet,如果您有静态 WAN IP 地址,则可以使用 SNAT (指定 WAN IP 地址)。内核的连接跟踪会在接口关闭和重新启动时跟踪所有连接。使用 MASQUERADE 时情况并非如此。当您拥有动态 WAN IP 地址时,使用 MASQUERADE 是一个更好的主意,因为您不必指定使用的 IP 地址,但可以指定使用的接口。无论该接口上的 IP 地址是什么,它都会应用于所有传出数据包。

除了数据包过滤,防火墙还可以进行端口和 IP 转发。在第六个示例 (使用端口转发) 中,外部客户端将连接到防火墙上的端口。对于客户端,连接将在防火墙上终止,但防火墙知道接收端口上的哪些内部服务器连接应该转发。防火墙会将 DESTINATION NAT 应用到数据包并传递它们。

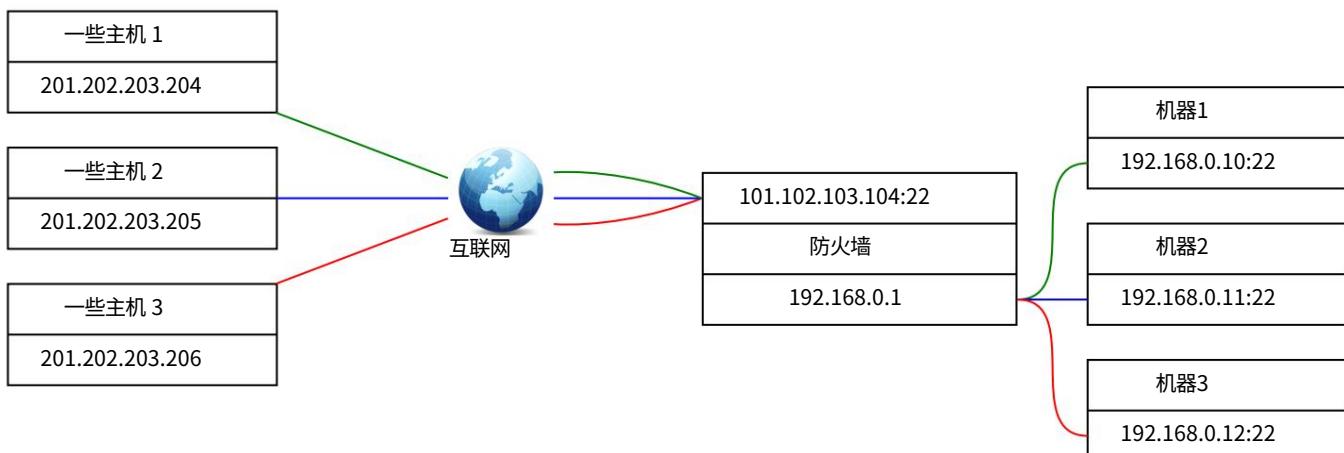
转发的流量也可以应用 SOURCE NAT,但在大多数情况下这是不希望的,因为这会严重妨碍审核接收服务器上的连接。在这种情况下,无法得知原始源地址 所有流量似乎都源自防火墙。

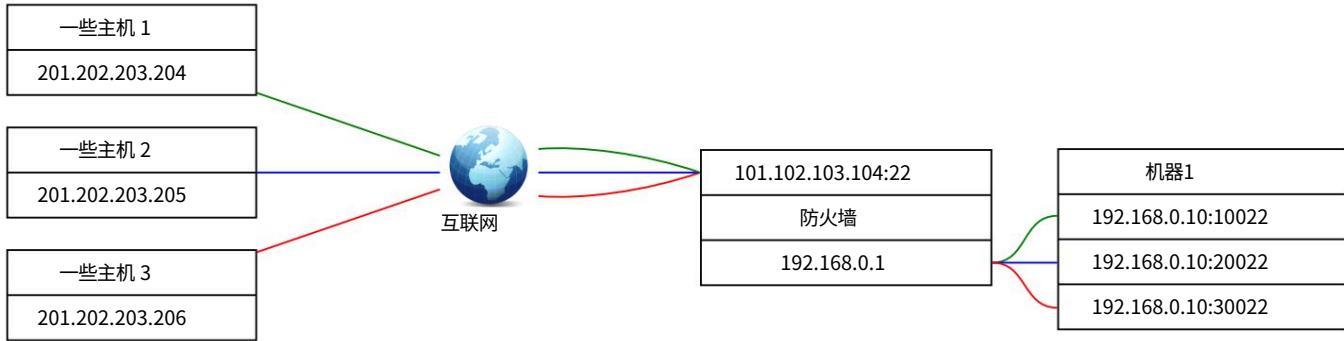
使用端口转发的情况是 (除其他外) :

- 不希望透明 (安全或其他考虑) :
  - 示例 : 通过防火墙可用的一项特定服务在多个内部服务器上运行,例如为我们组织的每个客户提供一个。根据源地址,可以决定将流量转发到哪个服务器。
- 透明是不可能的 (私有 IP 地址不在互联网上路由) :
  - 示例 1: 如果一个组织只有一个外部公共 IP 地址,但必须从在不同服务器上运行的 Internet 访问多个服务,则这些服务将需要对外开放,因为这些服务看似来自一个 IP 地址。
  - 示例 2: 一个内部服务器提供单一服务但侦听不同的端口号 (例如在客户之间进行分隔)。客户端将连接到 IANA 分配的服务端口,并且基于源 IP 地址,防火墙会将流量转发到为该源地址分配的目标端口。
- 缩放注意事项:
  - 示例 : 如果一个组织只有一个外部公共 IP 地址,但必须从在不同服务器上运行的 Internet 访问多项服务,则这些服务将需要向外部提供,因为这些服务看似来自一个 IP 地址。

大多数情况下,端口和/或 IP 转发用于启用从 Internet 到具有私有 IP 地址的服务器的传入连接。

端口转发表示例:





## 拒绝服务 (DoS) 攻击

### 描述

DoS 攻击者滥用 Internet 上的资源是有限的这一事实,并且可以通过拿走 (其中之一)他们的资源来中断服务:存储容量、带宽或处理器容量。这通常是通过“数据包泛洪”来实现的。

可以使用 TCP、ICMP 和 UDP 进行数据包泛洪。当使用 TCP 时,主要使用 SYN、ACK 和 RST 标志。

使用 ICMP 时,使用消息类型 echo request 和 echo reply。这称为“ping 泛洪”。使用 UDP 时,会使用 chargen (字符生成器协议) 和 echo UDP 服务。

此外,两个系统 (A 和 B) 可以通过第三个系统 (C) 相互对抗。系统 C 向系统 A 发送数据包,但将其发送的数据包的源 IP 地址更改为系统 B 的 IP 地址。系统 A 认为数据包来自系统 B,并开始向系统 B 发送回复。这种方法称为“DoS IP 地址欺骗”。

查看站点<http://www.cert.org/>以获得 DoS 和 DDoS (分布式 DoS) 攻击的完整历史记录。并查看描述网络入口过滤的 RFC2827,这是一种防止 IP 地址欺骗的方法。这不能防止 DoS 攻击,但可以追踪攻击者的真实 IP 地址。

### 预防

如果不断开与 Internet 的连接,就不可能完全防止 DoS 攻击。将 DoS 和 DDoS 攻击的影响降到最低的方法是对防火墙应用一些数据包过滤和速率限制规则。

## 使用 /proc/sys/net/ipv4 (sysctl) 来防止简单的 DOS 攻击

内核文档描述了以下 sysctl 选项以防止简单的 DoS 攻击:

- `tcp_max_orphans` - 整数:
  - 系统持有的未附加到任何用户文件句柄的 TCP 套接字的最大数量。如果超过此数量,孤立连接将立即重置并打印警告。
- `tcp_max_tw_buckets` - 整数:
  - 系统同时持有的最大时间等待套接字数。如果超过这个数字,时间等待套接字是即时的  
diateely 销毁并打印警告。
- `rp_filter` - 整数:
  - 0 - 无来源验证。
  - 1 - RFC3704 严格反向路径中定义的严格模式:每个传入数据包都根据 FIB 进行测试,如果接口不是最佳反向路径,则数据包检查将失败。默认情况下丢弃失败的数据包。
  - 2 - RFC3704 松散反向路径中定义的松散模式:每个传入数据包的源地址也针对 FIB 进行测试,如果无法通过任何接口访问源地址,则数据包检查将失败。

## 路由

在使用动态路由的环境中,可以使用路由守护进程。 routed 守护进程管理内核中的路由表。 routed 守护进程只实现了 RIP (路由信息协议)协议。当您希望动态路由其他类型的协议时,可以使用 gated 代替。 Gated 是一个经典的路由守护进程,支持 RIPv2、RIPng、OSPF、OSPF6、BGP4+ 和 BGP4-。

routerd 守护进程查找直接连接的主机和网络的接口,这些主机和网络已配置到系统中并标记为已启动。 (使用 ifconfig 命令将网络标记为正常运行。)如果存在多个接口,路由守护程序假定本地主机在网络之间转发数据包。 routerd 守护程序在每个接口上传输 RIP 请求数据包,在接口支持时使用广播消息。

routerd 守护进程然后侦听来自其他主机的 RIP 路由请求和响应数据包。当 routerd 守护进程向其他主机提供 RIP 信息时,它每 30 秒向所有直接连接的主机和网络发送一次 RIP 更新数据包 (包含其路由表的副本)。

当路由守护程序接收到路由信息协议 (RIP) 请求包以提供 RIP 路由信息时,路由守护程序生成响应包形式的回复。响应数据包基于内核路由表中维护的信息,并包含已知路由列表。每条路由都标有跳数度量,即源网络和目标网络之间的网关跳数。每条路由的指标是相对于发送主机的。 16 或更大的指标被认为是无限的或无法达到的。

## 何时使用路由

如果到某个目的地有多个可能的路径,并且您希望自动选择到该目的地的替代路线 (以防由于某种原因到该目的地的默认路线不可用),路由程序可以自动为您完成此操作。

## 用于管理路由表的工具、命令和实用程序

### 路线

route 操作内核的 IP 路由表。它的主要用途是在使用 ifconfig 命令配置后通过接口设置到特定主机或网络的静态路由。 (最近的 Linux 发行版使用 ip route 而不是 route。)

**概要 (最重要的观点) :**

路线

```
route [-v] add [-net|-host] target [netmask NM] [gw GW] [metric N] [[dev] If] route [-v] del [-net|-host] target [netmask NM] [gw GW] [公制 N] [[dev] 如果]
```

路由本身,不带任何参数,显示路由表。 -ee 选项将生成一条很长的行,其中包含路由表中的所有参数。

-v 选择详细操作。

-net|-host 目标是网络或主机。

netmask Nm 添加网络路由时,要使用的网络掩码。

gw GW 通过网关路由数据包。指定的网关必须首先可达。

metric N 将路由表中的 metric 字段 (由路由守护程序使用) 设置为 N。

dev If 强制路由与指定设备相关联。如果 dev If 是命令行上的最后一个选项,则可以省略 dev 一词,因为它是默认值。否则路由修饰符 (metric、netmask、gw、dev) 的顺序无关紧要。

例子:

```
路由添加-net 192.168.10.0 网络掩码 255.255.255.0 dev eth0
路由添加-net 192.168.20.0 网络掩码 255.255.255.0 gw 192.168.1.10 路由添加默认 gw 192.168.1.1 eth1
```

内核路由表的输出组织在以下列中：

Destination 目标网络或目标主机。

网关 网关地址或“\*”（如果未设置）。

Genmask 目标网络的网络掩码；“255.255.255.255”代表主机目的地，“0.0.0.0”代表默认路由。

标志 可能的标志包括：U - 路由已启

- 动 H - 目标是主机 G - 使用网
- 关 R - 为动态路由恢复路由 D
- 由守护进程或重定向动态安
- 装 M - 从路由守护进程或重定向修改 C - 缓存条
- 目！ - 拒绝路线

度量 到目标的“距离”（通常以跳数计算）。

Ref 对该路由的引用数。

Iface 将此路由的数据包发送到的接口。

例子：

内核IP路由表					
目的地	网关	基因掩码	标志公制参考		使用界面
192.168.20.0	*	255.255.255.0 你	0	0	0 eth0
链接本地	*	255.255.0.0	ü	1002 0	0 eth0
默认	192.168.20.1	0.0.0.0	优格	0	0 eth0

网络统计

打印网络连接、路由表、接口统计信息、伪装连接和多播成员资格。netstat 有很多可能的选项，但最常用的选项如下：

```
# 网络统计-rn
内核 IP 路由表 Destination Gateway 0.0.0.0
                                基因掩码          标记 MSS 窗口 irtt Iface
192.168.20.0                  255.255.255.0 你          0 0          0 eth0
169.254.0.0                   0.0.0.0           255.255.0.0          ü 0 0          0 eth0
0.0.0.0                         192.168.20.1       0.0.0.0          优格 0 0          0 eth0
```

netstat -rn 还将显示路由表。-r 选项将显示路由表，其中 -n 将阻止将 IP 地址和网络解析为名称。请查看手册页以获取更多有趣的选项。

ip6表

ip6tables 是 iptables 的 ipv6 等价物。除了使用 128 位地址而不是 32 位地址外，其语法与其对应的 ipv4 相同。

以下示例允许 ICMPv6：

```
ip6tables -A 输入 -p icmpv6 -j 接受 ip6tables -A 输出 -p icmpv6 -j 接受
```

iptables 和 ip6tables 可以同时使用。有关详细信息，请参阅 [iptables\(8\)](#) 联机帮助页。

## 管理 FTP 服务器 (212.2)

考生应该能够配置用于匿名下载和上传的 FTP 服务器。此目标包括配置用户访问权限,以及在允许匿名上传时要采取的预防措施。

### 关键知识领域

Pure-FTPd 和 vsftpd 的配置文件、工具和实用程序

对 ProFTPD 的认识

了解被动与主动 FTP 连接

### 条款和实用程序

- vsftpd.conf
- 重要的 Pure-FTPd 命令行选项

### FTP连接方式

FTP 是一种使用两个端口进行通信的服务。端口 21 用于命令端口（也称为控制端口）,端口 20 用于数据端口。FTP 有两种模式,主动FTP和被动FTP。这些模式在启动连接的方式上有所不同。

在主动模式下,客户端发起控制连接,服务器发起数据连接。在被动模式下,客户端发起两个连接。

### 活动模式

在主动模式下,客户端启动 FTP 会话。这是通过启动从非特权端口 (>1023) 到服务器上的端口 21 的控制连接来完成的。客户端向服务器发送客户端将侦听数据连接的 IP 地址和端口号。通常此端口是客户端上使用的控制连接端口之上的下一个端口。服务器向客户端命令端口发送 ACK,并主动打开从端口 20 到客户端的数据连接。客户端在数据连接上发回 ACK。

#### 活动模式示例：

- 客户端打开从客户端端口1050 到服务器端口21 的命令通道。
- 客户端向服务器发送端口1051 (1050 + 1),服务器在命令通道上确认。
- 服务器打开一条从服务器端口20到客户端端口1051的数据通道。
- 客户端在数据通道上确认。

### 被动模式

在客户端位于防火墙后面且无法接受传入 TCP 连接的情况下,可以使用被动模式。

在被动模式下,客户端启动 FTP 会话。这是通过启动从非特权端口 (>1023) 到服务器上的端口 21 的控制连接来完成的。在这种模式下,客户端向服务器发送 PASV 命令并接收 IP 地址和端口号作为返回。服务器回复 PORT XXXX 其中 XXXX 是服务器侦听数据连接并被动等待数据连接的非特权端口。客户端打开从控制连接端口上方的下一个端口到服务器上 PORT 回复中指定的端口的数据连接。服务器在数据连接上向客户端发回 ACK。

#### 被动模式示例：

- 客户端打开从客户端端口1050到服务器端口21的命令通道。
- 客户端在命令通道上向服务器发送 PASV 命令。
- 服务器在开始侦听该端口后发回（在命令通道上）PORT 1234。
- 客户端打开客户端1050到服务器端口1234的数据通道。
- 服务器在数据通道上确认。

## 通过防火墙启用连接

要在使用 iptables 时启用被动 FTP 连接，必须将“ip\_conntrack\_ftp”模块加载到防火墙中，并且必须允许状态为“related”的连接。

## vsftpd

vsftpd（非常安全的 FTP 守护程序）是一个非常流行、多功能、快速且安全的 FTP 服务器。

### 匿名上传和下载的最小配置示例

#### 示例 vsftpd.conf

```
# 如果启用,vsftpd 将以独立模式运行。这意味着 # vsftpd 不能从某种 inetc 运行。相反,#vsftpd 可执行文件直接运行一次。然后 vsftpd  
本身将负责监听和处理传入的连接。  
  
# 默认值:否  
listen=否  
  
# 控制是否允许本地登录。如果启用 ,#/etc/passwd 中的普通用户帐户（或任何你的 PAM 配置 # 引用）可以用于登录。这必须启用任何非  
匿名登录才能工作,包括虚拟用户。  
  
# 默认值:否  
local_enable=是  
  
# 这控制是否允许任何更改文件系统的 FTP 命令。这些命令是:STOR,DELE,RNFR,#RNTO,MKD,RMD,APPE 和 SITE。  
  
# 默认值:否  
write_enable=YES  
  
# 控制是否允许匿名登录。 # 如果启用,用户名 ftp 和匿名都被识别为 # 匿名登录。  
  
# 默认值:是  
anonymous_enable=YES  
  
# 此选项代表 vsftpd 将尝试在匿名登录后更改到的目录。失败被默默地忽略。  
  
# 默认值: (无)anon_root=/  
var/ftp/pub  
  
# 如果设置为 YES,匿名用户将被允许上传文件 # 在某些条件下。为此,必须激活选项 # write_enable,并且匿名 ftp 用户必须 # 对所需的上  
传位置具有写入权限。这个设置
```

```
#也是虚拟用户上传需要的;默认情况下,虚拟 # 用户被视为匿名 (即最大限制)特权。  
# 默认值:否  
anon_upload_enable=YES  
  
# 启用后,匿名用户将只被允许下载 # 世界可读的文件。这是认识到 ftp # 用户可能拥有文件,尤其是在存在上传的情况下。  
  
# 默认值:是  
anon_world_readable_only=NO
```

创建 ftp 用户：

```
useradd --home /var/ftp --shell /bin/false ftp
```

创建 FTP 目录：

```
mkdir -p --mode 733 /var/ftp/pub/incoming
```

设置 inetc 以侦听 FTP 流量并启动 vsftpd。将以下行添加到 /etc/inetc.conf:

FTP流	tcp nowait root /usr/sbin/tcpd /usr/sbin/vsftpd
------	---

重新加载 inetc 守护进程。

在线 HTML 版本的手册页列出了所有 vsftpd 配置选项,可以在以下位置找到：[vsftpd.conf](#) 的手册页。

当只允许匿名用户上传文件时,例如,为了将文件发送给远程支持进行分析,请确保该目录对所有者、root 可读写,并且可写但对组成员和其他人不可读。这允许匿名用户写入传入目录但不能更改它。

## 纯FTPD

Pure-FTPD 是一个高度灵活、安全和快速的 FTP 服务器。

### 配置

与许多守护进程不同,Pure-FTPD 不读取任何配置文件 (使用时 LDAP 和 SQL 除外)。相反,它使用命令行选项。为了方便起见,提供了一个包装器,它读取配置文件并使用正确的命令行选项启动 Pure-FTPD。

pure-ftpd 的具体配置选项可以在：[Pure-FTPD 配置文件中找到。](#)

### 重要的命令行选项

如果你想监听非标准端口上的传入连接,只需附加 -S 和端口号:

```
/usr/local/sbin/pure-ftpd -S 42
```

如果您的系统有许多 IP 地址,并且您希望 FTP 服务器只能通过这些地址之一访问,比如 192.168.0.42,只需使用以下命令:

```
/usr/local/sbin/pure-ftpd -S 192.168.0.42,21
```

---

### 备注21

端口号可以省略,因为这是默认端口。

---

要限制同时连接的数量,请使用 -c 选项:

```
/usr/local/sbin/pure-ftpd -c 50 &
```

### 匿名上传和下载的最小配置示例

#### 创建 ftp 用户：

```
useradd --home /var/ftp --shell /bin/false ftp
```

#### 创建具有正确权限的 ftp 目录结构：

```
# 设置适当的权限以禁止写入 mkdir -p --mode 555 /var/ftp mkdir -p --mode 555 /var/ftp/pub  
  
# 设置适当的权限以启用写入 mkdir -p --mode 755 /var/ftp/pub/incoming
```

#### 更改所有权：

```
chown -R ftp:ftp /变量/ftp/
```

192552	0 dr-xr-xr-x 3 ftp 0 dr-xr-xr-x 3 ftp 0 drwxr-	FTP	11 年 3 月 16 日 11:54 /var/ftp
192588	xr-x 2 ftp	FTP	11 年 3 月 8 日 11:07 /var/ftp/pub
192589		FTP	11 年 3 月 8 日 11:55 /var/ftp/pub/incoming

#### 设置 inetd 以侦听 FTP 流量并启动 pure-ftpd。将以下行添加到 /etc/inetd.conf：

```
ftp流tcpnowaitroot/usr/sbin/tcpd/usr/sbin/pure-ftpd-e
```

#### 重新加载 inetd 守护进程：

```
killall -HUP inetd
```

或者

```
kill -HUP $(cat /var/run/inetd.pid)
```

### 其他 FTP 服务器

在 Linux 系统上有许多可用和正在使用的 FTP 服务器。上述服务器的一些替代品是 :wu-ftpd 和 ProFTPD。

#### ProFTPD

ProFTPD - 专业的可配置、安全的文件传输协议服务器。

概要

```
proftpd [-hlntv] [-c 配置文件] [-d 调试级别] [-p 0|1]
```

proftpd 是专业文件传输协议 (FTP) 服务器守护进程。每次建立到 FTP 服务的连接时，该服务器可能会被互联网“超级服务器”inetd(8) 调用，或者它可以作为一个独立的守护进程运行。

当 proftpd 以独立模式运行并收到 SIGHUP 时，它将重新读取其配置文件。当在没有 -n 选项的情况下以独立模式运行时，主 proftpd 守护程序将其进程 ID 写入 /var/run/proftpd.pid 以便轻松了解要 SIGHUP 的进程。

有关此 ftp 服务器的详细信息，请参阅 proftpd 的手册页。可以在以下位置找到详细信息：[ProFTPD 项目](#)。

### 安全外壳 (SSH) (212.3)

考生应该能够配置和保护 SSH 守护进程。此目标包括为用户管理密钥和配置 SSH。考生还应该能够通过 SSH 转发应用程序协议并管理 SSH 登录。

## 关键知识领域

- OpenSSH 配置文件、工具和实用程序
- 超级用户和普通用户的登录限制
- 管理和使用服务器和客户端密钥以使用和不使用密码登录
- 使用来自多个主机的多个连接来防止在配置更改后丢失与远程主机的连接

## 条款和实用程序

- ssh
- sshd
- /etc/ssh/sshd\_config
- /etc/ssh/
- 私钥和公钥文件
- PermitRootLogin、PubkeyAuthentication、AllowUsers、PasswordAuthentication、协议

## SSH 客户端和服务器

ssh 是一个客户端程序,用于登录远程机器并在远程机器上执行命令。

sshd 是 ssh 的服务器 (守护进程)程序。

这两个程序共同取代了 rlogin 和 rsh,在不安全网络上的两个不受信任的主机之间提供安全的加密通信。

要通过不安全的网络复制文件,可以使用 scp 命令。 Scp 代表安全复制。它使用 ssh 进行数据传输。

SSH (Secure SHell) 使用数字密钥进行数据加密和身份验证。

## 密钥及其用途

使用两种类型的密钥:主机密钥和用户密钥。这些将在接下来的段落中讨论。

### 主机密钥

SSH 协议版本 1 和 2 在所谓的前向安全性方面存在细微差别。

SSH 协议版本:

SSH 协议版本 1 此版本仅支持 RSA 密钥。每个节点都有一个主机密钥 (通常为 2048 位) 来标识它。

当守护进程启动时,会生成一个额外的服务器密钥 (通常为 768 位)。它不存储在磁盘上,使用时每小时重新创建一次。

当客户端连接时,服务器守护程序使用其公共主机和服务器密钥进行响应。客户端将 RSA 主机密钥与自己的数据库进行比较,以验证它没有更改。然后客户端生成一个 256 位的随机数。

它使用主机和服务器密钥加密这个随机数,并将加密后的数字发送到服务器。然后双方都使用这个随机数作为密钥,用于加密会话中所有进一步的通信。会话的其余部分使用传统密码进行加密,目前是 Blowfish 或 3DES,默认情况下使用 3DES。客户端从服务器提供的算法中选择加密算法。

SSH 协议版本 2 此协议版本为默认版本,它支持 DSA、ECDSA 和 RSA 密钥。前向安全性是通过 Diffie-Hellman 密钥协议提供的。此密钥协议导致共享会话密钥。

会话的其余部分使用对称密码加密,目前为 128 位 AES、Blowfish、3DES、CAST128、Arcfour、192 位 AES 或 256 位 AES。客户端从服务器提供的算法中选择要使用的加密算法。此外,会话完整性是通过加密消息身份验证代码 (hmac-md5、hmac-sha1、umac-64、umac-128、hmac-ripemd160、hmac-sha2-256 或 hmac-sha2-512) 提供的。

最后,在协议的两个版本中,服务器和客户端进入身份验证对话。客户端尝试使用基于主机的身份验证、公钥身份验证、挑战-响应身份验证或密码身份验证来验证自己。

如果可能,请选择 SSH 协议版本 2 作为唯一要使用的版本。

#### 用户密钥,公钥和私钥

ssh 自动实现 RSA 认证机制。用户通过运行 ssh-keygen 创建 RSA 密钥对。

这会将私钥存储在用户主目录的 \$HOME/.ssh/id\_rsa 中,将公钥存储在 \$HOME/.ssh/id\_rsa.pub 中。然后用户应该将 id\_rsa.pub 复制到远程计算机上他的主目录中的 \$HOME/.ssh/authorized\_keys 文件中:

```
# cat id_rsa.pub >> ~/.ssh/authorized_keys
```

authorized\_keys 文件每行只有一个密钥,可能会很长。此后,用户无需提供密码即可登录。除了使用 rsa,也可以使用 dsa。键的名称反映了您创建的键的种类。确保 ~/.ssh 目录及其中的文件具有适当的权限。例如,在 .ssh 目录上使用 700,在其中的文件上使用 600。

如果不这样做,在某些情况下您将无法使用数字密钥登录。

#### 配置 sshd

可以使用命令行选项或编辑配置文件 /etc/ssh/sshd\_config 来配置 sshd。

-4 强制 sshd 仅使用 IPv4 地址。

-6 强制 sshd 仅使用 IPv6 地址。

-b bits 指定临时协议版本 1 服务器密钥中的位数 (默认为 1024)。

-C connection\_spec 指定用于 -T 扩展测试模式的连接参数。

-D 指定此选项时,sshd 不会分离,也不会成为守护进程。这样可以轻松监控 sshd。

-d 调试模式 服务器将详细的调试输出发送到系统日志,并且不会将自身置于后台。

-e 指定此选项时,sshd 会将输出发送到标准错误而不是系统日志。

-f config\_file 指定配置文件的名称。默认为 /etc/ssh/sshd\_config。如果没有,sshd 拒绝启动  
配置文件。

-g login\_grace\_time 为客户端提供宽限时间来验证自己 (默认 120 秒)。

-h host\_key\_file 指定从中读取主机密钥的文件。

-i 指定 sshd 从 inetd 运行。sshd 通常不从 inetd 运行,因为它需要生成服务器密钥  
在它可以响应客户端之前,这可能需要几十秒。

-k key\_gen\_time 指定临时协议版本 1 服务器密钥重新生成的频率 (默认 3600 秒,或一  
小时)。

-o option 可用于以配置文件中使用的格式提供选项。这对于指定选项很有用  
没有单独的命令行标志。

-p port 指定服务器侦听连接的端口 (默认为 22)。允许多个端口选项。

-q 安静模式。不会向系统日志发送任何内容。通常是每个连接的开始、验证和终止被记录。

-T 扩展测试模式。检查配置文件的有效性，将生效的配置输出到stdout，然后退出。

-t 测试模式。只检查配置文件的有效性和密钥的完整性。这对于可靠地更新 sshd 很有用因为配置选项可能会改变。

-u len 此选项用于指定 utmp 结构中保存远程主机名的字段的大小。

/etc/ssh/sshd\_config 目录中的 sshd 配置文件也可用于配置 sshd。该文件应该只能由 root 用户写入，但建议（虽然不是必需的）它是所有人都可读的。

#### 允许或拒绝 root 登录

允许或拒绝 root 登录是通过将配置文件中的关键字 PermitRootLogin 设置为适当的值来完成的。为了让某人更难获得完全访问权限，您不应该允许 root 登录。没有远程 root 访问的可能性，想要在远程服务器上获得 root 访问的人必须首先通过普通用户获得访问权限。这是一个额外的安全层。

PermitRootLogin 的可能值：

yes 这是默认设置。当设置为 yes 时，root 可以使用 ssh 登录。

no 当设置为 no 时，root 无法使用 ssh 登录。

without-password 这意味着对 root 禁用密码验证。

forced-commands-only 表示 root 只能使用 ssh 通过公钥认证登录系统，执行命令行给出的命令。允许的命令必须添加到 ~/.ssh/authorized\_keys 文件中的公钥。例如，如果您在 SERVERA 上的公钥行的开头添加 command= /bin/date，则使用该公钥登录时只允许 date 命令。如果您执行 ssh root@SERVERA，那么只会在远程系统上执行 date 命令。（即使通常不允许 root 登录，这对于进行远程备份也可能很有用。）请阅读手册页以获取更多信息。

#### 允许或拒绝非 root 登录

有许多关键字可用于影响 sshd 与登录相关的行为。

SSHD 关键字：

AllowUsers 此关键字后跟用户名列表，以空格分隔。仅允许与其中一种模式匹配的用户名登录。您可以使用 “\*” 和 “?” 作为模式中的通配符。

DenyUsers 此关键字后跟用户名列表，以空格分隔。与其中一种模式匹配的用户名无法登录。您可以使用 “\*” 和 “?” 作为模式中的通配符。

AllowGroups 该关键字后跟一组组名称，以空格分隔。仅允许属于与其中一种模式匹配的一个或多个组成员的用户登录。您可以使用 “\*” 和 “?” 作为模式中的通配符。

DenyGroups 此关键字后跟一组组名称，以空格分隔。不允许以下用户登录  
匹配其中一种模式的一个或多个组的成员。您可以使用 “\*” 和 “?” 作为模式中的通配符。

PasswordAuthentication 指定是否允许密码验证。默认值为“是”。

协议 指定 sshd 支持的协议版本。可能的值为“1”和“2”。多个版本必须以逗号分隔。默认值为“2,1”。请注意，协议列表的顺序并不表示偏好，因为客户端会在服务器提供的多个协议版本中进行选择。指定“2,1”等同于“1,2”。除非有使用协议版本“1”的严肃争论，否则只使用版本“2”，因为它更安全。

UsePAM 启用可插入身份验证模块接口。如果设置为“yes”，它将启用 PAM 身份验证，使用 ChallengeResponseAuthentication 和 PasswordAuthentication 除了 PAM 帐户和会话模块处理所有身份验证类型。默认为“是”。

ChallengeResponseAuthentication 指定是否允许质询-响应身份验证（例如，通过 PAM 或通过 login.conf(5) 中支持的身份验证样式）。默认为“是”。

如果要完全禁用基于密码的登录，应将以下 sshd\_config 设置设置为 no：

- 密码认证
- 质询响应认证
- 使用PAM

#### 启用或禁用 X 转发

本主题保留在这里是为了向您提供有关该主题的一些信息，但它不再位于 lpi.org 的关键知识领域中。

有许多关键字可用于影响 sshd 与 X Windows 系统相关的行为。

X WINDOWS 关键字(SSHD\_CONFIG)：

X11Forwarding X11 转发是一种机制，程序在一台机器上运行，X Windows 输出显示在另一台机器上。命令 ssh -X remote 会将 server-shell 中的 DISPLAY 设置为 localhost:num:0，这实际上是映射回客户端上下文中的原始 DISPLAY 的隧道端点。此隧道使用 ssh 进行保护。 X11Forwarding 可以设置为 yes 或 no。默认为否。

X11DisplayOffset 这指定了可用于 sshd 的 X11 转发的第一个显示编号。这可以防止 sshd 干扰真实服务器。默认值为 10。

XAuthLocation 这指定了 xauth 命令的完全限定位置。默认位置是 /usr/bin/X11/xauth。 xauth 用于编辑和显示连接 X 服务器时使用的授权信息。

例如，如果您使用 ssh your\_account@machineB 从机器 A 连接到机器 B 并启动 xterm，该进程将在机器 B 上运行并且 X 输出将通过 SSH 隧道传输到机器 A。要显示终端，机器 B 将连接到由 SSH 为 X11 转发打开的本地主机上的显示。 SSH 会将 X 连接转发到机器 A 上的 X 服务器，它将在本地显示中显示。由于 X 输出看似是在本地创建的，因此无需更改 xhost 设置即可显示内容。

要启用 X11 转发，还必须在客户端启用它。 ForwardX11 必须在客户端的 SSH 配置中设置为“yes”，或者在启动连接时在命令行中设置。

#### 无密码认证

使用 SSH 有不同的身份验证方法。一种方法是使用公钥/私钥对进行身份验证。

如果启用公钥身份验证，用户无需使用密码即可登录。当尝试使用密钥对进行身份验证时，将要求用户输入密码（或查询 ssh\_agent）。此外，可以使用无密码密钥（例如，用于自动会话）。

使用无密码密钥的注意事项：因为使用无密码密钥不需要密码，任何有权访问该密钥的人都可以使用它。这会带来严重的安全风险。应通过仅使用强制命令允许这些连接，并且最好来自一组有限的客户端来减少使用无密码密钥的影响（请参阅 man authorized\_keys）。

#### 无密码验证关键字：

PubkeyAuthentication 该参数指定是否允许公钥认证。默认值为“是”。注意

此选项仅适用于协议版本 2。

## ssh代理

此目标已移至 LPIC-1。

ssh-agent 是一个程序,用于保存用于公钥身份验证 (RSA,DSA) 的私钥。这个想法是 ssh-agent 在 X 会话或登录会话开始时启动,所有其他窗口或程序都作为 ssh-agent 程序的客户端启动。通过使用环境变量,可以定位代理并在使用 ssh 登录到其他机器时自动使用它。

### 启用代理转发

在 ssh\_config (系统范围或 \$HOME/.ssh/config 中) 将 ForwardAgent 设置为 “yes”。

### 登录会话

将以下两行添加到您的 \$HOME/.bash\_profile 文件或等效文件 (取决于您使用的 shell),以便无需每次都输入密码即可登录:

```
eval `ssh-agent` & ssh-add
```

eval `ssh-agent` 设置了一些环境变量。事实上,ssh-agent 返回设置它们所需的字符串,eval 确保它们得到设置。

如前所述,不带参数的 ssh-add 命令读取包含私钥的文件 \$HOME/.ssh/id\_rsa (或 id\_dsa) 的内容。如有必要,系统将提示您输入密码。

当用户从系统注销时,程序 ssh-agent 必须终止。要确保这会自动发生,请将以下行添加到您的 .bash\_logout 文件或等效文件中,具体取决于您使用的 shell:

```
ssh-agent -k
```

当前代理的进程 ID 是通过检查环境变量 \$SSH\_AGENT\_PID 的内容来确定的,该变量已由 eval `ssh-agent` 命令设置。

### 使用 ssh-agent 启用 X 会话

有几种方法可以做到这一点,具体取决于 X 的启动方式和您使用的显示管理器。

如果您使用 startx 从命令行启动 X,您可以键入 ssh-agent startx,在 X 中打开一个终端窗口并键入 ssh-add,这将提示您输入密码并加载您的密钥。

## 使用端口映射通过 ssh 隧道化应用程序协议

### 描述

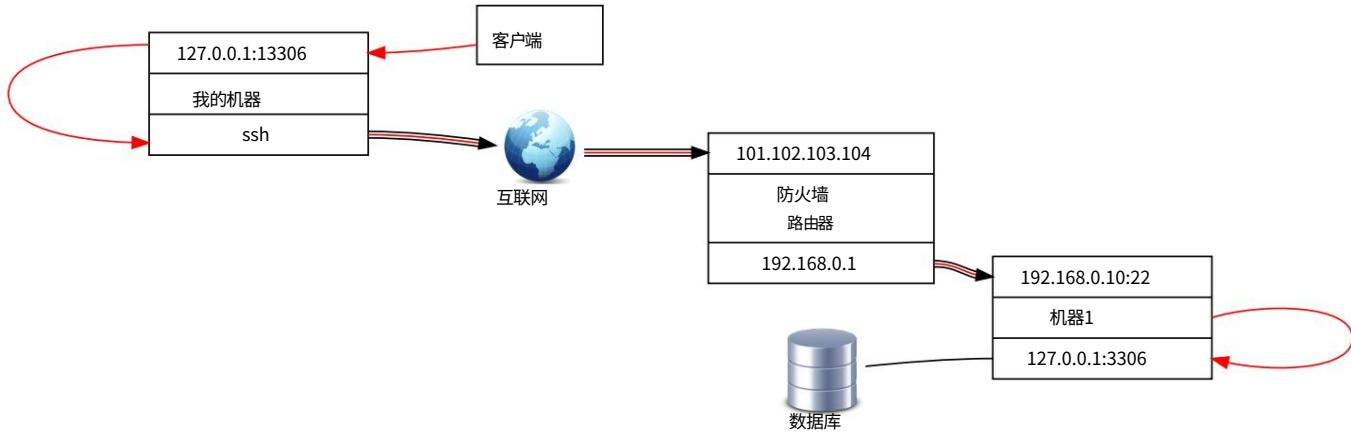
使用端口转发,SSH 将绑定到一个端口,并通过 SSH 连接隧道传输指向该端口的所有流量。流量被转发到 SSH 连接另一端的主机。远程主机可能是 SSH 连接终止的服务器,但流量也可以转发到另一台主机。端口转发可以双向配置:本地到远程,也可以远程到本地。

语法是:

```
ssh -R|L [绑定地址:]端口:主机:主机端口[用户名@]主机名[命令]
```

## 例子

作为示例（示例 1），请考虑图片中显示的情况。我们在 MyMachine 上工作，我们想连接到机器 1 上的 mysql 服务器。防火墙不允许 sql 连接。防火墙配置了端口转发，因此来自 Internet 的端口 22 上的传入流量将转发到 Machine1 上的端口 22。John 是 Machine1 上的用户。



首先通过使用“-L”选项运行 ssh 打开隧道：

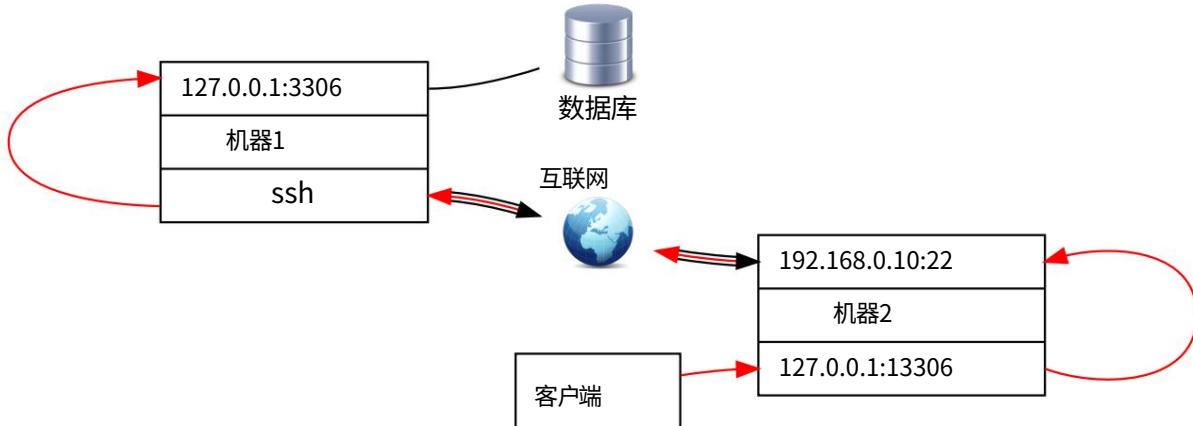
```
我的机器# ssh -L 13306:localhost:3306 john@101.102.103.104
```

然后运行您的应用程序，连接到由 SSH 隧道转发的本地端口：

```
我的机器# mysql -P 13306
```

您现在已连接到远程 MySQL 服务器，而无需通过网络启用 SQL 连接。

另一个示例（示例 2）是连接的一侧想要/需要初始化 SSH 连接以使远程方能够通过隧道连接回来。在这个简化的示例中，使用 -R 在 Machine1 到 machine2 上启动了一个 ssh 隧道。然后 machine2 上的 sql 客户端可以连接到 localhost:13306 并通过 SSH 隧道在端口 3306 上与 mysql 服务器建立连接。



只要 SSH 会话正在运行或只要有隧道会话通过它，隧道就会保持打开状态。这意味着有两种方法可以使用 SSH 隧道：

· 打开带有端口转发的 SSH 会话，并在需要时保持打开状态。如果您希望能够通过隧道重新连接而无需打开新的 SSH 会话，这将很有用。用于连接到远程主机并在相反方向初始化隧道（示例 2）。

· 在后台打开一个带有端口转发的 SSH 连接，并保持它打开足够长的时间，以便能够通过隧道初始化连接。（“`ssh -f -L <your:port:definition> remote_user@remote_host sleep 10 ;<your application>`”）。

## 安全任务 (212.4)

考生应该能够从各种来源接收安全警报，安装、配置和运行入侵检测系统，并应用安全补丁和错误修复。

关键知识领域：

· 用于扫描和测试服务器端口的工具和实用程序

· 将安全警报报告为 Bugtraq、CERT 或其他来源的位置和组织

· 实施入侵检测系统 (IDS) 的工具和实用程序

· 了解 OpenVAS 和 Snort

条款和实用程序：

· 远程登录

· nmap

· fail2ban

· 数控

· 开放增值服务

· Snort IDS

## 数控（网猫）

描述

Netcat (nc) 是一个非常通用的网络工具。 Netcat 是一种计算机网络服务，用于使用 TCP 或 UDP 读取和写入网络连接。 Netcat 被设计成一个可靠的“后端”设备，可以直接使用或由其他程序和脚本轻松驱动。同时，它还是一个功能丰富的网络调试和排查工具。 Netcat 的功能很多；例如，Netcat 可以用作代理或端口转发器。它可以使用任何本地源端口，或使用松散的源路由。它通常被称为 TCP/IP 瑞士军刀。

netcat 的一些主要特性是：

· 进出任何端口的出站或入站连接，TCP 或 UDP

· 完整的 DNS 正向/反向检查，并带有适当的警告

· 能够使用任何本地源端口

· 能够使用任何本地配置的网络源地址

· 内置端口扫描功能，带有随机发生器

- 内置松散源路由功能
- 可以从标准输入读取命令行参数
- 慢速发送模式,每 N 秒一行
- 传输和接收数据的十六进制转储
- 让另一个程序服务建立连接的可选能力
- 可选的远程登录选项响应器

因为 netcat 不对跨链路使用的协议做任何假设,所以它比 telnet 更适合调试连接。

#### 示例网猫。使用 netcat 执行端口扫描

使用 -z 选项,netcat 将对命令行中给出的端口执行端口扫描。默认情况下,netcat 不会产生任何输出。当仅扫描一个端口时,退出状态指示扫描结果,但对于多个端口,如果其中一个端口正在侦听,则退出状态将始终为“0”。出于这个原因,使用“详细”选项将有助于查看实际结果:

```
# nc -vz 本地主机 75-85
nc:连接到本地主机端口 75 (tcp) 失败:连接被拒绝 nc:连接到本地主机端口 76 (tcp) 失败:连接被拒绝 nc:连接到本地主机
端口 77 (tcp) 失败:连接被拒绝 nc:连接到本地主机端口 78 (tcp) 失败:连接被拒绝连接到本地主机 79 端口 [tcp/finger]
成功!

连接到本地主机 80 端口 [tcp/http] 成功! nc:连接到本地主机端口 81 (tcp) 失败:连接被拒绝 nc:
连接到本地主机端口 82 (tcp) 失败:连接被拒绝 nc:连接到本地主机端口 83 (tcp) 失败:连接被拒绝 nc:连接到本地主机端
口 84 (tcp) 失败:连接被拒绝 nc:连接到本地主机端口 85 (tcp) 失败:连接被拒绝
```

netcat 的手册页显示了更多有关如何使用 netcat 的示例。

Netcat 可以很容易地用在脚本中,用于许多你想自动运行的测试。

#### fail2ban 命令

##### 描述

Fail2ban 扫描 /var/log/pwdfail 或 /var/log/apache/error\_log 等日志文件,并禁止导致密码尝试被拒绝次数过多的 IP 地址。它更新防火墙规则以阻止 IP 地址。

Fail2ban 的主要功能是阻止属于可能试图破坏系统安全的主机的 IP 地址。它通过监控日志文件(例如 /var/log/pwdfail、/var/log/auth.log 等)来确定这些,并禁止在设置的时间范围内尝试登录次数过多或执行任何其他不需要的操作的任何主机 IP 由管理员。Fail2ban 通常配置为在一段时间后取消禁止被阻止的主机,以免“锁定”任何真正的连接。几分钟的解禁时间通常足以防止网络连接被恶意尝试淹没,并降低字典攻击成功的可能性。

#### nmap命令

##### 描述

nmap 是一个网络探索工具和安全扫描器。它可用于扫描网络,确定哪些主机已启动以及它们提供的服务。

nmap 支持大量扫描技术如:UDP、TCP connect()、TCP SYN（半开）、ftp proxy（反弹攻击）、Reverse-ident、ICMP（ping sweep）、FIN、ACK sweep、Xmas Tree、SYN 扫描、IP 协议和 Null 扫描。

如果您构建了防火墙，并且希望检查是否没有打开您不想打开的端口，则可以使用 nmap 工具。

#### 使用 nmap 命令

如果一台机器被 rootkit 感染，一些系统实用程序如 top、ps 和 netstat 通常会被攻击者替换。这些命令的修改版本通过不显示所有可用进程和侦听端口来帮助攻击者。通过对我们的主机执行端口扫描，我们可以探索哪些端口是开放的，并将其与已知服务列表进行比较。例如，这里有一个针对我们本地主机的 TCP 端口扫描示例：

```
$ nmap -sT 本地主机

在 2013-07-04 06:33 启动 Nmap 6.25 ( http://nmap.org ) CDT Nmap 扫描本地主机 (127.0.0.1) 的报告

主机启动 (0.0011 秒延迟)。
本地主机的其他地址 (未扫描) :127.0.0.1 未显示:993 已关闭端口

港口      国家服务
22/tcp  开启ssh 25/tcp  开启smtp 53/
tcp  开启domain 80/tcp  开启http 111/
tcp  开启rpcbind 389/tcp  开启ldap 3000/tcp
开启ppp

Nmap 完成:在 0.20 秒内扫描了 1 个 IP 地址 (1 个主机启动)
```

---

#### 笔记

默认情况下，nmap 只会扫描 1000 个最常用的端口。使用 -p 1-65535 或 -p -开关扫描所有可用端口。

---

让我们使用 UDP 协议执行相同的扫描：

```
$ nmap -sU localhost 您请求的扫描类型需
要 root 权限。
退出!
```

Nmap 是一个非常强大的网络扫描器，但有些选项需要 root 权限。如果您将以 root 身份并使用您自己的权限执行命令 nmap localhost，则 nmap 将以 root 身份使用 -sS 选项，并在以普通用户权限运行时使用 -sT。

现在，让我们通过 sudo 使用 root 权限再次运行 UDP 扫描：

```
$ sudo nmap -sU localhost [sudo] bob 的密码:
*****
在 2013-07-04 06:51 启动 Nmap 6.25 ( http://nmap.org ) CDT Nmap 扫描本地主机 (127.0.0.1) 的报告

主机启动 (0.000040 秒延迟)。
本地主机的其他地址 (未扫描) :127.0.0.1 未显示:995 已关闭端口 PORT SERVICE

状态
53/udp  打开 68/udp  打开|          领域
过滤 dhcpc 111/udp  打开 rpcbind
```

```
1900/udp 打开过滤 upnp 5353/udp 打开过滤 zeroconf
```

```
Nmap 完成:在 1.48 秒内扫描了 1 个 IP 地址 (1 个主机启动)
```

Nmap 是一个用途广泛且功能强大的工具，并提供了多种有关其功能的选项。例如，Nmap 可用于主动 TCP/IP 堆栈指纹识别以确定远程操作系统。您需要管理员权限才能执行此操作：

```
[root@lnx1 ~]# nmap -A 192.168.1.183
```

```
在 2015-10-30 16:03 CET 启动 Nmap 5.51 ( http://nmap.org ) 192.168.1.183 主机的 Nmap 扫描报告已启动 (0.0012 秒延迟)。
```

```
未显示:999 个关闭的端口
```

```
港口国服务版本
```

```
22/tcp 打开 ssh | ssh 主机密钥: OpenSSH 6.0p1 Debian 4+deb7u2 (协议 2.0)
```

```
1024 65:44:cf:c2:b8:e9:6a:a5:21:18:9f:55:70:1d:d9:57 (DSA) |_2048 87:15:36:b4:28:09:3c:84:fa:ea:9f:b3:9d:33:39:f9 (RSA)
```

```
MAC 地址:00:0F:60:02:BA:0D (Lifetron Co.)
```

```
没有与主机完全匹配的操作系统 (如果您知道其上运行的是什么操作系统,请参阅 http://nmap.org/ ←)。
```

```
提交 / )。
```

```
TCP/IP 指纹:
```

```
OS:SCAN(V=5.51%D=10/30%OT=22%CT=1%CU=34692%PV=Y%DS=1%DC=D%G=Y%M=000F60 %TM=5 OS:63386C7%P=x86_64-
```

```
redhat-linux-gnu)SEQ(SP=109%GCD=1%ISR=103%TI=Z%CI=I%II=OS:I%TS=7)
```

```
SEQ(SP=107%GCD=1%ISR=104%TI=Z%CI=I%II=I%TS=7)OPS(O1=M5B4ST11NW6%
```

```
OS:2=M5B4ST11NW6%O3=M5B4NNT11NW6%O4=M5B4ST11NW6 %O5=M5B4ST11NW6%O6=M5B4ST11)
```

```
操作系统:OPS(O1=M5B4ST11NW6%O2=NNT11%O3=M5B4NNT11NW6%O4=M5B4ST11NW6%O5=M5B4ST11NW 操作系统:
```

```
6%O6=M5B4ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5 =7120%W6=7120)ECN(R=操作系
```

```
统:Y%DF=Y%T=40%W=7210%O=M5B4NNNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T= 40%S=O%A=S+F=AS%R
```

```
OS:D=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R %O=%RD=0%Q=)T5(R=Y%
```

```
OS:DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40% W=0%S=A%A=Z%F=R%
```

```
OS:O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=4
```

```
OS:0%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

```
网络距离:1 跳
```

```
服务信息:操作系统:Linux
```

```
追踪路线
```

```
跳转时间 地址
```

```
1 1.16 毫秒 192.168.1.183
```

从输出中可以看出，被测试的机器是 Debian linux 主机。

请查阅 nmap(1) 的联机帮助页以了解有关其功能的更多信息。

## 开放增值服务

Open Vulnerability Assessment System (OpenVAS) 是一个包含多种服务和工具的开源框架，可提供全面而强大的漏洞扫描和漏洞管理解决方案。

实际的安全扫描器伴随着每天更新的网络漏洞测试 (NVT) 提要，总共超过 30,000 个（截至 2013 年 4 月）。

有关 OpenVAS 的详细信息，请访问：[Openvas - 开放式漏洞评估系统社区站点](#)。

## Snort IDS（入侵检测系统）

Snort 是一种开源网络入侵检测系统 (NIDS)，能够在 IP 网络上执行实时流量分析和数据包日志记录。它可以执行协议分析、内容搜索/匹配，并可用于检测各种攻击和探测，例如缓冲区溢出、隐形端口扫描、CGI 攻击、SMB 探测、操作系统指纹识别尝试等等。

更多的。 Snort 使用灵活的规则语言来描述它应该收集或传递的流量,以及使用模块化插件架构的检测引擎。 Snort 也具有实时警报功能,将系统日志、用户指定文件、UNIX 套接字或使用 Samba 的 smbclient 的 WinPopup 消息的警报机制整合到 Windows 客户端。 Snort 具有三个主要用途。它可以用于直接的数据包嗅探器,如 tcpdump、数据包记录器(对网络流量调试等很有用),或用作完整的网络入侵检测系统。 Snort 以 tcpdump 二进制格式或 Snort 的解码 ASCII 格式将数据包记录到根据外部主机的 IP 地址命名的日志目录。

### Snort规则的基本结构

所有的 Snort 规则都有两个逻辑部分:规则头和规则选项。

规则标头包含有关规则采取的操作的信息。它还包含用于将规则与数据包进行匹配的标准。选项部分通常包含一条警报消息和有关应该使用数据包的哪一部分来生成警报消息的信息。选项部分包含用于将规则与数据包进行匹配的附加条件。规则可以检测一种类型或多种类型的入侵活动。智能规则应该能够应用于多个入侵特征。

### Snort 规则头的结构

规则的操作部分决定了当满足条件并且规则与数据包完全匹配时采取的操作类型。典型的操作是生成警报或日志消息或调用另一个规则。您将在本章后面了解有关操作的更多信息。

协议部分仅用于将规则应用于特定协议的数据包。这是规则中提到的第一个标准。使用的协议示例有 IP、ICMP、UDP 等。

地址部分定义源地址和目标地址。地址可以是单个主机、多个主机或网络地址。

您还可以使用这些部分从完整网络中排除某些地址。有关地址的更多信息将在稍后讨论。

请注意,规则中有两个地址字段。源地址和目的地址是根据方向字段确定的。例如,如果方向字段为“->”,则左侧的地址是源地址,右侧的地址是目标地址。

在 TCP 或 UDP 协议的情况下,端口部分确定应用规则的数据包的源端口和目标端口。

对于 IP 和 ICMP 等网络层协议,端口号没有任何意义。

规则的方向部分实际上决定了哪个地址和端口号用作源,哪个用作目标。

只是一些例子:

```
alert icmp any any -> any any (msg: Ping with TTL=100 ; ttl: 100;) alert udp any 1024:2048 -> any any (msg: UDP ports ;) alert tcp 192.168.2.0/24 23 <-> 任何任何 (内容: "机密" ;消息: "检测到 <- 机密的" ;)
log udp any 53 -> any any 日志udp
```

有关 Snort 的详细信息,请访问: [Snort IDS](#)。

## 入侵检测和预防系统

在谈论入侵检测系统(IDS)时,我们可以区分主机入侵检测系统(HIDS)和网络入侵检测系统(NIDS)。当主机遭受可疑活动时,HIDS 会发出警报。NIDS 通常会检查网络流量,最好是在低级别,如果检测到可疑流量就会发出警报。

某些 IDS 系统的配置方式不仅可以发出警报,还可以阻止对特定资源的访问。

此资源可以是 TCP/IP 或 UDP 端口、网络设备上的物理端口或通过路由器或防火墙完全访问特定主机或网段。由于这些系统不仅可以检测,还可以防止它们被称为入侵防御系统(IPS)。与 IDS 系统一样,我们可以区分 HIPS 和 NIPS 系统。

入侵检测和入侵防御系统都使用检测定义系统。这些定义描述了某些特征,当满足这些特征时,会触发警报或对策。如果检测发生并且是正确的,我们称之为真阳性。如果发生检测但不准确,则称为误报。当系统未检测到

没有发生的事情,这被称为真阴性。当实际存在系统未检测到的事件时,这称为漏报。

通常,通过使用启发式检测方法可以扩展 IDS 的检测能力。为了使这些既有效又准确,需要对系统进行培训。在此期间,可能会检测到很多误报,这并不是坏事。但是系统需要进行调整,以便将误报的数量减少到最低限度。漏报等同于没有 IDS,这是 IDS 最不可取的行为。

## 跟踪安全警报

### 安全警报

安全警报是关于某些软件漏洞的警告。这些漏洞可能会导致您的服务水平下降,因为某些人非常善于滥用这些漏洞。这可能会导致您的系统被黑客入侵或彻底崩溃。

大多数情况下,问题已经有了解决方案,或者有人已经在着手解决这个问题,这将在本节的其余部分进行描述。

### 错误提示

#### 描述

BugTraq 是[securityfocus.com](http://securityfocus.com)上的一个完全公开的适度邮件列表,用于详细讨论和公布计算机安全漏洞:它们是什么、如何利用它们以及如何修复它们。

### Bugtraq 网站

[SecurityFocus](http://SecurityFocus.com)网站汇集了许多与安全相关的不同资源。其中之一是 Bugtraq 邮件列表。

还有一个 Bugtraq 常见问题解答。

#### 如何订阅 Bugtraq

使用位于<http://www.securityfocus.com/>的网络表单订阅任何 SecurityFocus 邮件列表。

### CERT

#### 描述

CERT 协调中心 (CERT/CC) 是互联网安全专业知识的中心,位于软件工程学院,这是一个由卡内基梅隆大学运营的联邦政府资助的研发中心。他们研究 Internet 安全漏洞、处理计算机安全事件、发布安全警报、研究网络系统的长期变化以及开发信息和培训以帮助您提高站点的安全性。

### 网站

CERT 维护着一个名为[The CERT Coordination Center](http://The CERT Coordination Center)的网站

#### 如何订阅 CERT 咨询邮件列表

请参阅 [us-cert.gov](http://us-cert.gov)列表和提要页面以注册 CERT 咨询邮件列表或在各种 NCAS 出版物上发布的 RSS 提要。

中国国家计算机中心

## 描述

CIAC 是美国能源部的计算机事件咨询能力。 CIAC 成立于 1989 年,在 Internet 蠕虫病毒爆发后不久,CIAC 向 DOE 的员工和承包商免费提供各种计算机安全服务,例如:事件处理咨询、计算机安全信息、现场研讨会、白帽审计。

## 网站

有一个[CIAC 网站](#)。

## 订阅邮件列表

CIAC 有几个电子出版物的自订阅邮件列表:

CIAC-BULLETIN for Advisory,最高优先级 - 时间关键信息,以及 Bulletins,重要的计算机安全信息。

CIAC-NOTES for Notes,计算机安全文章集。

SPI-ANNOUNCE 有关安全配置文件检查器 (SPI) 软件更新、新功能、分发和可用性的官方新闻。

SPI-NOTES,用于讨论有关使用 SPI 产品的问题和解决方案。

邮件列表由一个名为 ListProcessor 的公共域软件包管理,它忽略电子邮件标题行。要订阅 (添加自己) 到其中一个邮寄列表,请发送以下形式的请求:订阅列表名称 LastName、FirstName、PhoneNumber 作为电子邮件正文,替换为 CIAC-BULLETIN、CIAC-NOTES、SPI-ANNOUNCE 或 “list-name”的 SPI-NOTES 以及 “LastName”、“FirstName” 和 “PhoneNumber”的有效信息。发送至:ciac-listproc@llnl.gov。

您将收到一封确认函,其中包含地址和初始 PIN,以及有关如何更改其中任何一个、取消订阅或获得帮助的信息。

## 取消订阅邮件列表

要从 CIAC 邮件列表中删除,请通过电子邮件将以下请求发送到 ciac-listproc@llnl.gov:取消订阅列表名称。

## 使用 telnet 测试开放邮件中继

### 描述

开放邮件中继是一种邮件服务器,它接受来自任何地方的 SMTP 连接并将电子邮件转发到任何域。这意味着每个人都可以连接到该邮件服务器上的端口 25 并将邮件发送给他们想要的任何人。因此,您服务器的 IP 可能最终会出现在反垃圾邮件黑名单中。

### 测试开放邮件中继

可以通过将收件人的电子邮件传送到不应为收件人域进行任何中继的服务器来测试邮件中继。如果服务器接受并发送电子邮件,则它是开放中继。

在以下示例中,我们使用 telnet 连接到运行在端口 25 上的 SMTP 服务器:

```
$ telnet 本地主机 25 正在尝试 ::1...
连接到本地主机。
转义符是 ^] 。 220 linux.mailserver ESMTP Exim
4.80 Wed, 03 Jul 2013 08:08:06 -0500 MAIL FROM: bob@example.com 250 OK

RCPT 至 :root@localhost
250 已接受
数据
354 输入消息,以 “.” 结尾单独一行 Open Mail Relay 测试邮件

.
250 好 id=1UuMnI-0001SM-Pe
QUIT
221 linux.mailserver closing connection 连接被外部主机关闭。
```

邮件被接受是因为邮件服务器被配置为接受来自本地主机的连接，并且根据 SMTP 服务器，root@localhost 是一个有效的电子邮件地址。

Telnet 被认为不太适合作为远程登录协议，因为所有数据都以明文形式通过网络传输。但是 telnet 命令对于检查打开的端口非常有用。目标端口可以作为参数给出，如上例所示。

## OpenVPN (212.5)

考生应该能够配置 VPN（虚拟专用网络）并创建安全的点对点或站点到站点连接。

参考资料：[OpenVPN](#)

关键知识领域：

打开VPN

条款和实用程序：

- /etc/openvpn/
- 开放式VPN

打开VPN

OpenVPN 是一个免费的开源软件应用程序，它实现了虚拟专用网络 (VPN) 技术，用于在路由或桥接配置和远程访问设施中创建安全的点对点或站点到站点连接。它使用 SSL/TLS 安全性进行加密，能够穿越网络地址转换器 (NAT) 和防火墙。

OpenVPN 允许对等点使用预共享密钥、证书或用户名/密码相互验证。在多客户端服务器配置中使用时，它允许服务器使用签名和证书颁发机构为每个客户端发布身份验证证书。它广泛使用 OpenSSL 加密库，以及 SSLv3/TLSv1 协议，并包含许多安全和控制功能。

## 安装中

OpenVPN 几乎可以在任何现代操作系统上使用，并且可以从源代码构建或作为预构建包安装。

OpenVPN 与 IPsec 或任何其他 VPN 包不兼容。整个包包含一个用于客户端和服务器连接的二进制文件、一个可选的配置文件，以及一个或多个密钥文件，具体取决于所使用的身份验证方法。

### openvpn 选项

OpenVPN 允许将任何选项放在命令行或配置文件中。虽然所有命令行选项前面都有一个双前导破折号（“--”），但是当一个选项被放置在一个配置文件中时，这个前缀可以被删除。

--config file 从文件中加载额外的配置选项，其中每一行对应一个命令行选项，但与  
删除前导“--”。

-dev tunX|tapX|null TUN/TAP 虚拟网络设备（动态设备可以省略 X。）。

--nobind bits 不要绑定到本地地址和端口。IP 堆栈将为返回的数据包分配一个动态端口。由于对等方无法提前知道动态端口的值，因此此选项仅适用于将使用 --remote 选项启动连接的对等方。

--ifconfig l rn connection\_spec 设置 TUN/TAP 参数。l 是本地 VPN 端点的 IP 地址。对于 TUN 设备，rn 是远程 VPN 端点的 IP 地址。对于 TAP 设备，rn 是正在创建或连接到的虚拟以太网段的子网掩码。

秘密文件 [方向] 启用静态密钥加密模式（非 TLS）。使用生成的预共享秘密文件  
--genkey

## 配置

### 简单的点对点示例

此示例使用静态密钥进行身份验证。这是一个非常简单的设置，非常适合点对点网络。在以下示例中，将使用 tun 接口。另一种可能性是使用 tap 接口，但配置也会有所不同。有关使用这些接口的更多信息，请参见手册页。

将使用服务器端点 10.10.10.10 和客户端端点 10.10.10.11 创建 VPN 隧道。服务器的公共 IP 地址由 vpnserver.example.com 引用。这些端点之间的通信将被加密并通过默认的 OpenVPN 端口 1194 进行。

要设置此示例，必须创建一个密钥：openvpn --genkey --secret static.key。将此密钥（static.key）复制到客户端和服务器。

#### 服务器配置文件（server.conf）：

```
开发屯
ifconfig 10.10.10.10 10.10.10.11 keepalive 10 60 ping-timer-
rem persist-tun persist-key secret static.key
```

#### 客户端配置文件（client.conf）：

```
远程 vpnserver.example.com dev tun
ifconfig 10.10.10.11 10.10.10.10 keepalive 10 60 ping-timer-
rem persist-tun persist-key secret static.key
```

通过运行 openvpn server.conf 并在客户端上运行 openvpn client.conf 在服务器上启动 vpn。

## 问题和解答

### 系统安全

1. 列出IANA定义的私有网络地址范围

- 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
- 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
- 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

[私有网络地址\[339\]](#)

2. NAT 是什么意思？

网络地址解读。[网络地址转换\[340\]](#)

3. 拥有私有IP地址的服务器如何连接到互联网上的服务器？

通过执行网络地址转换的路由器（或具有路由器功能的服务器）进行连接。[网络地址转换实现\[340\]](#)

4. Linux内核中IPv4包过滤规则表的建立、维护、检查用什么工具？

[iptables iptables \[340\]](#)

5. 将 netfilter 命名为 CHAINS

PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING [iptables 链](#)

6. 将 netfilter 命名为 TABLES。

过滤、Nat、Mangle [iptables 表](#)

7. 对 FTP 流量执行状态防火墙需要什么模块？

[ip\\_conntrack\\_ftp iptables ip\\_conntrack\\_ftp \[342\]](#)

8. FTP 需要什么才能穿过防火墙（“传入”）？

- 必须加载模块 ip\_conntrack\_ftp · 必须接受到端口 21 的传入新连接
- 必须接受 RELATED 和 ESTABLISHED 连接的传入流量
- 必须接受传出流量（最少来自端口 21 和 ESTABLISHED 和 RELATED）。

[通过防火墙的 FTP \[357\]](#)

9. 命名数据包在到达有状态防火墙时可能处于的连接状态。

新的、已建立的、相关的、无效的 [iptables 连接状态\[342\]](#)

10. 响应来自网络上任何服务器的 ping 的最小 iptables 规则是什么？

假设网络连接到 eth0：

```
iptables -t filter -A INPUT -i eth0 -p icmp --icmp-type echo-request -m state -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

[iptables 允许 ICMP \[349\]](#)

11. 主动和被动模式下的FTP有什么区别？

在主动模式下，客户端向服务器发送客户端将监听的 IP 地址和端口号，然后服务器启动 TCP 连接。在被动模式下，客户端向服务器发送 PASV 命令并接收 IP 地址和端口号作为返回。客户端使用这些来打开与服务器的数据连接。[FTP 被动主动\[356\]](#)

12. routed 守护进程实现了什么协议?

RIP 路由协议。[路由实现 RIP \[354\]](#)

13. iptables 默认知道哪 4 个目标?

ACCEPT, DROP, QUEUE, RETURN [iptables 默认目标](#)

14. 至少命名三个扩展的 iptables 目标

LOG, MARK, REJECT, TOS, MIRROR, SNAT, DNAT, MASQUERADE, REDIRECT [iptables 扩展目标](#)

15. iptables 默认包含哪些模块? (姓名 4)

tcp, udp, icmp, mac, limit, multiport, mark, owner, state, unclean, tos [iptables 模块](#)

16. 描述“带有IP地址欺骗的DoS”

系统 A 向系统 B 发送数据包。这些数据包伪造了系统 C 的源地址。因此,系统 B 将向系统 C 发送响应。[带有 IP 地址欺骗的 DoS \[353\]](#)

17. 如何防止DoS攻击?

无法阻止 DOS 攻击,但可以通过对防火墙应用过滤和速率限制规则来降低影响。

[防止 DoS \[353\]](#)

18. 什么是 SSH?

SSH 是 rlogin 和 rsh 的安全替代品。[SSH 使用\[360\]](#)

19. 命名 sshd 配置选项 PermitRootLogin 的可能值

是的,不,没有密码,仅强制命令[SSH PermitRootLogin \[362\]](#)

20. 在 ssh 会话中显示 x 内容的首选方式是什么?

启用 X11 转发以通过 SSH 连接将 X 数据转发到本地显示器。[SSH 启用 X 转发](#)

21. 说出使用密码短语 ssh 密钥的安全影响以及至少一种减少影响的方法。

任何有权访问无密码密钥的人都可以访问,因为使用该密钥不需要密码。为密钥设置一个强制命令,最好限制对一个(或几个)客户端主机的访问。[无密码密钥风险\[363\]](#)

22. 如何确保不必为每个新连接输入密码(使用相同的密钥)?

使用 ssh-agent 加载密钥并启用代理转发。[SSH 代理\[364\]](#)

23. 什么是 SNORT?

Snort 是一种网络入侵检测系统。[第12.4.7节](#)

24. 如何让内部服务器上的私有IP地址的服务可以从互联网上访问?

为防火墙上的传入连接配置端口转发。[iptables 端口转发\[352\]](#)

25. 启用无密码登录需要做什么?

创建一对公钥/私钥,并将公钥的内容添加到远程用户的 authorized\_keys 文件中。[SSH authorized\\_keys \[361\]](#)

26. 不对来自 Internet 的传入连接执行 SOURCE NAT 的最重要原因是什么?

它妨碍了对接收服务器上这些连接的审核,因为所有流量似乎都来自同一个客户端(防火墙)。[iptables Source Nat 考虑\[352\]](#)

27. 运行 pure-ftpd 与运行任何其他 FTP 服务器有何不同?

与许多守护进程不同,Pure-FTPD 不读取任何配置文件(使用时 LDAP 和 SQL 除外)。相反,它使用命令行选项。[纯 FTPD 配置\[358\]](#)

28. 如果我们想显示在 SSH 中生成的 X 输出,是否需要运行 xhost 以允许连接到本地 X 服务器

启用 X11 转发的会话?

不需要。因为 X 输出看似是在本地创建的,所以无需更改 xhost 设置即可显示内容。[没有 xhost 的 SSH X11 转发\[363\]](#)

29. SSH 的端口转发是如何工作的?

SSH 绑定一个本地端口,通过与绑定的本地端口关联的开放 SSH 连接将来自该端口的所有流量隧道传输到该连接另一端的服务器上的端口。  
[SSH端口映射【364】](#)

30、nmap有什么用?

Nmap 可用于扫描网络以确定哪些主机已启动以及它们提供哪些服务。[地图【367】](#)

31.描述openVAS。

OpenVAS 是一个包含多种服务和工具的框架,提供全面而强大的漏洞扫描和漏洞管理[openvas【369】](#)

32. 列举几个安全警报的来源。

- Bugtraq

- 计算机紧急响应小组
- 中国国际仲裁中心

[安全警报](#)

## 第13章

### 参考书目

- [阿尔比茨01] Paul Albitz,Cricket Liu,DNS 和 BIND (第四版) ,O'Reilly,2001 年,ISBN 0-596-00158-4。
- [Apache01] Apache HTTP 服务器指令索引,Apache 软件基金会,2013 年。
- [Aplus901902] Mike Meyers,CompTIA A+ 认证综合考试指南,第 9 版,麦格劳希尔出版社,2016 年,ISBN 13 978-1-25-958951-5。
- [Wikipedia.org 的 BIND] Wikipedia.org 的 BIND,维基百科。
- [绑定常见问题] 互联网系统联盟。
- [班德尔97] David Bandel,Linux Journal,1997 年 1 月。
- [酒吧00] Moshe Bar,Dobbs 博士,2000 年 3 月 29 日。
- [鸟01] 蒂娜·伯德,2001 年 8 月。
- [引导常见问题]
- [Brockmeier01] Joe Brockmeier,Unix 评论。
- [Btrfs]
- [选择正确的 DNS 配置] DNS 服务器比较:如何选择正确的 DNS 配置,Justin Elling 木头。
- [ChrootBind9] Scott Wunsch,Linux 文档项目。
- [Coar00] Ken Coar,INT Media Group, Incorporated,2000 年 2 月。
- [Colligan00] 未知,维基百科,2012 年 11 月 2 日。
- [Wikipedia.org 上的 DANE] Wikipedia.org 上的 DANE,维基百科。
- [DNSHowto] Nicolai Langfeldt,Jamie Norrish,v3.1:2001 年 10 月。
- [道森00] Terry Dawson,Olf Kirch,Linux 文档项目,2000 年 3 月。
- [Dean01] Jeffrey Dean,LPI Linux Certification In A Nutshell,桌面快速参考,O'Reilly,2001 年 6 月,第一版。
- [Wikipedia.org 上的 DigiNotar] Wikipedia.org:DigiNotar,维基百科。
- [唐99] don@sabotage.org,1999 年 4 月。
- [德雷克00] 约书亚德雷克,2000 年 12 月。
- [Engelschall00] Ralph S. Engelschall,2000。

[自由/广域网]

[弗里德01] Jeffrey EF Friedl,精通正则表达式,O'Reilly,1997年,ISBN 0-56592-257-3。

【生成TLSA记录】虎雀网:生成TLSA记录,虎雀书门。

[Hinds01] David Hinds,Sourceforge.net,2001年7月13日。

[如何使用systemctl] 如何使用Systemctl管理Systemd服务和单元,Justin Ellingwood。

[休伯特00] Bert HubertRichard Allen,Linux文档项目,2000年4月28日。

[杰克逊01] 伊恩·杰克逊,2001年7月24日。

[约翰逊01] 理查德·约翰逊,1999年4月。

[Kiracofe01] Daniel Kiracofe,v1.3,2001年1月。

[克劳斯01] 拉尔夫·克劳斯,2001年1月10日。

[LPIC2sybex2nd] Christine BresnahanRichard Blum,LPIC-2 Linux Professional Institute Certification Study Guide (第二版),Sybex,2016年10月,ISBN 978-1-119-15079-4。

[卢特尔] Sander van Vugt,Linux Under the Hood Livelessons 2.,Prentice Hall,2017年1月13日,ISBN-13 978-0-13-466299-

[让我们加密TLSA] Internetsociety.org:让我们为邮件服务器和DANE加密证书,Jan or。

[Lindgreen01] Ted Lindgreen,Snow BV,2001年9月14日。

[LinuxRef01],红帽。

[LinuxRef02],红帽。

[LinuxRef03],Linux星球。

[LinuxRef04],自民党。

[LinuxRef05],XFree86项目公司..

[LinuxRef06],Apache,2001年2月28日。

[LinuxRef07],中央昆士兰大学,1999年。

[LinuxRef08],阿帕奇。

[刘00] Cricket Liu,Acme Byte and Wire LLC,2000。

[卢戈00] 大卫卢戈

[McGough01] Nancy McGough,版权所有©1994 Nancy McGough和Infinite Ink。

[NFS],开源开发者网络。

[NFSv4]

[NFSv4.2]

[NginX01] NginX网站,Nginx公司,2007年。

[NginX02] NGINX+https 101基础知识和入门,Cloudflare,2016年2月4日。

[尼尔森01] Mark Nielsen,Linux Gazette,2001年2月14日。

[尼尔森98] Mark Nielsen,Linux Gazette,Januari 1998.

[Nijssen99] Theo Nijssen,CERT NL,1999年8月。

[Pearson00] 未知,Squid-cache.org,2012年8月29日。

- [PerlRef01] Stas Bekman,Apache Group,2001 年 9 月。
- [PerlRef02] , ActivePerl 文档。
- [PerlRef04] Randal L. SchwartzTom Phoenix,O'Reilly,2001 年 7 月。
- [Poet99] 诗人,Linux 评论,1999 年 8 月 21 日。
- [PostfixGuide] Kyle D. Dent,Postfix:权威指南,O'Reilly,2003 年,ISBN 0-000-00000-0。
- [PostfixTFSreadme] Postfix 文档:Postfix 中的 TLS 前向保密,Postfix.org。
- [PostfixTLSreadme] Postfix 文档:Postfix TLS 支持,Postfix.org。
- [普拉萨德01] Tanmoy PrasadChris Snell,Linux 文档项目,2001 年 7 月 25 日。
- [RFC2487] 基于 TLS 的安全 SMTP 的 SMTP 服务扩展,Paul Hoffman。
- [RFC6698] 基于 DNS 的命名实体身份验证 (DANE) 传输层安全 (TLS) 协议:TLSA,互联网工程任务组。
- [RFC7671] 基于 DNS 的命名实体身份验证 (DANE) 传输层安全 (TLS) 协议:更新和指导,互联网工程任务组。
- [RIPE: BIND 10] RIPE: The Decline and Fall of BIND 10,Shane Kerr。
- [罗宾斯01] Daniel Robbins,IBM DeveloperWorks,2001 年 2 月。
- [Robbins01.2] Daniel Robbins,IBM DeveloperWorks,2001 年 7 月。
- [罗宾斯96] 阿诺德·D·罗宾斯,2001 年。
- [SMTauthHowto] Postfix SMTP AUTH (和 TLS) HOWTO,Patrick Ben Koetter。
- [SSL01] SSL/TLS 强加密常见问题解答,Apache 软件基金会。
- [SSL02] Dan Poirier,SSL with Virtual Hosts Using SNI,Httpd Wiki,2009 年 4 月。
- [SSL03] Moxie Marlinspike,SSL 和真实性的未来,Defcon,2012 年 3 月。
- [SSL04] Moxie Marlinspike,SSL 和真实性的未来,TLDP,2001 年 1 月。
- [Sayle98] 罗伯特·赛尔,1998 年 6 月 6 日。
- [夏普01] Richard Sharpe,使用 Samba,特别版,Que Corporation,2000 年 7 月,首次印刷。
- [直到01] 大卫提尔
- [Truemper00] Winfried Truemper,2000 年 7 月 23 日。
- [DANE 和 DNSSEC 教程] DANE 和 DNSSEC 教程,Wes Hardaker。
- [UEFI] 统一可扩展固件接口论坛,各种。
- [USBRef01] ,linux-usb.org。
- [Vasudevan02] Alavoor Vasudevan,linuxdocs.org,2002 年 1 月。
- [Wall01] Larry Wall,Tom Christiansen,Jon Orwant,Programming Perl (第 3 版),O'Reilly,2000 年,ISBN 0-596-00027-8。
- [韦塞尔01] 杜安韦塞尔斯
- [Will00] Michael Will,Linux 文档项目。
- [威尔逊00] Brian Wilson,O'Reilly OnLamp.com,2000 年 3 月 17 日。
- [Wirzenius98] Lars Wirzenius,Joanna Oja,Stephen Stafford,Linux 文档项目,版本 0.7。

[Yap98] Ken Yap, Linux Focus, 1998 年 9 月。

[Zadok01] Erez Zadok, Linux NFS and Automounter Administration (Craig Hunt Linux Library),, Sybex, 2001., ISBN 0-7821-2739-8.

[apache24upgrade] 从 2.2 升级到 2.4, Apache 软件基金会。

[apache24upgrade] Apache LogLevel 指令文档, Apache 软件基金会。

[apache24upgrade] Apache 2.4 中的新增功能, Rich Bowen。

[阿帕奇文档] Apache HTTP 服务器文档, Apache 软件基金会。

[apachesslhowto] SSL/TLS 强加密方法, Apache 软件基金会。

[archDKMS] Arch 的 DKMS, 各种。

[archNVMe] NVMe, 各种。

[archUEFI] Arch 的 UEFI, 各种。

[密码列表] Cipherli.st: 适用于 Apache、nginx 和 Lighttpd 的强密码, Remy van Elst 和 Juerd。

[debianDKMS] Arch 的 DKMS, 各种。

[digochanges] 如何将 Apache 配置从 2.2 语法迁移到 2.4 语法, Justin Ellingwood。

[剧情] Dracut 主项目页面, 各种。

[剧情] Phil Lembo, 关于 Linux 上 LDAP 身份验证的 SSSD 的更多技术博客, Wordpress.com。

[githubDKMS] 戴尔公司 Github 上的动态内核模块系统。

[linuxversions] 维基百科。

[mozsslconf] , 摩斯拉。

[raymii.org] Raymii.org 关于 SSL 的博客, Remy van Elst。

[tomsUEFI] 与您的 BIOS 说再见: 您好, UEFI!, Patrick Schmid 和 Achim Roos。

[图卡尼] Tukaani.org 上的 XZ Utils, 未知。

[wikiCRIME] 犯罪, 多种多样。

[wikiDKMS] Wikipedia.org 上的 DKMS, 各种。

[维基UEFI] 维基百科上的 UEFI, 各种。

[维基XZ] 维基百科上的 XZ, 各种。

[wikipedia\_apachemodules] 各种, 维基百科。

## 附录 A

# LPIC 2 级目标

这些是 LPIC 2 级目标,版本 4.5.0,自 2017 年 2 月 13 日起生效。

LPIC 2 级考试 201		60
200	容量规划	
200.1	衡量资源使用情况并排除故障	6个
200.2	预测未来的资源需求	2个
201	内核	
201.1	内核组件	2个
201.2	编译内核	3个
201.3	内核运行时管理和故障排除	4个
202	系统启动	
202.1	自定义系统启动	3个
202.2	系统恢复	4个
202.3	备用引导加载程序	2个
203	文件系统和设备	
203.1	操作 Linux 文件系统	4个
203.2	维护 Linux 文件系统	3个
203.3	创建和配置文件系统选项	2个
204	高级存储设备管理	
204.1	配置RAID	3
204.2	调整存储设备访问	2
204.3	逻辑卷管理器	3
205	网络配置	
205.1	基本网络配置	3
205.2	高级网络配置和故障排除	4
205.3	排除网络问题	4
206	系统维护	
206.1	从源代码制作和安装程序	2
206.2	备份操作	3
206.3	通知用户系统相关问题	1

表 A.1:LPIC 200 - 206 级目标及其相对权重

LPIC 2 级考试 202		60
207	域名服务器	
207.1	基本 DNS 服务器配置	3
207.2	创建和维护 DNS 区域	3
207.3	保护 DNS 服务器	2个
208	网页服务	
208.1	实施网络服务器	4个
208.2	HTTPS 的 Apache 配置	3个
208.3	实施代理服务器	2个
208.4	将 Nginx 实现为 Web 服务器	2个
209	文件共享	
209.1	Samba 服务器配置	5个
209.2	NFS 服务器配置	2个
210	网络客户端管理	
210.1	DHCP 配置	2个
210.2	PAM 认证	3个
210.3	LDAP 客户端使用	2
210.4	配置 OpenLDAP 服务器	4
211	电子邮件服务	
211.1	使用电子邮件服务器	4
211.2	管理本地电子邮件传送	2
211.3	管理远程电子邮件传送	2
212	系统安全	
212.1	配置路由器	3
212.2	保护 FTP 服务器	2
212.3	安全外壳 (SSH)	4
212.4	安全任务	3
212.5	打开VPN	2

表 A.2:LPIC 级别 207 - 212 目标及其相对权重

# 第14章

## 指数

-  
 .config, 20 /  
 dev/md0, 104 /  
 dev/nst\*, 151 /  
 dev/st\*, 151 /  
 dev/zero, 81 /  
 etc/auto.master, 93 /  
 etc(exports, 276, 277 /  
 etc/fstab , 80, 112 /etc/  
 init.d/autofs, 93 /etc/  
 init.d/bind, 163 /etc/  
 init.d/pcmcia, 35 /etc/  
 init.d/rc, 52 /etc/  
 inittab, 51 /etc/  
 mdadm.conf , 104 /etc/  
 modules.conf, 35 /etc/  
 rc.boot, 52 /etc/rcN.d,  
 52 /proc, 114 /proc/  
 interrupts, 114 /proc/  
 meminfo, 81 /proc /  
 mounts, 41, 79 /proc/  
 sys/kernel, 42 /proc/  
 sys/net/ipv4/ip\_forward,  
 114 /sbin/sulogin, 65 /usr/src/, 145 /  
 var/lib/ldap, 312 /var /named, 163  
 0.0.0.0, 123 10/8, 339 127.0.0.1, 122  
 172.16/12, 339 192.168/16, 339 8.3 文  
 件名格式, 95  
 啊, 129 反  
 中继, 319 Apache  
 \*, 223 .htaccess,  
 217 ?, 223  
 443, 228  
 access\_log,  
 213  
 AllowOverride,  
 218 apache2, 210  
 Apache2 配置文件,  
 210 apache2ctl>, 221 apachectl>,  
 221 APXS, 212 AuthDBMGroupFile,  
 2,17 AuthFile 219 AuthType, 217  
 AuthUserFile, 217 CLF, 213  
 CustomLog, 225 自主访问控制, 214  
 DNS, 223 DocumentRoot, 223  
 htpasswd, 218 httpd, 212 基于IP的  
 虚拟主机, 224 libssl.so, 212 Limit,  
 218 Listen, 223 Mandatory Access  
 控制, 214 MaxClients, 222  
 MaxKeepAliveRequests, 222  
 MaxSpareServers, 222  
 MinSpareServers, 222 mod\_access,  
 215 mod\_auth, 214  
 mod\_auth\_anon, 215  
 mod\_auth\_digest, 215 mod\_ssl,  
 228 个模块, 211 多个守护进程, 224  
 Name-base2 虚拟主机Name2Host ,  
 224

号码 67,  
 289 68,  
 289

A  
 访问日志, 213  
 access.db, 318  
 ACK扫描, 368

PerlSetVar, 220  
 重定向, 225  
 需要有效用户, 217  
 服务器管理员, 225  
 服务器别名, 223  
 服务器名称, 223  
 服务器根目录, 224  
 SSL证书文件, 233  
 SSL证书密钥文件, 233  
 启动服务器, 222  
 传输日志, 225  
 用户, 224  
 虚拟主机, 223  
 APXS, 212  
 ARP  
     缓存, 135 arp,  
     125, 132, 135  
 arpwatch, 135  
 攻击  
     拒绝服务, 353  
     SYN, 353  
 automount, 93, 282 可用性, 12  
  
 乙  
 备份  
     阿曼达, 151岁  
     备份PC, 152  
     巴库拉, 151  
     Bareos, 152 计划, 149 测试, 149 验证, 149  
  
 badblocks, 84 带宽使用, 1 绑定, 163 //, 165 ;, 165 @, 167, 175 #, 165 {, 165 }, 165 允许查询, 202 允许传输, 202 类别, 167 chrooted, 203 当前来源, 176 db.127, 175 db.local, 175 拨号, 166 目录, 165 exworks, 200 fetch-glue, 204 文件, 165 forward, 166 forward first;, 166 forward only;, 166

转发器, 165, 203 心跳间隔, 200 提示, 176 监狱, 204 本地主机, 175 named.conf, 164 named.pid, 204 选项, 165 递归, 204 重新加载, 169 resolv.conf, 202 SIGHUP, 169 从属, 203 stand-alone master, 200 start, 169 stop, 169 version, 166 zone file, 167 blacklisting, 372 blank, 97 blkid, 82 boot, 51 boot option initrd=, 41 boot sequence, 41 bootwait, 52 bottlenecks, 1 bounce attack, 368 广播, 294 广播地址, 123 bugtraq, 371 总线, 96 总线

SCSI、96 bzImage、15, 16 bzip2, 145

C  
 CA, 326  
 CA.pl, 231 仅缓存名称服务器, 164 卡内基梅隆大学, 371 CD-ROM 文件系统, 95 cdrecord, 96 CERT, 371 http://www.cert.org, 371 证书自签名, 326 证书颁发机构, 228, 230, 327 证书签名请求, 230 chkconfig, 58 CIAC, 372 BULLETIN, 372 ciac-listproc@llnl.gov, 372 NOTES, 372 SPI-ANNOUNCE, 372 SPI-NOTES, 372

订阅, 372 取消订阅, 372  
通用日志格式, 213  
通用名称, 230  
配置\_KMOD, 37  
配置模块, 37 配置, 145

配置  
阿帕奇, 221  
Apache 身份验证模块, 217  
阿帕奇 mod\_perl, 219  
Apache mod\_php, 220  
bind, 162 disks, 112 kernel modules, 35  
LDAP 身份验证, 302  
Linux 内核, 20  
Linux 内核选项, 114  
逻辑卷管理器, 116  
网络接口, 122  
网络文件系统, 273  
NIS 身份验证, 302  
开放天鹅, 129  
聚丙烯酰胺, 300  
突袭, 101  
SMB 服务器, 251, 273  
网络服务器, 209  
cpio, 151  
CPU 使用率, 1 创  
建文件系统, 78  
创建文件系  
统, 78  
SSL 服务器证书, 230  
板球, 212  
密码学  
公钥, 227  
企业社会责任, 326  
CTRL-ALT-DEL, 52  
ctrlaltdel, 52 自定义内核, 37 柱面, 112

D  
dd, 97, 151  
debugfs, 83 默  
认网关, 123 默认路由, 123  
depmod, 36 设备或资源  
繁忙, 33 DHCP, 289  
BOOTP, 297 客户端, 289 客户端标  
识符, 296 default-lease-time, 298 dhcpcd.conf, 289  
dhcpcd.leases, 298 域名服务  
器, 293

以太网地址, 296  
全局参数, 289 组声明, 290  
主机声明, 290  
IP 地址, 290 最大  
租用时间, 298 nntp 服  
务器, 293  
正常参数, 289 选项, 293 弹出  
服务器, 293 中继, 298 重新  
加载, 298  
服务器, 289  
共享网络, 289 smtp 服  
务器, 293  
静态主机, 296 子网  
声明, 290 dhcpcd, 289  
dhcrelay, 298  
诊断资源使用情况, 13 dig, 170, 184  
目录块, 78  
磁盘, 150  
磁盘 I/O, 1  
DKMS  
dkms 命令, 31  
dmesg, 112  
内核运行时管理, 44  
排除网络问题, 142  
脱氧核糖核酸酶, 344  
DNS, 163  
dnssec-keygen, 197  
dnssec-signzone, 197  
国家经济委员会, 198  
RRSIG, 197  
DoS 攻击, 353  
拒绝服务攻击  
IP 地址欺骗, 353  
网络入口过滤, 353  
数据包泛洪, 353  
同步, 353 系  
统控制, 353  
带有 IP 地址欺骗的 DoS, 353  
鸽舍, 328  
dracut, 31  
dump2fs, 83, 85  
动态共享对象, 211

乙  
电子邮件, 316  
电子邮件客户端, 316  
ESP, 129 以  
太网接口, 123  
进出口银行,  
320 exportfs, 276,  
279 ext2, 78

F

fail2ban, 367  
FAT, 79  
fdisk, 116  
文件系统, 77  
火墙和路由吞吐量、1转发、352空闲、81  
fsck, 64

, 83, 84

-A, 84

-C, 84

-R, 84

-a, 84

-c, 84

-f, 84

-n, 84

-p, 84

-y, 84

FTP, 239

Active, 356

接模式, 356 firewalll, 357

被动, 356

完全合格的域名, 230

G

网关, 123  
getty, 51  
gnutls-cli, 336  
gopher, 239  
蛴螬, 60 63,  
蛴螬 2, 60  
GRUB Legacy, 60  
grub-install, 63  
gunzip, 145 gzip,  
26 145 ,

HHD参数, 113

HFS, 95

主机, 170, 186

主机名, 142

htop, 9 http 代

理, 239 https,

228

我

ICMP, 133

身份证, 96

IDE, 112

ifcon 图, 122 , 132, 141

IKE, 129

include, 313间

接块, 78 init, 50 64 order

of scripts, 553 init scripts,

initdefault, 52初  
始 RAM 磁盘, 26 initrd,  
26 , 63  
手动创建, 26  
mkinitrd, 48 inittab,  
64 inode, 78 insmod, 32  
install, 146检查间隔, 85  
iostat, 2 iotop, 3

知识产权, 122

第 1 类, 339

类别 2, 339

类别 3, 339私人,

339公共, 339 ip,

123 138

ip\_conntrack, 342

ip\_conntrack\_ftp, 342

IPSEC, 128

IP表, 340

过滤器, 341

粉碎, 341

NAT, 341

有状态, 341

iptables --state,

341

接受, 343

脱氧核糖核酸酶, 344

DROP, 343扩

展模块, 344转发, 352 icmp,

344 ip\_conntrack, 342

ip\_conntrack\_ftp, 342限

制, 344

日志, 344

mac, 344

马克, 344马克,

345

MASQUERADE, 344匹配模

块, 344

MIRROR, 344多端

口, 344

NF\_接受, 341

NF\_DROP, 341

NF\_队列, 341

NF\_重复, 341

NF\_STOLEN, 341所

有者, 345

排队, 343

重定向, 344

拒绝, 344恢

复, 351

返回, 344

保存, 351  
 SNAT, 344  
 状态, 345 目标, 343  
 tcp, 344  
 TOS, 344  
 tos, 345  
 udp, 344  
 unclean, 345  
 iptables-restore, 351  
 iptables-save, 351  
 iptraf, 6 iscsiadm, 107  
 ISO9660, 95  
 iso9660, 79 iw, 125 iwconfig, 125 iwlist, 127 , 126

逻辑卷管理器, 116 最大内核大小, 16  
 NFS 客户端, 274  
 NFS 客户端 v3, 274  
 NFS 服务器, 274  
 NFS Server v3, 274 系统恢复, 60  
 Linux 防火墙, 340  
 linuxrc, 41 lo, 122

## K

kbdrequest, 52 内核压缩, 16 混合, 16  
 微型, 16 单片, 16 路由表, 5 内核文档, 15 内核映像, 15 内核模块, 17 别名, 35 dep 文件, 35 安装, 36 保留, 35 位置, 17 options, 35 path=, 35 post-install, 36 post-remove, 36 pre-install,, 36 pre-remove, 36 remove, 36 kerneld, 36 kill, 169 kmmod, 36

Logical Volume, 116 loop mount, 96 loopback interface, 122 lsdev, 43 lsmod, 32 lsof, 8 lspci, 42 lsusb, 42 , 136 LUN, 96 lvchange, 118 lvcreate, 117 118, lvdisplay, 118 lvextend, 117 lvmp, 118 116 lvmdiskscan, 118 lvmreduce, 118 lvremove, 118 lvrename, 118 lvmresize, 118 lvol, 117 lvremove, 118, 18 lvls 1,

## 大号

LDAP, 309  
 RFC 2116, 305  
 RFC 2251, 305  
 ldapadd, 307  
 ldapdelete, 307  
 ldappasswd, 307  
 ldapsearch, 306  
 LDIF, 312  
 Linux 启动过程, 49

M  
 m4, 317 邮件, 320  
 件服务器、 316 邮件  
 传输代理, 320  
 Maildir, 332  
 maillog, 324  
 mailq, 325 主要版本, 16 make, 19  
 145 binrpm-pkg, 37  
 pkg, 37 mrproper,  
 19 rpm-pkg, 37

make bzImage, 25  
 make clean, 25 make  
 con fig, 21 make  
 gcon fig, 22 make  
 menucon fig, 21 make  
 modules, 25 make  
 modules\_install, 25 make  
 oldcon fig, 25 make xcon fig,  
 22 make zImage, 25  
 makemap, 318 making a文件  
 系统, 78 个伪装连接, 5 个主  
 文件格式, 184

Mbox, 332  
 mbox\_min\_index\_size, 336  
 医学博士,  
 103医学博士, 104  
 内存物

理、8虚拟、  
 8内存使用、  
 1次要版本、16镜  
 像、102 mkcert.sh、  
 334 mke2fs、99  
 mkfs、78 mkfs.ext2、  
 78 mkintrams、  
 26 mkisofs、84, 5117  
 mkswap、81  
 modinfo、34  
 modprobe、33监视  
 器资源使用, 12监  
 控 IO 负载, 2挂载,  
 65 117挂载计数, 85  
 挂载, 276 MRTG, 212 mt,  
 151 MTA, 320 MTU, 123多用  
 户运行级别, 50 MX 记  
 录, 320 , 78 , 281

netbios  
 名称服务, 251网络过  
 滤器, 340挂钩, 340网络掩  
 码, 123 netstat, 4网  
 络, 150网络, 2网络地址转换,  
 340网络 I/O, 117网络入口过  
 滤, 353网络扫描, 368新别  
 名, 319 325 NFS, 122 --all,  
 280 --directories, 280 -r, 279 -ua, 280  
 1024, 281 4096, 281 8192, 281  
 all\_squash, 278 bg, 282客户端, 273 fg,  
 282文件句柄, 286防火墙, 285硬盘, 282  
 intr, 282内核, 274内核空间/271内核线程,LKSIZE,  
 281 nfsvers=2, 282 no\_all\_squash, 278  
 no\_root\_squash, 278 noatime,  
 282 noauto, 282 noexec, 282  
 nointr, 282 nosuid, 282  
 portmapper, security = 274  
 portmapper retry , 274  
 portmapper , 282 ro, 278 281  
 root\_squash, 278 rpc.lockd, 276  
 rpc.mountd, 276 rpc.nfsd, 276  
 rpc.statd, 276 rsize, 281 rw, 278  
 securing , 285 server, 273 SIGHUP,  
 279 soft, 282 squashing, 278

N  
 named, 163  
 named-checkconf, 163  
 named-checkzone, 184  
 named-compilezone, 184  
 named.conf, 163 NAT, 340  
 nc, 137 316 ncat, 316 ncd,  
 163 net, 254 , 366  
 , 366

tcp, 282  
 timeo=, 282  
 udp, 282 user  
 space, 276  
 version 4, 286  
 without portmapper, 274  
 wsize, 281 nfsstat, 284  
 nginx, 245 NIC address, 290 NIS, 122 nmap, 135 ACK sweep, 368  
 bounce attack, 368 network扫描, 368 NULL 扫描, 368 选项, 369 ping 扫描, 367 368 反向识别, 368 SYN 扫描, 368 TCP SYN, 368 测试防火墙, 368 圣诞树, 368 nmblookup, 252  
 nslookup, 184  
 nsswitch.conf, 163 NULL 扫描, 368  
 O  
 奇数发布, 16 off, 52  
 olcLogLevel, 312 once, 51  
 ondemand, 52 open files, 8  
 open relay, 372 how to test, 372  
 required, 300  
 requisite, 300  
 session, 302  
 ssh, 300  
 sufficient, 300  
 try\_first\_pass, 301  
 use\_first\_pass, 301  
 panic, 42 partition, 78  
 patch, 28 29 -quiet, 29 --remove-empty-files, 29 --silent, 29 --strip, 29 -E, 29 -R, 29 -p, 29 -s, 29 补丁级别, 16 PEM, 230 pflogsumm, 324 PHP, 220 物理范围, 116 物理卷, 116 ping, 132 ping 扫描, 368 ping6, 133 PKC, 227 端口映射, 364 portmapper, 283 postconf, 321 Post fix, 320 332 main.cf, 321 TLS, 325 post fix, 322 post fix-tls, 327 postmap, 324 postqueue, 325 powerfail, 52 powerfailnow, 52 powerokwait, 52 powerwait, 52 个专用网络地址, 339 个进程阻塞在 I/O, 11 procmail, 328 procmailrc, 333 ps, 6 pstree, 7 pure-ftpd, 358 pvchange, 118 pvck, 118 pvcreate, 117 pvdisk, 118 pvmove, 118 pvremove, 118 pvs, 118 119

打开 LDAP, 305  
 openssl, 230, 231, 325  
 开放天鹅, 129  
 OpenVAS, 369 开放式 VPN, 374  
 光学媒体, 150

P

Packet Flooding, 353  
 PAM account, 301 auth, 301 login, 300  
 nullok, 301  
 optional, 300  
 pam.conf, 300  
 pam\_ldap.so, 302  
 pam\_nis.so, 302  
 passwd, 300  
 password, 301

pvscan, 118

R

无线电, 299

RAID, 101 0,  
    102 1,  
    102 4,  
    102 5,  
    103 硬  
    件, 103  
    线性, 103 软  
    件, 103

raidstart, 105

重启, 50  
删除补  
丁, 29  
保留块, 85

资源使用

    测量, 1  
    故障排除, 1 次重  
生, 51 次反向区域, 174  
次反向识别, 368 次

RFC1631, 340

RFC2827, 353

rmmmod, 33  
rndc, 168

Rock Ridge, 95

流氓主机, 135 路  
由, 122 133 路由, , 141  
123 路由表, 122

RPC, 122 , 274

rpc 信息, 283

RRD 工具, 212

RSA, 230

RSA 密钥, 360

rsync, 150

runlevel, 50

runlevel 1, 50

runlevel 2-5, 50

runlevel 6, 50

runlevel S, 50

runlevel s, 50

小写

Samba, 251

samba

    global, 257  
    homes, 257  
    inetd, 251  
    ldapsam, 256  
    logon scripts, 269  
    nmbd, 251  
    passwd  
    backend, 256 port 137,  
    251 port 139, 251  
    printers, 257

身份验证, 242  
 cache\_dir, 240  
 cache\_mem, 245  
 cache\_swap, 245拒  
 绝访问, 241  
 http\_access, 240  
 http\_access 允许, 243  
 http\_access 拒绝, 243  
 http\_port, 240  
 maximum\_object\_size, 245  
 minimum\_object\_size, 245重  
 定向程序, 241重定向器, 241  
 squid.conf, 243 SSL, 239商店  
 入口, 244 137

ss, 5 ,  
 SSH, 128  
 ssh, 360 , 361  
 AllowGroups, 362  
 AllowUsers, 362  
 authorized\_keys, 361  
 Blow fish, 360配置sshd,  
 361 DenyGroups, 362  
 DenyUsers, 362  
 ForwardAgent, 364 Host  
 Keys, 360 id\_rsa, 361  
 id\_rsa.pub, 361 keys,  
 360  
 PasswordAuthentication,  
 362 passwordless, 363  
 PermitRootLogin, 362 Port  
 mapping, 364 Protocol, 362  
 protocol version 1, 360  
 PubkeyAuthentication, 363 RSA,  
 360 ssh-add, 364 ssh-agent, 364  
 SSH\_AGENT\_PID, 364 sshd\_config,  
 361 The X Window System, 363隧  
 道、 364用户密钥、 361 X 会话、  
 364 X11DisplayOffset、 363  
 X11Forwarding、 363  
 XAuthLocation、 363 ssh-  
 keygen、 361 sshd、 360  
 sshd\_config、 361 SSL/TLS、 227  
 状态、 341 345状态防火墙、 341统  
 计

传输速率, 11  
 strace, 39条带化, 102  
 超级块, 78超级块位  
 置, 86 swap, 480  
 swapoff, 82 swapon, 81  
 ,  
 SYN 攻击, 353  
 SYN 扫描, 368同  
 步, 83 sysctl, 44  
 sysinit, 52 115 , 353  
 systemctl, 54  
 systemd-delta,, 153

吨  
 胶带, 149  
 焦油150焦  
 油145  
 TCP SYN, 368  
 tcp 包装器, 275  
 tcpdump, 135  
 telinit, 64 telnet,  
 122 testparm, 3165 367

TIME\_EXCEEDED, 134  
 TLS, 227  
 tlsmgr, 327  
 tlsproxy, 327  
 top, 8 12  
 traceroute, 134透  
 明代理, 239  
 三重 DES, 230故障  
 排除 /proc, 37阻  
 塞流量, 141  
 涉及的组件, 139防火  
 墙, 141第一步, 139

跳, 140  
 ICMP, 140  
 ltrace, 40网  
 络, 139物理问题,  
 141 ping, 140路由, 141  
 strace, 39字符串, 40  
 traceroute, 140

TTL, 134  
 tune2fs, , 85 , 113  
 83 tunefs  
 -C, 85  
 -c, 85  
 -i, 85  
 -m, 85

-r, 85  
 隧道, 128  
 道, 364

U  
 udevadm, 45  
 udevmonitor, 45  
 umount, 79  
 uname, 41 , 146  
     --all, 41  
     --hardware-platform, 42 --kernel-name, 41 --kernel-release, 42 --kernel-version, 42 --machine, 42 --nodename, 41 --operating-system, 42 --processor, 42  
     -a, 41 -i, 42 -m, 42 -n, 41 -o, 42 -p, 42 -r, 42 -s, 41 -v, 42  
 unmount, 78 unresolved symbol, 33 update- rc.d, 53  
 UPS, 52正常运行时间, 10

漏洞, 371

W  
 w, 7  
 等待, 51  
 墙, 153网  
 络缓存, 239  
 WINS, 269胜  
 服务器, 251

X  
 X.500,305  
 XFS, 77  
 xfs\_check, 90  
 xfs\_grow, 117  
 xfs\_info, 90  
 xfs\_repair, 90  
 圣诞树, 368 xz、145 xz 压缩, 15

Z  
 zimage, 15 , 16  
 区, 174

V  
 版本编号, 16 vgcfgbackup, 118 vgchange, 119 vgck, 119 vgconvert, 119 vgcreate, 117 119 vgdisplay, 119 vgexport, 119 vgextend, 117 vgimport, 119 vgmerge, 119 vgmknodes, 119 vgreduce, 119 vgres19ve, 119 vgres19ve, 119 vgexport, 119 vgextend 119 vgscan, 119 vgsplit, 119虚拟主机, 222虚拟内存统计, 3 12

vmstat, 2,  
 VPN, 128