红帽企业版 **Linux 7**
安全性指南

红帽企业版 Linux 7安全性指南

Martin Prpič
Yoana Ruseva
翻译、校对：陈坤
翻译、校对：张可

Tomáš Čapek
Miroslav Svoboda
翻译、校对：邓明晗
校对、编辑：任浩

Stephen Wadeley
Robert Krátký
翻译、校对：吴洁蕾
校对、责任编辑：鄭中

红帽企业版 Linux 7安全性指南

Martin Prpič
Red Hat 工程部出版中心
mprpic@redhat.com

Tomáš Čapek
Red Hat 工程部出版中心
tcapek@redhat.com

Stephen Wadeley
Red Hat 工程部出版中心
swadeley@redhat.com

Yoana Ruseva
Red Hat 工程部出版中心
yruseva@redhat.com

Miroslav Svoboda
Red Hat 工程部出版中心
rkratky@redhat.com

Robert Krátký
Red Hat 工程部出版中心
msvoboda@redhat.com

翻译、校对：陈坤
澳大利亚昆士兰大学 笔译暨口译研究所
cuteckhaha@sina.com

翻译、校对：邓明晗
澳大利亚昆士兰大学 笔译暨口译研究所
342024612@qq.com

翻译、校对：吴洁蕾
澳大利亚昆士兰大学 笔译暨口译研究所
cielxphantom@163.com

翻译、校对：张可
澳大利亚昆士兰大学 笔译暨口译研究所
m.zk.dreamer@gmail.com

校对、编辑：任浩
澳大利亚昆士兰大学 笔译暨口译研究所
renhao0823@gmail.com

校对、责任编辑：鄭中
红帽工程部翻译中心 & 澳大利亚昆士兰大学笔译暨口译研究所
ccheng@redhat.com, uqcchun1@uq.edu.au

法律通告

摘要

本书可帮助用户和管理员了解保证工作站和服务器不受本地和远程入侵、攻击代码和恶意活动影响的过程和实践。 侧重于红帽企业 Linux，但细节的概念和技术适用于所有Linux系统，该指南详细介绍了一些规划和工具，这些规划和工具可以为数据中心、工作场所以及家庭创建一个安全的计算环境。使用正确的管理知识、警告和工具，运行 Linux 的系统可在充分发挥功能以及保障自身安全的情况下对抗最常见入侵行为和攻击方法。

目录

第 1 章 安全话题概述

由于对使用强大的联网计算机进行业务运作以及对个人信息管理的依赖性不断增加，各个行业都要了解网络和计算机安全实践。企业要求具有专业知识和技能的安全专家正确审核系统并量身定制解决方案以适应其机构的操作要求。因为大多数机构都在不断壮大，其员工要在本地或者远程访问关键公司 IT 资源，因此安全的计算环境变得更加重要。

遗憾的是，很多机构（以及个人用户）对于安全问题都是马后炮，对于安全的考虑总是放在功能、生产力、便利性、易于使用以及预算之后才考虑。正确的安全部署通常都是事后考虑 — 即在未授权入侵发生*"之后"*才考虑。在还没连接到不可信的网络例如互联网之前，采取正确的方法可有效阻止入侵尝试。

注意

这个文档会经常参考在 **/lib** 目录中的文件。当使用 64 位系统时，有些提到的文件可能位于 **/lib64** 目录中。

1.1. 什么是计算机安全？

计算机安全是一个笼统术语，其意义涵盖了从计算到信息处理的很大领域。依赖计算机系统和网络进行日常业务处理以及访问重要信息的行业视其数据为总资产的重要组成。有些名词和度量已经成为我们日常业务词汇，比如总拥有成本（TCO）、投资回报（ROI）和服务质量（QoS）。使用这些度量，各行业可将某些方面，比如数据整合和高可用性（HA）作为其计划和过程管理成本的一部分。在有些行业中，比如电子商务，数据的可用性和可信性意味着成功与失败。

1.1.1. 标准化的安全性

每个行业中的企业都要依赖由标准制定团体（比如美国医疗协会，AMA；电气与电子工程师协会，IEEE）设定的法规和规则。这同样也适用于信息安全。很多安全顾问和卖方都认同标准安全模式，即 CIA，或称*"保密、完整及可用"*。这个三层的模式是一般被人们接受的评估敏感信息并建立安全策略的元素。下面我们将进一步详细描述 CIA 模式：

- 保密 — 敏感信息必须只能对预先定义的一组个体可用。应限制未授权传输以及使用信息。例如：信息保密可确保客户个人或者财务信息不会被未授权个人以恶意目的获取，比如盗窃或者伪造信用。
- 完整 — 不应以任何方式修改信息以致其不完整或者不正确。应限制未授权用户修改或者销毁敏感信息的能力。
- 可用 — 授权用户应该可以在需要时随时访问信息。可用性可保证以共同商定的频率和时效获取信息。这经常要根据百分比进行计算，并在网络服务供应商及其企业客户使用的服务等级协议（SLA）中有具体说明。

1.2. 安全控制

计算机安全通常可分为三个主要类型，通常指的是 *controls*：

- 物理控制
- 技术控制
- 管理控制

这三个主要类型定义了正确安全部署的主要任务。在这些控制中有一些子分类可进一步细化这些控制以及如何部署它们。

1.2.1. 物理控制

物理控制是在定义的结构内实施保证安全的方法，用来阻止或者防止对敏感资料的未授权访问。物理控制示例包括：

- 闭路监控摄像机
- 动作或者热量报警系统
- 警卫
- 照片 ID
- 上锁并有固定锁的钢制门
- 生物识别（包括指纹、声音、面部、笔迹及其它用来识别个体的自动方法）

1.2.2. 技术控制

技术控制使用技术作为基础来控制对整个物理构架和网络中名敏感数据的访问和使用。技术控制影响范围很大，包括以下技术方面：

- 加密
- 智能卡
- 网络认证
- 访问控制（ACL）
- 文件完整审核软件

1.2.3. 管理控制

管理控制定义安全性中人的因素。它们涉及机构中所有级别的个人，并决定哪些用户可以访问哪些资源和信息：

- 培训和认知
- 灾难准备及恢复计划
- 人员招聘和分离策略
- 人员注册及使用

1.3. 漏洞评估

如果有充分的时间、资源和热情，攻击者几乎可以攻陷任何系统。所有安全过程和技术目前都不能保证系统绝对不会受到入侵。路由器可帮助保护到互联网的网关安全。防火墙可帮助保护网络终端安全。虚拟私用网络可安全地以保密流方式传送数据。入侵探测系统可警告您那些恶意动作。但是这些技术能否成功依赖各种不同因素，其中包括：

- 负责配置、监控和维护技术的专业人士。
- 迅速有效地补丁和更新服务及内核的能力。

- 可让您对网络一直保持警惕的反应能力。

由于数据系统和技术的不断变化，保证企业资源安全是很复杂的。鉴于这个复杂性，通常很难找到可用于您所有系统的专业资源。虽然人们可以在很多领域达到很高的水准，但是很难找到在多个领域都非常精通的人。主要是因为信息安全的每个领域都要求您的持续关注，信息安全不是一成不变的。

漏洞评估是您网络和系统安全的内部审计。其结果表明您网络信息的保密性、完整性和可用性（如 第 1.1.1 节 “标准化的安全性” > 所作出的解释）。通常，在侦查阶段启用漏洞评估。在此阶段，对目标系统和资源这些重要数据进行收集。此阶段会导致进入系统准备阶段，借此对所有已知漏洞进行必要的目标检测。准备阶段会以报告阶段终止，在报告阶段中会对所有发现的风险进行等级分类，分为高、中和低三级；关于目标的安全增强方式（或降低漏洞风险的方式）也会进行讨论。

如果您在家里进行弱点评估，您一般会检查家里的每扇门看看是否关上并锁好。您还会查看每扇窗户，确定关上并锁好。这个理念也同样适用于系统、网络和电子数据。恶意用户就是您数据的小偷和劫匪。清楚他们的工具、心态和动机，您就可以对其动作迅速作出反应。

1.3.1. 定义评估和测试

弱点评估可分为两类：“*由外而内*”和“*由内而外*”。

当进行“由外而内”的弱点评估时，您要从外部对抗您的系统。身处公司之外为您提供了黑客的视角。您看到的就是黑客看到的 — 公开路由的 IP 地址、您 *DMZ* 中的系统、您防火墙的对外接口等等。DMZ 代表“停火区”，对应那些处于可信内部网络，比如企业专用 LAN和不可信的外部网络，比如公共互联网之间的计算机或者小的子网。通常 DMZ 包括可连接到内部网络流量的设备，比如 Web（HTTP）服务器、FTP 服务器、SMTP（电子邮件）服务器和 DNS 服务器。

当进行“由内而外”的弱点评估时，您处于有利地位，因为您在内部且被认为是可信的。这就是您和您的同事登录到系统后的视角。您会看到打印服务器、文件服务器、数据库和其它资源。

这两种弱点评估有很大不同。作为公司的内部用户可拥有比外部用户更多的特权。在大多数机构中是将安全配置为防止入侵者进入。很少是用来防备机构的内部用户（比如部门防火墙、用户等级访问控制以及对内部资源的授权过程。）一般来说，作为大多数系统从内部看来有很多资源是公司的内部资源。一旦您身处公司之外，您的状态就成为不可信。您在公司外部可使用的系统和资源通常非常有限。

请考虑弱点评估和“*入侵测试*”之间的不同。将弱点评估作为入侵测试的第一步。弱点评估收集的信息可用于测试。进行该评估是来检查漏洞和潜在弱点，而入侵测试则是要利用所发现的漏洞和弱点。

评估网络设备是一个动态过程。无论是信息安全还是物理安全，都是动态的。执行评估所显示的概况，可能会出现误报和漏报。误报是指工具所发现的漏洞实际上并不存在。漏报是指遗漏了实际漏洞。

安全管理员只能通过其所使用的工具和所拥有的知识来发挥其作用。采用任何当前可行的评估工具，在您的系统运行这些评估工具，但几乎可确定的是其中存有误报。无论是程序错误还是用户错误，其结果是相同的。这些工具可能会发现误报，或更有甚者，发现漏报。

现在明确了弱点评估和入侵测试之间的不同，请在执行入侵测试前仔细考虑评估结果以便获得新的最佳实践方法。

警告

不要尝试利用生产系统的漏洞。这样做可能会对您系统和网络的生产和效率起到截然相反的效果。

以下列表给出了一些执行弱点评估的优点。

- 主动关注信息安全。
- 在黑客发现潜在漏洞之前找到这些潜在漏洞。

- 保持系统的更新和修补。
- 提升员工的专业知识并给与帮助。
- 减少经济损失和负面宣传。

1.3.2. 漏洞评估方法

在选择弱点评估工具方面，建立弱点评估方法论是很必要的。遗憾的是目前还没有预先确定或者业内公认的方法论，但尝试和最佳实践可作为有效的指导方法。

"*目标是什么？我们要查看的是某台服务器还是整个网络以及网络中的所有东西？我们是在公司外部还是内部？*"这些问题的答案在帮助您决定工具选择乃至使用方法时至关重要。

要更多地了解所发布的方法，请参阅以下网站：

- http://www.owasp.org/ — *开放式 Web 应用程序安全项目（OWASP，The Open Web Application Security Project）*

1.3.3. 漏洞评估工具

通过使用某些形式的信息收集工具，可启动评估。在评估整个网络时，首先绘制出布局，查找正在运行的主机。一旦找到主机的位置，将分别检查每个主机。关注这些主机需要另一套工具。知道使用哪一种工具是查找漏洞时最关键的一步。

就像日常生活的每个方面一样，很多工具可执行同一任务。这个概念也可用于执行弱点评估。有些具体到操作系统、应用程序甚至网络的工具（要看所使用的协议）。有些工具是免费的，有些不是。有些工具很直观易用，而有些则比较神秘，且文档支持很差，但拥有其它工具没有的一些功能。

使用合适的工具可能是一项艰巨的任务，最终，还是经验决定一切。如果可能的话，建立一个测试实验室，尽您所能尝试许多不同的工具，记下每一个工具的优点和缺点。查核 **README** 文件，或这些工具的手册页。此外，在互联网上查阅更多信息，如文章、分步指南，或是针对这些工具的邮件列表。

下面讨论的工具只是众多可用工具中的一个例子。

1.3.3.1. 使用 Nmap 扫描主机

Nmap 是一种常用工具，可用于判定网络的布局。**Nmap** 多年来一直被使用，它可能是收集信息时最经常使用的工具。其优秀的手册页中对其选项和使用方法进行了详细描述。管理员可以在网络上使用 **Nmap** 来查找主机系统以及打开这些系统的端口。

Nmap 是漏洞评估的第一步。您可以映射出在您网络上所有的主机，甚至可以传递选项，允许 **Nmap** 尝试对特定主机正在运行的操作系统进行识别。 **Nmap** 为制定关于使用安全服务和限制未使用服务的政策奠定了良好基础。

要安装 **Nmap** ，则须作为 **root** 用户运行 **yum install nmap** 命令。

1.3.3.1.1. 使用 Nmap

通过在所扫描机器的主机名或 **IP** 地址下输入 **nmap** 命令，**Nmap** 就可以在 shell 提示符中运行：

```
nmap <hostname>
```

例如，扫描机器的主机名 **foo.example.com**，用 shell 提示输入以下命令：

```
~]$ nmap foo.example.com
```

基本扫描（只需花几分钟，根据主机所在位置以及其他网络状况）的结果与以下结果相似：

```
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
80/tcp  open   http
113/tcp closed auth
```

Nmap 可测试最常见网络通信端口，以用于侦听或等待服务。这个常识对于想关闭不必要或未使用的服务的管理员来说，是非常有用的。

关于使用 **Nmap** 的更多信息，请参阅以下 URL 的官方主页：

http://www.insecure.org/

1.3.3.2. Nessus

Nessus 是一个可提供全方位服务的安全扫描程序。**Nessus** 的插件式结构允许用户自定义其系统和网络。与其他的扫描程序一样，**Nessus** 只能在其依赖的签名数据库中发挥作用。好在 **Nessus** 会时常更新，且具有全面报告、主机扫描以及实时漏洞搜索的功能。请记住，即使是像 **Nessus** 时常更新的强大工具，也可能会出现误报和漏报。

注意

Nessus 客户端和服务器软件需要支付订阅费才能使用。这一点已加到此文档中，以供那些有兴趣使用该程序的用户参考。

关于 **Nessus** 的更多信息，请参阅以下 URL 的官方网站：

http://www.nessus.org/

1.3.3.3. OpenVAS

OpenVAS （*Open Vulnerability Assessment System，开放式漏洞评估系统*）是一套可用于扫描漏洞和全面漏洞管理的工具和服务系统。 **OpenVAS** 框架可提供许多基于网络、桌面和命令行的工具，用于控制解决方案的不同组件。 **OpenVAS** 的核心功能是其所提供的安全扫描器，可使用超过 33,000 每日更新的网络漏洞测试（NVT， Network Vulnerability Test ）。与 **Nessus** （请参阅 第 1.3.3.2 节 “Nessus”）不同，**OpenVAS** 并不需要任何订阅费。

关于 openVAS 的更多信息，请参阅以下 URL 的官方网站：

http://www.openvas.org/

1.3.3.4. Nikto

Nikto 是一款杰出的“ *通用网关接口* ”(CGI，common gateway interface) 脚本扫描器。 **Nikto** 不仅可用于检查 CGI 漏洞，还可以躲避的方式运行，以便躲避入侵探测系统。Nikto 所提供完整的文档资料，在运行程序前，应当仔细查核。如果您有提供 CGI 脚本的网络服务器，那么**Nikto** 就是用于检查此类服务器安全的最佳资源。

关于 **Nikto** 的更多信息，可见以下 URL ：

http://cirt.net/nikto2

1.4. 安全威胁

1.4.1. 网络安全威胁

如果不能在以下方面很好地配置网络，就会增加被袭击的风险。

不安全的构架

错误配置的网络是未授权用户的主要切入点。让一个可信任并且开放的本地网络暴露于高风险的互联网上就如同开门揖盗— 有时可能什么都不会发生，但 *最终* 会有人利用这样的机会。

广播网络

系统管理员通常无法意识到其安全方案中联网硬件的重要性。简单的硬件，比如集线器和路由器，它们依赖的是广播或者非切换的原则，即，无论何时，某个节点通过网络将数据传送到接收节点时，集线器或者路由器都会向接受者发送该数据包的广播并处理该数据。这个方法是外部入侵者以及本地主机的未授权用户进行地址解析（*ARP*）或者介质访问控制 （*MAC*）地址嗅探的最薄弱的环节。

集中管理的服务器

另一个潜在的联网陷阱是使用集中管理的计算机。很多企业常用的削减支出的方法是将所有服务都整合到一个强大的机器中。这很方便，因为它容易管理，同时费用相对多台服务器配置来说更加便宜。但是集中管理的服务器也会造成网络单点的失败。如果中央服务器被破坏，则会造成整个网络完全不能使用，甚至更糟糕的是，有可能造成数据被篡改或者被盗。在这些情况下，中央服务器就成为访问整个网络的开放通道。

1.4.2. 服务器安全威胁

服务器安全与网络安全同样重要，因为服务器通常拥有机构的大量重要数据。如果服务器被破坏，则其所有内容都可被破解者偷走或者任意篡改。下面的小节详细论述了一些主要问题。

未使用的服务及开放端口

一般系统管理员安装操作系统时不会注意实际安装了哪些程序。这可能会造成一些问题，因为可能安装了并不需要的服务，而这些服务可能被安装和配置了默认设置，并且有可能被开启。这样可能会造成在服务器或者工作站中运行不必要的服务，比如 Telnet、DHCP 或者 DNS，而管理员并没有意识到这一点，从而造成不必要的流量经过该服务器，甚至成为破解者进入该系统的潜在通道。有关关闭端口以及禁用未使用服务的详情请参考 〈第 4.3 节 “安全服务”〉。

未打补丁的服务

默认安装包括的大多数服务器应用程序都是经过严格测试的安全软件。经过在产品环境中的长期应用后，将彻底改进其代码，并会发现和修复很多 bug。

但是世界上不存在十全十美的软件，而且总是有可以改进的空间。另外，较新的软件通常不会进行您希望的严格测试，因为它最近才用于产品环境，或者因为它可能不如其它服务器软件那么受欢迎。

开发者和系统管理员经常会在服务器应用程序中找到可开发的 bug，并将该信息在 bug 跟踪和与安全相关的网页中发布，比如 Bugtraq 邮件列表（http://www.securityfocus.com）或者计算机紧急反应团队（CERT）网站（http://www.cert.org）。虽然这些机制是警告社区安全隐患的有效方法，但关键还是要系统管理员可正确为其系统打补丁。这是事实，因为破解者也可访问同样的弱点跟踪服务，并在可能的情况下使用那些信息破解未打补丁的系统。良好的系统管理需要警惕、持续的 bug 跟踪，同时严格的系统维护可保证您有一个更安全的计算环境。

有关保持系统更新的详情请参考〈 *第 3 章 及时更新系统*〉。

疏忽的管理

管理员不能为其系统打补丁是服务器安全的最大威胁之一。根据“*系统管理、审核、网络、安全研究院*”（即 *SANS*）资料，造成计算机安全漏洞的主要原因是“让未经培训的人员维护系统安全，不为其提供培训，也没有足够的时间让其完成这项工作。” [1]这指的是那些缺乏经验的管理员以及过度自信或者缺乏动力的管理员。

有些管理员无法为其服务器和工作站打补丁，而有些则不会检查来自系统内核或者网络流量的日志信息。另一个常见的错误是不修改默认的密码或者服务密钥。例如：有些数据包使用默认的管理员密码，因为数据库开发者假设系统管理员会在安装后立刻更改这些密码。如果数据库管理员没有更改这个密码，那么即使是缺乏经验的破解者也可使用广为人知的默认密码获得该数据库的管理特权。这里只是几个疏忽管理造成服务器被破坏的示例。

自身有安全问题的服务

即使最谨慎的机构，如果选择自身就有安全问题的网络服务，也可能成为某些安全漏洞的受害者。例如：很多服务的开发是假设在可信网络中使用，一旦这些服务可通过互联网使用，即其本身变得不可信，则这些假设条件就不存在了。

一种不安全的网络服务那些使用不加密用户名和密码认证的服务。Telnet 和 FTP 就是这样的服务。如果数据包嗅探软件正在监控远程用户间的数据流量，那么这样的服务用户名和密码就很容易被拦截。

此类服务还更容易成为安全业内名词 “*中间人*” 攻击的牺牲品。在这类攻击中，破解者会通过愚弄网络中已经被破解的名称服务器，将网络流量重新指向其自己的机器而不是预期的服务器。一旦有人打开到该服务器的远程会话，攻击者的机器就成为隐形中转人，悄无声息地在远程服务和毫无疑心的用户间捕获信息。使用这个方法，破解者可在服务器或者用户根本没有意识到的情况下收集管理密码和原始数据。

另一个不安全的类型是网络文件系统和信息服务，比如 NFS 或者 NIS，它们是专门为 LAN 使用而开发的，但遗憾的是 WAN 网络（用于远程用户）现在也使用。NFS 默认情况下没有配置任何验证或者安全机制以防止破解者挂载到 NFS 共享并访问其中包括的内容。NIS 同样也有重要信息，网络中的每台计算机都必须了解这些信息，其中包括密码和文件权限，而且它们是纯文本 ASCII 或者 DBM（ASCII 衍生的）数据库。获得这个数据库访问的破解者可访问网络中的每个帐户，包括管理员帐户在内。

默认情况下 Red Hat Enterprise Linux 7 的发布是关闭此类服务的。但是由于管理员通常会发现他们必须使用这些服务，所以谨慎的配置很关键。有关使用安全方式设定服务的详情请参考 <第 4.3 节 “安全服务” >。

1.4.3. 工作站和家庭 PC 安全威胁

工作站和家庭 PC 可能比网络或者服务器受到攻击的可能性小，但因为它们通常都有敏感数据，比如信用卡信息，因此它们也是破解者的目标。工作站还可在用户不知情的情况下在合作攻击中被指派作为破解者的“奴隶”机器。因此，了解工作站的安全漏洞可让用户免于经常重新安装操作系统，或者防止数据被盗。

不安全的密码

不安全的密码是攻击者获取系统访问的最简单的方法之一。有关如何在生成密码时避免常见缺陷，详情请参考 < 第 4.1.1 节 “密码安全” >。

有漏洞的客户端应用程序

虽然管理员可保障服务器安全并进行修补，但这并不意味着远程用户在访问该服务器时是安全的。例如：如果该服务器通过公共网络提供 Telnet 或者 FTP 服务，那么攻击者就可以捕获通过该网络的用户名和密码，然后使用该帐户信息访问远程用户的工作站。

即使使用安全协议，比如 SSH，如果远程用户没有即时更新其客户端应用程序，那么对于某些攻击来说，他们也是不堪一击的。例如：v.1 SSH 客户端无法抵御恶意 SSH 服务器的 X 转发攻击。一旦它连接到该服务器，那么攻击者就可悄无声息地捕获所有该客户端通过该网络执行的击键和鼠标动作。这个问题在 v.2 SSH 协议中得到了解决，但这也取决于该用户是否留意什么样的应用程序有此类漏洞并根据需要更新它们。

<第 4.1 节 “计算机安全”>中详细论述了管理员和家庭用户应采取什么步骤限制计算机工作站的安全漏洞。

1.5. 常见的漏洞和攻击

表 1.1 “常见漏洞”详细论述一些入侵者访问机构网络资源的最常见漏洞和切入点。这些常见漏洞的重点是解释了攻击是如何进行的以及管理员怎样正确保护其网络免受类似攻击。

表 1.1. 常见漏洞

漏洞	描述	备注
空值或者默认密码	使管理密码空白或者使用有产品供应商所设定的默认密码。这在硬件中非常常见，比如说路由器和防火墙，但是这些服务在Linux上运行可包含默认管理者密码（通过 Red Hat Enterprise Linux 7 并没有产品附赠）。	通常与联网硬件有关，比如路由器、防火墙、VPN 以及网络附带存储（NAS）装置。 常见于传统操作系统，特别是那些捆绑服务（比如 UNIX 和 Windows）。 管理员有时会仓促创建特权用户账户，并将密码设为空，这样就为发现该账户的恶意用户提供了最佳切入点。
默认共享密钥	安全服务有时可因开发或者评估测试为目的而打包默认安全密钥。如果不更改这些密钥并将其放在互联网的产品环境中，拥有同一默认密钥的*所有*用户都可访问共享密钥资源及其所包含的敏感信息。	预先配置到安全服务器装置中，这在无线访问点中最常见。
IP 欺诈	远程机器会装作是您本地网络中的节点，查找服务器弱点并安装后门程序或者木马以获得您网络资源控制权。	欺骗是很困难的，因为它包括攻击者预测的 TCP/IP 序列号以便与目标系统链接，但有些工具可帮助攻击者完成这样的任务。 相比 PKI 或者其它在 **ssh** 或者 SSL/TLS 中所使用加密认证的其它形式，我们不建议您使用根据目标系统运行的使用*根据资源认证*技术的服务（比如 **rsh**、**telnet**、FTP 及其它）。
窃听	通过窃听两个节点之间的连接收集通过网络的两个活跃节点的数据。	这类攻击最可能在纯文本传输协议中有效，比如 Telnet、FTP 和 HTTP 传输。 远程攻击者必须可访问 LAN 中受威胁的系统以便执行此类攻击。通常黑客会使用活跃攻击（比如 IP 欺诈或者中间人）威胁 LAN 中到系统。 预防方法包括使用加密密钥交换到服务、一次性密码或者加密的认证防止密码欺诈，还建议您在传输过程中使用强大加密。

漏洞	描述	备注
服务弱点	攻击者找到在网络中运行的服务的缺陷或者漏洞，通过这些弱点攻击者可威胁整个系统及其所有数据，并可能威胁该网络中到其它系统。	基于 HTTP 的服务，比如 CGI 对于远程命令执行甚至互动的 shell 访问来说都是容易受到攻击的。即时作为非特权用户，比如 “nobody” 运行 HTTP 服务，也可读取类似配置文件以及网络映射等信息，或者攻击者可启动拒绝服务攻击，这样就可耗尽系统资源或者让其他用户无法使用系统资源。 在开发和测试阶段服务有时候会有一些被忽视的弱点，这些弱点（比如*缓冲溢出*，攻击者可使用任意值填满程序的内存缓冲从而使服务崩溃，给攻击者一个互动命令提示符即可允许其执行所有任务）可让攻击者完全拥有管理控制。 管理员应确定不要作为 root 用户运行那些服务，且应该随时注意供应商或者类似 CERT 和 CVE 安全性机构为程序提供的补丁会勘误更新。
应用程序弱点	攻击者要查找桌面和工作站程序（比如电子邮件客户端）的缺陷，并执行任意代码，植入木马以便进一步破坏，或者使系统崩溃。如果被破坏的工作站对剩余网络有管理特权，则会发生进一步的攻击。	工作站和桌面更容易被攻击因为工作人员没有防止或者探测这种侵害的专业知识或者经验。有必要在安装未授权软件或者打开不明电子邮件附件时告知他们可能存在的危险。 可使用一些防护软件以便电子邮件客户端软件不会自动打开或者执行附件。另外，通过红帽网络；或者其它系统管理服务自动更新工作站软件可减轻多种安全性部署的负担。
拒绝服务（DoS）攻击	攻击者或者一组攻击者通过向目标主机（可以是服务器、路由器或者工作站）发送未授权数据包联合攻击某个机构的网络或者服务器资源。这样可迫使合法用户无法使用该资源。	在美国大多数报告的 DoS 案例发生在 2000 年。一些有很高流量的商业和政府网站受到 ping flood 攻击而变得不可用，这些攻击是利用几个被破坏的系统使用宽带连接作为 *zombies*，或者重新指向广播节点进行的。 通常会伪装源数据包（也会重新广播），让调查攻击的真实源变得困难。 使用 **`iptables`** 和网络入侵探测系统，比如 **`snort`** 进行进入过滤（IETF rfc2267）的优点是可帮助管理员追踪并阻止发布的 DoS 攻击。

[1] http://www.sans.org/security-resources/mistakes.php

第 2 章 安装的安全提示

当您第一次将 CD 或者 DVD 放入磁盘驱动器安装 Red Hat Enterprise Linux 7 时就由安全性问题。开始就安全配置您的系统可让您今后实施额外的安全性设置变得更轻松。

2.1. 安全 BIOS

使用密码保护 BIOS（或者与 BIOS 对等的程序）以及引导装载程序可防止未授权用户使用可移动介质，或者通过单用户模式获取 root 特权引导机器而获得对该系统的物理访问。您应该采用的防止此类攻击的安全工具，取决于该工作站中的信息敏感性以及机器的位置。

例如：如果是在贸易展览中使用，且不包含任何敏感信息，那么防止此类攻击就不那么重要。但是如果在同一贸易展览中，某雇员的笔记本电脑中有该企业网络的专用未加密的 SSH 密钥，且没有小心保管，那么就可能导致严重的安全泄漏，并为整个公司造成无法预料的损失。

如果该工作站位于授权或者可信用户可以访问的位置，那么保障 BIOS 或者引导装载程序安全就不那么紧要。

2.1.1. BIOS 密码

使用密码保护计算机的 BIOS 主要有两个原因 [2]：

1. *防止更改 BIOS 设置* — 如果某个入侵者可访问该 BIOS，他们就可以将其设定为从磁盘或者光盘引导。这就让他们可以进入安全模式或者单用户模式，继而让他们可以在该系统中启动随机进程或者复制敏感数据。

2. *防止系统引导* — 有些 BIOS 允许引导过程中的密码保护。当激活 BIOS 时，攻击者会在 BIOS 启动引导装载程序前被迫输入密码。

因为不同计算机生产商提供的设置 BIOS 的方法不同，具体步骤请查询计算机手册。

如果您忘记了 BIOS 密码，您可以使用主板中的跳线或者断开 CMOS 电池连接重新设置该密码。因此，请尽可能锁上您的机箱。但是在尝试断开 CMOS 电池前请查看计算机或者主板手册。

2.1.1.1. 保证非 x86 平台安全

其它构架使用不同的程序执行那些与 x86 系统中 BIOS 基本对等的低层任务。例如：Intel® Itanium™ 计算机使用*可扩展固件接口*（*EFI*）shell。

有关在其它构架中使用密码保护类似 BIOS 程序的使用说明，请参考生产商的解释说明。

2.2. 其他资源

更多有关安装的详情，请参考〈Red Hat Enterprise Linux 7 安装手册〉。

[2] 因为不同制造商提供的系统， BIOS 会有所不同，有些可能不支持任何类型的密码保护，而其它可能支持某种类型，但不支持其它类型。

第 3 章 及时更新系统

这一章阐述了及时更新系统的过程，它包括计划和配置安装安全更新的方法，应用最新升级包所引入的变更，并且使用红帽 Red Hat 客户门户来了解安全更新公告。

3.1. 维护安装的软件

如果发现安全漏洞，为了限制潜在的安全威胁必须更新受影响的软件。如果该软件是 Red Hat Enterprise Linux 分布现在支持的软件包的一部分，Red Hat 尽快发布修复漏洞的更新软件包。

有关特定的安全漏洞的公告经常会伴有补丁（或者源代码）来解决问题。这个补丁会直接应用于 Red Hat Enterprise Linux 软件包并且在经过测试后作为勘误更新来公布。然而，如果公告不包括补丁， Red Hat 的开发者会先和软件维护者共同来解决这个问题。一旦解决了该问题，软件包就会在测试后作为勘误更新来公布。

如果您系统中的使用软件发布勘误更新，我们强烈建议您尽快更新受影响的软件包以便尽量减少系统出现潜在漏洞的时间。

3.1.1. 计划和配置安全更新

所有的软件都包含 bug，通常这些 bug 会造成漏洞让恶意用户侵入您的系统。未更新软件包是造成电脑入侵的共同原因。及时地安装安全补丁计划能快速删除被找到的漏洞，这样它们就不会被利用。

当安全更新可用时，安排安装更新并进行测试。我们需要使用其他的控件在发布更新以及系统安装更新期间保护我们的系统。这些控件取决于每一个匹配的漏洞，但是也包括其他的防火墙原则，外部防火墙的使用和软件设置的变化。

通过使用勘误机制来修复支持软件包中的 bugs。勘误包含一个或者多个 RPM 软件包，并伴有简单的解释说明每一个特定的勘误所处理的问题。所有的勘误都通过 **Red Hat 订阅管理** 服务分配给积极订阅的客户。处理安全问题的勘误被称为 *Red Hat 安全建议*。

3.1.1.1. 使用 Yum 的安全特征

Yum 软件包管理包含许多与安全相关的特征，可以用来搜索、列表、显示和安装安全勘误。这些特征有可能使用 **Yum** 来安装安全更新。

在您的系统中检查可用的安全有关的更新，请以 **root** 运行以下的命令：

```
~]# yum check-update --security
Loaded plugins: langpacks, product-id, subscription-manager
rhel-7-workstation-rpms/x86_64                    | 3.4 kB  00:00:00
No packages needed for security; 0 packages available
```

请注意以上命令是在非交互状态下运行，所以它可以在脚本中自动检测是否有可用更新。当安全更新可用时，命令会返回 100 的退出值。当安全更新不可用时，则变为0 。一旦遭遇错误，它就返回 1。

在模拟情况下，使用以下命令值安装安全有关的更新：

```
~]# yum update --security
```

使用 **updateinfo** 子命令来显示或者依照可用更新的储存库所提供的信息。**updateinfo** 子命令本身接受许多命令，其中有与安全相关的使用方法。为获取这些命令的概述，请参考 表 3.1 “可用安全相关的命令以及 yum updateinfo”。

表 3.1. 可用安全相关的命令以及 yum updateinfo

命令	描述
***advisory* [advisories]**	显示有关一个或者多个建议。使用一个或者多个建议号码来替代 *advisory*。
cves	显示子设备的信息，关于 CVE (*常见弱点与揭露*)。
security 或者 **sec**	显示所有安全相关的信息。
severity 或者 **sev** *severity_level*	在提供的 *severity_level* 中显示与安全相关的软件包的信息。

3.1.2. 更新和安装软件包

当更新系统中的软件包时，从可信资源下载更新是很重要的。攻击者可轻易重建本应用来解决问题的同一版本号的软件包，通过不同的安全漏洞并发布到互联网中。如果发生这种情况，采取例如验证针对原始 RPM 的文件之类的安全措施是无法探测到漏洞。因此，只从可信来源下载 RPMs 是非常重要的，例如从 Red Hat 下载并检查软件包签名以确定其完整性。

想要更多有关如何使用 **Yum** 软件包管理器的信息，请参考《红帽企业版Linux 7 系统管理指南》。

3.1.2.1. 验证签名的软件包

所有 Red Hat Enterprise Linux 的软件包都标有 Red Hat**GPG** 密钥。**GPG** 代表 **GNU 隐私防护**、或者 **GnuPG**，是用来确保分布式文件真实性的免费软件包。如果验证软件包签名失败，则软件包可能被修改，因此就不能信任此软件包。

Yum 软件包管理器允许所有安装和更新软件包进行自动验证，此为默认的特性。为了在您的系统中配置这个选项，在 **/etc/yum.conf** 配置文件中就必须把**gpgcheck** 配置指令设定为 **1**。

在您的文件系统中，使用以下命令手动验证软件包信息。

```
rpmkeys --checksig package_file.rpm
```

请参考〈产品签名 (GPG) 密钥〉有关 Red Hat 客户门户的文章，以获取其他有关 Red Hat 软件包签名实践方法的信息。

3.1.2.2. 安装签名的软件包

从您的文件系统中安装验证的软件包（请参考〈第 3.1.2.1 节 “验证签名的软件包”〉，获取更多有关如何验证软件包的信息）。作为 **root** 用户，请使用 **yum install** 命令。

```
yum install package_file.rpm
```

使用 Shell glob 即刻安装多个软件包。例如，以下命令在现有的目录中安装所有的 **.rpm** 软件包。

```
yum install *.rpm
```

重要

在安装任何安全勘误之前，请确保阅读包含在勘误报告中的所有具体步骤并依次执行。请参考〈第 3.1.3 节 “应用安装更新所引入的变化”〉以获取有关勘误更新所引入变化的基本指令。

3.1.3. 应用安装更新所引入的变化

下载并安装安全勘误和更新后，停止使用旧的软件并开始使用新软件是很重要的。如何做取决于所安装软件的类型。以下列表列出了软件常规分类并提供在软件包升级后使用更新版本的步骤。

备注

通常重启系统是保证使用软件最新版本的最确定的方法，但是并不经常提出此要求，而且系统管理员也无法经常执行这个操作。

应用程序

用户空间应用程序可以是由系统用户启动的任意程序。通常此类程序只有在用户、脚本或者自动任务工具启动时才使用。

当更新这种用户空间程序后，停止系统中该程序的所有事务，并再次启动该程序以便使用更新的版本。

内核

内核是 Red Hat Enterprise Linux 7 操作系统的核心软件组建。它对访问内存、处理器及外围设备进行管理，并调度所有任务。

由于其核心角色，所以无法在不停机的情况下重启 Kernel。因此只有重启系统后方可使用 Kernel 的更新版本。

KVM

当更新 *qemu-kvm* 与 *libvirt* 软件包时，必须停止所有的客户虚拟机，重载相关的虚拟模块（或者重新启动主体系统）并且重启虚拟机。

使用 **lsmod** 命令来确定从以下文件中下载哪个模块：**kvm**、**kvm-intel** 或者 **kvm-amd**。然后使用 **modprove -r** 命令进行删除，之后使用 **modprobe -a** 命令重新加载受影响的模块。例如：

```
~]# lsmod | grep kvm
kvm_intel             143031  0
kvm                   460181  1 kvm_intel
~]# modprobe -r kvm-intel
~]# modprobe -r kvm
~]# modprobe -a kvm kvm-intel
```

共享库

共享库是代码单元，例如 **glibc**，它们可用于很多应用程序和服务。使用共享库的应用程序通常在启动时载入共享代码，因此所有使用更新库的应用程序都必须停止并重启。

为确定某个特定的库相联的正在运行的应用程序，请使用 **lsof** 命令：

```
lsof library
```

例如：为确定与 **libwrap.so.0** 库相联的正在运行的应用程序，请输入：

```
~]# lsof /lib64/libwrap.so.0
COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF     NODE NAME
```

```
pulseaudi 12363 test mem    REG  253,0    42520 34121785
/usr/lib64/libwrap.so.0.7.6
gnome-set 12365 test mem    REG  253,0    42520 34121785
/usr/lib64/libwrap.so.0.7.6
gnome-she 12454 test mem    REG  253,0    42520 34121785
/usr/lib64/libwrap.so.0.7.6
```

这个命令会返回所有正在运行的使用 **TCP** 包装进行主机访问控制的程序。因此，如果更新 *tcp_wrappers* 软件包，则必须停止并重启列出的程序。

系统服务

系统服务是在引导过程中经常启动的可保留的服务器程序。系统服务的示例包括 **sshd** 或者 **vsftpd**。

只要机器运行，这些程序就经常被保留在内存中，每项更新的系统服务在软件包更新之后，必须停止并重启。这可以透过 **root** 用户使用**systemctl** 命令来完成：

```
systemctl restart service_name
```

使用您所希望重启服务的名称来覆盖 *service_name*，例如 **sshd**。

其他软件

请按照以下的说明，该说明是由链接到以下正确更新的应用程序所概括的。

- **红帽目录服务器**：请参考《发行备注》以获取在《Red Hat目录服务器产品文档页》中的正在讨论的 Red Hat Directory 服务器版本。
- **红帽企业虚拟管理器** ：请参考《Red Hat Enterprise Linux 7 安装指南》以获取在《Red Hat Enterprise 虚拟产品文档页》中的正在讨论的 Red Hat Enterprise 虚拟化版本。

3.2. 其他资源

如需获取更多有关安全更新、安装安全更新方法、Red Hat Customer Portal（红帽客户门户）以及相关的主题的信息，请参考以下列出的资源。

安装的文挡

- yum(8) — **Yum** 用于解释程序包管理器的手册页提供有关在您的系统中使用 **Yum** 安装、更新以及删除软件包方法的信息。
- rpmkeys(8) —**rpmkeys** 用于解释实用程序的手册页解释这款程序可用来验证下载程序包真伪的方法。

在线文档

- 《Red Hat Enterprise Linux 7 系统管理员指南》 — 《*系统管理员指南*》解释**Yum** 以及 **rpm** 程序可用来安装、更新和删除在 Red Hat Enterprise Linux 7 系统中软件包。
- 《Red Hat Enterprise Linux 7 SELinux 用户和管理员的指南》 —《*SELinux 用户和管理员指南*》解释 **SELinux** *强制性访问控制* 机制的配置。

红帽客户门户

- 红帽客户门户 — 客户门户主页包含通往最重要资源以及有关可用新内容的更新的链接。

- 安全联系和程序— 提供有关Red Hat安全响应团队信息以及何时与之联系的操作说明。
- 红帽安全博客 — 提供来自 Red Hat 红帽安全专业人员的与安全有关的最新问题的文章。

参见

- 〈第 2 章 *安装的安全提示*〉描述了如何在开始阶段安全地配置您的系统，让系统更容易执行后来的其外安全设置。
- 〈第 4.8.2 节 “创建 GPG 密钥”〉描述了如何创作个人 **GPG** 密钥来鉴定您的通讯。

第 4 章 用工具和服务强化您的系统

4.1. 计算机安全

密码是 Red Hat Enterprise Linux 7 用来确认用户身份的主要方法。这是为什么密码安全对保护用户、工作站以及网络是那么的重要。

出于安全目的，安装程序会对系统进行配置，从而可使用 *安全哈希算法 512*（*SHA512*）和影子密码。强烈建议您不要更改这个设置。

如果在安装过程中取消选择影子密码，则所有密码都会以单向哈希的形式保存在可读的 **/etc/passwd** 文件中，这样就使得该系统在离线密码破解攻击面前变得很脆弱。如果入侵者可作为常规用户访问该机器，他就可以将 **/etc/passwd** 文件复制到他自己的机器中，并对其运行密码破解程序。如果该文件中存在不安全的密码，那么密码被破解只是时间问题。

影子密码可通过在 **/etc/shadow** 文件中保存密码哈希消除这种类型的攻击，该文件只能由 root 用户读取。

这就迫使潜在的攻击者要登录该机器中的远程服务（比如 SSH 或者FTP）进行远程密码破解。这种暴力破解速度会慢很多，并且会留下明显的痕迹，因为在系统文件中会出现几百条失败登录尝试。当然，如果攻击者在夜间对使用薄弱密码的系统进行攻击，那么他可能在黎明前就可获得访问权限，并修改日志文件以掩盖其踪迹。

除要考虑格式和存储外，内容也是要考虑的问题。用户如要保护其帐户不被破解，最重要的是创建强大的密码。

4.1.1. 密码安全

4.1.1.1. 创建强大的密码

要创建一个安全可靠的密码，用户须牢记长密码比短而复杂的密码强。创建一个仅有八个字符的密码，就算它含有数字、特殊符号和大写字母，这也不是个好主意。优化密码破解工具，例如约翰开膛手（John The Ripper），以便破解连人也难以记住的密码。

在信息论中，熵 (entropy) 表示的是不确定性的量度，与随机变量有关，并以“位”为单位来表现信息量度。熵值越高，密码就越安全。根据美国国家标准与技术研究院（National Institute of Standards and Technology， NIST）特别出版物 <电子认证指南> （NIST SP 800-63-1，Electronic Authentication Guideline），在一本收录 5 万个常用密码的字典里，某一密码没有出现过的熵值应该至少有 10 位。这样说来，一个由 4 个随机字组成的密码，其熵值大约有 40 位。一个由多个字组成的密码，旨在增强安全性，也被称为“*密码短语*”，例如：

```
randomword1 randomword2 randomword3 randomword4
```

如果系统强制要求使用大写字母、数字或特殊符号，那么采用上述建议的密码短语可以轻易地被修改，例如修改第一个字符为大写字母，在末尾增添 "**1!**"。要注意这样的修改 *并不能* 显著地增强密码短语的安全性。

创建密码的另一种方法是使用密码生成器。**pwmake** 是一个命令行程序，用于生成随机密码，可由四种字符–组成：大写字母、小写字母、数字和特殊符号。其功能让您能够详细了解用于生成密码的具体熵值。而熵值产生于 **/dev/urandom**。这项功能让您能 指定最小熵值为 56 位，这对于不常出现暴力破解的系统和服务密码，这个熵值已足够。对于攻击者无法直接访问哈希密码文件的运用程序， 64 位就足以适用于此类运用程序。当攻击者可能获取直接访问哈希密码的权限，或密码被用作加密钥匙时，对于此类情况应使用 80 到 128 位。如果您无法明确指定一个具体的熵值，**pwmake** 将会使用默认值。创建一个 128 位的密码，则要运行下列命令：

```
pwmake 128
```

虽然有不同的方法可以创建一个安全可靠的密码，但都要避免以下不明智的做法：

- 使用字典里的单词，外语单词，逆序单词，或仅使用数字。
- 使用少于 10 字符的密码或密码短语。
- 使用键盘布局的系列键。
- 写下您的密码。
- 在密码中使用个人信息，如出生日期、周年纪念日、家庭成员姓名、或宠物名字。
- 在不同的机器上使用相同的密码短语或密码。

虽然创建密码非常重要，但合理地管理密码，特别是对于大型机构中的系统管理员而言，这也同样重要。下面的小节详细介绍了在机构中如何很好地创建并管理用户密码。

4.1.1.2. 强制使用强大的密码

如果一所机构拥有大量的用户，那么系统管理员有两个基本选择可用于强制使用强大的密码。他们可以为用户创建密码，或是他们可以让用户创建他们自己的密码，同时验证密码是否拥有足够的强度。

为用户创建密码，就要确保这个密码是好密码。但随着机构的发展，这变成了一项艰巨的任务。这也增加了用户的风险，由于他们要写下他们的密码，因而这就暴露了密码。

基于这些原因，大多数系统管理员更喜欢让用户创建自己的密码，但积极地验证这些密码是否足够强大。在某些情况下，管理员可能会强制用户定期更改密码，防止密码过期。

当用户被要求创建或更改密码时，可以使用 **passwd** 命令行实用程序，这就是*PAM-*检测软件（*可插入验证模块（Pluggable Authentication Modules）*），可检查密码是否过短或是否容易被破解。这个检查过程是由 **pam_pwquality.so** PAM 模块执行的。

注意

红帽企业版 Linux 7 中，**pam_pwquality** PAM 模块取代了 **pam_cracklib**，这原先用于红帽企业版 Linux 6 作为密码质量检测的默认模块。它与 **pam_cracklib** 使用相同的后端。

pam_pwquality 模块是根据一系列规则，用于检查密码的强度。其程序有两个步骤：首先，它检查所提供的密码是否能在字典中找到。如果不能，它将继续进行另外一些额外检查。**pam_pwquality** 与 其他 PAM 模块一起堆叠在**/etc/pam.d/passwd** 文件下的 **密码** 部分。而自定义规则将在 **/etc/security/pwquality.conf** 配置文件中具体说明。至于这些检查步骤的完整列表，请参阅 **pwquality.conf (8)** 手册页。

例 4.1. 在 pwquality.conf 中密码强度检查的参数配置

为了能够使用 **pam_quality** ，须在**/etc/pam.d/passwd** 文件下的 **password** 堆叠中添加以下命令行：

```
password    required    pam_pwquality.so retry=3
```

选择这些检查步骤有明确的要求，要每行一项。例如，要求一个密码的长度至少有 8 个字符，包含全部四种字符，则须添加以下命令行到**/etc/security/pwquality.conf** 文档：

```
minlen=8
minclass=4
```

要设置一个密码强度检查以检测是否有连续或重复的字符，则须在 **/etc/security/pwquality.conf** 中添加以下命令行：

```
maxsequence=3
maxrepeat=3
```

在本例中，输入的密码不能够含有超过 3 个连续字符，如“**abcd**”或“**1234**”。此外，完全相同的连续字符也不能超过 3 个。

注意

由于root 用户是施行密码创建规则的人，尽管有出现警告消息，他也能够为自己或普通用户设置任何密码。

4.1.1.3. 密码有效期的参数配置

密码有效期是另一个系统管理员用来保护在机构中防止不良密码的技术。密码有效期的意思就是在指定时段后（通常为 90 天），会提示用户创建新密码。它的理论基础是如果强制用户周期性修改其密码，那么破解的密码对与入侵者来说只在有限的时间内有用。密码有效期的负面影响是用户可能需要写下这些密码。

在 Red Hat Enterprise Linux 7 中有两个用来指定密码有效期的主要程序：**chage** 命令或者图形 **用户管理者**（**system-config-users**）应用程序。

重要

在红帽企业版 Linux 7 中，影子口令是默认启用的。更多信息，请参阅《红帽企业版 Linux 7 系统管理员指南》。

chage 命令的 **-M** 选项指定该密码有效的最长天数。例如：要将用户的密码设定为 90 天内有效，请执行以下命令：

```
chage -M 90 <username>
```

在上面的命令中使用用户名称替换 *<username>*。要禁用密码过期功能，通常在 **-M** 选项后使用值 **99999**（这相当于 273 年多一点）。

关于 **chage** 命令的可使用选项的更多信息，请参阅下表。

表 4.1. chage 命令行选项

选项	描述
--hesiodlhs=*<lhs>*	指定了从 1970 年 1 月 1 日密码更改后的天数
--hesiodlhs=*<lhs>*	指定帐户被锁的日期，以年－月－日的格式出现。除了使用日期，还可以使用从 1970 年 1 月 1 日以来的天数。
--hesiodlhs=*<lhs>*	指定了在密码过期后，但在锁住帐户前的非活跃天数。如果数值是 0，密码过期后帐户不会被锁住。
-l	列出当前账户的过期设置参数。

选项	描述
--hesiodlhs=<*lhs*>	指定了用户必须修改密码的最小天数间隔。如果数值为 **0** ，则密码未到期。
--hesiodlhs=<*lhs*>	指定了有效密码的最大天数。当此选项指定的天数加上 **-d** 选项指定的天数未到当前日期，用户必须在使用账户前修改密码。
--hesiodlhs=<*lhs*>	指定在密码到期日期之前对用户发出警告的天数。

您还可以使用 **chage** 命令以互动形式修改多个密码过期功能以及帐户信息。请使用以下命令进入互动模式：

```
chage <username>
```

以下是使用这个命令的示例互动会话：

```
~]# chage juan
Changing the aging information for juan
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
```

您可以在用户首次登录时，对密码进行参数配置，使密码过期。这就可迫使用户及时修改密码。

1. 设置初始密码。有两种常用的方法可实现这个步骤：您可以指定默认的密码，或使用空密码。

   要指定默认的密码，则须作为 **root** 用户使用 shell 提示符打出下列信息 ：

   ```
   passwd username
   ```

   或者，您可以分配一个空值密码，而不要一个原始密码。如果想要这样进行的话，请使用以下命令：

   ```
   passwd -d username
   ```

   

   警告

   尽管使用空密码十分便利，却是极不安全的做法。因为任何第三方都可以先行登录，使用这个不安全的用户名进入系统。可能的话，请避免使用空密码。如果无法不使用的话，请一定要确保用户在未用空密码锁定账户前登录。

2. 要迫使密码即刻到期，则须作为 **root** 用户运行以下命令：

   ```
   chage -d 0 username
   ```

   这个命令会将密码上次作出改动的日期设定为（1970 年 1 月 1 日）这个时间。这样，无论有什么密码到期政策，它都会迫使密码作出即时到期这一行动。

在用户 初次登录时，则立即会提示输入新密码。

您还可以使用图形 **用户管理者** 程序创建密码过期策略，如下。请注意：您需要管理员特权执行这个过程。

1. 请点击面板中的 **系统** 菜单，指向 **管理** 并点击 **用户和组群** 显示用户管理器。您还可在 shell 提示符后输入命令 **system-config-users**。
2. 请点击 **用户** 标签，并选择用户列表中需要的用户。
3. 请点击工具栏中的 **首选项**显示用户属性对话框（或者选择**文件**菜单中的**首选项**）。
4. 点击 **密码信息** 标签，并选择 **启用密码过期** 单选框。
5. 在 **多少天前需要更改** 字段输入所需值，并点击 **确定**。

4.1.2. 锁定未激活的用户账户

4.1.3. 登录尝试失败后锁定用户账户

在红帽企业版 Linux 6 中， **pam_faillock** PAM 模块允许系统管理员锁定在指定次数内登录尝试失败的用户账户。限制用户登录尝试的次数主要是作为一个安全措施，旨在防止可能针对获取用户的账户密码的暴力破解。

通过 **pam_faillock** 模块，将登录尝试失败的数据储存在 **/var/run/faillock** 目录下每位用户的独立文件中。

注意

在登录尝试失败的文件中，命令行的顺序很重要。在此顺序中有任何改变都会导致所有用户账户的锁定。当使用了 **even_deny_root** 选项，也会导致 root 用户账户的锁定。

根据这些步骤对账户锁定进行参数配置：

1. 要实现在三次失败尝试后，对任何非 root 用户进行锁定，并在十分钟后对该用户解锁，则须添加以下命令行到 **/etc/pam.d/system-auth** 文件和**/etc/pam.d/password-auth** 文件中的 **auth** 区段：

```
auth        required      pam_faillock.so preauth silent audit
deny=3 unlock_time=600
auth        sufficient    pam_unix.so nullok try_first_pass
auth        [default=die] pam_faillock.so authfail audit deny=3
unlock_time=600
```

2. 在前一步骤指定的两个文件中的 **account** 区段中添加以下命令行：

```
account     required      pam_faillock.so
```

3. 要让账户锁定也适用于 root 用户，则须在 **/etc/pam.d/system-auth** 文件和 **/etc/pam.d/password-auth** 文件中的**pam_faillock** 条目里添加 **even_deny_root** 选项：

```
auth        required      pam_faillock.so preauth silent audit
deny=3 even_deny_root unlock_time=600
auth        sufficient    pam_unix.so nullok try_first_pass
auth        [default=die] pam_faillock.so authfail audit deny=3
```

```
even_deny_root unlock_time=600
auth        sufficient    pam_faillock.so authsucc audit deny=3
even_deny_root unlock_time=600
```

用户 **john** 在前三次登录失败后，尝试第四次登录时，他的账户在第四次尝试中被锁定：

```
[yruseva@localhost ~]$ su - john
Account locked due to 3 failed logins
su: incorrect password
```

要让一个用户即使在数次登录失败之后，其账户仍未被锁定，则须在 **/etc/pam.d/system-auth** 和 **/etc/pam.d/password-auth** 中的 "first call of" pam_faillock 之前添加以下命令行。也可以用**user1**, **user2**, **user3** 代替实际用户名。

```
auth [success=1 default=ignore] pam_succeed_if.so user in
user1:user2:user3
```

要查看每个用户的尝试失败次数，则须作为 root 用户运行以下命令行：

```
[root@localhost ~]# faillock
john:
When                Type  Source
Valid
2013-03-05 11:44:14 TTY   pts/0
V
```

要解锁一个用户的账户，则须作为 root 用户运行以下命令行：

```
faillock --user <username> --reset
```

当使用 **authconfig** 功能对验证配置参数进行修改时， **authconfig** 功能的设置参数会覆盖 **system-auth** 文件和 **password-auth** 文件。要同时使用配置文件和**authconfig** ，您必须使用以下步骤对账户锁定进行参数配置：

1. 创建以下符号链接：

```
~]# ln -s /etc/pam.d/system-auth /etc/pam.d/system-auth-local
~]# ln -s /etc/pam.d/password-auth /etc/pam.d/password-auth-local
```

2. **/etc/pam.d/system-auth-local** 文件应含有以下命令行：

```
auth        required      pam_faillock.so preauth silent audit
deny=3 unlock_time=600 include system-auth-ac
auth        [default=die]  pam_faillock.so authfail silent audit
deny=3 unlock_time=600

account     required      pam_faillock.so
account     include       system-auth-ac

password    include       system-auth-ac

session     include       system-auth-ac
```

3. **/etc/pam.d/password-auth-local** 文件应含有以下命令行：

```
auth        required      pam_faillock.so preauth silent audit
deny=3 unlock_time=600 include password-auth-ac
auth        [default=die]  pam_faillock.so authfail silent audit
deny=3 unlock_time=600

account     required      pam_faillock.so
account     include       password-auth-ac

password    include       system-auth-ac

session     include       system-auth-ac
```

关于 **pam_faillock** 不同配置选项的更多信息，请参阅 **pam_faillock(8)** 手册页。

4.1.4. 会话锁定

在每天的操作中，用户可能会因一些原因需离开他们的工作站，使得工作站无人值守。这可能会让攻击者有物理访问机器的机会，尤其在物理安全措施不完备的情况下（参阅第 1.2.1 节 “物理控制”）。这个问题在笔记本电脑中尤为突出，因为它们的便携性影响了其物理安全。您可以通过利用会话锁定来减少这些风险。会话锁定的特征就是除非输入了正确的密码，否则禁止访问系统。

备注

锁定屏幕，而不是进行注销，这一做法的主要优势是允许用户进程（例如文件传输）持续进行。而注销则会停止这些进程。

4.1.4.1. 使用 vlock 锁定虚拟控制台

用户可能也需要锁定虚拟控制台。这可以通过使用一个名为 **vlock** 实用程序来实现。要安装这个实用程序，则须作为 root 用户执行以下命令：

```
~]# yum install vlock
```

安装之后，可以通过使用 **vlock** 命令，无需其他任何参数，对任何控制台会话进行锁定。这能够在锁定当前活动的虚拟控制台会话的同时，仍允许访问其他虚拟控制台。要禁止访问工作站所有的虚拟控制台，则须执行以下命令：

```
vlock -a
```

在本例中， **vlock** 锁定了当前活动的控制台，而 **-a** 选项则是防止切换到其他虚拟控制台。

其他信息请参阅 **vlock(1)** 手册页。

重要

那些与 **vlock** 版本有关的问题仍存在于当前的 Red Hat Enterprise Linux 7 。

- 这个程序目前不允许通过使用 root 密码对控制台进行解锁。其他信息可见 BZ#895066。
- 锁定控制台并不能清除滚动控制台屏幕缓冲区，但允许任何人物理访问工作台，查看原先在控制台上发出的命令和任何所显示的输出内容。更多信息请参阅 BZ#807369。

4.2. 控制 root 访问

当管理家庭机器时，该用户必须作为 root 用户或者使用 *setuid* 程序获得有效 root 特权，比如 **sudo** 或者 **su** 执行一些任务。setuid 程序是使用程序拥有者的用户 ID（*UID*）进行操作，而不是用户操作该程序。这样的程序可在详细列表的拥有者部分的 **s** 表示，如以下示例所示：

```
~]$ ls -l /bin/su
    -rwsr-xr-x. 1 root root 34904 Mar 10  2011 /bin/su
```

注意

s 可以是大写也可以是小写。如果是大写，则意味着还没有设定基本权限。

然而，对于机构的系统管理员而言，必须决定此机构的用户应有多大的管理访问权限访问机器。通常仅为 root 用户所能进行的一些操作，如重启和安装可移动媒体，通过一个名为 **pam_console.so** 的 PAM 模块，可允许首位登录物理控制台的用户进行操作。但是，其他重要的系统管理任务，如网络参数设置变更、新鼠标的参数配置、或网络设备的安装，这些都须有管理权限才能进行操作。因此，系统管理员必须决定用户应有多大的权限访问网络。

4.2.1. 不允许 root 访问

如果管理员因为总总理由认为允许用户作为 root 登录不妥，则不应当泄露 root 密码，且不允许通过引导装载程序密码保护进入运行级别 1 或单用户模式（有关此话题的更多信息，请参阅 <第 4.2.5 节 “引导装载程序的保护” >）。

以下有四种不同的方式能让管理员可进一步确保禁止 root 登录：

变更 root shell

要防止用户作为 root 用户直接登录，系统管理员可将 root 账户的 shell 参数设置到 **/etc/passwd** 文件下的 **/sbin/nologin**。

表 4.2. 禁用 root shell

效果	不影响
禁止访问 root shell 或将任何此类尝试载入日志。禁止以下程序访问 root 账户： **login** **gdm** **kdm** **xdm** **su** **ssh** **scp** **sftp**	有些程序无需 shell ，如文件传输协议（ FTP ，File Transfer Protocol）客户端、邮件客户端和很多 setuid 程序。 *不* 禁止以下程序访问 root 账户： **sudo** FTP clients Email clients

禁止通过任何控制台设备 (tty) 进行 root 访问

要进一步限制访问 root 账户，管理员可以通过编辑 **/etc/securetty** 文件在控制台禁止 root 登录。此文件列出了 root 用户允许登录的所有设备。如果此文件不存在，则 root 用户可以通过系统上任何通信设备进行登录，无论是通过控制台还是通过原始网络接口。这十分危险，因为用户可以作为 root 用户通过 Telnet 登录他们的机器，也就是通过网络在纯文本中进行密码传输。

在默认情况下，Red Hat Enterprise Linux 7 的 **/etc/securetty** 文件只允许 root 用户登录物理连接到机器的控制台。要防止 root 用户登录，则须作为 root 用户用 shell 提示符打出以下命令，删除此文件的内容：

```
echo > /etc/securetty
```

要使 **securetty** 能够支持 KDM、 GDM 和 XDM 登录管理器，则须添加以下命令行：

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad]
pam_securetty.so
```

添加到以下列出的文件中：

- **/etc/pam.d/gdm**
- **/etc/pam.d/gdm-autologin**
- **/etc/pam.d/gdm-fingerprint**
- **/etc/pam.d/gdm-password**
- **/etc/pam.d/gdm-smartcard**
- **/etc/pam.d/kdm**
- **/etc/pam.d/kdm-np**
- **/etc/pam.d/xdm**

警告

空白的 **/etc/securetty** 文件“ *不*”能防止 root 用户远程使用 OpenSSH 工具套件登录，因为在认证前无法打开该控制台。

表 4.3. 禁用 root 登录

效果	不影响
通过控制台或者网络防止对 root 账户的访问。防止以下程序访问 root 账户： • **login** • **gdm** • **kdm** • **xdm** • 可打开 tty 的其他网络服务	有些程序无需作为 root 用户登录，但可通过 setuid 或其他途径完成管理任务。*不* 允许以下程序访问 root 账户： • **su** • **sudo** • **ssh** • **scp** • **sftp**

禁止 root SSH 登录

要防止 root 通过 SSH 协议登录，则须编辑 SSH 守护进程的配置文件 **/etc/ssh/sshd_config**，且变更以下命令行：

```
#PermitRootLogin yes
```

将其改为：

```
PermitRootLogin no
```

表 4.4. 禁止 Root SSH 登录

效果	不影响
通过工具的 OpenSSH 套件防止 root 访问。防止以下程序访问 root 账户： • **ssh** • **scp** • **sftp**	有些程序并非 OpenSSH 工具套件的一部分。

使用 PAM 限制 root 访问服务

通过 **/lib/security/pam_listfile.so** 模块，PAM 在拒绝特定账户方面提供了极大的灵活性。管理员可用此模块来引用一份不允许登录的用户名单。要限制 root 访问系统服务，则须编辑在 **/etc/pam.d/** 目录下的目标服务文件，且确保身份验证是需要使用 **pam_listfile.so** 模块。

The following is an example of how the module is used for the **vsftpd** FTP server in the **/etc/pam.d/vsftpd** PAM configuration file (the \ character at the end of the first line is *not* necessary if the directive is on a single line):

```
auth    required    /lib/security/pam_listfile.so    item=user \n
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

这样可让 PAM 参考 **/etc/vsftpd.ftpusers** 文件，并让所有列出的用户拒绝访问该服务。管理员可更改这个文件的名称，且可为每个服务保存独立的列表，或者使用一中央列表拒绝访问多个服务。

如果管理员想要拒绝访问多个服务，可在 PAM 配置文件中添加类似的行，比如 **/etc/pam.d/pop** 和 **/etc/pam.d/imap** 中为电子邮件客户端添加，在 **/etc/pam.d/ssh** 中为 SSH 客户端添加。

关于 PAM 的更多信息，请参阅 **/usr/share/doc/pam-<version>/html/** 目录下的《*Linux-PAM 系统管理员指南*》。

表 4.5. 使用 PAM 来禁用 root

效果	不影响
PAM 检测软件能够禁止 root 访问网络服务。禁止以下服务访问 root 账户： • **login** • **gdm** • **kdm** • **xdm** • **ssh** • **scp** • **sftp** • FTP clients • Email clients • 任何 PAM 检测服务	PAM 无法识别的程序和服务。

4.2.2. 允许 root 访问

如果机构中的用户是可信且具有计算机知识，那么允许他们有 root 访问就不是什么问题。根据用户允许 root 访问意味着个人用户可处理一些次要活动，比如添加设备或者配置网络接口，那么可让系统管理员处理网络安全和其它重要问题。

另一方面，个人用户有 root 访问可导致以下问题：

- *机器错误配置* — 具有 root 访问的用户可错误配置其机器，并需要帮助方可解决问题。更有甚者他们可能在不知情的情况下开启安全漏洞。
- *运行不安全的服务* — 有 root 访问的用户可能会在其机器中运行不安全的服务，比如 FTP 或者 Telnet，并可能让用户名和密码处于危险。这些服务可通过网络以纯文本传送这个信息。
- *作为 root 运行电子邮件附件* — 对 Linux 有影响的病毒虽然少见，但确实存在。但只有在作为 root 用户运行它们时才有威胁。
- *保持审计线索完整* — 因为 root 账户经常为多个用户所共享，如此一来就有多个系统管理员可以维持系统，所以就无法弄清在一固定时间内究竟是哪个用户是 root 用户。使用单独登录时，用户所登录的账户，以及用来表示会话跟踪目的的唯一值将被放入任务结构，而这是该用户启动的每一个程序的父类别。当使用并发登录时，唯一值就可以用于特定登录的跟踪行为。当一个行为引起了审计事件，那么就记录下登录账户以及与唯一值关联的会话。使用 **aulast** 命令可查看这些登录和会话。**aulast** 命令中的 **--proof** 选项可用于表示一个特定的 **ausearch** 查询，以便隔离由一个特定会话产生的可审计事件。关于审计系统的更多信息，请参阅 < 第 5 章 *系统审核* > 。

4.2.3. 限制 root 访问

管理员也许是希望允许只通过 setuid 程序进行访问，而不是完全拒绝访问 root 用户，例如 **su** 或 **sudo**。关于 **su** 和 **sudo** 的更多信息，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》和 **su(1)** 与 **sudo(8)** 的手册页。

4.2.4. 允许自动注销用户登录

当用户作为 **root** 登录时，无人看管的登录会话可能会造成重大的安全风险。要降低这种风险，您可以配置系统来实现在一段时间后自动注销空闲用户：

1. 请确保 *screen* 工具包已安装。您可以作为 **root** 通过运行以下命令来实现：

```
yum install screen
```

关于如何在 Red Hat Enterprise Linux 7 安装工具包的更多信息，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》。

2. 作为 **root**，在 **/etc/profile** 文件的开头添加以下命令行来确保此文件的进程不被中断：

```
trap "" 1 2 3 15
```

3. 在 **/etc/profile** 文件的结尾添加以下命令行，以实现用户每次登录虚拟控制点或远程控制台就启动 **screen** 会话：

```
SCREENEXEC="screen"
if [ -w $(tty) ]; then
trap "exec $SCREENEXEC" 1 2 3 15
echo -n 'Starting session in 10 seconds'
sleep 10
exec $SCREENEXEC
fi
```

请注意，每当一个新的会话启动时，就会显示一条信息，用户则必须等待十秒。要调整启动会话前的等待时间，则须在 **sleep** 命令后改变数值：

4. 在 **/etc/screenrc** 配置文件中添加以下命令行，来实现在不活动周期后关闭 **screen** 会话：

```
idle 120 quit autodetach off
```

这将设置时间的限制为 120 秒。要调整这个限制时间，则须在 **idle** 指令后改变数值：

或者您可以通过使用以下命令行来配置系统，以实现仅锁定会话：

```
idle 120 lockscreen autodetach off
```

这种方式将要求使用密码来解锁会话。

此变更将在下一次用户登录系统时生效。

4.2.5. 引导装载程序的保护

使用密码保护 Linux 引导装载程序的主要原因如下：

1. *防止进入单用户模式* — 如果攻击者可将系统引导至单用户模式，他们就可以自动成为 root ，而不会被提示其输入 root 密码。

警告

不建议通过编辑 **/etc/sysconfig/init** 文件下的 ***SINGLE*** 参数，来实现禁止用密码访问单用户模式。 攻击者可以通过在 GRUB 2 的 kernel 命令行指定一个自定义的初始命令（使用 ***init=*** 参数）来跳过密码。如《Red Hat Enterprise Linux 7 系统管理员指南》所介绍，推荐使用密码保护 GRUB 2 引导装载程序。

2. *禁止访问 GRUB 2 控制台* — 如果机器使用 GRUB 2 作为其引导装载程序，攻击者可使用 GRUB 2 编辑器界面来改变其配置，或使用 **cat** 命令来收集信息。

3. *禁止访问不安全的操作系统* — 如果是双重引导系统，攻击者可以在启动时选择操作系统，例如 DOS，这就可忽略访问控制和文件权限。

Red Hat Enterprise Linux 7 在 Intel 64 和 AMD 64 平台上使用 GRUB 2 引导装载程序。关于 GRUB 2 的详细资料，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》。

4.2.5.1. 不允许交互式启动

在启动顺序的开头按 **I** 键，可允许您交互式启动系统。在交互式启动中，系统会提示您逐一启动每项服务。然而，这可能会导致那些通过物理访问您系统的攻击者禁用安全相关的服务，以及获取访问系统的权限。

要防止用户交互式启动系统，则须作为 root 用户禁用 **/etc/sysconfig/init** 文件下的 ***PROMPT*** 参数：

```
PROMPT=no
```

4.3. 安全服务

虽然用户访问管理控制对机构管理员来说是个重要问题，但监控哪些网络处于活跃状态对任何一位管理员以及 Linux 系统操作者来说都更为重要。

Red Hat Enterprise Linux 7中的很多服务都类似网络服务器。如果在一个机器上运行网络服务，那么服务器应用程序（亦称为 *daemon*），就会侦听一个或者多个网络端口的连接。这些服务器被视为潜在的攻击手段。

4.3.1. 服务的风险

网络服务可为 Linux 系统造成很多危险。以下是一些主要问题列表：

- *拒绝服务攻击（DoS）* — 通过向服务发出大量请求，拒绝服务攻击可让系统无法使用，因为它会尝试记录并回应每个请求。
- *分布的拒绝服务攻击（DDoS）* — 一种 DoS 攻击类型，可使用多台被入侵的机器（经常是几千台或者更多）对某个服务执行联合攻击，向其发送海量请求并使其无法使用。
- *脚本漏洞攻击* — 如果某台服务器使用脚本执行服务器端动作，网页服务器通常这样做，那么破解者就可以攻击那些没有正确编写的脚本。这些脚本漏洞攻击导致缓存溢出，或者允许攻击者更改系统中的文件。
- *缓存溢出攻击* — 连接到特权端口为1023 的服务器必须作为管理用户来运行。如果应用程序有可利用的缓存溢出，那么攻击者就可作为运行该应用程序的用户访问系统。因为有可利用的缓存溢出存在，破解者可使用自动工具来识别有漏洞的系统，并在获得访问后，使用自动工具套件保持其对该系统的访问。

注意

在 Red Hat Enterprise Linux 7 中可使用 *ExecShield* 缓和缓冲溢出漏洞的威胁，这是可执行内存片段和保护技术，由 x86 兼容的唯一或者多处理器内核支持。ExecShield 可通过将虚拟内存分成可执行片段以及不可执行片段降低缓冲溢出的风险。所有尝试执行可执行片段之外程序代码（比如缓存溢出漏洞注入的恶意代码）可触发片段失败并终止。

Execshield 还支持 *禁止执行* (NX) 在 AMD64 平台的技术 *eXecute Disable* (XD) 在 Itanium 上的技术以及 Intel® 64 系统。这些技术与 Execshield 合作可防止恶意代码在有 4KB 可执行代码单位的虚拟内存的可执行部分运行，降低来自隐藏缓存溢出漏洞的攻击风险。

重要

要限制通过网络进行攻击，应该将所有不使用的服务关闭。

4.3.2. 识别并配置服务

要提高安全性，默认关闭在 Red Hat Enterprise Linux 7 中安装的大多数服务。但有些是例外：

- **cups** — Red Hat Enterprise Linux 7 的默认打印服务器。
- **cups-lpd** — 备用打印服务器。
- **xinetd** — 控制与一系列下级服务器连接的超级服务器，比如 **gssftp** 和 **telnet** 超级服务器。
- **sshd** — OpenSSH 服务器，是 Telnet 的安全替代产品。

在决定是否要让服务保持运行时，最好根据常识，并避免冒任何风险。例如：如果无法使用打印机时，那就不要让 **cups** 继续运行。同样也适用于 **portreserv**。如果您没有挂载 NFSv3 卷或者使用 NIS（**ypbind** 服务），则应该禁用 **rpcbind**。检查哪些可用的网络服务可以在开机时启动是不够的。我们推荐还应该检查哪些端口已打开并在侦听。详情请参阅 <第 4.4.2 节 “验证使用侦听的端口” > 。

4.3.3. 不安全的服务

无疑，任何网络服务都是不安全的。这就是为什么要关闭不使用的服务是如此的重要。我们会常规发现并修补服务漏洞，这些工作对常规更新与网络服务有关的软件包非常重要。详情请参阅 <第 3 章 *及时更新系统* > 。

某些网络协议本身就比其它协议更不安全。这些协议包含一些服务：

- *以不加密的方式在网络中传输用户名和密码* — 很多老的协议，比如 Telnet 和 FTP，它们对认证会话都不加密，应尽量避免使用。
- *以不加密方式传输敏感数据* — 很多协议在网络间传输数据时不加密。这些协议包括 Telent、FTP、HTTP 和 SMTP。很多网络文件系统，比如 NFS 和 SMB 也以不加密的方式在网络间传输信息。用户在使用这些协议时有责任限制要传输的数据类型。

本身就不安全的服务示例包括 **rlogin**、**rsh**、 **telnet**、以及**vsftpd**。

所有远程登录和 shell 程序 (**rlogin**、**rsh**、 以及 **telnet**) 应避免使用以支持 SSH。详情请参阅〈 第 4.3.10 节 “保障 SSH” 〉有关 **sshd**。

FTP 并不象远程 shell 那样天生对系统安全有威胁，但需要小心配置并监控 FTP 服务器以免出问题。有关保证 FTP 服务器安全的详情请参阅〈第 4.3.8 节 “保证 FTP 安全”〉。

应小心使用并在防火墙后使用的服务包括：

- **auth**
- **nfs-server**
- **smb** 以及 **nbm** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

有关保证网络服务安全的更多信息，请参阅〈第 4.4 节 “安全访问网络”〉。

4.3.4. 保障 rpcbind

rpcbind 服务是为 NIS 和 NFS 等 RPC 服务进行动态端口分配的守护进程。它的认证机制比较薄弱，并可以为其控制的服务分配大范围的端口。因此很难保证其安全。

备注

因为 NFSv4 不再需要 rpcbind ，所以保障 **rpcbind** 安全只影响 NFSv2 和 NFSv3 的执行。如果您要运行 NFSv2 或者 NFSv3 服务器，就需要**rpcbind** ，且在以下章节会运用到 rpcbind 。

如果运行 RPC 服务，请遵守以下基本规则。

4.3.4.1. 使用 TCP Wrapper 保护 rpcbind

因为 TCP Wrapper 没有内嵌的认证形式，所以使用 TCP Wrapper 限制哪些网络或者主机可以访问 **rpcbind** 服务很重要。

另外，限制对服务的访问时，“*只*”使用 IP 地址。由于通过使 DNS 中毒和其它方法可以伪造主机名，所以请避免使用主机名。

4.3.4.2. 使用防火墙保护 rpcbind

要进一步限制访问 **rpcbind** 服务，最好是为该服务器添加 **firewalld** 规则，并限制对具体网络的访问。

以下是 **firewalld** Rich Text 命令的两个示例。第一个是实现从网络 192.168.0.0/24 到 111 端口（**rpcbind** 服务使用的端口）的 TCP 连接的示例。第二个是实现从本地主机到同一端口的 TCP 连接的示例。丢弃所有其它数据包。

```
~]# firewall-cmd --add-rich-rule='rule family="ipv4" port port="111"
protocol="tcp" source address="192.168.0.0/24" invert="True" drop'
~]# firewall-cmd --add-rich-rule='rule family="ipv4" port port="111"
protocol="tcp" source address="127.0.0.1" accept'
```

同样地，要限制 UDP 流量，则须使用以下命令：

```
~]# firewall-cmd --add-rich-rule='rule family="ipv4" port port="111"
protocol="udp" source address="192.168.0.0/24" invert="True" drop'
```

备注

将 **--permanent** 添加到 **firewalld** Rich Text 命令中，以实现永久设置。有关执行防火墙的更多信息，请参阅〈 第 4.5 节 “使用防火墙” 〉。

4.3.5. 保证 NIS 安全

“*网络信息服务*” (NIS) 是一个 RPC 服务，亦称之为 **ypserv** ， 可与 **rpcbind** 及其它相关服务一同使用，向自称在其域中的所有计算机发布用户名、密码以及其它敏感信息映射。

NIS 服务器由许多应用程序组成。它包括以下的应用程序：

- **/usr/sbin/rpc.yppasswdd** — 也称为 **yppasswdd** 服务，这个守护进程允许用户更改其 NIS 密码。
- **/usr/sbin/rpc.ypxfrd** — 也称为 **ypxfrd** 服务，这个守护进程负责通过网络的 NIS 映射传输。
- **/usr/sbin/ypserv** — 这是 NIS 服务器守护进程。

就当今的标准而言， NIS 在某种程度上并不安全。它没有主机认证机制，且所有通过网络的传输都是不加密的，包括哈希密码。因此设置使用 NIS 的网络时，要特别小心。事实上， NIS 的默认配置本身就不安全，这也让情况变得更为复杂。

建议任何想要运行 NIS 服务器的人先要保障 **rpcbind** 服务的安全，正如在〈第 4.3.4 节 “保障 rpcbind”〉中概括的那样，然后解决以下的问题，比如网络计划。

4.3.5.1. 谨慎规划网络

由于 NIS 通过网络传输敏感信息时未经加密，所以在防火墙后，且在隔离和安全的网络中运行就非常重要。使用不安全的网络传输 NIS 信息，无论何时都有被截获的风险。谨慎规划网络有助于防止严重的安全漏洞。

4.3.5.2. 使用类似密码的 NIS 域名和主机名

只要用户知道 NIS 服务器的 DNS 主机名和 NIS 域名，那么在 NIS 域中的任何计算机都可以在未经认证的情况下使用命令从服务器中提取信息。

例如：如果有人是从笔记本电脑连接到网络或者从外部侵入（并要嗅探内部 IP 地址），那么以下命令就可揭示 **/etc/passwd** 映射：

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

如果这个攻击者是 root 用户，那么他们就可通过以下命令获取 **/etc/shadow** 文件：

```
ypcat -d <NIS_domain> -h <DNS_hostname> shadow
```

备注

如果使用 Kerberos ，那么 **/etc/shadow** 文件就不会储存在 NIS 映射中。

要让攻击者更难访问 NIS 映射，则须让 DNS 主机名生成一个随机字符串 ，比如 **o7hfawtgmhwg.domain.com**。同样地，也可创建一个“*不同的*”随机 NIS 域名。这就让攻击者访问该 NIS 服务器变得更加困难。

4.3.5.3. 编辑 /var/yp/securenets 文件

如果 **/var/yp/securenets** 文件是空白文件，或是根本不存在（默认安装后就是这种情况），那么 NIS 就会侦听所有网络。首先要做的就是在该文件中添加子网掩码/网络对，这样一来 **ypserv** 只会响应来自对应网络的请求。

以下是 **/var/yp/securenets** 文件的条目示例：

```
255.255.255.0     192.168.0.0
```

警告

首次启动 NIS 服务器时，一定要有已生成的 **/var/yp/securenets** 文件。

这个技术并不提供对 IP 嗅探式攻击的保护，但至少可以限制 NIS 服务器提供服务的网络。

4.3.5.4. 分配静态端口并使用 Rich Text 规则

所有与 NIS 关联的服务器都可以分配到指定的端口，**rpc.yppasswdd** 除外 — 该守护进程允许用户更改其登录密码。其它两个 NIS 服务器守护进程 **rpc.ypxfrd** 和 **ypserv** 分配端口，这就可允许创建防火墙规则，以便进一步防止入侵者破坏 NIS 服务器守护进程。

要做到这一点，怎行在 **/etc/sysconfig/network** 中添加以下命令行：

```
YPSERV_ARGS="-p 834"
YPXFRD_ARGS="-p 835"
```

以下 rich text **firewalld** 规则可用于强制设定服务器用这些端口进行侦听的网络：

```
~]# firewall-cmd --add-rich-rule='rule family="ipv4" source
address="192.168.0.0/24" invert="True" port port="834-835"
protocol="tcp" drop'
~]# firewall-cmd --add-rich-rule='rule family="ipv4" source
address="192.168.0.0/24" invert="True" port port="834-835"
protocol="udp" drop'
```

这就是说，如果请求来自 **192.168.0.0/24** 网络，那么服务器就只可连接到 834 和 835 端口。第一规则用于 **TCP** ，第二规则用于 **UDP**。

备注

有关用 iptables 命令运行防火墙的更多信息，情参阅〈 第 4.5 节 “使用防火墙”〉。

4.3.5.5. 使用 Kerberos 认证

NIS 用于认证操作时，其中要考虑的问题是，无论用户何时登录机器，**/etc/shadow** 映射上的哈希密码都是通过网络进行传送。如果入侵者可以访问 NIS 域或者探查网络流量，那么他们就可以收集用户名以及哈希密码。在拥有充足时间的情况下，密码破译程序可以猜对较弱的密码，那么攻击者就可以访问网络上的有效账户。

因为 Kerberos 使用密钥加密，那么就不用通过网络发送哈希密码，所以系统就更加安全。关于 Kerberos 的更多信息，请参阅〈 Linux域身份，认证，策略指导 〉。

4.3.6. 保证 NFS 安全

重要

NFS 流量可通过使用不同版本的 TCP 进行传送，但它应在 NFSv3 下使用，而不是 UDP ；在使用 NFSv4 时，NFS 流量是必要的。所有版本的 NFS 都支持 Kerberos 用户和分组认证，作为 **RPCSEC_GSS** 内核模块的一部分。因为 Red Hat Enterprise Linux 7 支持 NFSv3 使用 **rpcbind**，所以有关 **rpcbind** 信息也包括在内。

4.3.6.1. 谨慎规划网络

NFSv2 和 NFSv3 传统上来说，不能安全地传输数据。现在所有版本的 NFS 都有能力对使用Kerberos 的普通文件系统进行认证（且进行选择性加密）。 在 NFSv4 下，可以使用Kerberos；在 V2 或 V3 下，锁定文件和挂载文件仍无法使用 Kerberos 。当使用 NFSv4 时，如果客户处于 NAT 或者防火墙的保护下，那么可能会关闭授权。关于如何使用 NFSv4.1 通过 NAT 和防火墙来运行授权的具体信息，请参阅《红帽企业版Linux7储存管理手册》。

4.3.6.2. 保障 NFS 挂载选项

《红帽企业版 Linux 7 储存管理手册》详细解释了 **/etc/fstab** 文件中 **mount** 命令的使用。从安全管理的角度来说，值得注意的是， **/etc/nfsmount.conf** 也详细讲解了 NFS 挂载选项，这可用于设定客户默认选项。

4.3.6.2.1. 审查 NFS 服务器

警告

只能导出整个文件系统。导出文件系统的子目录成为一个安全问题。因为某些情况，客户可能会“跳出”文件系统的导出的子目录，获取文件系统中未导出的目录（请参阅 **exports(5)** 手册页中的子树检查）。

使用 **ro** 选项可使文件系统导出的属性为“只读取”，这在任何时候都会减少可对挂载文件系统进行写入操作的用户数量。只有在明确要求的情况下，才能使用**rw** 选项。更多信息，请参阅 **exports(5)** 的手册页。例如，允许写入访问，则会加大符号链接攻击的风险。这包括临时目录，如**/tmp** 和 **/usr/tmp**。

用 **rw** 选项挂载目录时，要避免全域可写，这在任何时候都可降低风险。就像某些应用程序以明文储存密码或是储存加密强度较弱的密码，导出主目录也被视为有风险的操作。审查和改进应用代码可以减少这种风险。一些用户没有在他们的 SSH 密钥上设定密码，因此这也意味着主目录存在风险。强制使用密码或者使用 Kerberos 可以减少风险。

限定只有需要访问权限的客户才能导出目录。 在 NFS 服务器上使用 **showmount -e** 命令来审查该服务器导出的内容。请勿导出没有明确需求的任何内容。

请勿使用 **no_root_squash** 选项，并且审查现有的安装程序，以确保并未使用该选项。更多信息，请参阅〈第 4.3.6.4 节 “请勿使用 no_root_squash 选项”〉。

secure 选项是服务器端导出选项，用于限定只能从 “保留” 端口进行导出。默认情况下，服务器只允许客户通过 “保留” 端口（端口编号不超过 1024）进行通讯，因为传统上来说，客户只允许通过““可信”” 代码（例如内核 NFS 客户）来使用这些端口。然而，因为在许多网络上，任何人要成为某些客户端的 root 并不难。因此，假定保留端口所进行的通讯拥有特权，对于服务器而言，通常都是不安全的。因此，限制保留端口具有有限的价值；最好还是依靠 kerneros ，防火墙，以及限定只有特定客户才能进行导出。

如果可能的话，大多数的客户仍使用保留端口。然而，保留端口是有限的资源，因此客户（尤其是那些拥有大量 NFS 装载的客户）可以选择编号更高的端口。Linux 客户可以通过使用 “noresvport” 挂载选项来完成。如果您希望在导出目录中允许此运作，那么您可以通过 “insecure” 导出选项来完成。

禁止用户登录服务器是一个很好的做法。在审查 NFS 服务器的上述设置时，也审查能访问和进入服务器的人和内容。

4.3.6.2.2. 审查 NFS 客户

使用 **nosuid** 选项来禁止使用 **setuid** 程序。**nosuid** 选项可禁用 **set-user-identifier** 或 **set-group-identifier** 位。这可阻止远程用户通过运行 setuid 程序获取更高的特权。在客户端和服务器端使用该选项。

noexec 选项可禁止客户端上的所有可执行文件。使用此选项可防止用户无意中执行了文件系统中所共享的的文件。对于大多数的（即使不是全部的）文件系统而言，**nosuid** 和 **noexec** 选项都是标准选项。

使用 **nodev** 选项可防止客户端将 “device-files” 作为硬件设备进行处理。

resvport 选项是客户端挂载选项，**secure** 是相应的服务器端导出选项（请参阅上述说明）。它限定只有使用“保留端口”才能进行通讯。保留端口或是“知名”端口会保留给特权用户或程序，比如 root 用户。设置这个选项会促使客户使用保留的源端口与服务器进行通讯。

现在，所有版本的 NFS 都支持挂载 Kerberos 认证。启用这个挂载选项： **sec=krb5**。

NFSv4 支持用 Kerberos 进行挂载，通过使用 **krb5i** 来确保完整性，使用 **krb5p** 来确保隐私保护。在使用 **sec=krb5** 进行挂载时，上述这些都会使用到，但需要在 NFS 服务器上配置。有关导出目录（**man 5 exports**）的更多信息，请参阅手册页。

NFS 手册页（**man 5 nfs**）中，“安全注意事项” 部分解释了在 NFSv4 中增强安全的问题，以及包含了所有 NFS 详细的挂载信息。

4.3.6.3. 注意语法错误

NFS 服务器通过查阅 **/etc/exports** 文件，决定导出哪些文件系统以及将这些目录导出到哪些主机中。编辑此文件时，请小心，不要添加多余的空格。

例如，**/etc/exports** 文件中的以下命令行可实现与主机 **bob.example.com** 共享 **/tmp/nfs/** 目录的读／写权限。

```
/tmp/nfs/     bob.example.com(rw)
```

另一方面，由于主机名的一个空格，这使 **/etc/exports** 中的以下命令行可实现与主机 **bob.example.com** 共享同一目录的只读权限，同时实现与“*所有人*”共享它的读／写权限。

```
/tmp/nfs/     bob.example.com (rw)
```

最好使用 **showmount** 命令检查所有已配置的 NFS 共享，以确定共享的内容：

```
showmount -e <hostname>
```

4.3.6.4. 请勿使用 no_root_squash 选项

默认情况下，NFS 共享会将 root 用户更改为一个非特权用户帐户，即 **nfsnobody** 用户。这会将所有 root 创建的文件的所有者更改为 **nfsnobody**，这可防止用 setuid 位组来设置程序的上传。

如果使用 **no_root_squash**，那么远程 root 用户就可以更改共享文件系统中的任何文件，并留下感染木马的应用程序给其它用户去执行。

4.3.6.5. NFS 防火墙配置

NFSv4 是红帽企业版 Linux 7 默认的 NFS 版本，且它要求只对TCP 开放 2049 端口。如果使用 NFSv3 ，那么就需要四个额外的端口，如下述说明。

为NFSv3配置端口

NFS 使用的端口是由 rpcbind 进行动态分配，在创建防火墙规则时，可能会造成问题。要简化这个步骤，则须使用 */etc/sysconfig/nfs* 文件指定要使用的端口：

- **MOUNTD_PORT** — 用于挂载的 TCP 和 UDP 端口（rpc.mountd）
- **STATD_PORT** — 用于显示 TCP 和 UDP 状态的端口（rpc.statd）
- **LOCKD_TCPPORT** — 用于 nlockmgr 的 TCP 端口（rpc.lockd）
- **LOCKD_UDPPORT** — 用于 nlockmgr 的 UDP 端口（rpc.lockd）

指定的端口号绝对不能用于其它服务。对您的防火墙进行配置，可指定端口号以及 TCP 和 UDP 的 2049 端口（NFS）。

在 NFS 服务器上运行 **rpcinfo -p** 命令，可查看所使用的端口和 RPC 程序。

4.3.7. 保证 Apache HTTP 服务器安全

Apache HTTP 服务器是 Red Hat Enterprise Linux 7 中最稳定、最安全的服务之一。有很多可用的选项和技术可用于保证 Apache HTTP 服务器安全 — 由于数量过多，在此就不进行深入探讨。以下小节简要介绍了在运行 Apache HTTP 服务器时可采用的操作。

在投入生产“*之前*”，一定要核实所有脚本都可如预期在系统中运行。另外，请确保只有 root 用户才有权限写入含脚本或者 CGI 的任何目录。要做到这一点，则须作为 root 用户运行以下命令：

```
chown root <directory_name>
```

```
chmod 755 <directory_name>
```

系统管理员应谨慎使用以下配置选项（在 **/etc/httpd/conf/httpd.conf** 进行配置）：

FollowSymLinks

此指令为默认启用，因此在创建符号链接到网页服务器的文档 root 目录时，请慎重行事。例如，请勿为“**/**”提供符号链接。

Indexes

虽然此指令为默认启用，但并非必要。要防止访问者浏览在服务器上的文件，则须删除这个指令。

UserDir

因为此指令可确认系统中用户帐户是否存在，所以要默认禁用 **UserDir** 指令。要在服务器上启用用户名目录浏览，则须使用以下指令：

```
UserDir enabled
        UserDir disabled root
```

这些指令用于 **/root/** 之外的所有用户目录，可激活其用户目录浏览这一功能。要在禁用帐户列表中添加用户，则须在 **UserDir disabled** 命令行添加以空格分隔的用户列表。

ServerTokens

ServerTokens 指令控制着服务器响应标题头信息，这信息会传送回给客户。它包括不同的信息，通过使用下列参数，可以对其进行自定义操作：

- **ServerTokens Full**（默认选项）— 提供所有可用信息（OS类型以及所使用的模块），例如：

```
Apache/2.0.41 (Unix) PHP/4.2.2 MyMod/1.2
```

- **ServerTokens Prod** 或者 **ServerTokens ProductOnly** — 提供以下信息：

```
Apache
```

- **ServerTokens Major** — 提供以下信息：

```
Apache/2
```

- **ServerTokens Minor** — 提供以下信息：

```
Apache/2.0
```

- **ServerTokens Min** 或者 **ServerTokens Minimal** — 提供以下信息：

```
Apache/2.0.41
```

- **ServerTokens OS** — 提供以下信息：

```
Apache/2.0.41 (Unix)
```

建议使用 **ServerTokens Prod** 选项，这样一来，潜在攻击者就无法获取关于您系统的任何有用信息。

重要

请勿删除 **IncludesNoExec** 指令。默认情况下，“*服务器端嵌入*”（SSI）模块无法执行命令。除非绝对必要，建议您不要更改这个设置，因为它可能会允许攻击者在系统中执行命令。

删除 httpd 模式

在某些情况下，最好删除特定的 **httpd** 模式，以限制 HTTP 服务器的功能。要实现这一目的，只须为整个命令行添加注释，该命令行用于加载在 **/etc/httpd/conf/httpd.conf** 文件中您想要删除的模块。例如，要删除代理模块，则须通过给下列命令行新增“ # ”字符，为下列命令行添加注释：

```
#LoadModule proxy_module modules/mod_proxy.so
```

请注意， **/etc/httpd/conf.d/** 目录包含了可用于加载模块的配置文件。

httpd 以及 SELinux

有关信息，请参阅《红帽企业版 Linux 7 SELinux 用户和管理员手册》。

4.3.8. 保证 FTP 安全

“*文件传输协议*”（FTP）是一个比较旧的 TCP 协议，用来通过网络传输文件。因为服务器所处理的所有传输，包括用户认证，都是未经加密，所以它被认为是一个不安全的协议，且应该谨慎地进行配置。

Red Hat Enterprise Linux 7 提供两个FTP 服务器：

- **Red Hat Content Accelerator（tux）** — 具有 FTP 功能的内核空间网页服务器。
- **vsftpd** — 重视安全的单机 FTP 服务执行工具。

下列安全指南可用于设置 **vsftpd** FTP 服务。

4.3.8.1. FTP 登录信息

提交用户名和密码前，所有用户都会看到登录信息。默认情况下，这个信息包含了版本信息，这对于尝试识别系统弱点的破解者十分有用。

要为 **vsftpd** 更改登录信息，则须在 **/etc/vsftpd/vsftpd.conf** 文件中添加以下指令：

```
ftpd_banner=<insert_greeting_here>
```

用登录信息文本替换上述指令中的 *<insert_greeting_here>*。

对于多行信息而言，最好使用信息文件。要简化多提示信息管理，则须将所有提示信息放入名为 **/etc/banners/** 的新目录。在本示例中，用于 FTP 连接的提示信息文件为 **/etc/banners/ftp.msg**。以下为此类文件的示例：

```
######### Hello, all activity on ftp.example.com is logged. #########
```

备注

正如〈第 4.4.1 节 “使用 TCP Wrappers 以及 xinetd 保证服务安全”〉所述，没有必要在文件的每一行中都使用 **220** 。

要在 **vsftpd** 中引用这个登录信息，则须在 **/etc/vsftpd/vsftpd.conf** 文件中添加以下指令：

```
banner_file=/etc/banners/ftp.msg
```

还可以发送附加信息提示给使用 TCP Wrapper 的连入连接，如〈第 4.4.1.1 节 “TCP Wrapper 和连接提示”〉所述。

4.3.8.2. 匿名访问

/var/ftp/ 目录的存在可激活匿名帐户。

创建这个目录的最简单的方法是安装 **vsftpd** 软件包。这个软件包可为匿名用户建立目录树，并为匿名用户配置目录的只读权限。

默认情况下，匿名用户不能写入任何目录。

警告

如果启用对 FTP 服务器的匿名访问，那么就要注意保存敏感数据的位置。

4.3.8.2.1. 匿名上传

要允许匿名用户上传文件，那么建议在 **/var/ftp/pub/** 中生成只写目录。要完成此操作，则须作为 root 用户运行以下命令：

```
~]# mkdir /var/ftp/pub/upload
```

下一步，更改权限以防止匿名用户查看该目录中的内容：

```
~]# chmod 730 /var/ftp/pub/upload
```

该目录的详细格式列表应如下所示：

```
~]# ls -ld /var/ftp/pub/upload
drwx-wx---. 2 root ftp 4096 Nov 14 22:57 /var/ftp/pub/upload
```

允许匿名用户在目录中读取和写入的管理员经常会发现他们的服务器成为盗窃软件的窝脏之处。

另外，在 **vsftpd** 下，在 **/etc/vsftpd/vsftpd.conf** 文件中添加以下行：

```
anon_upload_enable=YES
```

4.3.8.3. 用户帐户

因为 FTP 用不安全的网络传输未经加密的用户名和密码进行认证，所以最好拒绝系统用户从其用户帐户访问服务器。

要禁用 **vsftpd** 中的所有用户帐户，则须在 **/etc/vsftpd/vsftpd.conf** 中添加以下指令：

```
local_enable=NO
```

4.3.8.3.1. 限制用户帐户

要禁止 FTP 访问特殊账户或者特殊群组账户，例如 root 用户以及那些拥有 **sudo** 特权的用户，最简单的方法就是使用 PAM 列表文件，如〈第 4.2.1 节 “不允许 root 访问”〉所述。用于 **vsftpd** 的 PAM 配置文件是 **/etc/pam.d/vsftpd**。

还可以在每个服务中直接禁用用户帐户。

要在 **vsftpd** 中禁用特定帐户，则须在 **/etc/vsftpd/ftpusers** 中添加用户名。

4.3.8.4. 使用 TCP Wrapper 控制访问

使用 TCP Wrapper 控制对 FTP 守护进程的访问，如〈第 4.4.1 节 “使用 TCP Wrappers 以及 xinetd 保证服务安全”〉所述。

4.3.9. 保障 Postfix 的安全

Postfix 是邮件传输代理（MTA），它使用简单邮件传输协议（SMTP）在其它 MTA 和电子邮件客户端或者传递代理之间传递电子信息。虽然很多 MTA 都可以在彼此之间加密流量，但大多数并不这样做，因此使用任何公共网络发送电子邮件都被视为不安全的沟通形式。Postfix 替代 Sendmail 成为 Red Hat Enterprise Linux 7 默认的 MTA。

建议使用 Postfix 服务器的用户解决以下问题。

4.3.9.1. 限制拒绝服务攻击

因为电子邮件的本质，坚定的攻击者可以极其容易地使用邮件对服务器进行洪水攻击，导致拒绝服务。通过对 **/etc/postfix/main.cf** 文件中的指令进行限制设定，可以阻止有效的此类攻击。您可以更改已经存在的指令赋值，或是以下列格式，将所要的值添加到所需的指令中：

```
<directive> = <value>
```

以下一系列指令可用于限制拒绝服务攻击：

- **smtpd_client_connection_rate_limit** — 单位时间内，任何客户被允许与这个服务进行的最大连接尝试次数（如下所述）。如果默认值是0，这就意味着在单位时间内，客户可进行的连接次数与 Postfix 能接收的连接次数一样多。默认情况下，可排除在信任网络中的客户。
- **anvil_rate_time_unit** — 该单位时间可用于进行速率限制计算。默认值是 60 秒。
- **smtpd_client_event_limit_exceptions** — 从连接和速率限制命令中所排除的客户。默认情况下，也可排除在信任网络中的客户。
- **smtpd_client_message_rate_limit** — 单位时间内，客户被允许进行请求传递信息的最大次数（不管 Postfix是否真的接收这些信息）。
- **default_process_limit** — 提供特定服务的 Postfix 子进程默认的最大值。这种限制可能因为在 **master.cf** 文件中的特定服务而取消。默认情况下，赋值为 100。
- **queue_minfree** — 在队列文件系统中，接收邮件所需的最小可用空间（以字节为单位）。Postfix SMTP 服务器当前使用此指令来决定是否可以接收任何邮件。默认情况下，当可用空间的最小值小于 message_size_limit 的 1.5 倍时， Postfix SMTP 服务器则会拒绝 **MAIL FROM** 指令。要具体制定一个更高的可用空间最小值限定，则须具体制定一个 queue_minfree 值，其大小至少是 message_size_limit 的 1.5 倍。默认情况下，queue_minfree 值是 0。
- **header_size_limit** — 用于储存信息标题的最大内存（以字节为单位）。如果标题太大，那么超出的部分就会被舍弃。默认情况下，赋值为 102400。
- **message_size_limit** — 信息的最大值（以字节为单位），包括信封信息。默认情况下，赋值为 10240000。

4.3.9.2. NFS 以及 Postfix

请勿将邮件 spool 目录，**/var/spool/postfix/**，放到 NFS 共享卷上。因为 NFSv2 和 NFSv3 不会保持对用户 ID 和组群 ID 的控制，所以两个或者更多用户可以有相同的 UID，并接收和读取彼此的邮件。

注意

在 NFSv4 中使用 Kerberos，就不会出现这种情况。因为 **SECRPC_GSS** 内核模块不会根据 UID 进行认证。但是，最好还是“*不要*”将邮件池目录放到 NFS 共享卷中。

4.3.9.3. 只使用邮件的用户

要防止本地用户利用 Postfix 服务器上的漏洞，那么最好是让邮件用户只能使用电子邮件程序访问 Postfix 服务器。应该禁止邮件服务器上的 shell 帐户访问，并且**/etc/passwd** 文件中的所有 shell 用户都应设定到 **/sbin/nologin** 中（可能除了 root 用户之外）。

4.3.9.4. 禁用 Postfix 网络侦听

默认情况下，Postfix 被设定为只侦听本地回路地址。您可以通过查看 **/etc/postfix/main.cf** 文件来核实这一点。

查看 **/etc/postfix/main.cf** 文件，以确保只出现下列 ***inet_interfaces*** 命令行：

```
inet_interfaces = localhost
```

这确保 Postfix 只接收来自本地系统而非来自网络的邮件信息（比如定时任务报告）。这是默认设置，并且保护 Postfix 免受网络攻击。

inet_interfaces = all 设置可用于删除本地主机限制，并且允许Postfix 侦听所有接口。

4.3.10. 保障 SSH

Secure Shell （SSH）是一个强大的网络协议，可通过安全的渠道与其他系统进行通讯。通过 **SSH** 的传输都经过加密，可避免被拦截。关于 **SSH** 协议，以及在 Red Hat Enterprise Linux 7 中如何使用**SSH** 服务的常用信息，请参阅《红帽企业版 Linux 7 系统管理员指南》。

重要

此章节特别关注于保障 **SSH** 安全设置的最常用方法。这张列表中所推荐方法绝不可认为是详尽的，或是最权威的方法。关于可用于修改 **sshd** 守护进程的所有配置指令，请参阅 **sshd_config(5)** ；关于 **SSH** 基本概念的详细介绍，请参阅 **ssh(1)** 。

4.3.10.1. 加密登录

SSH 支持使用加密密钥登录电脑。这比只使用密码要更安全。如果您可以把这种方法与其他受到认证的方法相结合，那么这就被认为是多因素认证。有关如何使用多种认证方法的更多信息，请参阅〈第 4.3.10.2 节 “多种认证方法”〉。

为了启用加密密钥进行认证， 在 **/etc/ssh/sshd_config** 文件中的**PubkeyAuthentication** 配置指令需要设定为 **yes**。请注意，这是默认设置。把 **PasswordAuthentication** 指令设定为 **no** ，则会消除使用密码登录的可能性。

使用 **ssh-keygen** 命令可以生成**SSH** 密钥。如果在没有其它参数的情况下，调用 SSH 密钥，则会生成 2048 位RSA 密钥集。在默认情况下，密钥储存在 **~/.ssh** 目录中。您可以使用 **-b** 切换更改密钥强度。正常情况下，使用 2048 位密钥就足够了。《红帽企业版 Linux 7 系统管理手册》包含了有关生成密钥对的详细信息。

在**~/.ssh**目录中，您应该会看到两个密钥。当运行**ssh-keygen** 命令时，如果您接受这种默认情况，那么所生成文件就会命名为 **id_rsa** 和 **id_rsa.pub** ，并且分别含有公钥和私钥。您应当随时保护私钥，将其设置为除文件所有者外其他任何人都不可读取，使其免于暴露。然而，公钥则需要传送到您将要登录的系统。您可以使用**ssh-copy-id** 命令来传送密钥至服务器：

```
~]$ ssh-copy-id -i [user@]server
```

这个命令会自动把公钥添加到*服务器*上的 **~/.ssh/authorized_key** 文件中。当您试图登录服务器时，**sshd** 守护进程就会检查此文件。

同样地，对于密码以及其他认证机制，您也应该时常更改 **SSH** 密钥。当您这样做的时候，请确保从**authorized_key** 文件中移除所有不用的密钥。

4.3.10.2. 多种认证方法

使用多种认证方法或者多因素认证，会提升保护水平以防止未经授权的访问；强化系统以防止被入侵，也可起到同样的效果。尝试使用多因素认证登录系统的用户，必须成功通过所有指定的认证方法，才能得到授权进行访问。

使用 **/etc/ssh/sshd_config** 文件中的 **AuthenticationMethods** 配置指令，可指定要使用的认证方法。请注意，使用此指令可以定义多份所需的认证方法列表。如果是那样的话，用户必须在完成至少一份列表上的每种方法。列表需用空格进行分隔，且列表中，每个认证方法的名称必须用逗号分隔。例如：

```
AuthenticationMethods publickey,gssapi-with-mic publickey,keyboard-
interactive
```

如果尝试登录成功的用户是通过 **publickey** 认证和 **gssapi-with-mic** 认证，或是 **publickey** 认证和 **keyboard-interactive** 认证，那么只有使用上述的 **AuthenticationMethods** 指令进行配置的 **sshd** 守护进程才能得到授权进行访问。请注意，每个所要求的认证方法都要使用对应的配置指令（例如，**/etc/ssh/sshd_config** 文件中的**PubkeyAuthentication**），方可准确地启用。关于可用认证方法的常用列表，请参阅 **ssh(1)** 的〈*AUTHENTICATION*〉章节。

4.3.10.3. 其他方法保障 SSH 安全

协议版本

由 Red Hat Enterprise Linux 7 所提供的 **SSH** 协议，即使此协议的运行支持 SSH-1 以及 SSH-2 版本的协议，但是可能的情况下，只使用后者。 SSH-2 版本比起旧版 SSH-1 作了一些的改进，并且大多数高级配置选项只在使用 SSH-2 时才可用。

建议用户使用 SSH-2，这可使 **SSH** 协议对所使用的认证和通讯的保护范围达到最大化。通过使用 **/etc/ssh/sshd_config** 文件中的 **Protocol** 配置指令，可指定 **sshd** 守护进程所支持的协议版本或是其他版本的协议。默认设置是 **2**。

密钥类型

默认情况下，**ssh-keygen** 命令会生成一对 SSH-2RSA 默认密钥；使用 **-t** 选项，通过指令它也可生成 DSA 或 ECDSA 密钥。ECDSA（Elliptic Curve Digital Signature Algorithm，椭圆曲线数字签名算法）在同等的密钥长度下可提供更好的操作。它也可生成较短的密钥。

非默认端口

默认情况下， **sshd** 守护进程会侦听 **22** 网络端口。更改端口会减少系统受到基于自动网络扫描而造成的攻击，从而增加其安全性。通过使用 **/etc/ssh/sshd_config** 配置文件中 **Port** 指令，可指定端口。请注意，要允许使用非默认端口，必须更改 SELinux 默认设置。通过作为 **root** 输入以下指令，修改 **ssh_port_t** SELinux 类型，您可以完成此操作：

```
~]# semanage -a -t ssh_port_t -p tcp port_number
```

在上述命令中，用 **Port** 指令指定的新端口号代替*port_number* 。

非 root 登录

如果特殊使用情况下，无需作为 **root** 用户登录，那么您应该考虑在 **/etc/ssh/sshd_config** 文件中把 **PermitRootLogin** 配置指令设置成 **no**。通过禁止作为 **root** 用户登录，管理者可以审核哪个用户作为常规用户登录后运行了什么特权命令，且之后可获取**root** 权限。

4.4. 安全访问网络

4.4.1. 使用 TCP Wrappers 以及 xinetd 保证服务安全

TCP （Transmission Control Protocol，传输控制协议） Wrapper 程序不仅仅是可以拒绝对某种服务的访问。这个部分将阐明如何使用它们传送连接提示信息程序，警告存在来自某些主机的攻击，以及增强日志记录功能。关于 TCP Wrapper 功能和控制语言的信息，请参阅 **hosts_options(5)** 手册页。关于可用的状态标志寄存器，请参阅 **xinetd.conf(5)** 的手册页。您可将此状态标志寄存器作为选项运用于某一服务。

4.4.1.1. TCP Wrapper 和连接提示

在用户连接到服务时显示适当的提示可让潜在的攻击者知道已经惊动了系统管理员。您还可以控制展示给用户的信息内容。要在服务中添加 TCP Wrapper 提示，则须使用 **banner** 选项。

本例使用 **vsftpd** 来执行 banner 。要启用，则须创建提示信息文件。它是在系统中随处可见，但必须与守护进程的名称保持一致。在本例中，名为 **/etc/banners/vsftpd** 的文件含有以下命令行：

```
220-Hello, %c
220-All Activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being
removed.
```

%c 令牌提供各种客户端信息，比如用户名和主机名，或者用户名和 IP 地址，这些信息可用来生成连接，甚至造成威胁。

要在连入连接显示此提示信息，则须将下列命令行添加到 **/etc/hosts.allow** 文件：

```
vsftpd : ALL : banners /etc/banners/
```

4.4.1.2. TCP Wrapper 和攻击警告

如果已经探测出某个主机或者网络正在攻击该服务器，那么通过使用 **spawn** 指令，使用 TCP Wrapper 可警示管理员关于来自该主机或者网络的后续攻击。

在这个示例中，假设已经探测到来自 206.182.68.0/24 网络的破解者尝试攻击服务器。请将下列命令行放入 **/etc/hosts.deny** 文件，以拒绝任何来自该网络的连接尝试，并将这些尝试记录在特定的文件中：

```
ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >>
/var/log/intruder_alert
```

%d 令牌提供破解者企图访问的服务名称。

要允许连接并予以记录，则须将 **spawn** 指令放入 **/etc/hosts.allow** 文件 。

备注

因为 **spawn** 指令可执行所有 shell 命令，所以最好生成一个特定脚本以提示管理员，或者在特定客户端尝试连接到服务器的事件中，执行一系列命令。

4.4.1.3. TCP Wrapper 和改进的日志

如果比起其它连接类型更令人担忧，那么可使用 **severity** 选项提升该服务的日志级别。

在这个示例中，假设尝试连接到 FTP 服务器 23 端口（Telnet 端口）的任何人就是破解者。要指出这一点，则须在日志文件中使用 **emerg** 标记替换默认标记 **info**，并拒绝连接。

要做到这一点，则须将下列命令行放入 **/etc/hosts.deny** ：

```
in.telnetd : ALL : severity emerg
```

这使用默认 **authpriv** 日志工具，但会将优先权从默认值 **info** 提高到 **emerg**，即将日志信息直接发送到控制台。

4.4.2. 验证使用侦听的端口

应当避免打开不必要的端口，因为这会增加您系统受到攻击的可能性。如果在系统运行之后，您发现有意外打开的端口处于侦听状态，那么这可能就是入侵的迹象，应该对此进行调查。

作为 root 用户，从控制台发出以下命令，以判定哪个端口正在侦听来自网络的连接：

```
~]# netstat -pan -A inet,inet6 | grep -v ESTABLISHED
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address       Foreign Address     State
PID/Program name
tcp        0      0 0.0.0.0:111         0.0.0.0:*           LISTEN
1608/rpcbind
tcp        0      0 127.0.0.1:53        0.0.0.0:*           LISTEN
2581/unbound
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN
2048/sshd
tcp        0      0 127.0.0.1:631       0.0.0.0:*           LISTEN
3202/cupsd
tcp        0      0 0.0.0.0:54136       0.0.0.0:*           LISTEN
2279/rpc.statd
tcp        0      0 127.0.0.1:25        0.0.0.0:*           LISTEN
2708/master
tcp        0      0 127.0.0.1:8953      0.0.0.0:*           LISTEN
2581/unbound
tcp        0      0 127.0.0.1:8955      0.0.0.0:*           LISTEN
2634/dnssec-trigger
tcp6       0      0 :::111              :::*                LISTEN
1608/rpcbind
tcp6       0      0 :::60881            :::*                LISTEN
2279/rpc.statd
tcp6       0      0 ::1:53              :::*                LISTEN
2581/unbound
tcp6       0      0 :::22               :::*                LISTEN
```

```
2048/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN
3202/cupsd
tcp6       0      0 ::1:25                  :::*                    LISTEN
2708/master
tcp6       0      0 ::1:8953                :::*                    LISTEN
2581/unbound
udp        0      0 127.0.0.1:766           0.0.0.0:*
2279/rpc.statd
udp        0      0 0.0.0.0:59186           0.0.0.0:*
674/avahi-daemon: r
udp        0      0 0.0.0.0:33639           0.0.0.0:*
2279/rpc.statd
udp        0      0 0.0.0.0:889             0.0.0.0:*
1608/rpcbind
udp        0      0 127.0.0.1:53            0.0.0.0:*
2581/unbound
udp        0      0 0.0.0.0:68              0.0.0.0:*
2642/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*
1608/rpcbind
udp        0      0 0.0.0.0:46198           0.0.0.0:*
2642/dhclient
udp        0      0 0.0.0.0:123             0.0.0.0:*
697/chronyd
udp        0      0 0.0.0.0:5353            0.0.0.0:*
674/avahi-daemon: r
udp        0      0 127.0.0.1:323           0.0.0.0:*
697/chronyd
udp6       0      0 :::3885                 :::*
2642/dhclient
udp6       0      0 :::889                  :::*
1608/rpcbind
udp6       0      0 ::1:53                  :::*
2581/unbound
udp6       0      0 :::111                  :::*
1608/rpcbind
udp6       0      0 :::123                  :::*
697/chronyd
udp6       0      0 ::1:323                 :::*
697/chronyd
udp6       0      0 :::33235                :::*
2279/rpc.statd
raw6       0      0 :::58                   :::*                    7
2612/NetworkManager
```

请注意，输入 **-l** 选项时，并不会显示 SCTP （stream control transmission protocol，流控制传输协议）服务器。

查核系统所需的服务和命令的输出信息时，关闭那些非特别需要或未经授权的，再重复检查。继续执行，然后使用来自另一系统的 **nmap** 来进行外部检查，此系统是通过网络连接到第一个系统的。这可用于验证 **iptables** 的规则。扫描来自外部系统的 **ss** 输出信息（除了本机 127.0.0.0 或 ::1 区间）中显示的每一个 IP 地址。使用 **-6** 选项，对 IPv6 （Internet Protocol Version 6，网际网路通讯协定第六版）地址进行扫描。更多信息，请参阅 **man nmap(1)** 。

以下示例是从另一系统的控制台发出的命令，用于判定哪个端口正在侦听 TCP 网络连接：

```
~]# nmap -sT -O 192.168.122.1
```

关于 **ss**，**nmap**，以及 **services** 的更多信息，请参阅手册页。

4.4.3. 禁用源路由

源路由是一种互联网协议机制，可允许 IP 数据包携带地址列表的信息，以此分辨数据包沿途经过的路由器。通过某一路径时，会出现一可选项，记录为中间路径。所列出的中间路径，即“路径记录”，可提供返回至源路由路径上的目的地。这就允许源路由可指定某一路径，无论是严格的还是松散的，可忽略路径列表上的一些或全部路由器。它可允许用户恶意重定向网络流量。因此，应禁用源路由。

accept_source_route 选项会导致网络接口接收“*严格源路由选项*”（SSR，Strict Source Route）或“*松散源路由选项*”（LSR，Loose Source Routing） 数据包。源路由数据包的接收是由 sysctl 设置所控制。作为 root 用户，发出以下命令，丢弃 SSR 或 LSR 数据包：

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_source_route=0
```

如上述所言，可能的话（禁止转发可能会干扰虚拟化技术），也应禁止数据包的转发。作为 root 用户，发出以下所列出的命令：

这些命令禁止在所有界面上对 IPv4 和 IPv6 数据包进行转发。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.forwarding=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.forwarding=0
```

这些命令禁止在所有界面上对所有组播数据包进行转发。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.mc_forwarding=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.mc_forwarding=0
```

接收 ICMP （Internet Control Message Protoco，Internet控制报文协议）重定向多为非法使用。除非有特定需要，禁止对此类 ICMP 重定向数据包的接收和传送。

这些命令禁止在所有界面上对所有的 ICMP 重定向数据包进行接收。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_redirects=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0
```

此命令禁止在所有界面上对 ICMP 安全重定向数据包进行接收。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.secure_redirects=0
```

此命令禁止在所有界面上对所有的 IPv4 ICMP 重定向数据包进行接收。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.send_redirects=0
```

这是禁止传送 IPv4 重定向数据包的指示。关于 “IPv6 节点要求” （IPv6 Node Requirements）导致 IPv4 与 IPv6 不同的详细解释，请参阅 〈RFC4294〉 。

要实现永久设定，则必须将其添加到 **/etc/sysctl.conf** 。

更多信息，请参阅 sysctl 手册页 **sysctl(8)** 。关于对源路由及其变量的相关互联网选项的详细解释，请参阅 〈RFC791〉。

警告

以太网可提供其他方式来实现重定向流量，如 ARP （地址解析协议，Address Resolution Protocol）或 MAC （介质访问控制，Medium/Media Access Control）地址欺骗、未经授权的 DHCP （动态主机配置协议，Dynamic Host Configuration Protocol）服务器、以及 IPv6 路由器或邻居通告。此外，偶尔广播的单播流量会导致信息泄露。这些缺点只能通过网络操作员执行的特定对策才能解决。基于主机的对策并非全部有效。

4.4.4. 反向路径过滤

反向路径过滤可用于防止数据包从一接口传入，又从另一不同的接口传出。输出路径与输入路径不同，这有时被称为“非对称路由”（asymmetric routing）。路由器通常会按某种路径传送数据包，但大多数主机并不需要这么做。在以下此类应用程序中常出现异常现象：从一链接输出流量，又从另一不同的服务提供者链接那接收流量。例如，使用结合 xDSL 的租用线路，或是与 3G 网络调制解调器连接的卫星。如果此类场景适用于您，那么就有必要关闭输入接口的反向路径过滤。简而言之，除非必要，否则最好将其关闭，因为它可防止来自子网络的用户采用 IP 地址欺骗手段，并减少 DDoS （分布式拒绝服务，Distributed Denial of Service）攻击的机会。

注意

红帽企业版 Linux 7 根据 RFC 3704 网络入口滤波器的入站过滤文件所推荐的“严格反向路径”（Strict Reverse Path），默认使用严格反向路径过滤。目前只适用于 IPv4 。

警告

如果要启用转发程序，那么只能禁用反向路径过滤，若有其他方式可用于验证源地址（如 iptables 规则示例）。

rp_filter

通过 **rp_filter** 指令启用反向路径过滤。 **rp_filter** 选项可用于指导 kernel （操作系统内核）从三种模式中选择一种。

设置默认行为时，则须采取以下形式：

```
~]# /sbin/sysctl -w net.ipv4.conf.default.rp_filter=INTEGER
```

如果 *INTEGER* 处于以下状态：

- **0** ——未进行源验证。
- **1** ——处于如 RFC3704 所定义的严格模式。
- **2** ——处于如 RFC3704 所定义的松散模式。

通过使用 **net.ipv4.*interface*.rp_filter** 可实现对每一网络接口设置的覆写。要在重启时，实现这些设置能够持续存在，则须修改 **/etc/sysctl.conf** 文件。

4.4.4.1. 附加资源

以下资源对反向路径过滤进行更多的解释。

- **相关网站**

  关于对网络入口滤波器的入站过滤的详细解释，请参阅〈 RFC3704 〉。

- 关于 **/proc/sys/net/ipv4/** 目录下的一系列文件以及可用选项，请参阅 https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt 。

4.5. 使用防火墙

4.5.1. 防火墙简介

动态防火墙后台程序 **firewalld** 提供了一个 动态管理的防火墙，用以支持网络 “zones” ，以分配对一个网络及其相关链接和界面一定程度的信任。它具备对 **IPv4** 和 **IPv6** 防火墙设置的支持。它支持以太网桥，并有分离运行时间和永久性配置选择。它还具备一个通向服务或者应用程序以直接增加防火墙规则的接口。

4.5.2. 了解防火墙

一个图像化的配置工具，**firewall-config**，用于配置 **firewalld**：它依次用 **iptables工具** 与执行数据包筛选的内核中的 **Netfilter** 通信，

使用图像化的 **firewall-config** 工具，按下 **Super** 键进入活动总览，点击 **firewall** ，然后按下 **Enter**。**firewall-config** 工具就出现了。您将被提示输入管理员密码。

firewall-config 工具里有一个标记为 **Configuration** 的下拉菜单，可以在 **运行时间** 和 **永久** 两种模式之间进行选择。要注意，如果您选择了 **Permanent** ，在左上角会出现一排附加的图标。因为不能在运行模式下改变一个服务参数，所以这些图标仅在永久配置模式中出现。

由 **firewalld** 提供的是动态的防火墙服务，而非静态的。因为配置的改变可以随时随地立刻执行，不再需要保存或者执行这些改变。现行网络连接的意外中断不会发生，正如防火墙的所有部分都不需要重新下载。

提供命令行客户端，**firewall-cmd**，用于进行永久性或非永久性运行时间的改变，正如在 **man firewall-cmd(1)**所解释的一样。永久性改变需要按照 **firewalld(1)** 手册页的解释来进行。注意，**firewall-cmd** 命令可以由 **root** 用户运行，也可以由管理员用户——换言之， **wheel** 群体的成员运行。在后一种情况里，命令将通过 **polkit** 进程来授权。

firewalld 的配置储存在 **/usr/lib/firewalld/** 和 **/etc/firewalld/** 里的各种 XML 文件里，这样保持了这些文件被编辑、写入、备份的极大的灵活性，使之可作为其他安装的备份等等。

其他应用程序可以使用 D-bus 和 **firewalld** 通信。

4.5.3. 比较 system-config-firewall 以及 iptables 的 firewalld

firewalld 和 **iptables service** 之间最本质的不同是：

- **iptables service** 在 **/etc/sysconfig/iptables** 中储存配置，而 **firewalld** 将配置储存在 **/usr/lib/firewalld/** 和 **/etc/firewalld/** 中的各种 XML 文件里，。要注意，当 **firewalld** 在Red Hat Enterprise Linux上安装失败时， **/etc/sysconfig/iptables** 文件就不存在。

- 使用 **iptables service**，每一个单独更改意味着清除所有旧有的规则和从 **/etc/sysconfig/iptables**里读取所有新的规则，然而使用 **firewalld** 却不会再创建任何新的规则；仅仅运行规则中的不同之处。因此，**firewalld** 可以在运行时间内，改变设置而不丢失现行连接。

使用 **iptables tool** 与内核包过滤对话也是如此。

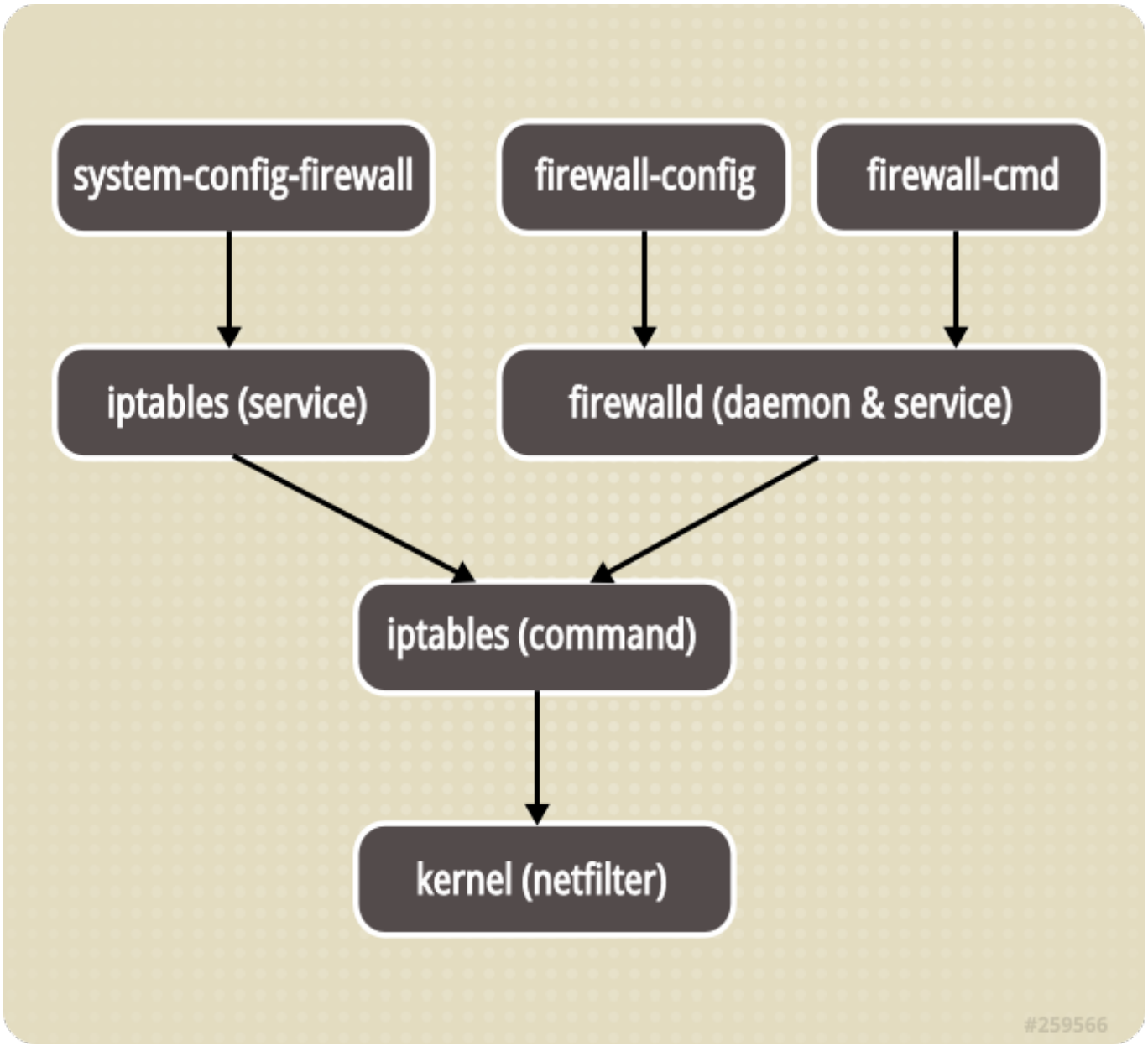

图 4.1. 防火墙堆栈

4.5.4. 对网络区的理解

基于用户对网络中设备和交通所给与的信任程度，防火墙可以用来将网络分割成不同的区域。**NetworkManager** 通知 **firewalld** 一个接口归属某个区域。接口所分配的区域可以由 **NetworkManager** 改变，也可以通过能为您打开相关 **NetworkManager** 窗口的 **firewall-config** 工具进行。

在**/etc/firewalld/**的区域设定是一系列可以被快速执行到网络接口的预设定。列表并简要说明如下：

drop（丢弃）

任何接收的网络数据包都被丢弃，没有任何回复。仅能有发送出去的网络连接。

任何接收的网络数据包都被丢弃，没有任何回复。仅能有发送出去的网络连接。

block（限制）

任何接收的网络连接都被 **IPv4** 的 icmp-host-prohibited 信息和 **IPv6** 的 icmp6-adm-prohibited 信息所拒绝。

public（公共）

在公共区域内使用，不能相信网络内的其他计算机不会对您的计算机造成危害，只能接收经过选取的连接。

external（外部）

特别是为路由器启用了伪装功能的外部网。您不能信任来自网络的其他计算，不能相信它们不会对您的计算机造成危害，只能接收经过选择的连接。

dmz（非军事区）

用于您的非军事区内的电脑，此区域内可公开访问，可以有限地进入您的内部网络，仅仅接收经过选择的连接。

work（工作）

用于工作区。您可以基本相信网络内的其他电脑不会危害您的电脑。仅仅接收经过选择的连接。

home（家庭）

用于家庭网络。您可以基本信任网络内的其他计算机不会危害您的计算机。仅仅接收经过选择的连接。

internal（内部）

用于内部网络。您可以基本上信任网络内的其他计算机不会威胁您的计算机。仅仅接受经过选择的连接。

trusted（信任）

可接受所有的网络连接。

指定其中一个区域为默认区域是可行的。当接口连接加入了 **NetworkManager**，它们就被分配为默认区域。安装时，**firewalld** 里的默认区域被设定为公共区域。

4.5.5. 选择一个网络区域

网络区域名已经选定为不加说明，即可明了，并允许用户快速地做出合理决定。但是，应对默认配置的设定进行检查，而且根据您的需要和风险评估，不必要的服务将不能使用。

4.5.6. 对预先定义的服务的理解

一项服务可以是本地和目的地端口的列表，如果服务被允许的话，也可以是一系列自动加载的防火墙辅助模块。预先定义的服务的使用，让客户更容易被允许或者被禁止进入服务。与对开放端口或者值域，或者端口截然不同，使用预先定义服务，或者客户限定服务，或许能够让管理更容易。 **firewalld.service(5)** 中的手册页描述了服务配置的选择和通用文件信息。服务通过单个的 XML 配置文件来指定，这些配置文件则按以下格式命名：***service-name*.xml**。

用图形化 **firewall-config** 工具查看服务列表，按下 **Super** 键进入开始菜单，输入 **firewall** 然后按下 **Enter**，**firewall-config** 工具就出现了。您将被提示输入管理员密码。现在，在 **Services** 标签下，您可以查看服务列表了。

要使用命令行列出默认的预先定义服务，以 **root** 身份执行以下命令：

```
~]# ls /usr/lib/firewalld/services/
```

请勿编辑**/usr/lib/firewalld/services/** ，只有 **/etc/firewalld/services/** 的文件可以被编辑。

要列出系统或者用户创建的系统，以 **root** 身份执行以下命令：

```
~]# ls /etc/firewalld/services/
```

使用图形化 **firewall-config** 工具和通过编辑 **/etc/firewalld/services/** 中的 XML 文件，服务可以被增加和删除。如果服务没有被用户增加或者改变，那么 **/etc/firewalld/services/** 中不会发现相应的 XML 文件。如果您希望增加或者改变服务， **/usr/lib/firewalld/services/** 文件可以作为模板使用。以 **root** 身份执行以下命令：

```
~]# cp /usr/lib/firewalld/services/[service].xml
/etc/firewalld/services/[service].xml
```

然后您可以编辑最近创建的文件。**firewalld** 优先使用 **/etc/firewalld/services/** 里的文件，如果一份文件被删除且服务被重新加载后，会切换到 **/usr/lib/firewalld/services/**。

4.5.7. 理解直接接口

firewalld 有一个被称为 “direct interface” （直接接口），它可以直接通过 **iptables**、**ip6tables** 和 **ebtables** 的规则。它适用于应用程序，而不是用户。如果您不太熟悉 **iptables**，那么使用直接接口是很危险的，因为您可能无意中导致防火墙被入侵。**firewalld** 保持对所增加项目的追踪，所以它还能质询 **firewalld** 和发现由使用直接端口模式的程序造成的更改。直接端口由增加 **--direct** 选项到 **firewall-cmd** 命令来使用。

直接端口模式适用于服务或者程序，以便在运行时间内增加特定的防火墙规则。这些规则不是永久性的，它们需要在每次通过 D-BU S从 **firewalld** 接到启动、重新启动和重新加载信息后运用。

4.5.8. 检查是否已安装防火墙

在 Red Hat Enterprise Linux 7 中，默认安装 **firewalld** 和图形化用户接口配置工具 **firewall-config**。作为 **root** 用户运行下列命令可以检查：

```
~]# yum install firewalld firewall-config
```

4.5.9. 禁用防火墙

要禁用 **firewalld**，则作为 **root** 用户运行下列命令：

```
~]# systemctl disable firewalld
# systemctl stop firewalld
```

4.5.10. 使用 iptables 服务

要用 **iptables** 和 **ip6tables** 服务代替 **firewalld**，则以 **root** 身份运行以下命令，先禁用 **firewalld**：

```
~]# systemctl disable firewalld
# systemctl stop firewalld
```

然后安装 *iptables-services* 程序包，以 **root** 身份输入以下命令：

```
~]# yum install iptables-services
```

iptables-services 程序包包含了 **iptables** 服务和 **ip6tables** 服务。

然后，以 **root** 身份运行 **iptables** 和 **ip6tables** 命令：

```
  # systemctl start iptables
  # systemctl start ip6tables
  # systemctl enable iptables
  # systemctl enable ip6tables
```

4.5.11. 启动防火墙

要启动 **firewalld**，则以 **root** 用户身份输入以下命令：

```
~]# systemctl start firewalld
```

4.5.12. 检查防火墙是否运行

如果 **firewalld** 在运行，输入以下命令检查：

```
~]$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
   Active: active (running) since Sat 2013-04-06 22:56:59 CEST; 2 days
ago
 Main PID: 688 (firewalld)
   CGroup: name=systemd:/system/firewalld.service
```

另外，检查 **firewall-cmd** 是否可以通过输入以下命令来连接后台程序：

```
~]$ firewall-cmd --state
running
```

4.5.13. 安装防火墙

要安装*firewalld*，则以 **root** 用户身份运行以下命令：

```
~]# yum install firewalld
```

要安装图形化用户接口工具 *firewall-config*，则以 **root** 用户身份运行下列命令：

```
~]# yum install firewall-config
```

4.5.14. 配置防火墙

防火墙可以通过使用图形化用户接口工具 **firewall-config**、命令行接口工具 **firewall-cmd** 和编辑 XML 配置文件来配置。下面会以此详述这些方法：

4.5.14.1. 使用图形化用户接口配置防火墙

4.5.14.1.1. 启动图形化防火墙设置工具

要启动图形化 **firewall-config** 工具，按下 **Super** 键进入开始菜单，点击 **firewall**，然后按 **Enter** 键，**firewall-config** 工具就出现了。您会被提示输入一个管理员密码。

要用命令行启动图形化防火墙配置工具，则以 **root** 用户身份输入以下命令：

```
~]# firewall-config
```

Firewall Configuration 窗口就打开了。注意，这个命令可以由普通用户运行，但随后您会被反复提示输入管理员密码。

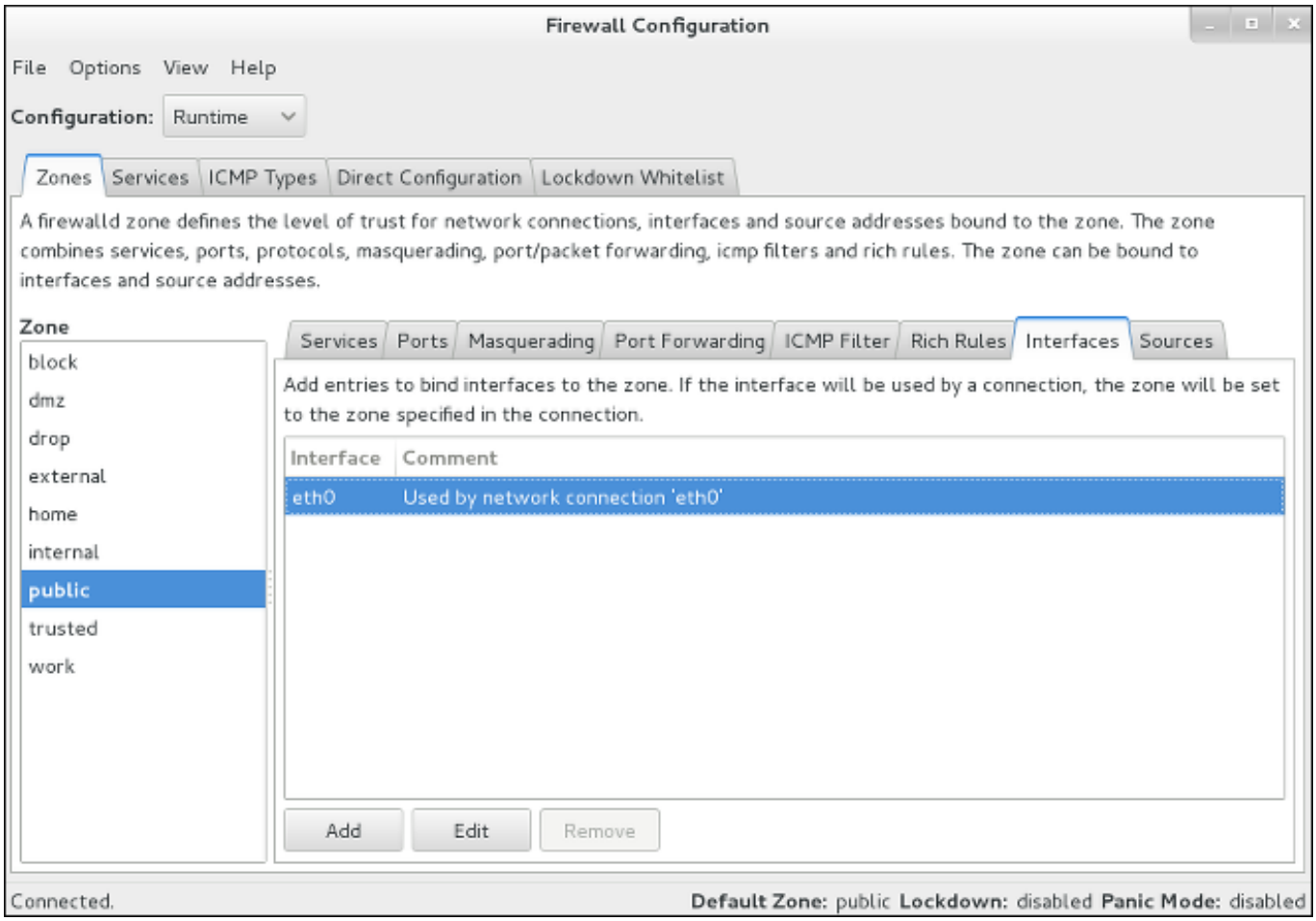

图 4.2. 防火墙配置工具

在左下方角落寻找 “Connected” 字符，这标志着 **firewall-config** 工具已经连接到用户区后台程序 **firewalld**。注意，**ICMP Types**、**Direct Configuration** 和 **Lockdown Whitlist** 标签只在从 **View** 下拉菜单中选择之后才能看见。

4.5.14.1.2. 改变防火墙设置

要立刻改变现在的防火墙设置，须确定当前视图设定在 **Runtime**。或者，从下拉菜单中选择 **Permanent**，编辑下次启动系统或者防火墙重新加载时执行的设定。

注意

在 **Runtime** 模式下更改防火墙的设定时，一旦您启动或者清除连接服务器的复选框，选择立即生效。当您在一个也许还被其他用户使用的系统上工作时，应当谨记这一点。

在**Permanent** 模式下更改防火墙的设定，您的选择将仅仅在您重新加载防火墙或者系统重启之后生效。您可以使用 **文件** 菜单下的重新加载图标，或者点击 **选项** 菜单，选择 **重新加载防火墙** 。

您可以选择左边列里的分区。您将注意到这些分区包含一些可用的服务，您可能需要调整或者滚动窗口才能看见整个列表。您可以通过选择和取消选择一个服务来自定义设定。

4.5.14.1.3. 增加一个接口到分区

要增加或者重新分配一个连接到分区的接口，则启动 **firewall-config**，从菜单栏选择 **Options**，由下拉菜单里选择 **更改连接的分区** ，**Connections** 列表就出现了。选择被分配的连接，出现 **Select Zone for Connection** 窗口。从下拉菜单中选择新的防火墙分区并点击 **OK**。

4.5.14.1.4. 设置默认分区

要设定一个将要被分配新接口的分区作为默认值，则启动 **firewall-config**，从菜单栏选择**Options**，由下拉菜单中选择 **Change Default Zone**，出现**Default Zone** 窗口。从给出的列表中选择您需要用的分区作为默认分区，点击 **OK**。

4.5.14.1.5. 配置服务

要使用或者禁用一个预先设定或用户服务，则启动 **firewall-config**工具并选择将要配置服务的网络分区。选中**Services**标签并选择每个您需要信任的服务类型的复选框。清除复选框则限制服务。

要编辑一项服务，开始 **firewall-config** 工具，然后从标记为 **Configuration** 的下拉选项菜单选择 **Permanent** 模式。其余的图标和菜单案件会出现在 **Services** 窗口的底部。选择您想要配置的服务。

Ports and Protocols 标签可以为选择的服务执行增加、更改、移除端口和协议。模块标签用于配置 **Netfilter** 辅助模块。**Destination**模块使得受限的流量进入一个特定的目的地址和互联网协议 (**IPv4** or **IPv6**)。

4.5.14.1.6. 打开防火墙里的端口

要允许流量通过防火墙到达某个端口，则启动 **firewall-config** 并选择您想更改设定的网络区域。选择 **Ports** 图标并点击右边的 **Add** 按钮，**Port and Protocol** 就打开了。

输入端口数量或者端口号范围，获得许可。从下拉菜单中选择 **tcp** 或者 **udp**。

4.5.14.1.7. 使用伪装的 IP 地址

要将 **IPv4** 地址转换为一个单一的外部地址，则启动 **firewall-config** 工具并选择需要转换地址的网络区域。选择 **Masquerading** 标签和复选框以便把 **IPv4** 地址转换成一个单一的地址。

4.5.14.1.8. 配置端口转发

为一个特定端口转发入站网络流量或 “packets” 到一个内部地址或者替代端口，首先激活伪装IP地址，然后选择 **Port Forwarding** 标签。

在窗口靠上部分选择入站流量协议和端口或者端口范围。靠下部分是用于设置目的端口细节的。

要转发流量到一个本地端口即同一系统上的端口，需选择 **Local forwarding** 复选框，输入要转发的流量的本地端口或者端口值范围。

要转发流量到其他的 **IPv4** 地址，则选择 **Forward to another port** 复选框，输入目的地 IP 地址和端口或者端口范围。如果端口位置空缺则默认发送到同一个端口。点击 **OK** 执行更改。

4.5.14.1.9. 配置 ICMP 过滤

要使用或者禁用一个 **ICMP** 过滤，则启动 **firewall-config** 工具并选择要过滤其信息的网络区域。选择 **ICMP Filter** 图标并选择每种您需要过滤的 **ICMP** 信息类型的复选框。清除复选框以禁用过滤。这种设定是单向的，默认允许全部。

要编辑一个 **ICMP** 类型，则启动 **firewall-config**然后从标签为 **Configuration** 的下拉菜单里选择 **Permanent** 模式。在 **Services** 窗口底部会出现附加图标。

4.5.14.2. 用命令行工具 firewall-cmd 配置防火墙

命令行工具 **firewall-cmd** 是默认安装的应用程序 **firewalld** 的一部分。您可以查证到它是为检查版本或者展示帮助结果而安装的。输入如下命令来检查版本：

```
~]$ firewall-cmd --version
```

输入如下命令来查看帮助输出：

```
~]$ firewall-cmd --help
```

我们在下面选列出一些命令，完整列表请查看操作说明 **man firewall-cmd(1)**。

注意

为了设置一个永久或者可执行命令，除了 **--direct** 命令（它们本质上是暂时的）之外，要向所有命令添加 **--permanent** 选择。注意，这不只是意味着永久更改，而且更改将仅仅在防火墙重新加载、服务器重启或者系统重启之后生效。用 **firewall-cmd** 设置的缺少 **--permanent** 选项的设定能立即生效，但是它仅仅在下次防火墙重新加载、系统启动或者 **firewalld** 服务重启之前可用。防火墙不会在断开连接时重新加载，而会提示您通过重新加载，放弃临时更改。

4.5.14.3. 用命令行接口（CLI）查看防火墙设置

输入以下命令，得到 **firewalld** 的状态的文本显示：

```
~]$ firewall-cmd --state
```

输入以下命令，查看活动分区的列别，并附带一个目前分配给它们的接口列表：

```
~]$ firewall-cmd --get-active-zones
public: em1 wlan0
```

输入以下命令，找出当前分配了接口（例如 em1）的区域：

```
~]$ firewall-cmd --get-zone-of-interface=em1
public
```

以 **root** 身份输入以下命令，找出分配给一个区域（例如公共区域）的所有接口：

```
~]# firewall-cmd --zone=public --list-interfaces
em1 wlan0
```

从 **NetworkManager** 可以得到这个信息，并且仅显示接口而非连接。

以 **root** 用户身份输入以下命令，找出像公共区域这样的一个区域的所有设置：

```
~]# firewall-cmd --zone=public --list-all
public
  interfaces:
  services: mdns dhcpv6-client ssh
  ports:
  forward-ports:
  icmp-blocks: source-quench
```

以 **root** 身份输入以下命令，查看目前活动的网络区域：

```
~]# firewall-cmd --get-service
cluster-suite pop3s bacula-client smtp ipp radius bacula ftp mdns samba
dhcpv6-client dns openvpn imaps samba-client http https ntp vnc-server
telnet libvirt ssh ipsec ipp-client amanda-client tftp-client nfs tftp
libvirt-tls
```

这样将列出 **/usr/lib/firewalld/services/** 中的服务器名称。注意，配置文件是以服务本身命名的 ***service-name*.xml**。

以 **root** 身份输入以下命令，查看所有在防火墙下次加载后将活跃的网络区域：

```
~]# firewall-cmd --get-service --permanent
```

4.5.14.4. 用命令行接口（CLI）更改防火墙设置

4.5.14.4.1. 终止所有数据包（Panic模式）

以 **root** 身份输入以下命令，开始终止所有输入和输出的数据包：

```
~]# firewall-cmd --panic-on
```

所有输入和输出的数据包都将被终止。在一段休止状态之后，活动的连接将被终止；花费的时间由单个会话的超时值决定。

以 **root** 身份输入以下命令，开始再次传输输入和输出的数据包：

```
~]# firewall-cmd --panic-off
```

禁用 panic 模式之后，如果 panic 模式被运行一小段时间，建立的连接可以再次工作。

输入命令，确定 panic 模式被使用或者禁用：

```
~]$ firewall-cmd --query-panic
```

如果在运行模式，屏幕会显示 **yes**，退出状态为 **0**，如果被启用，屏幕会显示 **no**，退出状态为 **0**。

4.5.14.4.2. 用命令行接口（CLI）重新加载防火墙

以 **root** 身份输入以下命令，重新加载防火墙，并不中断用户连接，即不丢失状态信息：

```
~]# firewall-cmd --reload
```

以 **root** 身份输入以下信息，重新加载防火墙并中断用户连接，即丢弃状态信息：

```
~]# firewall-cmd --complete-reload
```

通常在防火墙出现严重问题时，这个命令才会被使用。比如，防火墙规则是正确的，但却出现状态信息问题和无法建立连接。

4.5.14.4.3. 用命令行接口（CLI）为分区增加接口

要为一个分区增加接口，比如，把 em1 增加到公共分区，则以 **root** 身份输入以下命令：

```
~]# firewall-cmd --zone=public --add-interface=em1
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

4.5.14.4.4. 通过编辑接口配置文件为分区增加接口

要通过编辑 **ifcfg-em1** 配置文件来为一个分区增加接口，比如，把 em1 增加到工作分区，需以 **root** 身份用一个编辑器增加以下行到 **ifcfg-em1**：

```
ZONE=work
```

注意，如果您遗漏 **ZONE** 选项，或者使用 use **ZONE=**或**ZONE=''**，那么默认区将被使用。

NetworkManager 程序将自动连接，相应地，分区将被设定。

4.5.14.4.5. 通过编辑防火墙配置文件来配置默认分区

以 **root** 用户身份，打开 **/etc/firewalld/firewalld.conf** 并按如下方式编辑文件：

```
# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=home
```

以 **root** 身份输入以下命令，以重新加载防火墙：

```
~]# firewall-cmd --reload
```

这样可以在不丢失状态信息的同时重新加载防火墙（TCP对话不会被中断）。

4.5.14.4.6. 使用命令行接口（CLI）设置默认分区

以 **root** 用户身份输入以下命令来设置默认分区，比如设置为公共区域：

```
~]# firewall-cmd --set-default-zone=public
```

这个更改将立刻生效，而且在此情况下不需要重新加载防火墙。

4.5.14.4.7. 用命令行接口打开防火墙的端口

通过以 **root** 身份输入以下命令，列出一个区域，例如 dmz 的所有开放端口：

```
~]# firewall-cmd --zone=dmz --list-ports
```

要将一个端口加入一个分区，例如，允许 **TCP** 的流量通过端口 **8080** 的 进入dmz分区，则以 **root** 身份输入以下命令：

```
~]# firewall-cmd --zone=dmz --add-port=8080/tcp
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

要将一系列端口加入一个分区，比如允许从 5060 到 5061 的端口都接入公共分区，则以 **root** 身份输入以下命令：

```
~]# firewall-cmd --zone=public --add-port=5060-5061/udp
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

4.5.14.4.8. 使用命令行接口（CLI）将一个服务加入到分区

要把一个服务加入到分区，例如允许 **SMTP** 接入工作区，则以 **root** 身份运行以下命令：

```
~]# firewall-cmd --zone=work --add-service=smtp
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

4.5.14.4.9. 使用命令行接口（CLI）从一个分区移除服务

要从分区移除服务，比如从工作区移除 **SMTP**，则以 **root** 身份输入以下命令：

```
~]# firewall-cmd --zone=work --remove-service=smtp
```

增加 **--permanent** 可使这个更改在系统启动后被允许。如果用这个选项，并且希望立刻产生更改，以 **root** 身份输入以下命令，重新加载防火墙：

```
~]# firewall-cmd --reload
```

注意，这并不会中断已经建立的连接。如果您打算中断，您可以使用 **--complete-reload** 选项，但这不仅仅中断您已经移除的服务，还会中断所有已经建立的连接。

4.5.14.4.10. 通过编辑 XML 文件为一个分区增加服务

以 **root** 身份输入以下命令，查看默认分区文件：

```
~]# ls /usr/lib/firewalld/zones/
block.xml  drop.xml      home.xml      public.xml   work.xml
dmz.xml    external.xml  internal.xml  trusted.xml
```

这些文件不能编辑。如果 **/etc/firewalld/zones/** 目录里没有等效文件存在，它们被默认为可使用。

以 **root** 身份输入以下命令，查看从默认区被更改的分区文件：

```
~]# ls /etc/firewalld/zones/
external.xml  public.xml  public.xml.old
```

在上述示例中，工作区域文件不存在。以 **root** 身份输入以下命令，加入工作区文件：

```
~]# cp /usr/lib/firewalld/zones/work.xml /etc/firewalld/zones/
```

现在您可以在 **/etc/firewalld/zones/** 目录中编辑该文件。如果您删除该文件，**firewalld** 将切换到使用 **/usr/lib/firewalld/zones/** 里的默认文件。

要将一个服务加入分区，比如允许 **SMTP** 进入工作区，则以 **root** 权限编辑程序，编辑 **/etc/firewalld/zones/work.xml** 文件，使之包括如下行：

```
<service name="smtp"/>
```

4.5.14.4.11. 通过编辑 XML 文件从一个分区中移除服务

编辑 XML 区域文件，必须以 **root** 权限运行编辑程序。以 **root** 身份输入以下命令，查看过去配置的分区的文件：

```
~]# ls /etc/firewalld/zones/
external.xml  public.xml  work.xml
```

以 **root** 权限来编辑程序，编辑 **/etc/firewalld/zones/work.xml** 文件来移除如下行：

```
<service name="smtp"/>
```

就能从一个分区移除服务，比如从工作区移除 **SMTP**。如果 **work.xml** 文件没有进行其他更改，它可以被移除，并且 **firewalld** 会在下一次重新加载或者系统启动之后使用默认的 **/usr/lib/firewalld/zones/work.xml** 配置。

4.5.14.4.12. 配置伪装 IP 地址

如果伪装 IP 不能为一个外部区域启用，则以 **root** 身份输入以下命令来检查：

```
~]# firewall-cmd --zone=external --query-masquerade
```

如果可用，屏幕会显示 **yes**，退出状态为 **0**； 否则，屏幕显示 **no**，退出状态为 **1**。如果省略**zone** ，默认区域将被使用。

以 **root** 身份输入以下命令，允许伪装IP：

```
~]# firewall-cmd --zone=external --add-masquerade
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

以 **root** 身份输入以下命令，禁用伪装IP：

```
~]# firewall-cmd --zone=external --remove-masquerade
```

增加 **--permanent** 选项并重新加载防火墙，使之成为永久性设置。

4.5.14.4.13. 使用命令行接口（CLI）配置端口转发

要将进入网络的程序包从一个端口转发到一个替代端口或者地址，首先需以 **root** 身份输入以下命令来为一个区域（比如外部区域），运行伪装 IP 地址：

```
~]# firewall-cmd --zone=external --add-masquerade
```

以 **root** 身份输入以下命令，把程序包转发到一个本地端口，即相同系统上的一个端口：

```
~]# firewall-cmd --zone=external --add-forward-
port=port=22:proto=tcp:toport=3753
```

在这个例子里，本来要送到 22 端口的程序包现在被转发到 3753 端口。源目的端口用 **port** 选项指定。这个选项可以是一个端口，或者一组端口范围并加上协议。如果指定协议的话，这个协议必须是 **tcp** 或 **udp**。这个新的本地端口，即流量被转发过去的端口或者端口范围，需用 **toport** 选项指定。增加 **--permanent** 选项并重新加载防火墙，可以使设置永久保存。

以 **root** 身份输入以下命令，不改变目的端口将程序包转发到另一个通常是内部地址的 **IPv4** 地址：

```
~]# firewall-cmd --zone=external --add-forward-
port=port=22:proto=tcp:toaddr=192.0.2.55
```

在这个示例中，原本发往22端口的程序包现在被转发到相同的端口，地址则由 **toaddr** 提供。源目的地端口用 **port** 指定。这个选项可能是一个端口，或者一组端口范围并加上协议。如果被指定，协议必须是 **tcp** 或 **udp** 中的一个。这个新端口，即流量被转发过去的端口或者端口范围，用 **toport** 指定。增加 **--permanent** 选项并重新加载防火墙，使这个设定永久保存。

以 **root** 身份输入以下命令，把程序包转发到通常是内部地址的另一个 **IPv4** 地址：

```
~]# firewall-cmd --zone=external /
      --add-forward-
port=port=22:proto=tcp:toport=2055:toaddr=192.0.2.55
```

在这个示例中，原本发往 22 端口的程序包现在被转发到和 **toaddr** 选项一起给出地址的2055端口。源目的端口用 **port** 选项指定。这个选项可以是一个端口，或者打包了协议的端口范围。如果被指定，这个协议一定是 **tcp** 或 **udp** 中的一个。这个新的目的端口，即流量被转发过去的端口或者端口范围，用 **toport** 指定。增加 **--permanent** 选项并重新加载防火墙，使这个设置永久保留。

4.5.14.5. 用 XML 文件配置防火墙

firewalld 的配置设定存储在**/etc/firewalld/** 目录下的 XML 文件里。切勿编辑 **/usr/lib/firewalld/** 目录下的文件，因为它们是为默认设定准备的。查看和编辑这些 XML 文件，您需要 **root** 的用户许可。三个操作手册对 XML 文件进行了解说：

- **firewalld.icmptype(5)** 操作手册 — 描述了 **ICMP** 过滤的 XML 配置文件。
- **firewalld.service(5)** 操作手册 — 描述了 **firewalld service** 的 XML 配置文件。
- **firewalld.zone(5)** 操作手册 — 描述了配置 **firewalld** 区域的 XML 配置文件。

用图形化工具和命令行工具可以对 XML 文件进行直接创建、编辑或者间接创建。组织可以把它们分配到 RPM 文件里，使管理和版本控制更容易。例如 **Puppet** 的工具可以分配这种配置文件。

4.5.14.6. 使用直接接口

通过 **firewall-cmd** 工具，可以使用 **--direct** 选项在运行时间里增加或者移除链。现提供一些例子，请查阅 **firewall-cmd(1)** 操作说明获取更多信息。

如果不熟悉 **iptables** ，使用直接接口非常危险，因为您可能无意间导致防火墙被入侵。

直接端口模式适用于服务或者程序，以便在运行时间内增加特定的防火墙规则。这些规则不是永久性的，它们需要在每次通过 D-BU S从 **firewalld** 接到启动、重新启动和重新加载信息后运用。

4.5.14.6.1. 使用直接接口增加一个自定义规则

以 **root** 身份按照以下格式发布一个命令，增加一个自定义规则到 “IN_public_allow” 链里：

```
~]# firewall-cmd --direct --add-rule ipv4 filter IN_public_allow \
      0 -m tcp -p tcp --dport 666 -j ACCEPT
```

4.5.14.6.2. 用直接接口移除一个自定义规则

以 **root** 用户身份按照以下格式发布一个命令，从 “IN_public_allow” 链移除一个自定义规则：

```
~]# firewall-cmd --direct --remove-rule ipv4 filter IN_public_allow \
      0 -m tcp -p tcp --dport 666 -j ACCEPT
```

4.5.14.6.3. 用直接接口列出自定义规则

以 **root** 用户身份按照以下格式发布一个命令，列出 “IN_public_allow” 链中的规则：

```
~]# firewall-cmd --direct --get-rules ipv4 filter IN_public_allow
```

4.5.15. 给复杂防火墙规则配置“Rich Language" 语法

通过 “rich language” 语法，可以用比直接接口方式更易理解的方法建立复杂防火墙规则。此外，还能永久保留设置。这种语言使用关键词值，是 **iptables** 工具的抽象表示。这种语言可以用来配置分区，也仍然支持现行的配置方式。

4.5.15.1. 多语言命令的格式

在这个部分，所有命令都必须以 **root** 用户身份运行。增加一项规则的命令格式如下：

```
firewall-cmd [--zone=zone] --add-rich-rule='rule' [--timeout
9=seconds]
```

这样将为 *zone* 分区增加一项多语言规则 *rule* 。这个选项可以多次指定。如果分区被省略，将使用默认分区。如果出现超时，规则将在指定的秒数内被激活，并在之后被自动移除。

移除一项规则：

```
firewall-cmd [--zone=zone] --remove-rich-rule='rule'
```

这将为 *zone* 分区移除一项多语言规则 （*rule*） 。这个选项可以多次指定。如果分区被省略，将使用默认分区。

检查一项规则是否存在：

```
firewall-cmd [--zone=zone] --query-rich-rule='rule'
```

这将复查是否已经为区域（*zone*） 增加一个多语言规则 （*rule*） 。如果可用，屏幕会显示 **yes**，退出状态为 **0**； 否则，屏幕显示 **no**，退出状态为 **1**。如果省略 zone，默认区域将被使用。

使用在分区配置文件里的多语言表述的相关信息，可查阅 **firewalld.zone(5)** 说明。

4.5.15.2. 理解多规则结构

多规则命令的格式或结构如下所示：

```
rule [family="<rule family>"]
    [ source address="<address>" [invert="True"] ]
    [ destination address="<address>" [invert="True"] ]
    [ <element> ]
    [ log [prefix="<prefix text>"] [level="<log level>"] [limit
value="rate/duration"] ]
    [ audit ]
    [ accept|reject|drop ]
```

一个规则是关联某个特定分区的，一个分区可以有几个规则。如果几个规则互相影响或者冲突，则执行和数据包相匹配的第一个规则。如果提供了规则系列，它可以是 **ipv4** 或者 **ipv6** 。规则系列把规则限定在 **IPv4** 或 **IPv6** 。如果没有提供规则系列， 将为 **IPv4** 和 **IPv6** 增加规则。如果源地址或者目标地址在一个规则中被使用，那么必须提供规则系列。端口转发也存在这种情况。

4.5.15.3. 理解多规则命令

source

通过制定源地址，一个尝试连接的源头可以被限制在源地址中。一个源地址或者地址范围是一个为 **IPv4** 或者 **IPv6** 做掩护的 IP 地址或者一个网络 IP 地址。网络系列（ **IPv4** 或 **IPv6** ）将被自动覆盖。针对 **IPv4** 的伪装可以是一个网络伪装或者一个普通数字。针对 **IPv4** 的伪装是一个简单数字。不支持使用主机名。可以通过增加 **invert**="*true*" 或 **invert**="*yes*" 来颠倒源地址命令的意思。所有提供的地址都匹配。

destination

通过制定目的地址，目标可以被限制在目的地址中。目标地址使用跟源地址相同的语法。原地址和目标地址的使用是有选择的，不可能目标地址的所有要素都使用。这取决于目标地址的使用，例如在服务项中，这个要素只可以是以下要素类型之一： **service** ， **port** ， **protocol** ， **masquerade** ， **icmp-block** 和 **forward-port** 。

service

服务名称是 **firewalld** 提供的其中一种服务。要获得被支持的服务的列表，输入以下命令： **firewall-cmd --get-services** 。如果一个服务提供了一个目标地址，它将和规则中的目标地址冲突，并且导致一个错误。使用内部目的地址的服务大多是使用了多路传送的服务。命令为以下形式：

```
service name=service_name
```

port

端口既可以是一个独立端口数字，又或者端口范围，例如，5060-5062。协议可以指定为 **tcp** 或 **udp** 。命令为以下形式：

```
port port=number_or_range protocol=protocol
```

protocol

协议值可以是一个协议 ID 数字，或者一个协议名。预知可用协议，请查阅 **/etc/protocols**。命令为以下形式：

```
protocol value=protocol_name_or_ID
```

。

icmp-block

用这个命令阻绝一个或多个 **ICMP** 类型。 **ICMP** 类型是 **firewalld** 支持的 **ICMP** 类型之一。要获得被支持的 **ICMP** 类型列表，输入以下命令：

```
~]$ firewall-cmd --get-icmptypes
```

在此，指定一个动作是不被允许的。**icmp-block** 在内部使用 **reject** 动作。命令为以下形式：

```
icmp-block name=icmptype_name
```

masquerade

打开规则里的 IP 伪装。用源地址而不是目的地址来把伪装限制在这个区域内。在此，指定一个动作是不被允许的。

forward-port

从一个带有指定为 **tcp** 或 **udp** 协议的本地端口转发数据包到另一个本地端口，或另一台机器，或另一台机器上的另一个端口。 **port** 和 **to-port** 可以是一个单独的端口数字，或一个端口范围。而目的地址是一个简单的 IP 地址。在此，指定一个动作是不被允许的。 **forward-port** 命令使用内部动作 **accept** 。这个命令为以下形式：

```
forward-port port=number_or_range protocol=protocol /
             to-port=number_or_range to-addr=address
```

log

注册含有内核记录的新的连接请求到规则中，比如系统记录。您可以定义一个前缀文本——可以把记录信息作为前缀加入。记录等级可以是 **emerg** 、 **alert** 、 **crit** 、 **error** 、**warning** 、 **notice** 、 **info** 或者 **debug** 中的一个。可以选择记录的用法，可以按以下方式限制注册：

```
log [prefix=prefix text] [level=log level] limit
value=rate/duration
```

等级用正的自然数 [1, ..] 表达，持续时间的单位为 **s** 、 **m** 、 **h** 、 **d** 。 **s** 表示秒， **m** 表示分钟， **h** 表示小时， **d** 表示天。最大限定值是 **1/d** ，意为每天最多有一条日志进入。

audit

审核为发送到 **auditd** 服务的审核记录来注册提供了另一种方法。审核类型可以是 **ACCEPT** 、 **REJECT** 或 **DROP** 中的一种，但不能在 **audit** 命令后指定，因为审核类型将会从规则动作中自动收集。审核不包含自身参数，但可以选择性地增加限制。审核的使用是可选择的。

accept|reject|drop

可以是 **accept** 、**reject** 或 **drop** 中的一个行为。规则中仅仅包含一个要素或者来源。如果规则中包含一个要素，那么行为可以处理符合要素的新连接。如果规则中包含一个来源，那么指定的行为可以处理来自源地址的一切内容。

```
accept | reject [type=reject type] | drop
```

选择 **accept** 所有新的连接请求都会被允许。选择 **reject** ，连接将被拒绝，连接来源将接到一个拒绝信息。拒绝的类型可以被设定为使用另一种值。选择 **drop** ， 所有数据包会被丢弃，并且不会向来源地发送任何信息。

4.5.15.4. 使用多规则登录命令

使用 **Netfilter** 登录目标可以完成登录，也可以使用审核目标。用 “*zone*_log” 格式命名的新链可以加入到所有分区，其中 *zone* 为该分区名。在 **deny** 链之前进行该项处理，以便获得适当的排序。根据规则的行为，整个规则或者部分规则会按照规则被分别放置在独立链中，如下所示：

```
zone_log
zone_deny
zone_allow
```

所有登录规则将放在 “*zone*_log” 链中，这会最先被解析。所有 **reject** 和 **drop** 规则都被放置在 “*zone*_deny” 链，在登录链之后被解析。所有 **accept** 规则被放在 “*zone*_allow” 链里，它将在 **deny** 链之后被解析。如果规则中既包含了 **log** ，又有 **deny** 或者 **allow** ，各部分将被放在相应的链中。

4.5.15.4.1. 多规则登录命令使用示例 1

为认证报头协议 **AH** 使用新的**IPv4** 和 **IPv6** 连接：

```
rule protocol value="ah" accept
```

4.5.15.4.2. 多规则登录命令使用示例 2

同意新的 **IPv4** 和 **IPv6** 连接 **FTP** ，并使用审核每分钟登录一次：

```
rule service name="ftp" log limit value="1/m" audit accept
```

4.5.15.4.3. 多规则登录命令使用示例 3

为 **TFTP** 协议同意来自 **192.168.0.0/24** 地址的新的 **IPv4** 连接，并且使用系统日志每分钟登录一次：

```
rule family="ipv4" source address="192.168.0.0/24" service name="tftp"
log prefix="tftp" level="info" limit value="1/m" accept
```

4.5.15.4.4. 多规则登录命令使用示例 4

为 **RADIUS** 协议拒绝所有来自 **1:2:3:4:6::** 的新 **IPv6** 连接，并每分钟在级别3登录。接受来自其他来源的新的 **IPv6** 连接：

```
rule family="ipv6" source address="1:2:3:4:6::" service name="radius" log
prefix="dns" level="info" limit value="3/m" reject
rule family="ipv6" service name="radius" accept
```

4.5.15.4.5. 多规则登录命令使用示例 5

转发带有 **TCP** 协议的端口 4011 上的来自 **1:2:3:4:6::** 的 **IPv6** 包，到端口 4012 上的 **1::2:3:4:7** 。

```
rule family="ipv6" source address="1:2:3:4:6::" forward-port to-
addr="1::2:3:4:7" to-port="4012" protocol="tcp" port="4011"
```

4.5.15.4.6. 多规则登录命令使用示例 6

把一个源地址加入白名单，以便允许来自这个源地址的所有连接

```
rule family="ipv4" source address="192.168.2.2" accept
```

更多示例请查阅 **firewalld.richlanguage(5)** 说明页。

4.5.16. 锁定防火墙

如果以 **root** 身份运行本地应用或者服务（比如 **libvirt** ），就能更改防火墙设置。因为这个功能，管理员可以锁定防火墙设置，这样无论是不向锁定的白名单添加应用，还是仅允许添加应用，都可以要求防火墙更改。锁定设置默认不启动，如果启动，用户可以确保本地应用或者服务不需要对防火墙做任何设置更改。

4.5.16.1. 设置防火墙锁定

以 **root** 身份运行一个编辑器，把以下行增加到 **/etc/firewalld/firewalld.conf** 文件：

```
Lockdown=yes
```

以 **root** 身份使用以下命令重启防火墙：

```
~]# firewall-cmd --reload
```

欲在默认区内使用 **imaps** 服务，则以管理员账户，也就是 **wheel** 组中的用户（通常是系统的第一位用户），使用以下命令：

```
~]$ firewall-cmd --add-service=imaps
Error: ACCESS_DENIED: lockdown is enabled
```

欲使用 **firewall-cmd** ，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python -Es
/usr/bin/firewall-cmd*'
```

如果需要重启后会继续使用此设定，增加 **--permanent** 选项。

以 **root** 身份重启防火墙：

```
~]# firewall-cmd --reload
```

以管理员账户输入以下命令，尝试在默认区里再次启动 **imaps** 服务。您将被提示输入用户密码：

```
~]$ firewall-cmd --add-service=imaps
```

这样，命令成功运行。

4.5.16.2. 用命令行客户端配置锁定

查询锁定是否执行，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --query-lockdown
```

如果是锁定状态，打印退出状态为 **0** 的 **yes**。否则，打印退出状态为 **1** 的 **no**。

启动锁定，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --lockdown-on
```

关闭锁定，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --lockdown-off
```

4.5.16.3. 用命令行配置锁定白名单选项

锁定白名单可以包含命令，安全环境，用户和用户ID。如果白名单上输入的一个命令以一个星号 “*” 结束，那么所有以这个命令开始的命令行都匹配。如果没有 “*” ，那么包括参数的绝对命令必须匹配。

环境，是指一个正在运行的应用或者服务的安全 (SELinux) 环境。用以下命令获取一个正在运行的应用的环境：

```
~]$ ps -e --context
```

这个命令检查所有运行中的应用。通过 **grep** 工具将输出转移，得到需要的应用。比如：

```
~]$ ps -e --context | grep example_program
```

列出白名单上的所有命令行，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --list-lockdown-whitelist-commands
```

增加一个*command* 命令到白名单，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python -Es
/usr/bin/command'
```

从白名单移除一个 *command* 命令，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --remove-lockdown-whitelist-command='/usr/bin/python
-Es /usr/bin/command'
```

查询 *command* 命令是否在白名单上，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --query-lockdown-whitelist-command='/usr/bin/python -
Es /usr/bin/command'
```

如果存在，显示退出状态为 **0** 的 **yes** ，否则，显示退出状态为 **1** 的 **no** 。

列出白名单上的所有安全环境，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --list-lockdown-whitelist-contexts
```

增加一个环境 *context* 到白名单，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --add-lockdown-whitelist-context=context
```

要使这个命令持续，增加 **--permanent** 选项。

从白名单移除一个环境 *context* ，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --remove-lockdown-whitelist-context=context
```

要使这个命令持续，增加 **--permanent** 选项。

查询白名单上是否有环境 *context* ，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --query-lockdown-whitelist-context=context
```

如果存在，显示退出状态为 **0** 的 **yes** ，否则，显示退出状态为 **1** 的 **no** 。

列出白名单上所有用户 ID ，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --list-lockdown-whitelist-uids
```

增加一个用户 ID *uid* 到白名单，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --add-lockdown-whitelist-uid=uid
```

要使这个命令持续，增加 **--permanent** 选项。

从白名单上移除一个用户 ID *uid* ，以**root** 身份输入以下命令：

```
~]# firewall-cmd --remove-lockdown-whitelist-uid=uid
```

要使这个命令持续，增加 **--permanent** 选项。

查询用户 ID *uid* 是否在白名单上，输入以下命令：

```
~]$ firewall-cmd --query-lockdown-whitelist-uid=uid
```

如果存在，显示退出状态为 **0** 的 **yes** ，否则，显示退出状态为 **1** 的 **no** 。

列出白名单上所有用户名，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --list-lockdown-whitelist-users
```

增加一个用户名 *user* 到白名单，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --add-lockdown-whitelist-user=user
```

要使这个命令持续，增加 **--permanent** 选项。

从白名单移除一个用户名 *user* ，以 **root** 身份输入以下命令：

```
~]# firewall-cmd --remove-lockdown-whitelist-user=user
```

要使这个命令持续，增加 **--permanent** 选项。

查询用户名 *user* 是否在白名单上，输入以下命令：

```
~]$ firewall-cmd --query-lockdown-whitelist-user=user
```

如果存在，显示退出状态为 **0** 的 **yes** ，否则，显示退出状态为 **1** 的 **no** 。

4.5.16.4. 用配置文件来配置锁定白名单选项

默认的白名单配置文件包括 **NetworkManager** 环境和 **libvirt** 的默认环境。列表里也有用户 ID 0。

```
<?xml version="1.0" encoding="utf-8"?>
<whitelist>
  <selinux context="system_u:system_r:NetworkManager_t:s0"/>
  <selinux context="system_u:system_r:virtd_t:s0-s0:c0.c1023"/>
  <user id="0"/>
</whitelist>
```

这里跟随了一个示例白名单配置文件，它启动用于 **firewall-cmd** 功能的所有命令，为名为 *user*、用户 ID 为 **815** 的用户：

```
<?xml version="1.0" encoding="utf-8"?>
<whitelist>
  <command name="/usr/bin/python -Es /bin/firewall-cmd*"/>
  <selinux context="system_u:system_r:NetworkManager_t:s0"/>
  <user id="815"/>
  <user name="user"/>
</whitelist>
```

在这个范例里，我们出示了 **user id** 和 **user name** 两样，但只需要一个即可。Python 是一个解释器，所以写在命令行的最前面。您也可以使用一个非常特别的命令，比如：

```
/usr/bin/python /bin/firewall-cmd --lockdown-on
```

在这例子里，只有 **--lockdown-on** 命令会被允许。

注意

在 Red Hat Enterprise Linux 7 中，所有功能现在都放在 **/usr/bin/** 中，而且 **/bin/** 目录被系统链接到 **/usr/bin/** 目录。换言之，尽管以 **root** 身份运行的 **firewall-cmd** 路径可能解析到 **/bin/firewall-cmd**，但是现在会使用 **/usr/bin/firewall-cmd** 。所有新的脚本可以使用新的地址，但要意识到，如果以 **root** 身份运行的脚本被写入使用 **/bin/firewall-cmd** 路径，那么，命令路径除了是传统意义上仅用于非 **root** 用户的 **/usr/bin/firewall-cmd** 路径以外，还必须被加入白名单。

在一个命令的名字属性结尾的 “*” 意味着所有以此行开头的命令都匹配。如果没有 “*” ，那么包括参数的绝对命令必须匹配。

4.5.17. 附加资源

下列信息的来源提供了关于 **firewalld** 的附加资源。

4.5.17.1. 已安装的文档

- **firewalld(1)** 说明页——描述 **firewalld** 的命令选项。
- **firewalld.conf(5)** 说明页——包括配置 **firewalld** 的信息。
- **firewall-cmd(1)** 说明页——描述 **firewalld** 命令行客户端的命令选项。
- **firewalld.icmptype(5)** 操作手册 — 描述了 **ICMP** 过滤的 XML 配置文件。
- **firewalld.service(5)** 操作手册 — 描述了 **firewalld service** 的 XML 配置文件。
- **firewalld.zone(5)** 操作手册 — 描述了配置 **firewalld** 区域的 XML 配置文件。
- **firewalld.direct(5)** 说明页——描述 **firewalld** 直接接口配置文件。
- **firewalld.lockdown-whitelist(5)** 说明页——描述 **firewalld** 白名单锁定配置文件。
- **firewall.richlanguage(5)** 说明页——描述 **firewalld** 多语言规则语法。
- **firewalld.zones(5)** 说明页——概述分区情况以及如何配置它们。

4.6. 用 DNSSEC 保护 DNS 流量

4.6.1. 介绍 DNSSEC

DNSSEC是一套“ *域名系统安全扩展* ”(DNSSEC，Domain Name System Security Extensions) ，能让“ **域名系统** ”（DNS，Domain Name System）客户端进行身份验证以及检查来自 **DNS** 域名服务器响应的完整性，以此鉴定它们的来源，并判断它们是否在传输过程中被篡改过。

4.6.2. 了解 DNSSEC

对于通过互联网连接，现在有越来越多的网站使用 **超文本传输协议安全（HTTPS，Hyper Text Transfer Protocol Security）** 来提供安全的链接。然而，除非您直接输入 IP 地址，在连接到 **HTTPS** 网络服务器之前，必须执行 **DNS** 查询。由于缺少身份验证，执行这些 **DNS** 查询是不安全的，且会遭到“ *中间人* ”攻击（ MITM，man-in-the-middle attacks）。换句话说， **DNS** 客户端无法确信疑似来自特定的 **DNS** 域名服务器的应答是否可信，以及是否被篡改过。更重要的是，递归服务器无法确定从其他域名服务器获取的记录是真实的。 **DNS** 协议无法提供客户端可确保不遭受中间人攻击的机制。 DNSSEC 的引入解决了在使用 **DNS** 解析域名时，缺少身份验证和完整性检查的问题。但它不能解决机密性的问题。

DNSSEC 所发布的信息包括 **DNS** 资源记录的数字签字和公用加密密钥的分配，以这样的方式让 **DNS** 解析器建立起多层次的信息链。因所有 **DNS** 资源记录而生成的数字签名，添加到此 DNS 区域作为资源记录签名（RRSIG）。此区域所添加的公用加密密钥作为资源记录的域名系统密钥（ DNSKEY ）。要建立起多层次的信息链，则须将 DNSKEY 的散列值发布到父区域作为“ *代理签名* ”（DS，Delegation of Signing）资源记录。要验证不存在性，则须使用 *NextSECure* （NSEC，下一代安全）和 NSEC3 （NSEC的替换或备用方案）资源记录。在 DNSSEC 区域签名中，每一个“ *资源记录集* ”（RRset，resource record set） 都有其对应的 RRSIG 资源记录。请注意， 用作子区域代理的记录（域名服务器粘附记录，NS and glue records）并没有进行签名；这些记录要显示在子区域，并在此区域进行签名。

运用 root 区域公用加密密钥进行配置的解析器会完成处理 DNSSEC 的信息。使用这种密钥，解析器可以验证用于 root 区域的签名。例如， root 区域对 **.com** 的 DS 记录进行签名。 root 区域也为 **.com** 域名服务器提供域名服务器粘附记录。解析器会跟踪代理和查询使用代理域名服务器 **.com** 的DNSKEY 记录。所获取的 DNSKEY 记录散列值应当与 root 区域的 DS 记录相匹配。如果匹配，则解析器将会信任所获取的 **.com**

DNSKEY 。在 **.com** 区域内， RRSIG 记录是由 **.com** DNSKEY 所创建。同样地，在 **.com** 中的代理也是重复此程序，例如 **redhat.com**。用这种方法，尽管 **DNS** 验证解析器在其正常操作期间收集了很多 DNSKEY ，但只需用一个 root 密钥对其进行配置。如果密码检查失败了，则解析器将 SERVFAIL 会返回给应用程序。

DNSSEC 的设计是根据以下这种方式：对于不支持 DNSSEC 的应用程序完全不可见。如果非 DNSSEC 应用程序查询支持 DNSSEC的解析器 ，则它所接收的答复没有任何新资源记录类型，如 RRSIG 。然而， 支持 DNSSEC 的解析器仍将执行所有密码检查，若探测到恶意 **DNS** 答复，它仍会将 SERVFAIL 错误返回给应用程序。 DNSSEC 会保护 **DNS** 服务器（权威服务器和递归服务器）数据的完整性，但却不为应用程序和解析器提供的安全保护。因此，予以一个从应用程序到其解析器的安全传输方式十分重要。实现这一目的最容易的方法就是运行 **localhost** （本地主机）上支持 DNSSEC 的解析器，使用 **/etc/resolv.conf**下的 **127.0.0.1** 。或者可以使用虚拟专用网络（ VPN ，Virtual Private Network）连接到远程 **DNS** 服务器。

了解热点问题

使用无线网络热点（ Wi-Fi Hotspot ，Wireless Fidelity Hotspot）或 VPN 时，就会依赖 “DNS 欺骗”（DNS lies）。所获取的端口往往会发生 **DNS** 劫持，以便重定向用户跳转到需要身份验证（或支付）的 Wi-Fi 服务网页。连接 VPN 的用户常常需使用 “内部专用” **DNS** 服务器，以便定位那些在公司网络外不存在的资源。这需要软件进行额外处理。例如， **dnssec-trigger** 可用于探测一个无线热点（ Hotspot ）是否劫持 **DNS** 查询，或 **unbound** 是否充当代理域名服务器处理 DNSSEC 查询。

选择支持 DNSSEC 的递归解析器

要部署支持 DNSSEC 的递归解析器，则可使用 **BIND** 或 **unbound** 。这两者在默认情况下都使用 DNSSEC ，并用 DNSSEC root 密钥进行配置。在服务器上使用 DNSSEC ，两者都可正常工作。然而， **unbound** 更常用于移动设备，如笔记本电脑。因为它允许本机用户对 DNSSEC 覆写动态重配置，无论是使用 **dnssec-trigger** 时无线热点所需求的， 亦或是使用 **Libreswan** 时 VPNs 所需求的。 **unbound** 守护进程进一步支持对列入 **etc/unbound/*.d/**目录的 DNSSEC 异常状况进行部署，这对服务器和移动设备都有用。

4.6.3. 了解 Dnssec-trigger

一旦 **unbound** 完成安装，并在 **/etc/resolv.conf**下进行配置，则所有来自应用程序的 **DNS** 查询都会通过 **unbound**进行处理。**dnssec-trigger** 只有在被触发时，才会对 **unbound** 解析器进行重配置。这大多数运用于漫游的客户机，如笔记本电脑，这种可连接到不同 Wi-Fi 网络的机器。其过程如下：

- 通过 **动态主机配置协议（DHCP，Dynamic host configuration protocol）** 获取新的 **DNS** 服务器时，则 **NetworkManager** “会触发” **dnssec-trigger** 。
- 随后，**Dnssec-trigger** 会对服务器执行一系列测试，判断其是否完全支持 DNSSEC 。
- 如果支持，那么 **dnssec-trigger** 会重配置 **unbound** ，以用于作为所有查询转发程序的 **DNS** 服务器。
- 如果测试失败，则**dnssec-trigger** 将忽略新的 **DNS** 服务器，并尝试一些可行的退却方法。
- 如果它判定一个不受限制的 53 端口（**用户数据报协议（UDP，User Datagram Protocol）** 以及 **传输控制协议（TCP，Transmission Control Protocol）** ）可以使用，则它将告知 **unbound** 可成为全递归 **DNS** 服务器，无需使用任何转发程序。
- 如果无法完成操作，如因 53 端口被防火墙阻拦，此防火墙会阻挡除连接网络的 **DNS** 服务器之外的所有程序，则它将会尝试通过使用 **DNS** 到 80 端口，亦或通过使用 **DNS** 封装的 **安全传输层协议（TLS，Transport Layer Protocol）** 到 443 端口。在 80 端口和 443 端口运行 **DNS** 的服务器可在 **/etc/dnssec-trigger/dnssec-trigger.conf**下进行配置。注释的范例可在默认配置文件中找到。
- 如果这些退却方法也失败了，则 **dnssec-trigger** 将提供一种不安全的操作，这将完全忽略 DNSSEC ；亦或它将在 “缓存专用” （cache only）模式下运行，此模式下它将不会尝试新的 **DNS** 查询，但将会应答所有已在缓存器中的数据。

无线热点更是常常在授予访问网络权限之前，重定向用户到登录页面。在探测上述编列期间，如果探测到重定向命令，则会提示用户，以询问是否通过要求登录来获取网络访问权限。**dnssec-trigger** 守护进程将继续对 DNSSEC 解析器每十秒进行探测。关于使用 **dnssec-trigger** 图形化工具的更多信息，请参阅第 4.6.8 节 “使用 Dnssec-trigger”。

4.6.4. 提供域和域名服务器的 VPN

VPN 一些连接类型可传输域和一系列域名服务器， 可用于作为 VPN 隧道安装部分的域。在 **红帽企业版 Linux** 中，这是由 **NetworkManager** 所支持的。这就是说， **unbound**、 **dnssec-trigger** 和 **NetworkManager** 的结合产物能够完全支持 VPN 软件所提供的域和域名服务器。一旦 VPN 隧道完成，就可以清除关于所有接收域名的登录在本机的 **unbound** 缓存，从而通过 VPN 获取的内部域名服务器中提取最新对域名名称的查询。终止 VPN 隧道时，则会再次清除 **unbound** 缓存，以确保任何对域的查询会返回给公用 IP 地址，而不会返回到原先获取的 IP 地址。请参阅〈第 4.6.11 节 “对连接所提供的域进行 DNSSEC 验证配置”〉。

4.6.5. 建议的命名惯例

Red Hat 建议，静态域名和动态域名都要与用于 **DNS** 机器的“ *完全合格域名* ”（FQDN，fully-qualified domain name）相匹配，如 **host.example.com**。

互联网名称与数字地址分配机构（ICANN，The Internet Corporation for Assigned Names and Numbers）有时会将原先未注册的顶级域名（TLD， Top-Level Domain ）（如 **.yourcompany**）添加到公共注册平台。因此，Red Hat 强烈建议，请勿使用不代表自己的域名，即使在专用网络。因为这可能导致同一域名根据网络配置进行不同的解析。结果，可能导致无法使用网络资源。使用一个不代表自己的域名也使 DNSSEC 的部署和维持更加困难，因为域名冲突需要手动配置才能启用 DNSSEC 验证。关于此问题的更多信息，请参阅〈ICANN 域名冲突的常见问题（ICANN FAQ on domain name collision）〉。

4.6.6. 了解信任锚（Trust Anchor）

信任锚由 **DNS** 域名以及此域名相关的公用密钥（或公用密钥的散列值）组成。其表述为一个基本的 64 比特加密密钥。其类似于一种信息交换方式的证书，含有公用密钥，可用于对 **DNS** 记录进行核实和身份验证。关于信任锚更加完整的定义，请参阅〈RFC 4033〉。

4.6.7. 安装 DNSSEC

4.6.7.1. 安装 unbound

要通过在本机上使用 DNSSEC 对 **DNS** 进行验证，则必须安装 **DNS** 解析器 **unbound** （或 **bind** ）。移动设备上只需安装 **dnssec-trigger** 。对于服务器而言，安装**unbound** 就应当足够了，尽管根据服务器所在地（局域网（LAN，local area network） 或 Internet），可能会要求对本地域进行转发配置。 **dnssec-trigger** 当下只在全球公共 DNS 区域提供帮助。**NetworkManager**，**dhclient**， 以及 VPN 应用程序通常可以自动收集域列表（和域名服务器列表），但 **dnssec-trigger** 和 **unbound** 却不行。

要安装 **unbound** ，则须作为 **root** 用户允许以下命令：

```
~]# yum install unbound
```

4.6.7.2. 检查 unbound 是否在运行

要判定 **unbound** 守护进程是否在运行，则须输入以下命令：

```
~]$ systemctl status unbound
 unbound.service - Unbound recursive Domain Name Server
   Loaded: loaded (/usr/lib/systemd/system/unbound.service; disabled)
   Active: active (running) since Wed 2013-03-13 01:19:30 CET; 6h ago
```

systemctl status 命令将会报告 **unbound Active: inactive (dead)** ，若 **unbound** 服务未在运行。

4.6.7.3. 启动 unbound

要启动 **unbound** 守护进程用于当前会话，则须作为 **root** 用户运行以下命令：

```
~]# systemctl start unbound
```

运行 **systemctl enable** 命令，以确保每次启动系统时， **unbound** 开始运行：

```
~]# systemctl enable unbound
```

unbound 守护进程允许使用以下目录对本地数据或覆写进行配置：

- **/etc/unbound/conf.d** 用于为特定的域名添加配置。这用于重定向域名查询到特定的 **DNS** 服务器。这通常用于只存在于企业广域网（WAN，wide area network）的子域。
- **/etc/unbound/keys.d** 目录用于为特定的域名添加信任锚。这在 DNSSEC 对内部专用域名进行签名时才需要，但并没有公用现有的 DS 记录来建立信任途径。另一种使用情况是，当对内部域进行签名时，使用不同的 DNSKEY ，而不是使用企业广域网之外可行的公用域名。
- **/etc/unbound/local.d** 目录用于添加特定的 **DNS** 数据作为本地覆写。者可用于建立黑名单，或创建手动覆写。这个日期将会通过 **unbound** 返回给客户端，但是不会被标记为有 DNSSEC 签名。

NetworkManager 和一些 VPN 软件可改变动态配置。这些配置目录含有注释范例。更多信息请参阅 **unbound.conf(5)** 手册页。

4.6.7.4. 安装 Dnssec-trigger

dnssec-trigger 应用程序作为 **dnssec-triggerd**守护进程来运行。要安装 **dnssec-trigger** ，则须作为 **root** 用户运行以下命令：

```
~]# yum install dnssec-trigger
```

4.6.7.5. 检查 Dnssec-trigger 守护进程是否在运行

要判定 **dnssec-triggerd** 是否在运行，则须输入以下命令：

```
~]$ systemctl status dnssec-triggerd
systemctl status dnssec-triggerd.service
dnssec-triggerd.service - Reconfigure local DNS(SEC) resolver on network
change
   Loaded: loaded (/usr/lib/systemd/system/dnssec-triggerd.service;
enabled)
   Active: active (running) since Wed 2013-03-13 06:10:44 CET; 1h 41min
ago
```

systemctl status 命令将会报告 **dnssec-triggerd** **Active: inactive (dead)** ，若 **dnssec-triggerd** 守护进程未在运行。要在当前会话中启用，则须作为 **root** 用户运行以下命令：

```
~]# systemctl start dnssec-triggerd
```

运行 **systemctl enable** 命令，以确保每次启动系统时， **unbound** 开始运行：

```
~]# systemctl enable dnssec-triggerd
```

4.6.8. 使用 Dnssec-trigger

dnssec-trigger 应用程序有GNOME panel 的功能，用于显示 DNSSEC 探测结果，以及用于执行 DNSSEC 探测命令请求。要启用此功能，则须按 **Super** 键进入应用程序概览视图（ Activities Overview），输入 **DNSSEC** ，然后再按 **Enter** 。一个形似船锚的图标将会添加到屏幕底部的消息托盘。可通过按屏幕底部右侧的蓝色圆形通知图标来显示。右击锚状图标，则会出现弹出式菜单。

正常操作下， **unbound** 在本机可用作域名服务器， **resolv.conf** 会指向 **127.0.0.1**。当您在 **无线热点登录（Hotspot Sign-On）** 界面点击 **OK** 时，这就会改变。 **DNS** 服务器受到 **NetworkManager** 的查询，并被放入 **resolv.conf**。然后您就可以在无线热点登录页面进行身份验证。锚状图标会显示巨大的红色感叹号以作警示，提醒您 **DNS** 查询的执行并不安全。身份验证后， **dnssec-trigger** 可自动检测，并转换到安全模式。尽管在某些情况下，它无法自动检测，则用户必须手工操作，选择 **重新检测（Reprobe）** 。

正常情况下，**Dnssec-trigger** 不需要用户进行交互操作。一旦 启用，它会在后台工作。如果出现问题，它会弹出消息框来通知用户。它也会将 **resolv.conf** 文件的变更通知 **unbound** 。

4.6.9. 对 DNSSEC 使用 dig 命令

要查看 DNSSEC 是否在工作，则可用不同的命令行工具。最好的使用工具就是 *bind-utils* 工具包中的 **dig** 命令。其他可用的工具分别是， *ldns* 工具包中的 **drill** 和 *unbound* 工具包中的 **unbound-host** 。旧版的 **DNS** 实用程序 **nslookup** 和 **host** 都已过时，不应再使用。

要使用 **dig** 发送 DNSSEC 数据查询请求，则须添加 **+dnssec** 选项到命令中，例如：

```
~]$ dig +dnssec whitehouse.gov
; <<>> DiG 9.9.3-rl.13207.22-P2-RedHat-9.9.3-4.P2.el7 <<>> +dnssec
whitehouse.gov
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21388
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;whitehouse.gov.   IN A

;; ANSWER SECTION:
whitehouse.gov.  20 IN A 72.246.36.110
whitehouse.gov.  20 IN RRSIG A 7 2 20 20130825124016 20130822114016 8399
whitehouse.gov. BB8VHWEkIaKpaLprt3hq1GkjDROvkmjYTBxiGhuki/BJn3PoIGyrftxR
HH0377I0Lsybj/uZv5hL4UwWd/lw6Gn8GPikqhztAkgMxddMQ2IARP6p
wbMOKbSUuV6NGUT1WWwpbi+LelFMqQcAq3Se66iyH0Jem7HtgPEUE1Zc 3oI=
```

```
;; Query time: 227 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Aug 22 22:01:52 EDT 2013
;; MSG SIZE  rcvd: 233
```

除 A 记录之外，所返回的 RRSIG 记录含有 DNSSEC 签名以及签名的初始时间和截止时间。据 **unbound** 服务器显示，通过返回的顶端 **flags:** 区段下 **ad** 比特可知数据已经过 DNSSEC 身份验证。

如果 DNSSEC 验证失败，则 **dig** 命令将会返回 SERVFAIL 错误：

```
~]$ dig badsign-a.test.dnssec-tools.org
; <<>> DiG 9.9.3-rl.156.01-P1-RedHat-9.9.3-3.P1.el7 <<>> badsign-
a.test.dnssec-tools.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 1010
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;badsign-a.test.dnssec-tools.org. IN A

;; Query time: 1284 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Aug 22 22:04:52 EDT 2013
;; MSG SIZE  rcvd: 60]
```

要请求查看关于失败的更多信息，则须指定 **+cd** 选项加入到 **dig** 命令中，以禁止 DNSSEC 检查：

```
~]$ dig +cd +dnssec badsign-a.test.dnssec-tools.org
; <<>> DiG 9.9.3-rl.156.01-P1-RedHat-9.9.3-3.P1.el7 <<>> +cd +dnssec
badsign-a.test.dnssec-tools.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26065
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;badsign-a.test.dnssec-tools.org. IN A

;; ANSWER SECTION:
badsign-a.test.dnssec-tools.org. 49 IN A 75.119.216.33
badsign-a.test.dnssec-tools.org. 49 IN RRSIG A 5 4 86400 20130919183720
20130820173720 19442 test.dnssec-tools.org.
E572dLKMvYB4cgTRyAHIKKEvdOP7tockQb7hXFNZKVbfXbZJOIDREJrr
zCgAfJ2hykfY0yJHAlnuQvM0s6xOnNBSvc2xLIybJdfTaN6kSR0YFdYZ
n2NpPctn2kUBn5UR1BJRin3Gqy20LZlZx2KD7cZBtieMsU/IunyhCSc0 kYw=

;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Aug 22 22:06:31 EDT 2013
;; MSG SIZE  rcvd: 257
```

通常， DNSSEC 错误会自己显示错误的初始时间或截止时间。尽管在本例中， www.dnssec-tools.org 的访问者蓄意损坏 RRSIG 签名，这是我们无法通过手动查看此输出来进行探测。错误将会显示在 **systemctl status unbound** 的输出中，且 **unbound** 守护进程会将这些错误记录到 **syslog** ，如下所示：

```
Aug 22 22:04:52 laptop unbound: [3065:0] info: validation failure
badsign-a.test.dnssec-tools.org. A IN
```

使用 **unbound-host**的示例：

```
~]$ unbound-host -C /etc/unbound/unbound.conf -v whitehouse.gov
whitehouse.gov has address 184.25.196.110 (secure)
whitehouse.gov has IPv6 address 2600:1417:11:2:8800::fc4 (secure)
whitehouse.gov has IPv6 address 2600:1417:11:2:8000::fc4 (secure)
whitehouse.gov mail is handled by 105 mail1.eop.gov. (secure)
whitehouse.gov mail is handled by 110 mail5.eop.gov. (secure)
whitehouse.gov mail is handled by 105 mail4.eop.gov. (secure)
whitehouse.gov mail is handled by 110 mail6.eop.gov. (secure)
whitehouse.gov mail is handled by 105 mail2.eop.gov. (secure)
whitehouse.gov mail is handled by 105 mail3.eop.gov. (secure)
```

4.6.10. 装配 Dnssec-trigger 无线热点探测设备

连接网络时， **dnssec-trigger** 会尝试探测无线热点。无线热点通常是一种会在可使用网络之前迫使用户进行网页交互的设备。通过尝试下载一已知内容的指定网页，来完成探测。如果存在无线热点，则不会有如预期所料的接收内容。

要设置一已知的固定网页，使其可通过 **dnssec-trigger** 用于探测无线热点，则须如下执行：

1. 对某些在互联网上可公开访问的机器，设置其网页服务器。关于网页服务器的更多信息，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》。
2. 一旦您让服务器开始运行，则会将已知内容的静态页面发布到服务器上。此页面无需一定是有效的 HTML 页面。例如，您可使用一个名为 **hotspot.txt**只含有 **OK** 字符串的纯文本文件。假设您的服务器位于 **example.com** ，您可将 **hotspot.txt** 文件发布到网页服务器的 ***document_root*/static/** 子目录，那么您静态网页服务器的地址将是 **example.com/static/hotspot.txt** 。请参阅《 Red Hat Enterprise Linux 7 系统管理员指南》下的 **DocumentRoot** 指令。
3. 将以下命令行添加到 **/etc/dnssec-trigger/dnssec-trigger.conf** 文件：

   ```
   url: "http://example.com/static/hotspot.txt OK"
   ```

   此命令添加了一个可通过 **HTTP** （80端口）探测到的 URL 。第一部分就是可解析的 URL 以及可下载的页面。命令的第二部分是所下载的网页预期含有的文本字符串。

关于配置选取的更多信息，请参阅手册页的 **dnssec-trigger.conf(8)** 。

4.6.11. 对连接所提供的域进行 DNSSEC 验证配置

在默认情况下，转发区及其固有的域名服务器会通过 **dnssec-trigger** 自动添加到 **unbound** ，以用于任何通过 **NetworkManager** 的连接所提供的域，除了 Wi-Fi 连接之外。默认情况下，所有添加到 **unbound** 的转发区都已进行 DNSSEC 验证。

用于验证转发区的默认行为可被更改，从而所有的转发区在默认情况下将 **不** 会进行 DNSSEC 验证。要做到这一点，则须更改 **dnssec-trigger** 配置文件 **/etc/dnssec.conf** 下的 **validate_connection_provided_zones** 变量。作为 **root** 用户，打开并编辑以下命令行：

```
validate_connection_provided_zones=no
```

无法更改任何现有的转发区，只能更改未来的转发区。因此，如果您想禁止 DNSSEC 用于当前所提供的域，那么您需要重新连接。

4.6.11.1. 对 Wi-Fi 所提供的域进行 DNSSEC 验证配置

为 Wi-Fi 所提供的区域添加转发区即可启用。要实现此功能，则须更改 **dnssec-trigger** 配置文件 **/etc/dnssec.conf** 下 **add_wifi_provided_zones** 变量。作为 **root** 用户，打开并编辑以下命令行：

```
add_wifi_provided_zones=yes
```

对任何已存在的转发区无法进行更改，只能对将要执行的转发区进行更改。因此，如果您要禁止 DNSSEC 用于当前 Wi-Fi 所提供的域，那么您需要重新连接（重新开启） Wi-Fi 。

警告

要“ **打开** ”添加到 **unbound** 作为转发区的 Wi-Fi 所提供的域，则可能会出现安全隐患，例如：

1. 一个 Wi-Fi 接入点可能有意通过 **DHCP（ Dynamic host configuration protocol，动态主机配置协议）** 给您提供一个域，而它并无 DHCP 的权限，也无法将您所有的 **DNS** 查询发送到其 **DNS** 服务器。
2. 如果您对“ **关闭** ”的转发区进行 DNSSEC 验证，那么 Wi-Fi 所提供的 **DNS** 服务器可从所提供的域中，伪造用于域名的 **IP** 地址，而您并不知情。

4.6.12. 附加资源

以下这些资源将对 DNSSEC 进行更多的 解释。

4.6.12.1. 已安装的文档

- **dnssec-trigger(8)** 手册页 —— 描述用于 **dnssec-triggerd**，**dnssec-trigger-control** 以及 **dnssec-trigger-panel** 的命令选项。
- **dnssec-trigger.conf(8)** 手册页 —— 描述用于 **dnssec-triggerd** 的配置选项。
- **unbound(8)** 手册页 —— 描述用于 unbound 以及 **DNS** 验证解析器的命令选项。
- **unbound.conf(5)** 手册页 —— 含有配置 **unbound** 的信息。
- **resolv.conf(5)** 手册页 —— 含有解析器例程所读取的信息。

4.6.12.2. 在线文档

http://tools.ietf.org/html/rfc4033

RFC 4033 DNS 安全介绍及其要求（ DNS Security Introduction and Requirements）。

http://www.dnssec.net/

有链接到许多 DNSSEC 资源的网站。

http://www.dnssec-deployment.org/

DNSSEC 部署计划（DNSSEC Deployment Initiative）由国土安全部赞助（Department for Homeland Security），含有大量 DNSSEC 信息，并通过 邮件列表来讨论 DNSSEC 部署事项。

http://www.internetsociety.org/deploy360/dnssec/community/

国际互联网大会（Internet Society）的“Deploy 360” 计划是为了促进并协调 DNSSEC 部署，这是在全球范围内发现团体和 DNSSEC 活动的良好资源。

http://www.unbound.net/

此文档含有关于 **unbound DNS** 服务的基本信息。

http://www.nlnetlabs.nl/projects/dnssec-trigger/

此文档含有关于 **dnssec-trigger** 的基本信息。

4.7. 保护虚拟私用网络（VPN）

在 Red Hat Enterprise Linux 7 中，VPN可以用受到 **Libreswan** 应用支持的 **IPsec** 加密通道协议来进行配置（ *Virtual Private Network*(VPN) ）。**Libreswan** 是 **Openswan** 应用的一个分支，是可交换文档中的例子。**NetworkManager IPsec** 插件称为 *NetworkManager-libreswan*。GNOME Shell 的用户需要安装带有 *NetworkManager-libreswan* 附件的 *NetworkManager-libreswan-gnome* 数据包。

在 Red Hat Enterprise Linux 7中，**Libreswan** 是一个开放源，用户空间 **IPsec**的实践项目可以从中获得。它使用 *Internet key exchange* (IKE) 协议， **IKE** 版本 1 和版本 2 被作为用户级别的后台程序来执行。手动密钥也可以通过 **ip xfrm** 命令建立，但不推荐这样做。 **Libreswan**与 Linux 内核连接， 用网络链接来转移加密秘钥。加密包和解密包在 Linux 内核中发生。

Libreswan 使用 *network security services* (NSS) 加密库， 这是 *Federal Information Processing Standard* (FIPS) 安全合规要求的。

4.7.1. 使用 Libreswan 的 IPsec VPN

要安装 **Libreswan**，以 **root** 身份输入以下命令：

```
~]# yum install libreswan
```

检查 **Libreswan**是否已安装，输入以下命令：

```
~]$ yum info libreswan
```

新安装 **Libreswan** 之后， NSS 数据库将作为安装过程的一部分被初始化。但是，如果您要开始一个新的数据库，首先要按以下方式移除旧的数据库：

```
~]# rm /etc/ipsec.d/*db
```

然后，初始化一个新的 NSS 数据库，以 **root** 身份输入以下命令：

```
~]# ipsec initnss
Enter a password which will be used to encrypt your keys.
```

```
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

如果您不想使用 NSS 密码，那么在被提示输入密码的时候，直接按两次 **Enter**。如果您输入了密码，那么每次 **Libreswan** 启动时， 您需要再次输入密码，就像每次系统启动时一样。

检查由 **Libreswan** 提供的 **ipsec** 后台程序是否运行，输入以下命令：

```
~]$ systemctl status ipsec
ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled)
   Active: inactive (dead)
```

启动由 **Libreswan** 提供的 **ipsec** 后台程序，以 **root** 身份输入以下命令：

```
~]# systemctl start ipsec
```

确定后台程序正在运行：

```
~]$ systemctl status ipsec
ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled)
   Active: active (running) since Wed 2013-08-21 12:14:12 CEST; 18s ago
```

确定启动系统式，**Libreswan** 也会启动，以 **root** 身份输入以下命令：

```
~]# systemctl enable ipsec
```

配置媒介以及基于主机的防火墙来允许 **ipsec** 服务。查阅 <第 4.5 节 “使用防火墙” > 得到防火墙和允许指定程序通过的有关信息。**Libreswan** 要求防火墙允许以下数据包通过：

- 针对 **Internet Key Exchange** (IKE) 协议的 **UDP** 端口 500
- 针对 **IKE NAT-Traversal** 的 **UDP** 端口 4500
- 针对 **Encapsulated Security Payload** (ESP) **IPsec** 数据包的端口 50
- 针对 **Authenticated Header** (AH) **IPsec** 数据包（非常见）的端口 51

我们提供了三个例子，用 **Libreswan** 建立一个 **IPsec** VPN。第一个例子是将两个主机连接在一起，使之可以安全通讯。第二个例子是将两个站点连接起来组成一个网络。第三个例子是支持漫游用户，在此环境里被称为 *road warriors*。

4.7.2. 使用 Libreswan 的 VPN 配置

Libreswan 不使用术语 “source”（来源） 或 “destination”（目的）。相反，它用术语 “left”（左边） 和 “right”（右边） 来代指终端（主机）。虽然大多数管理员用 “left” 表示本地主机，“right” 表示远程主机，但是这样可以再大多数情况下在两个终端上使用相同的配置。

有三种常用的方法为终端提供认证：

- *Pre-Shared Keys* （PSK） 是最简单的证明方法。PSK 由随机字符组成，长度至少为 20 个字符。考虑到非随机和和短的 PSK 的危险，当系统在 FIPS 模式下运行时，这个方法不能使用。

- Raw RSA 值常用于静态的主机对主机，或者子网对子网的 **IPsec** 配置。这些主机用彼此的公共 RSA 秘钥手动配置。当许多或者更多主机都需要彼此建立 **IPsec** 通道时，这个方法不能很好地扩展。
- X.509 认证常用于有许多主机需要连接到一个常用的 **IPsec** 通道的大规模配置。一个中央认证中心（*certificate authority* （CA））被用于为主机或者用户注册 RSA 认证。这个中央 CA 负责转播信任关系，包括取消每个主机和用户。

4.7.3. 使用 Libreswan 的主机对主机 VPN

要配置 **Libreswan** 创建一个主机对主机 **IPsec** VPN，在两个被指定为 “left” 和 “right” 的主机之间，以 **root** 身份在指定为 “left” 的主机上输入以下命令，创建一个新的 RSA 秘钥组：

```
~]# ipsec newhostkey --configdir /etc/ipsec.d \
          --output /etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
```

这样产生一个用于主机的 RSA 秘钥组。产生 RSA 的过程要花上好几分钟，尤其是在带低熵的虚拟机上。

要查看公共秘钥，以 **root** 身份在指定为 “left” 的主机上输入以下命令：

```
~]# ipsec showhostkey --left
# rsakey AQOrlo+hO
leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
```

您需要这个秘钥来增加配置文件，如下文所示：

以 **root** 身份在指定为 “right” 的主机上输入以下命令：

```
~]# ipsec newhostkey --configdir /etc/ipsec.d \
          --output /etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
```

要查看公共秘钥，以 **root** 身份在指定为 “right” 的主机上输入以下命令：

```
~]# ipsec showhostkey --right
# rsakey AQO3fwC6n
rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
```

您将需要把这个秘钥增加到配置文件。

秘密的部分被存储在 **/etc/ipsec.d/*.db** 文件里，也称为 “NSS 数据库”。

要为这种主机对主机的通道建立配置文件，要把上面的行 **leftrsasigkey=** 和 **rightrsasigkey=** 增加到一个位于 **/etc/ipsec.d/** 目录中的自定义配置里。要让 **Libreswan** 读出用户配置文件，则以 **root** 身份使用编辑器来编辑主配置文件 **/etc/ipsec.conf** ，并通过移除 **#** 注释符来使用以下行，这一行看起来是这样：

```
include /etc/ipsec.d/*.conf
```

以 **root** 身份使用编辑器，用如下格式创建一个带有合适名称的文件：

```
/etc/ipsec.d/my_host-to-host.conf
```

按照如下方式编辑文件：

```
conn mytunnel
    leftid=@west.example.com
    left=192.1.2.23
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...]
W2n417C/4urYHQkCvuIQ==
    rightid=@east.example.com
    right=192.1.2.45
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    authby=rsasig
    # load and initiate automatically
    auto=start
```

您可以在左右主机上使用完全相同的配置文件。系统会自动侦测 “left” 或 “right”。如果其中一个主机是移动主机，致使 **IP** 地址无法提前获取，那么就在移动主机上把 **%defaultroute** 用作它的 **IP** 地址。它能自动获取动态 **IP** 地址。在接受了来自接入手机的连接的静态主机上，用 **%any** 指定移动主机的 **IP** 地址。

确保 **leftrsasigkey** 值从 “left” 主机上获取，确定 **rightrsasigkey** 从 “right” 主机上获取。

重启 **ipsec** 来确保它读取新的配置：

```
~]# systemctl restart ipsec
```

以 **root** 输入以下命令来加载 **IPsec** 通道：

```
~]# ipsec auto --add mytunnel
```

要建立通道，在 left 或者 right，以 root 目录输入以下命令：

```
~]# ipsec auto --up mytunnel
```

4.7.3.1. 查证使用 Libreswan 的主机对主机 VPN

IKE 协议产生于 **UDP** 端口 500。 **IPsec** 数据包展示为 **Encapsulated Security Payload** （ESP）数据包。当 VPN 连接需要通过一个 NAT 路由器时，**ESP** 数据包在端口 4500 上被打包在 **UDP** 数据包里。

要核实数据包正在通过 VPN 通道被发送，以 **root** 身份按照以下格式输入一条命令：

```
~]# tcpdump -n -i interface esp and udp port 500 and udp port 4500
00:32:32.632165 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1a),
length 132
00:32:32.632592 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1a),
length 132
00:32:32.632592 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489,
seq 7, length 64
00:32:33.632221 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1b),
length 132
00:32:33.632731 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1b),
length 132
00:32:33.632731 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489,
seq 8, length 64
00:32:34.632183 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1c),
length 132
00:32:34.632607 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1c),
length 132
```

```
00:32:34.632607 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489,
seq 9, length 64
00:32:35.632233 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1d),
length 132
00:32:35.632685 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1d),
length 132
00:32:35.632685 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489,
seq 10, length 64
```

其中 *interface* 就是用来负荷通信的接口。要停止使用 **tcpdump** 捕获的数据包，按下 **Ctrl+C**。

注意

tcpdump 命令完全无法和 **IPsec** 互动。它仅仅识别向外的加密程序包，而不是向外的纯文档文件程序包。它可以识别进入的加密程序包和进入的解码程序包。如果可以，在两个机器之间的路由器而非其中一个终端上运行 **tcpdump**。

4.7.4. 使用 Libreswan 的点对点 VPN

要为 **Libreswan** 创建一个点对点 **IPsec** VPN，并连接两个网络，要在两个主机之间创建一个 **IPsec** 通道，配置终端允许一个或者更多子网通过。所以，它们可以被看作是通向网络远程部分的门户。点对点 VPN 的配置和主机对主机 VPN 仅有的不同在于，必须在配置文件中指定一个或者更多的网络或子网。

要配置 **Libreswan** 来创建一个点对点 **IPsec** VPN，首先按照 < 第 4.7.3 节 “使用 Libreswan 的主机对主机 VPN” > 所述，配置一个主机对主机 **IPsec** VPN，然后拷贝或者移动文件到一个带有适当名称的文件里，例如 **/etc/ipsec.d/my_site-to-site.conf**。以 **root** 身份使用编辑器编辑，编辑自定义配置文件 **/etc/ipsec.d/my_site-to-site.conf** 如下：

```
conn mysubnet
     also=mytunnel
     leftsubnet=192.0.1.0/24
     rightsubnet=192.0.2.0/24

conn mysubnet6
     also=mytunnel
     connaddrfamily=ipv6
     leftsubnet=2001:db8:0:1::/64
     rightsubnet=2001:db8:0:2::/64

conn mytunnel
    auto=start
    leftid=@west.example.com
    left=192.1.2.23
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...]
W2n417C/4urYHQkCvuIQ==
    rightid=@east.example.com
    right=192.1.2.45
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    authby=rsasig
```

要建立通道，需重启 **Libreswan** 或者手动加载，并初始化所有连接，以 **root** 身份使用以下命令：

```
~]# ipsec auto --add mysubnet
```

```
~]# ipsec auto --add mysubnet6
```

```
~]# ipsec auto --add mytunnel
```

```
~]# ipsec auto --up mysubnet
104 "mysubnet" #1: STATE_MAIN_I1: initiate
003 "mysubnet" #1: received Vendor ID payload [Dead Peer Detection]
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
106 "mysubnet" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "mysubnet" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "mysubnet" #1: received Vendor ID payload [CAN-IKEv2]
004 "mysubnet" #1: STATE_MAIN_I4: ISAKMP SA established
{auth=OAKLEY_RSA_SIG cipher=aes_128 prf=oakley_sha group=modp2048}
117 "mysubnet" #2: STATE_QUICK_I1: initiate
004 "mysubnet" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel
mode {ESP=>0x9414a615 <0x1a8eb4ef xfrm=AES_128-HMAC_SHA1 NATOA=none
NATD=none DPD=none}
```

```
~]# ipsec auto --up mysubnet6
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
117 "mysubnet" #2: STATE_QUICK_I1: initiate
004 "mysubnet" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel
mode {ESP=>0x06fe2099 <0x75eaa862 xfrm=AES_128-HMAC_SHA1 NATOA=none
NATD=none DPD=none}
```

```
~]# ipsec auto --up mytunnel
104 "mytunnel" #1: STATE_MAIN_I1: initiate
003 "mytunnel" #1: received Vendor ID payload [Dead Peer Detection]
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
106 "mytunnel" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "mytunnel" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "mytunnel" #1: received Vendor ID payload [CAN-IKEv2]
004 "mytunnel" #1: STATE_MAIN_I4: ISAKMP SA established
{auth=OAKLEY_RSA_SIG cipher=aes_128 prf=oakley_sha group=modp2048}
117 "mytunnel" #2: STATE_QUICK_I1: initiate
004 "mytunnel" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel
mode {ESP=>0x9414a615 >0x1a8eb4ef xfrm=AES_128-HMAC_SHA1 NATOA=none
NATD=none DPD=none}
```

4.7.4.1. 核实带有 Libreswan 的点对点VPN

核实数据包正在通过 VPN 通道被发送，是和 <第 4.7.3.1 节 “查证使用 Libreswan 的主机对主机 VPN” >中所解释的完全一样的过程。

4.7.5. 使用 Libreswan 的点对点单一隧道 VPN

通常，当点对点的通道创建完成，网关需要使用它们内在的 **IP** 地址，而不是它们公共的 **IP** 地址来互相联系。这使用单一隧道可以实现。如果名为 **west** 的 left 的主机，拥有内在的 **IP** 地址 **192.0.1.254**，如果名为**east** 的 right 的主机拥有内在的 **IP** 地址 **192.0.2.254**，可以使用单一隧道的配置可以被使用：

```
conn mysubnet
```

```
    leftid=@west.example.com
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...]
W2n417C/4urYHQkCvuIQ==
    left=192.1.2.23
    leftsourceip=192.0.1.254
    leftsubnet=192.0.1.0/24
    rightid=@east.example.com
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    right=192.1.2.45
    rightsourceip=192.0.2.254
    rightsubnet=192.0.2.0/24
    auto=start
    authby=rsasig
```

4.7.6. 子网延伸使用 Libreswan

经常， **IPsec** 被部署在集散体系结构。每个叶节点都有 **IP** 范围，它是更大范围的一部分。叶通过集线器相互联系。这被称为 “ *子网延伸*”。在下列例子中，我们使用 **10.0.0.0/8** 配置总部及使用更小两个 **/24** 子网的分支。

在总部：

```
conn branch1
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=5.6.7.8
    rightid=@branch1
    righsubnet=10.0.1.0/24
    rightrsasigkey=0sAXXXX[...]
    #
    auto=start
    authby=rsasigkey

conn branch2
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=10.11.12.13
    rightid=@branch2
    righsubnet=10.0.2.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasigkey
```

在 “branch1”的办公室，我们使用相同的链接。另外我们使用传递链接来排除我们从隧道被运送的本地 LAN 流量：

```
conn branch1
    left=1.2.3.4
```

```
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=10.11.12.13
    rightid=@branch2
    righsubnet=10.0.1.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasigkey

conn passthrough
    left=1.2.3.4
    right=0.0.0.0
    leftsubnet=10.0.1.0/24
    rightsubnet=10.0.1.0/24
    authby=never
    type=passthrough
    auto=route
```

4.7.7. 使用 Libreswan 的 Road Warrior 应用

Road Warrior 是具有动态分配 **IP** 地址的流动客户端的旅行用户，比如说笔记本电脑。这些通过证书进行身份验证。

在服务器上：

```
conn roadwarriors
    left=1.2.3.4
    # if access to the LAN is given, enable this
    #leftsubnet=10.10.0.0/16
    leftcert=gw.example.com
    leftid=%fromcert
    right=%any
    # trust our own Certificate Agency
    rightca=%same
    # allow clients to be behind a NAT router
    rightsubnet=vhost:%priv,%no
    authby=rsasigkey
    # load connection, don't initiate
    auto=add
    # kill vanished roadwarriors
    dpddelay=30
    dpdtimeout=120
    dpdaction=%clear
```

在流动客户端上，也就是 Road Warrior 的设备上，我们需要稍微修改以上配置：

```
conn roadwarriors
    # pick up our dynamic IP
    left=%defaultroute
    leftcert=myname.example.com
    leftid=%fromcert
    # right can also be a DNS hostname
```

```
    right=1.2.3.4
    # if access to the remote LAN is required, enable this
    #rightsubnet=10.10.0.0/16
    # trust our own Certificate Agency
    rightca=%same
    authby=rsasigkey
    # Initiate connection
    auto=start
```

4.7.8. Road Warrior 应用使用了 Libreswan 与 X.509 的 XAUTH

当使用 XAUTH **IPsec** 扩展名来建立链接时，**Libreswan** 本身提供了分配 **IP** 地址的方法以及 DNS 信息去漫游VPN客户端。也可使用 PSK 或 X.509 证书来部署 XAUTH，使用 X.509 部署更安全。客户端证书可以被证书吊销列表或 *"在线证书状态协议"* (OCSP） 吊销。使用 X.509 证书，个体客户端不能模拟服务器。使用 PSK，也被称为组密码，在理论上是可行的。

此外，XAUTH 要求 VPN 客户端使用用户名和密码来识别自身。用一次性密码（OTP，One time Passwords),比如谷歌验证器或 RSA 安全 ID 标记，一次性标记可被附加到用户密码之后。

对XAUTH有三种可能的后端：

xauthby=pam

它使用在 **/etc/pam.d/pluto** 的配置来验证用户。 它自身可以使用多种后端来配置 **Pam**。它可以使用系统账户用户密码方案在、LDAP 目录、RADIUS 服务器或自定义密码验证模块。

xauthby=file

它使用配置文件 **/etc/ipsec.d/passwd**（不要与 **/etc/ipsec.d/nsspassword** 混淆）。这个文件的格式与 **Apache 、htpasswd** 文档类似和 **Apache htpasswd** 命令可以被用来创建此文件里的条目。但是，在用户名和密码之后，要求第三列使用 **IPsec** 链接名的链接，比如说当使用"链接远程用户"提供VPN来删除用户，密码文件的条目应该看起来如下：

```
user1:$apr1$MIwQ3DHb$1I69LzTnZhnCT2DPQmAOK.:remoteusers
```

NOTE：当使用 **htpasswd** 命令，链接名须被手动添加在 *用户之后：每一行user:password 之后*。

xauthby=alwaysok

服务器总是会假定 XAUTH 用户和密码的组合是正确的。尽管服务器忽略了这些，客户端也需要指定用户名和密码。这些只有当用户被 X.509 证书识别之后才能被使用，或者在不需要 XAUTH 后端时检测 VPN。

使用 X.509 证书的配置示例

```
conn xauth-rsa
    auto=add
    authby=rsasig
    pfs=no
    rekey=no
    left=ServerIP
    leftcert=vpn.example.com
    #leftid=%fromcert
    leftid=vpn.example.com
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    rightaddresspool=10.234.123.2-10.234.123.254
```

```
    right=%any
    rightrsasigkey=%cert
    modecfgdns1=1.2.3.4
    modecfgdns2=8.8.8.8
    modecfgdomain=example.com
    modecfgbanner="Authorized Access is allowed"
    leftxauthserver=yes
    rightxauthclient=yes
    leftmodecfgserver=yes
    rightmodecfgclient=yes
    modecfgpull=yes
    xauthby=pam
    dpddelay=30
    dpdtimeout=120
    dpdaction=clear
    ike_frag=yes
    # for walled-garden on xauth failure
    # xauthfail=soft
    #leftupdown=/custom/_updown
```

当 **xauthfail** 被设定为“soft"，而不是 "hard"，验证失败便被忽略，VPN 被设定为像验证用户是成功的一样。从上至下的自定义脚本可以被用来检查环境变量 **XAUTH_FAILED**。这些用户可以被重新定向，比如使用 iptables DNAT 重新定向至 “墙内的花园” ，在那里他们可以联系管理员，或者更新这项服务的付费订阅。

VPN 客户端使用 **modecfgdomain** 值和 DNS 条目去重新定向为指定的域查询指定的名称服务器。这使得漫游用户可以使用内部 DNS 名称，来访问仅供内部使用的资源。

如果 **leftsubnet** 不是 **0.0.0.0/0**，拆分隧道配置请求会被自动送到客户端。比如说，当使用 **leftsubnet=10.0.0.0/8**，VPN 客户端只会通过 VPN 把流量送到 **10.0.0.0/8**。

4.7.9. 附加资源

接下来的信息源会提供关于 **LibreSwan** 以及 **ipsec** 后台程序的额外资源。

4.7.9.1. 已安装的文档

- **ipsec(8)** 手册页——为 **ipsec** 描述命令选项。
- **ipsec.conf(5)** 手册页 – 包含配置 **ipsec** 的信息。
- **ipsec.secrets(5)** 手册页—包含配置**ipsec** 的信息。
- **ipsec_auto(8)** man page —描述 **auto**的命令行客户端的命令选项，以操作自动键入的 **LibreSwan IPsec** 链接。
- **ipsec_rsasigkey(8)** 手册页 – 描述生成 RSA 签名秘钥的工具。
- **/usr/share/doc/libreswan-*version*/README.nss**——描述用于原始 RSA 秘钥的命令及使用 **Libreswan pluto** 程序的加密库的证书。

4.7.9.2. 在线文档编制

https://libreswan.org

上游项目的网站。

http://www.mozilla.org/projects/security/pki/nss/

网络安全服务（Network Security Services）项目

4.8. 加密

4.8.1. 使用 LUKS 硬盘加密

磁盘格式的 Linux 统一密钥设置（或称为 LUKS）可让您加密 Linux 计算机中的分区。这对可移动计算机以及可移动介质尤为重要。LUKS 可允许使用多个用户密钥解密用于分区批加密的主密钥。

LUKS 概要

LUKS 能做什么

- LUKS 能对全区设备加密，因此，非常适用于保护移动设备的内容，如可移动的储存媒体或笔记本电脑硬盘驱动器。
- 加密区设备的基本内容可为任意内容。这对于加密 **swap** 设备十分有用。这对用于特定的格式化区设备数据储存的某些数据库也是很有用的。
- LUKS 使用现有设备映射器的内核子系统。
- LUKS 密码短语增强，可提供防止字典攻击。
- LUKS 设备含有多个密钥槽，允许用户添加备用密钥或密码短语。

LUKS "*不能*" 做的是：

- LUKS 不适用于需要很多（超过 8 个）用户对同一设备有不同访问密钥的程序。
- LUKS 不适用于需要文件级别加密的程序。

4.8.1.1. 红帽企业版 Linux 中的 LUKS 实施

红帽企业版 Linux 6 采用 LUKS 执行文件系统加密。默认情况下不会在安装过程中选择加密文件系统的选项。如果您选择该选项加密您的硬盘，则每次您引导计算机时都会提示您输入密码短语。这个密码短语可为您用于分区解密的批加密密钥"解锁"。如果您选择要修改默认分区表，则您可以选择您要加密的分区。这是在分区表设置值设定的。

用于 LUKS （请参阅 **cryptsetup --help**）的默认密码是 aes-cbc-essiv:sha256 （ESSIV - Encrypted Salt-Sector Initialization Vector，加密密钥的 hash，是 Linux 系统中 dm-crpt 默认使用的 IV ）。请注意，默认情况下，在 XTS 模式（aes-xts-plain64）下使用此安装程序 **Anaconda** 。 LUKS 的默认密钥长度为 256 位。 LUKS **Anaconda** （XTS 模式） 的默认密钥长度为 512 位。 可用的密码为：

- AES - 高级加密标准 - 〈FIPS PUB 197〉
- Twofish（128 位块密码）
- Serpent
- cast5 - 〈 RFC 2144〉
- cast6 - 〈 RFC 2612〉

4.8.1.2. 手动加密目录

警告

按照这个步骤执行将删除您要加密的分区中的所有数据。您将会丢失所有信息！在开始执行这个步骤前，请确保您在外部信源中备份了数据。

1. 作为 root 用户，用 shell 提示符输入以下命令，进入运行等级 1：

```
telinit 1
```

2. 卸载您现有的 **/home**:

```
umount /home
```

3. 如果上一步的命令失败，那么使用 **fuser** 来查找占用 **/home** 的程序并将其终止：

```
fuser -mvk /home
```

4. 检验是否还有装载 **/home** ：

```
grep home /proc/mounts
```

5. 将随机数据填入您的分区：

```
shred -v --iterations=1 /dev/VG00/LV_home
```

此命令会以您设备的顺序写入速度执行，可能要花些时间才能完成。它是确保未经加密的数据不会保留在使用过的设备上，并混淆部分含有加密数据却又不是随机数据的设备。

6. 初始化您的分区：

```
cryptsetup --verbose --verify-passphrase luksFormat
/dev/VG00/LV_home
```

7. 打开新加密的设备：

```
cryptsetup luksOpen /dev/VG00/LV_home home
```

8. 确保设备存在：

```
ls -l /dev/mapper | grep home
```

9. 创建文件系统：

```
mkfs.ext3 /dev/mapper/home
```

10. 装载文件系统：

```
mount /dev/mapper/home /home
```

11. 确保文件系统可见：

```
df -h | grep home
```

12. 添加以下命令到 **/etc/crypttab** 文件：

```
home /dev/VG00/LV_home none
```

13. 编辑 **/etc/fstab** 文件，移除 **/home** 旧的入口，并添加以下命令行：

```
/dev/mapper/home /home ext3 defaults 1 2
```

14. 恢复默认的 SELinux 安全环境：

```
/sbin/restorecon -v -R /home
```

15. 重启机器：

```
shutdown -r now
```

16. **/etc/crypttab** 条目在启动时，您的电脑会询问您的 **luks** 密码短语。
17. 作为 root 用户登录，并恢复您的备份。

现在您就有一个加密的分区，即便电脑处于关机的状态，可安全地放置您所有的数据。

4.8.1.3. 为现有设备添加新密码短语

使用以下命令，添加新的密码短语到现有的设备：

```
cryptsetup luksAddKey <device>
```

提示您输入现有密码短语进行验证后，将提示您输入新密码短语。

4.8.1.4. 从现有的设备中移除密码短语

使用以下命令，从现有设备中移除密码短语：

```
cryptsetup luksRemoveKey <device>
```

将提示您要删除的密码短语，然后是剩下用来验证的任意密码短语。

4.8.1.5. 在 Anaconda 中创建加密块设备

您可以在系统安装过程中创建加密块设备。这可允许您轻松使用加密分区配置系统。

要对块设备加密，则须在选择自动分区时检查 **加密系统** （Encrypt System）复选框，或在创建独立分区、软件 RAID （Redundant Arrays of independent Disks，磁盘阵列）阵列或逻辑卷时，勾选 **加密（Encrypt）** 复选框。在您完成分区之后，系统会提示您输入加密的密码短语。要求此密码短语可访问加密设备。如果您有预先存在的 LUKS 设备，并有其在早期安装过程中所提供的密码短语，那么输入密码短语对话框中将也会含有复选框。检查此复选框表明，您将新的密码短语添加到预先存在的加密块设备中每个可用的槽。

注意

在 **自动分区** 屏幕上检查（Automatic Partitioning） **加密系统** （Encrypt System）复选框，然后选择 **创建自定义分区** （Create custom layout），这样就不会引起任何块设备进行自动加密。

注意

您可使用 **kickstart** 为每个新加密的块设备设置单独的密码。

4.8.1.6. 附加资源

关于 LUKS 或是在 Red Hat Enterprise Linux 7 下加密硬盘的其他信息，请访问以下链接：

- LUKS home page
- LUKS/cryptsetup FAQ
- LUKS - Linux Unified Key Setup Wikipedia article
- HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive and pvmove

4.8.2. 创建 GPG 密钥

GPG （GNU Privacy Guard，GNU 隐私卫士）用于识别您的身份，并对您的通信进行身份验证，包括那些您不认识的人。 GPG 允许任何通过读取 GPG 签名邮件来验证其身份的人使用。换句话说， 对于某些十分确定您所签名的通信实际上就是来源于您， GPG 允许那些人使用。 GPG 有用是因为它能防止第三方更改编码或中途拦截对话，更改信息。

4.8.2.1. 在 GNOME 中创建 GPG 密钥

要在 GNOME 中创建 GPG 密钥，则须遵循这些步骤：

1. 安装 **海马（Seahorse）** 实用程序，更易于 GPG 密钥的管理：

```
~]# yum install seahorse
```

2. 要创建密钥，则须从 **应用程序（Applications）** → **辅助程序（Accessories）** 菜单，选择 **密码和加密密钥（Passwords and Encryption Keys）**，这就启动 **Seahorse** 应用程序。

3. 从 **文件（File）** 菜单中选择 **新文件（New）** ，再选 **PGP 密钥（PGP Key；Pretty Good Privacy，加密软体）** 。然后点击 **继续（Continue）** .

4. 输入您的全名、电子邮箱以及可用于描述您的选择性注释（例如：约翰· C ·史密斯（ John C. Smith），jsmith@example.com， 软件工程师（Software Engineer））。点击 **创建（Create）** 。然后会出现对话框，要求输入密钥的密码短语。选择一个强大又容易记的密码短语。点击 **OK** ，密钥就创建好了。

警告

如果您忘记了您的密码短语，那么您将无法解码数据。

要查找您的 GPG 密钥 ID，则须在新创建密钥旁的 **密钥 ID（Key ID）** 栏中查找。在大多数情况下，如果要求您输入密钥 ID ，那么在密钥 ID 之前加入 **0x** ，如 **0x6789ABCD** 。您应当备份您的私钥，并保存在安全的地方。

4.8.2.2. 在 KDE 中创建 GPG 密钥

要在 KDE（桌面环境）中创建 GPG 密钥，则须遵循这些步骤：

1. 从主菜单中选择 **应用程序（Applications）** → **实用程序（Utilities）** → **加密工具（Encryption Tool）** ，启动 KGpg 程序。如果您从未使用过 KGpg 程序，那么此程序会指导您完成创建 GPG 密钥对的过程。
2. 会出现对话框，提示您创建新的密钥对。输入您的姓名、电子邮箱以及选择是否添加注释。您也可为您的密钥选择有效期，以及密钥强度（位数）和算法。
3. 在下一个对话框输入您的密码短语。此时，您的密钥会出现在 **KGpg** 的主窗口。

警告

如果您忘记了您的密码短语，那么您将无法解码数据。

要查找您的 GPG 密钥 ID，则须在新创建密钥旁的 **密钥 ID（Key ID）** 栏中查找。在大多数情况下，如果要求您输入密钥 ID ，那么在密钥 ID 之前加入 **0x** ，如 **0x6789ABCD** 。您应当备份您的私钥，并保存在安全的地方。

4.8.2.3. 使用命令行创建 GPG 密钥

1. 使用以下 shell 命令：

```
~]$ gpg2 --gen-key
```

这个命令生成由公钥和私钥组成的密钥对。其它人可使用您的公钥认证和（或者）解密您的会话。尽量广泛发布您的公钥，特别是对那些您知道要从您哪里接收认证会话的人，比如邮件列表。

2. 一系列的提示会指导您完成此过程。若需要，按下 **回车** （Enter）键，赋予默认值。第一个提示会询问您要选择怎样的密钥：

```
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection?
```

在几乎所有情况下，默认值都是正确的选择。RSA / RSA 密钥不仅允许您对通信签名，还允许您加密文件。

3. 选择密钥长度：

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

默认值 2048 位，对于几乎所有的用户来说，都是足够的，而且有着极强的安全级别。

4. 选择密钥的有效期。选择有效期，而不是使用 **none** 的默认值，这是很好的想法。例如，如果密钥中的电子邮箱无效了，那么有效期将提醒其他人停止使用此公钥。

```
Please specify how long the key should be valid.
0 = key does not expire
d = key expires in n days
w = key expires in n weeks
m = key expires in n months
y = key expires in n years
key is valid for? (0)
```

例如，输入一个 **1y** 值，会使密钥的有效期为一年。（如果您改变主意的话，您可以在密钥生成之后更改其有效期。）

5. 在 **gpg2** 应用程序询问签名信息之前，则会出现以下提示：

```
Is this correct (y/N)?
```

输入 **y** 完成此程序。

6. 为您的 GPG 密钥输入您的姓名和电子邮箱。记住此程序是要验证您的个人真实身份。因此，包括您的真实姓名。如果您选择假的电子邮箱，那么其他人要找到您的公钥就更困难了。这会使您的通信身份验证很难进行。例如，如果您在邮件列表中将此 GPG 密钥用于您的个人介绍，那么在此列表中输入您使用的电子邮箱。

使用注释字段添加别名或者其它信息。（有些人为不同目的使用不同的密钥，并使用注释互相识别，比如“Office”或者“Open Source Projects”。）

7. 在确认的提示信息中，如果所有输入都是正确的，请输入 **O** 字母；或者使用其他选择来解决任何问题。最后，为您的安全密钥输入密码短语。 **gpg2** 程序会要求您输入两次密码短语，以确保您没有输入错误。

8. 最后， **gpg2** 会产生随机数据，以尽可能地确保您的密钥是独一无二的。移动您的鼠标，输入随机密钥，或在系统运行此步骤期间执行其他任务来加速此进程。一旦完成此步骤，您的密钥就生成完毕，可以使用：

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A
FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

9. 密钥指纹是您密钥的简写“签名”。它允许您确认其他人是否接收过您的真实公钥，有没有进行篡改。您不需要写下此指纹。在任何时间要显示此指纹，则须使用此命令，替换您的电子邮箱：

```
~]$ gpg2 --fingerprint jqdoe@example.com
```

您的 “GPG key ID” 由 8 个十六进制数字组成，用于识别公钥。在上述示例中， GPG 密钥 ID 是 **1B2AFA1C** 。在大多数情况下，如果要求您输入密钥 ID ，那么在密钥 ID 之前加入 **0x** ，如 **0x6789ABCD** 。

警告

如果您忘记了您的密码短语，则该密钥就无法使用，且使用该密钥加密的数据将会丢失。

4.8.2.4. 有关公钥加密

1. Wikipedia - Public Key Cryptography
2. HowStuffWorks - Encryption

4.8.3. 在公钥密码学中使用 openCryptoki

openCryptoki 是一个 Linux 下的 *PKCS#11* 开源实现，是一种“*公钥加密标准*”（PKCS，Public-Key Cryptography Standard），定义了通常称为令牌的加密设备的应用程序接口（API）。令牌可在硬件或软件中执行。此特点概述了 **openCryptoki** 系统是如何安装、配置，以及如何在 Red Hat Enterprise Linux 7 中使用。

4.8.3.1. 安装 openCryptoki 并启动服务

要在您的系统中安装 **openCryptoki** 基本工具包，包括用于检测的令牌的软件实现，则须作为 **root** 用户运行以下命令：

```
~]# yum install opencryptoki
```

根据您打算使用的硬件令牌的类型，您可能需要安装其他工具包以支持具体使用情况。例如，要获取对“*可信计算平台模块*”（TPM，Trusted Platform Module）设备的支持，您需要安装 *opencryptoki-tpmtok* 工具包。

关于如何使用 **Yum** 工具包管理器安装工具包的基本信息，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》。

要启用 **openCryptoki** 服务，您需要运行 **pkcsslotd** 守护进程。作为 **root** 用户执行以下命令，就可启动当前会话的守护进程：

```
~]# systemctl start pkcsslotd
```

要确保在启动时可自动启用服务，则须运行以下命令：

```
~]# systemctl enable pkcsslotd
```

关于如何使用 systemd 来管理服务的更多信息，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》。

4.8.3.2. 配置并使用 openCryptoki

启动时，**pkcsslotd** 守护进程会读取**/etc/opencryptoki/opencryptoki.conf** 配置文件，它可用于收集关于在系统工作时所配置的令牌以及其槽的信息。

此文件定义了使用键值对的独立槽。每个槽的定义可含有描述、可使用的令牌库的规格说明书，以及槽制造者的 ID 。另外，可对槽的硬件和固件的版本进行定义。关于文件格式的描述以及独立键和可分配给这些键的值的详细描述，请参阅 opencryptoki.conf(5) 手册页。

要在运行时修改 **pkcsslotd** 守护进程的行为，则须使用 **pkcsconf** 实用程序。此工具允许您显示和配置守护进程的状态，以及列出并修改当前所配置的槽和令牌。例如，要显示关于令牌的信息，则须发出以下命令（请注意， 所有需用 **pkcsslotd** 守护进程进行通信的非 root 用户必须是 **pkcs11** 系统组的成员）：

```
~]$ pkcsconf -t
```

关于可用于 **pkcsconf** 工具的参数列表，请参阅 pkcsconf(1) 手册页。

警告

请牢记，只有完全可信的用户应可成为 **pkcs11** 组的成员，因为此组的所有成员有权限阻止其他 **openCryptoki** 服务的用户访问所配置的 PKCS#11 令牌。

第 5 章 系统审核

Linux 审核系统为追踪系统中与安全相关的信息提供了途径。基于预配置原则，审核将生成日志项从而记录尽可能多的在系统中发生的事件。这一信息对执行关键任务的环境尤其重要，它可以确定那些违反安全策略的人以及他们的行为。审核不会为系统提供额外的安全保护；相反，它能用来发现系统中违反安全策略的行为。通过额外的措施例如 SELinux 可以进一步地防止这些违反行为。

下面的列表总结了一些信息有关审核能够记录的日志文件：

- 日期和时间，类型，以及事件结果。
- 主题和对象的敏感性标签。
- 事件关联与触发事件的用户身份。
- 所有对审核配置的修改以及尝试访问审核日志文件。
- 所有认证机制的使用，例如 SSH、Kerberos、以及其他。
- 对于任何信任数据库的改变，例如 **`/etc/passwd`**.
- 尝试把信息输入系统，或者从系统中输出信息
- 包含或者排除以用户身份，主题和对象标签以及其他属性为基础的事件

使用审核系统也要求与安全有关的认证。设计审核是为了能满足甚至超过以下认证或者服从指南的要求：

- 受控制访问保护文件（CAPP）
- 卷标式安全保护设定文件（LSPP）
- 基于规则集的访问控制（RSBAC）
- 国家工业安全计划操作手册（NISPOM）
- 联邦信息安全管理法案（FISMA）
- 支付卡行业数据安全标准 (PCI-DSS)
- 安全技术实施指南 (STIG)

审核也可以是：

- 由国家信息安全保障联盟（NIAP）以及最佳安全行业（BSI）评估
- 红帽企业版 Linux 5 通过 LSPP/CAPP/RSBAC/EAL4+ 认证。
- 红帽企业版 Linux 6 通过操作系统保护文件/评估保障等级4+ (OSPP/EAL4+) 认证。

用例

访问监测文件

审核能够追踪是否有人访问、修改或者运行某个文件或者目录，或者是否更改了文件属性。这是很有用的，例如检测访问重要文件以及备有审计记录以防其中的某个文件被破坏。

调用监测系统

每次使用特定的系统调用时，配置审核来生成日志项。例如，这可以通过监测来追踪系统中的变化 **`settimeofday`**、**`clock_adjtime`** 和其他与时间相关的系统调用。

用户记录指令运行

因为审核可以追踪该文件是否被运行，因此设定许多规则来记录每一个执行过的特定指令。例如，为每一个可执行的 **/bin**目录设定规则。通过用户的身份可以搜寻所产生的日志项从而生成每一位用户所执行指令的审计记录。

记录安全事件

pam_faillock 认证模块能够记录失败的登录尝试，也可以通过建立审核来记录失败的登录尝试，并提供有关尝试登录用户的额外信息。

查找事件

审核提供 **ausearch** 实用程序，这被用来筛选日志项，并且提供基于许多情况的审计记录。

运行总结报告

aureport 实用程序此外还可以被用来生成所记录的事件的日常报告。系统管理员能够分析报告并且进一步调查可疑活动。

监测网络访问

iptables 以及 **ebtables** 实用程序可以被配置用来触发审核事件，允许系统管理员监测网络访问。

注意

审核所收集的信息量可能影响系统性能。

5.1. 审核系统架构

审核系统包含两个主要部分：用户空间的应用程序、实用程序，以及 kernel-side 系统调用处理。Kernel 的组件从用户空间的应用程序接受系统调用，并且通过三个过滤器中的一个过滤器来进行筛选： *user*、*task* 或者 *exit*。一旦系统调用通过其中的一个过滤器，就将通过 *exclude* 过滤器进行传送，这是基于审核规则的配置，并把它传送给审核的守护程序做进一步的处理。图 5.1 “审核系统架构” 说明这一过程。

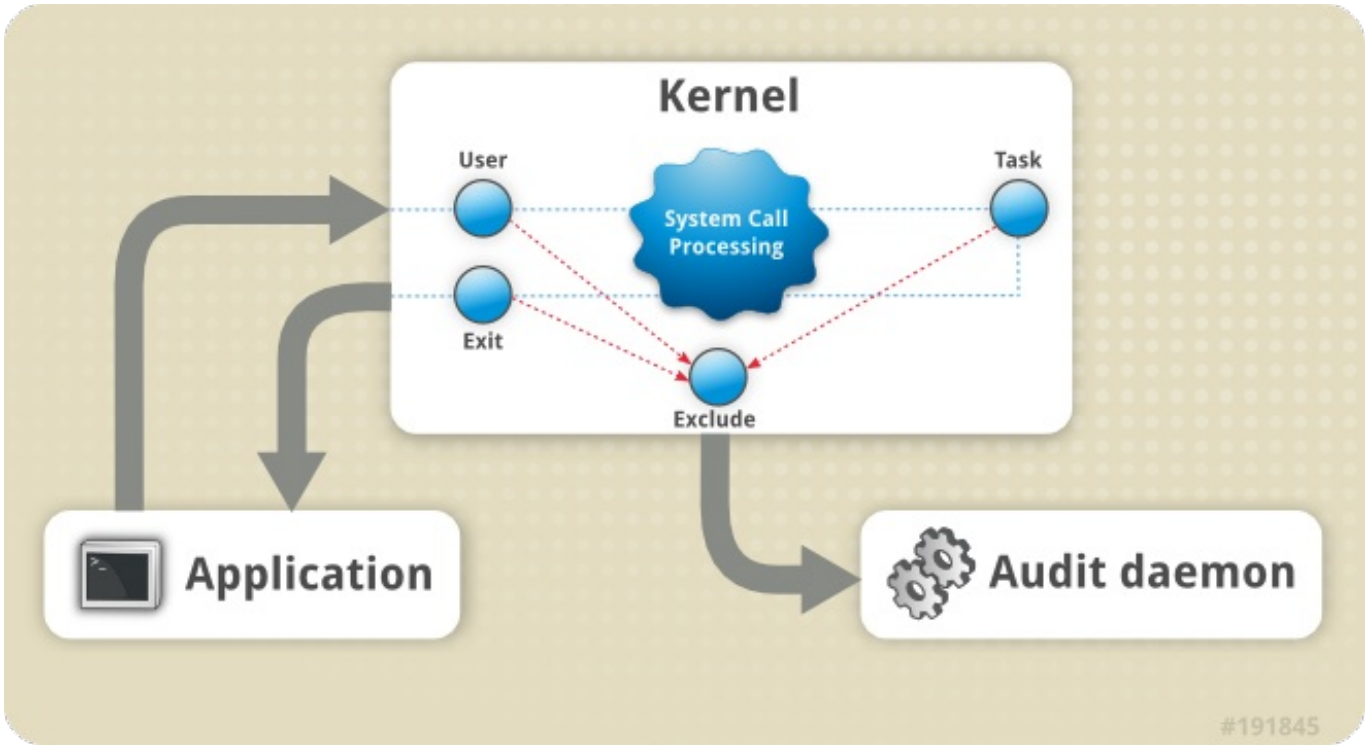

图 5.1. 审核系统架构

用户空间的审核守护进程收集来自于 Kernel 的信息，并在日志文件中创造日志文件项。其他审核用户空间的实用程序与审核守护进程进行信息交互，Kernel 审核组件，或者审核日志文件：

- **audisp** — 审核调度守护进程与审核守护进程进行交互，把事件传送给其他应用程序做进一步处理。守护进程的目的是为了提供插入机制，这样的话实时分析程序就能与审核事件进行交互。
- **auditctl** — 审核控制实用程序与 Kernel 审核组件进行交互来控制生成事件过程的许多设定和参数。
- 剩余的审核实用程序把审核日志文件内容作为输入信息，并基于用户要求生成输出信息。例如，**aureport** 实用程序生成所有记录事件的报道。

5.2. 安装 *audit* 软件包

为了使用审核系统，您必须在系统中安装 *audit* 软件包。*audit* packages（*audit* 及 *audit-libs*）是默认安装在红帽企业版 Linux 6 中。如果您不想要安装这些软件包，作为 root 用户执行以下命令来安装。

```
~]# yum install audit
```

5.3. 配置 `audit` 服务

审核守护程序可以在 **/etc/audit/auditd.conf** 配置文件中进行配置。这个文件包括修改审核守护进程特性的配置参数。紧跟 # 字符（#）的任何空行或者文本都被忽略。所有配置参数的列表以及它们的解释都可以在 audit.conf(5) 手册页中找到。

5.3.1. 为了 在 CAPP 环境配置 `auditd`

默认 **auditd** 配置应该对大多数环境都适合。但是如果您的环境符合由 *可控制存取保护档案*（CAPP）所建立的标准，这将是公共标准认证的一部分，审核守护程序必须用以下设定配置：

- 保存审核日志文件的目录（**/var/log/audit/**）经常应该在另一个分区。这将防止其他过程耗费此目录中的空间，并且为剩余的审核守护程序提供准确的检测。
- ***max_log_file*** 参数详细说明了每个审核日志文件最少的占用空间，参数必须设定为充分利用保存审核日志文件分区所在的可用空间。
- ***max_log_file_action*** 参数决定采取何种行动，一旦到达在 ***max_log_file***中所设定的极限，则应该设定为 **keep_logs** 防止审核日志文件被重写。
- ***space_left*** 参数明确说明磁盘中可用空间的数量，这样的话在***space_left_action*** 参数中所设定的行动会被触发。此参数必须被设定为一个数字它会给予管理者足够的时间来回应和刷新磁盘空间。***space_left*** 价值取决于审核日志文件生成的速度。
- 我们推荐您采用合适的通知方法把 ***space_left_action*** 参数设定为**email** 或者 **exec**。
- ***admin_space_left*** 参数明确说明自由空间的绝对最小数量，为了在 ***admin_space_left_action*** 参数中所设定的行动会被触发，必须设定一个会给予管理者的日志行动总够空间的值。
- ***admin_space_left_action*** 参数必须设定 **single** 使系统属于单一用户模式，并且允许管理者开放一些磁盘空间。
- ***disk_full_action*** 参数明确说明当保存审核日志文件的分区没有可用空间时，应该触发行动，并且必须设定为 **halt** 或者 **single**。这保障了当审核不再记录事件时，系统也能在单一用户模式下关闭或者运行。
- ***disk_error_action***，明确说明如果保存在审核日志文件的分区检测到错误时，应该采取行动，必须设定 **syslog**、**single** 或者 **halt**，这取决于当地的安全政策有关硬件故障的处理。
- ***flush*** 配置参数必须设定为 **sync** 或者 **data**。这些参数保证所有的审核事件数据能与磁盘中的日志文件同步。

剩余的配置选择应该根据当地安全政策建立。

5.4. 开始 audit 服务

一旦 **auditd** 进行适当配置，就可以开始服务来收集审核信息，并在日志文件中储存。作为 root 用户来开始执行以下指令 **auditd**：

```
~]# service auditd start
```

您可以可选择性地配置 **auditd**，作为 root 用户在启动事件开始使用以下指令：

```
~]# chkconfig auditd on
```

在 **auditd**上可以执行一些其他的行动， 使用 **service auditd *action*** 命令，*action* 可能是以下其中之一：

- **stop** — 停止 **auditd**。
- **restart** — 重启**auditd**。
- **reload** 或者 **force-reload** — 重新加载 **auditd** 在 **/etc/audit/auditd.conf** 文件中的配置。
- **rotate** — 在 **/var/log/audit/** 目录中旋转日志文件。
- **resume** — 在推迟审核事件日志之后重新开始，例如存在没有足够的磁盘分区空间来保存审核日志文件情况。

- **condrestart** 或者 **try-restart** — 只有当它已经在运行时，重启 **auditd** 。
- **status** — 显示运行状态 **auditd**。

5.5. 定义审核规则

审核系统根据一组规则运行，这组规则定义了日志文件中所获取的内容。有三种类型的审核规则可以详细说明：

- 控制规则 — 允许审核系统的行为和它的一些被修改的配置。
- 文件系统规则 — 也被称为文件监视，允许审核进入特定文件或者目录。
- 系统调用规则 — 允许记录任何指定程序所做的系统调用。

审核规则可以在命令行上使用 **auditctl** 实用程序进行详细说明（请注意这些规则并不是在重新启动时一直有效），或者写在 **/etc/audit/audit.rules** 文件中。以下两个部分总结了定义审核规则的两个方法。

5.5.1. 使用 auditctl 实用程序来定义审核规则

注意

所有与审核服务交互的命令以及审核日志文件都需要 root 特权。作为 root 用户确保您执行这些命令。

auditctl 命令允许您控制审核系统的基本功能并且限定规则来决定哪些审核项目要记录。

定义控制规则

以下是一些控制规则允许您修改审核系统的行为：

-b

在 Kernel 中设定最大数量的已存在的审核缓冲区，例如：

```
~]# auditctl -b 8192
```

-f

当追踪重要错误时设定所要完成的行动，例如：

```
~]# auditctl -f 2
```

以上配置触发 kernel 恐慌以防重要错误。

-e

启动或者禁用审核系统或者锁定它的配置，例如：

```
~]# auditctl -e 2
```

以上命令锁定审核配置。

-r

设定每秒生成信息的速率，例如：

```
~]# auditctl -r 0
```

以上配置在生成信息方面不设定限制速率。

-s

报告审核系统状态，例如：

```
~]# auditctl -s
AUDIT_STATUS: enabled=1 flag=2 pid=0 rate_limit=0
backlog_limit=8192 lost=259 backlog=0
```

-l

列出所有当前装载的审核规则，例如：

```
~]# auditctl -l
LIST_RULES: exit,always watch=/etc/localtime perm=wa key=time-
change
LIST_RULES: exit,always watch=/etc/group perm=wa key=identity
LIST_RULES: exit,always watch=/etc/passwd perm=wa key=identity
LIST_RULES: exit,always watch=/etc/gshadow perm=wa key=identity
⋮
```

-D

删除所有当前装载的审核规则，例如：

```
~]# auditctl -D
No rules
```

定义文件系统规则

定义文件系统规则，使用以下语法：

```
auditctl -w path_to_file -p permissions -k key_name
```

其中：

- *path_to_file* 是审核过的文件或者目录：
- *permissions* 是被记录的权限：
  - **r** — 读取文件或者目录。
  - **w** — 写入文件或者目录。
  - **x** — 运行文件或者目录。
  - **a** — 改变在文件或者目录中的属性。
- *key_name* 是可选字符串，可帮助您判定哪个规则或者哪组规则生成特定的日志项。

例 5.1. 文件系统规则

为了定义所有的输写访问权限以及在 **/etc/passwd** 文件中每个属性更改的规则，执行以下命令：

```
~]# auditctl -w /etc/passwd -p wa -k passwd_changes
```

请注意以下字符串 **-k** 选项是任意的。

为了定义记录所有输写访问权限，以及在 **/etc/selinux/** 目录中所有文件属性更改的规则，执行以下命令：

```
~]# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

为了定义可以记录执行 **/sbin/insmod** 命令的规则，在 Linux Kernel 中插入模块，执行以下命令：

```
~]# auditctl -w /sbin/insmod -p x -k module_insertion
```

定义系统调用规则

为了定义系统调用规则，使用以下语法：

```
auditctl -a action,filter -S system_call -F 输入栏=value -k key_name
```

其中：

- *action* 以及 *filter* 详细说明某个事件何时被记录。 *action* 可能是 **always**（经常是）或者**never**（从不是）其中之一。 *filter* 详细说明哪个 Kernel 规则匹配过滤器应用在事件中。以下是其中之一的与规则匹配的过滤器： **task**、**exit**、**user** 以及 **exclude**。如果想要更多有关这些过滤器的信息，请参考〈第 5.1 节 “审核系统架构”〉的开始部分。
- *system_call* 通过它的名字详细说明系统调用。所有的系统调用都可以在**/usr/include/asm/unistd_64.h** 文件中找到。许多系统调用都能形成一个规则，每个都在 **-S** 选项之后详细说明。
- *field=value* 详细说明其他选项，进一步修改规则来与以特定架构、组 ID、进程 ID和其他内容为基础的事件相匹配。为了列出完整可用的输入栏类型和它们的数值，请参考 auditctl(8) 手册页。
- *key_name* 是可选字符串，可帮助您判定哪个规则或者哪组规则生成特定的日志项。

例 5.2. 系统调用规则

为了定义创造日志项 的规则，每次通过程序使用系统调用 **adjtimex** 或者 **settimeofday**。当系统使用 64 位架构，请执行以下命令：

```
~]# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -
k time_change
```

为了定义创造日志项的规则，每次由 ID 是 500 或更大的系统用户删除或者重命名文件时，使用（**-F auid!=4294967295** 选项排除没有设定登录 UID的用户），执行以下命令：

```
~]# auditctl -a always,exit -S unlink -S unlinkat -S rename -S
renameat -F auid>=500 -F auid!=4294967295 -k delete
```

使用系统调用语法来定义文件系统也是有可能的。对于 **-w /etc/shadow -p wa** 文件系统规则来说，以下命令为模拟的系统调用创造了规则：

```
~]# auditctl -a always,exit -F path=/etc/shadow -F perm=wa
```

5.5.2. 在 /etc/audit/audit.rules 文件中定义持久的审核规则和控制

为了定义在重新启动时可以一直有效的审核规则，您必须把它们包含在 **/etc/audit/audit.rules** 文件中。这个文件使用相同的 **auditctl** 命令行语法来详细说明规则。任何在 # 之后的空行或者文本（#）可以忽略。

auditctl 指令可以被用来读取来自指定文件的规则，使用 **-R** 选项，例如：

```
~]# auditctl -R /usr/share/doc/audit-version/stig.rules
```

定义控制规则

文件可以只包括以下的控制规则，修改审核系统的行为：**-b**、**-D**、**-e**、**-f**、或者 **-r**。如果想获取更多信息，请参考 第 5.5.1 节 “定义控制规则”。

例 5.3. 在 audit.rules中控制规则。

```
# Delete all previous rules
-D

# Set buffer size
-b 8192

# Make the configuration immutable -- reboot is required to change
audit rules
-e 2

# Panic when a failure occurs
-f 2

# Generate at most 100 audit messages per second
-r 100
```

定义文件系统和系统调用规则

使用 **auditctl** 语法定义文件系统和系统调用原则。在〈第 5.5.1 节 “使用 auditctl 实用程序来定义审核规则”〉中的例子可以用以下规则文件来表示：

例 5.4. 在 audit.rules 中的文件系统和系统调用规则

```
-w /etc/passwd -p wa -k passwd_changes
-w /etc/selinux/ -p wa -k selinux_changes
-w /sbin/insmod -p x -k module_insertion
```

```
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
-a always,exit -S unlink -S unlinkat -S rename -S renameat -F
auid>=500 -F auid!=4294967295 -k delete
```

预配置规则文件

在 **/usr/share/doc/audit-*version*/** 目录中, 根据不同的认证标准 *audit* 软件包提供一组预配置规则文件：

- **nispom.rules** — 审核规则配置符合《国家行业安全程序操作运行指南》的第八章中详细说明的要求。
- **capp.rules** — 审核规则配置满足由 CAPP 设定的要求，是公共标准认定的一部分。
- **lspp.rules** —审核规则配置满足由 LSPP 设定的要求是公共标准认定的一部分。
- **stig.rules** — 审核规则配置满足由 STIG 所设定的要求。

为了使用这些配置文件，需要创造您原始文件的备份 **/etc/audit/audit.rules** 并且复制您所选择的有关 **/etc/audit/audit.rules** 文件的配置文件:

```
~]# cp /etc/audit/audit.rules /etc/audit/audit.rules_backup
~]# cp /usr/share/doc/audit-version/stig.rules /etc/audit/audit.rules
```

5.6. 理解审核日志文件

默认情况下，在 **/var/log/audit/audit.log** 文件中的审核系统储存日志项；如果启用日志旋转，就可以旋转储存在同一目录中的 **audit.log** 文件。

以下的审核规则记录了每次读取或者修改 **/etc/ssh/sshd_config** 文件的尝试：

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

如果 **auditd** 守护程序在运行，就需在审核日志文件中运行以下命令创造新事件：

```
~]# cat /etc/ssh/sshd_config
```

在 **audit.log** 文件中的事件如下所示：

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2
success=no exit=-13 a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1
ppid=2686 pid=3538 auid=500 uid=500 gid=500 euid=500 suid=500 fsuid=500
egid=500 sgid=500 fsgid=500 tty=pts0 ses=1 comm="cat" exe="/bin/cat"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="sshd_config"
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0
name="/etc/ssh/sshd_config" inode=409248 dev=fd:00 mode=0100600 ouid=0
ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
```

以上事件由三个记录组成（每个以 **type=** 密码作为开始），共享相同的时间戳和编号。每个记录包含好几对 ***name=value*** ，由空格或者逗号分开。以下是关于以上事件的详细分析：

第一个记录

type=SYSCALL

type 输入栏包含这类记录。在这个例子中， **SYSCALL** 数值详细说明连接到 Kernel 的系统调用触发了这个记录。

为了列出所有可能的类型值和它们的解释，请参考〈第 B.2 节 “审核记录类型”〉。

msg=audit(1364481363.243:24287):

msg 输入栏记录：

- **audit(*time_stamp*:*ID*)** 表格中记录的时间戳和特殊 ID。如果多种记录生成为相同审核事件的一部分，那么它们可以共享相同的时间戳和 ID。
- Kernel 或者用户空间应用提供不同的事件特定 ***name=value*** 组。

arch=c000003e

arch 输入栏包括关于系统CPU架构的信息。值 **c000003e** 是使用 16 进制表示法编码。当使用 **ausearch** 命令搜寻审核记录时, 使用 **-i** 或者 **--interpret** 选项自动转化为 16 进制值可供人读取的对等语。 **c000003e** 值被解释为 **x86_64**。

syscall=2

syscall 输入栏记录了传输给 Kernel 的输入栏类型。值 **2** 可以与在 **/usr/include/asm/unistd_64.h** 文件中可供人读取的对等语相匹配。 在这种情况下，**2** 是 **open** 系统调用。 请注意 **ausyscall** 实用程序允许您把系统调用数字转换成可供人读取的对等语。 使用 **ausyscall --dump** 命令来展示所有的系统调用和它们的号码。如想要获取更多信息，请参考 ausyscall(8) 手册页。

success=no

success 输入栏记录了系统调用是否被成功地记录在特定事件中。在这种情况下，调用不会成功。

exit=-13

exit 输入栏包含详细说明由系统调用所返回的退出代码的值。在不同的系统调用中，值各不相同。您可以用以下命令把值解释为可供人读取的对等语：**ausearch --interpret --exit -13**（假设您的审核日志中包含的事件没有退出代码 **-13**）。

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

a0 到 **a3** 输入栏记录了前四个参数，在这个事件中使用 16 进制编码系统调用。这些参数取决于使用的系统调用；它们可以通过 **ausearch** 实用程序来解释。

items=1

items 输入栏包含事件中路径记录的数量。

ppid=2686

items 输入栏记录了父进程 ID（PPID）。在这个情况下，**2686** 是 **bash** 进程的 PPID。

pid=3538

pid 输入栏记录了进程 ID（PID）。在这个情况下，**3538** 是 **cat** 进程的 PID。

auid=500

auid 输入栏记录了审核用户 ID，这个是 loginuid。这个ID是用户在登录时使用的并且即使当用户身份改变时，也可以通过每个进程获取该ID。（例如，通过切换用户账户，使用 **su - john**命令）。

uid=500

uid 输入栏记录了开始分析进程的用户 ID。使用以下指令：**ausearch -i --uid *UID***，用户 ID 就可以被解释为用户名字。在这个情况下， **500** 是 **shadowman** 的用户 ID。

gid=500

gid 输入栏记录了开始分析进程用户的 ID组。

euid=500

euid 输入栏记录了开始分析进程用户的有效用户 ID。

suid=500

suid 输入栏记录了开始分析进程的用户的设置用户 ID。

fsuid=500

fsuid 输入栏记录了开始分析进程用户的文件系统用户 ID。

egid=500

egid 输入栏记录了开始分析进程用户的有效群组 ID。

sgid=500

sgid 输入栏记录了开始分析进程用户的设置群组 ID。

fsgid=500

fsgid 输入栏记录了开始分析进程的用户的文件系统群组 ID。

tty=pts0

tty 输入栏记录了调用分析进程的终端。

ses=1

ses 输入栏记录了调用分析进程会话的会话 ID。

comm="cat"

comm 输入栏记录了命令行的名字，它被用于调用分析进程。在这种情况下 **cat** 命令被用来触发审核事件。

exe="/bin/cat"

exe 输入栏记录了被用来调用分析进程的可执行的路径。

subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

subj 输入栏记录了 SELinux 语境，运用此语境可以在执行时间中标注分析进程。

key="sshd_config"

key 输入栏记录了所有管理者定义的字符串，它与在审核日志中生成该事件的规则有关。

第二记录

type=CWD

在第二记录中，**type** 输入栏数值是 **CWD** — 当前工作目录。这种类型用于记录在被执行的第一记录中详细说明的触发系统调用的进程。

这个记录的目的是为了记录当前进程的位置以防在相关的 PATH 记录中捕捉到相对路径。运用这个方法可以重塑绝对路径。

msg=audit(1364481363.243:24287)

msg 输入栏持有与第一记录中的数值相同的时间戳和 ID 值。

cwd="/home/shadowman"

cwd 输入栏含有进入目录的路径，在目录中触发系统调用。

第三记录

type=PATH

在第三记录中， **type** 输入栏值是 **PATH**。每个审核事件包含一个 **PATH** 对于每条路径种类的记录作为一个参数，传输给系统调用。在审核事件中，只有一条路径（**/etc/ssh/sshd_config**）被用来作为参数。

msg=audit(1364481363.243:24287):

msg 输入栏持有与第一和第二记录中的值相同的时间戳和 ID 值。

item=0

item 输入栏表明在所有项目中，哪个项目在 **SYSCALL** 类型记录中，参考了当前记录。这个是数字是以零为基准；值 **0** 意味着它是第一项。

name="/etc/ssh/sshd_config"

name 输入栏记录了文件或者目录的所有路径，作为参数被传输给系统调用。在这种情况下，它是 **/etc/ssh/sshd_config** 文件。

inode=409248

inode 输入栏包含索引结点数字，与记录在事件中的文件和目录有关。以下命令体现了与 **409248** 索引结点数字相关的文件和目录：

```
~]# find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00

dev 输入栏明确说明了设备的次要和主要 ID，它包含记录在事件中的文件和目录。在这种情况下，值代表 **/dev/fd/0** 设备。

mode=0100600

mode 输入栏记录了文件和目录权限，用 16 进制表示法编码。在这种情况下，**0100600** 可以被解释为 **-rw-------**，意味着对于 **/etc/ssh/sshd_config** 文件，只有 root 用户拥有读取并且输入权限。

ouid=0

ouid 输入栏记录了对象所有者的用户 ID。

ogid=0

ogid 输入栏记录对象拥有者的群组 ID。

rdev=00:00

rdev 输入栏包含记录的设备识别器只用于特殊文件。在这种情况下，正常文件是不用来作为记录文件的。

obj=system_u:object_r:etc_t:s0

obj 输入栏记录了 SELinux 语境，运用此语境可以在执行时间中标注分析进程。

以上分析过的审核事件是事件所包含的所有可能位置栏的一小部分。为了列出所有事件的位置栏及解释，请参考〈第 B.1 节 “审核事件字段”〉。为了列出所有事件类型以及解释，请参考〈第 B.2 节 “审核记录类型”〉。

例 5.5. 其他的 audit.log 事件。

以下审核事件记录了成功启动的 **auditd** 守护程序。 **ver** 位置栏显示了已经开始的审核守护程序的版本。

```
type=DAEMON_START msg=audit(1363713609.192:5426): auditd start, ver=2.2
format=raw kernel=2.6.32-358.2.1.el6.x86_64 auid=500 pid=4979
subj=unconfined_u:system_r:auditd_t:s0 res=success
```

以下审核事件记录了作为 root 用户使用 UID 500 登录失败。

```
type=USER_AUTH msg=audit(1364475353.159:24270): user pid=3280 uid=500
auid=500 ses=1 subj=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023 msg='op=PAM:authentication acct="root" exe="/bin/su"
hostname=? addr=? terminal=pts/0 res=failed'
```

5.7. 搜索审核日志文件

ausearch 实用程序允许您为特定事件搜索审核日志文件。默认情况下，**ausearch** 寻找 **/var/log/audit/audit.log** 文件。您可以使用 **ausearch *options* -if *file_name***命令来详细说明不同的文件。在一个 **ausearch** 命令中提供多种选择等同于使用 *AND* 运算符。

例 5.6. 使用 ausearch 搜索审核日志文件

因登录失败而搜索 **/var/log/audit/audit.log** 文件，请使用以下命令。

```
~]# ausearch --message USER_LOGIN --success no --interpret
```

搜索所有的账户，群组，角色变更，请使用以下命令：

```
~]# ausearch -m ADD_USER -m DEL_USER -m ADD_GROUP -m USER_CHAUTHTOK
-m DEL_GROUP -m CHGRP_ID -m ROLE_ASSIGN -m ROLE_REMOVE -i
```

搜索所有的由特定用户所执行的记录操作，使用用户的登录 ID（**auid**），请使用以下命令：

```
~]# ausearch -au 500 -i
```

搜寻从昨天至今所有的失败的系统调用，请使用以下命令：

```
~]# ausearch --start yesterday --end now -m SYSCALL -sv no -i
```

列出所有 **ausearch** 选项，请参考 ausearch(8) 手册页。

5.8. 创建审核报告

aureport 实用程序允许您生成有关记录在审核日志文件中事件的总结和分栏式报告。默认情况下，查询在 **/var/log/audit/** 目录中的所有 **audit.log** 文件来创建报告。您可以指定不同的文件来运行报告而不使用 **aureport *options* -if *file_name*** 命令。

例 5.7. 使用 aureport 来生成审核报告。

生成有关过去三天内不包括示例日在内的记录的事件，请使用以下命令：

```
~]# aureport --start 04/08/2013 00:00:00 --end 04/11/2013 00:00:00
```

生成所有可执行文件事件的一份报告，请使用以下命令：

```
~]# aureport -x
```

生成以上可执行文件事件的总结，请使用以下命令：

```
~]# aureport -x --summary
```

生成所有用户失败事件的总结报告，请使用以下命令：

```
~]# aureport -u --failed --summary -i
```

生成每个系统用户登录失败的总结报告，请使用以下命令：

```
~]# aureport --login --summary -i
```

通过 **ausearch** 查询搜索 用户 **500** 所有的文件访问事件生成一份报告，请使用以下命令：

```
~]# ausearch --start today --loginuid 500 --raw | aureport -f --
summary
```

生成一份报告有关所有被查询的审核文件以及所包含事件的事件范围，使用以下命令：

```
~]# aureport -t
```

列出所有的 **aureport** 选项，请参考 aureport(8) 手册页。

5.9. 其他资源

如需获取更多有关审核系统的信息，请参考以下资料来源：

网上来源

- Linux 审核系统项目页 ：http://people.redhat.com/sgrubb/audit/。
- Hack in the Box 杂志的文章〈*调查 Linux 审核系统的 Kernel 返回代码*〉：http://magazine.hackinthebox.org/issues/HITB-Ezine-Issue-005.pdf。

安装的文档

文档由 *audit* 软件包提供，可以在 **/usr/share/doc/audit-*version*/** 目录中找到。

手册页

- audispd.conf(5)
- auditd.conf(5)
- ausearch-expression(5)
- audit.rules(7)
- audispd(8)
- auditctl(8)
- auditd(8)
- aulast(8)
- aulastlog(8)
- aureport(8)
- ausearch(8)
- ausyscall(8)
- autrace(8)
- auvirt(8)

第 6 章 合规性与漏洞扫描

6.1. 红帽企业版 Linux 的安全合规性

"*合规审计*" 是用来解决给定对象是否遵循合规性策略中写明的所有规定的一个过程。"*合规策略*" 由负责指定所期望设置的安全专家定义，经常以清单的形式，使用在计算环境中。

合规策略在不同的组织之间有着很大的差别，甚至在同一组织的不同系统下也是如此。策略之间的差异基于这些系统的用途以及它们对于这些组织的重要程度而定。定制软件的设置以及部署的特性也对自定义策略清单提出了需求。

红帽企业版 Linux 提供了支持完全自动化合规审计的工具。这些工具基于安全内容自动化协议（SCAP）标准，专门为合规策略自动化调整设计。

支持红帽企业版 Linux 7 安全合规性工具

- **SCAP Workbench** — **scap-workbench** 图形化工具被设计为在单一的本地或者远程系统上执行配置和漏洞扫描。此外该工具也可以被用来生成基于这些扫描与评估的安全报告。
- **OpenSCAP** — **oscap** 命令行实用工具被设计为在本地系统上执行配置和漏洞扫描，验证安全合规性内容，以及生成基于这些扫描与评估的报告和指南。

如果您需要远程在多个系统上执行自动化合规审核，您可以利用 OpenSCAP 红帽卫星解决方案。欲了解更多信息，请参阅〈第 6.5 节 “在红帽 Satellite 上使用 OpenSCAP”〉 及 〈第 6.7 节 “附加资源”〉。

注意

需要注意的是红帽公司不随红帽企业版 Linux 7 分发提供任何默认的合规策略。原因在〈第 6.2 节 “典型的合规策略”〉中有解释。

6.2. 典型的合规策略

安全策略或者合规策略很少从头开始编写。**ISO 27000**系列标准，衍生产品，以及其他来源提供的安全策略模板和实践建议应该对启动编写有所帮助。然而，各机构组织在建立自己的安全程序时，需要对策略模板做修改，以便与他们自己的需求相匹配。策略模板选择的依据应该是挑选那些与企业环境相关联的模板，然后必须针对该模板进行调整，因为该模板要么包含了一些不能被应用于组织中的内置假定，要么明确的要求必须做出某些决定。

红帽企业版 Linux 的审核功能是基于 SCAP （安全内容自动化协议）标准的。SCAP 是一种综合的可互操作的规范，这种规范对格式与术语进行了标准化，通过这种标准化的规范向人类和机器传达软件缺陷以及安全配置信息。SCAP 是一种多用途的框架规范，它支持自动化配置、漏洞和补丁检查、技术控制达标活动以及安全性度量。

换句话说，SCAP 是一个独立于供应商外用于表达安全策略的方式，因此它被广泛的应用于现代企业中。SCAP 的规格打造了一个生态系统，其中安全性内容的格式著名且标准，同时扫描或者策略编辑的执行也不是强制性的。这种状态使得企业或者机构一旦建立起他们自己的安全策略(含SCAP内容)，就无需在意他们究竟雇佣了多少安全提供商。

SCAP 的最新版本包含了几个基本标准。这些组件根据他们自身的功能在 SCAP 内部被整理成组，如下所述：

SCAP 组件

- *语言* — 这组由 SCAP 语言组成，为表达合规策略定义了标准的词汇和约定。
  - *拓展配置清单描述格式 (XCCDF)* — 一种为表达、组织和管理安全指导的语言。
  - *开放脆弱性和评估语言 （OVAL）* — 一种被开发出来为已经过扫描的系统执行逻辑声明的语言。
  - *开放清单互动语言 （OCIL）* — 一种被设计用来为查询用户提供标准方法，解读用户对于给定问题的反馈的语言。
  - *资产识别 （AI）* — 一种被开发用于提供数据模型、研究方法以及引导鉴别安全资产的语言。
  - *资产报告格式 （ARF）* — 一种经过设计的语言，主要用来表达信息的传输格式，而这些信息则包含了收集好的安全资源，以及资源和安全报告之间的关系。
- *列举* — 本组包含 SCAP 标准定义的命名格式，以及从某些与安全相关领域利益相关而产生的项目的官方清单或者字典。
  - *普通参数列举 （CCE）* — 一种为应用程序和操作系统的安全相关的配置元素所列出的枚举。
  - *普通平台列举 (CPE)* — 一种结构化的命名方案，通常用来识别信息技术（IT）系统、平台以及软件包。
  - *普通漏洞与危险性 (CVE)* — 一种可用于参考公开的软件漏洞与风险集的方法。
- *度量* — 这组由一系列框架组成，用于识别和评估安全风险。
  - *普通参数划分系统(CCSS)* — 一种用于评估与安全相关的配置元素的度量系统，同时它也可以以打分的方式帮助用户优先考虑适当的应对措施。
  - *普通漏洞划分系统 (CVSS)* — 一种用于评估软件安全隐患的度量系统，同时它也可以以打分的方式帮助用户优先应对安全风险。
- *完整性* — 一种维护 SCAP 内容与扫描结果完整性的 SCAP 规范。
  - *信任模型的安全自动化数据 (TMSAD)* — 一组建议，这些推荐解释了现有规范的使用方法，在安全自动化领域里的 XML 文件上下文环境中，用来代表签名、哈希值、关键信息以及身份信息。

每个 SCAP 组件都有自己的基于 XML 的文档格式及 XML 名称空间。一个 SCAP 中所表达的合规策略既可以采用单个 OVAL 定义的 XML 文件、数据流文件和单个 zip 档案的方式，又可以采用一组各自包含表示策略清单的 XCCDF 文件的 XML 文件集这样的方式。

6.2.1. XCCDF 文件格式

XCCDF 语言被设计为支持信息交换、文档生成、组织化和情境化调整、自动一致性测试以及符合性评分。该语言主要是描述性质的，并不包含任何用来执行安全扫描的命令。然而，XCCDF 文档可以作为其他 SCAP 组件的参考，而且就其本身而言，它也可以被用于制作合规策略，移植到除相关的评估文档（OVAL、 OCIL）以外的所有目标平台。

通常，可以用一组 XML 文件中包含一个 XCCDF 清单的方法来表示合规策略。该 XCCDF 文件通常指向了评估资源、多重 OVAL，OCIL 以及脚本检查引擎（SCE）文件。此外，该文件集可以包含有 CPE 字典文件和为此字典定义了对象的 OVAL 文件。

作为一种基于 XML 的语言，XCCDF 定义并使用了大量可供选择的 XML 元素以及特性。下表简要介绍了主要的 XCCDF 元素；有关 XCCDF 更多的细节，请查阅 NIST跨机构报告7275 第 4修订版.

XCCDF 文档中的主要 XML 元素

- **<xccdf:Benchmark>** — 这是一个涵盖整个 XCCDF 文档的根元素。它也可以包含清单的元数据，例如标题、描述、作者列表、最近修改日期以及清单验收状态。
- **<xccdf:Rule>** — 这是一个关键元素，这个元素代表了清单的需求，同时保留了它的描述。它可以包含子元素，这些子元素定义了使用给定的规则验证或者执行合规性的动作，或者干脆修改这条规则自身。
- **<xccdf:Value>** — 该关键元素被用于表达其他 XCCDF 元素处于基准范围内的属性。
- **<xccdf:Group>** — 该元素被用于整理成一个XCCDF 文档，在相同环境下或者需求领域内，通过收集 **<xccdf:Rule>**、**<xccdf:Value>** 和 **<xccdf:Group>** 元素的方式，该元素将整理生成的 XCCDF 文档组合成架构。
- **<xccdf:Profile>** — 该元素为 XCCDF 基准的一个指定的调整服务。它允许基准保留数个不同的调整。**<xccdf:Profile>** 利用多个选择器元素，例如 **<xccdf:select>** 或者 **<xccdf:refine-rule>**，去判断即将修改和处理哪些正处于生效状态的元素。
- **<xccdf:Tailoring>** — 该元素允许从基准外部定义基准档案，这在某些时候是很理想的合规策略手工调整。
- **<xccdf:TestResult>** — 该元素用于记录目标系统上对于给定基准的扫描结果。每一个 **<xccdf:TestResult>** 都应该参考特定的资料，这些资料被用来定义为特定的扫描而制定的合规策略，而且它也包括与扫描密切相关的目标系统的重要信息。
- **<xccdf:rule-result>** — 这是 **<xccdf:TestResult>** 的一个子元素，用于保存从基准到目标系统应用特定规则的结果。
- **<xccdf:fix>** — 这是 **<xccdf:Rule>** 的一个子元素，用于修复那些不符合给定规则的目标系统。它可以包含一个运行在目标系统中的命令或者脚本，这个命令或脚本为了使系统符合规则而设计。
- **<xccdf:check>** — 这是 **<xccdf:Rule>** 的一个子元素，是一个外部来源，这个外部来源定义了如何评估给定的规则。
- **<xccdf:select>** — 这是一个选择器元素，用于包括或者排除选定的规则或者策略中的规则组。
- **<xccdf:set-value>** — 这是一个选择器元素，用于重写指定**<xccdf:Value>** 元素的当前值，而并不修改该元素的其他属性。
- **<xccdf:refine-value>** — 这是一个选择器元素，用于在策略调整过程中具体说明特定 **<xccdf:Value>** 元素的约束。
- **<xccdf:refine-rule>** — 这个选择器元素允许重写选定规则的属性。

例 6.1. XCCDF 文件示例

```
<?xml version="1.0" encoding="UTF-8"?>
<Benchmark xmlns="http://checklists.nist.gov/xccdf/1.2"
    id="xccdf_com.example.www_benchmark_test">
  <status>incomplete</status>
  <version>0.1</version>
  <Profile id="xccdf_com.example.www_profile_1">
    <title>Profile title is compulsory</title>
    <select idref="xccdf_com.example.www_group_1"
        selected="true"/>
    <select idref="xccdf_com.example.www_rule_1"
```

```
        selected="true"/>
    <refine-value idref="xccdf_com.example.www_value_1"
        selector="telnet service"/>
  </Profile>
  <Group id="xccdf_com.example.www_group_1">
    <Value id="xccdf_com.example.www_value_1">
      <value selector="telnet_service">telnet-server</value>
      <value selector="dhcp_servide">dhcpd</value>
      <value selector="ftp_service">tftpd</value>
    </Value>
    <Rule id="xccdf_com.example.www_rule_1">
      <title>The telnet-server Package Shall Not Be Installed </title>
      <rationale>
        Removing the telnet-server package decreases the risk
        of the telnet service's accidental (or intentional) activation
      </rationale>
      <fix platform="cpe:/o:redhat:enterprise_linux:6"
          reboot="false"
          disruption="low"
          system="urn:xccdf:fix:script:sh">
        yum -y remove
        <sub idref="xccdf_com.example.www_value_1"/>
      </fix>
      <check system="http://oval.mitre.org/XMLSchema/oval-definitions-
5">
        <check-export value-id="xccdf_com.example.www_value_1"
            export-name="oval:com.example.www:var:1"/>
        <check-content-ref href="examplary.oval.xml"
            name="oval:com.example.www:def:1"/>
      </check>
      <check system="http://open-scap.org/page/SCE">
        <check-import import-name="stdout"/>
        <check-content-ref href="telnet_server.sh"/>
      </check>
    </Rule>
  </Group>
</Benchmark>
```

6.2.2. OVAL 文件格式

OVAL（开放式漏洞评估语言）是 SCAP 中必不可少的和最初始的组成部分。OVAL 标准的主要目标是开启安全产品之间的互通互用能力。这由下面三个领域的标准化实现：

1. 目标系统配置的表现。
2. 为特定机器状态的存在而对目标系统所做的分析。
3. 报告指定机器状态和受观测机器状态之间的比较结果。

有别于其他工具或者自定义脚本，OVAL 语言以声明的形式描述了资源的理想状态。OVAL 语言代码不能被直接执行，而是依靠一个叫做 *扫描软件*的 OVAL 解释工具去执行。OVAL 所具备的声明性质保证了受评估系统的状态不会被意外地改变，这一点是非常重要的，因为安全扫描工具通常运行在可能获取的最高权限上。

OVAL 规范对公众意见与贡献、各类与 MITRE 合作的 IT 公司，以及由联邦政府资助的非营利组织开放。OVAL 规范一直在持续地进化中，不同版本间通过版本号进行区分。当前版本5.10.1发布于2012年1月。

类似所有其他的 SCAP 组件，OVAL 基于 XML 。OVAL 标准定义了几种文档格式。它们各自包含了不同种类的信息，服务于不同的目的。

OVAL 文档格式

- *OVAL Definitions* 格式是最常见的 OVAL 文件格式，直接用于系统扫描。OVAL定义文档描述了目标系统的理想状态。
- *OVAL Variables* 格式定义了一些变量用于修改 OVAL 定义文档。OVAL 变量文档通常与 OVAL 定义文件一起使用，以调整目标系统在运行时的安全内容。
- *OVAL System Characteristics* 格式保存有关评估系统的信息。OVAL 系统特性文档通常用于实际系统状态与 OVAL 定义文档中所定义的预期状态进行比对。
- *OVAL Results* 是用来报告系统评估结果的最全面的 OVAL 格式。OVAL 结果文档通常包括受评估 OVAL 定义的副本、受约束的 OVAL 变量、OVAL 系统特性以及经过计算的基于系统特性和定义的测试结果。
- *OVAL Directives* 格式通过包括或者排除某些细节的方式对 OVAL 结果文档中的冗余部分加以调整。
- *OVAL Common Model* 格式包含了用于其他几种 OVAL 方案中的构造和枚举的定义。它被用来再次利用 OVAL 定义，这样就可以避免在多个文档中发生重复的现象。

OVAL定义文档由一组配置需求所组成，每组需求在以下五个基本层面做了定义：*定义*、*测试*、*目标*、*声明*，和 *变量*。 定义部分内的元素描述了哪些测试应该被实现以便满足给定的定义。测试元素将对象与状态联系在了一起。在系统评估过程中，当一个受评估系统的资源可以用给定对象元素符合给定状态元素来表示，那么这个测试就会被认为是通过的。 变量部分定义了外部变量，这些外部变量可能被用于调整来自状态部分的元素。除了这些部分以外， OVAL 定义文档通常也包括 *发生器* 和 *签名* 部分。*发生器* 部分保存有关文档来源的信息以及各种与自身内容相关的额外信息。

每一种 OVAL 文档基础部分中的元素都可以明确地通过下表中的标识符进行识别：

```
oval:namespace:type:ID
```

namespace 是一个由命名空间定义的标识符 *type* 要么是定义元素 *def* ，要么是测试元素的 *tst* ，要么是对象元素 *obj* ，要么是状态元素 *ste* ，要么是变量元素 *var* ，而且*ID* 是标识符的一个整数值。

例 6.2. OVAL 定义文档示例

```
<?xml version="1.0" encoding="utf-8"?>
<oval_definitions
    xmlns:lin-def="http://oval.mitre.org/XMLSchema/oval-definitions-
5#linux"
    xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5"
    xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <generator>
    <oval:product_name>vim</oval:product_name>
    <oval:schema_version>5.10.1</oval:schema_version>
    <oval:timestamp>2012-11-22T15:00:00+01:00</oval:timestamp>
  </generator>
```

```
  <definitions>
    <definition class="inventory"
        id="oval:org.open-scap.cpe.rhel:def:7"
        version="1">
      <metadata>
        <title>Red Hat Enterprise Linux 7</title>
        <affected family="unix">
          <platform>Red Hat Enterprise Linux 7</platform>
        </affected>
        <reference ref_id="cpe:/o:redhat:enterprise_linux:7"
            source="CPE"/>
        <description>
          The operating system installed on the system is Red Hat
Enterprise Linux 7
        </description>
      </metadata>
      <criteria>
        <criterion comment="Red Hat Enterprise Linux 7 is installed"
            test_ref="oval:org.open-scap.cpe.rhel:tst:7"/>
      </criteria>
    </definition>
  </definitions>
  <tests>
    <lin-def:rpminfo_test check_existence="at_least_one_exists"
        id="oval:org.open-scap.cpe.rhel:tst:7"
        version="1"
        check="at least one"
        comment="redhat-release is version 7">
      <lin-def:object object_ref="oval:org.open-scap.cpe.redhat-
release:obj:1"/>
      <lin-def:state state_ref="oval:org.open-scap.cpe.rhel:ste:7"/>
    </lin-def:rpminfo_test>
  </tests>
  <objects>
    <lin-def:rpmverifyfile_object id="oval:org.open-scap.cpe.redhat-
release:obj:1"
        version="1">
      <!-- This object represents rpm package which owns /etc/redhat-
release file -->
      <lin-def:behaviors nolinkto='true'
          nomd5='true'
          nosize='true'
          nouser='true'
          nogroup='true'
          nomtime='true'
          nomode='true'
          nordev='true'
          noconfigfiles='true'
          noghostfiles='true' />
      <lin-def:name operation="pattern match"/>
      <lin-def:epoch operation="pattern match"/>
      <lin-def:version operation="pattern match"/>
      <lin-def:release operation="pattern match"/>
      <lin-def:arch operation="pattern match"/>
      <lin-def:filepath>/etc/redhat-release</lin-def:filepath>
    </lin-def:rpmverifyfile_object>
```

```
  </objects>
  <states>
    <lin-def:rpminfo_state id="oval:org.open-scap.cpe.rhel:ste:7"
        version="1">
      <lin-def:name operation="pattern match">^redhat-release</lin-
def:name>
      <lin-def:version operation="pattern match">^7[^\d]</lin-
def:version>
    </lin-def:rpminfo_state>
  </states>
</oval_definitions>
```

6.2.3. 数据流格式

SCAP 数据流是一种文件格式，自 SCAP 1.2 版本起开始使用，它代表了 XCCDF、OVAL 还有其他组件文件组成的包，可以被用来定义一个由 XCCDF 清单所表达的合规策略。它还包含一个索引和目录，允许按照 SCAP 组件把已知数据流分解成为文件。

数据流使用 XML 格式，包含了一个由一整个表的内容所构成的数据头以及一系列 **<ds:component>** 元素。每一个元素均包含一个 SCAP 组件，例如 XCCDF、OVAL、CPE 以及其他。数据流文件可以包含相同类型的多个组件，并且因此可以覆盖到所有您您的企业所需要的安全策略。

例 6.3. 一个数据流头示例

```
<ds:data-stream-collection
xmlns:ds="http://scap.nist.gov/schema/scap/source/1.2"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:cat="urn:oasis:names:tc:entity:xmlns:xml:catalog"
    id="scap_org.open-scap_collection_from_xccdf_ssg-rhel7-xccdf-
1.2.xml"
    schematron-version="1.0">
  <ds:data-stream id="scap_org.open-scap_datastream_from_xccdf_ssg-
rhel7-xccdf-1.2.xml"
      scap-version="1.2" use-case="OTHER">
  <ds:dictionaries>
    <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel7-
cpe-dictionary.xml"
        xlink:href="#scap_org.open-scap_comp_output--ssg-rhel7-cpe-
dictionary.xml">
      <cat:catalog>
        <cat:uri name="ssg-rhel7-cpe-oval.xml"
            uri="#scap_org.open-scap_cref_output--ssg-rhel7-cpe-
oval.xml"/>
      </cat:catalog>
    </ds:component-ref>
  </ds:dictionaries>
  <ds:checklists>
    <ds:component-ref id="scap_org.open-scap_cref_ssg-rhel7-xccdf-
1.2.xml"
        xlink:href="#scap_org.open-scap_comp_ssg-rhel7-xccdf-1.2.xml">
      <cat:catalog>
```

```
        <cat:uri name="ssg-rhel7-oval.xml"
            uri="#scap_org.open-scap_cref_ssg-rhel7-oval.xml"/>
      </cat:catalog>
    </ds:component-ref>
  </ds:checklists>
  <ds:checks>
    <ds:component-ref id="scap_org.open-scap_cref_ssg-rhel7-oval.xml"
        xlink:href="#scap_org.open-scap_comp_ssg-rhel7-oval.xml"/>
    <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel7-
cpe-oval.xml"
        xlink:href="#scap_org.open-scap_comp_output--ssg-rhel7-cpe-
oval.xml"/>
    <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel7-
oval.xml"
        xlink:href="#scap_org.open-scap_comp_output--ssg-rhel7-
oval.xml"/>
  </ds:checks>
</ds:data-stream>
<ds:component id="scap_org.open-scap_comp_ssg-rhel7-oval.xml"
    timestamp="2014-03-14T16:21:59">
  <oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-
definitions-5"
      xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5"
      xmlns:ind="http://oval.mitre.org/XMLSchema/oval-definitions-
5#independent"
      xmlns:unix="http://oval.mitre.org/XMLSchema/oval-definitions-
5#unix"
      xmlns:linux="http://oval.mitre.org/XMLSchema/oval-definitions-
5#linux"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://oval.mitre.org/XMLSchema/oval-common-5
      oval-common-schema.xsd
          http://oval.mitre.org/XMLSchema/oval-definitions-5
      oval-definitions-schema.xsd
          http://oval.mitre.org/XMLSchema/oval-definitions-
5#independent
      independent-definitions-schema.xsd
          http://oval.mitre.org/XMLSchema/oval-definitions-5#unix
      unix-definitions-schema.xsd
          http://oval.mitre.org/XMLSchema/oval-definitions-5#linux
      linux-definitions-schema.xsd">
```

6.3. 使用 SCAP 工作台

SCAP Workbench (*scap-workbench*) 是一个图形化的工具，它允许用户在本地或远程系统上执行配置和漏洞扫描，实现系统的修复，以及生成基于扫描评估的报告。需要注意的是，与 **oscap** 命令行实用工具比起来，SCAP 工作台只具备有限的功能。SCAP 工作台也可以处理只以 XCCDF 文件和数据流文件形式存在的安全内容。

以下各节说明如何安装，启动和使用 SCAP 工作台，以便进行系统扫描、修复和自定义扫描，并显示与这些任务相关的例子。

6.3.1. 安装 SCAP 工作台

若要在系统中安装 SCAP 控制台，请以 **root** 身份运行以下命令：

```
~]# yum install scap-workbench
```

该命令安装所有保证 SCAP 工作台能够正常工作的软件包，包括 *scap-workbench* 软件包提供的工具本身。需要注意的是，所需的依赖项，例如 *qt* 和 *openssh* 软件包，如果已经安装在您的系统中的话，将会被自动更新到可用的最新版本。

在您可以开始有效率地使用 SCAP 工作台之前，您还需要安装或者导入一些安全内容到您的系统中。您可以从相应的网站下载 SCAP 内容，或者，如果指定 RPM 文件或者安装包的话，您可以使用 **Yum** 软件包管理器从指定的位置或者资料库进行安装。

例如，您可以安装《SCAP 安全指南》（SSG）包， *scap-security-guide*, 包含目前 Linux 系统最先进最详尽的安全策略设置。请参阅 SSG 项目 页，了解怎样在系统中部署该软件包的具体步骤。

当您在系统中安装完 *scap-security-guide* 以后，除非另有指明，否则 SSG 安全内容可以在 **/usr/share/xml/scap/ssg/rhel7/** 目录下找到，而且您可以继续进行其他安全合规操作。

为了找到其他可能符合您需求的现有 SCAP 内容来源，请参阅 〈第 6.7 节 “附加资源”〉 。

6.3.2. 运行 SCAP 工作台

在成功安装 SCAP 工作台工具和 SCAP 内容以后，您就可以在您的系统中开始使用 SCAP 工作台了。为了从 GNOME 传统桌面环境中使用 SCAP 工作台，请按下 **Super** 键进入活动概览，输入 **scap-workbench**，然后按下 **Enter**。**Super** 键以各种形式出现，取决于键盘和其他硬件，但往往不是“Windows”键就是“Command”键，而且通常出现在 **Spacebar** 键的左侧。

一旦您启动该实用程序， **SCAP Workbench** 窗口就会出现。SCAP 工作台窗口包含许多交互式组件，您应该在开始扫描您的系统之前先熟悉这些组件：

输入文件

该字段包含了所选安全策略的完整路径。您可以通过点击 **Browse** 按钮搜索您系统中适用于 SCAP 的内容。

清单

该下拉列表框显示的是将被应用于所选安全策略中的清单的名称。如果存在不止一个清单，您可以通过点击此下拉列表框来选择一个特定的清单。

调整

该下拉列表框会通知您给定安全策略的定制情况。您可以通过点击该下拉列表框来选择自定义规则，这些规则将会被应用在系统评估中。默认值是 **(no tailoring)**，这意味着所使用的安全策略将不会有任何改变。如果您对所选的安全配置文件做了任何改动，可以通过点击 **Save Tailoring** 按钮以 XML 文档的方式保存这些改动内容。

配置文件

该下拉列表框包含所选安全策略配置文件的名称。通过点击该下拉列表框，您可以从给定的 XCCDF 或者数据流文件中筛选出安全配置文件。若要创建一个继承了所选安全策略配置文件属性的新配置文件，请点击 **Customize** 按钮。

目标

这两个单选按钮允许您选择待评估系统是本地计算机还是远程计算机。

选中的规则

该字段显示的是一系列受安全策略影响的安全规则。将鼠标悬停在某个特定的安全规则上以获取详细信息。

保存内容

该菜单允许您将由 **Input file** 以及 **Tailoring** 字段中选出的 SCAP 文件保存到选定的目录或者以 RPM 包的形式保存下来。

状态栏

这是一种图形化的工具条，指示着正在执行的操作状态。

在线修复

该复选框允许在系统评估中开启修复功能。如果您选中该复选框，SCAP 工作台将尝试校正那些无法匹配策略定义状态的系统设置。

扫描

该按钮允许您启动对指定系统的评估。

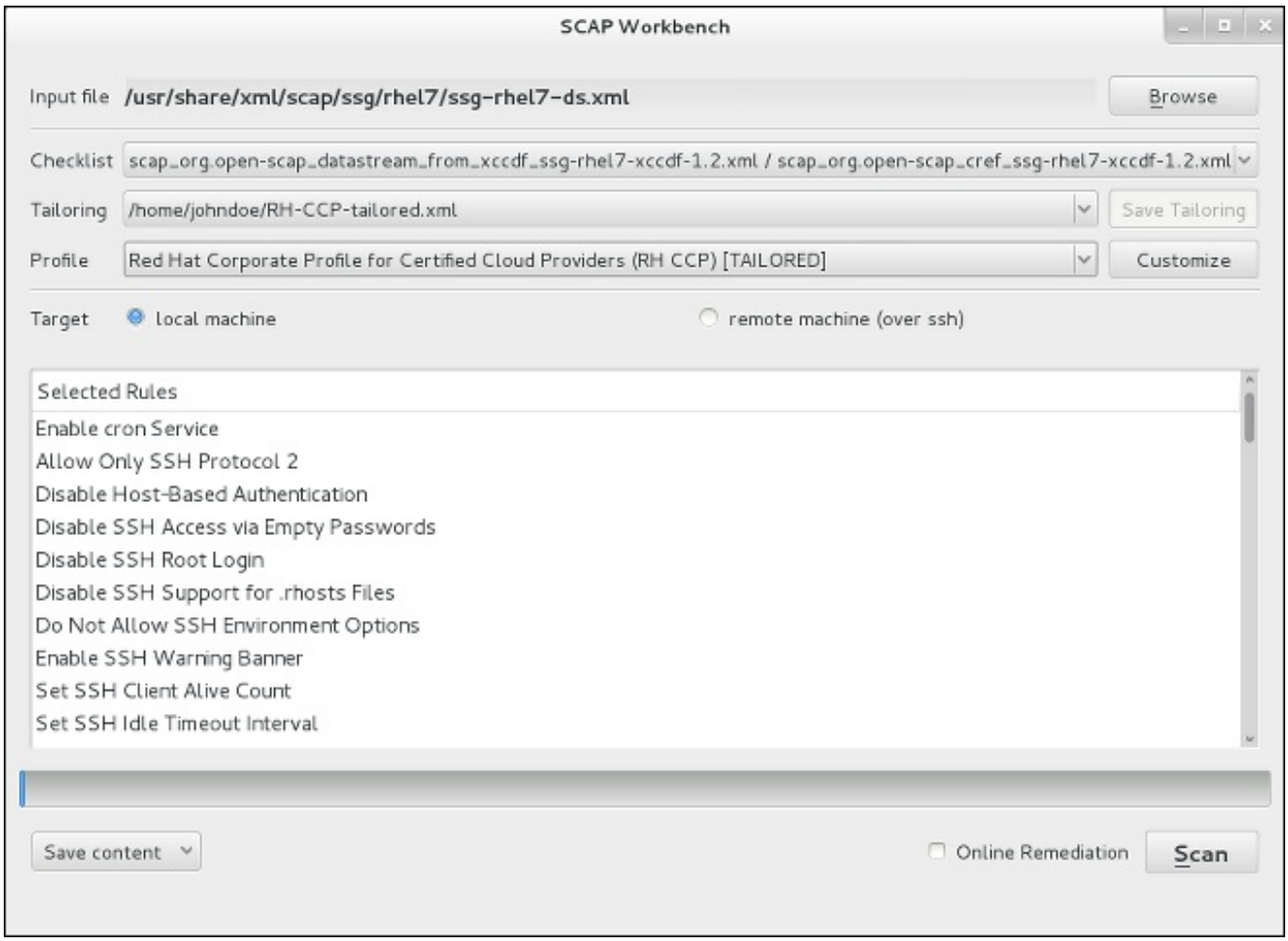

图 6.1. SCAP 工作台窗口

6.3.3. 扫描系统

SACP 工作台的主要功能是依照给定的 XCCDF 或者数据流文件，在被选中的系统中执行安全扫描。若要评估您的系统有没有违反所选的安全策略，请遵循下列步骤：

1. 通过点击 **Browse** 按钮还有寻找相应的 XCCDF 或者数据流文件来选择一项安全策略。

警告

选择一项安全策略会导致先前没有保存过的任何调整变更丢失。若要重新应用丢失的设置，您必须选择可用的配置文件并且重新调整内容。需要注意的是，您过去的自定义内容未必适用于新的安全策略。

2. 如果被选中的 SCAP 文件是一个数据流文件，提供了不止一个清单，您可以通过点击 **Checklist** 下拉列表框来选择特定的清单。

警告

更改清单可能会导致不同配置文件的选择和任何先前的自定义设置并不适用于新的清单。

3. 如果您已经预先安排了一个针对您的使用案例的自定义安全内容文件，可以通过点击 **Tailoring** 下拉列表框来加载此文件。您也可以通过变更现有安全配置文件的方式去创建一个自定义调整文件。欲了解更多信息，请参阅 〈第 6.3.4 节 “定制安全配置文件”〉。

   a. 如果您不希望使用任何当前系统评估的自定义设置，请选中 **(no tailoring)** 选项。如果之前没有任何自定义设置被选中话，该项目即为默认选项。

   b. 选中 **(open tailoring file...)** 选项来搜索特定的调整文件，这些文件被用于当前的系统评估。

   c. 如果您曾经使用过某个调整文件，SCAP 工作台会记住这个文件并把它添加到列表中。这简化了同一扫描中重复的应用程序。

4. 通过点击 **Profile** 下拉列表框来选择一个合适的安全配置文件。

   a. 若需进一步修改所选的配置文件，请点击 **Customize** 按钮。有关配置文件自定义的详细信息，请参阅 〈第 6.3.4 节 “定制安全配置文件”〉。

5. 分别在两个 **Target** 单选按钮中选中一个来扫描本地或者远程计算机。

   a. 如果您选择了远程系统，通过输入用户名、主机名以及端口信息的方式来指定它，如下例所示：

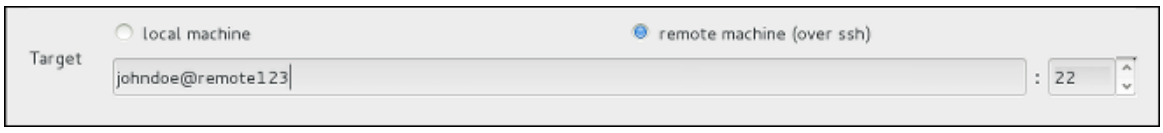

图 6.2. 指定一个远程系统

6. 您可以通过选中 **Online remediation** 复选框来允许系统设置自动校正。 启用该选项时，如果在系统扫描中相关的检查失败的话，SCAP工作台将会按照策略中所应用的安全规则尝试去改变系统配置。

警告

如果使用不谨慎，在修复选项启用的情况下运行系统评估可能会导致系统丧失功能。

7. 点击 **Scan** 按钮来启动系统扫描。

6.3.4. 定制安全配置文件

选择好适合您的安全策略的安全配置文件以后，您可以通过点击 **Customize** 按钮来进一步调整。这将打开一个新的调整窗口，该窗口允许您修改当前选中的 XCCDF 配置文件，而实际上并不用改动各自的 XCCDF 文件。

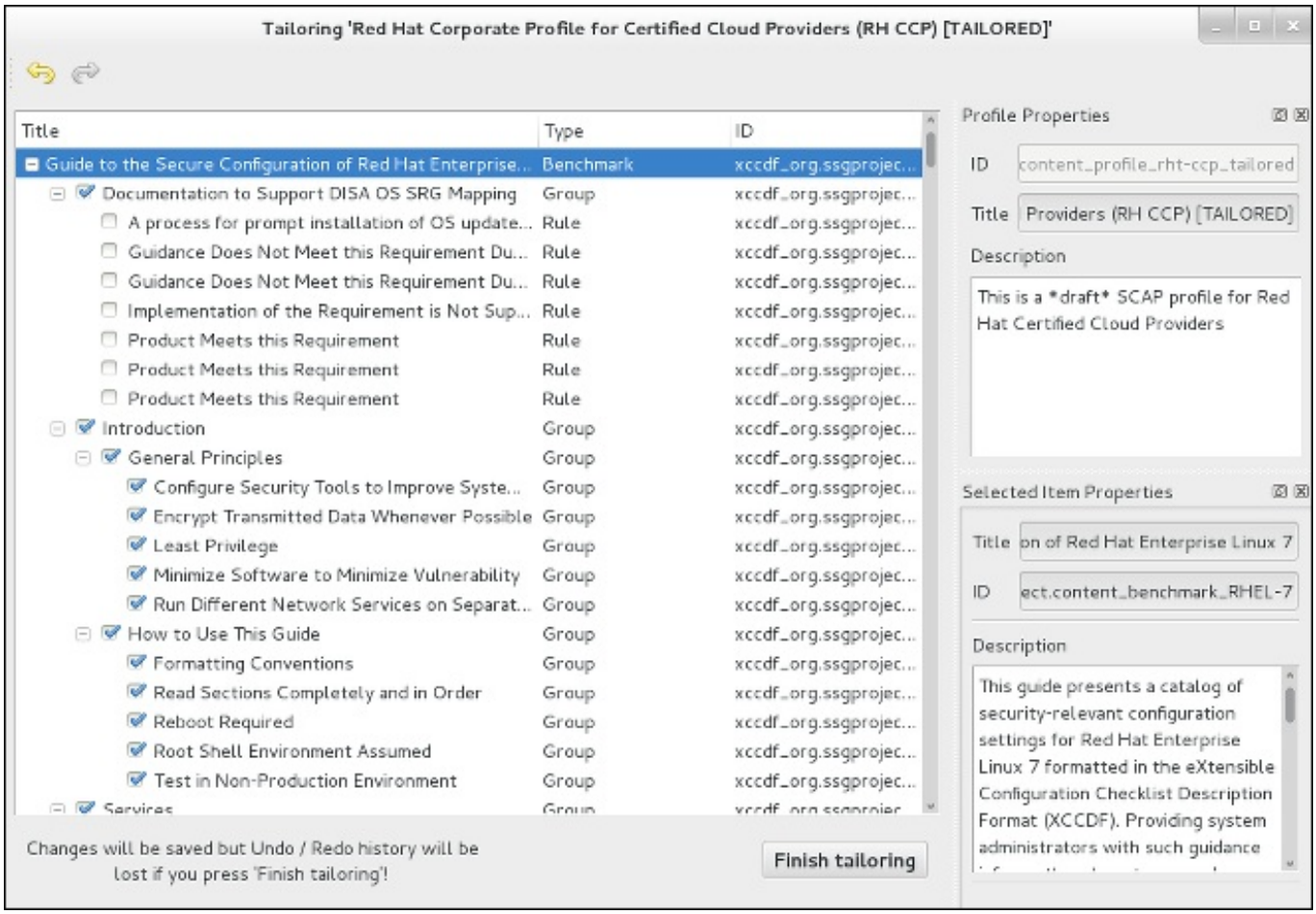

图 6.3. 定制所选的安全配置文件

Tailoring 窗口包含了一整套与选中的安全配置文件相关的 XCCDF 元素，这些安全配置文件包含了每个元素及其功能的详细信息。您可以通过在该窗口的主要领域选择或者反向选择相应的复选框来打开或者关闭这些元素。该调整窗口还支持 **undo** 和 **redo** 功能；您可以通过点击窗口左上角各自的箭头图标来撤销或者重做您的选择。

当您完成您的配置文件定制后，通过点击 **Finish Tailoring** 按钮来确认这些变更。您所做的变更被保存在内存中，如果 SCAP 工作台被关闭或者产生了某些变化，比如选择了一个新的 SCAP 内容或者另一个调整选项，您所做的变更将不复存在。如果您希望这些变更被储存下来，请点击 **SCAP Workbench** 窗口中的**Save Tailoring** 按钮。该操作允许您以一个 XCCDF 调整文件的方式在选中的目录下保存您对安全配置文件所做的变更。需要注意的是该调整文件以后也可以与其他配置文件一起选择。

6.3.5. 保存 SCAP 内容

SCAP 工作台也可以允许您保存被用于您的系统评估中的 SCAP 内容。您既可以分开保存调整文件 (请参阅〈第 6.3.4 节 “定制安全配置文件”〉)，也可以通过点击 **Save content** 下拉列表框，选择 **Save into a directory** 或者 **Save as RPM** 选项来一次性保存所有的安全内容。

通过选中 **Save into a directory** 选项，SCAP 工作台将 XCCDF 或数据流文件和调整文件两者都保存到指定的位置。这可以作为一个有效的备份方案。

通过选中 **Save as RPM** 选项，您可以令 SCAP 工作台创建一个包含 XCCDF 或数据流文件和调整文件的 RPM 包。这对于分发期望的安全内容到那些无法被远程扫描的系统，或者仅仅是为了实现今后对内容的进一步处理，是非常有用处的。

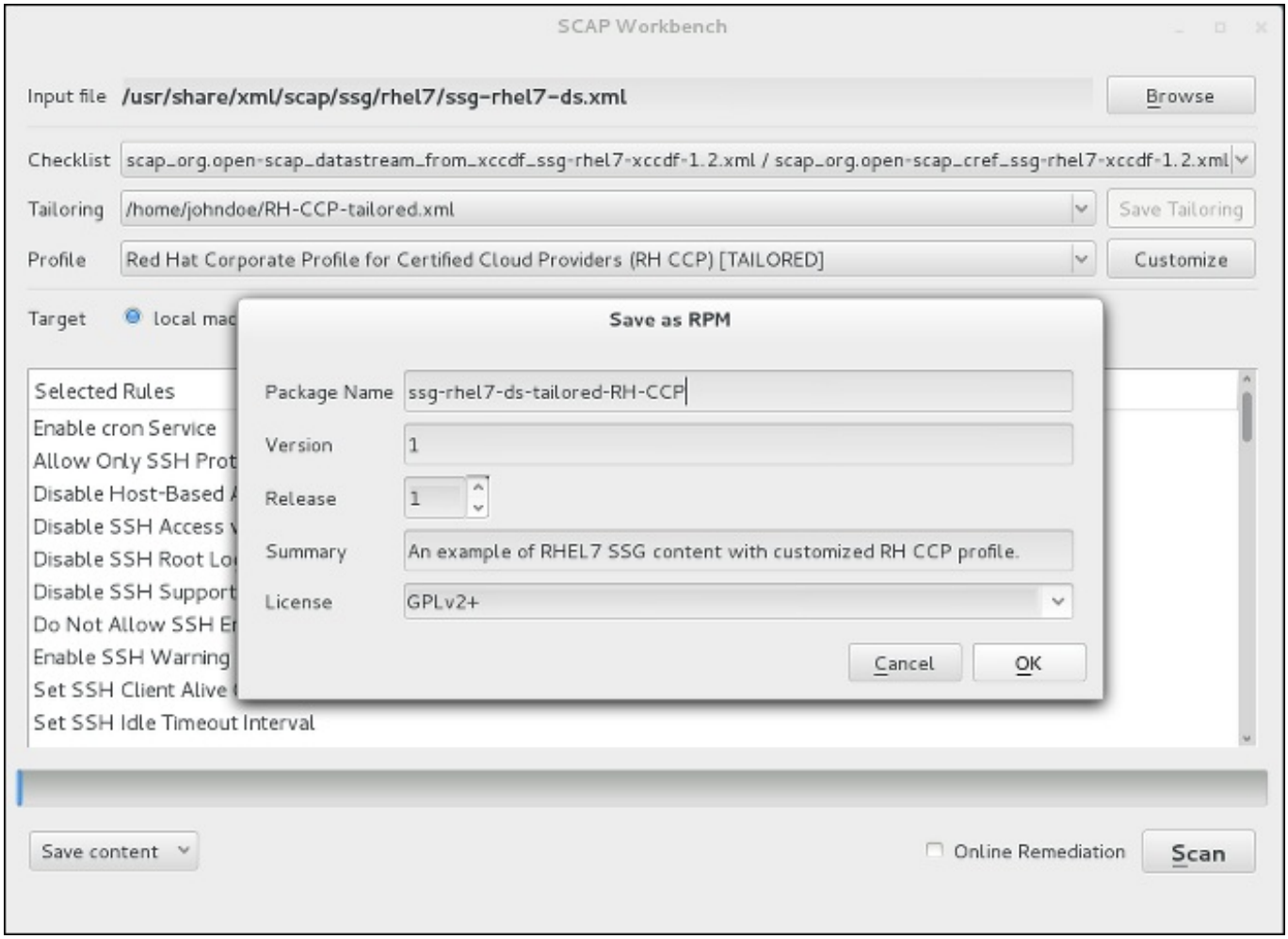

图 6.4. 以 RPM 包的形式保存当前的 SCAP 内容

6.3.6. 查看扫描结果并生成扫描报告

当系统扫描结束以后，两个新的按钮， **Clear** 和 **Report**，会出现并取代 **Scan** 按钮。

警告

点击 **Clear** 按钮会永久移除扫描结果。

您可以通过点击 **Report** 按钮来展示和进一步处理扫描结果，该操作会打开 **Evaluation Report** 窗口。此窗口包含 **Save** 下拉列表框，还有两个按钮， **Open in Browser**，和 **Close**。

您可以通过点击 **Save** 下拉列表框来以 XCCDF、ARF 或者 HTML 文件的形式来保存扫描结果。选中 **HTML Report** 选项来生成可读的扫描报告。XCCDF 和 ARF（数据流）格式适合进一步地自动化处理。您可以反复去选择这三个选项。

如果您喜欢便能立即查看而不保存扫描报告，您可以点击 **Open in Browser** 按钮，这将以一个临时的 HTML 文件默认的网络浏览器中打开这些扫描报告。

6.4. 使用 oscap

oscap 命令行工具允许用户扫描本地系统， 验证安全合规内容，生成基于这些系统扫描与评估的报告和指南。该工具作为一个 OpenSCAP 库的前端，基于它处理的一种类型的 SCAP 内容，将其功能分组模块化（子命令）。

以下各节解释了如何安装 **oscap**，执行最常见的操作，并且显示与这些工作相关的例子。要了解更多与特定的子命令有关的内容，请使用 **--help** 选项加上**oscap** 命令：

```
oscap [options] module module_operation
[module_operation_options_and_arguments] --help
```

module 代表一种正在被处理的 SCAP 内容类型， *module_operation* 是一种对 SCAP 内容进行特定操作的子命令。

例 6.4. 获取有关具体 oscap 操作的帮助

```
~]$ oscap ds sds-split --help
oscap -> ds -> sds-split

Split given SourceDataStream into separate files

Usage: oscap [options] ds sds-split [options] SDS TARGET_DIRECTORY

SDS - Source data stream that will be split into multiple files.
TARGET_DIRECTORY - Directory of the resulting files.

Options:
   --datastream-id <id>          - ID of the datastream in the
collection to use.
   --xccdf-id <id>               - ID of XCCDF in the datastream that
should be evaluated.
```

要了解所有 **oscap** 特性及其设置的完整列表，请参阅 **oscap(8)** 手册页。

6.4.1. 安装 oscap

为了安装 **oscap** 到您的系统中，需要以 **root** 用户身份运行以下命令：

```
~]# yum install openscap-utils
```

此命令允许您安装保证 **oscap** 正常运行所需的所有安装包，包括提供实用工具自身的 *openscap* 软件包。如果您想编写您自己的安全内容，您也应该安装 *openscap-engine-sce* 包，这个安装包提供了脚本检查引擎（SCE）。SCE 是 SCAP 的一个扩展协议，允许内容作者使用脚本语言去编写自己的安全内容，例如 Bash 语言，Python 语言或者 Ruby 语言。该安装包可以以和 *openscap-utils* 软件包同样的方式进行安装。

根据需要，在安装完 **oscap** 后，您可以检查您所安装 **oscap** 版本的功能，比如它支持什么样的规格，某个 **oscap** 文件储存在什么位置，能使用什么样的 SCAP 对象，以及其他有用的信息。要显示此信息，请输入以下命令：

```
~]$ oscap -V
OpenSCAP command line tool (oscap) 1.0.4
Copyright 2009--2014 Red Hat Inc., Durham, North Carolina.

==== Supported specifications ====
XCCDF Version: 1.2
OVAL Version: 5.10.1
CPE Version: 2.3
CVSS Version: 2.0
CVE Version: 2.0
Asset Identification Version: 1.1
Asset Reporting Format Version: 1.1

==== Capabilities added by auto-loaded plugins ====
SCE Version: 1.0 (from libopenscap_sce.so.8)

==== Paths ====
Schema files: /usr/share/openscap/schemas
Schematron files: /usr/share/openscap/xsl
Default CPE files: /usr/share/openscap/cpe
Probes: /usr/libexec/openscap

==== Inbuilt CPE names ====
Red Hat Enterprise Linux - cpe:/o:redhat:enterprise_linux
Red Hat Enterprise Linux 5 - cpe:/o:redhat:enterprise_linux:5
Red Hat Enterprise Linux 6 - cpe:/o:redhat:enterprise_linux:6
Red Hat Enterprise Linux 7 - cpe:/o:redhat:enterprise_linux:7
Fedora 16 - cpe:/o:fedoraproject:fedora:16
Fedora 17 - cpe:/o:fedoraproject:fedora:17
Fedora 18 - cpe:/o:fedoraproject:fedora:18
Fedora 19 - cpe:/o:fedoraproject:fedora:19
Fedora 20 - cpe:/o:fedoraproject:fedora:20
Fedora 21 - cpe:/o:fedoraproject:fedora:21
Red Hat Enterprise Linux Optional Productivity Applications -
cpe:/a:redhat:rhel_productivity
Red Hat Enterprise Linux Optional Productivity Applications 5 -
cpe:/a:redhat:rhel_productivity:5

==== Supported OVAL objects and associated OpenSCAP probes ====
system_info                  probe_system_info
family                       probe_family
filehash                     probe_filehash
environmentvariable          probe_environmentvariable
textfilecontent54            probe_textfilecontent54
textfilecontent              probe_textfilecontent
variable                     probe_variable
xmlfilecontent               probe_xmlfilecontent
```

```
environmentvariable58          probe_environmentvariable58
filehash58                     probe_filehash58
inetlisteningservers           probe_inetlisteningservers
rpminfo                        probe_rpminfo
partition                      probe_partition
iflisteners                    probe_iflisteners
rpmverify                      probe_rpmverify
rpmverifyfile                  probe_rpmverifyfile
rpmverifypackage               probe_rpmverifypackage
selinuxboolean                 probe_selinuxboolean
selinuxsecuritycontext         probe_selinuxsecuritycontext
file                           probe_file
interface                      probe_interface
password                       probe_password
process                        probe_process
runlevel                       probe_runlevel
shadow                         probe_shadow
uname                          probe_uname
xinetd                         probe_xinetd
sysctl                         probe_sysctl
process58                      probe_process58
fileextendedattribute          probe_fileextendedattribute
routingtable                   probe_routingtable
```

在可以开始有效率地使用 **oscap** 实用工具之前，您还必须安装或者导入一些安全内容到您的系统中。您可以从相应的网站中下载 SCAP 内容，或者，如果指定为 RPM 文件或者软件包，您可以使用 **Yum** 安装包管理器从指定位置或者资料库进行安装。

例如，您可以安装《SCAP 安全指南》（SSG）软件包，*scap-security-guide*，该软件包包含了 Linux 系统最新的一套安全策略。请参阅 SSG project 页，了解在您的系统中部署该软件包的具体步骤。

当您在系统中安装完 *scap-security-guide* 以后，除非另有指明，否则 SSG 安全内容可以在 **/usr/share/xml/scap/ssg/rhel7/** 目录下找到，而且您可以继续进行其他安全合规操作。

为了找到其他可能符合您需求的现有 SCAP 内容来源，请参阅 〈第 6.7 节 “附加资源”〉。

在您的系统中安装好 SCAP 内容以后，**oscap** 可以通过指定文件路径到内容的方式处理这些内容。**oscap** 工具支持 SCAP 1.2 版本，同时它向下兼容 SCAP 1.1和1.0版本，这样它就可以直接处理 SCAP 内容的早期版本而无需任何特殊的需求。

6.4.2. 显示 SCAP 内容

SCAP 标准定义了众多文件格式。**oscap** 工具可以处理或者创建符合许多格式的文件。为了进一步处理与 SCAP 内容相关的给定文件，您需要了解如何根据给定的文件类型来运用 **oscap** 。如果不确定如何使用一个特定的文件，您既可以打开这个文件，也可以读取这个文件，或者您也可以使用 **oscap**的 **info** 模块，解析文件，并以可读的格式提取相关信息。

运行下列命令来检查 SCAP 文档的内部结构，并显示有用的信息，例如文档类型、规范版本、文档的状态、文档的发布日期，以及文档被复制到系统中的日期：

```
oscap info file
```

其中 *file* 是正在被检查的安全内容的完整路径。下面的示例更能说明 **oscap info** 命令的用法：

例 6.5. 显示 SCAP 内容信息

```
~]$ oscap info /usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xmal
Document type: Source Data Stream
Imported: 2014-03-14T12:22:01

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-rhel7-xccdf-
1.2.xml
Generated: (null)
Version: 1.2
Checklists:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-xccdf-1.2.xml
                Profiles:
                        xccdf_org.ssgproject.content_profile_test
                        xccdf_org.ssgproject.content_profile_rht-ccp
                        xccdf_org.ssgproject.content_profile_common
                        xccdf_org.ssgproject.content_profile_stig-
rhel7-server-upstream
                Referenced check files:
                        ssg-rhel7-oval.xml
                                system:
http://oval.mitre.org/XMLSchema/oval-definitions-5
Checks:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-oval.xml
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel7-cpe-oval.xml
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel7-oval.xml
Dictionaries:
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel7-cpe-
dictionary.xml
```

6.4.3. 扫描系统

oscap 最重要的功能是在本地系统上执行配置与漏洞扫描。以下是各个命令的一般语法：

```
oscap [options] module eval [module_operation_options_and_arguments]
```

oscap 工具可以针对由两方代表的 SCAP 内容扫描系统，这两方包括**XCCDF** （可扩展的配置检查清单描述格式）基准和 **OVAL** （开放弱点评估语言）定义。安全策略可以以单独的OVAL文件或者XCCDF文件的形式存在，也可以以多个单独的 XML 文件的形式存在 ，这里每个 XML 文件代表了不同的组件（XCCDF, OVAL, CPE, CVE, 还有其他）。扫描结果可以打印为两种，标准输出和 XML 文件。结果文件可以经由 **oscap** 做进一步处理以便生成可读的报告。下面的例子说明了该命令最常见的用法。

例 6.6. 使用SSG OVAL 定义扫描系统

要针对 SSG OVAL 定义文件扫描您的系统，同时评估所有的定义，请运行以下命令：

```
~]$ oscap oval eval --results scan-oval-results.xml
/usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

扫描结果将会以 **scan-oval-results.xml** 文件的方式保存在当前目录中。

例 6.7. 使用SSG OVAL 定义扫描系统

为了评估来自由 SSG 数据流文件代表的安全策略中的特别的 OVAL 定义，请运行以下命令：

```
~]$ oscap oval eval --id oval:ssg:def:100 --results scan-oval-
results.xml /usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

扫描结果将会以 **scan-oval-results.xml** 文件的方式保存在当前目录中。

例 6.8. 使用 SSG XCCDF 基准扫描系统

要在系统中为 **xccdf_org.ssgproject.content_profile_rht-ccp** 配置文件执行 SSG XCCDF 基准测试，请运行以下命令：

```
~]$ oscap xccdf eval --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results scan-xccdf-
results.xml scan-xccdf-results.xml /usr/share/xml/scap/ssg/rhel7/ssg-
rhel7-ds.xml
```

扫描结果将会以 **scan-xccdf-results.xml** 文件的方式保存在当前目录中。

注意

--profile 命令行参数从给定的 XCCDF 或者数据流文件中选择安全配置文件。可用的配置文件列表可以通过运行 **oscap info** 命令来获取。如果 **--profile** 命令行参数被省略了，默认的 XCCDF 配置文件将根据 SCAP 标准的要求被使用。需要注意的是默认的 XCCDF 配置文件可能是，也可能不是一个合适的安全策略。

6.4.4. 生成报告和指南

oscap 的另一个有用的功能是能够生成可读的 SCAP 内容。**oscap** 实用工具允许您将一个XML文件转换成 HTML 或者纯文本格式。该功能被用于生成安全指南或者清单，这些指南或清单可以作为信息的来源，同样也可以用于指导安全系统配置。系统扫描结果也可以被转换成高可读性的结果报告。一般的命令语法如下：

```
oscap module generate sub-module [specific_module/sub-
module_options_and_arguments] file
```

这里 *module* 是 **xccdf** 或 **oval** 两者之一， *sub-module* 是一种生成的文档，并且 *file* 代表一个 XCCDF 或者 OVAL 文件。

下面展示的是该命令在使用过程中最常见的例子：

例 6.9. 生成一份包含清单的指南

要为 **xccdf_org.ssgproject.content_profile_rht-ccp** 配置文件生成一份包含清单的指南，请运行以下命令：

```
~]$ oscap xccdf generate guide --profile
xccdf_org.ssgproject.content_profile_rht-ccp
/usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml > ssg-guide-
checklist.html
```

这份指南将会以 **ssg-guide-checklist.html** 文件的方式储存在当前目录下。

例 6.10. 将 SSG OVAL 扫描结果转换为报告

要将一份SSG OVAL 扫描结果转换为 HTML 文件，请运行以下命令：

```
~]$ oscap oval generate report scan-oval-results.xml > ssg-scan-oval-report.html
```

这份结果报告将会以 **ssg-scan-oval-report.html** 文件的方式储存在当前目录下。此示例假定您从与 **scan-oval-results.xml** 文件存放的相同位置运行该命令。否则，您需要指定该文件及包含其扫描结果的完整路径。

例 6.11. 将SSG XCCDF 扫描结果转换为报告

要将一份 SSG XCCDF 扫描结果转换为 HTML 文件，请运行以下命令：

```
~]$ oscap xccdf generate report scan-xccdf-results.xml > scan-xccdf-report.html
```

这份结果报告将会以 **ssg-scan-xccdf-report.html** 文件为名储存在当前目录下。 或者，您可以使用 **--report** 命令行参数在扫描过程中生成此报告。

```
~]$ oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_rht-ccp --resultsscan-xccdf-results.xml --report scan-xccdf-report.html/usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

6.4.5. 验证 SCAP 内容

在系统中使用安全策略之前，您应该首先验证所使用的策略，以避免该策略可能包含的任何语法或者语意上的错误。 **oscap** 实用工具可被用于验证针对标准 SCAP XML 架构的安全内容。验证结果会被打印到标准错误流 (stderr) 中。该验证命令的一般语法如下：

```
oscap module validate [module_options_and_arguments] file
```

这里 *file* 被验证是该文件的完整路径。唯一例外的是数据流模块 (ds)，这里使用的是 **ds-validate** 操作来代替 **validate**。需要注意的是，所有给定数据流中的 SCAP 组件都会被自动验证，而且没有任何组件会被单独指定，这点从下面的例子中就可以看出：

```
~]$ oscap ds sds-validate /usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

对于某些 SCAP 内容，比如 OVAL 规范，您也可以执行 Schematron 验证。 Schematron 验证比标准验证慢，但是提供了更深入的分析，并因此能够检测出更多的错误。下面的 SSG 示例显示了该命令的典型用法：

```
~]$ oscap oval validate --schematron /usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

6.5. 在红帽 Satellite 上使用 OpenSCAP

当运行多个红帽企业版Linux 系统时，保持所有的系统均符合您的安全策略，且从一个位置远程执行安全扫描和评估是非常重要的。这些可以通过安装在您的 Satellite 客户端（需红帽 Satellite 5.5 以上的版本）上的 *spacewalk-oscap* 软件包来实现。该软件包可以从 **Red Hat Network Tools** 频道找到。

该解决方案支持两种方式执行合规扫描、查看还有进一步处理扫描结果。您可以使用 **OpenSCAP Satellite Web Interface** 或者通过 **Satellite API** 运行命令和脚本。有关此解决方案的安全合规性，及其需求和能力的详细信息，请参阅 《红帽卫星 5.6 用户指南》。

6.6. 应用实例

这一部分展示了为红帽产品提供的某个安全内容的实际使用情况。

6.6.1. 红帽产品的审计安全漏洞

红帽会为其产品提供持续不断的 OVAL 定义。这些定义允许在已安装的软件中开启漏洞完全自动化审计。要了解项目的更多信息，请查阅 http://www.redhat.com/security/data/metrics/。要下载这些定义，请运行以下命令：

```
~]$ wget http://www.redhat.com/security/data/oval/com.redhat.rhsa-all.xml
```

红帽 Satellite 5的用户可能会发现补丁定义中有帮助的 XCCDF 部分。要下载这些定义，请运行以下命令：

```
~]$ wget http://www.redhat.com/security/data/metrics/com.redhat.rhsa-all.xccdf.xml
```

要审核系统中已安装软件的安全漏洞，请运行以下命令：

```
~]$ oscap oval eval --results rhsa-results-oval.xml --report oval-report.html com.redhat.rhsa-all.xml
```

oscap 工具将Red Hat Security Advisories 映射到了 CVE 标识符中，这些标识符与国家漏洞数据库相连，且会报告哪些安全公告没有被应用到系统中。

注意

需要注意的是，这些 OVAL 定义被设计为仅用于涉及红帽所发布的软件和更新之中。您需要提供额外的定义以便能及时检测第三方软件的补丁状态。

6.6.2. 使用 SCAP 安全指南审核系统设置

SCAP 安全策略 （SSG） 项目软件包， *scap-security-guide*，包含了 Linux 系统最新的安全策略设置。请参阅 SSG project 页面了解如何在您的系统中部署该软件包。部分 *scap-security-guide* 也可以为红帽企业版 Linux 7的设置提供指导。要检查 *scap-security-guide* 中存在的安全内容，请使用 **oscap info** 模块：

```
~]$ oscap info /usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

这个命令输出的是 SSG 文档的一个概述，它包含了可用的配置文件。要对您的系统设置进行审核，请选择一个合适的配置文件，并运行恰当的评估命令。例如，针对草拟的配置文件，下面的命令为经过认证的红帽云供应商对给定的系统进行评估：

```
~]$ oscap xccdf eval --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results ssg-rhel7-xccdf-
result.xml --report ssg-rhel7-report.html
/usr/share/xml/scap/ssg/rhel7/ssg-rhel7-ds.xml
```

6.7. 附加资源

有关各类安全合规领域内的更多感兴趣的信息，请参阅如下资源：

安装的文档

- **oscap**(8) — **oscap** 命令行工具手册页提供了可用选项的完整列表及其用法的使用说明。
- **scap-workbench**(8) — **SCAP Workbench** 应用程序手册页提供了应用程序的基本信息，以及一些潜在的 SCAP 内容源链接。
- 红帽企业版 Linux 7的安全设置指南 — 一份HTML文档，位于 **/usr/share/doc/scap-security-guide-0.1.5/** 目录下，以 XCCDF 清单的形式为您的系统提供详细的安全指南。

在线文档

- The OpenSCAP project page — OpenSCAP项目的主页提供了 **oscap** 实用工具以及其他 SCAP 相关组件和项目的详细信息。
- The SCAP Workbench project page — SCAP工作台项目的主页提供了 **scap-workbench** 应用程序的详细信息。
- The SCAP Security Guide (SSG) project page — SSG 项目的主页提供了有关红帽企业版 Linux 的最新安全内容。
- National Institute of Standards and Technology (NIST) SCAP page — 此页代表了 SCAP 相关材料的一个庞大集合，包括 SCAP 的 出版物、技术参数以及 SCAP 验证程序。
- National Vulnerability Database (NVD) — 此页代表了最大规模的 SCAP 内容资料库，以及最大规模的其他基于漏洞管理数据的 SCAP 标准。
- Red Hat OVAL content repository — 这是一个包含了红帽企业版Linux系统 OVAL 定义的储存库。
- MITRE CVE — 这是一个由 MITRE 公司提供的公开的安全漏洞数据库。
- MITRE OVAL — 该页代表了一个由 MITRE 公司提供的 OVAL 相关项目。除去其他 OVAL 相关信息，这些页面包含了 OVAL 语言的最新版本以及一个巨大的 OVAL 内容资料库，总计超过22,000条 OVAL 定义。
- 红帽卫星 5.6 用户指南 — 该书在众多话题中，描述了如何在多系统上使用 OpenSCAP 来维护系统安全。

第 7 章 联邦标准和法规

要保证安全等级，可能需要您的机构在符合联邦和行业安全规格、标准和规则方面有所努力。本章论述了这些标准和规则的一部分。

7.1. 联邦信息处理标准（FIPS）

FIPS （美国联邦信息处理标准）出版物 140-2，是一个计算机的安全标准，有美国政府和业界工作组来验证密码模块的质量。FIPS 出版物（包括140-2）可以在以下网址中找到：http://csrc.nist.gov/publications/PubsFIPS.html。注意在编写的时候，出版物 140-3 处于草稿阶段，并不能代表完成的标准。 FIPS 标准提供了四项安全 *等级*来确保可以足够涵盖不同的行业、加密模块的执行和组织大小和要求。这些层次描述如下：

- 等级1 — 安全等级 1 提供了最低等级的安全性。基本的安全要求对于加密模块有详细规定（比如，至少需要使用一个 Approved 算法或者 Approved 安全性能）。没有具体的物理安全机制在安全等级 1 的加密模块中必须超过生产等级组成的基本要求。一个安全等级为1的加密模块的例子就是 PC（个人电脑）加密板。
- 等级2 — 安全等级2通过增加了篡改证据的要求，以加强安全等级1加密模块的物理安全机制，这包括了使用篡改争取的涂膜和封章，或模块中可移动覆盖或门的防盗锁。篡改证据的涂膜和印章被置于加密模块上因此涂膜或封印必须被破解以物理进入模块中的纯文本加密秘钥和 CSP （重要安全参数）。篡改证据印章或防盗锁都被置于封面或者门上以防止未授权的物理访问。
- 等级3— 除了安全等级2所要求的篡改证据安全机制外，安全等级3尝试阻止入侵者获得进入加密模块中 CSP 的权限。安全等级3所要求的物理安全机制目的是得到尝试物理访问探测和反应的高可能性，使用或者修改加密模块。当加密模块中可移动的封面/门被打开时，物理安全机制也许包括使用强大的附件和篡改检测器/响应电路中将所有的纯文本 CSP 归零。
- 等级4—安全等级4 提供了这一标准中最高的安全等级。在这一安全等级上，物理安全机制提供了在加密模块周围的完整保护套，目的是监测并对所有不授权的物理访问尝试作出回应。加密模块从任何方向渗透有非常高的几率会被探测出，导致了所有纯文本 CSP 的即刻归零。安全等级4的加密模块对于物理不受保护的环境中的运作十分有用。

更多关于这些等级和 FIPS 标准的其他说明，请见http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf完整 FIPS 140-2标准。

7.1.1. 启动 FIPS 模式

为了让红帽 Linux 系统符合 FIPS （联邦信息处理标准）的 140-2 出版物，您需要作出一些修改以确保得到认可的加密模块得到使用。将您的系统（内核或者用户空间）切换到 FIPS 模式，请遵守以下步骤：

1. 为了适当核查操作插入式模块完整性，预链接不得不废止。这可以通过设置**/etc/sysconfig/prelink** 配置文件中的**PRELINKING=no** 来完成。如有任何既存的预链接，应该使用 **prelink -u -a**要求，在所有系统文件中解除。
2. 接下来，安装 *dracut-fips*包：

```
~]# yum install dracut-fips
```

3. 重命名**initramfs**文档：

```
~]# dracut -f
```

警告

这个操作会改写既存的**initramfs**文档。

4. 修改在 **/boot/grub/grub.conf** 文档中当前内核的内核命令行，添加以下选项：

```
fips=1
```

注意

如果 **/boot** 或者 **/boot/efi** 位于单独的分区，内核参数***boot=<partition of /boot or /boot/efi>*** 必须被添加到内核命令行。你可以分别通过运行 **df /boot** 或者 **df /boot/efi**这两个指令。

```
~]$ df /boot
Filesystem           1K-blocks      Used Available Use%
Mounted on
/dev/sda1               495844     53780    416464  12% /boot
```

即使是在启动时设备名称改变，通过运行以下要求来辨认 UUID （通用唯一标准）以确保 the ***boot=*** 参数选择仍然运作。

```
~]$ blkid /dev/sda1
/dev/sda1: UUID="05c000f1-f899-467b-a4d9-d5ca4424c797"
TYPE="ext4"
```

根据上述例子，以下字符串需要被附到内核命令行上；

```
boot=UUID=05c000f1-f899-467b-a4d9-d5ca4424c797
```

5. 重启系统。

您应该需要严格的 FIPS 合规性，**fips=1** 内核选项需要在系统安装时被添加到内核命令行，因此密码的生成是通过计算法则以及运作中持续的监控而完成的。用户应该同样通过移动鼠标以确保系统在安装过程中有足够的信息熵，或如果没有鼠标，也要确保可以多次按下按键。对于敲击键盘的次数应该多于 256 次。少于 256 次的键盘敲击次数会产生非唯一的钥匙。

7.2. 国家工业安全计划操作手册

NISPOM（也称DoD 5220.22-M），作为 NISP （国家行业安全项目）的构成部分，为所有的政府承包者关于分类信息而建立了一系列的程序和要求。当前的 NISPOM 于2006年2月28日更新，主要变化从2013年3月28日更改。 NISPOM 文档可以从以下链接中下载：http://www.nispom.org/NISPOM-download.html。

7.3. 支付卡行业数据安全标准

https://www.pcisecuritystandards.org/about/index.shtml：*PCI 安全标准委员会是一个开放的全球论坛，成立于 2006 年，它负责 PCI 安全标准的开发、管理、培训以及普及，其中包括数据安全标准（DSS）。*

您可以在 https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml 下载 PCI DSS 标准。

7.4. 安全技术实施指南

《安全技术实施指南》（或称 STIG，Security Technical Implementation Guide）是计算机软件和硬件标准化安全安装的方法。

更多关于 STIG 的信息请看以下链接：http://iase.disa.mil/stigs/index.html。

加密标准

A.1. 同步加密

A.1.1. 高级加密标准 — AES

在加密中， AES （高级加密标准）是被美国政府所采用的加密标准。这一标准包括三大块密码，AES-128, AES-192 和 AES-256, 通过一个更大的合集并最初发表为 Rijindael。每个 AES 密码都有128个比特位大小，秘钥位大小分别是128、192和256比特。AES 密码被广泛分析，目前被全球使用，正如其前一代 DES （数据加密标准）一样。 [3]

A.1.1.1. AES 历史

AES 是由 NIST （美国国家标准技术研究所）在经过5年标准化过程，于2001年11月26日在第197份出版物中发布的。在 Rijindael 的设计被选择之前，15个竞争设计会被展示和评选出最适合的一个。在2002年5月26日被作为一个标准而有效使用。这个设计在很多不同的加密包中被使用。 AES 是第一个被 NSA （美国国家安全局）通过的作为顶级机密公开使用的密码。（详见下方 AES 安全体系） [4]

Rijindael 密码由两位比利时编码者 Joan Daemen （琼·德门）和 Vincent Rijmen （文森特·瑞捷门）共同开发，并由他们递交到 AES 筛选过程。 Rijindael 是两个开发者姓名的混合。 [5]

A.1.2. 数据加密标准 — DES

数据加密标准（DES）是由国家标准局为美国在 1976 年选择作为联邦官方信息处理标准（FIPS）的块密码（共享秘密加密的形式），随后在国际上广为应用。它的依据是使用 56 位密钥的对称密钥算法。该算法最初与分类别的设计元素不符，密钥长度较短，并被怀疑是国家安全局的（NSA）的后门。DES 后经大量的学术研究，这些研究引发了对块加密以及加密分析的现代化理解。 [6]

A.1.2.1. DES 历史

DES 目前对于很多程序来说都不安全。主要是因为56比特秘钥长度太短。在1999年1月，distributed.net网站和电子前沿基金会联合在2小时15分钟内公开破解了一个 DES 秘钥。尽管它们在实际操作中不易被装载，也有一些同样的分析结果证实了密码的理论性弱点。尽管有很多理论攻击，但以三位DES 编码的计算被认为实际上很安全。在近些年，密码已经被 AES（高级加密标准）取代。 [7]

在一些文档中，在作为标准的 DES 和被称为 DEA （数据加密计算法则）的运算法则之间是有区别的。 [8]

A.2. 公钥加密

公钥加密是一种由很多加密算法和加密系统采用的加密方法，其与众不同之处在于使用不对称密钥算法，而不是使用，或者另外使用对称密钥算法。使用公钥-私钥密钥技术已经让很多之前未知的保护通讯或者验证信息的方法变为现实。它们不要求使用对称密钥算法时需要的一个或者多个保密密钥的安全初始交换。它还可用于创建数字签名。 [9]

公钥加密法是世界广泛使用的基础技术，是类似传输层安全性（TLS，SSL 的后续）、PGP 和 GPG 等互联网标准基础的加密方法。 [10]

在公钥加密法中使用的特别技术是不对称密钥算法，这个算法中用来加密信息的密钥与解密信息的密钥不是同一个。每个用户都有一对加密密钥 — 一个公钥和一个私钥。私钥是秘密保存，而公钥则会广泛传播。信息是使由接收方到公钥加密，且该信息只能使用对应的私钥解密。两个密钥间是数学计算关系，但很难在例如具体情况或者项目实践中使用公钥演算出私钥。这个算法的发现对从二十世纪七十年代开始使用的加密法实践有革

命性影响。 [11]

相比而言，对称密钥算法及其变体已经使用了几千年，该方法由发送者和接受者共享一个保密密钥（该密钥还必须保密，因此还要考虑常用术语带来的争议），此保密秘钥同时用于加密和解密。要使用对称加密方案，发送者和接受者必须事先安全地共享保密密钥。 [12]

因为对称密钥算法所需计算量最小，通常可使用密钥交换算法互换密钥，并使用那个密钥和对称密钥算法传送数据。例如： PGP 和 SSL/ TLS 产品线可这样做，结果是可生成混合的密码系统。 [13]

A.2.1. Diffie-Hellman

Diffie-Hellman 密钥交换（D-H）是可让事先彼此不了解的双方通过不稳定的通讯频道联合建立共享保密密钥的加密协议。可使用这个密钥加密以后使用对称密钥密码的通讯。 [14]

A.2.1.1. Diffie-Hellman 历史

该方案是由 Whit输入栏 Diffie 和 Martin Hellman 于 1976 年首次发布，虽然它晚于 GCHQ （英国信号情报机构）的 Malcolm J. Williamson 发明，但英国一直将该发明列为机密。2002 年，Hellman 建议将该算法改名为 Diffie–Hellman–Merkle 密钥交换以纪念 Ralph Merkle's 对发明公钥加密法的贡献（Hellman 2002） [15]

虽然 Diffie–Hellman 密钥协议本身是匿名（非认证）密钥合约协议，但它为各种认证协议提供了基础，并用来在传输层安全性的短期模式中提供最佳转发保密（根据密码组件可以时 EDH 或者 DHE）。 [16]

现在已经过期的美国专利 4,200,770 论述了这个算法，并给予作为发明者的 Hellman、Diffie 和 Merkle 很高评价。 [17]

A.2.2. RSA

在密码学中，RSA（代表着首次公开描述它的 Rivest 李威斯特、Shamir 沙米尔和 Adleman 阿德尔曼）是一个对于公钥加密的计算法则。这个法则是第一个被认作适合和加密一样适合认证的计算法则，并且是在公钥密码学中第一个最大的进步之一。RSA 在电子商务协议中被广泛使用，并且因为足够长的钥匙以及及时的安装启动而被认为很安全。

A.2.3. DSA

DSA （数字签名算法）是数字签名的标准，即美国联邦政府用于数字签名的标准。DSA 只可用于签名，且不是加密算法。 [18]

A.2.4. SSL/TLS

传输层安全性（TLS）及其之前的产品安全套接字层（SSL）都是可为通过网络（比如互联网）进行的通讯提供安全性的加密法协议。TLS 和 SSL 在传输层端到端加密网络链接片段。

很多版本的协议被应用程序广泛使用，如，网页浏览、电子邮件、互联网传真、即时短信以及 VoIP （IP语音呼叫）。 [19]

A.2.5. Cramer-Shoup 加密系统

Cramer–Shoup 系统是一个非对称秘钥加密算法，而且被证实是第一个针对适应性选择米文攻击所采用的标准加密猜想所用的安全有效的项目。其安全性是基于决定性Diffie–Hellman猜想的计算难解性（广泛接受，但未被证实）的计算难解性的。由 Ronald Cramer （罗纳德·克莱默）和 Victor Shoup （维克多·苏伯）于

1998年研发，是 ElGamal 加密系统的延伸。与 ElGamal 相反，它具有很强的延展性，Cramer–Shoup 添加了另外的成分来确保甚至是遭受广泛的攻击时保证其非延展性。这一非延展性是通过使用抗撞击哈希函数和额外的计算而取得的，导致了密文是 ElGamal 的两倍大。 [20]

A.2.6. ElGamal 加密

在密码学中，ElGamal加密体系基于Diffie-Hellman加密协议的公钥加密的非对称秘钥加密算法。有Tahel ElGamal在1985年描述。ElGamal 加密被用于免费的GNU隐私保护软件、PGP 近期的版本和其他密码系统中。 [21]

---

[3] "高级加密标准。"*维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[4] "高级加密标准。"*维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[5] "高级加密标准。"*维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[6] "数据加密标准。"*维基百科。* 2009 年 11 月 14 日http://en.wikipedia.org/wiki/Data_Encryption_Standard

[7] "数据加密标准。"*维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[8] "数据加密标准。"*维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[9] "公钥加密" *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[10] "公钥加密" *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[11] "公钥加密" *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[12] "公钥加密" *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[13] "公钥加密" *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[14] "Diffie-Hellman." *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[15] "Diffie-Hellman." *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[16] "Diffie-Hellman." *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[17] "Diffie-Hellman." *维基百科。* 2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[18] "DSA." *维基百科。* 2010 年 2 月 24 日http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[19] "TLS/SSL。" *维基百科。* 2010 年 2 月 24 日 http://en.wikipedia.org/wiki/Transport_Layer_Security

[20] "Cramer-Shoup cryptosystem。" *维基百科。* 2010 年 2 月 24 日 http://en.wikipedia.org/wiki/Cramer–Shoup_cryptosystem

[21] "ElGamal encryption" *维基百科。* 2010 年 2 月 24 日 http://en.wikipedia.org/wiki/ElGamal_encryption

审核系统引用

B.1. 审核事件字段

表 B.1 “事件字段” 列举了目前所有获得支持的审核事件字段。一个事件字段是在审计日志文档中等号前的值。

表 B.1. 事件字段

事件字段	解释
a0, **a1**, **a2**, **a3**	记录系统调用前四个参数，标为十六进制。
acct	记录用户账号名。
addr	记录 IPv4 或者 IPv6 地址，这一字段通常紧随一个 **hostname** 字段并包含主机名解释地址。
arch	记录关于中央处理器结构的系统信息，以十六进制编码。
auid	记录审核用户身份。当用户身份改变的时候，身份在登录时被指定给另一个用户并在每次操作时继承（例如，用 **su - john** 切换用户名）。
capability	记下二进制位的数字，用来设置 Linux 功能。 更多关于 Linux 功能的信息，参见 capabilities(7) 页。
cap_fi	记录的数据与文件系统功能设置。
cap_fp	记录与设置一个与许可文件系统功能相关的数据。
cap_pe	记下与设置有效处理功能相关的数据。
cap_pi	记下与设置继承处理功能相关的数据。
cap_pp	记下与设置许可处理功能相关的数据。
cap_pp	记下 **cgroup** 的路径，其中包含处理审计时间生产的处理。
cap_pp	记下整条执行的命令行。这在 shell 解释程序在 **exe** 输入栏 record 领域记录时有用，例如，**/bin/bash** 在 shell 解释程序和 **cmd** 其余执行的命令行字段记录，例如， **helloworld.sh --help**.
comm	记录被执行的要求。这在 **exe** 的字段记录在 shell 解译程序中很有用，例如**/bin/bash** 作为套解译程序以及 **comm** 字段记录的脚本被执行，或者在执行**helloworld.sh** 时很有用。
cwd	记录系统调用目录路径。
data	记下TTY记录相关数据。
dev	记录设备中的次要和主要ID，包含事件记录的文件或目录。
devmajor	记录主要设备ID。
devminor	记录次要设备ID。
egid	记录开始分析进程用户的有效组ID。
euid	记录开始分析进程用户的有效用户ID。
exe	记录曾调用分析进程的可执行路径。
exit	记录由系统调用返回的退出代码。此值随系统调动变化。可以将此值解释给它的人类可以读得懂的对应值，根据以下命令：**ausearch --interpret --exit *exit_code***
family	记录使用的地址协议类型，IPv4 还是 IPv6。
filetype	记录文件类型。
flags	记录文件系统名标志。
fsgid	记录开始分析进程的用户文件系统组ID。
fsuid	记录开始分析进程的用户文件系统用户 ID。
gid	记录组 ID。
hostname	记录主机名称。
icmptype	记录接收到的 ICMP （因特网信息控制协议）包类型。审核包含这一字段的信息，通常由**iptables** 生成。

事件字段	解释
id	记录变动账户的用户 ID。
inode	记录在审核时，关联文件或目录的 inode 数字。
inode_gid	记录 inode 所有者的组 ID。
inode_uid	记录 inode 所有者的用户 ID。
items	记录路径数目，该记录会被附加。
key	记录与在审核日志中能产生一个特定事件条例相关的用户定义字符串。
list	记录审查规则列表 ID。以下是列表已知的 ID 是： - 0 — **user** - 1 — **task** - 4 — **exit** - 5 — **exclude**
mode	记录文件目录权限，以数值表示法编码。
msg	记录时间戳和记录中的唯一 ID ，或者不同的事件特定 ***<name>=<value>*** 匹配，由内核或用户空间应用提供。
msgtype	记录基于用户的 AVC 拒绝返回的信息类型 。信息类型由 D-Bus 决定。
name	记录以自变量形式传到系统调用的全部文件或目录路径。
new-disk	记录分配给虚拟机的新磁盘资源名字。
new-mem	记录分配给虚拟机的新记忆资源数量。
new-vcpu	记录分配给虚拟机的新虚拟 CPU 资源的数量。
new-net	记录分配给虚拟机的新网络界面资源的 MAC 地址。
new_gid	记录指定给用户的组 ID。
oauid	记录登录进入系统用户的用户 ID（预期相对的是，例如，使用**su**）并开始目标进程。这一字段专用于记录类型**OBJ_PID**。
ocomm	记录用来启动目标进程的命令。这一字段专用于记录类型 **OBJ_PID**。
opid	记录目标进程的进程 ID。这一字段专用于记录类型 **OBJ_PID**。
oses	记录目标进程的会话 ID。这一字段专属于记录类型 **OBJ_PID**。
ouid	记录目标进程中的真实用户 ID。
obj	记录以 SELinux 为对象的内容，该对象可以是文档、目录、承接，或任何一个被动接受的物体。
obj_gid	记录对象的组 ID。
obj_lev_high	记录对象的高 SELinux 级别。
obj_lev_low	记录对象的低 SELinux 级别。
obj_role	记录对象的 SELinux 角色。
obj_uid	记录对象的 UID。
obj_user	记录对象的关联用户。
ogid	记录对象所有者的群组 ID。
old-disk	当一个新的磁盘资源被分配给一个虚拟机时记录旧磁盘资源的名字。
old-mem	当新内存额被指定给虚拟机时，记录旧的内存资源总额。
old-vcpu	当新虚拟 CPU 被指定给虚拟机，记录旧虚拟 CPU 资源。
old-net	当新的网络界面被指定给虚拟机时，记录 MAC 地址的旧网络界面资源。
old_prom	记录网络混杂标志的上一个值。
ouid	记录开始目标进程的真实用户 ID。
path	记录以自变量形式传到系统调用的全部文件或目录路径。
perm	记录曾调用分析进程的可执行路径。（即，读、写、操作或者变更属性）

事件字段	解释
pid	**pid**语义场取决于该领域起源的价值。 由用户控件衍生出来的领域，该领域拥有一个操作 ID。 在由内核衍生的领域，该领域拥有一个线程号。线程号等同于单线进程中的进程标识。注明该线程号的值和 pthread_t 在用户空间账号所用值是不一样的。更多信息，参考 gettid(2) 操作说明。
ppid	记录父进程标识（Parent PID）。
prom	记录网络简短社交活动标志。
proto	记录曾用的网络协议。这个领域是专门针对由 **iptables** 衍生的审核事件的。
res	记录引发审核事件操作的结果。
result	记录引发审核事件操作的结果。
saddr	记录套接地址。
sauid	记录发送者的审计登录用户账号。这个账号由作为内核的 D-Bus 提供，并不能看到哪个用户发送了最初的**auid**。
ses	记录会话 ID，在该会话中分析过程被调用。
sgid	记录开始分析进程用户的群组 ID。
sig	记录导致项目异常终止的信号代码。通常，这是一个系统被入侵的迹象。
subj	记录以 SELinnux 为对象的内容，这个对象可以是一个过程、一个用户、或者是任何对目标起作用的事物。
subj_clr	记录 SELinux 对象的清除许可。
subj_role	记录对象的 SELinux 的角色。
subj_sen	记录对象 SELinux 的敏感度。
subj_user	记录和主题相关的用户。
success	记录系统调用是成功还是失败。
suid	记录开启分析过程用户的设置用户 ID。
syscall	记录发送到内核的系统调用类型。
terminal	记录终端名称（不包括**/dev/**）。
tty	记录控制终端的名称。如果过程没有控制终端，则用 **(none)**的值。
uid	记录启动分析进程的用户的真实 ID。
vm	记录产生审核事件的虚拟机名称。

B.2. 审核记录类型

表 B.2 “记录类型” 列举了所有支持当前审核记录的类型。事件类型在每个审核记录开头的**type=** 的字段中被指定。

表 B.2. 记录类型

事件类型	解释
ADD_GROUP	当添加用户空间组时被触发。
ADD_USER	当添加用户空间的用户账号时被触发。
ANOM_ABEND [a]	当一个过程非正常终止（如果被启用，一个信号会导致核心转储）时被触发。
ANOM_ACCESS_FS[a]	当文档或目录访问非正常终止时被触发。
ANOM_ADD_ACCT[a]	当用户空间账号添加非正常终止时被触发。
ANOM_AMTU_FAIL[a]	当 AMTU（AMTU 抽象机器测试工具）的失败被检测到时被触发。

事件类型	解释
ANOM_CRYPTO_FAIL[a]	当加密系统的失败被检测到时被触发。
ANOM_DEL_ACCT[a]	当用户空间账号删除非正常终止时被触发。
ANOM_EXEC[a]	当文档的执行非正常终止时被触发。
ANOM_LOGIN_ACCT[a]	当尝试登录账号非正常终止时被触发。
ANOM_LOGIN_FAILURES[a]	当已达到失败登录的限制时被触发。
ANOM_LOGIN_LOCATION[a]	当登录尝试发生在禁区时被触发。
ANOM_LOGIN_SESSIONS[a]	当登录尝试达到最大并发会话量时被触发。
ANOM_LOGIN_TIME[a]	当登录尝试在某时被例如 **pam_time** 阻止时被触发。
ANOM_MAX_DAC[a]	当达到 DAC （自定义访问控制）失败的最大值时被触发。
ANOM_MAX_MAC[a]	当已达到 MAC （强制访问控制）的最大量时被触发。
ANOM_MK_EXEC[a]	当文档可执行时被触发。
ANOM_MOD_ACCT[a]	当用户空间账号的修改非正常终止时被触发。
ANOM_PROMISCUOUS[a]	当启用或停用混杂模式时被触发。
ANOM_RBAC_FAIL[a]	当检测到 RBAC（基于角色访问控制）自测失败时被触发。
ANOM_RBAC_INTEGRITY_FAIL[a]	当检测到 RBAC （基于角色访问控制）文档完整性测试失败时被触发。
ANOM_ROOT_TRANS[a]	当用户变成根用户时被触发。
AVC	被触发后以记录 SELinux 的权限检查。
AVC_PATH	当 SELinux 进行权限检查时被触发来记录 **dentry** 和**vfsmount** 组。
BPRM_FCAPS	当用户以文档系统的许可范围来执行一个项目时被触发。
CAPSET	被触发来记录为基于过程而设置的能力，例如，作为根用户运行时抵御的能力。
CHGRP_ID	当用户空间群组 ID 被改变时被触发。
CHUSER_ID	当用户空间用户 ID 被改变时被触发。
CONFIG_CHANGE	当审核系统配置被修改时被触发。
CRED_ACQ	当用户需要用户空间凭证时被触发。
CRED_DISP	当用户释放用户空间凭据时被触发。
CRED_REFR	当用户刷新其用户空间凭据时被触发。
CRYPTO_FAILURE_USER	当解密、加密或随机加密操作失败时被触发。
CRYPTO_KEY_USER	被触发以记录用于加密目的的密钥标示符。
CRYPTO_LOGIN	当加密管理员登录尝试被觉察时被触发。
CRYPTO_LOGOUT	当加密管理员注销尝试被觉察时被触发。
CRYPTO_PARAM_CHANGE_USER	当加密参数改变被觉察时被触发。
CRYPTO_REPLAY_USER	当重播攻击被觉察时被触发。
CRYPTO_SESSION	被触发以记录在 TLS 会话建立过程中的参数。
CRYPTO_TEST_USER	被触发来记录根据 FIPS-140 标准要求的加密测试结果。
CWD	被触发来记录当前工作目录。
DAC_CHECK	被触发来记录 DAC 检查结果。
DAEMON_ABORT	当守护程序因为错误终止时被触发。
DAEMON_ACCEPT	当 **auditd** 守护程序接受远程连接时被触发。

事件类型	解释
DAEMON_CLOSE	当 **auditd** 守护程序关闭远程连接时被触发。
DAEMON_CONFIG	当守护程序配置变换被检测到时被触发。
DAEMON_END	当守护程序被成功停止时被触发。
DAEMON_RESUME	当 **auditd** 守护程序恢复登录时被触发。
DAEMON_ROTATE	当 **auditd** 守护程序切换审核日志文件时被触发。
DAEMON_START	当 **auditd** 守护程序启动时被触发。
DEL_GROUP	当用户空间组被删除时被触发。
DEL_USER	当用户空间用户被删除时被触发。
DEV_ALLOC	当设备被分配时被触发。
DEV_DEALLOC	当设备被释放时被触发。
EOE	被触发以记载多记录事件。
EXECVE	被触发以记录 **execve(2)** 系统调用的参数。
FD_PAIR	触发以记录使用 **pipe** 和 **socketpair** 的系统调用。
FS_RELABEL	当文件系统重新标记的操作被检测到时被触发。
GRP_AUTH	当一组密码被用来验证用户空间组时被触发。
INTEGRITY_DATA [b]	被触发以记录由内核运作的数据完整性验证事件。
INTEGRITY_HASH[b]	被触发以记录由内核运作的散列型完整性验证事件。
INTEGRITY_METADATA[b]	被触发以记录由内核运作的元数据完整性验证事件。
INTEGRITY_PCR[b]	被触发以记录 PCR（平台配置寄存器）的无效信息。
INTEGRITY_RULE[b]	被触发以记录政策规则。
INTEGRITY_STATUS[b]	被触发以记录完整性验证的状态。
IPC	被触发以记录关于由系统调用的关于进程间通信对象的信息。
IPC_SET_PERM	被触发以记录 **IPC_SET** 关于一个 IPC 客体的管理操作设定的新值的相关信息。
KERNEL	被触发以记录审核系统的初始化。
KERNEL_OTHER	被触发以记录第三方内核模块的信息。
LABEL_LEVEL_CHANGE	当对象的层次结构被修改时被触发。
LABEL_OVERRIDE	当管理员重写对象的层次结构时被触发。
LOGIN	当用户登录并进入系统时被触发以记录相关登录信息。
MAC_CIPSOV4_ADD	当 CIPSO（商业网络条款安全选项）用户名增加了新的 DOI（域名解释）时被触发。增加 DOI（域名解释）是由 NetLabel 提供的内核组合标签容量的一部分。
MAC_CIPSOV4_DEL	当一个 CIPSO 用户名删除了已存在的 DOI。增加 DOI 是由 NetLabel 提供的内核组合标签容量的一部分。
MAC_CONFIG_CHANGE	当一个 SELinux 的布尔值被改变时被触发。
MAC_IPSEC_EVENT	当一个 IPSec 的事件被检测到或者当 IPSec 配置改变时被触发以记录信息。
MAC_MAP_ADD	当一个新的 LSM（Linux 安全模式）的域映射被添加时被触发。LSM 域映射是由 NetLabel 提供的内核组合标签容量的一部分。
MAC_MAP_DEL	当一个几寸的域映射被添加时被触发。LSM 域映射是由 NetLabel 提供的内核组合标签容量的一部分。
MAC_POLICY_LOAD	当 SELinux 政策文档被加载时被触发。
MAC_STATUS	当 SELinux 模式（启动、批准、关闭）被改变时被触发。
MAC_UNLBL_ALLOW	当使用由 NetLabel 提供的内核组合标签容量时且未贴标的流量被允许时被触发。
MAC_UNLBL_STCADD	当使用由 NetLabel 提供的内核组合标签容量并静态贴标被添加时被触发。

事件类型	解释
MAC_UNLBL_STCDEL	当使用由 NetLabel 提供的内核组合标签容量且一个静态贴标被删除时被触发。
MMAP	被触发以记录文档的描述符以及**mmap(2)**系统调用的标志。
MQ_GETSETATTR	被触发以记录**mq_getattr(3)**和**mq_setattr(3)**的信息队列特性。
MQ_NOTIFY	被触发以记录 **mq_notify(3)**系统调用的参数。
MQ_OPEN	被触发以记录 **mq_open(3)**系统调用的参数。
MQ_SENDRECV	被触发以记录 **mq_send(3)**和 **mq_receive(3)** 系统调用的参数。
NETFILTER_CFG	当网络过滤器的链修改被检测到时被触发。
NETFILTER_PKT	被触发以记录遍历网络过滤器链的数据包。
OBJ_PID	被触发以记录关于信号被发出的过程信息。
PATH	被触发以记录文档名字路径信息。
RESP_ACCT_LOCK [c]	当用户账号被锁定时被触发。
RESP_ACCT_LOCK_TIMED[c]	当用户账号在一个特定时间内被锁定时被触发。
RESP_ACCT_REMOTE[c]	当用户账号被远程锁定时被触发。
RESP_ACCT_UNLOCK_TIMED[c]	当用户账号在已配置的时间后被解锁时被触发。
RESP_ALERT[c]	当警报电子邮件被发送时被触发。
RESP_ANOMALY[c]	当一个异常没有在操作时被触发。
RESP_EXEC[c]	当一个入侵检测项目对于源于项目执行的威胁做出反应时被触发。
RESP_HALT[c]	当系统被关闭时被触发。
RESP_KILL_PROC[c]	当进程被终止时被触发。
RESP_SEBOOL[c]	当 SELinux 的布尔值被设置时被触发。
RESP_SINGLE[c]	当系统被设定为单一用户模式时被触发。
RESP_TERM_ACCESS[c]	当会话终止时被触发。
RESP_TERM_LOCK[c]	当终端被锁定时被触发。
ROLE_ASSIGN	当管理员指定了一个用户的 SELinux 角色时被触发。
ROLE_MODIFY	当管理员修改一个 SELinux 角色时被触发。
ROLE_REMOVE	当管理员从 SELinux 角色中将用户名移除时被触发。
SELinux_ERR	当内部 SELinux 错误被检测到时被触发。
SERVICE_START	当服务启动时被触发。
SERVICE_STOP	当服务停止时被触发。
SOCKADDR	被触发以记录套接字地址或者被系统调用调回。
SOCKETCALL	被触发以记录 **sys_socketcall** 系统调用的参数（被用来复合多数套接字相关的系统调用）。
SYSCALL	被触发以记录内核的系统调用。
SYSTEM_BOOT	当系统被启动时被触发。
SYSTEM_RUNLEVEL	当系统的允许水平被改变时被触发。
SYSTEM_SHUTDOWN	当系统被关闭时被触发。
TEST	被触发以记录测试信息的成功值。
TRUSTED_APP	此种类型的记录可以被需要审核的第三方应用使用。
TTY	当 TTY 输入被发送到管理过程时被触发。
USER_ACCT	当用户空间用户账号被修改时被触发。
USER_AUTH	当用户空间的身份验证尝试被检测到时被触发。
USER_AVC	当用户空间的 AVC 信息生成时被触发。

事件类型	解释
USER_CHAUTHTOK	当用户账号特性被修改时被触发。
USER_CMD	当用户空间的 shell 命令被执行时被触发。
USER_END	当用户空间会话被终止时被触发。
USER_ERR	当用户账号状态错误被检测到时被触发。
USER_LABELED_EXPORT	当一个对象被导出了 SELinux 标签时被触发。
USER_LOGIN	当用户登录时被触发。
USER_LOGOUT	当用户注销时被触发。
USER_MAC_POLICY_LOAD	当用户空间的守护程序在上载一项 SELinux 政策时被触发。
USER_MGMT	被触发以记录用户空间管理数据。
USER_ROLE_CHANGE	当用户的 SELinux 角色被改变时被触发。
USER_ SELinux_ERR	当用户空间 SELinux 错误被检测到时被触发。
USER_START	当用户空间会话开始时被触发。
USER_TTY	当关于 TTY 输入到一个管理过程的解释信息是从用户空间发送时被触发。
USER_UNLABELED_EXPORT	当对象被导出并没有 SELinux 标签时被触发。
USYS_CONFIG	当用户空间系统的参数变化被检测到时被触发。
VIRT_CONTROL	当虚拟机器被启动、暂停或停止时被触发。
VIRT_MACHINE_ID	被触发以记录虚拟机器的标签绑定。
VIRT_RESOURCE	被触发以记录虚拟机器的资源配置。

[a] 所有预制**ANOM**的审核事件类型都预计被入侵检测程序处理。
[b] 这个事件类型与 IMA（完整性度量架构）有关，并与 TPM（可信平台模块）芯片运行得最好。
[c] 所有预置**RESP**的审核事件类型都是设定好的对于预防其检测出系统内有恶性事件对于入侵检测系统的反应。

修订历史

修订 1-14.1.1	Thu Nov 20 2014	Chester Cheng

说明：翻译完成。
翻译、校对：陈坤、邓明晗、吴洁蕾、张可。
校对、编辑：任浩。
校对、责任编辑：鄭中。
附注：本简体中文版来自「**红帽工程部翻译中心**」与「**澳大利亚昆士兰大学笔译暨口译研究所**」之合作计划。若有疏漏之处，期盼各方先进透过以下网址，给予支持指正：https://bugzilla.redhat.com/。

修订 1-14.1	Tue Jun 03 2014	Tomáš Čapek

7.0 GA 发布版本

修订 1-12.35	Tue May 20 2014	Tomáš Čapek

为风格更改进行重建

修订 1-12	Tue, Mar 05 2013	Martin Prpič

本书的最初创作