Red Hat Enterprise Linux 7

安全指南

保护 RHEL 服务器和工作站的概念和技术

Last Updated: 2024-02-28

Red Hat Enterprise Linux 7 安全指南

保护 RHEL 服务器和工作站的概念和技术

Mirek Jahoda
Red Hat Customer Content Services
mjahoda@redhat.com

Jan Fiala
Red Hat Customer Content Services
jafiala@redhat.com

Stephen Wadeley
Red Hat Customer Content Services

Robert Krátký
Red Hat Customer Content Services

Martin Prpič
Red Hat Customer Content Services

Ioanna Gkioka
Red Hat Customer Content Services

Tomáš Čapek
Red Hat Customer Content Services

Yoana Ruseva
Red Hat Customer Content Services

Miroslav Svoboda
Red Hat Customer Content Services

法律通告

摘要

本书帮助用户和管理员学习保护工作站和服务器免受本地和远程入侵、利用和恶意活动的流程和实践。 侧重于 Red Hat Enterprise Linux，但详细介绍了适用于所有 Linux 系统的概念和技术，本指南详细介绍了为数据中心、工作区和家庭创建安全的计算环境所涉及的规划和工具。 通过拥有正确的管理知识、对安全的重视及相关的工具，Linux 系统可以完全正常工作，并防止大多数安全入侵和攻击。

目录

第 1 章 安全主题概述

由于日益依赖于强大的网络计算机来帮助经营业务并跟踪个人信息，整个行业围绕着网络和计算机安全实践而建立起来。企业已征求安全专家的知识和技能来适当地审核系统和定制解决方案，以满足组织的运营要求。因为大多数机构动态程度更高，所以相关员工会在本地和远程访问关键的公司 IT 资源，因此对安全计算环境的需求也随之变得更高。

不幸的是，许多组织（以及个人用户）将安全性视为事后考虑的事情，是一种为提高能力、生产力、便利性、易用性和预算问题而被忽略的流程。适当的安全实现通常是在事后制定的 – 在发生未授权入侵 *后*。在将站点连接到不可信网络（如互联网）之前，采取正确的措施是抵御入侵尝试的有效方法。

注意

本文档对 **/lib** 目录中的文件进行了一些引用。使用 64 位系统时，提到的一些文件可能位于 **/lib64** 中。

1.1. 什么是计算机安全性？

计算机安全性是一个涵盖计算和信息处理范围的一般术语。依靠计算机系统和网络进行日常业务交易和访问重要信息的行业将数据视为其整体资产的重要组成部分。一些术语和指标已进入我们日常的业务词汇，如总拥有成本(TCO)、投资回报(ROI)和服务质量(QoS)。借助这些指标，行业可以核算数据完整性和高可用性(HA)等方面，来作为规划和流程管理成本的一部分。在电子商务等行业中，数据的可用性和可信赖性可能意味着成功与失败的区别。

1.1.1. 标准化安全

每个行业中的企业都依赖制定标准的机构（如美国医疗协会(AMA)或电气与电子工程师协会(IEEE)）所制定的法规和规则。同样的理念也适用于信息安全。许多安全顾问和供应商都同意称为 CIA，或 *机密性*、*完整性和可用性* 的标准安全模型。这种三层模式是普遍认可的组件，用于评估敏感信息的风险和建立安全策略。下面进一步详细描述了 CIA 模型：

- 机密性 - 敏感信息必须只对一组预定义的个人可用。应限制未经授权的信息的传播和使用。例如，信息的机密性确保客户的个人或财务信息不会被未经授权的人出于恶意目的而获得，如身份失窃或信用欺诈。
- 完整性 - 不应以导致信息不完整或不正确的方式更改信息。应限制未经授权的用户修改或销毁敏感信息的能力。
- 可用性 - 被授权的用户可随时根据需要访问信息。可用性是一种保证，即可以按照商定的频率和及时性获得信息。这通常以百分比来衡量，并在网络服务提供商及其企业客户的服务级别协议(SLA)中正式约定。

1.1.2. 加密软件和认证

以下红帽知识库文章概述了 Red Hat Enterprise Linux 核心加密组件、记录它们是什么、如何选择它们、它们是如何集成到操作系统中的、它们如何支持硬件安全模块和智能卡，以及加密证书如何应用于它们。

- RHEL7 核心加密组件

1.2. 安全控制

计算机安全性通常被分为三个不同的主要类别，通常称为 *控制* ：

- 物理的

- 技术的
- 管理的

这三大类定义了适当安全实现的主要目标。这些控制中有一些子类，其进一步详细说明了控制及如何实现它们。

1.2.1. 物理控制

物理控制是在定义的结构中实施的安全措施，用来阻止或防止未经授权访问敏感材料。物理控制的示例如下：

- 闭路监控摄像机
- 运动或热报警系统
- 安全保护
- 照片 ID
- 金属门锁定
- 生物统计学（包括指纹、声音、脸部、虹膜、笔迹和其他用于识别个人的自动方法）。

1.2.2. 技术控制

技术控制使用技术作为基础，来控制通过物理结构和网络对敏感数据的访问和使用。技术控制范围很广，包含如下技术：

- 加密
- 智能卡
- 网络验证
- 访问控制
- 文件完整性审核软件

1.2.3. 管理控制

管理控制确定了安全的人为因素。它们涉及机构内各级人员，并确定哪些用户可以通过以下方式访问哪些资源和信息：

- 培训并认知
- 灾难和恢复计划
- 人员与隔离策略
- 人员注册和核算

1.3. 漏洞评估

只要有时间、资源和动机，攻击者几乎可以侵入任何系统。当前提供的所有安全流程和技术都无法保证所有系统完全安全，不受入侵。路由器有助于保护通往互联网的网关。防火墙有助于保护网络边缘。虚拟专

用网络以加密流的方式安全地传输数据。入侵检测系统对恶意活动发出警告。然而，每项技术的成功取决于许多变量，包括：

- 负责配置、监控和维护技术的人员的专业技能.
- 能够快速高效地修补和更新服务及内核的能力。
- 负责人员对网络时刻保持警觉的能力。

考虑到数据系统和技术的动态状态，保护企业资源可能非常复杂。由于这种复杂性，通常很难为所有系统找到专家资源。虽然有可能拥有在许多信息安全领域具有高水平知识的人员，但很难留住在多个主题领域都是专家的人员。这主要是因为信息安全的每个主题领域都需要持续关注和专注。信息安全不会停滞不前。

漏洞评估是对网络和系统安全性的内部审计；其结果表示网络的机密性、完整性和可用性（如 第 1.1.1 节 "标准化安全"中所述）。通常，漏洞评估从勘察阶段开始，在此期间收集有关目标系统和资源的重要数据。此阶段将进入系统就绪阶段，在此阶段，将对目标进行所有已知漏洞的基本检查。准备阶段在报告阶段达到顶峰，在此阶段，调查结果被分为高、中、低风险类别;并讨论了提高目标安全性(或降低漏洞风险)的方法。

如果您要对您的家进行漏洞评估，您可能会检查您家的每一扇门，看看它们是否关闭和上锁了。您还要检查每个窗口，确保它们完全关闭并插好插销了。同样的概念也适用于系统、网络和电子数据。恶意用户是您数据的窃贼和破坏者。然后，您可以专注于自己的工具、精力和措施来应对恶意用户。

1.3.1. 定义评估和测试

漏洞调查可分为两种类型：*outside looking in* 和 *inside looking around*。

当进行外部漏洞评估时，您会试图从外部破坏您的系统。站在公司的外部，为您提供破解者的观点。您会看到攻击者可以看到的内容 - 可公开路由的 IP 地址、您的 *DMZ* 系统、防火墙的外部接口等等。DMZ 代表"非军事区"，对应一个计算机或小子网络，该网络位于可信内部网络（如公司专用 LAN）与不可信外部网络（如公共互联网）之间。通常，DMZ 包含可供互联网流量访问的设备，如 Web (HTTP)服务器、FTP 服务器、SMTP （电子邮件）服务器和 DNS 服务器。

当您进行内部漏洞评估时，您处于优势地位，因为您是内部的，而且您的状态被提升到可信。这是您和您的同事登录系统后的观点。您会看到打印服务器、文件服务器、数据库和其他资源。

这两种类型的漏洞评估会有分大区别。作为内部公司，为您提供比外部更多的特权。在大多数机构中，安全性被配置为把入侵者挡在外部。在保护组织内部(如部门防火墙、用户级访问控制和内部资源的身份验证流程)方面所做的工作很少。通常，由于大多数系统都是公司内部的，所以在公司内部有更多的资源。一旦您位于公司以外，您的状态就不被信任。外部可用的系统和资源通常非常有限。

漏洞评估和*渗透测试*之间是有区别的。可将漏洞评估作为入渗透试的第一步。从评估中获得的信息用于测试。虽然评估是为了检查漏洞及潜在的漏洞，但渗透测试实际上试图利用这些发现。

设计网络基础结构是一个动态过程。安全性信息和物理安全是动态的。执行评估会显示一个概述，它可能会报告假的正状态和假的负状态。假的正状态代表，攻击发现安全漏洞，但这些漏洞实际并不存在。假的负状态代表，没有发现存在的安全漏洞。

安全管理员的所起到的效果取决于使用的工具及自己所具有的知识。使用目前任何一个评估工具对您的系统运行，几乎可以保证会有一些假的正状态。无论是因为程序错误还是用户错误，其结果都是相同的。工具可能会发现假的正状态，但更严重的是假的负状态。

现在,已定义了漏洞评估与中路测试之间的差异,请仔细检查评估结果,然后先仔细检查插入性测试,作为您最新最佳实践方法的一部分。

警告

不要尝试在生产环境中利用漏洞。这样做会对您的系统和网络的生产率效率造成负面影响。

以下列表检查执行漏洞的一些好处。

- 树立主动关注信息安全的意识。
- 在攻击这发现潜在的漏洞之前，发现潜在的漏洞。
- 使系统保持最新，并应用了补丁程序。
- 在开发专业人士方面促进增长和协助。
- 中止商业损失和负面的公共形象。

1.3.2. 为漏洞评估建立方法论

为了帮助选择用于漏洞评估的工具，建立漏洞评估方法是很有帮助的。不幸的是，目前还没有预定义或行业认可的方法；但是，常识和最佳实践可以作为充分的指南。

*目标是什么？我们是在看一台服务器，还是在看我们的整个网络和网络内的一切？我们是外部还是内部的？*这些问题的答案非常重要，因为它们不仅有助于确定选择哪些工具，而且有助于确定使用这些工具的方式。

要了解有关建立方法的更多信息，请参阅以下网站：

- https://www.owasp.org/ - *开放的 Web 应用程序安全项目*

1.3.3. 漏洞评估工具

评估可以从使用某种形式的信息收集工具开始。评估整个网络时，请首先画出布局，以查找正在运行的主机。定位后，单独检查每个主机。专注于这些主机需要另外一组工具。了解使用哪些工具可能是查找漏洞的最关键步骤。

与日常生活的各个方面一样，有很多执行相同工作的不同工具。这个概念也适用于执行漏洞评估。有特定于操作系统、应用程序甚至网络的工具（根据使用的协议）。有些工具是免费的，有些不是。有些工具比较直观且易于使用，而其他工具比较神秘且文档很少，但具有其他工具没有的功能。

寻找合适的工具可能是一项艰巨的任务，最终经验很重要。如果可能，建立一个测试实验室，并尝试尽可能多的工具，注意每种工具的优缺点。查看工具的 **README** 文件或手册页。此外，可以在互联网上查找更多信息，如文章、分步指南，甚至特定于工具的邮件列表。

以下讨论的工具只是可用工具的一个小抽样。

1.3.3.1. 使用 Nmap 扫描主机

Nmap 是一个流行的工具，可用于确定网络的布局。**Nmap** 已存在多年，可能是收集信息时最常用的工具。其中包括了一个很好的手册页，它提供了其选项和用法的详细描述。管理员可以在网络上使用 **Nmap**，来查找主机系统和这些系统上开放的端口。

Nmap 是漏洞评估中称职的第一步。您可以勾勒出网络中的所有主机，甚至传递一个选项，以允许 Nmap 尝试识别运行在特定主机上的操作系统。Nmap 是使用安全服务并限制未使用服务建立政策的一个良好基础。

要安装 Nmap，请以 **root** 用户身份运行 **yum install nmap** 命令。

1.3.3.1.1. 使用 Nmap

Nmap 可以在 shell 提示符中运行，方法是输入 **nmap** 命令，后跟要扫描的机器的主机名或 **IP** 地址：

> **nmap** *<hostname>*

例如，要扫描主机名为 **foo.example.com** 的机器，请在 shell 提示符下输入以下内容：

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

基本扫描结果（可能需要几分钟时间，具体取决于主机所在的位置和其他网络状况）类似如下：

```
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 测试用于侦听或等待服务的最常见网络通信端口。对于希望关闭不必要的或未使用服务的管理员，这些知识会很有用。

有关使用 Nmap 的更多信息，请参见以下 URL 的官方主页：

http://www.insecure.org/

1.3.3.2. Nessus

Nessus 是一个完整的服务安全扫描程序。Nessus 的插件架构允许用户为其系统和网络自定义它。与任何扫描程序一样，Nessus 就像它所依赖的签名数据库一样。不幸的是，Nessus 经常更新，并具有完整报告、主机扫描和实时漏洞搜索功能。请记住，即使工具中的功能强大且经常更新为 Nessus，也可能会有误报和假的负数。

注意

Nessus 客户端和服务器软件需要使用订阅。本文档中已包含在本文档中，作为可能有兴趣使用此流行应用程序的用户参考。

有关 Nessus 的更多信息，请参见以下 URL 的官方网站：

http://www.nessus.org/

1.3.3.3. OpenVAS

OpenVAS (*开放式漏洞评估系统*)*是一组工具和服务，可用于扫描漏洞和全面的漏洞管理。*OpenVAS 框架提供了多个基于 Web 的桌面和命令行工具，用于控制解决方案的各种组件。OpenVAS 的核心功能由安全扫描程序提供，它使用超过 33 个每日更新的网络漏洞测试(NVT)。与 Nessus 不同（请参阅

第 1.3.3.2 节 “Nessus”），OpenV AS 不需要任何订阅。

有关 OpenVAS 的更多信息，请访问以下 URL 的官方网站：

http://www.openvas.org/

1.3.3.4. Nikto

Nikto 是非常好的 *通用网关接口* (CGI)脚本扫描程序。Nikto 不仅检查 CGI 漏洞，而是以当前的方式进行检查，因此影响入侵检测系统。附带全面的文档，在运行程序之前应仔细检查。如果您有提供 CGI 脚本的 Web 服务器，Nikto 可以作为检查这些服务器安全性的绝佳资源。

有关 Nikto 的更多信息，请访问以下 URL：

http://cirt.net/nikto2

1.4. SECURITY THREATS

1.4.1. 网络安全隐患

在配置网络的以下方面时，错误的做法可能会增加攻击的风险。

不安全的构架
一个错误配置的网络是未授权用户的主要入口点。让一个基于信任的、开放的本地网络暴露在高度不安全的互联网上，就像让一扇门在一个犯罪猖獗的社区里敞开一样--在一段时间内可能不会发生任何事情，但终究会有人利用这个机会。

广播网络
系统管理员往往没有意识到网络硬件在其安全计划中的重要性。简单硬件（如集线器和路由器）依赖于广播或非交换原则；也就是说，每当节点通过网络将数据传输到接收节点时，集线器或路由器会发送数据包的广播，直到接收节点接收和处理数据。这种方法最容易受到外部入侵者和本地主机上未经授权用户的地址解析协议（*ARP*）或媒体访问控制（*MAC*）地址欺骗的攻击。

集中式服务器
另一个潜在的网络弱点是集中式的计算环境。许多企业常用的降低成本措施是将所有服务整合到一台功能强大的机器上。这很方便，因为它比多个服务器配置更易于管理，而且成本也低得多。但是，集中式服务器在网络上引入了单点故障。如果中央服务器受损，可能会导致网络完全不可用或更糟，容易发生数据操作或失窃。在这些情况下，中央服务器成为允许访问整个网络的开放门。

1.4.2. 服务器安全隐患

服务器安全性与网络安全性同样重要，因为服务器通常包含组织的大量重要信息。如果服务器被入侵，则其所有内容可能变得可供攻击者窃取或随意操作。以下小节详细介绍了一些主要问题。

未使用的服务和开放端口
Red Hat Enterprise Linux 7 的完整安装包含超过 1000 个应用程序和库软件包。但是，大多数服务器管理员并没有选择安装发行版中的每一个软件包，而更喜欢安装软件包的基本安装，包括几个服务器应用。有关限制安装的软件包数量以及其他资源的原因，请参阅 第 2.3 节 “安装最小软件包挂载”。

系统管理员经常出现的情况是，安装操作系统时没有注意到底安装了哪些程序。这可能有问题，因为可能会安装不需要的服务，使用默认设置进行了配置，并且可能开启了服务。这可能导致不需要的服务（如 Telnet、DHCP 或 DNS）在服务器或工作站上运行，而管理员没有意识到，这可能会给服务器造成不必要的流量，甚至造成进入系统的潜在通途，导致攻击者进入系统。有关关闭端口和禁用未使用的服务的详情，请参考 第 4.3 节 “保护服务”。

未修补的服务
默认安装中包含的大多数服务器应用程序都是经过全面测试的可靠的软件。多年来一直在生产环境中使用，其代码得到了全面优化，发现并修复了许多 bug 。

然而，没有完美的软件，总有进一步提升的空间。较新的软件通常不会象预期的一样严格测试，因为它最近才开始在生产环境中使用，或者可能不像其它服务器软件一样流行。

开发人员和系统管理员通常会在服务器应用程序中找到可被利用的错误，并在 bug 跟踪和安全相关 Web 站点（如 Bugtraq 邮件列表http://www.securityfocus.com）或 Computer Emergency Response Team (CERT)网站(http://www.cert.org)中发布信息。虽然这些机制是提醒社区了解安全隐患的有效方法，但效果取决于系统管理员是否立即对系统进行了补丁。这一点尤其正确，因为攻击者可以访问这些相同的漏洞跟踪服务，并随时使用这些信息来破解未修补的系统。良好的系统管理需要保持警惕，不断地跟踪程序漏洞，并进行适当的系统维护，以确保更安全的计算环境

有关保持系统最新的更多信息，请参阅 第 3 章 *使您的系统保持最新状态*。

管理
管理员如果没有对系统进行补丁，则会对服务器安全性造成最大的威胁。根据 *SysAdmin*、*审计*、*网络*、*安全研究所* (*SANS*)的主要原因是，计算机安全漏洞的主要原因是"为维护安全性分配未经培训，也没有时间来学习和完成工作"。[1] 这既适用于没有经验的管理者，也适用于过于自信或积极进取的管理者。

有些管理员没有给服务器和工作站打补丁，而有些管理员则没有观察系统内核或网络流量的日志消息。另一个常见错误是服务的默认密码或密钥没有改变。例如，一些数据库有默认的管理密码，因为数据库开发人员假定系统管理员在安装后会立即修改这些密码。如果数据库管理员没有修改这个密码，即使是没有经验的破解者也可以使用一个广为人知的默认密码来获得数据库的管理权限。这些只是几个例子，说明不注意管理会导致服务器被入侵。

固有的 Insecure 服务
如果选择的网络服务本身就不安全，即使是最警惕的组织也会成为漏洞的受害者。例如，有许多服务是在假设它们是在受信任的网络上使用的情况下开发的；然而，一旦服务在互联网上变得可用这一假设就失效了（互联网本身就是不受信任的）。

一个不安全的网络服务是那些需要未加密的用户名和密码进行身份验证的用户。Telnet 和 FTP 是两个这样的服务。如果数据包嗅探软件监控远程用户和此类服务用户名和密码之间的流量，则可以轻松地截获服务用户名和密码。

从本质上讲，这类服务也更容易成为安全行业所说的*中间人*攻击的猎物。在这种类型的攻击中，攻击者通过欺骗网络上被破解的名称服务器指向他的机器而不是目标服务器来重定向网络流量。当有人打开到服务器的远程会话后，攻击者的机器就充当一个不可见的机构，在远程服务与捕获信息的用户之间保持静默。这样，攻击者可以在没有服务器或者用户的情况下收集管理密码和原始数据。

另一类不安全的服务包括网络文件系统和信息服务，如NFS或NIS，它们是明确为局域网使用而开发的，但不幸的是，它们被扩展到包括广域网（为远程用户）。默认情况下,NFS 没有配置任何验证或安全机制以防止供节者挂载 NFS 共享并访问包含的任何内容。NIS 也有网络中的每个计算机都必须知道的重要信息，包括明文 ASCII 或 DBM（ASCII 派生）数据库中的密码和文件权限。获得对此数据库访问权限的攻击者，然后可以访问网络中的每个用户帐户，包括管理员的帐户。

默认情况下，Red Hat Enterprise Linux 7 会发布，所有此类服务都关闭。但是，由于管理员经常发现自己被迫使用这些服务，因此仔细配置是至关重要的。有关以安全的方式设置服务的更多信息，请参阅 第 4.3 节 “保护服务”。

1.4.3. 工作站和家庭 PC 安全的威胁

工作站和家庭 PC 可能不像网络或服务器那样容易受到攻击，而且由于它们通常包含敏感数据，如信用卡信息，它们都是系统攻击者的目标。工作站也可以在用户不知情的情况下被使用，并被攻击者用作协调攻击中的"从"机器。因此，了解工作站的漏洞可让用户避免重装操作系统的麻烦，或者更糟糕的是从数据失

窃中恢复。

错误**密**码
不好的密码是攻击者获得系统访问权限的最简单的方法之一。有关如何避免在创建密码时避免常见缺陷的更多信息，请参阅 第 4.1.1 节 “密码安全性”。

存在安全漏洞的客户**端**应**用程序**
虽然管理员可能拥有一个完全安全且打过补丁的服务器，但这并不表示远程用户在访问时是安全的。例如，如果服务器通过公共网络提供 Telnet 或 FTP 服务，攻击者可以捕获通过网络传递的纯文本用户名和密码，然后使用帐户信息访问远程用户的工作站。

即使使用安全协议（如 SSH），如果远程用户不更新其客户端应用，它们也可能会受到某些攻击。例如，v.1 SSH 客户端容易受到恶意 SSH 服务器的 X 转发攻击。一旦连接到服务器，攻击者可以悄悄地捕获客户端通过网络进行的任何击键动作和鼠标点击动作。这个问题已在 v.2 SSH 协议中解决，但用户需要跟踪哪些应用程序有此类漏洞并根据需要进行更新。

第 4.1 节 “桌面安全性” 详细讨论管理员和家庭用户应采取什么步骤来限制计算机工作站的漏洞。

1.5. COMMON EXPLOITS 和 ATTACKS

表 1.1 “通用扩展” 详细介绍入侵者访问组织网络资源时使用的一些最常见的漏洞和入口点。这些常见漏洞的关键在于解释了如何进行攻击以及管理员如何正确地保护其网络免受此类攻击。

表 1.1. 通用扩展

漏洞	描述	备注
空密码或默认密码	将管理密码置为空，或使用产品供应商设置的默认密码.这在路由器和防火墙等硬件中最常见，但一些在 Linux 上运行的服务也可以包含默认的管理员密码（虽然 Red Hat Enterprise Linux 7 不附带）。	通常与网络硬件（如路由器、防火墙、VPN 和网络附加存储(NAS)设备）相关。 在很多传统操作系统中很常见，尤其是那些捆绑了服务（如 UNIX 和 Windows）的操作系统。 管理员有时会匆忙创建特权用户帐户，并将密码置为空，从而为发现该帐户的恶意用户创建了一个完美的入口点。
默认共享密钥	出于开发或评估测试的目的，安全服务有时会打包默认的安全密钥。如果这些密钥保持不变，并放在互联网上的生产环境中，则拥有相同默认密钥的 *所有* 用户都可以访问该共享密钥的资源以及其包含的任何敏感信息。	最常在无线接入点和预配置的安全服务器设备中。
IP Spoofing	远程计算机充当本地网络上的节点，找到您服务器的漏洞，并安装一个后门程序或特洛伊木马来控制您的网络资源。	欺骗比较困难，因为它涉及到攻击者预测 TCP/IP 序列号以协调到目标系统的连接，但有几个工具都可帮助攻击者执行此类攻击。 具体取决于目标系统运行的服务（如 **rsh**、**telnet**、FTP等），这些服务使用 *基于源* 的身份验证技术，与 **ssh** 或 SSL/TLS 中使用的其他形式的加密身份验证相比，不建议这样做。

漏洞	描述	备注
窃听	通过窃听两个节点之间的连接，来收集网络上两个活动节点之间传递的数据。	这种类型的攻击主要适用于明文传输协议，如 Telnet、FTP 和 HTTP 传输。 远程攻击者必须能够访问局域网中一个被破坏的系统才能执行此类攻击；通常黑客使用主动攻击（如 IP 欺骗或中间人）来破坏局域网上的系统。 安全措施包括带有加密密钥交换、一次性密码或加密身份验证的服务，以防止密码嗅探；还建议在传输过程中进行强加密。
服务漏洞	攻击者发现在互联网上运行的服务有缺陷或漏洞；通过此漏洞，攻击者破坏整个系统以及其可能保存的任何数据，并可能破坏网络中的其他系统。	基于 HTTP 的服务（如 CGI）容易受到远程命令执行甚至交互式 shell 访问的攻击。即使 HTTP 服务以非特权用户（如"nobody"）的身份运行，可以读取的配置文件和网络映射等信息，或者攻击者可以发起拒绝服务攻击，从而耗尽系统资源或使其对其他用户不可用。 服务有时可能会有在开发和测试过程中没有被注意到的漏洞；这些漏洞（如 *缓冲区溢出*，攻击者会使用填充应用内存缓冲区的任意值使服务崩溃，从而给攻击者一个交互式命令提示，他们可以从中执行任意命令）可以为攻击者提供完整的管理控制。 管理员应确保服务不以 root 用户身份运行，并应该对来自供应商或安全组织(如 CERT 和 CVE)的应用程序补丁和勘误表更新保持警惕。
应用程序漏洞	攻击者在桌面和工作站应用程序（如电子邮件客户端）中发现错误，执行任意代码，植入特洛伊木马以备将来入侵，或使系统崩溃。如果被入侵的工作站对网络的其余部分具有管理特权，则可能会被进一步利用。	工作站和桌面更易被利用，因为工作者不具备防止或检测威胁的专业知识或经验；必须告知个人在安装未授权软件或打开未经请求的电子邮件附件时所承担的风险。 可以实施保护，如电子邮件客户端软件不自动打开或执行附件。此外，使用红帽网络自动更新工作站软件；或使用其他可以减轻多套安全部署负担的系统管理服务。
拒绝服务(DoS)攻击	攻击者或攻击者组通过向目标主机（服务器、路由器或工作站）发送未经授权的数据包，来针对组织的网络或服务器资源进行协调。这将迫使合法用户无法使用该资源。	美国报告的最新 DoS 问题单在 2000 年发生。几个高流量的商业和政府站点被协同的 ping 洪水攻击造成不可用，这些攻击使用了几个被破坏的系统，这些系统的高带宽连接被作为 *僵尸*，或重定向广播节点。 源数据包通常会被伪造（以及重播），从而使调查攻击的真正来源变得困难。 使用 **iptables** 和网络入侵检测系统（如 **snort**）进行入口过滤(IETF rfc2267)的进步有助于管理员跟踪并防止分布式 DoS 攻击。

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

第 2 章 安装的安全提示

安全从您第一次将 CD 或者 DVD 放入您的磁盘驱动器时开始，以安装 Red Hat Enterprise Linux 7。从一开始就安全地配置系统可以使以后更容易实施其他安全设置。

2.1. 保护 BIOS

对 BIOS（或与 BIOS 等效的）和引导加载程序的密码保护可防止具有系统物理访问权限的未授权用户使用可移动介质引导，或通过单用户模式获得 root 权限。您为防止此类攻击而需要采取的安全措施取决于工作站中信息的敏感程度和机器的位置。

例如，如果机器是在交易展示中使用并且不包含敏感信息，那么防止此类攻击可能并不重要。但是，如果员工使用私有的、未加密的 SSH 密钥进行公司网络的笔记本电脑，则在同一交易显示下以无人值守方式处理，则可能导致整个公司的严重安全漏洞。

但是，如果工作站位于只有授权的或可信任的人员才有权访问的地方，则可能不需要保护 BIOS 或引导加载程序。

2.1.1. BIOS 密码

密码保护计算机 BIOS 的两个主要原因是[2]:

1. *防止对 BIOS 设置的更改* - 如果入侵者可以访问 BIOS，他们可以将其设置为从 CD-ROM 或闪存驱动器引导。这使得他们能够进入救援模式或单用户模式，从而使他们可以在系统上启动任意进程或复制敏感数据。

2. *防止系统引导* - 一些 BIOS 允许对引导过程进行密码保护。激活后，攻击者必须在 BIOS 启动引导加载程序前输入密码。

由于设置 BIOS 密码的方法因计算机制造商而异，因此请查阅计算机手册以了解具体说明。

如果您忘记 BIOS 密码，可以通过主板上的跳线来重置，也可以通过断开 CMOS 电池来重置。因此，如果可能的话，最好锁好计算机机箱。但是，在尝试断开 CMOS 电池之前，请查阅计算机或主板的手册。

2.1.1.1. 保护基于非 BIOS 的系统

其他系统和架构使用不同的程序来执行大致相当于 x86 系统上 BIOS 的低级别任务。*例如，统一可扩展固件接口* (*UEFI*)shell。

有关密码保护类似 BIOS 程序的步骤，请查看制造商的说明。

2.2. 对磁盘进行分区

红帽建议为 **/boot**、/、**/home**、**/tmp** 和 **/var/tmp/** 目录创建单独的分区。每种分区的原因不同，我们将解决每个分区。

/boot

这个分区是系统在启动过程中读取的第一个分区。用于将系统引导至 Red Hat Enterprise Linux 7 的引导装载程序和内核镜像存储在这个分区中。此分区不应加密。如果这个分区包含在 / 中，且该分区已加密或者不可用，则您的系统将无法引导。

/home

当用户数据（**/home**）存储在 / 而不是独立分区中时，分区可能会填满，从而导致操作系统不稳定。另

外，当将您的系统升级到 Red Hat Enterprise Linux 7 的下一版本时，当您可以将数据保存在 **/home** 分区中时，在安装过程中不会覆盖它。如果 root 分区(/)损坏，则您的数据将永久丢失。通过使用单独的分区，对数据丢失有稍微多一点的保护。您还可以将此分区作为频繁备份的目标。

/tmp 和 **/var/tmp/**。

/tmp 和 **/var/tmp/** 目录都是用来存储不需要长期存储的数据。但是，如果大量数据填充了其中一个目录，则它可能会消耗掉您的所有存储空间。如果发生这种情况，且这些目录存储在 / 中，则您的系统可能会变得不稳定并崩溃。因此，将这些目录移到它们自己的分区中是一个不错的想法。

注意

在安装过程中，您可以选择加密分区。您必须提供密码短语。此密码短语充当解锁批量加密密钥的密钥，该密钥用于保护分区的数据。如需更多信息，请参阅 第 4.9.1 节 “使用 LUKS 磁盘加密”。

2.3. 安装最小软件包挂载

最好只安装您要使用的软件包，因为计算机上的每一款软件都可能包含漏洞。如果您要从 DVD 介质安装，请仔细选择要在安装过程中安装的软件包。如果您发现需要其他软件包，您可在以后将其添加到系统中。

有关安装 **Minimal** 安装 环境的更多信息，请参阅 Red Hat Enterprise Linux 7 安装指南中的 软件选择 章节。最少的安装也可以通过 Kickstart 文件使用 **--nobase** 选项执行。有关 Kickstart 安装的详情，请查看 Red Hat Enterprise Linux 7 安装指南中的软件包选择部分。http://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-kickstart-syntax.html#sect-kickstart-packages

2.4. 在安装过程中限制网络连接

安装 Red Hat Enterprise Linux 时，安装介质代表系统在特定时间的快照。因此，它可能没有最新的安全修复程序，并且可能容易受到某些问题的攻击，这些问题是在安装介质提供的系统发布后才修复的。

安装有潜在漏洞的操作系统时，始终将暴露限制在最近的必要网络区内。最安全的选择是"无网络"区，这意味着在安装过程中使计算机断开连接。在某些情况下，LAN 或内部网连接就足够了，而互联网连接的风险最大。要遵循最佳安全实践，请从网络安装 Red Hat Enterprise Linux 时，选择与您的软件仓库最接近的区域。

有关配置网络连接的更多信息，请参阅 Red Hat Enterprise Linux 7 安装指南中的网络和主机名章节。https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-network-hostname-configuration-x86.html

2.5. 安装后的步骤

以下步骤是安装 Red Hat Enterprise Linux 后应立即执行的安全相关步骤。

1. 更新您的系统。以 root 用户身份输入以下命令：

```
~]# yum update
```

2. 尽管安装 Red Hat Enterprise Linux 后会自动启用防火墙服务 **firewalld**，但在某些情况下，它可能会被明确禁用，例如在 kickstart 配置中。在这种情况下，建议考虑重新启用防火墙。

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

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

3. 要提高安全性，请禁用您不需要的服务。例如，如果您的计算机上没有安装打印机，使用以下命令禁用 **cups** 服务：

   ```
   ~]# systemctl disable cups
   ```

   要查看活动状态的服务，请输入以下命令：

   ```
   ~]$ systemctl list-units | grep service
   ```

2.6. 其它资源

有关常规安装的详情，请查看 Red Hat Enterprise Linux 7 安装指南 。

---

[2] 由于不同厂家的系统 BIOS 不同，一些可能任何一种类型的密码保护都不支持，另一些则可能支持其中一种类型，但不支持另一种类型。

第 3 章 使您的系统保持最新状态

本章论述了保持系统最新的过程，涉及计划和配置安装安全更新的方式，应用新更新的软件包引入的变化，以及使用红帽客户门户网站来跟踪安全公告。

3.1. 维护已安装的软件

发现安全漏洞时，必须更新受影响的软件，以限制任何潜在的安全风险。如果软件是当前支持的 Red Hat Enterprise Linux 发行版中的一个软件包的一部分，红帽会发布可尽快修复漏洞的更新软件包。

通常，针对给定安全漏洞的公告会附带一个修复此问题的补丁（或源代码）。然后，这个补丁适用于 Red Hat Enterprise Linux 软件包，并作为勘误更新进行测试并发布。但是，如果公告不包括补丁，红帽开发人员首先与软件维护人员合作，以解决问题。修复此问题后，软件包将作为勘误更新进行测试并发布。

如果为系统上使用的软件发布勘误更新，强烈建议您尽快更新受影响的软件包，以最大程度降低系统可能存在安全漏洞的时间。

3.1.1. 规划和配置安全更新

所有软件都包含 bug。通常，这些 bug 可能会导致漏洞使您的系统暴露给恶意用户。未更新的软件包是计算机入侵的常见原因。实施及时安装安全补丁的计划，以快速消除发现的漏洞，因此无法利用它们。

当安全更新可用时，测试安全更新，并将其调度到安装。在更新发行及其系统中安装时，需要使用其他控制来保护系统。这些控制取决于确切的漏洞，但可能包括其他防火墙规则、使用外部防火墙或更改软件设置。

支持的软件包中的错误使用勘误机制修复。勘误由一个或多个 RPM 软件包组成，并附带特定勘误处理的问题的简短说明。所有勘误都通过 **红帽订阅管理服务** 提供给具有有效订阅的客户。解决安全问题的勘误被称为 *红帽安全公告*。

有关使用安全勘误的更多信息，请参阅 第 3.2.1 节 “在客户门户网站中查看安全公告”。有关红帽订阅管理服务的详细信息，包括如何从 **RHN Classic** 迁移的说明，请参阅与此服务相关的文档： 红帽订阅管理。

3.1.1.1. 使用 Yum 的安全功能

Yum 软件包管理器包含若干与安全相关的功能，可用于搜索、列出、显示和安装安全勘误。这些功能还支持使用 **Yum** 安装安全更新。

要检查系统可用的与安全相关的更新，请以 **root** 用户身份输入以下命令：

```
~]# 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，上述命令以非交互模式运行，因此可以在脚本中使用它来自动检查是否有可用的更新。当有任何可用的安全更新，如果不存在，则命令会返回 100 的 exit 值。0在遇到错误时，它会返回 1。

类似地，使用以下命令只安装与安全相关的更新：

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

使用 **updateinfo** 子命令显示有关可用更新的存储库提供的信息。**updateinfo** 子命令本身接受多个命令，一些与安全相关的用途相关。有关这些命令的概述信息，请参阅 表 3.1 "与安全相关的命令，用于 yum updateinfo"。

表 3.1. 与安全相关的命令，用于 yum updateinfo

命令	描述
公告 [公告]	显示有关一个或多个公告的信息。使用 *公告* 号或数字替换公告。
CVE	显示与 CVE 相关的信息子集(*常见漏洞和暴露*)。
Security 或 **sec**	显示所有与安全相关的信息。
[severity_level] 或 **sev [severity_level]**	显示提供的 *severity_level* 的安全相关软件包的信息。

3.1.2. 更新和安装软件包

更新系统上的软件时，务必要从可信源下载更新。攻击者可以轻松重建具有相同版本号的软件包，其版本号应该与应该解决问题但存在不同的安全漏洞，并在互联网上释放它。如果发生这种情况，使用安全措施（如针对原始 RPM 验证文件）不会检测漏洞。因此，仅从可信源（如红帽）下载 RPM 非常重要，并检查软件包签名以验证其完整性。

有关如何使用 Yum 软件包管理器的详细信息，请参见 Red Hat Enterprise Linux 7 系统管理员指南中的 Yum 章节。

3.1.2.1. 验证签名的软件包

所有 Red Hat Enterprise Linux 软件包都使用 Red Hat **GPG** 密钥签名。**GPG** 代表 **GNU Privacy Guard** 或 **GnuPG**，这是用于确保分布式文件真实性使用的免费软件包。如果软件包签名验证失败，则可以更改软件包，因此无法信任。

Yum 软件包管理器允许自动验证其安装或升级的所有软件包。此功能默认为启用。要在您的系统中配置这个选项，请确保在 **/etc/yum.conf** 配置文件中将 **gpgcheck** 配置指令设置为 **1**。

使用以下命令手动验证文件系统中的软件包文件：

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

有关红帽软件包签名实践的更多信息，请参阅红帽客户门户网站上的 产品签名(GPG)密钥 文章。

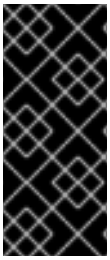
3.1.2.2. 安装签名软件包

要安装经过验证的软件包（请参阅 第 3.1.2.1 节 "验证签名的软件包" 以了解如何从您的文件系统中验证软件包），请使用 **yum install** 命令，如下所示：

```
yum install package_file.rpm
```

使用 **shell** 通配一次性安装多个软件包。例如，以下命令会在当前目录中安装所有 **.rpm** 软件包：

```
yum install *.rpm
```



重要

在安装任何安全勘误前，请务必阅读勘误报告中包含的任何特殊指令，并相应地执行它们。有关应用勘误更新所做的更改的一般说明，请参阅 第 3.1.3 节 “应用由安装更新引入的变化”。

3.1.3. 应用由安装更新引入的变化

下载并安装安全勘误和更新后，停止使用旧软件并开始使用新软件非常重要。具体操作方式取决于已更新的软件类型。以下列表列出了软件的一般类别，并提供在软件包升级后使用更新的版本的说明。

注意

通常，重新引导系统是确保使用最新版本的软件包的确定方式；但是，此选项并非始终是必需的，也并非总是可用于系统管理员的。

应用程序

用户空间应用程序是用户可以启动的任何程序。通常，只有在用户、脚本或自动任务实用程序启动它们时，才会使用此类应用。

更新这样的用户空间应用程序后，停止系统上的任何应用程序实例，然后再次启动程序以使用更新的版本。

内核

内核是 **Red Hat Enterprise Linux 7** 操作系统的核心软件组件。它管理对内存、处理器和外围设备的访问，并且调度所有任务。

由于其中央角色，在没有重新启动计算机的情况下，无法重新启动内核。因此，在系统重启前，无法使用内核版本。

KVM

更新 **qemu-kvm** 和 **libvirt** 软件包后，需要停止所有客户虚拟机，重新载入相关的虚拟化模块（或重启主机系统），并重新启动虚拟机。

使用 **lsmod** 命令确定从以下哪些模块被加载： **kvm**、**kvm-intel** 或 **kvm-amd**。然后，使用 **modprobe -r** 命令删除并随后使用 **modprobe -a** 命令重新加载受影响的模块。**fox** 示例：

```
~]# 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，如 **glibc**，它们由多个应用程序和服务使用。使用共享库的应用程序通常会在应用程序初始化时加载共享代码，因此任何使用更新库的应用程序都必须停止并重新启动。

要确定哪个应用程序针对特定库链接，请使用 **lsof** 命令：

```
lsof library
```

例如，要确定哪些运行的应用程序链接到 **libwrap.so.0** 库，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
```

此命令返回一个使用 **TCP** 包装程序进行主机访问控制的所有运行程序的列表。因此，在更新 **tcp_wrappers** 软件包时，任何列出的程序都必须停止并重新启动。

systemd 服务

systemd 服务是通常在引导过程中启动的持久服务器程序。**systemd** 服务示例包括 **sshd** 或 **vsftpd**。

因为只要机器正在运行，这些程序通常会在内存中保留，因此在升级其软件包后，每个更新的 **systemd** 服务都必须停止并重新启动。这可以以 **root** 用户身份使用 **systemctl** 命令完成：

-

systemctl restart *service_name*

使用您要重启的服务的名称替换 ***service_name***，如 **sshd**。

其他软件

按照下方链接的资源概述的说明，正确更新以下应用程序。

- **Red Hat Directory Server -** 请参阅 *中有关的 Red Hat Directory Server 版本的发行注记。https://access.redhat.com/documentation/zh-CN/Red_Hat_Directory_Server/*
- ***Red Hat Enterprise Virtualization Manager - 请参阅有关的 Red Hat Enterprise Virtualization 版本的 安装指南，网址为 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Virtualization/。***

3.2. 使用红帽客户门户网站

红帽客户门户网站 https://access.redhat.com/ 是与红帽产品相关的官方信息的主要面向客户的资源。您可以使用它来查找文档、管理订阅、下载产品和更新、创建支持问题单以及了解安全更新。

3.2.1. 在客户门户网站中查看安全公告

要查看与具有有效订阅的系统相关的安全公告(errata)，请登录到客户门户网站 https://access.redhat.com/，然后点击主页中的 Download Products and Updates 按钮。当您进入 Software & Download Center 页面时，点 Errata 按钮查看您注册的系统相关的公告列表。

要浏览所有活跃红帽产品的所有安全更新列表，请使用页面顶部的导航菜单进入 Security → Security Updates → Active Products。

点击表左侧的勘误代码，以显示有关各个公告的更多详细信息。下一页仅包含给定勘误的描述，包括其原因、后果和所需的修复，以及特定勘误更新的所有软件包列表，以及如何应用更新。该页面还包括相关引用的链接，如相关的 CVE。

3.2.2. CVE 客户门户网站页面

CVE (常见漏洞和风险)项目由 MITRE 公司维护，是漏洞和安全暴露的标准化名称列表。要浏览与客户门户网站中红帽产品相关的 CVE 列表，请在 https://access.redhat.com/ 登录您的帐户，并使用页面顶

部的导航菜单导航到 Security → Resources → CVE Database。

点击表左侧的 CVE 代码，以显示有关各个漏洞的更多详细信息。下一页仅包含对给定 CVE 的描述，以及受影响的红帽产品列表以及相关红帽勘误的链接。

3.2.3. 了解问题严重性分级

红帽产品中发现的所有安全问题都会根据问题的严重性为 红帽产品安全 影响等级。四点评级由以下级别组成： Low、Moderate、Important 和 Critical。此外，每个安全问题还使用 通用漏洞评分系统 (CVSS)基础评分进行评级。

这些评级结合可帮助您了解安全问题的影响，允许您为您的系统调度和优先级升级策略。请注意，评级反映了给定漏洞的潜在风险，这基于对程序错误的技术分析，而不是当前的威胁级别。这意味着，如果为特定漏洞发布漏洞，安全影响评级不会改变。

要查看客户门户网站中单个严重性级别的详细描述，请访问 严重性评级 页面。

3.3. 其它资源

有关安全更新、应用它们、红帽客户门户网站和相关主题的方法的更多信息，请参阅以下列出的资源。

安装的文档

- yum(8) - Yum 软件包管理器的手册页提供了有关 Yum 可用来在您的系统上安装、更新和删除软件包的方式的信息。
- rpmkeys(8) - rpmkeys 实用程序的手册页描述了该程序可用于验证下载的软件包的真实性的方式。

在线文档

- Red Hat Enterprise Linux 7 系统管理员指南 - Red Hat Enterprise Linux 7 系统管理员指南记录了在 Red Hat Enterprise Linux 7 系统上用于安装、更新和删除软件包的 Yum 和 rpm 命令。
- Red Hat Enterprise Linux 7 SELinux 用户和管理员指南 - Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南记录了 SELinux 强制访问控制机制 的配置。

红帽客户门户网站

- **Red Hat Customer Portal, Security** - 客户门户网站的安全部分包含到最重要的资源的链接，包括 Red Hat CVE 数据库，以及红帽产品安全联系。

- 红帽安全博客 - 有关红帽安全专家的最新与安全相关问题的文章。

另请参阅

- 第 2 章 安装的安全提示 描述如何从开始安全地配置您的系统，以便以后更轻松地实施其他安全设置。

- 第 4.9.2 节 “创建 GPG 密钥” 描述如何创建一组个人 GPG 密钥以验证您的通信。

第 4 章 使用工具和服务强化您的系统

4.1. 桌面安全性

Red Hat Enterprise Linux 7 提供了多种方法来强化桌面免受攻击，并防止未经授权的访问。这部分论述了用户密码、会话和帐户锁定以及可移动介质的安全处理的建议实践。

4.1.1. 密码安全性

密码是 Red Hat Enterprise Linux 7 用于验证用户身份的主要方法。这就是为什么密码安全性如此重要，对保护用户、工作站和网络至关重要。

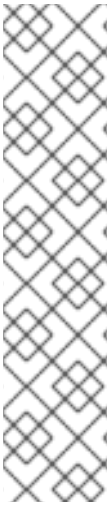
为安全起见，安装程序会将系统配置为使用 安全哈希算法 512 (SHA512)和影子密码。强烈建议您不要更改这些设置。

如果在安装过程中取消选择影子密码，则所有密码都以单向哈希形式存储在全局可读的 /etc/passwd 文件中，这样会使系统容易受到离线密码攻击。如果入侵者能够以普通用户身份获取对计算机的访问权限，他可将 /etc/passwd 文件复制到自己的计算机上，并对其运行任意数量的密码阻止程序。如果文件中有不安全的密码，则只需要在密码破解程序发现它之前的时间。

影子密码通过将密码哈希存储在文件 /etc/shadow 中消除此类攻击，这只对 root 用户可读。

这会强制潜在的攻击者通过登录到机器上的网络服务（如 SSH 或 FTP）来远程破解密码。此类 brute-force 攻击非常慢，因为数百个失败的登录尝试会写入系统文件。当然，如果攻击者在带有弱密码的系统上在夜中启动攻击，则攻击者可能会在 dawn 前获得访问权限，并编辑日志文件以覆盖其跟踪。

除了格式和存储注意事项外，还需要考虑内容。用户可以防止其帐户遭受攻击的单个最重要的事情是创建一个强大的密码。



注意

红帽建议使用中央身份验证解决方案，如 Red Hat Identity Management (IdM)。首选使用中央解决方案使用本地密码。详情请查看：

- Red Hat Identity Management 简介
- 定义密码策略

4.1.1.1. 创建强密码

在创建安全密码时，用户必须记住长的密码比短和复杂的密码更强。创建仅有 8 个字符的密码是个好主意，即使它包含数字、特殊字符和大写字母。密码破解工具（如 John The Ripper）针对破坏此类密码进行了优化，这也很难记住。

在信息理论中，熵是与随机变量关联的不确定性级别，以位为单位。熵值越大，保护密码越高。根据 NIST SP 800-63-1，在字典中没有包括 50000 的密码，通常选择的密码应至少有 10 位熵。因此，由四个随机单词组成的密码包含大约 40 位熵。包含用于添加安全性的多个词语的长密码也称为 密码短语，例如：

```
randomword1 randomword2 randomword3 randomword4
```

如果系统强制使用大写字母、数字或特殊字符，则遵循上述建议的密码短语可以以简单方式修改，例如，将第一个字符更改为大写并附加 "1!"。请注意，此类修改 不会 显著增加密码短语的安全性。

另一种创建密码的方式是使用密码生成器。pwmake 是一个命令行工具，用于生成由所有四个字符组成的随机密码 - 大写、小写、数字和特殊字符。实用程序允许您指定用于生成密码的熵位数。熵从 /dev/urandom 拉取。您可以指定的最小位数为 56，它足以用于系统和服务的密码，其中 brute 强制攻击非常罕见。64 位对于攻击者无法直接访问密码哈希文件的应用程序来说是足够的。对于攻击者可能会获得对密码哈希的直接访问或者密码用作加密密钥的情况，应使用 80 到 128 位。如果您指定了无效数量的熵位，pwmake 将使用默认位。要创建 128 位的密码，请输入以下命令：

```
pwmake 128
```

虽然创建安全密码的方法不同，但总是避免以下错误的实践：

- 使用单个字典词语，一个外部语言中的词语、一个 inverted 字词或仅数字。

- 使用少于 10 个字符的密码或密码短语。
- 使用键盘布局中的一系列键。
- 写出您的密码。
- 在密码中使用个人信息，如过期日期、年金、家庭成员名称或片断名称。
- 在多台机器上使用相同的密码短语或密码。

虽然创建安全密码是必然的，但正确管理它们也很重要，特别是大型组织内的系统管理员。以下部分详细介绍了在组织内创建和管理用户密码的良好做法。

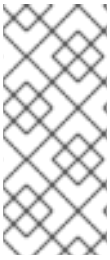
4.1.1.2. 强制密码

如果组织有大量用户，系统管理员可以有两个基本选项来强制使用强密码。它们可以为用户创建密码，或者允许用户在验证密码时创建自己的密码。

为用户创建密码可确保密码正常，但随着组织不断增长，它就成为了任务。它还会增加用户停用密码的风险，从而公开密码。

因此，大多数系统管理员更喜欢用户创建自己的密码，但主动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在密码过期时定期更改密码。

当要求用户创建或更改密码时，他们可以使用 passwd 命令行工具，该实用程序为 PAM- 感知(可插拔验证模块)，并检查密码是否太短，或者易于破解。此检查由 pam_pwquality.so PAM 模块执行。



注意

在 Red Hat Enterprise Linux 7 中，pam_pwquality PAM 模块替换了 pam_cracklib，它在 Red Hat Enterprise Linux 6 中使用它作为密码质量检查的默认模块。它使用与 pam_cracklib 相同的后端。

pam_pwquality 模块用于检查密码对一组规则的强度。其流程由两个步骤组成：首先它检查在字典中是否找到提供的密码。如果没有，它会继续执行很多额外的检查。pam_pwquality 与 /etc/pam.d/passwd 文件 的密码 组件中的其他 PAM 模块一同堆叠，在 /etc/security/pwquality.conf 配置文件中指定自定义规则集。有关这些检查的完整列表，请查看 pwquality.conf (8) 手册页。

例 4.1. 在 pwquality.conf中配置密码强度检查

要使用 pam_quality 启用，请将以下行添加到 /etc/pam.d/passwd 文件中的 密码 堆栈中：

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

检查的选项会每行指定一个。例如，要要求密码至少为 8 个字符，包括所有四个字符类，请在 /etc/security/pwquality.conf 文件中添加以下行：

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

要为字符序列和相同的连续字符设置密码强度检查，请将以下行添加到 /etc/security/pwquality.conf 中：

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

在这个示例中，输入的密码无法在单调序列中包含 3 个字符，如 abcd，以及 3 个相同的连续字符，如 1111。

注意

由于 root 用户是强制创建密码的规则，他们可以为自己或普通用户设置任何密码，尽管警告消息也是如此。

4.1.1.3. 配置密码过期

密码过期是系统管理员用于防御机构中损坏的密码的另一种技术。密码过期意味着在指定周期（通常为 90 天）后，系统将提示用户创建新密码。这样做的理论是，如果用户强制定期更改其密码，则攻击者密码只对有限时间的入侵者有用。但是，在密码过期之外，用户更有可能将其密码写出。

要在 Red Hat Enterprise Linux 7 下指定密码过期，请使用age 命令。

重要

在 Red Hat Enterprise Linux 7 中，默认启用影子密码。如需更多信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南。

concurrency 命令的 - M 选项指定密码有效的最大天数。例如，要将用户的密码设置为在 90 天后过期，请使用以下命令：

```
chage -M 90 username
```

在上述命令中，使用用户名替换 username。要禁用密码过期，请在 -M 选项后使用 -1。

有关可用选项的更多信息，请参阅下表。

表 4.1. age 命令行选项

选项	描述
-d *days*	指定 1970 年 1 月 1 日之后的天数，密码已改变。
-E *date*	指定帐户被锁定的日期，格式为 YYYY-MM-DD。也可以使用 1970 年 1 月 1 日之后的天数。
-I *days*	指定在锁定帐户前密码过期的非活动天数。如果值为 0，则帐户在密码过期后不会被锁定。
-l	列出当前帐户过期设置。
-m *days*	指定用户必须更改密码的最小天数。如果值为 **0**，则密码不会过期。
-M *days*	指定密码有效的最大天数。当此选项指定的天数加上 **-d** 选项指定的天数小于当前日期的天数时，用户必须先更改密码。
-W *days*	指定密码到期日期前的天数，以警告用户。

您也可以在交互模式中使用 interaction 命令修改多个密码过期和帐户详细信息。使用以下命令进入交互模式：

-

```
chage <username>
```

以下是使用以下命令的互动会话示例：

```
~]# 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]:
```

您可以将密码配置为在用户首次登录时过期。这会强制用户立即更改密码。

1. 设置初始密码。要分配默认密码，以 root 用户身份在 shell 提示符后输入以下命令：

```
passwd username
```

警告

passwd 实用程序具有设置 null 密码的选项。使用 null 密码是非常方便的做法，因为任何第三方都可以使用不安全的用户名登录并访问系统。尽可能避免使用 null 密码。如果无法实现，请始终确保用户已准备好登录，然后再使用 null 密码解锁帐户。

2. 以 root 用户身份运行以下命令来强制立即密码过期：

```
chage -d 0 username
```

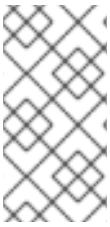
此命令设置密码上次更改为 epoch (January 1, 1970)的日期的值。这个值会强制立即过期密码过期策略（如果有的话）。

在初始登录时，用户现在会提示您输入新密码。

4.1.2. 帐户锁定

在 Red Hat Enterprise Linux 7 中，pam_faillock PAM 模块允许系统管理员在指定次数尝试失败后锁定用户帐户。限制用户登录尝试主要作为安全措施，旨在防止针对获取用户帐户密码的可能的暴力攻击。

使用 pam_faillock 模块时，失败的登录尝试存储在 /var/run/faillock 目录中每个用户的独立文件中。



注意

失败的尝试日志文件中的行顺序非常重要。这个顺序的任何更改都可以锁定所有用户帐户，包括在使用 even_deny_ root 选项时包括 root 用户帐户。

按照以下步骤配置帐户锁定：

1. 要在 3 次失败后锁定任何非 root 用户，并在 10 分钟后解锁该用户，请将两行添加到 /etc/pam.d/system- auth 和 / etc/pam.d/password-auth 文件的 auth 部分。编辑后，两个文件中的整个 auth 部分都应该类似如下：

```
auth        required      pam_env.so
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
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so
```

添加了行号 2 和 4。

2. 在上一步中指定的两个文件的 account 部分添加以下行：

```
account     required      pam_faillock.so
```

3. 要对 root 用户应用帐户锁定，请将 even_deny_root 选项添加到 /etc/pam.d/system-auth 和 /etc/pam.d/password-auth 文件中的 pam_faillock 条目：

```
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

account     required      pam_faillock.so
```

当用户 **john** 在之前登录三次后尝试登录时，其帐户会在第四个尝试时被锁定：

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

要防止系统在多次登录失败后锁定用户，请在行的上面添加以下行：在 **/etc/pam.d/system-auth** 和 **/etc/pam.d/password-auth** 中首次调用 **pam_faillock**。另外，将 **user1**、**user2** 和 **user3** 替换为实际用户名。

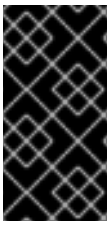
```
auth [success=1 default=ignore] pam_succeed_if.so user in user1:user2:user3
```

要查看每个用户失败的尝试次数，请以 **root** 用户身份运行 ：

```
~]$ faillock
john:
When                Type  Source                                           Valid
2013-03-05 11:44:14 TTY   pts/0                                               V
```

要解锁用户帐户，请以 **root** 用户身份运行：

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



重要

运行 **cron** 作业会重置该用户的 **pam_faillock** 的故障计数器，因此不应为 **cron** 配置 **pam_faillock**。如需更多信息，请参阅 知识库支持(KCS)解决方案。

使用 authconfig 保留自定义设置

当使用 **authconfig** 工具修改身份验证配置时，**system-auth** 和 **password-auth** 文件会被 **authconfig** 工具中的设置覆盖。这可以通过创建符号链接来代替配置文件，**authconfig** 识别且不会被覆盖。要在配置文件和 **authconfig** 中同时使用自定义设置，请按照以下步骤配置帐户锁定：

1. 检查 **system-auth** 和 **password-auth** 文件是否已指向 **system-auth-ac** 和 **password-auth-ac**（这是系统默认设置）：

```
~]# ls -l /etc/pam.d/{password,system}-auth
```

如果输出类似如下，符号链接就就位，您可以跳过到第 3 步：

```
lrwxrwxrwx. 1 root root 16 24. Feb 09.29 /etc/pam.d/password-auth -> password-auth-ac
lrwxrwxrwx. 1 root root 28 24. Feb 09.29 /etc/pam.d/system-auth -> system-auth-ac
```

如果 system-auth 和 password-auth 文件不是符号链接，请继续下一步。

2. **重命名配置文件：**

```
~]# mv /etc/pam.d/system-auth /etc/pam.d/system-auth-ac
~]# mv /etc/pam.d/password-auth /etc/pam.d/password-auth-ac
```

3. **使用自定义设置创建配置文件：**

```
~]# vi /etc/pam.d/system-auth-local
```

/etc/pam.d/system-auth-local 文件应包含以下行：

```
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        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
```

```
~]# vi /etc/pam.d/password-auth-local
```

/etc/pam.d/password-auth-local 文件应包含以下行：

```
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        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       password-auth-ac

session     include       password-auth-ac
```

4. 创建以下符号链接：

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

有关各种 pam_faillock 配置选项的更多信息，请参阅 pam_faillock(8) 手册页。

删除 nullok 选项

如果 /etc/shadow 文件中的 password 字段为空，则 nullok 选项允许用户使用空白密码登录。要禁用 nullok 选项，请从 /etc/pam.d/ 目录中的配置文件中删除 nullok 字符串，如 /etc/pam.d/system-auth 或 /etc/pam.d/password-auth。

请参阅 nullok 选项，允许用户在不输入密码的情况下登录？KCS 解决方案以了解更多信息。

4.1.3. 会话锁定

用户可能需要因为日常操作过程中有很多原因而使其工作站无人值守。这可能会给攻击者物理访问机器提供了机会，特别是在物理安全措施不足的环境中（请参阅 第 1.2.1 节 “物理控制”）。笔记本电脑特别公开，因为其移动性会干扰物理安全性。您可以使用会话锁定功能来缓解这些风险，这些功能会阻止访问系统，直到输入正确的密码为止。

注意

锁定屏幕而不是注销的主要优点是，锁允许用户的进程（如文件传输）继续运行。注销将停止这些进程。

4.1.3.1. 使用 vlock 锁定虚拟控制台

要锁定虚拟控制台，请使用 vlock 工具。以 root 身份输入以下命令安装它：

```
~]# yum install kbd
```

安装后，您可以使用 vlock 命令锁定任何控制台会话，而无需附加参数。这会锁定当前活动的虚拟控制台会话，同时仍然允许访问其他控制台。要防止访问工作站上的所有虚拟控制台，请执行以下操作：

```
vlock -a
```

在这种情况下，vlock 会锁定当前活跃的控制台，而 -a 选项会阻止切换到其他虚拟控制台。

详情请查看 vlock (1) 手册页。

4.1.4. 强制只读挂载 Removable Media

要强制对可移动介质进行只读挂载（如 USB 闪存磁盘），管理员可以使用 udev 规则来检测可移动介质，并使用 blockdev 工具将它们配置为只读挂载。这足以强制物理介质的只读挂载。

使用 blockdev 强制只读挂载 Removable Media

要强制以只读方式挂载所有可移动介质，请在 /etc/ udev /rules.d/ 目录中创建一个名为 的 udev 配置文件，例如： 80-readonly-removables.rules:

```
SUBSYSTEM=="block",ATTRS{removable}=="1",RUN{program}="/sbin/blockdev --setro %N"
```

以上 udev 规则确保所有新连接的可移动块（存储）设备都使用 blockdev 工具自动配置为只读。

应用新的 udev 设置

要使这些设置生效，需要应用新的 udev 规则。udev 服务自动检测对其配置文件的更改，但新设置不会应用到已存在的设备。只有新连接的设备会受到新设置的影响。因此，您需要卸载并拔出所有连接的可移动介质，以确保在下一次插入时将新设置应用到它们。

要强制 udev 将所有规则重新应用到已存在的设备，请以 root 用户身份输入以下命令：

```
~# udevadm trigger
```

请注意，强制 udev 使用上述命令重新应用所有规则不会影响任何已经挂载的存储设备。

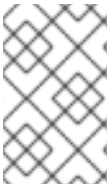
要强制 udev 重新加载所有规则（出于某种原因不会自动检测到新规则），请使用以下命令：

```
~# udevadm control --reload
```

4.2. 控制根访问

在管理家计算机时，用户必须以 root 用户身份执行一些任务，或者使用 setuid 程序（如 sudo 或 su ）获取有效的 root 特权。setuid 程序是与程序的所有者的用户 ID (UID)而不是用户操作程序运行的用户 ID (UID)。这些程序由长格式列表的所有者部分的 s 表示，如下例所示：

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



注意

s 可能为大写或小写。如果显示为大写，这表示没有设置底层权限位。

但是，对于组织的系统管理员，必须做出选择，因为该机构中的管理访问权限用户应该对其机器具有多少管理访问权限。通过名为 pam_console.so 的 PAM 模块，通常只为 root 用户保留一些活动，如重新启动和挂载可移动介质，则允许物理控制台登录的第一个用户。但是，在没有管理特权的情况下，无法更改网络设置、配置新鼠标或挂载网络设备等其他重要的系统管理任务。因此，系统管理员必须决定应收到其网络上的用户数量。

4.2.1. 禁止 Root 访问

如果管理员不可更改地允许用户以 root 身份登录，或者出于其他原因，应保留了 root 密码，并且应保持 secret，并且应禁止访问运行级别一个或多个用户模式（有关此主题的更多信息，请参阅 第 4.2.5 节 “保护 Boot Loader”）。

以下是管理员可以进一步确保禁止 root 登录的四个不同方法：

更改 root shell

为防止用户直接以 root 身份登录，系统管理员可以将 root 帐户的 shell 设置为 /etc/passwd 文件中的 /sbin/nologin。

表 4.2. 禁用 Root Shell

影响	未受影响
阻止访问 root shell 并记录任何此类尝试。以下程序无法访问 root 帐户： • login • gdm • kdm • XDM • su • ssh • scp • sftp	不需要 shell 的程序，如 FTP 客户端、邮件客户端和许多 setuid 程序。以下程序 不会阻止 访问 root 帐户： • sudo • FTP 客户端 • 电子邮件客户端

使用任何控制台设备(tty)禁用 root 访问权限

要进一步限制对 root 帐户的访问，管理员可以通过编辑 /etc/securetty 文件来禁用在控制台中的 root 登录。此文件列出了允许 root 用户登录的所有设备。如果文件根本不存在，root 用户可以通过系

统上的任何通信设备（无论是通过控制台还是原始网络接口）登录。这很危险，因为用户可以使用 Telnet 以 root 身份登录其计算机，这会通过网络以纯文本形式传输密码。

默认情况下，Red Hat Enterprise Linux 7 的 /etc/securetty 文件只允许 root 用户在与机器物理连接的控制台中登录。要防止 root 用户登录，请以 root 用户身份在 shell 提示符下输入以下命令来删除此文件的内容：

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

要在 KDM、GDM 和 XDM 登录管理器中启用 securetty 支持，请添加以下行：

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

以下列出的文件：

- /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 /etc/securetty 文件不会阻止 root 用户使用 OpenSSH 工具套件进行远程登录，因为在身份验证后不会打开控制台。

表 4.3. 禁用 root 登录

影响	*未受影响*
阻止使用控制台或网络访问 root 帐户。以下程序无法访问 root 帐户： • *login* • *gdm* • *kdm* • *XDM* • *打开 tty 的其他网络服务*	*不以 root 身份登录但通过 setuid 或其他机制执行管理任务的程序。以下程序 不会阻止 访问 root 帐户：* • *su* • *sudo* • *ssh* • *scp* • *sftp*

禁用 root SSH 登录

要防止 root 通过 SSH 协议登录，请编辑 SSH 守护进程的配置文件 /etc/ssh/sshd_config，并更改如下行：

```
#PermitRootLogin yes
```

如下所示：

```
PermitRootLogin no
```

表 4.4. 禁用 root SSH 登录

影响	未受影响
使用 OpenSSH 工具套件进行 root 访问。以下程序无法访问 root 帐户： • ssh • scp • sftp	不属于 OpenSSH 工具套件的程序。

使用 PAM 限制对服务的 root 访问权限

PAM 通过 /lib/security/pam_listfile.so 模块，在拒绝特定帐户时具有很大的灵活性。管理员可以使用此模块来引用不允许登录的用户列表。要限制对系统服务的 root 访问权限，请编辑 /etc/pam.d/ 目录中目标服务的 文件，并确保验证需要 pam_listfile.so 模块。

以下是如何将模块用于 /etc/pam.d/ vsftpd PAM 配置文件中的 vsftpd FTP 服务器（如果指令位于一行中，则不需要在 第一行末尾的 \ 字符）：

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

-

这指示 PAM 查阅 /etc/vsftpd.ftpusers 文件，并拒绝访问任何列出用户的服务。管理员可以更改此文件的名称，并可为每个服务保留单独的列表，或使用一个中央列表拒绝对多个服务的访问。

如果管理员希望拒绝对多个服务的访问，可以将类似的行添加到 PAM 配置文件，如 /etc/pam.d/pop 和 /etc/pam.d/imap 用于邮件客户端，或 /etc/pam.d/ssh 用于 SSH 客户端。

有关 PAM 的更多信息，请参阅 Linux-PAM 系统管理员指南，位于 /usr/share/doc/pam-<version>/html/ 目录中。

表 4.5. 使用 PAM 禁用 Root

影响	未受影响
防止对 PAM 感知的网络服务的 root 访问。以下服务无法访问 root 帐户： • login • gdm • kdm • XDM • ssh • scp • sftp • FTP 客户端 • 电子邮件客户端 • 任何 PAM 感知服务	不了解 PAM 的程序和服务.

4.2.2. 允许根访问

如果机构中的用户受信任且计算机同步，则允许他们 root 访问权限不是问题。用户允许 root 访问权限意味着，像添加设备或配置网络接口等小活动可由单个用户处理，让系统管理员可以自由处理网络安全性和其他重要问题。

另一方面，为单个用户提供 root 访问权限可能会导致以下问题：

- Machine Misconfiguration - 具有 root 访问权限的用户可能会错误地配置其机器，并需要帮助解决问题。甚至更糟糕，它们可能会在不知情的情况下打开安全漏洞。
- 运行 Insecure Services - 具有 root 访问权限的用户可能会在其计算机上运行不安全的服务器，如 FTP 或 Telnet，可能会使用户名和密码面临风险。这些服务通过网络以纯文本形式传输此信息。
- 以 Root 身份运行电子邮件附件 - 尽管存在影响 Linux 的电子邮件病毒。恶意计划由 root 用户运行时构成了最大的威胁。
- 保持审计跟踪不变 - 因为 root 帐户通常由多个用户共享，因此多个系统管理员可以维护系统，因此无法找出这些用户在给定时间是 root 用户。使用单独的登录时，通过 登录帐户以及会话跟踪目的的唯一编号将置于任务结构中，由用户启动的每个进程继承。使用并发登录时，可以使用唯一数字来跟踪操作到特定登录。当某个操作生成审计事件时，它会与登录帐户以及与该唯一数字关联的会话记录。使用 aulast 命令查看这些登录和会话。aulast 命令的 -- effective 选项可以建议一个特定的 ausearch 查询来隔离特定会话生成的可审计事件。有关审计系统的更多信息，请参阅 第 7 章 系统审计。

4.2.3. 限制根访问

管理员可能只想通过 setuid 程序（如 su 或 sudo ）允许访问，而不是完全拒绝对 root 用户的访问。有关 su 和 sudo 的更多信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的 获取特权 章节，以及 su (1) 和 sudo (8) 手册页。

4.2.4. 启用自动注销

当用户以 root 身份登录时，无人值守登录会话可能会导致严重的安全风险。要降低这个风险，您可以将系统配置为在固定时间段内自动注销闲置用户。

1. 以 root 用户身份，在 /etc/profile 文件的开头添加以下行，以确保无法中断此文件的处理：

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

2. 以 root 用户身份，将以下行插入到 /etc/profile 文件中，以便在 120 秒后自动注销：

```
export TMOUT=120
readonly TMOUT
```

如果指定秒数没有活动（上例中设为 120），则 TMOUT 变量终止 shell。您可以根据特定安装的需求更改限制。

4.2.5. 保护 Boot Loader

密码保护 Linux 引导装载程序的主要原因如下：

1. 防止访问单用户模式 - 如果攻击者可以将系统引导至单用户模式，则它们会自动以 root 身份登录，而不会提示输入 root 密码。

警告

不建议通过编辑 /etc/sysconfig/init 文件中的 SINGLE 参数来保护对单用户模式的访问。攻击者可以通过在 GRUB 2 的内核命令行中指定自定义初始命令（使用 init= 参数）来绕过密码。建议对 GRUB 2 引导装载程序进行密码保护，如 Red Hat Enterprise Linux 7 系统管理员指南中的 使用密码保护 GRUB 2 章节中所述。

2. 防止访问 GRUB 2 控制台 - 如果机器使用 GRUB 2 作为其引导装载程序，攻击者可以使用 GRUB 2 编辑器界面更改其配置或使用 cat 命令来收集信息。

3. 防止对 Insecure Operating Systems 的访问 - 如果它是一个双引导系统，攻击者可以在引导时选择操作系统，例如 DOS，它会忽略访问控制和文件权限。

Red Hat Enterprise Linux 7 在 Intel 64 和 AMD64 平台上包括 GRUB 2 引导装载程序。有关 GRUB 2 的详细信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的 使用 GRUB 2 Boot Loader 章节。

4.2.5.1. 禁用交互式启动

在启动序列开始时按 I 键可让您以交互方式启动您的系统。在交互式启动过程中，系统会提示您逐一启动每个服务。但是，这可能会允许获得系统物理访问权限的攻击者禁用与安全相关的服务，并可以访问该系统。

要防止用户以 root 用户身份启动系统，以 root 用户身份在 /etc/sysconfig/init 文件中禁用 PROMPT 参数：

```
PROMPT=no
```

4.2.6. 保护硬链接和符号链接

为了防止恶意用户利用未经保护的硬链接和符号链接导致的潜在漏洞，Red Hat Enterprise Linux 7 包含一个仅允许创建或遵循某些条件的链接的功能。

如果是硬链接，则需要满足以下条件之一：

- 用户拥有其链接的文件。
- 用户已对其链接的文件具有读写访问权限。

如果是符号链接，只有当具有粘滞位的全局可写目录外，或者需要满足以下条件之一时，进程才被允许遵循链接：

- 符号链接后面的进程是符号链接的所有者。
- 目录的所有者与符号链接的所有者相同。

默认开启这个保护。它由 /usr/lib/sysctl.d/50-default.conf 文件中的以下选项控制：

```
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
```

要覆盖默认设置并禁用保护，请在 /etc/sysctl.d/ 目录中创建一个名为 51-no-protect-links.conf 的新配置文件，其内容如下：

```
fs.protected_hardlinks = 0
fs.protected_symlinks = 0
```

注意

请注意，为了覆盖默认系统设置，新配置文件需要具有 .conf 扩展名，且需要在默认系统 文件后 读取（文件以字典顺序读取，因此文件名开头的设置具有较高的数字）。

有关使用 sysctl 机制在引导时配置内核参数的详情，请查看 sysctl.d(5) 手册页。

4.3. 保护服务

虽然用户对组织内的系统管理员而言是管理控制的一个重要问题，但监控哪些网络服务对于管理和运行 Linux 系统的人员至关重要。

Red Hat Enterprise Linux 7 下的许多服务都是网络服务器。如果网络服务在计算机上运行，则服务器应用（称为 守护进程）正在侦听一个或多个网络端口的连接。这些服务器的每个服务器都应被视为潜在的攻击。

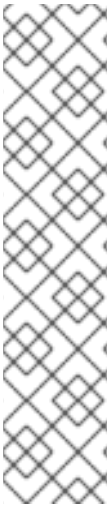
4.3.1. 对服务的风险

网络服务可能会给 Linux 系统带来很多风险。以下是一些主要问题的列表：

- 拒绝 Service Attacks (DoS) - 通过向服务填充请求，拒绝服务攻击可能会导致系统因为尝试记录并回答每个请求而不可用。
- Service Attack (DDoS)的分布式拒绝(DDoS) - 使用多个受损机器（通常以千计或更多个）对服务进行协调攻击，利用请求并无法使用它。
- 脚本漏洞攻击 - 如果服务器使用脚本来执行服务器端操作，作为 Web 服务器通常做的，攻击

者可以将不正确的编写脚本作为目标。这些脚本漏洞攻击可能会导致缓冲区溢出状况，或者允许攻击者更改系统上的文件。

- buffer Overflow Attacks - 希望侦听端口 1 到 1023 的服务必须以管理特权启动，或者需要为它们设置 CAP_NET_BIND_SERVICE 功能。当进程绑定到端口并正在侦听它后，通常会丢弃特权或功能。如果没有丢弃特权或功能，且应用程序有可被利用的缓冲区溢出，攻击者可能会作为运行守护进程的用户访问系统。由于存在可利用的缓冲区溢出，因此攻击者使用自动化工具来识别具有漏洞的系统，一旦获得访问，它们使用自动化的 rootkits 来保持对系统的访问权限。



注意

在 Red Hat Enterprise Linux 7 中，执行Shield （可执行内存分段和保护技术）中可以缓解缓冲区溢出漏洞的威胁。execshield 通过将虚拟内存划分为可执行和非可执行文件段来降低缓冲区溢出的风险。试图在可执行段外执行的任何程序代码（如从缓冲区溢出漏洞注入的恶意代码）都会触发分段错误并终止。

execshield 还包括对 AMD64 平台和 Intel® 64 系统上的 No eXecute (NX)技术的支持。这些技术与 ExecShield 结合使用，以防止恶意代码在虚拟内存的可执行部分以 4KB 的可执行代码运行，从而降低攻击缓冲区溢出漏洞的风险。

重要

要限制暴露会受到攻击的网络，应关闭所有未使用的服务。

4.3.2. 识别和配置服务

为增强安全性，Red Hat Enterprise Linux 7 安装的大多数网络服务都会默认关闭。然而，有一些值得注意的例外：

- cups - Red Hat Enterprise Linux 7 的默认打印服务器。
- cups-lpd - 备用打印服务器。
- xinetd - 控制到一系列从属服务器（如 gssftp 和 telnet ）连接的超级服务器。
- sshd - OpenSSH 服务器，这是 Telnet 的安全替换。

在确定这些服务是否运行时，最好使用常见意义，并避免承担任何风险。例如，如果打印机不可用，请不要让 cups 运行。对于 portreserve，也是如此。如果您没有挂载 NFSv3 卷或使用 NIS (ypbind 服务)，则应禁用 rpcbind。检查哪些网络服务可在引导时启动是不够的。建议还要检查哪些端口处于打开状态并侦听。如需更多信息，请参阅 第 4.4.2 节 “验证正在列出哪些端口”。

4.3.3. 不安全的服务

任何网络服务都不安全。这就是关闭未使用的服务非常重要的原因。对服务的利用会频繁发现和修补，因此定期更新与任何网络服务关联的软件包非常重要。请参阅 第 3 章 使您的系统保持最新状态 了解更多信息。

某些网络协议本质上比其他协议更安全。这包括以下任何服务：

- 通过网络 Unencrypted - 很多较旧的协议（如 Telnet 和 FTP）传输 用户名和密码，不要加密身份验证会话，并应尽可能避免。
- 通过网络 Unencrypted 传输敏感数据 - 很多协议通过网络未加密的传输数据。这些协议包括 Telnet、FTP、HTTP 和 SMTP。许多网络文件系统（如 NFS 和 SMB ）也通过网络未加密的传输信息。在使用这些协议来限制传输的数据类型时，用户的职责。

本质上不安全的服务示例包括 rlogin、rsh、telnet 和 vsftpd。

所有远程登录和 shell 程序(rlogin、rsh 和 telnet)都应避免使用 SSH。有关 sshd 的详情，请查看 第 4.3.11 节 “保护 SSH”。

FTP 并不像远程 shell 一样对系统的安全性有一定的危险，但必须仔细配置并监控 FTP 服务器以避免出现问题。有关保护 FTP 服务器的详情，请查看 第 4.3.9 节 “保护 FTP”。

应仔细实现的服务，并在防火墙后面包括：

- auth
- nfs-server

- SMB 和 nbm (Samba)
- yppasswdd
- ypserv
- ypxfrd

有关保护网络服务安全的更多信息，请参阅 第 4.4 节 “保护网络访问”。

4.3.4. 保护 rpcbind

rpcbind 服务是 RPC 服务的动态端口分配守护进程，如 NIS 和 NFS。它有较弱的身份验证机制，并可为其控制的服务分配大量端口。因此，很难保护。

注意

保护 rpcbind 仅影响 NFSv2 和 NFSv3 实现，因为 NFSv4 不再需要它。如果您计划实施 NFSv2 或 NFSv3 服务器，则需要 rpcbind，并且适用以下部分。

如果运行 RPC 服务，请遵循以下基本规则。

4.3.4.1. 使用 TCP wrapper 保护 rpcbind

务必要使用 TCP wrapper 来限制哪些网络或主机可以访问 rpcbind 服务，因为它没有内置的身份验证形式。

此外，在限制对该服务的访问时，只使用 IP 地址。避免使用主机名，因为它们可以通过 DNS poisoning 和其他方法进行伪造。

4.3.4.2. 使用 firewalld 保护 rpcbind

要进一步限制对 rpcbind 服务的访问，最好将 firewalld 规则添加到服务器，并限制对特定网络的访

问。

以下是 firewalld 丰富的语言命令示例。第一个允许从 192.168.0.0/24 网络到端口 111 （由 rpcbind 服务使用）的 TCP 连接。第二个允许从 localhost 到同一端口的 TCP 连接。 所有其他数据包都将被丢弃。

```
~]# 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'
```

要类似限制 UDP 流量，请使用以下命令：

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

注意

将 --permanent 添加到 firewalld 丰富的语言命令中，使设置永久生效。有关实现防火墙的详情，请查看 第 5 章 使用防火墙。

4.3.5. 保护 rpc.mountd

rpc.mountd 守护进程实现 NFS MOUNT 协议的服务器端，NFS 版本 2 (RFC 1904)和 NFS 版本 3 (RFC 1813)使用的协议。

如果运行 RPC 服务，请遵循以下基本规则。

4.3.5.1. 使用 TCP Wrappers 保护 rpc.mountd

使用 TCP Wrappers 限制哪些网络或主机可以访问 rpc.mountd 服务非常重要，因为它没有内置的身份验证形式。

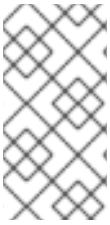
此外，在限制对该服务的访问时，只使用 IP 地址。避免使用主机名，因为它们可以通过 DNS poisoning 和其他方法进行伪造。

4.3.5.2. 使用 firewalld 保护 rpc.mountd

要进一步限制对 rpc.mountd 服务的访问，请在服务器中添加 firewalld 丰富的语言规则并限制对特定网络的访问。

以下是 firewalld 丰富的语言命令示例。第一个允许从 192.168.0.0/24 网络挂载连接。第二个 允许从本地主机挂载连接。所有其他数据包都将被丢弃。

```
~]# firewall-cmd --add-rich-rule 'rule family="ipv4" source NOT address="192.168.0.0/24" service
name="mountd" drop'
~]# firewall-cmd --add-rich-rule 'rule family="ipv4" source address="127.0.0.1" service
name="mountd" accept'
```



注意

将 --permanent 添加到 firewalld 丰富的语言命令中，使设置永久生效。有关实现防火墙的详情，请查看 第 5 章 使用防火墙。

4.3.6. 保护 NIS

网络信息服务 (NIS)是一个 RPC 服务，称为 ypserv，它将与 rpcbind 和其他相关服务结合使用，用于将用户名、密码和其他敏感信息分发到其域中的任何计算机。

NIS 服务器由多个应用程序组成。它们包括以下内容：

- /usr/sbin/rpc.yppasswdd - 也称为 yppasswdd 服务，此守护进程允许用户更改其 NIS 密码。
- /usr/sbin/rpc.ypxfrd - 还称 ypxfrd 服务，此守护进程负责 NIS 通过网络传输。
- /usr/sbin/ypserv - 这是 NIS 服务器守护进程。

NIS 因当今的标准而不安全。它没有主机身份验证机制，并通过未加密的网络传输所有信息，包括密码哈希。因此，设置使用 NIS 的网络时必须非常小心。这进一步复杂，NIS 的默认配置本质上不安全。

建议任何计划实施 NIS 服务器的用户首先保护 rpcbind 服务，如 第 4.3.4 节 “保护 rpcbind” 所述，然后解决以下问题，如网络规划。

4.3.6.1. 仔细规划网络

由于 NIS 通过网络传输未加密的敏感信息，因此该服务必须在防火墙后面和分段安全网络后面运行。每当通过不安全的网络传输 NIS 信息时，都会拦截它的风险。仔细的网络设计可帮助防止严重的安全漏洞。

4.3.6.2. 使用类似密码的 NIS 域名和主机名

NIS 域中的任何机器都可以使用命令从服务器中提取信息，而无需身份验证，只要用户知道 NIS 服务器的 DNS 主机名和 NIS 域名。

例如，如果某家计算机连接到网络，或者从外部中断到网络（并管理到欺骗内部 IP 地址），以下命令显示 /etc/passwd 映射：

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

如果这个攻击者是一个 root 用户，可以通过输入以下命令来获取 /etc/shadow 文件：

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

注意

如果使用 Kerberos，则 /etc/shadow 文件不会存储在 NIS 映射中。

要使对 NIS 映射的访问更难于攻击者，请为 DNS 主机名创建一个随机字符串，如 o7hfawtgmhwg.domain.com。同样，创建不同的 随机 NIS 域名。这使得攻击者更难以访问 NIS 服务器。

4.3.6.3. 编辑 /var/yp/securenets 文件

如果 /var/yp/securenets 文件为空或不存在（如默认安装后的情况），NIS 侦听所有网络。首先要做的事情之一是将子网掩码/网络对放在文件中，以便 ypserv 仅响应来自适当网络的请求。

以下是 /var/yp/securenets 文件中的示例条目：

```
255.255.255.0     192.168.0.0
```

警告

在不创建 /var/yp/securenets 文件的情况下，不要在第一次启动 NIS 服务器。

这个技术不提供对 IP 欺骗攻击的保护，但它至少对 NIS 服务器服务的网络上的限制。

4.3.6.4. 分配静态端口和使用 Rich Language 规则

与 NIS 相关的所有服务器都可以被分配除 rpc.yppasswdd 以外的特定端口 - 允许用户更改其登录密码的守护进程。将端口分配给其他两个 NIS 服务器守护进程 rpc.ypxfrd 和 ypserv，允许创建防火墙规则来进一步保护 NIS 服务器守护进程不受入侵者的影响。

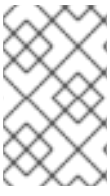
要做到这一点，请在 /etc/sysconfig/network 中添加以下行：

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

然后，以下丰富的语言 firewalld 规则可用于强制服务器侦听这些端口的网络：

```
~]# 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'
```

这意味着，如果请求来自 192.168.0.0/24 网络，服务器仅允许连接到端口 834 和 835。第一个规则用于 TCP，第二个规则用于 UDP。



注意

有关使用 iptables 命令实现防火墙的详情，请参考 第 5 章 使用防火墙。

4.3.6.5. 使用 Kerberos 身份验证

使用 NIS 进行身份验证时需要考虑的问题之一是，每当用户登录计算机时，都会通过网络发送

/etc/shadow 映射中的密码哈希。如果入侵者获得了对 NIS 域的访问并嗅探网络流量，他们可以收集用户名和密码哈希。如果有足够的时间，攻击者可能会猜测弱密码，攻击者可以获得对网络上有效帐户的访问权限。

由于 Kerberos 使用 secret 密钥加密，所以不会通过网络发送密码哈希，使系统更安全。有关 Kerberos 的更多信息，请参阅 Linux 域身份、身份验证和策略指南中的使用 Kerberos 登录到 IdM 部分。

4.3.7. 保护 NFS

重要

可以在所有版本中使用 TCP 发送 NFS 流量，它应当与 NFSv3 一起使用，而不是使用 UDP，在使用 NFSv4 时是必需的。作为 RPCSEC_GSS 内核模块的一部分，NFS 的所有版本都支持 Kerberos 用户和组身份验证。仍然包含 rpcbind 的信息，因为 Red Hat Enterprise Linux 7 支持使用 rpcbind 的 NFSv3。

4.3.7.1. 仔细规划网络

传统上 NFSv2 和 NFSv3 传递数据。现在，NFS 的所有版本都能够使用 Kerberos 验证（并选择性地加密）普通文件系统操作。在 NFSv4 下，所有操作都可以使用 Kerberos；在 NFSv2 或 NFSv3 下，文件锁定和挂载仍无法使用它。使用 NFSv4.0 时，如果客户端位于 NAT 或防火墙后面，则可以关闭委派。有关使用 NFSv4.1 来允许委托通过 NAT 和防火墙操作的详情，请参考 Red Hat Enterprise Linux 7 存储管理指南 的 pNFS 部分。

4.3.7.2. 保护 NFS 挂载选项

在 /etc/fstab 文件中介绍了使用 mount 命令，请参见 Red Hat Enterprise Linux 7 存储管理指南 中的" 使用 mount 命令 "章节。从安全管理的角度来看，值得注意，也可以在 /etc/nfsmount.conf 中指定 NFS 挂载选项，可用于设置自定义默认选项。

4.3.7.2.1. 查看 NFS 服务器

警告

仅导出整个文件系统。导出文件系统的子目录可能是安全问题。在有些情况下，客户端可能会"破坏"文件系统的导出部分，并得到取消导出部分（请参阅 exports (5) 手册页中有关子树检查的部分）。

使用 ro 选项将文件系统导出为只读文件系统，以减少用户可以写入挂载的文件系统的用户数量。仅在需要时使用 rw 选项。详情请查看 man exports (5) 页面。例如，允许写入访问会增加符号链接攻击的风险。这包括临时目录，如 /tmp 和 /usr/tmp。

必须使用 rw 选项挂载目录的位置，避免尽可能使目录全局可写，以降低风险。导出主目录也被视为风险，因为有些应用以明文或弱方式加密存储密码。随着应用程序代码被检查并改进，这个风险会降低。有些用户没有在 SSH 密钥上设置密码，因此这也意味着主目录会带来风险。强制使用密码或使用 Kerberos 可降低该风险。

只将导出限制给需要访问权限的客户端。在 NFS 服务器上使用 showmount -e 命令来检查服务器正在导出的内容。不要导出不需要的任何内容。

不要使用 no_root_squash 选项，并查看现有安装以确保不使用它。请参阅 第 4.3.7.4 节 “不要使用 no_root_squash 选项” 了解更多信息。

secure 选项是用于将导出限制到 “保留端口” 的服务器端导出选项。默认情况下，服务器仅允许来自 “保留端口” 的客户端通信（编号小于 1024 的端口），因为传统客户端只有允许的 “可信” 代码（如内核 NFS 客户端）使用这些端口。但是，在很多网络上，任何人无法在某些客户端上成为 root 用户，因此，对于服务器来说，假设来自保留端口的通信都具有特权非常安全。因此，对保留端口的限制具有有限的值；最好根据 Kerberos、防火墙和对特定客户端的导出限制来决定。

大多数客户端仍然尽可能使用保留的端口。但是，保留的端口是一个有限的资源，因此客户端（特别是那些具有大量 NFS 挂载的客户端）也可以选择使用高数字的端口。Linux 客户端可以使用 “noresvport” 挂载选项进行此操作。如果要在导出上允许此操作，您可以使用 “insecure” export 选项进行此操作。

最好不允许用户登录到服务器。查看 NFS 服务器上的以上设置时，请查看谁和什么可以访问服务器。

4.3.7.2.2. 查看 NFS 客户端

使用 nosuid 选项禁止使用 setuid 程序。nosuid 选项禁用 set-user-identifier 或 set-group-identifier 位。这可防止远程用户通过运行 setuid 程序获得更高的特权。在客户端和服务器端使用这个选项。

noexec 选项禁用客户端上的所有可执行文件。使用此选项来防止用户意外执行放在被共享的文件系统中的文件。nosuid 和 noexec 选项对于大多数都不是所有文件系统的标准选项。

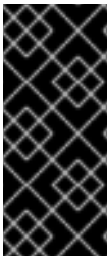
使用 nodev 选项防止 “device-files” 被客户端作为硬件设备处理。

resvport 选项是一个客户端挂载选项，而 secure 是对应的服务器端导出选项（请参阅上述说明）。它限制了与"保留端口"的通信。保留或"well known"端口为特权用户和进程保留，如 root 用户。设置此选项可让客户端使用保留源端口与服务器通信。

NFS 的所有版本现在支持使用 Kerberos 身份验证挂载。启用此选项的挂载选项为： sec=krb5。

NFSv4 支持对完整性使用 krb5i 的 Kerberos 挂载，使用 krb5p 进行隐私保护。使用 sec=krb5 挂载时会使用它们，但需要在 NFS 服务器上进行配置。如需更多信息，请参阅 exports (man 5 导出)的 man page。

NFS man page (man 5 nfs)有一个 “SECURITY CONSIDERATIONS” 部分，它解释了 NFSv4 中的安全增强，并包含所有 NFS 特定的挂载选项。



重要

krb5-libs 软件包提供的 MIT Kerberos 库不支持在新部署中使用数据加密标准(DES)算法。由于安全性和某些兼容性原因，在 Kerberos 库中，DES 默认被弃用并禁用。只有在您的环境不支持任何更新和更安全得算法时，才出于兼容性的原因使用 DES。

4.3.7.3. 语法错误

NFS 服务器通过咨询 /etc/exports 文件来确定要导出哪些主机以及要将这些目录导出到哪个文件系统。在编辑此文件时，请小心不要添加额外的空格。

例如，/etc/exports 文件中的以下行将目录 /tmp/nfs/ 共享到主机 bob.example.com，其读/写权限。

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

另一方面，/etc/exports 文件中的以下行与主机 bob.example.com 共享同一目录，并具有只读权限，因为主机名后面有一个空格字符来与 全局 共享。

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

使用 showmount 命令检查任何配置的 NFS 共享是不错的做法：

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

4.3.7.4. 不要使用 no_root_squash 选项

默认情况下，NFS 共享将 root 用户改为 nfsnobody 用户（非特权用户帐户）。这会将所有 root 创建文件的所有者更改为 nfsnobody，这样可防止上传设置了 setuid 位的程序。

如果使用 no_root_squash，则远程 root 用户可以更改共享文件系统上的任何文件，并将 Trojans 破坏的应用程序留给其他用户意外执行。

4.3.7.5. NFS 防火墙配置

NFSv4 是 Red Hat Enterprise Linux 7 的 NFS 的默认版本，它只需要为 TCP 打开端口 2049。如果使用 NFSv3，则需要四个额外的端口，如下所述。

为 NFSv3 配置端口

用于 NFS 的端口由 rpcbind 服务动态分配，这可能会在创建防火墙规则时造成问题。要简化这个过程，请使用 /etc/sysconfig/nfs 文件指定要使用的端口：

- MOUNTD_PORT - mountd 的 TCP 和 UDP 端口(rpc.mountd)
- STATD_PORT - status (rpc.statd)的 TCP 和 UDP 端口

在 Red Hat Enterprise Linux 7 中，在 /etc/modprobe.d/lockd.conf 文件中为 NFS 锁定管理器 (nlockmgr)设置 TCP 和 UDP 端口：

- nlm_tcpport - nlockmgr (rpc.lockd)的 TCP 端口
- nlm_udpport - UDP 端口 nlockmgr (rpc.lockd)

指定的端口号不得被任何其他服务使用。将您的防火墙配置为允许指定的端口号，以及 TCP 和 UDP 端口 2049 (NFS)。有关其他可自定义 NFS 锁定管理器参数的描述，请参阅

/etc/modprobe.d/lockd.conf。

在 NFS 服务器上运行 rpcinfo -p 命令，以查看正在使用的端口和 RPC 程序。

4.3.7.6. 使用红帽身份管理保护 NFS

在使用 Red Hat Identity Management （包括在 Red Hat Enterprise Linux 中）的环境中可以大大简化 Kerberos 感知 NFS 设置。

请参阅 Red Hat Enterprise Linux 7 Linux 域身份、身份验证和策略指南，特别是 设置 Kerberos 感知 NFS 服务器 以了解如何在使用 Red Hat Identity Management 时使用 Kerberos 保护 NFS。

4.3.8. 保护 HTTP 服务器

4.3.8.1. 保护 Apache HTTP 服务器

Apache HTTP 服务器是 Red Hat Enterprise Linux 7 中最稳定和安全的服务之一。有很多选项和技术可用于保护 Apache HTTP 服务器 - 这太多是为了深入处理 Apache HTTP 服务器。以下章节简要解释了运行 Apache HTTP 服务器时的良好做法。

在将脚本放入生产 之前，请始终验证系统上运行的任何脚本是否按预期工作。此外，确保只有 root 用户对包含脚本或 CGI 的任何目录具有写入权限。要做到这一点，以 root 用户身份输入以下命令：

```
chown root <directory_name>
```

```
chmod 755 <directory_name>
```

使用以下配置选项（在 /etc/httpd/conf/httpd.conf中配置）时，系统管理员应小心：

FollowSymLinks

默认情况下，这个指令是启用的，因此请务必在创建 Web 服务器文档根的符号链接时小心。例如，最好提供指向 / 的符号链接。

索引

这个指令默认为启用，但可能不需要。要防止 visitors 浏览服务器上的文件，请删除此指令。

UserDir

默认情况下，UserDir 指令被禁用，因为它可以确认系统中存在用户帐户。要在服务器上启用用户目录浏览，请使用以下指令：

```
UserDir enabled
    UserDir disabled root
```

*这些指令激活用户目录浏览 /**root**/ 以外的所有用户目录。要将用户添加到禁用帐户列表中，请在 **UserDir disabled** 行中添加以空格分隔的用户列表。*

ServerTokens

ServerTokens 指令控制发送到客户端的服务器响应标头字段。它包括可使用以下参数自定义的各种信息：

- ***ServerTokens Full （默认选项）- 提供所有可用信息(OS 类型和使用的模块)，例如：***

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

- ***ServerTokens Prod 或 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
  ```

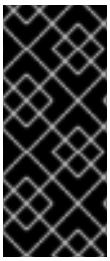
- ***ServerTokens Min 或 ServerTokens Minimal - 提供以下信息：***

  ```
  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，以便可能的攻击者不会获取您系统的任何宝贵信息。



重要

不要删除 IncludesNoExec 指令。默认情况下，Server-Side Includes (SSI)模块无法执行命令。建议您不要更改此设置，除非绝对必要，因为它可能会使攻击者在系统中执行命令。

删除 httpd 模块

在某些情况下，删除某些 httpd 模块来限制 HTTP 服务器的功能是很有帮助的。为此，请编辑 /etc/httpd/conf.modules.d 目录中的配置文件。例如，要删除代理模块：

```
echo '# All proxy modules disabled' > /etc/httpd/conf.modules.d/00-proxy.conf
```

请注意，/etc/httpd/conf.d/ 目录还包含用于加载模块的配置文件。

httpd 和 SELinux

如需更多信息，请参阅 Red Hat Enterprise Linux 7 SELinux 用户和管理员指南中的 Apache HTTP 服务器和 SELinux 章节。

4.3.8.2. 保护 NGINX

NGINX 是一个高性能 HTTP 和代理服务器。本节简要记录了强化 NGINX 配置的其他步骤。在 NGINX 配置文件的 server 部分中执行以下所有配置更改。

禁用版本字符串

要防止攻击者了解服务器上运行的 NGINX 版本，请使用以下配置选项：

```
server_tokens       off;
```

这会影响版本号，只需报告由 NGINX 提供的所有请求中的字符串 nginx 的影响：

```
$ curl -sI http://localhost | grep Server
Server: nginx
```

包括其他与安全相关的标头

NGINX 提供的每个请求都可以包括额外的 HTTP 标头来缓解某些已知的 Web 应用程序漏洞：

- add_header X-Frame-Options SAMEORIGIN； - 此选项拒绝域之外的任何页面来帧由 NGINX 提供的任何内容，从而有效地缓解了攻击。
- add_header X-Content-Type-Options nosniff; - 这个选项在某些较旧的浏览器中防止 MIME 类型嗅探。
- add_header X-XSS-Protection "1; mode=block"; - 这个选项启用跨站点脚本过滤(XSS)过滤，这可以防止浏览器渲染由 NGINX 响应中包含的潜在的恶意内容。

禁用 Potentially Harmful HTTP 方法

如果启用，某些 HTTP 方法可能会允许攻击者对专为开发人员测试 Web 应用程序的 Web 服务器执行操作。例如，TRACE 方法已知允许跨站点追踪(XST)。

您的 NGINX 服务器可以通过只列入允许的用户来禁止这些有害 HTTP 方法以及任何任意方法。例如：

```
# Allow GET, PUT, POST; return "405 Method Not Allowed" for all others.
if ( $request_method !~ ^(GET|PUT|POST)$ ) {
    return 405;
}
```

配置 SSL

要保护 NGINX web 服务器提供的数据，请考虑仅通过 HTTPS 提供它。要在 NGINX 服务器中为启用 SSL 生成安全配置配置文件，请参阅 Mozilla SSL 配置生成器。生成的配置可确保禁用已知存在安全漏洞的协议（如 SSLv2 或 SSLv3、密码和哈希算法（例如 3DES 或 MD5）。

您还可以使用 SSL 服务器测试 验证您的配置是否满足现代安全要求。

4.3.9. 保护 FTP

文件传输协议 (FTP)是一种较旧的 TCP 协议，旨在通过网络传输文件。因为与服务器进行的所有事务（包括用户身份验证）都是未加密的，所以它被视为不安全的协议，应该仔细配置。

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

- 红帽内容加速器 (tux)- 具有 FTP 功能的内核空间 Web 服务器。
- vsftpd - FTP 服务的独立、面向安全的实现。

以下安全指南是设置 vsftpd FTP 服务。

4.3.9.1. FTP Greeting Banner

在提交用户名和密码前，所有用户都会看到问候横幅。默认情况下，此横幅包含有助于识别系统中弱点的版本信息。

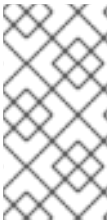
要更改 vsftpd 的问候横幅，请在 /etc/vsftpd/vsftpd.conf 文件中添加以下指令：

```
ftpd_banner=<insert_greeting_here>
```

将上述指令中的 < insert_greeting_here > 替换为问候消息的文本。

对于 Mutli-line banners，最好使用横幅文件。要简化对多个横幅的管理，请将所有横幅放在名为 /etc/banners/ 的新目录中。本例中 FTP 连接的横幅文件是 /etc/banners/ftp.msg。以下是此类文件的一个示例：

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



注意

不需要使用在 第 4.4.1 节 “使用 TCP wrapper 和 xinetd 保护服务” 中指定的 220 文件开始每行。

要引用 vsftpd 的这一问候标语文件，请在 /etc/vsftpd/vsftpd.conf 文件中添加以下指令：

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

您还可以使用 TCP Wrappers 将额外的横幅发送到传入的连接，如 第 4.4.1.1 节 “TCP 包装器和连接标语” 所述。

4.3.9.2. Anonymous Access（匿名访问）

存在 /var/ftp/ 目录可激活匿名帐户。

创建此目录的最简单方法是安装 vsftpd 软件包。这个软件包为匿名用户建立目录树，并为匿名用户将目录的权限配置为只读。

默认情况下，匿名用户无法写入任何目录。

警告

如果启用对 FTP 服务器的匿名访问，请注意敏感数据的存储位置。

4.3.9.2.1. 匿名上传

要允许匿名用户上传文件，建议在 /var/ftp/pub/ 中创建只写目录。要做到这一点，请以 root 用户身份输入以下命令：

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

接下来，更改权限，以便匿名用户无法查看目录的内容：

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

目录的长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进行读写的管理员通常会发现其服务器成为盗窃软件的存储库。

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

```
anon_upload_enable=YES
```

4.3.9.3. 用户帐户

由于 FTP 通过不安全的网络传输未加密的用户名和密码进行身份验证，因此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，请在 /etc/vsftpd/vsftpd.conf 中添加以下指令：

```
local_enable=NO
```

4.3.9.3.1. 限制用户帐户

要禁用特定帐户或特定帐户组（如 root 用户和具有 sudo 权限的用户）的 FTP 访问，最简单的方法是使用 PAM 列表文件，如 第 4.2.1 节 “禁止 Root 访问” 所述。vsftpd 的 PAM 配置文件为 /etc/pam.d/vsftpd。

也可以直接禁用每个服务中的用户帐户。

要在 vsftpd 中禁用特定的用户帐户，请将用户名添加到 /etc/vsftpd/ftpusers

4.3.9.4. 使用 TCP wrapper 控制访问

按照 第 4.4.1 节 “使用 TCP wrapper 和 xinetd 保护服务” 所述，使用 TCP wrapper 控制对 FTP 守护进程的访问。

4.3.10. 保护 Postfix

Postfix 是一个邮件传输代理(MTA)，它使用简单邮件传输协议(SMTP)在其他 MTA 之间发送电子邮件，以及电子邮件客户端或发送代理。虽然很多 MTA 能够在另一个 MTA 之间加密流量，但大多数都不允许，因此通过任何公共网络发送电子邮件被视为本质上是不安全的通信形式。Postfix 将 Sendmail 取代为 Red Hat Enterprise Linux 7 中的默认 MTA。

建议计划实施 Postfix 服务器的任何人都解决了以下问题。

4.3.10.1. 限制服务攻击

由于电子邮件的性质，确定的攻击者可能会非常轻松地使用邮件来填充服务器，并导致拒绝服务。可以通过设置 /etc/postfix/main.cf 文件中的指令限制来限制此类攻击的有效性。您可以更改已有指令的值，或者您可以使用以下格式所需的值添加所需指令：

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

.以下是可用于限制拒绝服务攻击的指令列表：

- smtpd_client_connection_rate_limit - 允许每个时间单位对这个服务进行的最大连接尝试次数（如下所述）。默认值为 0，这意味着客户端每次时间单位可以接收 Postfix 可以接受的连接数。默认情况下，可信网络中的客户端会被排除。
- anvil_rate_time_unit - 此时间单位用于速率限制计算。默认值为 60 秒。
- smtpd_client_event_limit_exceptions - 从连接和速率限制命令中排除的客户端。默认情况下，可信网络中的客户端会被排除。
- smtpd_client_message_rate_limit - 允许客户端按时间单位请求的最大消息数（无论 Postfix 是否实际接受这些消息）。
- default_process_limit - 提供给定服务的 Postfix 子进程的默认最大数量。对于 master.cf 文件中的特定服务，可以禁止这个限制。默认值为 100。
- queue_minfree - 接收邮件所需的队列文件系统中的最小可用空间量（以字节为单位）。Postfix SMTP 服务器目前使用此选项来确定它将接受任何邮件。默认情况下，当可用空间量小于 message_size_limit 的 1.5 倍时，Postfix SMTP 服务器会拒绝 MAIL FROM 命令。要指定较高的最小可用空间限制，请指定 message_size_limit 至少 1.5 倍的 queue_minfree 值。默认情况下，queue_minfree 值为 0。
- header_size_limit - 存储消息标头的最大内存量（以字节为单位）。如果标头更大，则丢弃过量。默认值为 102400。
-

message_size_limit - 消息的最大大小（以字节为单位），包括信封信息。默认值为 10240000。

4.3.10.2. NFS 和 Postfix

切勿将邮件假脱机目录 /var/spool/postfix/ 放置到 NFS 共享卷上。由于 NFSv2 和 NFSv3 不维护对用户和组 ID 的控制，因此两个或多个用户可以具有相同的 UID，并且接收和读取彼此的邮件。

注意

使用 Kerberos 的 NFSv4 时，情况并非如此，因为 SECRPC_GSS 内核模块不使用基于 UID 的身份验证。但是，最好不要将邮件假脱机 目录放在 NFS 共享卷上。

4.3.10.3. 仅邮件用户

为了帮助防止 Postfix 服务器上的本地用户利用，邮件用户最好使用电子邮件程序访问 Postfix 服务器。邮件服务器上的 shell 帐户不应被允许，并且 /etc/passwd 文件中的所有用户 shell 都应设置为 /sbin/nologin (root 用户可能例外)。

4.3.10.4. 禁用 Postfix 网络列表

默认情况下，Postfix 设置为仅侦听本地回送地址。您可以通过查看文件 /etc/postfix/main.cf 来验证这一点。

查看文件 /etc/postfix/main.cf，以确保仅显示以下 inet_interfaces 行：

```
inet_interfaces = localhost
```

这样可确保 Postfix 仅接受来自本地系统而不是来自网络的邮件（如 cron 作业报告）。这是默认设置，保护 Postfix 免受网络攻击。

要删除 localhost 限制并允许 Postfix 侦听所有接口，可使用 inet_interfaces = all 设置。

4.3.10.5. 将 Postfix 配置为使用 SASL

Postfix 的 Red Hat Enterprise Linux 7 版本可以使用 Dovecot 或 Cyrus SASL 实现进行 SMTP 身份验证 （或 SMTP AUTH）。SMTP 身份验证是 简单邮件传输协议的扩展。启用后，需要 SMTP 客户端使用服务器和客户端都支持并接受的身份验证方法向 SMTP 服务器进行身份验证。这部分论述了如何配置 Postfix 以使用 Dovecot SASL 实现。

要安装 Dovecot POP/IMAP 服务器，因此在您的系统中提供 Dovecot SASL 实现，以 root 用户身份运行以下命令：

```
~]# yum install dovecot
```

Postfix SMTP 服务器可以使用 UNIX-domain 套接字或 TCP 套接字 与 Dovecot SASL 实现通信。只有 Postfix 和 Dovecot 应用程序运行在单独的计算机上时，才需要后一种方法。本指南优先选择 UNIX 域套接字方法，其负担更好隐私。

为了指示 Postfix 使用 Dovecot SASL 实现，需要为这两个应用程序执行多个配置更改。按照以下步骤使这些更改生效。

设置 Dovecot

1. 修改主 Dovecot 配置文件 /etc/dovecot/conf.d/10-master.conf，使其包含以下行（已包含大多数相关部分，且只需要取消注释的行）：

```
service auth {
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
}
```

上面的示例假定使用 UNIX-domain socket 进行 Postfix 和 Dovecot 之间的通信。它还假定 Postfix SMTP 服务器的默认设置，其中包括位于 /var/spool/postfix/ 目录中的邮件队列，以及在 postfix 用户和组下运行的应用程序。这样，读取和写入权限仅限于 postfix 用户和组。

或者，您可以使用以下配置设置 Dovecot 以通过 TCP 侦听 Postfix 验证请求：

```
service auth {
  inet_listener {
    port = 12345
  }
}
```

在上例中，将 12345 替换为您要使用的端口数。

2. 编辑 /etc/dovecot/conf.d/10-auth.conf 配置文件，以指示 Dovecot 为 Postfix SMTP 服务器提供 普通 和 登录 身份验证机制：

```
auth_mechanisms = plain login
```

设置 Postfix

如果是 Postfix，则仅需要修改主配置文件 /etc/postfix/main.cf。添加或编辑以下配置指令：

1. 在 Postfix SMTP 服务器中启用 SMTP 身份验证：

```
smtpd_sasl_auth_enable = yes
```

2. 指示 Postfix 将 Dovecot SASL 实现用于 SMTP 身份验证：

```
smtpd_sasl_type = dovecot
```

3. 提供相对于 Postfix 队列目录的身份验证路径（请注意，无论 Postfix 服务器是否在 chroot 中运行，使用相对路径可确保配置可以正常工作）：

```
smtpd_sasl_path = private/auth
```

此步骤假设您要使用 UNIX-domain socket 进行 Postfix 和 Dovecot 之间的通信。如果您使用 TCP 套接字进行通信，要将 Postfix 配置为在不同机器上查找 Dovecot，请使用类似如下的配置值：

```
smtpd_sasl_path = inet:127.0.0.1:12345
```

在上例中，127.0.0.1 需要替换为 Dovecot 机器的 IP 地址，并使用 Dovecot 的/etc/dovecot/conf.d/10-master.conf 配置文件中指定的端口替换 12345。

4. 指定 Postfix SMTP 服务器为客户端提供的 SASL 机制。请注意，可以为加密和未加密的会话指定不同的机制。

```
smtpd_sasl_security_options = noanonymous, noplaintext
smtpd_sasl_tls_security_options = noanonymous
```

上面的例子指定，在未加密的会话中，不允许匿名身份验证，且不会允许传输未加密的用户名或密码的机制。对于加密的会话（使用 TLS），只允许非匿名身份验证机制。

有关限制允许 SASL 机制的所有支持策略列表，请参阅 http://www.postfix.org/SASL_README.html#smtpd_sasl_security_options。

其它资源

以下在线资源提供了用于通过 SASL 配置 Postfix SMTP 身份验证的附加信息。

- http://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL - 包含有关如何设置 Postfix 以对 SMTP 身份验证使用 Dovecot SASL 实现的信息。
- http://www.postfix.org/SASL_README.html#server_sasl - 包含有关如何将 Postfix 设置为对 SMTP 身份验证使用 Dovecot 或 Cyrus SASL 实现的信息。

4.3.11. 保护 SSH

Secure Shell (SSH)是一种强大的网络协议，用于通过安全通道与其他系统通信。通过 SSH 的传输会被加密和保护。有关 SSH 协议的常规信息以及 Red Hat Enterprise Linux 7 中的 SSH 服务的信息，请参阅 Red Hat Enterprise Linux 7 指南中的 OpenSSH 章节。

重要

本节注意保护 SSH 设置的最常见方法。否 意味着这个建议测量列表被视为详细或确定。有关修改 sshd 守护进程行为和 ssh (1) 的所有配置指令的说明，请参阅 sshd_config (5)。

4.3.11.1. 加密登录

SSH 支持使用加密密钥登录到计算机。这比仅使用密码更安全。如果您将此方法与其他身份验证方法相结合，则它被视为多因素身份验证。有关使用多个验证方法的详情，请参考 第 4.3.11.2 节 “多个身份验证方法”。

要启用使用加密密钥进行身份验证，需要将 /etc/ssh/sshd_config 文件中的 PubkeyAuthentication 配置指令设置为 yes。请注意，这是默认设置。将 PasswordAuthentication 指令设置为 no，以禁用可

能使用密码登录。

可以使用 ssh-keygen 命令生成 SSH 密钥。如果在没有附加参数的情况下调用，它会创建一个 2048 位 RSA 密钥集。默认情况下，密钥存储在 ~/.ssh/ 目录中。您可以使用 -b 参数修改密钥的位级。使用 2048 位密钥通常就足够了。Red Hat Enterprise Linux 7 系统管理员指南中的 配置 OpenSSH 章节包含有关生成密钥对的详细信息。

您应该在 ~/.ssh/ 目录中看到两个密钥。如果您在运行 ssh-keygen 命令时接受了默认值，则生成的文件分别命名为 id_rsa 和 id_rsa.pub，并分别包含私钥和公钥。您应该始终通过使私钥对除文件所有者以外的任何人都不可读来保护私钥。但是，公钥需要传送到您要登录的系统。您可以使用 ssh-copy-id 命令将密钥传送到服务器：

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

此命令还会自动将公钥附加到 服务器 上的 ~/.ssh/authorized_keys 文件中。当您尝试登录到服务器时，sshd 守护进程将检查此文件。

与密码和任何其他身份验证机制类似，您应该定期更改 SSH 密钥。完成后，请确保从 authorized_keys 文件中删除任何未使用的密钥。

4.3.11.2. 多个身份验证方法

使用多个身份验证方法或多因素验证会增加未授权访问的保护级别，因此在强化系统时应考虑防止它受到攻击。尝试登录到使用多因素身份验证的系统的用户必须成功完成所有指定的身份验证方法才能授予访问权限。

使用 /etc/ssh/sshd_config 文件中的 AuthenticationMethods 配置指令来指定要使用哪些身份验证方法。请注意，可以使用这个指令定义多个所需的身份验证方法列表。如果是这种情况，用户必须至少以其中一个列表完成每个方法。列表需要用空白空格分开，列表中的独立 authentication-method 名称必须用逗号分开。例如：

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

使用上述 AuthenticationMethods 指令配置的 sshd 守护进程仅在尝试成功完成 公钥身份验证 时授予访问权限，后跟 gssapi-with-mic 或 键盘交互 身份验证。请注意，每个请求的身份验证方法都需要使用 /etc/ssh/sshd_config 文件中对应的配置指令（如 PubkeyAuthentication）显式启用。有关可用身份验证方法的常规列表，请参阅 ssh (1) 的 AUTHENTICATION 部分。

4.3.11.3. 保护 SSH 的其他方法

协议版本

尽管 Red Hat Enterprise Linux 7 提供的 SSH 协议的实现仍然支持 SSH 客户端的 SSH-1 和 SSH-2 版本，但尽可能使用后者。SSH-2 版本包含有关较旧的 SSH-1 的改进，大多数高级配置选项仅在使用 SSH-2 时可用。

红帽建议使用 SSH-2 来最大化 SSH 协议可保护使用它的身份验证和通信的扩展。可使用 /etc/ssh/sshd _config 文件中的 Protocol 配置指令来指定 sshd 守护进程支持的协议版本或版本。默认设置为 2。请注意，SSH-2 版本是 Red Hat Enterprise Linux 7 SSH 服务器唯一支持的版本。

密钥类型

虽然 ssh-keygen 命令默认生成一对 SSH-2 RSA 密钥，但使用 -t 选项，也可以指示生成 DSA 或 ECDSA 密钥。ECDSA (Elliptic Curve Digital Signature Algorithm)以相同的对称密钥长度提供更好的性能。它还会生成较短的密钥。

非默认端口

默认情况下，sshd 守护进程侦听 TCP 端口 22。更改端口可降低系统因自动网络扫描而受到攻击的风险，从而通过模糊的方式提高安全性。可使用 /etc/ssh/sshd_config 配置文件中的 Port 指令来指定端口。另请注意，默认 SELinux 策略必须更改为允许使用非默认端口。您可以以 root 用户身份输入以下命令来修改 ssh_port_t SELinux 类型来完成此操作：

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

在上面的命令中，将 port_number 替换为使用 Port 指令指定的新端口号。

没有根登录

只要您的特定用例不需要以 root 用户身份登录，您应该考虑在 /etc/ssh/sshd_config 文件中将 PermitRootLogin 配置指令设置为 no。通过禁止以 root 用户身份登录，管理员可以审核哪些用户在以普通用户身份登录后运行了哪些特权命令，然后获得 root 权限。

使用 X 安全扩展

Red Hat Enterprise Linux 7 客户端中的 X 服务器不提供 X 安全扩展。因此，当连接到带有 X11 转发的不可信 SSH 服务器时，客户端无法请求另一个安全层。大多数应用程序都无法在启用此扩展的情况下运行。默认情况下，/etc/ssh/ssh_config 文件中的 ForwardX11Trusted 选项被设置为 yes，ssh -X remote_machine （不受信任的主机）和 ssh -Y remote_machine （可信主机）命令之间没有区别。

警告

红帽建议在连接到不可信主机时使用 X11 转发。

4.3.12. 保护 PostgreSQL

PostgreSQL 是一个对象相关数据库管理系统(DBMS)。在 Red Hat Enterprise Linux 7 中，postgresql-server 软件包提供 PostgreSQL。如果没有安装，请以 root 用户身份输入以下命令来安装它：

```
~]# yum install postgresql-server
```

在开始使用 PostgreSQL 之前，您必须在磁盘上初始化数据库存储区域。这称为数据库集群。要初始化数据库集群，请使用命令 initdb，该集群随 PostgreSQL 一起安装。数据库集群所需的文件系统位置由 -D 选项表示。例如：

```
~]$ initdb -D /home/postgresql/db1
```

如果尚未存在，initdb 命令将尝试创建您指定的目录。在这个示例中，我们使用名称 /home/postgresql/db1。/home/postgresql/db1 目录包含数据库中存储的所有数据，以及客户端身份验证配置文件：

```
~]$ cat pg_hba.conf
# PostgreSQL Client Authentication Configuration File
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access.  Records take one of these forms:
#
# local      DATABASE  USER  METHOD  [OPTIONS]
# host       DATABASE  USER  ADDRESS  METHOD  [OPTIONS]
# hostssl    DATABASE  USER  ADDRESS  METHOD  [OPTIONS]
# hostnossl  DATABASE  USER  ADDRESS  METHOD  [OPTIONS]
```

pg_hba.conf 文件中的以下行允许任何经过身份验证的用户使用其用户名访问任何数据库：

```
local   all             all                                     trust
```

当使用创建数据库用户和没有本地用户的层次应用程序时，这可能会造成问题。如果您不想显式控制

系统中的所有用户名，请从 pg_hba.conf 文件中删除这一行。

4.3.13. 保护 Docker

Docker 是一个开源项目，可在 Linux 容器内自动部署应用程序，并提供将应用与其运行时依赖项打包到容器中的功能。要使 Docker 工作流更加安全，请按照 Red Hat Enterprise Linux Atomic Host 7 容器安全指南中的步骤操作。

4.3.14. 针对 DDoS Attacks 保护 memcached

Memcached 是一个开源、高性能的分布式内存对象缓存系统。虽然它是通用的，但主要用于通过降低数据库负载来提高动态 Web 应用程序的性能。

Memcached 是一个内存键值存储，用于任意数据（如字符串和对象）的小块，来自于数据库调用、API 调用或页面渲染的结果。Memcached 允许应用程序从系统的一部分获取内存，超过其需求，并使它能被应用程序需要的区域访问。

Memcached 漏洞

2018 年，发现了向公共互联网公开的 memcached 服务器漏洞 DDoS 扩展攻击。这些攻击利用了使用 UDP 协议进行传输的 memcached 通信。攻击非常有效，因为具有高比例比例 - 几百字节大小的请求会产生几兆字节甚至几百兆字节的响应。这个问题已被记录为 CVE-2018-1000115。

在大多数情况下，memcached 服务不需要向公共互联网公开。此类风险可能有自己的安全问题，允许远程攻击者泄漏或修改存储在 memcached 中的信息。

强化 memcached

要降低安全风险，请根据您的配置执行以下步骤：

- 在 LAN 中配置防火墙。如果您的 memcached 服务器应该只可从本地网络访问，请不要允许 memcached 使用的端口的外部流量。例如，从允许的端口列表中删除默认情况下 memcached 使用的端口 11211：

```
~]# firewall-cmd --remove-port=11211/udp
~]# firewall-cmd --runtime-to-permanent
```

有关允许特定 IP 范围使用端口 11211 的 firewalld 命令，请参阅 第 5.8 节 “使用区域管理流量取决于源”。

- 通过将 -U 0 -p 11211 值添加到 /etc/sysconfig/memcached 文件中的 OPTIONS 变量来禁用 UDP，除非您的客户端确实需要这个协议：

```
OPTIONS="-U 0 -p 11211"
```

- 如果您在与应用程序相同的机器上只使用单个 memcached 服务器，请设置 memcached 以仅侦听 localhost 流量。将 -l 127.0.0.1,::1 值添加到 /etc/sysconfig/memcached 中的 OPTIONS ：

```
OPTIONS="-l 127.0.0.1,::1"
```

- 如果可能更改身份验证，请启用 SASL （简单身份验证和安全层）身份验证：

1. 在 /etc/sasl2/memcached.conf 文件中修改或添加：

```
sasldb_path: /path.to/memcached.sasldb
```

2. 在 SASL 数据库中添加帐户：

```
~]# saslpasswd2 -a memcached -c cacheuser -f /path.to/memcached.sasldb
```

3. 确保 memcached 用户和组可以访问数据库。

```
~]# chown memcached:memcached /path.to/memcached.sasldb
```

4. 通过将 -S 值添加到 /etc/sysconfig/memcached，在 memcached 中启用 SASL 支持：

-

```
OPTIONS="-S"
```

5. 重启 memcached 服务器以应用更改。

6. 将 SASL 数据库中创建的用户名和密码添加到应用程序的 memcached 客户端配置中。

- 使用 stunnel 加密 memcached 客户端和服务器之间的通信。由于 memcached 不支持 TLS，因此临时解决方案是使用代理，如 stunnel，它在 memcached 协议之上提供 TLS。

  您可以将 stunnel 配置为使用 PSK (Pre Shared Keys)，甚至最好使用用户证书。在使用证书时，只有经过身份验证的用户可以访问您的 memcached 服务器，且您的流量会被加密。

  重要

  如果您使用隧道访问 memcached，请确保该服务只侦听 localhost 或防火墙会阻止网络访问 memcached 端口。

  请参阅 第 4.8 节 “使用 stunnel” 了解更多信息。

4.4. 保护网络访问

4.4.1. 使用 TCP wrapper 和 xinetd 保护服务

TCP 封装器比拒绝对服务的访问要多。本节介绍如何使用它们来发送连接横幅、来自特定主机的攻击警告，并增强日志记录功能。有关 TCP Wrapper 功能和控制语言的详情，请查看 hosts_options(5) man page。有关可用标记，请参阅 xinetd.conf(5) man page，它作为您可以应用到服务的选项。

4.4.1.1. TCP 包装器和连接标语

当用户连接到服务时，显示合适的横幅是让潜在攻击者知道系统管理员正在警觉的好方法。您还可以

控制系统向用户呈现哪些信息。要为服务实施 TCP wrapper 横幅，请使用 横幅 选项。

此示例为 vsftpd 实施横幅。首先，创建一个横幅文件。它可以是系统上的任何位置，但它必须与守护进程的名称相同。在本例中，该文件名为 /etc/banners/vsftpd，包含以下行：

```
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，如用户名和主机名，或者的用户名和 IP 地址，以便更方便连接。

要使此横幅显示在传入连接中，请在 /etc/hosts.allow 文件中添加以下行：

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

4.4.1.2. TCP 封装器和攻击警告

如果特定的主机或网络已被检测到对服务器进行攻击，可以使用 TCP wrapper 来警告管理员使用 generate 指令从该主机或网络发出后续攻击。

在本例中，假设已检测到来自 206.182.68.0/24 网络的 cracker 试图攻击服务器。将以下行放在 /etc/hosts.deny 文件中，以拒绝来自该网络的任何连接尝试，并将尝试记录到特殊文件：

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

%d 令牌提供攻击者试图访问的服务名称。

要允许连接并记录它，请将 generate 指令放在 /etc/hosts.allow 文件中。

注意

由于 generate 指令执行任何 shell 命令，因此最好创建一个特殊的脚本来通知管理员或在特定客户端尝试连接到服务器时执行一系列命令。

4.4.1.3. TCP wrapper 和 Enhanced Logging

如果某些类型的连接比其他连接更关注，则可使用 **severity** 选项为该服务提升日志级别。

在本例中，假设尝试连接到 **FTP** 服务器上的端口 **23 (Telnet** 端口**)**的任何人都是攻击者。要表示这一点，请在日志文件中放置 **emerg** 标志，而不是默认的标志、**info** 和拒绝连接。

要做到这一点，请在 **/etc/hosts.deny** 中添加以下行：

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

这使用默认的 **authpriv** 日志记录工具，但将默认值 **info** 的优先级提升为，这会将日志消息直接发布到控制台。

4.4.2. 验证正在列出哪些端口

关闭未使用的端口非常重要，以避免出现可能的攻击。对于处于侦听状态的意外端口，您应该调查可能的入侵签名。

使用 netstat 进行开放端口扫描

以 **root** 用户身份输入以下命令，以确定哪些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
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      1/systemd
tcp        0      0 192.168.124.1:53        0.0.0.0:*               LISTEN      1829/dnsmasq
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1176/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1177/cupsd
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1664/master
sctp             0.0.0.0:2500                                       LISTEN   20985/sctp_darn
udp        0      0 192.168.124.1:53        0.0.0.0:*                           1829/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*                           977/dhclient
...
```

使用 **netstat** 命令的 **-l** 选项仅显示侦听的服务器套接字：

```
~]# netstat -tlnw
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

```
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 192.168.124.1:53        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp6       0      0 :::111                  :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 ::1:25                  :::*                    LISTEN
raw6       0      0 :::58                   :::*                    7
```

使用 ss 进行开放端口扫描

或者，使用 ss 实用程序列出处于侦听状态的打开端口。它可以显示比 netstat 更多的 TCP 和状态信息。

```
~]# ss -tlw
etid State      Recv-Q Send-Q     Local Address:Port                 Peer Address:Port
udp   UNCONN     0      0                 :::ipv6-icmp                      :::*
tcp   LISTEN     0      128               *:sunrpc                         *:*
tcp   LISTEN     0      5       192.168.124.1:domain                        *:*
tcp   LISTEN     0      128               *:ssh                           *:*
tcp   LISTEN     0      128       127.0.0.1:ipp                            *:*
tcp   LISTEN     0      100       127.0.0.1:smtp                           *:*
tcp   LISTEN     0      128               :::sunrpc                       :::*
tcp   LISTEN     0      128               :::ssh                          :::*
tcp   LISTEN     0      128               ::1:ipp                         :::*
tcp   LISTEN     0      100               ::1:smtp                         :::*
```

```
~]# ss -plno -A tcp,udp,sctp
Netid State      Recv-Q Send-Q      Local Address:Port                 Peer Address:Port
udp   UNCONN     0      0         192.168.124.1:53                        *:*                users:
(("dnsmasq",pid=1829,fd=5))
udp   UNCONN     0      0          *%virbr0:67                            *:*                users:
(("dnsmasq",pid=1829,fd=3))
udp   UNCONN     0      0                *:68                            *:*                users:
(("dhclient",pid=977,fd=6))
...
tcp   LISTEN     0      5         192.168.124.1:53                        *:*                users:
(("dnsmasq",pid=1829,fd=6))
tcp   LISTEN     0      128              *:22                            *:*                users:
(("sshd",pid=1176,fd=3))
tcp   LISTEN     0      128       127.0.0.1:631                           *:*                users:
(("cupsd",pid=1177,fd=12))
tcp   LISTEN     0      100       127.0.0.1:25                            *:*                users:
(("master",pid=1664,fd=13))
...
sctp  LISTEN     0      5                *:2500                          *:*                users:
(("sctp_darn",pid=20985,fd=3))
```

UNCONN 状态显示 UDP 侦听模式的端口。

从外部系统对 ss 输出中显示的每个 IP 地址进行扫描（除 localhost 127.0.0.0 或 ::1 范围除外）。使用 -6 选项扫描 IPv6 地址。

然后，使用 nmap 工具从通过网络连接到第一个系统的另一个远程机器进行外部检查。这可用于验证 firewalld 中的规则。以下是确定为 TCP 连接侦听哪些端口的示例：

```
~]# nmap -sT -O 192.168.122.65
   Starting Nmap 6.40 ( http://nmap.org ) at 2017-03-27 09:30 CEST
   Nmap scan report for 192.168.122.65
   Host is up (0.00032s latency).
   Not shown: 998 closed ports
   PORT    STATE SERVICE
   22/tcp  open  ssh
   111/tcp open  rpcbind
   Device type: general purpose
   Running: Linux 3.X
   OS CPE: cpe:/o:linux:linux_kernel:3
   OS details: Linux 3.7 - 3.9
   Network Distance: 0 hops

   OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
   Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
```

当 TCP SYN 扫描 (-sS) 不是选项时，TCP 连接扫描 (-sT) 是默认的 TCP 扫描类型。O 选项检测主机的操作系统。

使用 netstat 和 s 扫描 Open SCTP 端口

netstat 实用程序打印有关 Linux 网络子系统的信息。要显示开放流控制传输协议(SCTP)端口的协议统计信息，以 root 用户身份输入以下命令：

```
~]# netstat -plnS
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address   Foreign Address  State    PID/Program name
sctp              127.0.0.1:250                 LISTEN   4125/sctp_darn
sctp       0      0 127.0.0.1:260   127.0.0.1:250    CLOSE    4250/sctp_darn
sctp       0      0 127.0.0.1:250   127.0.0.1:260    LISTEN   4125/sctp_darn
```

```
~]# netstat -nl -A inet,inet6 | grep 2500
sctp              0.0.0.0:2500                          LISTEN
```

ss 工具也可以显示 SCTP 开放的端口：

```
~]# ss -an | grep 2500
sctp   LISTEN     0      5         *:2500                *:*
```

如需更多信息，请参阅 ss(8)、netstat(8)、nmap(1) 和 services(5) 手册页。

4.4.3. 禁用源路由

源路由是一种互联网协议机制，它允许 IP 数据包传输信息（地址列表），告知路由器数据包必须采用的路径。还有一个选项，可以在路由遍历时记录跃点。执行的跃点列表"路由记录"为目的地提供源的返回路径。这允许源（发送主机）指定路由，松散或严格，忽略部分或全部路由器的路由表。它可以允许用户重定向网络流量以进行恶意目的。因此，应该禁用基于源的路由。

accept_source_route 选项使网络接口接受设置了 Strict Source Routing (SSR)或 Loose Source Routing (LSR)选项的数据包。接收源路由数据包由 sysctl 设置控制。以 root 用户身份发出以下命令来丢弃设置了 SSR 或 LSR 选项的数据包：

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

禁用数据包转发也应该尽可能与上述操作进行（禁用转发可能会影响到虚拟化）。以 root 身份运行以下命令：

这些命令禁用所有接口上 IPv4 和 IPv6 数据包的转发：

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

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

这些命令禁用所有接口上所有多播数据包的转发：

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

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

接受 ICMP 重定向有一些合法用途。禁用接受并发送 ICMP 重定向数据包，除非特别需要。

这些命令禁用接受所有接口上的所有 ICMP 重定向数据包：

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

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

-

此命令禁用接受所有接口上的安全 ICMP 重定向数据包：

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

此命令禁用接受所有接口上的所有 IPv4 ICMP 重定向数据包：

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

重要

如果至少有一个 net.ipv4.conf.all.send_redirects 或 net.ipv4.conf.接口.send_redirects 选项被设置为 enabled，则发送 ICMP 重定向仍保持活动状态。确保将 net.ipv4.conf.接口.send_redirects 选项设为每个接口的 0 值。要在添加新接口时自动禁用 ICMP 请求发送，请输入以下命令：

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

只有 指令可以禁用发送 IPv4 重定向数据包。有关 “IPv6 节点要求” 的说明，请参阅 RFC4294，这会导致 IPv4 和 IPv6 之间的差别。

注意

要使这些设置在重启后持久保留，请修改 /etc/sysctl.conf 文件。例如，要禁用接受所有接口上所有 IPv4 ICMP 重定向数据包，请使用以 root 用户身份运行的编辑器打开 /etc/sysctl.conf 文件并添加以下行：

```
net.ipv4.conf.all.send_redirects=0
```

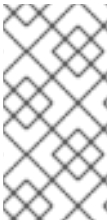
如需更多信息，请参阅 sysctl 手册页 sysctl (8)。有关基于源的路由及其变体的相关互联网选项的说明，请参阅 RFC791。

警告

以太网网络提供额外的重定向流量的方法，如 ARP 或 MAC 地址欺骗、未经授权的 DHCP 服务器和 IPv6 路由器或邻居公告。另外，单播流量偶尔会广播，从而导致信息泄漏。这些弱点只能由网络操作器实施的特定计数器解决。基于主机的计数器无效。

4.4.3.1. 反向路径转发

反向路径转发用于防止到达一个接口的数据包通过不同的接口离开。当传出路由和传入的路由不同时，它有时被称为非对称路由。路由器通常会以这种方式路由数据包，但大多数主机都不需要这样做。例外是，涉及通过一个链接发送流量并通过不同服务提供商的另一个链接接收流量的应用程序。例如，将租期行与 xDSL 或 satellite 链接与 3G 模式ms 结合使用。如果此类场景适用于您，则需要在传入接口上关闭反向路径转发。简而言之，除非您知道需要它，否则最好被启用，因为它会阻止用户使用本地子网的 IP 地址，并减少了 DDoS 攻击的机会。



注意

Red Hat Enterprise Linux 7 默认使用 Strict Reverse 路径转发，遵循 RFC 3704, Ingress Filtering for Multihomed Networks 的建议。

警告

如果启用了转发，则只有在有其他方法进行 source-address 验证（例如 iptables 规则）时，才应禁用反向路径转发。

rp_filter

通过 rp_filter 指令启用反向路径转发。sysctl 工具可用于更改正在运行的系统，并通过向 /etc/sysctl.conf 文件中添加行来进行永久更改。rp_filter 选项用于指示内核从三种模式之一中进行选择。

要进行临时的全局更改，请以 root 用户身份输入以下命令：

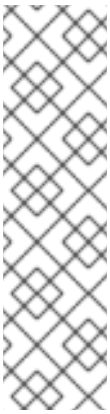
```
sysctl -w  net.ipv4.conf.default.rp_filter=integer
sysctl -w net.ipv4.conf.all.rp_filter=integer
```

其中 integer 是以下之一：

- 0 - 没有源验证。
- 1 - RFC 3704 中定义的严格模式。
- 2 - RFC 3704 中定义的松散模式。

可使用 net.ipv4.conf.接口.rp_filter 命令覆盖每个网络接口的设置，如下所示：

```
sysctl -w net.ipv4.conf.interface.rp_filter=integer
```



注意

要使这些设置在重启后持久保留，请修改 /etc/sysctl.conf 文件。例如，要更改所有接口的模式，请使用以 root 用户身份运行的编辑器打开 /etc/sysctl.conf 文件，并添加以下行：

```
net.ipv4.conf.all.rp_filter=2
```

IPv6_rpfilter

如果 IPv6 协议，firewalld 守护进程默认适用于 Reverse 路径转发。可以在 /etc/firewalld/firewalld.conf 文件中检查设置。您可以通过设置 IPv6_rpfilter 选项来更改 firewalld 行为。

如果您需要反向路径转发的自定义配置，您可以使用 ip6tables 命令 在不 firewalld 守护进程的情况下执行它：

```
ip6tables -t raw -I PREROUTING -m rpfilter --invert -j DROP
```

此规则应插入原始/PREROUTING 链的开头，以便它应用到所有流量，特别是在有状态匹配规则之前。有关 iptables 和 ip6tables 服务的详情请参考 第 5.13 节 “使用 iptables 设置和控制 IP 集”。

启用数据包转发

要启用来自系统外部的数据包转发到另一个外部主机，必须在内核中启用 IP 转发。以 root 身份登录，并将 /etc/sysctl.conf 文件中显示为 net.ipv4.ip_forward = 0 的行改为以下内容：

```
net.ipv4.ip_forward = 1
```

要载入 /etc/sysctl.conf 文件中的更改，请输入以下命令：

```
/sbin/sysctl -p
```

要检查是否已打开 IP 转发，以 root 身份运行以下命令：

```
/sbin/sysctl net.ipv4.ip_forward
```

如果上述命令返回 1，则启用 IP 转发。如果返回 0，您可以使用以下命令手动打开：

```
/sbin/sysctl -w net.ipv4.ip_forward=1
```

4.4.3.2. 其它资源

以下是解释 Reverse 路径转发的更多资源。

- 安装的文档

  /usr/share/doc/kernel-doc-version/Documentation/networking/ip-sysctl.txt - 此文件包含目录中可用文件和选项的完整列表。第一次访问内核文档前，以 root 用户身份输入以下命令：

  ```
  ~]# yum install kernel-doc
  ```

- 在线文档

有关 Multihomed Networks 的 Ingress Filtering 的说明，请参阅 RFC 3704。

4.5. 使用 DNSSEC 保护 DNS 流量

4.5.1. DNSSEC 简介

DNSSEC 是一组 域名系统安全扩展 (DNSSEC)，它允许 DNS 客户端验证并检查来自 DNS 名称服务器的响应的完整性，以验证其原始卷，并确定它们是否在传输中被篡改。

4.5.2. 了解 DNSSEC

对于通过互联网连接，现在有更多 Web 站点可以使用 HTTPS 安全地连接。但是，在连接到 HTTPS webserver 之前，必须执行 DNS 查找，除非您直接输入 IP 地址。这些 DNS 查找是不安全的完成的，受因为缺少身份验证的中间人攻击。换句话说，DNS 客户端不能确信来自给定 DNS 名称服务器的回复是真实的，且未被篡改。更重要的是，递归名称服务器无法确定它从其他名称服务器获得的记录是个例。DNS 协议没有为客户端提供了一种机制来确保它不受中间人攻击。引入 DNSSEC 以解决使用 DNS 解析域名时缺少身份验证和完整性检查。它没有解决保密性的问题。

发布 DNSSEC 信息涉及数字签名 DNS 资源记录，以及以这样一种方式分发公钥，从而使 DNS 解析器能够构建分层信任链。所有 DNS 资源记录的数字签名都会生成并添加到区域，作为数字签名资源记录 (RRSIG)。区域的公钥被添加为 DNSKEY 资源记录。要构建分层链，DNSKEY 的哈希值在父区域中发布，以 委派签名 (DS)资源记录。为便于验证非一致性，则使用 NextSECure (NSEC)和 NSEC3 资源记录。在 DNSSEC 签名区域中，每个资源记录集 (RRset)都有对应的 RRSIG 资源记录。请注意，用于委托到子区域(NS 和 glue 记录)的记录没有签名；这些记录会出现在子区域中，并在那里签名。

处理 DNSSEC 信息由配置了根区域公钥的解析器完成。使用这个密钥，解析器可以验证 root 区域中使用的签名。例如，root 区域已签署了 .com 的 DS 记录。root 区域还为 .com 名称服务器提供 NS 和 glue 记录。解析器遵循此委托并查询 .com 的 DNSKEY 记录，使用这些委派的名称服务器。获得的 DNSKEY 记录的哈希值应与 root 区域中的 DS 记录匹配。如果是，解析器将信任 .com 获取的 DNSKEY。在 .com 区域中，RSIG 记录由 .com DNSKEY 创建。对于 .com 中的委派，此过程的重复重复，如 redhat.com。使用此方法时，只需要配置一个 root 密钥来验证 DNS 解析器只需要配置一个 root 密钥，同时它在正常操作期间从全球收集多个 DNSKEY。如果加密检查失败，解析器会将 SERVFAIL 返回到应用程序。

DNSSEC 的设计方式对不支持 DNSSEC 的应用程序完全不可见。如果非 DNSSEC 应用程序查询 DNSSEC 功能解析器，它将在没有这些新的资源记录类型（如 RRSIG）的情况下收到回答。但是，DNSSEC 功能解析器仍将执行所有加密检查，如果检测到恶意 DNS 回答，仍会向应用程序返回 SERVFAIL 错误。DNSSEC 保护 DNS 服务器（权威和递归）之间数据的完整性，它不会在应用程序和解析器之间提供安全性。因此，务必要让应用程序为其解析器提供安全传输。完成的最简单方法是，在 localhost 上运行 DNSSEC 功能解析器，并在 /etc/resolv.conf 中使用 127.0.0.1。或者可以使用到远程 DNS 服务器的 VPN 连接。

了解 Hotspot 问题

Wi-Fi Hotspots 或 VPN 依赖 “DNS” 是：捕获门户倾向于劫持 DNS，以便将用户重定向到需要为其验证（或付费）进行 Wi-Fi 服务的页面。连接到 VPN 的用户通常需要使用 “内部” DNS 服务器来查找公司网络外不存在的资源。这要求软件进行额外的处理。例如，dnssec-trigger 可用于检测 Hotspot 是否劫持 DNS 查询，unbound 可以充当代理名称服务器来处理 DNSSEC 查询。

选择 DNSSEC Capable Recursive Resolver

要部署支持递归解析器的 DNSSEC，可以使用 BIND 或 unbound。两者都默认启用 DNSSEC，并使用 DNSSEC root 密钥进行配置。要在服务器上启用 DNSSEC，但其中一个操作都将在移动设备（如笔记本）上首选使用 unbound，因为它允许本地用户使用 dnssec-trigger 时动态重新配置 Hotspots 所需的 DNSSEC 覆盖，在使用 Libreswan 时，对于 VPN。unbound 守护进程进一步支持部署在 etc/unbound8:0:1::.d/ 目录中列出的 DNSSEC 异常，它们对服务器和移动设备都很有用。

4.5.3. 了解 Dnssec-trigger

在 /etc/resolv.conf 中安装和配置 unbound 后，所有来自应用程序的 DNS 查询都会被 unbound 处理。DNSSEC-trigger 仅在触发 unbound 解析器时重新配置 unbound 解析器。这主要适用于连接到不同 Wi-Fi 网络的 roaming 客户端机器，如笔记本电脑。此过程如下：

- *当通过 DHCP 获取新的 DNS 服务器时，NetworkManager “会触发” dnssec-trigger。*
- *然后 DNSSEC -trigger 对服务器执行多个测试，并确定它是否正确支持 DNSSEC。*
- *如果存在，则 dnssec-trigger 会重新配置 unbound，以使用该 DNS 服务器作为所有查询的转发器。*
- *如果测试失败，dnssec-trigger 将忽略新的 DNS 服务器，并尝试一些可用的回退方法。*
- *如果它确定有无限端口 53 (UDP 和 TCP)可用，它将告知 unbound 成为完整的递归 DNS 服务器，而无需使用任何转发器。*
- *如果这不可能，例如，因为防火墙阻止了端口 53，除了到达网络的 DNS 服务器本身外，它将尝试使用 DNS 到端口 80，或者 TLS 封装 DNS 到端口 443。在端口 80 和 443 上运行的服务器可以在 /etc/dnssec-trigger/dnssec-trigger.conf 中配置。默认配置文件中应提供了注释的示例。*
- *如果这些回退方法也失败，dnssec-trigger 将提供不安全的操作，这将会完全绕过*

DNSSEC，“或者只在缓存中”运行，它不会尝试新的 DNS 查询，但会回答它在缓存中已有的所有内容。

Wi-Fi Hotspots 越来越多地将用户重定向到登录页，然后向互联网授予访问权限。在上面概述的序列中，如果检测到重定向，系统会提示您询问是否需要登录才能访问互联网。dnssec-trigger 守护进程每 10 秒继续探测 DNSSEC 解析器。有关使用 dnssec-trigger 图形化工具的详情，请查看 第 4.5.8 节 “使用 Dnssec-trigger”。

4.5.4. VPN Supplied Domains 和 Name Servers

某些类型的 VPN 连接可以传递一个域和用于该域的名称服务器列表，作为 VPN 隧道设置的一部分。在 Red Hat Enterprise Linux 中，NetworkManager 支持它。这意味着 unbound、dnssec-trigger 和 NetworkManager 的组合可以正确支持 VPN 软件提供的域和名称服务器。VPN 隧道启动后，会为接收的域名的所有条目清除本地 unbound 缓存，以便从使用 VPN 访问的内部名称服务器获取对域名中的名称的查询。当 VPN 隧道终止时，不会再次清除 unbound 缓存，以确保对域的任何查询都将返回公共 IP 地址，而不是之前获取的专用 IP 地址。请参阅 第 4.5.11 节 “为连接分割域配置 DNSSEC 验证”。

4.5.5. 推荐的命名实践

红帽建议静态名称和临时名称与 DNS 中用于机器 的完全限定域名 (FQDN)匹配，如 host.example.com。

分配名称和编号(ICANN)的互联网公司有时会将之前未注册的顶级域（如 .yourcompany）添加到公共寄存器中。因此，红帽强烈建议您不要使用没有委托给您的域名，即使在专用网络上，这可能会导致根据网络配置的不同解析域名。因此，网络资源可能会不可用。使用未委托给您的域名也使得 DNSSEC 更难以部署和维护，因为域名冲突需要手动配置来启用 DNSSEC 验证。有关此问题的更多信息，请参阅有关域名冲突的 ICANN 常见问题解答。

4.5.6. 了解信任 Anchors

在分层加密系统中，信任锚 是被假定为可信的权威实体。例如，在 X.509 架构中，根证书是从中派生信任链的信任锚。信任锚必须事先拥有信任方，然后才能进行路径验证。

在 DNSSEC 上下文中，信任锚由与该名称关联的 DNS 名称和公钥（或公钥的哈希）组成。它表示为 base 64 编码密钥。它与一个证书类似，它是一种交换信息（包括公钥）的方法，可用于验证和验证 DNS 记录。RFC 4033 将信任定位符定义为 DNSKEY RR 的已配置 DNSKEY RR 或 DS RR 哈希。验证安全感知解析器使用此公钥或哈希作为起点，用于将身份验证链构建到签名的 DNS 响应中。通常，验证解析

器必须通过一些安全或可信的方法在 DNS 协议之外获取其信任定位符的初始值。存在信任定位符还意味着解析器应该预期信任锚指向的区域。

4.5.7. 安装 DNSSEC

4.5.7.1. 安装 unbound

要在机器上使用 DNSSEC 验证 DNS，需要安装 DNS 解析器 未绑定 （或 绑定 ）。只需要在移动设备上安装 dnssec-trigger。对于服务器，unbound 应该足够了，但可能需要本地域的转发配置，具体取决于服务器所在的位置(LAN 或 Internet)。DNSSEC-trigger 目前只会对全局公共 DNS 区域提供帮助。NetworkManager、dhclient 和 VPN 应用程序通常会自动收集域列表（以及名称服务器列表），但不能自动收集 dnssec-trigger 或 unbound。

要安装 unbound，请以 root 用户身份输入以下命令：

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

4.5.7.2. 检查 unbound 是否正在运行

要确定 unbound 守护进程是否正在运行，请输入以下命令：

```
~]$ 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
```

如果 unbound 服务没有运行，systemctl status 命令将会报告 unbound 作为 Active: inactive (dead)。

4.5.7.3. 启动 unbound

要为当前会话启动 unbound 守护进程，请以 root 用户身份输入以下命令：

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

运行 **systemctl enable** 命令，以确保每次系统引导时都启动 **unbound** ：

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

unbound 守护进程允许使用以下目录配置本地数据或覆盖：

- **/etc/unbound/conf.d** 目录用于为特定域名添加配置。这用于将对域名的查询重定向到特定的 **DNS** 服务器。这通常用于只存在于企业 **WAN** 中的子域。

- **/etc/unbound/keys.d** 目录用于为特定域名添加信任定位符。当仅限内部名称被 **DNSSEC** 签名时，这是必需的，但没有公开现有的 **DS** 记录来构建信任路径。另一个用例是使用与公司 **WAN** 外部公开名称不同的 **DNSKEY** 进行签名。

- **/etc/unbound/local.d** 目录用于添加特定的 **DNS** 数据作为本地覆盖。这可用于构建黑名单或创建手动覆盖。此数据将由 **unbound** 返回到客户端，但不会标记为 **DNSSEC** 签名。

NetworkManager 以及一些 **VPN** 软件可能会动态更改配置。这些配置目录包含注释掉的示例条目。详情请查看 **unbound.conf (5)** 手册页。

4.5.7.4. 安装 Dnssec-trigger

dnssec-trigger 应用作为守护进程运行，**dnssec-triggerd**。要安装 **dnssec-trigger**，请以 **root** 用户身份输入以下命令：

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

4.5.7.5. 检查 Dnssec-trigger 守护进程是否正在运行

要确定 **dnssec-triggerd** 是否正在运行，请输入以下命令：

```
~]$ 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
```

如果 dnssec-triggerd 守护进程没有运行，systemctl status 命令将报告为 Active: inactive (dead)。要为当前会话启动它，请以 root 用户身份输入以下命令：

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

运行 systemctl enable 命令，以确保 dnssec-triggerd 每次系统引导时启动：

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

4.5.8. 使用 Dnssec-trigger

dnssec-trigger 应用有一个 GNOME 面板实用程序，用于显示 DNSSEC 探测结果，以及按需执行 DNSSEC 探测请求。要启动该实用程序，请按 Super 键进入 Activities Overview，输入 DNSSEC，然后按 Enter。在屏幕底部的消息栏中添加了图标重新排序异常。按屏幕右下角的 round blue 通知图标显示它。右键单击 anchor 图标以显示弹出菜单。

在不正常操作 unbound 中，本地将 unbound 用作名称服务器，resolv.conf 则指向 127.0.0.1。当您单击 Hotspot Sign-On 面板中的 OK 时，这已更改。DNS 服务器从 NetworkManager 查询并放入 resolv.conf 中。现在，您可以在 Hotspot 的登录页面上进行身份验证。anchor 图标显示一个大的红色感叹号，警告您以不安全的方式进行 DNS 查询。经过身份验证后，dnssec-trigger 应该自动检测此模式并切回到安全模式，但在某些情况下，用户必须选择 Reprobe 来手动执行此操作。

DNSSEC-trigger 通常不需要任何用户交互。启动后，它在后台工作，如果遇到了一个问题，则通过弹出文本框中通知用户。它还告知 unbound 对 resolv.conf 文件的更改。

4.5.9. 使用带有 DNSSEC 的 dig

要查看 DNSSEC 是否正常工作，可以使用各种命令行工具。使用的最佳工具是 bind-utils 软件包中的 dig 命令。其他有用的工具可从 ldns 软件包和 unbound 软件包中的 unbound-host 深入了解。旧的 DNS 工具 nslookup 和 host 已被弃用，不应使用。

要使用 dig 发送请求 DNSSEC 数据的查询，选项 +dnssec 会添加到命令中，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. BB8VHWEklaKpaLprt3hq1GkjDROvkmjYTBxiGhuki/BJn3PolGyrftxR
HH0377l0Lsybj/uZv5hL4UwWd/lw6Gn8GPikqhztAkgMxddMQ2lARP6p
wbMOKbSUuV6NGUT1WWwpbi+LelFMqQcAq3Se66iyH0Jem7HtgPEUE1Zc 3ol=

;; 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
```

除了 A 记录外，还会返回包含 DNSSEC 签名的 RRSIG 记录，以及签名的时间和过期时间。unbound 服务器表示数据经过 DNSSEC 验证，方法是返回顶部的 flags: 部分中的 ad bit。

如果 DNSSEC 验证失败，则 dig 命令会返回 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]
```

要请求有关失败的更多信息，可以通过为 dig 命令指定 +cd 选项来禁用 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.
E572dLKMvYB4cgTRyAHlKKEvdOP7tockQb7hXFNZKVbfXbZJOlDREJrr
zCgAfJ2hykfY0yJHAlnuQvM0s6xOnNBSvc2xLlybJdfTaN6kSR0YFdYZ
n2NpPctn2kUBn5UR1BJRin3Gqy20LZlZx2KD7cZBtieMsU/lunyhCSc0 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 错误地错误地认为清单本身是不良的，但在本示例中，www.dnssec-tools.org 的人已强制使用这个 RRSIG 签名，但我们无法手动查看此输出来检测。这个错误将显示在 systemctl status unbound 的输出中，unbound 守护进程会将这些错误记录到 syslog 中，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)
```

4.5.10. 为 Dnssec-trigger 设置 Hotspot 检测基础架构

当连接到网络时，dnssec-trigger 会尝试检测 Hotspot。Hotspot 通常是一个设备，它会强制用户与网页进行交互，然后才能使用网络资源。检测是通过尝试下载带有已知内容的特定固定网页来完成的。如果存在 Hotspot，则收到的内容不会如预期一样。

要设置一个固定的网页，其中包含 dnssec-trigger 可以用来检测 Hotspot 的已知内容，如下所示：

1. 在某些计算机上设置 Web 服务器，可在 Internet 上公开访问。请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的 Web 服务器 章节。

2. 服务器运行后，使用已知内容发布静态页面。该页面不需要是有效的 HTML 页面。例如，您可以使用名为 hotspot.txt 的纯文本文件，该文件仅包含字符串 OK。假设您的服务器位于 example.com，并且您在 Web 服务器 document_root/static/ 子目录中发布您的 hotspot.txt 文件，那么静态 Web 页面的地址将是 example.com/static/hotspot.txt。请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的 Web 服务器 一章中的 DocumentRoot 指令。

3. 将以下行添加到 /etc/dnssec-trigger/dnssec-trigger.conf 文件中：

   ```
   url: "http://example.com/static/hotspot.txt OK"
   ```

   此命令添加使用 HTTP （端口 80）探测到的 URL。第一部分是将要解析的 URL 以及将要下载的页面。命令的第二部分是下载的网页应包含的文本字符串。

有关配置选项的详情，请查看 man page dnssec-trigger.conf (8)。

4.5.11. 为连接分割域配置 DNSSEC 验证

默认情况下，对于任何连接提供的每个域的 dnssec-trigger 会自动添加带有正确名称服务器的区域，但通过 NetworkManager 提供的 Wi-Fi 连接除外。默认情况下，添加到 unbound 的所有转发区域都是 DNSSEC 验证的。

可以更改验证转发区域的默认行为，以便在默认情况下，所有转发区域 都不会被 DNSSEC 验证。为此，请更改 dnssec-trigger 配置文件 /etc/dnssec.conf 中的 validate_connection_provided_zones 变量。以 root 用户身份，打开并编辑行，如下所示：

```
validate_connection_provided_zones=no
```

不会对任何现有转发区进行更改，但只适用于将来的转发区。因此，如果您要为当前提供的域禁用 DNSSEC，则需要重新连接。

4.5.11.1. 为 Wi-Fi Supplied 域配置 DNSSEC 验证

可以为 Wi-Fi 提供的区添加转发区域。为此，请更改 dnssec-trigger 配置文件 /etc/dnssec.conf 中的 add_wifi_provided_zones 变量。以 root 用户身份，打开并编辑行，如下所示：

```
add_wifi_provided_zones=yes
```

不会对任何现有转发区进行更改，但只适用于将来的转发区。因此，如果要为当前的 Wi-Fi 提供的域启用 DNSSEC，则需要重新连接（重新启动） Wi-Fi 连接。

警告

将 Wi-Fi 提供的域作为转发区启用到 未绑定 可能会导致安全影响，例如：

1. Wi-Fi 接入点可以有意通过 DHCP 为您提供域，其没有授权，并将所有 DNS 查询路由到其 DNS 服务器。

2. 如果您的转发区的 DNSSEC 验证 关闭了，Wi-Fi 提供的 DNS 服务器可能会欺骗来自提供的域的域名的 IP 地址，而无需知道它。

4.5.12. 其它资源

以下是解释 DNSSEC 的更多资源。

4.5.12.1. 安装的文档

- DNSSEC-trigger (8) 手册页 - 描述 dnssec-triggerd、dnssec-trigger-control 和 dnssec-trigger-panel 的命令选项。

- *DNSSEC-trigger.conf (8) 手册页 - 描述 dnssec-triggerd 的配置选项。*
- *unbound (8) 手册页 - 描述 unbound 的命令选项(DNS 验证解析器)。*
- *unbound.conf (5) 手册页 - 包含如何配置 unbound 的信息。*
- *resolv.conf (5) 手册页 - 包含解析器例程读取的信息。*

4.5.12.2. 在线文档

http://tools.ietf.org/html/rfc4033

RFC 4033 DNS 安全简介和要求.

http://www.dnssec.net/

包含指向许多 DNSSEC 资源的网站。

http://www.dnssec-deployment.org/

DNSSEC 部署计划由部为家庭安全赞助，包含很多 DNSSEC 信息，并有用于讨论 DNSSEC 部署问题的邮件列表。

http://www.internetsociety.org/deploy360/dnssec/community/

互联网 Society 的 “Deploy 360” 计划与协调 DNSSEC 部署是全球寻找社区和 DNSSEC 活动的良好资源。

http://www.unbound.net/

本文档包含有关 unbound DNS 服务的通用信息。

http://www.nlnetlabs.nl/projects/dnssec-trigger/

本文档包含有关 dnssec-trigger 的一般信息。

4.6. 使用 LIBRESWAN 保护虚拟专用网络(VPN)

在 Red Hat Enterprise Linux 7 中，可以使用 Libreswan 应用程序支持的 IPsec 协议配置 虚拟专用网络 (VPN)。Libreswan 是 Openswan 应用程序的延续，Openswan 文档中的许多示例可以通过 Libreswan 交换。NetworkManager IPsec 插件称为 NetworkManager-libreswan。GNOME Shell 的用户应该安装 NetworkManager-libreswan-gnome 软件包，该软件包具有 NetworkManager-libreswan 作为依赖项。请注意，NetworkManager-libreswan-gnome 软件包只包括在 Optional 频道中。请参阅启用补充和可选存储库。

VPN 的 IPsec 协议本身使用 互联网密钥交换 (IKE)协议进行配置。术语 IPsec 和 IKE 可互换使用。IPsec VPN 也称为 IKE VPN、IKEv2 VPN、XAUTH VPN、Cisco VPN 或 IKE/IPsec VPN。IPsec VPN 变体，它使用 Level 2 Tunneling Protocol (L2TP)，它通常被称为 L2TP/IPsec VPN，它需要 Optional 频道 xl2tpd 应用程序。

Libreswan 是 Red Hat Enterprise Linux 7 中提供的开源用户空间 IKE 实现。IKE 版本 1 和 2 作为用户级别的守护进程实现。IKE 协议本身也加密。IPsec 协议由 Linux 内核实现，Libreswan 配置内核以添加和删除 VPN 隧道配置。

IKE 协议使用 UDP 端口 500 和 4500。IPsec 协议由两个不同的协议组成，即 Encapsulated Security Payload (ESP)，其协议号为 50，以及协议号 51 的 Authenticated Header (AH)。不建议使用 AH 协议。建议 AH 用户迁移到使用 null 加密的 ESP。

IPsec 协议有两种不同的操作模式，即 Tunnel 模式 （默认）和 传输模式。可以使用没有 IKE 的 IPsec 配置内核。这称为 手动密钥。可以使用 ip xfrm 命令手动配置密钥，但为了安全起见，强烈建议您这样做。Libreswan 使用 netlink 与 Linux 内核连接。在 Linux 内核中进行数据包加密和解密。

Libreswan 使用 网络安全服务 (NSS)加密库。libreswan 和 NSS 均通过了 联邦信息处理标准(FIPS) 出版物 140-2 的认证。

重要

由 Libreswan 和 Linux 内核实现的 IKE/IPsec VPN 是 Red Hat Enterprise Linux 7 中推荐的唯一 VPN 技术。在不了解这样做风险的情况下不要使用任何其他 VPN 技术。

4.6.1. 安装 Libreswan

要安装 Libreswan，请以 root 用户身份输入以下命令：

```
~]# yum install libreswan
```

检查是否安装了 Libreswan ：

```
~]$ yum info libreswan
```

在一个新的 Libreswan 安装后，应初始化 NSS 数据库作为安装过程的一部分。在启动新的数据库前，请按如下所示删除旧数据库：

```
~]# systemctl stop ipsec
~]# rm /etc/ipsec.d/*db
```

然后，要初始化新的 NSS 数据库，请以 root 用户身份输入以下命令：

```
~]# ipsec initnss
Initializing NSS database
```

仅在 FIPS 模式下运行时，需要使用密码保护 NSS 数据库。要为 FIPS 模式初始化数据库，而不是上一个命令，请使用：

```
~]# certutil -N -d sql:/etc/ipsec.d
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:
```

要启动 Libreswan 提供的 ipsec 守护进程，以 root 身份运行以下命令：

```
~]# systemctl start ipsec
```

确认守护进程现在正在运行：

```
~]$ systemctl status ipsec
* ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled; vendor preset: disabled)
```

```
   Active: active (running) since Sun 2018-03-18 18:44:43 EDT; 3s ago
     Docs: man:ipsec(8)
           man:pluto(8)
           man:ipsec.conf(5)
  Process: 20358 ExecStopPost=/usr/sbin/ipsec --stopnflog (code=exited, status=0/SUCCESS)
  Process: 20355 ExecStopPost=/sbin/ip xfrm state flush (code=exited, status=0/SUCCESS)
  Process: 20352 ExecStopPost=/sbin/ip xfrm policy flush (code=exited, status=0/SUCCESS)
  Process: 20347 ExecStop=/usr/libexec/ipsec/whack --shutdown (code=exited, status=0/SUCCESS)
  Process: 20634 ExecStartPre=/usr/sbin/ipsec --checknflog (code=exited, status=0/SUCCESS)
  Process: 20631 ExecStartPre=/usr/sbin/ipsec --checknss (code=exited, status=0/SUCCESS)
  Process: 20369 ExecStartPre=/usr/libexec/ipsec/_stackmanager start (code=exited,
status=0/SUCCESS)
  Process: 20366 ExecStartPre=/usr/libexec/ipsec/addconn --config /etc/ipsec.conf --checkconfig
(code=exited, status=0/SUCCESS)
 Main PID: 20646 (pluto)
   Status: "Startup completed."
   CGroup: /system.slice/ipsec.service
           └─20646 /usr/libexec/ipsec/pluto --leak-detective --config /etc/ipsec.conf --nofork
```

要确保 Libreswan 在系统启动时启动，以 root 用户身份运行以下命令：

```
~]# systemctl enable ipsec
```

配置任何中间以及基于主机的防火墙以允许 ipsec 服务。有关防火墙以及允许特定服务通过的信息，请参阅 第 5 章 使用防火墙。libreswan 需要防火墙来允许以下数据包：

- **用于 互联网密钥交换 (IKE)协议的 UDP 端口 500 和 4500**
- **用于 封装安全负载 (ESP) IPsec 数据包的协议 50**
- **协议 51 用于 Authenticated Header (AH) IPsec 数据包(uncommon)**

我们提供了使用 Libreswan 设置 IPsec VPN 的三个示例。第一个示例是将两个主机连接在一起，以便它们可以安全地通信。第二个示例将两个站点连接在一起以组成一个网络。第三个示例支持远程用户，在此上下文中称为 road warriors。

4.6.2. 使用 Libreswan 创建 VPN 配置

Libreswan 不使用术语 “‘源和目的地’ 或 ‘“服务器和客户端”’”，因为 IKE/IPsec 是对等协议。而是使用 “左侧的” 术语 “来指代” 端点（主机）。这也允许大多数情况下在两个端点上使用相同的配置，尽管许多管理员选择始终为本地主机使用，并 “适合” 远程主机。“”

身份验证端点有四个常用的方法：

- 预共享密钥 (PSK)是最简单的验证方法。PSK 应该由随机字符组成，长度至少为 20 个字符。在 FIPS 模式中，PSK 需要根据所使用的完整性算法满足最低强度要求。建议您不要使用小于 64 个随机字符的 PSK。

- 原始 RSA 密钥通常用于静态主机到主机或子网到子网 IPsec 配置。主机使用其他的公共 RSA 密钥手动配置。当几十个或更多主机都需要相互设置 IPsec 隧道时，此方法无法很好地扩展。

- X.509 证书通常用于大型部署，其中有很多主机需要连接到一个通用的 IPsec 网关。中央证书颁发机构 (CA)用于为主机或用户签名 RSA 证书。此中央 CA 负责中继信任，包括单个主机或用户的撤销。

- NULL 身份验证用于在不进行身份验证的情况下获取网格加密。它可防止被动攻击，但不会防止主动攻击。但是，由于 IKEv2 允许非对称身份验证方法，因此 NULL 身份验证也可用于互联网扩展 Opportunistic IPsec，其中客户端验证服务器，但服务器不验证客户端。此模型与使用 TLS （也称为 https:// 网站）的安全网站类似。

除了这些身份验证方法外，还可以添加额外的身份验证，以防止量子计算机可能的攻击。这个额外的验证方法称为 Postquantum Preshared Keys （PPK。单个客户端或客户端组可以通过指定与带外配置的预共享密钥对应的PPKID 来使用它们自己的 PPK。请参阅 第 4.6.9 节 “对 Quantum 计算机使用保护”。

4.6.3. 使用 Libreswan 创建主机到主机 VPN

要将 Libreswan 配置为在称为“ ‘left’ ”和 “ ‘right’ ”的两个主机之间创建主机到主机的 IPsec VPN，请在两个主机（左和右侧 ）上以 root 身份输入以下命令来创建新的原始 RSA 密钥对：

```
~]# ipsec newhostkey --output /etc/ipsec.d/hostkey.secrets
Generated RSA key pair with CKAID 14936e48e756eb107fa1438e25a345b46d80433f was stored in
the NSS database
```

这会为主机生成 RSA 密钥对。生成 RSA 密钥的过程可能需要很长时间，特别是在具有低熵的虚拟机中。

要查看主机密钥，以便可以在配置中指定它，作为 root 用户添加到新主机密钥的主机上，使用 “newhostkey” 命令返回的 CKAID ： “”

```
~]# ipsec showhostkey --left --ckaid 14936e48e756eb107fa1438e25a345b46d80433f
 # rsakey AQPFKElpV

leftrsasigkey=0sAQPFKElpV2GdCF0Ux9Kqhcap53Kaa+uCgduoT2I3x6LkRK8N+GiVGkRH4Xg+WMrz
Rb94kDDD8m/BO/Md+A30u0NjDk724jWuUU215rnpwvbdAob8pxYc4ReSgjQ/DkqQvsemoeF4kimMU1
OBPNU7lBw4hTBFzu+iVUYMELwQSXpremLXHBNIamUbe5R1+ibgxO19l/PAbZwxyGX/ueBMBvSQ+H
0UqdGKbq7UgSEQTFa4/gqdYZDDzx55tpZk2Z3es+EWdURwJOgGiiiIFuBagasHFpeu9Teb1VzRyytny
NiJCBVhWVqsB4h6eaQ9RpAMmqBdBeNHfXwb6/hg+JIKJgjidXvGtgWBYNDpG40fEFh9USaFlSdiHO+
dmGyZQ74Rg9sWLtiVdlH1YEBUtQb8f8FVry9wSn6AZqPlpGgUdtkTYUCaaifsYH4hoIA0nku4Fy/Ugej8
9ZdrSN7Lt+igns4FysMmBOl9Wi9+LWnfl+dm4Nc6UNgLE8kZc+8vMJGkLi4SYjk2/MFYgqGX/COxSCPE
FUZFiNK7Wda0kWea/FqE1heem7rvKAPIiqMymjSmytZl9hhkCD16pCdgrO3fJXsfAUChYYSPyPQClkav
vBL/wNK9zlaOwssTaKTj4Xn90SrZaxTEjpqUeQ==
```

您需要此密钥来添加到两个主机上的配置文件中，如下所示。如果您忘记了 CKAID，您可以使用以下方法获取机器上所有主机密钥的列表：

```
~]# ipsec showhostkey --list
< 1 >  RSA keyid: AQPFKElpV ckaid: 14936e48e756eb107fa1438e25a345b46d80433f
```

密钥对的 secret 部分存储在 “NSS 数据库中”，该数据库位于 /etc/ipsec.d8:0:1::.db 中。

要使此主机到主机的隧道的配置文件，上面的行 leftrsasigkey= 和 rightrsasigkey= 将添加到位于 /etc/ipsec.d/ 目录中的自定义配置文件中。

以 root 用户身份运行的编辑器，以以下格式创建具有适当名称的文件：

/etc/ipsec.d/my_host-to-host.conf

按如下方式编辑该文件：

```
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
```

公钥也可以由其 CKAID 而不是其 RSAID 配置。在这种情况下，使用 “leftckaid=” 而不是 “leftrsasigkey=”

您可以在左侧和右侧主机上使用相同的配置文件。Libreswan 根据指定的 IP 地址或主机名自动检测它是否 “保留” 或 “右侧”。如果其中一个主机是移动主机，这表示之前不知道 IP 地址，那么在移动客户端上，移动客户端将使用 %defaultroute 作为其 IP 地址。这将自动获取动态 IP 地址。在接受来自传入移动主机的静态服务器主机上，使用 %any 指定其 IP 地址的移动主机。

确保 leftrsasigkey 值 “从左侧” 主机获取，并且从右侧主机获取 “ right rsasigkey” 值。使用 leftckaid 和 rightckaid 时也是如此。

重启 ipsec 以确保它读取新配置，如果配置为在引导时启动，确认隧道已建立：

```
~]# systemctl restart ipsec
```

使用 auto=start 选项时，应在几秒钟内建立 IPsec 隧道。您可以以 root 用户身份输入以下命令来手动加载和启动隧道：

```
~]# ipsec auto --add mytunnel
~]# ipsec auto --up mytunnel
```

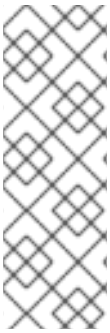
4.6.3.1. 使用 Libreswan 验证主机到主机 VPN

IKE 协商在 UDP 端口 500 和 4500 上发生。IPsec 数据包显示为 封装安全 Payload (ESP)数据包。ESP 协议没有端口。当 VPN 连接需要通过 NAT 路由器时，ESP 数据包会封装在端口 4500 的 UDP 数据包中。

要验证数据包是否通过 VPN 隧道发送，请以 root 用户身份以 root 用户身份运行以下命令：

```
~]# tcpdump -n -i interface esp or udp port 500 or 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 是已知传输流量的接口。要使用 tcpdump 结束捕获，请按 Ctrl+C。



注意

tcpdump 命令与 IPsec 意外交互。它只会看到传出的加密数据包，而不是传出的纯文本数据包。它确实会看到加密的传入数据包，以及解密的传入数据包。如果可能，在两台机器之间的路由器上运行 tcpdump，而不是在其中一个端点本身上运行。当使用虚拟 Tunnel 接口(VTI)时，物理接口的 tcpdump 会显示 ESP 数据包，而 VTI 接口上的 tcpdump 会显示明文流量。

要检查隧道已完全建立，另外查看通过隧道有多少流量，请以 root 用户身份输入以下命令：

```
~]# ipsec whack --trafficstatus
006 #2: "mytunnel", type=ESP, add_time=1234567890, inBytes=336, outBytes=336, id='@east'
```

4.6.4. 使用 Libreswan 配置站点到站点 VPN

为了让 Libreswan 创建站点到站点的 IPsec VPN，请将两个网络连接在一起，会在两个主机间创建一个 IPsec 隧道，该端点配置为允许来自一个或多个子网的流量来传递。因此，它们可以被视为网络远程部分的网关。站点到站点 VPN 的配置只能与主机到主机 VPN 不同，同时必须在配置文件中指定一个或多个网络或子网。

要将 Libreswan 配置为创建站点到站点的 IPsec VPN，首先配置主机到主机的 IPsec VPN，如 第 4.6.3 节 “使用 Libreswan 创建主机到主机 VPN” 所述，然后将文件复制到具有适当名称的文件，如 /etc/ipsec.d/my_site-to-site.conf。使用以 root 身份运行的编辑器，编辑自定义配置文件 /etc/ipsec.d/my_site-to-site.conf，如下所示：

```
conn mysubnet
    also=mytunnel
    leftsubnet=192.0.1.0/24
    rightsubnet=192.0.2.0/24
    auto=start

conn mysubnet6
    also=mytunnel
    connaddrfamily=ipv6
    leftsubnet=2001:db8:0:1::/64
    rightsubnet=2001:db8:0:2::/64
    auto=start

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
```

要启动隧道，重启 Libreswan 或手动加载并启动所有连接，以 root 用户身份启动所有连接：

```
~]# ipsec auto --add mysubnet
```

```
~]# ipsec auto --add mysubnet6
```

```
~]# 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}
```

4.6.4.1. 使用 Libreswan 验证站点到站点 VPN

验证数据包是否通过 VPN 隧道发送，与 第 4.6.3.1 节 “使用 Libreswan 验证主机到主机 VPN” 中所述的步骤相同。

4.6.5. 使用 Libreswan 配置站点到站点的单一 Tunnel VPN

通常，当构建站点到站点隧道时，网关需要使用其内部 IP 地址而不是其公共 IP 地址相互通信。这可以通过一个隧道来完成。如果左侧主机（主机名为 west ）具有内部 IP 地址 192.0.1.254 和正确的主机（主机名 east ）具有内部 IP 地址 192.0.2.254，请使用单一隧道将以下配置存储在两个服务器上的 /etc/ipsec.d/myvpn.conf 文件中：

```
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
```

4.6.6. 使用 Libreswan 配置子网扩展

IPsec 通常部署在 hub 和 spoke 架构中。每个叶节点都有一个 IP 范围，它是更大的范围。通过 hub 相互沟通。这称为 子网入侵。

例 4.2. 配置简单子网扩展设置

在以下示例中，我们使用 10.0.0.0/8 和两个使用更小 /24 子网的分支配置头办公室。

在头办公室：

```
conn branch1
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=5.6.7.8
    rightid=@branch1
    rightsubnet=10.0.1.0/24
    rightrsasigkey=0sAXXXX[...]
    #
    auto=start
    authby=rsasig

conn branch2
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=10.11.12.13
    rightid=@branch2
    rightsubnet=10.0.2.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasig
```

在“分支1”办公室中，我们使用相同的连接。另外，我们使用直通连接来排除通过隧道发送的本地 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
    rightsubnet=10.0.1.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasig

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
```

4.6.7. 配置 IKEv2 远程访问 VPN Libreswan

road warriors 是带动态分配 IP 地址（如笔记本电脑）的移动客户端的旅行用户。它们使用证书进行身份验证。为了避免需要使用旧的 IKEv1 XAUTH 协议，在以下示例中使用 IKEv2：

在服务器中：

```
conn roadwarriors
    ikev2=insist
    # Support (roaming) MOBIKE clients (RFC 4555)
    mobike=yes
    fragmentation=yes
    left=1.2.3.4
    # if access to the LAN is given, enable this, otherwise use 0.0.0.0/0
    # leftsubnet=10.10.0.0/16
    leftsubnet=0.0.0.0/0
    leftcert=vpn-server.example.com
    leftid=%fromcert
    leftxauthserver=yes
    leftmodecfgserver=yes
    right=%any
```

```
    # trust our own Certificate Agency
    rightca=%same
    # pick an IP address pool to assign to remote users
    # 100.64.0.0/16 prevents RFC1918 clashes when remote users are behind NAT
    rightaddresspool=100.64.13.100-100.64.13.254
    # if you want remote clients to use some local DNS zones and servers
    modecfgdns="1.2.3.4, 5.6.7.8"
    modecfgdomains="internal.company.com, corp"
    rightxauthclient=yes
    rightmodecfgclient=yes
    authby=rsasig
    # optionally, run the client X.509 ID through pam to allow/deny client
    # pam-authorize=yes
    # load connection, don't initiate
    auto=add
    # kill vanished roadwarriors
    dpddelay=1m
    dpdtimeout=5m
    dpdaction=%clear
```

其中：

left=1.2.3.4

1.2.3.4 值指定服务器的实际 IP 地址或主机名。

leftcert=vpn-server.example.com

这个选项**指定引用用于导入证书的友好名称或 nickname 的证书。通常，该名称作为 PKCSRG 证书捆绑包的一部分生成，格式为 .p12 文件。如需更多信息，请参阅 pkcs12 (1) 和 pk12util (1) 手册页。**

在移动客户端上，road warrior 的设备使用之前配置的稍微变化：

```
conn to-vpn-server
    ikev2=insist
    # pick up our dynamic IP
    left=%defaultroute
    leftsubnet=0.0.0.0/0
    leftcert=myname.example.com
    leftid=%fromcert
    leftmodecfgclient=yes
    # right can also be a DNS hostname
    right=1.2.3.4
    # if access to the remote LAN is required, enable this, otherwise use 0.0.0.0/0
    # rightsubnet=10.10.0.0/16
    rightsubnet=0.0.0.0/0
    # trust our own Certificate Agency
```

```
rightca=%same
authby=rsasig
# allow narrowing to the server's suggested assigned IP and remote subnet
narrowing=yes
# Support (roaming) MOBIKE clients (RFC 4555)
mobike=yes
# Initiate connection
auto=start
```

其中：

auto=start

这个选项允许用户在 ipsec 系统服务启动时连接到 VPN。如果要稍后建立连接，请使用 auto=add 替换它。

4.6.8. 使用 X.509 配置 IKEv1 远程访问 VPN Libreswan 和 XAUTH

Libreswan 提供了一种将 IP 地址和 DNS 信息原生分配给 roaming VPN 客户端的方法，因为使用 XAUTH IPsec 扩展建立连接。可以使用 PSK 或 X.509 证书部署扩展身份验证(XAUTH)。使用 X.509 部署更安全。客户端证书可以通过证书撤销列表或 在线证书状态协议 (OCSP)吊销。使用 X.509 证书时，单个客户端无法模拟服务器。使用 PSK （也称为组密码）是理论上可能的。

XAUTH 要求 VPN 客户端额外使用用户名和密码识别其自身。对于一次性密码(OTP)，如 Google Authenticator 或 RSA SecureID 令牌，一次性令牌会附加到用户密码中。

XAUTH 有三个可能的后端：

xauthby=pam

这使用 /etc/pam.d/pluto 中的配置来验证用户。可插拔验证模块(PAM)可以自行配置为使用各种后端。它可以使用系统帐户 user-password 方案、LDAP 目录、RADIUS 服务器或自定义密码身份验证模块。如需更多信息，请参阅使用可插拔验证模块(PAM) 章节。

xauthby=file

这使用 /etc/ipsec.d/passwd 配置文件（不应与 /etc/ipsec.d/nsspassword 文件混淆）。此文件的格式与 Apache .htpasswd 文件类似，可以使用 Apache htpasswd 命令在此文件中创建条目。但是，在用户名和密码后，使用使用的 IPsec 连接名称需要第三列，例如使用 conn remoteusers 来提供 VPN 来删除用户，密码文件条目应如下所示：

```
user1:$apr1$MIwQ3DHb$1I69LzTnZhnCT2DPQmAOK.:remoteusers
```

注意

在使用 **htpasswd** 命令时，必须在每行的 **user:password** 部分之后手动添加连接名称。

xauthby=alwaysok

服务器始终假定 **XAUTH** 用户和密码组合是正确的。客户端仍然需要指定用户名和密码，尽管服务器会忽略它们。只有在用户已经通过 **X.509** 证书标识时，或者测试 **VPN** 时，才应使用此选项，而无需 **XAUTH** 后端。

使用 **X.509** 证书的服务器配置示例：

```
conn xauth-rsa
    ikev2=never
    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
    modecfgdns="1.2.3.4,8.8.8.8"
    modecfgdomains=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
```

当将 **xauthfail** 设置为 **soft** 时，而不是 **hard**，则忽略身份验证失败，且 **VPN** 会像正确验证一样设置。自定义 **updown** 脚本可用于检查环境变量 **XAUTH_FAILED**。然后可以将此类用户重定向到一个

“walled garden”，例如，他们可以联系管理员或续订该服务的付费订阅。

VPN 客户端使用 modecfgdomain 值和 DNS 条目将对指定域的查询重定向到这些指定的名称服务器。这允许用户使用内部 DNS 名称访问内部资源。请注意，虽然 IKEv2 支持使用 modecfgdomains 和 modecfgdns 的域名和名称服务器 IP 地址的逗号分隔列表，但 IKEv1 协议只支持一个域名，libreswan 最多只支持两个名称服务器 IP 地址。另外，要将横幅文本发送到 VPN cliens，请使用 modecfgbanner 选项。

如果 leftsubnet 不是 0.0.0.0/0，则分割配置请求将自动发送到客户端。例如：在使用 leftsubnet=10.0.0.0/8 时，VPN 客户端只通过 VPN 为 10.0.0.0/8 发送流量。

在客户端上，用户必须输入用户密码，这取决于所使用的后端。例如：

xauthby=file

管理员生成密码并将其存储在 /etc/ipsec.d/passwd 文件中。

xauthby=pam

密码在 /etc/pam.d/pluto 文件的 PAM 配置中指定的位置获得。

xauthby=alwaysok

未检查密码，并且始终接受该密码。使用这个选项用于测试目的，或者要确保仅 xauth-only 客户端的兼容性。

其它资源

有关 XAUTH 的更多信息，请参阅 ISAKMP/Oakley (XAUTH) Internet-Draft 文档中的扩展身份验证。

4.6.9. 对 Quantum 计算机使用保护

使用带有预共享密钥的 IKEv1 可以防止量子攻击者。重新设计 IKEv2 不会原生提供这种保护。Libreswan 提供使用 Postquantum Preshared Keys (PPK)来保护 IKEv2 连接免受量子攻击。

要启用可选的 PPK 支持，请在连接定义中添加 ppk=yes。如需要 PPK，请添加 ppk=insist。然后，可为每个客户端分配一个带有一个 secret 值的 PPK ID，其 secret 值会被传递到带外（最好是使用半字

节安全）。PPK 的随机性应该非常强大，且不能基于字典的单词。PPK ID 和 PPK 数据本身存储在 ipsec.secrets 中，例如：

```
@west @east : PPKS "user1" "thestringismeanttobearandomstr"
```

PPKS 选项指的是静态 PPK。有一个实验性功能，可以使用基于一次性平板的 Dynamic PPK。在每个连接中，一次性平板的一个新部分用作 PPK。当使用时，文件中的该部分动态 PPK 被零覆盖，以防止重复使用。如果没有剩余一次性材料，连接会失败。详情请查看 ipsec.secrets(5) 手册页。

警告

动态 PPK 的实现是作为技术预览提供的，这个功能应该小心使用。如需更多信息，请参阅 Red Hat Enterprise Linux 7.5 发行注记。

4.6.10. 其它资源

以下信息源提供有关 Libreswan 和 ipsec 守护进程的其他资源。

4.6.10.1. 安装的文档

- IPsec (8) 手册页 - 描述 ipsec 的命令选项。
- IPsec.conf (5) 手册页 - 包含配置 ipsec 的信息。
- IPsec.secrets (5) 手册页 - 描述 ipsec.secrets 文件的格式。
- ipsec_ auto (8) 手册页 - 描述使用自动命令行客户端来处理使用自动交换密钥建立的 Libreswan IPsec 连接。
- ipsec_rsasigkey (8) 手册页 - 描述用于生成 RSA 签名密钥的工具。

- /usr/share/doc/libreswan-version/

4.6.10.2. 在线文档

https://libreswan.org

上游项目的网站。

https://libreswan.org/wiki

Libreswan Project Wiki。

https://libreswan.org/man/

所有 Libreswan man page。

NIST Special Publication 800-77: Guide to IPsec VPNs

在根据 IPsec 部署安全服务时为机构提供实际指导。

4.7. 使用 OPENSSL

OpenSSL 是一个为应用程序提供加密协议的库。openssl 命令行工具启用使用 shell 中的加密功能。它包含一个交互模式。

openssl 命令行工具有许多伪命令，用于提供有关系统上安装 openssl 版本的命令的信息。pseudo-commands list-standard-commands、list-message-digest-commands 和 list-cipher-commands 会分别输出所有标准命令、消息摘要命令或密码命令的列表，它们分别位于 present openssl 工具中。

pseudo-commands list-cipher-algorithms 和 list-message-digest-algorithms 列出所有密码和消息摘要名称。pseudo-command list-public-key-algorithms 列出所有支持的公钥算法。例如，要列出支持的公钥算法，请运行以下命令：

```
~]$ openssl list-public-key-algorithms
```

pseudo-command no-command-name 测试指定名称的 command-name 是否可用。旨在在 shell 脚本中使用。如需更多信息，请参阅 man openssl(1)。

4.7.1. 创建和管理加密密钥

使用 OpenSSL 时，公钥从对应的私钥衍生而来。因此，在决定算法后，第一步是生成私钥。在这些示例中，私钥称为 privkey.pem。例如，要使用默认参数创建 RSA 私钥，请运行以下命令：

```
~]$ openssl genpkey -algorithm RSA -out privkey.pem
```

RSA 算法支持以下选项：

- rsa_keygen_bits:numbits - 生成的密钥中的位数。如果没有指定 1024，则使用。
- rsa_keygen_pubexp:value - RSA public exponent 值。如果前面带有 0x，则可以是大的十进制值，也可以是十六进制值。默认值为 65537。

例如，要使用 3 作为公钥创建 2048 位 RSA 私钥，请运行以下命令：

```
~]$ openssl genpkey -algorithm RSA -out privkey.pem -pkeyopt rsa_keygen_bits:2048 \ -pkeyopt
rsa_keygen_pubexp:3
```

要使用 128 位 AES 和密码短语 "hello" 来加密私钥，请运行以下命令：

```
~]$ openssl genpkey -algorithm RSA -out privkey.pem -aes-128-cbc -pass pass:hello
```

有关生成私钥的更多信息，请参阅 man genpkey(1)。

4.7.2. 生成证书

要使用 OpenSSL 生成证书，需要一个有可用的私钥。在这些示例中，私钥称为 privkey.pem。如果您还没有生成私钥，请查看 第 4.7.1 节 "创建和管理加密密钥"

要使 证书由证书颁发机构 (CA)签名，需要生成一个证书，然后将其发送到 CA 进行签名。这称为证书签名请求。请参阅 第 4.7.2.1 节 "创建证书签名请求" 了解更多信息。另一种方法是创建一个自签名证书。请参阅 第 4.7.2.2 节 "创建自签名证书" 了解更多信息。

4.7.2.1. 创建证书签名请求

要创建提交给 CA 的证书，以以下格式发出命令：

```
~]$ openssl req -new -key privkey.pem -out cert.csr
```

这将以默认 的隐私增强型电子邮件 (PEM)格式创建经过编码的名为 cert.csr 的 X.509 证书。名称 PEM 派生自 RFC 1424 中描述的 “互联网 Electronic 邮件的隐私增强”。要以替代 DER 格式生成证书文件，请使用 -outform DER 命令选项。

在发出上述命令后，系统将提示您提供关于您和组织的信息，以便为证书创建 可分辨名称 (DN)。您需要以下信息：

- 您国家有两个字母国家/地区代码
- 您州的全名或省
- City 或 Town
- 您的机构名称
- 机构中单元的名称
- 您的系统名称或主机名
- 您的电子邮件地址

req(1) man page 描述了 PKCS# 10 证书请求和生成工具。创建流程中使用的默认设置包含在 /etc/pki/tls/openssl.cnf 文件中。如需更多信息，请参阅 man openssl.cnf (5)。

4.7.2.2. 创建自签名证书

要生成自签名证书，请对 366 天有效，以以下格式发出命令：

```
~]$ openssl req -new -x509 -key privkey.pem -out selfcert.pem -days 366
```

4.7.2.3. 使用 Makefile 创建证书

/etc/pki/tls/certs/ 目录包含一个 Makefile，可用于使用 make 命令创建证书。要查看使用说明，请运行以下命令：

```
~]$ make -f /etc/pki/tls/certs/Makefile
```

或者，切换到目录并发出 make 命令，如下所示：

```
~]$ cd /etc/pki/tls/certs/
~]$ make
```

详情请查看 make(1) man page。

4.7.3. 验证证书

CA 签名的证书称为可信证书。因此，自签名证书是不受信任的证书。verify 实用程序使用相同的 SSL 和 S/MIME 功能来验证 OpenSSL 在正常操作中使用的证书。如果发现错误，则会报告错误，然后尝试继续测试以报告任何其他错误。

要以 PEM 格式验证多个独立 X.509 证书，以以下格式发出命令：

```
~]$ openssl verify cert1.pem cert2.pem
```

要验证证书链，leaf 证书必须位于 cert.pem 中，您不信任的中间证书必须在 untrusted.pem 中直接连接。可信 root CA 证书必须在 /etc/pki/tls/certs/ca-bundle.crt 或 cacert.pem 文件中列出的默认 CA 中。然后，要验证链，以以下格式发出命令：

```
~]$ openssl verify -untrusted untrusted.pem -CAfile cacert.pem cert.pem
```

如需更多信息，请参阅 man verify(1)。

重要

由于这个算法的强度不足，Red Hat Enterprise Linux 7 中禁用了使用 MD5 哈希算法的签名验证。始终使用强大的算法，如 SHA256。

4.7.4. 加密和解密文件

要使用 OpenSSL 加密（和解密）文件，可以使用 pkeyutl 或 enc 内置命令。使用 pkeyutl 时，RSA 密钥用于执行加密和解密，而使用 enc，使用对称算法。

使用 RSA 密钥

要加密名为 纯文本 的文件，请按如下所示发出命令：

```
~]$ openssl pkeyutl -in plaintext -out cyphertext -inkey privkey.pem
```

密钥和证书的默认格式为 PEM。如果需要，使用 -keyform DER 选项指定 DER 密钥格式。

要指定加密引擎，请使用 -engine 选项，如下所示：

```
~]$ openssl pkeyutl -in plaintext -out cyphertext -inkey privkey.pem -engine id
```

其中 id 是加密引擎的 ID。要检查引擎的可用性，请运行以下命令：

```
~]$ openssl engine -t
```

要签名名为 plaintext 的数据文件，请运行以下命令：

```
~]$ openssl pkeyutl -sign -in plaintext -out sigtext -inkey privkey.pem
```

要验证签名数据文件并提取数据，请运行以下命令：

```
~]$ openssl pkeyutl -verifyrecover -in sig -inkey key.pem
```

要验证签名，例如使用 DSA 密钥，请按如下所示发出命令：

```
~]$ openssl pkeyutl -verify -in file -sigfile sig -inkey key.pem
```

pkeyutl(1) 手册页描述了公钥算法工具。

使用 Symmetric Algorithms

要列出可用的对称加密算法，请使用不支持的选项执行 enc 命令，如 -l ：

```
~]$ openssl enc -l
```

要指定算法，请使用其名称作为选项。例如，要使用 aes-128-cbc 算法，请使用以下语法：

```
openssl enc -aes-128-cbc
```

要使用 aes-128-cbc 算法加密名为 plaintext 的文件，请输入以下命令：

```
~]$ openssl enc -aes-128-cbc -in plaintext -out plaintext.aes-128-cbc
```

要解密上例中获取的文件，请使用 -d 选项，如下例所示：

```
~]$ openssl enc -aes-128-cbc -d -in plaintext.aes-128-cbc -out plaintext
```

重要

enc 命令无法正确支持 AEAD 密码，而 ecb 模式则被视为安全。为获得最佳结果，请不要使用 cbc 以外的其他模式、cfb、b 或 ctr。

4.7.5. 生成消息 Digests

dgst 命令以十六进制形式生成所提供的文件或文件的消息摘要。命令也可用于数字签名和验证。message digest 命令采用以下格式：

```
openssl dgst algorithm -out filename -sign private-key
```

其中 algorithm 是 md5|md4|md2|sha1|sha|mdc2|ripemd160|dss1 之一。编写本文时，首选使用 SHA1 算法。如果您需要使用 DSA 签名或进行验证，则必须将 dss1 选项与包含 -rand 选项指定的随机数据的文件一起使用。

要使用 sha1 算法以默认 Hex 格式生成消息摘要，请运行以下命令：

```
~]$ openssl dgst sha1 -out digest-file
```

要使用私钥 privekey.pem 对摘要进行数字签名，请运行以下命令：

```
~]$ openssl dgst sha1 -out digest-file -sign privkey.pem
```

如需更多信息，请参阅 man dgst(1)。

4.7.6. 生成密码散列

passwd 命令计算密码的哈希。要在命令行中计算密码的哈希值，请运行以下命令：

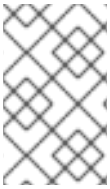
```
~]$ openssl passwd password
```

默认使用 -crypt 算法。

要从标准输入（使用基于 MD5 的 BSD 算法 1）计算密码的哈希值，请运行以下命令：

```
~]$ openssl passwd -1 password
```

-apr1 选项指定 BSD 算法的 Apache 变体。



注意

仅在禁用 FIPS 模式的 openssl passwd -1 password 命令。否则，命令不起作用。

要计算存储在文件中的密码的哈希值，并使用 salt xx，请运行以下命令：

```
~]$ openssl passwd -salt xx -in password-file
```

密码发送到标准输出，没有 -out 选项来指定输出文件。table 将生成包含其相应的明文密码的密码哈希表。

如需更多信息和示例，请参阅 man sslpasswd(1)。

4.7.7. 生成随机数据

要使用 seed 文件生成包含随机数据的文件，请运行以下命令：

```
~]$ openssl rand -out rand-file -rand seed-file
```

可以使用冒号：来指定用于查找随机数据进程的多个文件，作为列表分隔符。

如需更多信息，请参阅 man rand(1)。

4.7.8. 基准测试您的系统

要测试给定算法的系统的计算速度，以以下格式发出命令：

```
~]$ openssl speed algorithm
```

其中 algorithm 是您要使用的支持的算法之一。要列出可用的算法，请键入 openssl speed，然后按选项卡。

4.7.9. 配置 OpenSSL

OpenSSL 有一个配置文件 /etc/pki/tls/openssl.cnf，称为 master 配置文件，该文件由 OpenSSL 库读取。也可以为每个应用程序拥有单独的配置文件。配置文件包含多个部分，其部分名称如下：[section_name]。注意文件的第一个部分，直到第一个 [section_name] 被指代为 default 部分。当 OpenSSL 在配置文件中搜索名称时，首先搜索指定部分的名称。所有 OpenSSL 命令都使用 master OpenSSL 配置文件，除非命令中使用了一个选项来指定替代的配置文件。配置文件在 config (5) 手册页中详细介绍。

两个 RFC 解释了证书文件的内容。它们是：

- Internet X.509 公钥基础架构证书和证书撤销列表(CRL)配置文件
- 更新互联网 X.509 公钥基础架构证书和证书撤销列表(CRL)配置文件

4.8. 使用 STUNNEL

stunnel 程序是客户端和服务器之间的加密打包程序。它监听其配置文件中指定的端口，用客户端加密 communitation，并将数据转发到侦听其常用端口的原始守护进程。这样，您可以保护自身不支持任何类型的加密的服务，或者提高使用您要避免的加密类型的服务的安全性，如 SSL 版本 2 和 3，受 POODLE SSL 漏洞(CVE-2014-3566)的影响。详情请查看 https://access.redhat.com/solutions/1234773。CUPS 是一个组件示例，它没有提供在其自己的配置中禁用 SSL 的方法。

4.8.1. 安装 stunnel

以 root 身份输入以下命令安装 stunnel 软件包：

```
~]# yum install stunnel
```

4.8.2. 将 stunnel 配置为 TLS Wrapper

要配置 stunnel，请按照以下步骤执行：

1. 无论您使用哪个服务，您需要 stunnel 的有效证书。如果您没有合适的证书，您可以应用到证书颁发机构 来获取一个证书，或者您可以创建自签名证书。

警告

始终将由证书颁发机构签名的证书用于生产环境中运行的服务器。自签名证书仅适用于测试目的或专用网络。

有关证书颁发机构授予的证书的更多信息，请参阅 第 4.7.2.1 节 “创建证书签名请求”。另一方面，要为 stunnel 创建自签名证书，请输入 /etc/pki/tls/certs/ 目录，并以 root 用户身份运行以下命令：

```
certs]# make stunnel.pem
```

回答所有问题以完成该过程。

2. 当您有证书时，为 stunnel 创建配置文件。它是每行指定一个选项或服务定义开头的文本文件。您还可以在文件中保留注释和空行，以改进其法定性，其中注释以分号开头。

stunnel RPM 软件包包含 /etc/stunnel/ 目录，您可以在其中存储配置文件。虽然 stunnel 不需要文件名或其扩展名的任何特殊格式，请使用 /etc/stunnel/stunnel.conf。以下内容将 stunnel 配置为 TLS 包装器：

```
cert = /etc/pki/tls/certs/stunnel.pem
; Allow only TLS, thus avoiding SSL
sslVersion = TLSv1
chroot = /var/run/stunnel
setuid = nobody
setgid = nobody
pid = /stunnel.pid
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1

[service_name]
accept = port
connect = port
TIMEOUTclose = 0
```

或者，您可以通过使用以下行替换包含 sslVersion = TLSv1 的行来避免 SSL：

```
options = NO_SSLv2
options = NO_SSLv3
```

选项的目的如下：

- cert - 证书的路径
- sslVersion - SSL 的版本；请注意，您可以使用 TLS，即使 SSL 和 TLS 是两个独立的加密协议
- chroot - 更改在其中运行 stunnel 进程的根目录，以便提高安全性
- setuid, setgid - stunnel 进程的用户和组作为运行; nobody 是一个受限系统帐户
- pid - stunnel 文件保存其进程 ID，相对于 chroot

- socket - 本地和远程套接字选项；在这种情况下，禁用 Nagle 的算法 来提高网络延迟
- [service_name] - 服务定义的开头；此行下面使用的选项仅适用于给定服务，而上述选项则全局影响 stunnel
- accept - 要侦听的端口
- connect - 要连接到的端口；这必须是您要保护的服务的端口
- TIMEOUTclose - 从客户端等待 close_notify 警报的秒数； 0 指示 stunnel 不会被等待
- options - OpenSSL 库选项

> **例 4.3. 保护 CUPS**
>
> 要将 stunnel 配置为 CUPS 的 TLS 包装程序，请使用以下值：
>
> ```
> [cups]
> accept = 632
> connect = 631
> ```
>
> 您可以使用您喜欢的任何空闲端口，而不是 632。631 是 CUPS 通常使用的端口。

3. 创建 chroot 目录，并授予 setuid 选项对其写入访问权限所指定的用户。要做到这一点，请以 root 用户身份输入以下命令：

```
~]# mkdir /var/run/stunnel
~]# chown nobody:nobody /var/run/stunnel
```

这允许 stunnel 创建 PID 文件。

4. 如果您的系统使用不允许访问新端口的防火墙设置，请相应地更改它们。详情请查看 第 5.6.7 节 “使用 GUI 打开端口”。

5. 当您创建配置文件和 chroot 目录后，当您确定指定的端口可以访问后，就可以开始使用 stunnel。

4.8.3. 启动、停止和重启 stunnel

要启动 stunnel，请以 root 用户身份输入以下命令：

```
~]# stunnel /etc/stunnel/stunnel.conf
```

默认情况下，s tunnel 使用 /var/log/secure 来记录其输出。

要终止 stunnel，以 root 用户身份运行以下命令终止进程：

```
~]# kill `cat /var/run/stunnel/stunnel.pid`
```

如果在 stunnel 运行时编辑配置文件，请终止 stunnel，然后再次启动它以使更改生效。

4.9. 加密

4.9.1. 使用 LUKS 磁盘加密

Linux Unified Key Setup-disk-format （或 LUKS）允许您加密 Linux 计算机上的分区。这在涉及到移动计算机和可移动介质时尤为重要。LUKS 允许多个用户密钥解密主密钥，用于分区的批量加密。

LUKS 概述

LUKS 做什么

- LUKS 会加密整个块设备，因此非常适合保护移动设备的内容，如可移动介质或笔记本电脑磁盘驱动器。

- 加密块设备的底层内容是任意的。这使得加密 交换设备 非常有用。对于将特殊格式化块设备用于数据存储的某些数据库，这也很有用。

- LUKS 使用现有的设备映射器内核子系统。
- LUKS 增强了对字典攻击的保护。
- LUKS 设备包含多个密钥插槽，允许用户添加备份密钥或密码短语。

LUKS 不做什么：

- LUKS 不适用于需要许多（超过 8 个）用户对同一设备有不同的访问密钥的情况。
- LUKS 不适用于需要文件级加密的应用程序。

重要

LUKS 等磁盘加密解决方案仅在您的系统关闭时保护数据。一旦系统开启并且 LUKS 解密了磁盘后，通常有权访问该磁盘的任何人都可以使用该磁盘上的文件。

4.9.1.1. Red Hat Enterprise Linux 中的 LUKS 实施

Red Hat Enterprise Linux 7 使用 LUKS 执行文件系统加密。默认情况下，在安装过程中不选中加密文件系统的选项。如果您选择加密硬盘驱动器的选项，系统会提示您输入每次引导计算机时将要求您输入的密码短语。这个密码短语"解锁"用于解密分区的批量加密密钥。如果您选择修改默认分区表，您可以选择要加密的分区。这是在分区表设置中设定的。

LUKS 使用的默认密码（请参阅 cryptsetup --help）是 aes-cbc-essiv:sha256 (ESSIV - Encrypted Salt-Sector Initialization Vector)。请注意，安装程序 Anaconda 默认使用 XTS 模式(aes-xts-plain64)。LUKS 的默认密钥大小为 256 位。Anaconda (XTS 模式)的 LUKS 的默认密钥大小为 512 位。可用的加密系统包括：

- AES - 高级加密标准 - FIPS PUB 197

- Twofish（128 位块加密）
- Serpent
- cast5 - RFC 2144
- cast6 - RFC 2612

4.9.1.2. 手动加密目录

警告

按照以下步骤删除您要加密的分区中的所有数据。您 WILL 丢失了您的所有信息！在开始此流程前，请确保将数据备份到外部源！

1. 以 root 用户身份在 shell 提示符后输入以下内容进入运行级别 1：

```
telinit 1
```

2. 卸载您现有的 /home ：

```
umount /home
```

3. 如果上一步中的命令失败，请使用 fuser 查找进程切换 /home 并终止它们：

```
fuser -mvk /home
```

4. 验证 /home 不再被挂载：

```
grep home /proc/mounts
```

5. **使用随机数据填充分区：**

```
shred -v --iterations=1 /dev/VG00/LV_home
```

这个命令会按照设备的后续写入速度进行，可能需要一些时间才能完成。确保未加密数据保留在已用设备上的重要步骤，并且模糊处理包含加密数据的设备的部分，而不是只是随机数据。

6. **初始化分区：**

```
cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home
```

7. **打开新加密设备：**

```
cryptsetup luksOpen /dev/VG00/LV_home home
```

8. **确保该设备存在：**

```
ls -l /dev/mapper | grep home
```

9. **创建文件系统：**

```
mkfs.ext3 /dev/mapper/home
```

10. **挂载文件系统：**

```
mount /dev/mapper/home /home
```

11. **确保文件系统可见：**

```
df -h | grep home
```

12. **将以下内容添加到 /etc/crypttab 文件中：**

```
home /dev/VG00/LV_home none
```

13. 编辑 /etc/fstab 文件，删除 /home 的旧条目并添加以下行：

```
/dev/mapper/home /home ext3 defaults 1 2
```

14. 恢复默认 SELinux 安全上下文：

```
/sbin/restorecon -v -R /home
```

15. 重启机器：

```
shutdown -r now
```

16. /etc/crypttab 中的条目使您的计算机在引导时询问您的 luks 密码短语。

17. 以 root 身份登录并恢复您的备份。

现在，在计算机关闭时，您已有一个加密的分区，用于安全剩余的所有数据。

4.9.1.3. 向现有设备添加新密码

使用以下命令在现有设备中添加一个新的密码短语：

```
cryptsetup luksAddKey device
```

在提示输入现有验证密码短语之一后，会提示您输入新的密码短语。

4.9.1.4. 从现有设备中删除密码

使用以下命令从现有设备中删除密码短语：

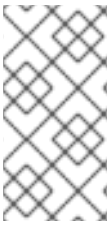
```
cryptsetup luksRemoveKey device
```

系统将提示您输入您要删除的密码短语，然后针对任何剩余的密码进行身份验证。

4.9.1.5. 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设备。这可让您使用加密分区轻松配置系统。

要启用块设备加密，请在创建单个分区、软件 RAID 阵列或逻辑卷时选择 Encrypt System 复选框。完成分区后，会提示您输入加密密码短语。需要此密码短语才能访问加密设备。如果您预先存在的 LUKS 设备，并在安装过程的前面为它们提供正确的密码短语，则密码短语条目对话框也会包含复选框。选中此复选框表示您要添加到每个预先存在的加密块设备中的可用插槽的新密码短语。



注意

在自动分区屏幕上选中 Encrypt System 复选框，然后选择 Create custom layout 不会导致任何块设备被自动加密。

注意

您可以使用 kickstart 为每个新加密块设备设置单独的密码短语。

4.9.1.6. 其它资源

有关 Red Hat Enterprise Linux 7 下 LUKS 或加密硬盘驱动器的附加信息，请访问以下链接之一：

- LUKS 主页
- LUKS/cryptsetup 常见问题解答
- LUKS - Linux Unified Key Setup Wikipedia 文章
- HOWTO ：使用第二个硬盘和 pvmove 创建加密的物理卷(PV)

4.9.2. 创建 GPG 密钥

GPG 用于识别自己并验证您的通信，包括您不知道的人。GPG 允许读取 GPG 签名电子邮件的任何人验证其真实性。换句话说，GPG 允许某人合理地与您签署的通信。GPG 非常有用，因为它有助于防止第三方更改代码或拦截对话并更改消息。

4.9.2.1. 在 GNOME 中创建 GPG 密钥

要在 GNOME 中创建 GPG 密钥，请按照以下步骤操作：

1. 安装 Seahorse 工具，其使 GPG 密钥管理更容易：

```
~]# yum install seahorse
```

2. 要创建密钥，请从 应用程序 → 附件 菜单中选择 Passwords and Encryption Keys，这将启动应用程序 Seahorse。

3. 从 File 菜单中，选择 New，然后选择 PGP Key。然后单击 Continue。

4. 输入您的全名、电子邮件地址和可选注释，描述您是谁（例如：John C. Smith、jsmith@example.com、软件工程师）。点 Create。此时会显示一个对话框，要求输入密钥的密码短语。选择强大的密码短语，但易于记住。点 OK 并创建密钥。

警告

如果您忘记了您的密码短语，您将无法解密数据。

要查找您的 GPG 密钥 ID，请查看新创建的密钥旁边的密钥 ID 列。在大多数情况下，如果您请求密钥 ID，请将 0x 添加到密钥 ID 前，如 0x6789ABCD 中。您应该备份您的私钥，并将它保存在某个地方安全。

4.9.2.2. 在 KDE 中创建 GPG 密钥

要在 KDE 中创建 GPG 密钥，请按照以下步骤操作：

1. 从主菜单中选择 应用程序 → 实用程序 加密工具，从主菜单启动 KGpg 程序。如果您之前使用 KGpg，则程序将引导您完成创建自己的 GPG 密钥对的过程。

2. 此时会出现一个对话框，提示您创建新密钥对。输入您的名字、电子邮件地址和可选注释。您还可以为您的密钥选择过期时间，以及密钥强度（位数）和算法。

3. 在下一个对话框中输入您的密码短语。此时，您的密钥会出现在主 KGpg 窗口中。

警告

如果您忘记了您的密码短语，您将无法解密数据。

要查找您的 GPG 密钥 ID，请查看新创建的密钥旁边的密钥 ID 列。在大多数情况下，如果您请求密钥 ID，请将 0x 添加到密钥 ID 前，如 0x6789ABCD 中。您应该备份您的私钥，并将它保存在某个地方安全。

4.9.2.3. 使用命令行创建 GPG 密钥

1. 使用以下 shell 命令：

```
~]$ gpg2 --gen-key
```

此命令生成由公钥和私钥组成的密钥对。其他人使用您的公钥来验证和解密您的通信。尽可能广泛地分发您的公钥，特别是您所知道希望接收您的身份通信（如邮件列表）的人员。

2. 一系列提示会指示您完成该过程。如果需要，按 Enter 键分配默认值。第一个提示要求您选择您喜欢的密钥类型：

```
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 密钥允许您仅签署通信，也允许您加密文件。

3. 选择密钥大小：

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

此外，默认的 2048 几乎对于所有用户来说都足够了，代表非常强大的安全性级别。

4. 选择 键将过期的时间。最好选择过期日期，而不使用默认值，即无。例如，如果密钥上的电子邮件地址无效，过期日期将提醒其他人停止使用该公钥。

```
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，输入值 1y 使密钥在一年内有效。（如果您更改您的主意，您可以在生成密钥后更改此过期日期。）

5. 在 gpg2 应用程序请求签名信息前，会出现以下提示：

```
Is this correct (y/N)?
```

输入 y 以完成该过程。

6. 输入您的 GPG 密钥的名称和电子邮件地址。请记住，这个过程将作为真实个人进行身份验证。因此，请包含您的真实名称。如果您选择了一个虚假的电子邮件地址，则其他电子邮件地址将更难以查找您的公钥。这使得验证您的通信变得困难。如果您在邮件列表中将这个 GPG 密钥用于自我引入，例如，请输入您在该列表中使用的电子邮件地址。

使用注释字段包含别名或其他信息。（某些人使用不同的密钥来满足不同的目的，并使用注

释来标识每个键，如"Office"或"开源项目"。"）

7. *在确认提示下，输入字母 O 以继续所有条目（如果所有条目都正确），或者使用其他选项来修复任何问题。最后，为您的 secret 密钥输入密码短语。gpg2 程序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 生成随机数据，使您的密钥尽可能唯一。在这一步中移动鼠标、类型随机密钥或在系统上执行其他任务以加快进程。完成此步骤后，您的密钥已完成并可使用：*

```
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. *密钥指纹是您的密钥的简写"signature"。它允许您确认他们已经收到了您的实际公钥，而无需篡改。您不需要将此指纹写出。要随时显示指纹，请使用这个命令替换您的电子邮件地址：*

```
~]$ gpg2 --fingerprint jqdoe@example.com
```

您的"GPG 密钥 ID"由标识公钥的 8 个十六进制数组成。在上例中，GPG 密钥 ID 为 1B2AFA1C。在大多数情况下，如果您请求密钥 ID，请将 0x 添加到密钥 ID 前，如 0x6789ABCD 中。

警告

如果您忘记了您的密码短语，则无法使用密钥，并且使用该密钥加密的任何数据都将丢失。

4.9.2.4. 关于公钥加密

1. *Wikipedia - Public Key Cryptography*

2. *HowStuffWorks - Encryption*

4.9.3. 对公共密钥加密使用 openCryptoki

openCryptoki 是一种 Linux 实现，它是一种公共密钥加密 标准，它定义了名为令牌的加密设备的应用编程接口(API)。令牌可以在硬件或软件中实施。本章概述了在 Red Hat Enterprise Linux 7 中安装、配置和使用 openCryptoki 系统的方式。

4.9.3.1. 安装 openCryptoki 并启动服务

要在您的系统上安装基本的 openCryptoki 软件包，包括令牌的软件实现用于测试目的，请以 root 用户身份输入以下命令：

```
~]# yum install opencryptoki
```

根据您要使用的硬件令牌类型，您可能需要安装为特定用例提供支持的其他软件包。例如：要获得对受信任的平台模块 (TPM)设备的支持，您需要安装 opencryptoki-tpmtok 软件包。

如需有关如何使用 Yum 软件包管理器安装软件包的信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的安装软件包部分。https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/7/html-single/system_administrators_guide/index#sec-Installing

要启用 openCryptoki 服务，您需要运行 pkcsslotd 守护进程。以 root 用户身份执行以下命令，启动当前会话的守护进程：

```
~]# systemctl start pkcsslotd
```

要确保该服务在引导时自动启动，请输入以下命令：

```
~]# systemctl enable pkcsslotd
```

有关如何使用 systemd 目标来管理服务的更多信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的使用 systemd 管理服务 章节。

4.9.3.2. 配置和使用 openCryptoki

启动时，pkcsslotd 守护进程会读取 /etc/opencryptoki/opencryptoki.conf 配置文件，该文件用于收集有关配置用于系统及其插槽的令牌信息。

该文件使用键值对定义各个插槽。每个插槽定义可以包含描述、要使用的令牌库的规格，以及插槽制造商的 ID。另外，也可以定义插槽的硬件和固件的版本。有关文件格式的描述信息请查看 opencryptoki.conf(5) 手册页，以及有关单个键的更详细描述以及可分配给它们的值。

要在运行时修改 pkcsslotd 守护进程的行为，请使用 pkcsconf 工具。此工具允许您显示和配置守护进程的状态，并列出和修改当前配置的插槽和令牌。例如，要显示令牌的相关信息，请发出以下命令（请注意，需要与 pkcsslotd 守护进程通信的所有非 root 用户都必须是 pkcs11 系统组群的一部分）：

```
~]$ pkcsconf -t
```

如需 pkcsconf 工具可用的参数列表，请查看 pkcsconf(1) 手册页。

警告

请记住，pkcs11 组中只能分配完全可信用户，因为此组的所有成员都有阻止 openCryptoki 服务其他用户访问配置的 PKCS628 令牌的权利。这个组的所有成员都可以使用对 openCryptoki 的任何用户的权限执行任意代码。

4.9.4. 使用智能卡向 OpenSSH Supply 凭证

智能卡是 USB 盘、微SD 或 SmartCard 表单因素中的轻量级硬件安全模块。它提供远程可管理的安全密钥存储。在 Red Hat Enterprise Linux 7 中，OpenSSH 支持使用智能卡进行身份验证。

要将智能卡与 OpenSSH 搭配使用，请将卡中的公钥保存到 ~/.ssh/authorized_keys 文件中。在客户端上安装由 opensc 软件包提供的 PKCS efi 库。PKCS vary 是一个公钥加密标准，它定义了一个应用程序编程接口(API)到名为令牌的加密设备。以 root 用户身份输入以下命令：

```
~]# yum install opensc
```

4.9.4.1. 从卡检索公钥

要列出卡中的密钥，请使用 ssh-keygen 命令。使用 -D 指令指定共享库（以下示例中的OpenSC）。

```
~]$ ssh-keygen -D /usr/lib64/pkcs11/opensc-pkcs11.so
ssh-rsa AAAAB3NzaC1yc[...]+g4Mb9
```

4.9.4.2. 在服务器上存储公钥

要使用远程服务器上的智能卡启用验证，请将公钥传送到远程服务器。通过复制检索到的字符串（密钥），并将其粘贴到远程 shell，或者将密钥存储到文件（以下示例中的smartcard.pub ）并使用 ssh-copy-id 命令进行此操作：

```
~]$ ssh-copy-id -f -i smartcard.pub user@hostname
user@hostname's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh user@hostname"
and check to make sure that only the key(s) you wanted were added.
```

在没有私钥文件的情况下存储公钥需要使用 SSH_COPY_ID_LEGACY=1 环境变量或 -f 选项。

4.9.4.3. 使用智能卡中的密钥向服务器进行身份验证

OpenSSH 可以从智能卡读取您的公钥，并使用您的私钥执行操作，而无需公开密钥本身。这意味着私钥不会离开卡。要使用智能卡连接到远程服务器进行验证，请输入以下命令并输入 PIN 保护您的卡：

```
[localhost ~]$ ssh -I /usr/lib64/pkcs11/opensc-pkcs11.so hostname
Enter PIN for 'Test (UserPIN)':
[hostname ~]$
```

使用您要连接的实际主机名替换 hostname。

要在下次连接到远程服务器时保存不必要的输入，请将路径存储在 ~/.ssh/config 文件中：

```
Host hostname
    PKCS11Provider /usr/lib64/pkcs11/opensc-pkcs11.so
```

运行不带任何附加选项的 ssh 命令进行连接：

```
[localhost ~]$ ssh hostname
Enter PIN for 'Test (UserPIN)':
[hostname ~]$
```

4.9.4.4. 使用 ssh-agent 自动执行 PIN Logging In

设置环境变量以使用 ssh-agent 开始。在大多数情况下，您可以跳过这一步，因为 ssh-agent 已在典

型的会话中运行。使用以下命令检查是否可以连接到您的身份验证代理：

```
~]$ ssh-add -l
Could not open a connection to your authentication agent.
~]$ eval `ssh-agent`
```

为了避免在每次使用这个密钥连接时写入 PIN，请运行以下命令将卡添加到代理中：

```
~]$ ssh-add -s /usr/lib64/pkcs11/opensc-pkcs11.so
Enter PIN for 'Test (UserPIN)':
Card added: /usr/lib64/pkcs11/opensc-pkcs11.so
```

要从 ssh-agent 中删除卡，请使用以下命令：

```
~]$ ssh-add -e /usr/lib64/pkcs11/opensc-pkcs11.so
Card removed: /usr/lib64/pkcs11/opensc-pkcs11.so
```

注意

FIPS 201-2 需要由个人身份验证(PIV)卡拥有者明确的用户操作，作为使用卡中存储的数字签名密钥的条件。OpenSC 正确强制实施这一要求。

但是，对于某些应用程序，需要卡拥有者为每个签名输入 PIN。要缓存智能卡 PIN，请在 /etc/opensc-x86_64.conf 中的 pin_cache_ignore_user_consent = true 之前删除 # 字符。

如需更多信息，请参阅 PIV Digital Signature 密钥(NISTIR 7863)的卡持有者身份验证。

4.9.4.5. 其它资源

Red Hat Enterprise Linux 7 的智能卡支持中 描述了设置您的硬件或软件令牌。

有关 pkcs11-tool 实用程序管理和使用智能卡和类似 PKCS facilities 安全令牌的更多信息，请参阅 pkcs11-tool (1) 手册页。

4.9.5. 可信和加密的密钥

可信和加密的密钥 是利用内核密钥环服务的内核生成的可变长度对称密钥。密钥从未以未加密的形式显示在用户空间中，意味着可以验证其完整性，这意味着可以通过扩展验证模块(EVM)来验证并确认正在运行的系统的完整性。用户级程序只能以加密 Blob 的形式访问密钥。

可信密钥需要硬件组件： 受信任的平台模块 (TPM)芯片，用于创建和加密(密封)密钥。TPM 使用名为存储 root 密钥(SRK)的 2048 位 RSA 密钥 密封密钥。

此外，还可以使用特定的 TPM 的平台配置寄存器(PCR)值集密封可信密钥。PCR 包含一组完整性管理值，它们反映了 BIOS、引导装载程序和操作系统。这意味着 PCR- 密封的密钥只能由加密的确切系统上的 TPM 解密。但是，当加载 PCR-sealed 可信密钥（添加到密钥环），并验证其关联的 PCR 值后，就可以使用新的（或将来） PCR 值进行更新，以便可以引导新的内核。单个键也可以保存为多个 blob，每个键都有不同的 PCR 值。

加密密钥不需要 TPM，因为它们使用内核 AES 加密，这使其比可信密钥更快。加密的密钥是使用内核生成的随机数字创建的，并在导入到用户空间 Blob 时由主密钥加密。此主密钥可以是可信密钥或用户密钥，它是其主要缺点 - 如果主密钥不是可信密钥，则加密的密钥仅与用于加密它的用户密钥的安全。

4.9.5.1. 使用密钥

在使用密钥执行任何操作前，请确保在系统中载入 trusted 和 encrypted-keys 内核模块。在不同 RHEL 内核构架中载入内核模块时请考虑以下点：

- 对于带有 x86_64 架构的 RHEL 内核，TRUSTED_KEYS 和 ENCRYPTED_KEYS 代码作为核心内核代码的一部分构建。因此，x86_64 系统用户可以使用这些密钥，而无需加载 trusted 和 encrypted-keys 模块。

- 对于所有其他架构，需要先加载 trusted 和 encrypted-keys 内核模块，然后才能使用密钥执行任何操作。要载入内核模块，请执行以下命令：

```
~]# modprobe trusted encrypted-keys
```

可以使用 keyctl 实用程序创建、加载、导出和更新可信和加密的密钥。有关使用 keyctl 的详情，请参考 keyctl(1)。

注意

要使用 TPM （如用于创建和密封可信密钥），需要启用并激活它。这通常可以通过机器的 BIOS 中的设置，或使用 tpm-tools 软件包中的 tpm_setactive 命令来实现。另外，还需要安装 TrouSers 应用程序(trousers 软件包)和 tcsd 守护进程，它们是 TrouSers 套件的一部分，与 TPM 进行通信。

要使用 TPM 创建可信密钥，请执行具有以下语法的 keyctl 命令：

```
~]$ keyctl add trusted name "new keylength [options]" keyring
```

使用上述语法时，可以按照如下所示构建示例命令：

```
~]$ keyctl add trusted kmk "new 32" @u
642500861
```

上面的示例创建一个名为 kmk 的可信密钥，长度为 32 字节(256 位)，并将其放置在用户密钥环中(@u)。密钥长度为 32 到 128 字节（256 到 1024 位）。使用 show 子命令列出内核密钥环的当前结构：

```
~]$ keyctl show
Session Keyring
       -3 --alswrv    500   500  keyring: _ses
 97833714 --alswrv    500    -1   \_ keyring: _uid.1000
642500861 --alswrv    500   500       \_ trusted: kmk
```

print 子命令将加密密钥输出到标准输出。要将密钥导出到用户空间 blob，请使用 pipe 子命令，如下所示：

```
~]$ keyctl pipe 642500861 > kmk.blob
```

要从 user-space blob 加载可信密钥，请再次使用带有 blob 的 add 命令作为参数：

```
~]$ keyctl add trusted kmk "load `cat kmk.blob`" @u
268728824
```

然后，可以使用 TPM密封的可信密钥来创建安全加密密钥。以下命令语法用于生成加密密钥：

```
~]$ keyctl add encrypted name "new [format] key-type:master-key-name keylength" keyring
```

根据上述语法，可以构建使用已创建的可信密钥生成加密密钥的命令，如下所示：

```
~]$ keyctl add encrypted encr-key "new trusted:kmk 32" @u
159771175
```

要在 TPM 不可用的系统中创建加密密钥，请使用随机数字序列生成用户密钥，然后使用该密钥密封实际加密密钥。

```
~]$ keyctl add user kmk-user "`dd if=/dev/urandom bs=1 count=32 2>/dev/null`" @u
427069434
```

然后，使用 random-number 用户密钥生成加密密钥：

```
~]$ keyctl add encrypted encr-key "new user:kmk-user 32" @u
1012412758
```

list 子命令可用于列出指定内核密钥环中的所有密钥：

```
~]$ keyctl list @u
2 keys in keyring:
427069434: --alswrv  1000  1000 user: kmk-user
1012412758: --alswrv  1000  1000 encrypted: encr-key
```

重要

请记住，未由 master 可信密钥密封的加密密钥仅作为用于加密它们的用户主密钥（随机数字密钥）安全。因此，master 用户密钥应该尽可能安全地加载，最好在引导过程中提前载入。

4.9.5.2. 其它资源

以下离线和在线资源可用于获取与使用可信和加密的密钥相关的其他信息。

安装的文档

- keyctl(1) - 描述 keyctl 工具及其子命令的使用。

在线文档

-

Red Hat Enterprise Linux 7 SELinux 用户和管理员指南 - Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南 介绍了 SELinux 的基本原则，以及如何配置和使用各种服务的 SELinux，如 Apache HTTP 服务器。

- *https://www.kernel.org/doc/Documentation/security/keys-trusted-encrypted.txt* - 有关 Linux 内核可信和加密密钥功能的官方文档。

另请参阅

- 第 A.1.1 节 “高级加密标准 - AES” 提供有关 高级加密标准 的简明描述。
- 第 A.2 节 “公钥加密” 描述公钥加密方法及其使用的各种加密协议。

4.9.6. 使用随机数字生成器

为了能够生成无法轻松中断的安全加密密钥，需要一个随机数字的来源。通常，数字越随机的是，获取唯一密钥的几率越好。生成随机数字的 熵 通常从计算环境"noise"或使用硬件 随机数生成器 获得。

rngd 守护进程是 rng-tools 软件包的一部分，能够同时使用环境 noise 和硬件随机数字生成器来提取熵。守护进程检查由随机性源提供的数据是否足够随机，然后将其存储在内核的随机的熵池中。它生成的随机数字可以通过 /dev/random 和 / dev/urandom 字符设备 提供。

/dev/random 和 /dev/urandom 之间的区别在于，前者是一个阻塞设备，这意味着它在确定熵的数量不足以生成正确的随机输出时停止提供数字。相反，/dev/urandom 是一个非阻塞源，它重复使用内核的熵池，因此可以提供不受限制的伪随机数字，带有较少的熵。因此，/dev/urandom 不应该用于创建长期加密密钥。

要安装 rng-tools 软件包，以 root 用户身份运行以下命令：

```
~]# yum install rng-tools
```

要启动 rngd 守护进程，请以 root 用户身份执行以下命令：

```
~]# systemctl start rngd
```

要查询守护进程的状态，请使用以下命令：

-

```
~]# systemctl status rngd
```

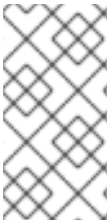
要使用可选参数启动 rngd 守护进程，请直接执行它。例如，要指定随机数字输入的替代源（除 /dev/hwrandom），请使用以下命令：

```
~]# rngd --rng-device=/dev/hwrng
```

上一命令使用 /dev/hwrng 作为读取随机数字的设备启动 rngd 守护进程。同样，您可以使用 -o （或 --random-device）选项为随机数字输出选择内核设备（不是默认的 /dev/random）。有关所有可用选项的列表，请查看 rngd(8) 手册页。

要检查给定系统中有哪些熵源可用，请以 root 用户身份执行以下命令：

```
~]# rngd -vf
Unable to open file: /dev/tpm0
Available entropy sources:
 DRNG
```



注意

输入 rngd -v 命令后，根据的进程会在后台继续运行。默认情况下应用 -b、--background 选项（成为后台程序）。

如果没有 TPM 设备，您将只看到 Intel numeric Random Number Generator (DRNG)作为熵源。要检查您的 CPU 是否支持 RDRAND 处理器指令，请输入以下命令：

```
~]$ cat /proc/cpuinfo | grep rdrand
```

注意

如需更多信息和软件代码示例，请参阅 Intel Digital Random Number Generator (DRNG)软件实现指南。

rng-tools 软件包还包含 rngtest 工具，可用于检查数据的随机性。要测试 /dev/random 输出的随机性级别，请使用 rngtest 工具，如下所示：

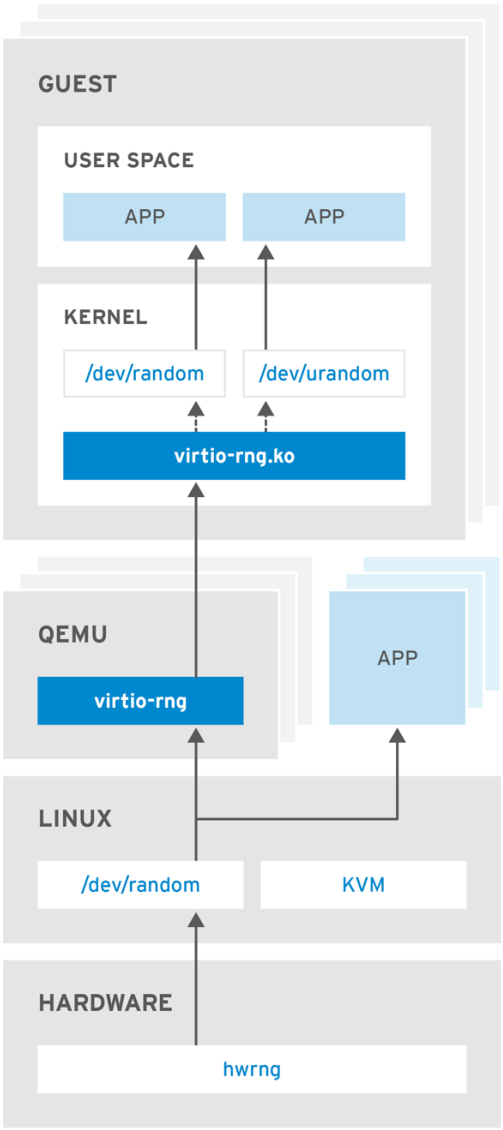
```
~]$ cat /dev/random | rngtest -c 1000
rngtest 5
```

```
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions. There is NO warranty; not even for
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rngtest: starting FIPS tests...
rngtest: bits received from input: 20000032
rngtest: FIPS 140-2 successes: 998
rngtest: FIPS 140-2 failures: 2
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 2
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.171; avg=8.453; max=11.374)Mibits/s
rngtest: FIPS tests speed: (min=15.545; avg=143.126; max=157.632)Mibits/s
rngtest: Program run time: 2390520 microseconds
```

rngtest 工具输出中显示的大量故障表示经过测试的数据的随机性不足，不应依赖于它。有关 rngtest 工具可用选项列表，请查看 rngtest(1) 手册页。

Red Hat Enterprise Linux 7 引入了 virtio RNG （随机数字生成器）设备，它为 KVM 虚拟机提供从主机机器访问熵的 KVM 虚拟机。在推荐的设置中，hwrng 发送到主机 Linux 内核的熵池（通过 /dev/random），QEMU 将使用 /dev/random 作为客户机请求的熵源。

图 4.1. virtio RNG 设备

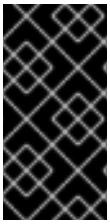


RHEL_453350_0717

[D]

在以前的版本中，Red Hat Enterprise Linux 7.0 和 Red Hat Enterprise Linux 6 客户机可以通过 rngd 用户空间守护进程从主机使用熵。设置守护进程是每个 Red Hat Enterprise Linux 安装的手动步骤。通过 Red Hat Enterprise Linux 7.1，手动步骤已被消除，使整个流程无缝且自动。现在不需要使用 rngd，当可用的熵低于特定阈值时，客户机内核本身从主机获取熵。然后，guest 内核在请求后马上为应用程序提供随机数字。

Red Hat Enterprise Linux 安装程序 Anaconda 现在在其安装程序镜像中提供 virtio-rng 模块，在 Red Hat Enterprise Linux 安装过程中提供可用的主机熵。



重要

要正确决定您应该用于您的场景的随机数字生成器，请参阅 了解 Red Hat Enterprise Linux 随机数字生成器接口 文章。

4.10. 使用基于策略的解密配置加密卷的自动锁定

基于策略的解密(PBD)是技术的集合，允许使用像用户密码、受信任的平台模块(TPM)设备、连接到系统（如智能卡）或特殊网络服务器等不同方法解锁加密的根和硬盘的辅助卷。

PBD 作为技术允许将不同的解锁方法合并到策略中，从而创建以不同方式解锁同一卷的功能。Red Hat Enterprise Linux 中 PBD 的当前实现由 Clevis 框架和名为 pins 的插件组成。每个 pin 都提供单独的解锁功能。现在，唯一可用的 pins 是允许卷使用 TPM 或网络服务器解锁的卷。

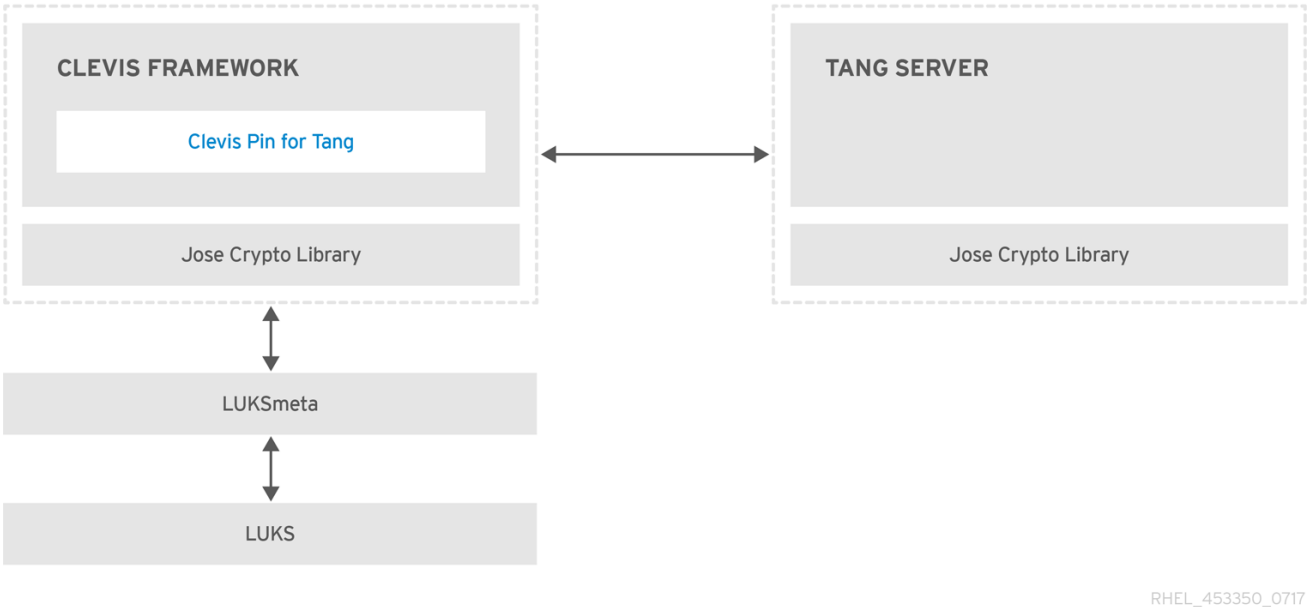
Network Bound Disc Encryption (NBDE)是 PBD 技术的一个子类别，允许将加密卷绑定到特殊的网络服务器。NBDE 的当前实现包括 Tang 服务器的 Clevis pin 和 Tang 服务器本身。

4.10.1. network-Bound Disk Encryption

Network-Bound Disk Encryption (NBDE)允许用户加密物理和虚拟机上的硬盘驱动器的根卷，而无需在系统重启时手动输入密码。

在 Red Hat Enterprise Linux 7 中，NBDE 通过以下组件和技术实现：

图 4.2. 使用 Clevis 和 Tang 的 Network-Bound Disk Encryption



Tang 是一个将数据绑定到网络存在的服务器。当系统绑定到某个安全网络时，它会使包含数据的系统变得可用。Tang 是无状态的，不需要 TLS 或身份验证。与基于 escrow 的解决方案不同，服务器存储所有加密密钥并了解以前使用的每个密钥，Tang 从不与任何客户端密钥进行交互，因此不会从客户端获得任何识别信息。

Clevis 是一个自动化解密的可插拔框架。在 NBDE 中，Clevis 提供 LUKS 卷的自动解锁。clevis 软件包提供该功能的客户端。

Clevis pin 是 Clevis 框架的一个插件。其中一个 pins 是实现与 NBDE 服务器进行交互的插件 - Tang。

Clevis 和 Tang 是通用的客户端和服务器组件，提供网络绑定加密。在 Red Hat Enterprise Linux 7 中，它们与 LUKS 一起使用，以加密和解密 root 和非 root 存储卷，以完成网络绑定磁盘加密。

客户端和服务器端组件都使用 José 库来执行加密和解密操作。

当您开始调配 NBDE 时，Tang 服务器的 Clevis pin 获取 Tang 服务器公告的非对称密钥的列表。或者，由于密钥是非对称的，因此 Tang 的公钥列表可以分发到带外，以便客户端能够在不访问 Tang 服务器的情况下进行操作。此模式称为 脱机调配。

Tang 的 Clevis pin 使用其中一个公钥来生成唯一的强加密的加密密钥。使用此密钥加密数据后，密钥将被丢弃。Clevis 客户端应将此调配操作生成的状态存储在方便的位置。这种加密数据的过程就是 调配步骤。NBDE 的调配状态利用 luksmeta 软件包存储在 LUKS 标头中。

当客户端准备好访问其数据时，它会加载再调配步骤中生成的元数据，并响应恢复加密密钥。此过程是 恢复步骤。

在 NBDE 中，Clevis 使用 pin 绑定 LUKS 卷，以便能自动解锁它。成功完成绑定流程后，可以使用提供的 Dracut 解锁程序解锁磁盘。

所有 LUKS 加密设备，例如 /tmp、/var 和 /usr/local/ 目录，其中包含建立网络连接前需要启动的文件系统，被视为是 root 卷。此外，在网络启动前运行的服务使用的所有挂载点，如 /var/log/、var/log/audit/ 或 /opt，需要在切换到 root 设备后提前挂载。您还可以通过在 /etc/fstab 文件中没有 _netdev 选项来识别根卷。

4.10.2. 安装加密客户端 - Clevis

要在带有加密卷（客户端）的机器上安装 Clevis 可插拔框架及其 pins，请以 root 用户身份输入以下命令：

```
~]# yum install clevis
```

要解密数据，请使用 clevis decrypt 命令，并提供密码文本(JWE)：

```
~]$ clevis decrypt < JWE > PLAINTEXT
```

如需更多信息，请参阅内置 CLI 帮助：

```
~]$ clevis
Usage: clevis COMMAND [OPTIONS]

  clevis decrypt      Decrypts using the policy defined at encryption time
  clevis encrypt http Encrypts using a REST HTTP escrow server policy
  clevis encrypt sss  Encrypts using a Shamir's Secret Sharing policy
  clevis encrypt tang Encrypts using a Tang binding server policy
  clevis encrypt tpm2 Encrypts using a TPM2.0 chip binding policy

~]$ clevis decrypt
Usage: clevis decrypt < JWE > PLAINTEXT

Decrypts using the policy defined at encryption time

~]$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE

Encrypts using a Tang binding server policy

This command uses the following configuration properties:

  url: <string>   The base URL of the Tang server (REQUIRED)

  thp: <string>   The thumbprint of a trusted signing key

  adv: <string>   A filename containing a trusted advertisement
  adv: <object>   A trusted advertisement (raw JSON)

Obtaining the thumbprint of a trusted signing key is easy. If you
have access to the Tang server's database directory, simply do:

    $ jose jwk thp -i $DBDIR/$SIG.jwk

Alternatively, if you have certainty that your network connection
is not compromised (not likely), you can download the advertisement
yourself using:

    $ curl -f $URL/adv > adv.jws
```

4.10.3. 在强制模式中使用 SELinux 部署 Tang 服务器

Red Hat Enterprise Linux 7.7 及更新版本提供了 tangd_port_t SELinux 类型，Tang 服务器可以在 SELinux enforcing 模式下部署为受限制的服务。

先决条件

- 已安装 **policycoreutils-python-utils** 软件包及其依赖项。

流程

1. 要安装 **tang** 软件包及其依赖项，请以 **root** 用户身份输入以下命令：

```
~]# yum install tang
```

2. 选择一个未占用的端口，如 **7500/tcp**，并允许 **tangd** 服务绑定到该端口：

```
~]# semanage port -a -t tangd_port_t -p tcp 7500
```

请注意，一个端口一次只能由一个服务使用，因此尝试使用已经占用的端口意味着 **ValueError: Port already defined** 错误消息。

3. 在防火墙中打开端口：

```
~]# firewall-cmd --add-port=7500/tcp
~]# firewall-cmd --runtime-to-permanent
```

4. 使用 **systemd** 启用 **tangd** 服务：

```
~]# systemctl enable tangd.socket
Created symlink from /etc/systemd/system/multi-user.target.wants/tangd.socket to
/usr/lib/systemd/system/tangd.socket.
```

5. 创建覆盖文件：

```
~]# systemctl edit tangd.socket
```

6.

在以下编辑器屏幕中，其打开了位于 /etc/systemd/system/tangd.socket.d/ 目录中的一个空 override.conf 文件，通过添加以下行将 Tang 服务器的默认端口从 80 改为之前选择的端口号：

```
[Socket]
ListenStream=
ListenStream=7500
```

保存文件并退出编辑器。

7. 重新载入更改的配置并启动 tangd 服务：

```
~]# systemctl daemon-reload
```

8. 检查您的配置是否正常工作：

```
~]# systemctl show tangd.socket -p Listen
Listen=[::]:7500 (Stream)
```

9. 启动 tangd 服务：

```
~]# systemctl start tangd.socket
```

由于 tangd 使用了 systemd 套接字激活机制，因此服务器会在第一次连接进来时就立即启动。在第一次启动时会自动生成一组新的加密密钥。

要执行手动生成密钥等加密操作，请使用 jose 工具。输入 jose -h 命令或查看 jose (1) 手册页以了解更多信息。

例 4.4. 轮转 Tang 密钥

定期轮转您的密钥非常重要。轮转它们的确切间隔取决于您的应用程序、密钥大小和机构策略。有关一些常见建议，请参阅 Cryptographic Key Length Recommendation 页面。

要轮转密钥，请从密钥数据库目录中生成新密钥开始，通常为 /var/db/tang。例如，您可以使用以下命令创建新的签名和交换密钥：

```
~]# DB=/var/db/tang
~]# jose jwk gen -i '{"alg":"ES512"}' -o $DB/new_sig.jwk
~]# jose jwk gen -i '{"alg":"ECMR"}' -o $DB/new_exc.jwk
```

重命名旧密钥，使其具有前导 . 以将它们隐藏在广告中。请注意，以下示例中的文件名与密钥数据库目录中的实际和唯一的文件名不同。

```
~]# mv $DB/old_sig.jwk $DB/.old_sig.jwk
~]# mv $DB/old_exc.jwk $DB/.old_exc.jwk
```

Tang 立即获取所有更改。不需要重启。

此时，新的客户端绑定会获取新密钥，旧客户端可以继续使用旧密钥。当您确定所有旧客户端都使用新密钥时，您可以删除旧的密钥。

警告

请注意，在客户端仍在使用旧密钥时删除旧密钥可能会导致数据丢失。

4.10.3.1. 部署高可用性系统

Tang 提供两种构建高可用性部署的方法：

1. 客户端冗余（推荐）

   客户端应配置成能够绑定到多个 Tang 服务器。在此设置中，每个 Tang 服务器都有自己的密钥，客户端可以通过联系这些服务器的子集来进行解密。Clevis 已通过其 sss 插件支持此工作流。

   有关此设置的详情，请查看以下手册页：

- *tang (8), 部分高可用性*
- *clevis (1), 第 Shamir 的 Secret 共享部分*
- *clevis-encrypt-sss(1)*

红帽建议对高可用性部署使用这个方法。

2. *密钥共享*

出于冗余的目的，可以部署多个 Tang 实例。要设置第二个或后续实例，请安装 tang 软件包，并使用 rsync 通过 SSH 将密钥目录复制到新主机。请注意，红帽不推荐此方法，因为共享密钥会增加密钥的风险，需要额外的自动化基础设施。

4.10.4. 为带有 Tang 的 NBDE 系统部署加密客户端

先决条件

- *Clevis 框架已安装。请查看 第 4.10.2 节 “安装加密客户端 - Clevis”*
- *Tang 服务器或其下载的公告可用。请查看 第 4.10.3 节 “在强制模式中使用 SELinux 部署 Tang 服务器”*

流程

要将 Clevis 加密客户端绑定到 Tang 服务器，请使用 clevis encrypt tang 子命令：

```
~]$ clevis encrypt tang '{"url":"http://tang.srv"}' < PLAINTEXT > JWE
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
```

更改上例中的 http://tang.srv URL，使其与安装 tang 的服务器的 URL 匹配。JWE 输出文件包含您的加密密码文本。这个密码文本是从 PLAINTEXT 输入文件中读取的。

要解密数据，请使用 clevis decrypt 命令，并提供密码文本(JWE)：

```
~]$ clevis decrypt < JWE > PLAINTEXT
```

如需更多信息，请参阅 clevis-encrypt-tang (1) 手册页或使用内置 CLI 帮助：

```
~]$ clevis
Usage: clevis COMMAND [OPTIONS]

  clevis decrypt      Decrypts using the policy defined at encryption time
  clevis encrypt http Encrypts using a REST HTTP escrow server policy
  clevis encrypt sss  Encrypts using a Shamir's Secret Sharing policy
  clevis encrypt tang Encrypts using a Tang binding server policy
  clevis luks bind    Binds a LUKSv1 device using the specified policy
  clevis luks unlock  Unlocks a LUKSv1 volume

~]$ clevis decrypt
Usage: clevis decrypt < JWE > PLAINTEXT

Decrypts using the policy defined at encryption time

~]$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE

Encrypts using a Tang binding server policy

This command uses the following configuration properties:

  url: <string>   The base URL of the Tang server (REQUIRED)

  thp: <string>   The thumbprint of a trusted signing key

  adv: <string>   A filename containing a trusted advertisement
  adv: <object>   A trusted advertisement (raw JSON)

Obtaining the thumbprint of a trusted signing key is easy. If you
have access to the Tang server's database directory, simply do:

    $ jose jwk thp -i $DBDIR/$SIG.jwk

Alternatively, if you have certainty that your network connection
is not compromised (not likely), you can download the advertisement
yourself using:

    $ curl -f $URL/adv > adv.jws
```

4.10.5. 使用 TPM 2.0 策略部署加密客户端

在 64 位 Intel 或 64 位 AMD 架构的系统上，要部署使用受信任的平台模块 2.0 (TPM 2.0)芯片加密的客户端，请使用 clevis encrypt tpm2 子命令，以 JSON 配置对象的形式唯一参数：

```
~]$ clevis encrypt tpm2 '{}' < PLAINTEXT > JWE
```

要选择不同的层次结构、哈希和密钥算法，请指定配置属性，例如：

```
~]$ clevis encrypt tpm2 '{"hash":"sha1","key":"rsa"}' < PLAINTEXT > JWE
```

要解密数据，请提供密码文本(JWE)：

```
~]$ clevis decrypt < JWE > PLAINTEXT
```

pin 还支持将数据封装到平台配置寄存器(PCR)状态。这样，只有 PCR 哈希值与密封时使用的策略匹配时，数据才能被解封。

例如，对于 SHA1 银行，使用索引 0 和 1 将数据封装到 PCR：

```
~]$ clevis encrypt tpm2 '{"pcr_bank":"sha1","pcr_ids":"0,1"}' < PLAINTEXT > JWE
```

如需更多信息以及可能的配置属性列表，请参阅 clevis-encrypt-tpm2 (1) 手册页。

4.10.6. 配置根卷的手动注册

要自动解锁现有的 LUKS 加密的根卷，请安装 clevis-luks 子软件包并使用 clevis luks bind 命令将卷绑定到 Tang 服务器：

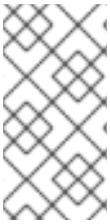
```
~]# yum install clevis-luks
```

```
~]# clevis luks bind -d /dev/sda tang '{"url":"http://tang.srv"}'
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
You are about to initialize a LUKS device for metadata storage.
Attempting to initialize it may result in data loss if data was
already written into the LUKS header gap in a different format.
A backup is advised before initialization is performed.

Do you wish to initialize /dev/sda? [yn] y
Enter existing LUKS password:
```

此命令执行四个步骤：

1. 使用与 LUKS 主密钥相同的无序状态测量法创建新的密钥。
2. 使用 Clevis 加密新密钥.
3. 使用 LUKSMeta 将 Clevis JWE 对象存储在 LUKS 标头中。
4. 启用与 LUKS 一起使用的新密钥。

现在，可以使用您的现有密码和 Clevis 策略解锁此磁盘。如需更多信息，请参阅 clevis-luks-bind (1) 手册页。



注意

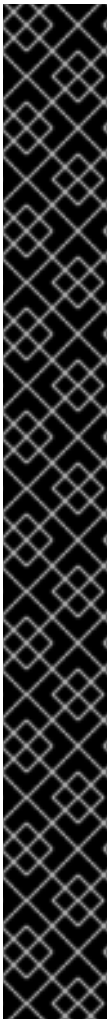
绑定过程假定至少有一个可用的 LUKS 密码插槽。clevis luks bind 命令占用了其中一个插槽。

要验证 Clevis JWE 对象是否已成功放置在 LUKS 标头中，请使用 luksmeta show 命令：

```
~]# luksmeta show -d /dev/sda
0   active empty
1   active cb6e8904-81ff-40da-a84a-07ab9ab5715e
2 inactive empty
3 inactive empty
4 inactive empty
5 inactive empty
6 inactive empty
7 inactive empty
```

要启用早期引导系统来处理磁盘绑定，请在已安装的系统中输入以下命令：

```
~]# yum install clevis-dracut
~]# dracut -f --regenerate-all
```



重要

要将 NBDE 用于带有静态 IP 配置（没有 DHCP）的客户端，请手动将网络配置传递给 dracut 工具，例如：

```
~]# dracut -f --regenerate-all --kernel-cmdline "ip=192.0.2.10 netmask=255.255.255.0
gateway=192.0.2.1 nameserver=192.0.2.45"
```

或者，在 /etc/dracut.conf.d/ 目录中创建一个带有静态网络信息的 .conf 文件。例如：

```
~]# cat /etc/dracut.conf.d/static_ip.conf
kernel_cmdline="ip=10.0.0.103 netmask=255.255.252.0 gateway=10.0.0.1
nameserver=10.0.0.1"
```

重新生成初始 RAM 磁盘镜像：

```
~]# dracut -f --regenerate-all
```

详情请查看 dracut.cmdline (7) 手册页。

4.10.7. 使用 Kickstart 配置自动注册

Clevis 可以与 Kickstart 集成，以提供完全自动化的注册过程。

1. 指示 Kickstart 对磁盘进行分区，以便使用临时密码为所有挂载点（除 /boot ）启用了 LUKS 加密。注册过程的这一步中的密码是临时密码。

```
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --grow --encrypted --passphrase=temppass
```

请注意，OSPP-complaint 系统需要更复杂的配置，例如：

```
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --size=2048 --encrypted --passphrase=temppass
part /var --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /tmp --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /home --fstype="xfs" --ondisk=vda --size=2048 --grow --encrypted --
passphrase=temppass
```

```
part /var/log --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /var/log/audit --fstype="xfs" --ondisk=vda --size=1024 --encrypted --
passphrase=temppass
```

2. 通过在 %packages 部分中列出它们来安装相关的 Clevis 软件包：

```
%packages
clevis-dracut
%end
```

3. 在 %post 部分中调用 clevis luks bind 来执行绑定。之后，删除临时密码：

```
%post
clevis luks bind -f -k- -d /dev/vda2 \
tang '{"url":"http://tang.srv","thp":"_OsIk0T-E2l6qjfdDiwVmidoZjA"}' \ <<< "temppass"
cryptsetup luksRemoveKey /dev/vda2 <<< "temppass"
%end
```

在上例中，请注意，我们指定在 Tang 服务器上信任的 thumbprint 作为绑定配置的一部分，从而允许绑定完全非交互式。

在使用 TPM 2.0 策略而不是 Tang 服务器时，您可以使用类似的流程。

有关 Kickstart 安装的详情，请查看 Red Hat Enterprise Linux 7 安装指南。有关 Linux Unified Key Setup-on-disk-format (LUKS)的详情，请参考 第 4.9.1 节 “使用 LUKS 磁盘加密”。

4.10.8. 配置可移动存储设备的自动锁定

要自动解锁 LUKS 加密的可移动存储设备，如 USB 驱动器，请安装 clevis-udisks2 软件包：

```
~]# yum install clevis-udisks2
```

重启系统，然后使用 clevis luks bind 命令执行绑定步骤，如 第 4.10.6 节 “配置根卷的手动注册” 所述：

```
~]# clevis luks bind -d /dev/sdb1 tang '{"url":"http://tang.srv"}'
```

现在，可以在 GNOME 桌面会话中自动解锁 LUKS 加密的可移动设备。绑定到 Clevis 策略的设备也可以通过 clevis luks unlock 命令解锁：

```
~]# clevis luks unlock -d /dev/sdb1
```

在使用 TPM 2.0 策略而不是 Tang 服务器时，您可以使用类似的流程。

4.10.9. 在引导时配置非 root 卷的自动锁定

要使用 NBDE 同时解锁 LUKS 加密的非 root 卷，请执行以下步骤：

1. 安装 clevis-systemd 软件包：

```
~]# yum install clevis-systemd
```

2. 启用 Clevis unlocker 服务：

```
~]# systemctl enable clevis-luks-askpass.path
Created symlink from /etc/systemd/system/remote-fs.target.wants/clevis-luks-askpass.path to
/usr/lib/systemd/system/clevis-luks-askpass.path.
```

3. 使用 clevis luks bind 命令执行绑定步骤，如 第 4.10.6 节 “配置根卷的手动注册” 所述。

4. 要在系统引导过程中设置加密块设备，请将带有 _netdev 选项的对应行添加到 /etc/crypttab 配置文件中。详情请查看 crypttab (5) 手册页。

5. 将卷添加到 /etc/fstab 文件中可访问文件系统的列表。此配置文件中也使用 _netdev 选项。详情请查看 fstab (5) 手册页。

4.10.10. 在 NBDE 网络中部署虚拟机

clevis luks bind 命令不会改变 LUKS 主密钥。这意味着，如果您创建了一个 LUKS 加密镜像以便在虚拟机或云环境中使用，则运行此镜像的所有实例都将共享主密钥。这极其不安全，应始终避免。

这不是 Clevis 的一个限制，而是 LUKS 的设计原则。如果您希望在云中有一个加密的根卷，则需要确保对云中的每个 Red Hat Enterprise Linux 实例都执行了安装过程（通常使用 Kickstart）。如果没有共享 LUKS 主密钥，就无法共享镜像。

如果要在虚拟化环境中部署自动解锁，红帽强烈建议您将 lorax 或 virt-install 等系统与 Kickstart 文件（请参阅 第 4.10.7 节 “使用 Kickstart 配置自动注册”）或其他自动配置工具一起使用，以确保每个加密的虚拟机都有一个唯一的主密钥。

4.10.11. 使用 NBDE 为云环境构建可自动注册的虚拟机镜像

在云环境中部署可自动注册的加密镜像会带来一系列独特的挑战。与其他虚拟化环境一样，建议减少从一个镜像启动的实例数量，以避免共享 LUKS 主密钥。

因此，最佳实践是创建自定义映像，这些映像不在任何公共存储库中共享，为部署有限数量的实例提供了基础。要创建的实例的确切数量应由部署的安全策略定义，并且基于与 LUKS 主密钥攻击向量关联的风险容错能力。

要构建启用 LUKS 的自动化部署，应当使用 Lorax 或 virt-install 等系统以及一个 Kickstart 文件，来确保镜像构建过程中主密钥的唯一性。

云环境支持我们在这里考虑的两种 Tang 服务器部署选项。首先，Tang 服务器可以在云环境本身中部署。其次，Tang 服务器可以部署在云外的独立的基础架构上，并且这两个基础架构之间有 VPN 连接。

在云中原生部署 Tang 可以轻松部署。但是，考虑到它与其他系统的密文数据持久性层共享基础设施，因此 Tang 服务器的私钥和 Clevis 元数据可以存储在同一个物理磁盘上。对这个物理磁盘的访问允许密文数据的完全泄露。

重要

因此，红帽强烈建议在存储数据的位置和运行 Tang 的系统之间保持物理隔离。在云和 Tang 服务器之间的这种隔离可确保 Tang 服务器的私钥不会被意外与 Clevis 元数据组合。如果云基础设施面临风险，它还提供了对 Tang 服务器的本地控制。

4.10.12. 其它资源

如何设置带有多个 LUKS 设备的 Network Bound Disk Encryption (Clevis+Tang unlocking) 知识库文章。

如需更多信息，请参阅以下手册页：

- ***tang(8)***
- ***clevis (1)***
- ***jose(1)***
- ***clevis-luks-unlockers(1)***
- ***tang-nagios(1)***

4.11. 使用 AIDE检查完整性

高级入侵检测环境（Advanced Intrusion Detection Environment，简称 AIDE）是一个实用工具，它可以创建系统上的文件数据库，然后利用该数据库来确保文件的完整性，并检测系统入侵。

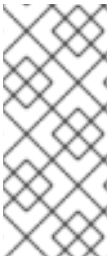
4.11.1. 安装 AIDE

要安装 aide 软件包，请以 root 用户身份输入以下命令：

```
~]# yum install aide
```

要生成初始数据库，请以 root 用户身份输入以下命令：

```
~]# aide --init

AIDE, version 0.15.1

### AIDE database at /var/lib/aide/aide.db.new.gz initialized.
```



注意

在默认配置中，aide --init 命令只检查 /etc/aide.conf 文件中定义的一组目录和文件。要在 AIDE 数据库中包含其他目录或文件，并更改其监视的参数，请相应地编辑 /etc/aide.conf。

要开始使用数据库，请从初始数据库文件名中删除 .new 子字符串：

```
~]# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

要修改 AIDE 数据库的位置，请编辑 /etc/aide.conf 文件并修改 DBDIR 值。要获得额外的安全性，请将数据库、配置和 /usr/sbin/aide 二进制文件存储在安全的位置，如只读介质。

重要

为了避免 AIDE 数据库位置更改后的 SELinux 拒绝，请相应地更新您的 SELinux 策略。如需更多信息，请参阅 SELinux 用户和管理员指南。

4.11.2. 执行完整性检查

要启动手动检查，请以 root 用户身份输入以下命令：

```
~]# aide --check
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2017-03-30 14:12:56

Summary:
  Total number of files: 147173
  Added files:   1
  Removed files:  0
  Changed files:  2
...
```

AIDE 至少应配置为运行每周扫描。多数情况下，AIDE 应该每天运行。例如，要使用 cron 计划在每天 4:05 执行 AIDE （请参阅系统管理员指南中的 Automating System Tasks 章节），请将以下行添加到 /etc/crontab ：

```
05 4 * * * root /usr/sbin/aide --check
```

4.11.3. 更新 AIDE 数据库

在验证了软件包更新或配置文件调整等系统更改后，更新您的基准 AIDE 数据库：

```
~]# aide --update
```

aide --update 命令创建 /var/lib/aide/aide.db.new.gz 数据库文件。要开始使用它进行完整性检查，

请从文件名中删除 .new 子字符串。

4.11.4. 其它资源

有关 AIDE 的更多信息，请参阅以下文档：

- aide(1) 手册页
- aide.conf(5) man page
- Red Hat Enterprise Linux 7 安全配置指南(OpenSCAP 安全指南)：使用 AIDE 验证完整性

4.12. 使用 USBGUARD

USBGuard 软件框架通过实施基于设备属性的基本白名单和黑名单功能，提供对入侵 USB 设备的系统保护。要强制执行用户定义的策略，USBGuard 使用 Linux 内核 USB 设备授权功能。USBGuard 框架提供以下组件：

- 带有进程间通信(IPC)接口的守护进程组件，用于动态交互和策略强制执行。
- 与正在运行的 USBGuard 实例交互的命令行界面。
- 编写 USB 设备授权策略的规则语言。
- 用于与共享库中实施的守护进程交互的 C++ API。

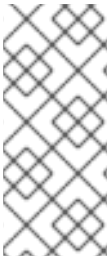
4.12.1. 安装 USBGuard

要安装 usbguard 软件包，请以 root 用户身份输入以下命令：

```
~]# yum install usbguard
```

要创建初始规则集，请以 root 用户身份输入以下命令：

```
~]# usbguard generate-policy > /etc/usbguard/rules.conf
```



注意

要自定义 USBGuard 规则集，请编辑 /etc/usbguard/rules.conf 文件。详情请查看 usbguard-rules.conf (5) 手册页。另外，请参阅 第 4.12.3 节 “使用规则语言创建您自己的策略”。

要启动 USBGuard 守护进程，请以 root 用户身份输入以下命令：

```
~]# systemctl start usbguard.service
~]# systemctl status usbguard
● usbguard.service - USBGuard daemon
   Loaded: loaded (/usr/lib/systemd/system/usbguard.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2017-06-06 13:29:31 CEST; 9s ago
     Docs: man:usbguard-daemon(8)
 Main PID: 4984 (usbguard-daemon)
   CGroup: /system.slice/usbguard.service
           └─4984 /usr/sbin/usbguard-daemon -k -c /etc/usbguard/usbguard-daem...
```

要确保 USBGuard 在系统启动时自动启动，请以 root 用户身份运行以下命令：

```
~]# systemctl enable usbguard.service
Created symlink from /etc/systemd/system/basic.target.wants/usbguard.service to
/usr/lib/systemd/system/usbguard.service.
```

要列出 USBGuard 识别的所有 USB 设备，请以 root 用户身份输入以下命令：

```
~]# usbguard list-devices
1: allow id 1d6b:0002 serial "0000:00:06.7" name "EHCI Host Controller" hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" parent-hash
"4PHGcaDKWtPjKDwYpIRG722cB9SlGz9l9lea93+Gt9c=" via-port "usb1" with-interface 09:00:00
...
6: block id 1b1c:1ab1 serial "000024937962" name "Voyager" hash
"CrXgiaWIf2bZAU+5WkzOE7y0rdSO82XMzubn7HDb95Q=" parent-hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" via-port "1-3" with-interface 08:06:50
```

要授权设备与系统交互，请使用 allow-device 选项：

```
~]# usbguard allow-device 6
```

要取消授权并从系统中删除设备，请使用 reject-device 选项。要只取消授权设备，请使用带有 block-device 选项的 usbguard 命令：

```
~]# usbguard block-device 6
```

usbguard 使用 block 和 reject 术语，其含义如下：

- block - 暂时不要与这个设备通信
- reject - 忽略这个设备，就像不存在一样

要查看 usbguard 命令的所有选项，请使用 --help 指令输入它：

```
~]$ usbguard --help
```

4.12.2. 创建白名单列表和黑名单列表

usbguard-daemon.conf 文件由 usbguard 守护进程加载，在解析其命令行选项后，用于配置守护进程的运行时参数。要覆盖默认配置文件(/etc/usbguard/usbguard-daemon.conf)，请使用 -c 命令行选项。详情请查看 usbguard-daemon (8) 手册页。

要创建白名单或黑色列表，请编辑 usbguard-daemon.conf 文件并使用以下选项：

usbguard 配置文件

RuleFile=<path>

usbguard 守护进程使用此文件从中加载策略规则集，并编写通过 IPC 接口接收的新规则。

IPCAllowedUsers=<username> [<username> ...]

守护进程将接受来自的 IPC 连接的用户名列表。

IPCAllowedGroups=<groupname> [<groupname> ...]

守护进程将接受来自的 IPC 连接的组名称列表。

IPCAccessControlFiles=<path>

保存 IPC 访问控制文件的目录的路径。

ImplicitPolicyTarget=<target>

如何对待策略中的任何规则都不匹配的设备。接受的值： allow、 block、 reject。

PresentDevicePolicy=<policy>

如何处理守护进程启动时已连接的设备：

- Allow - 授权每个存在的设备
- block - 取消授权每个存在的设备
- reject - 删除每个存在的设备
- keep - 只同步内部状态并保留它
- apply-policy - 评估每个存在设备的规则集

PresentControllerPolicy=<policy>

如何处理守护进程启动时已连接的 USB 控制器：

- **Allow - 授权每个存在的设备**
- **block - 取消授权每个存在的设备**
- **reject - 删除每个存在的设备**
- **keep - 只同步内部状态并保留它**
- **apply-policy - 评估每个存在设备的规则集**

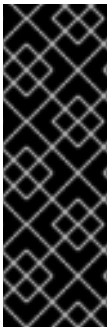
例 4.5. usbguard 配置

以下配置文件订购 usbguard 守护进程从 /etc/usbguard/rules.conf 文件中加载规则，它只允许 usbguard 组中的用户使用 IPC 接口：

```
RuleFile=/etc/usbguard/rules.conf
IPCAccessControlFiles=/etc/usbguard/IPCAccessControl.d/
```

要指定 IPC 访问控制列表(ACL)，请使用 usbguard add-user 或 usbguard remove-user 命令。如需了解更多详细信息，请参阅 usbguard (1)。在本例中，要允许 usbguard 组中的用户修改 USB 设备授权状态、列出 USB 设备、侦听异常事件以及列出 USB 授权策略，请以 root 用户身份输入以下命令：

```
~]# usbguard add-user -g usbguard --devices=modify,list,listen --policy=list --exceptions=listen
```



重要

守护进程提供 USBGuard 公共 IPC 接口。在 Red Hat Enterprise Linux 中，对此接口的访问默认仅限于 root 用户。考虑设置 IPCAccessControlFiles 选项（推荐）或 IPCAllowedUsers 和 IPCAllowedGroups 选项，来限制对 IPC 接口的访问。不要将 ACL 保留为未配置，因为这会向所有本地用户公开 IPC 接口，并允许他们操作 USB 设备的授权状态并修改 USBGuard 策略。

如需更多信息，请参阅 usbguard-daemon.conf (5) 手册页中的 IPC 访问控制部分。

4.12.3. 使用规则语言创建您自己的策略

usbguard 守护进程决定是否根据一组规则定义的策略授权 USB 设备。当 USB 设备插入系统时，守护进程会按顺序扫描现有规则，并在找到匹配规则时，根据规则目标授权(allows)、取消授权（阻止）或删除（拒绝）设备。如果没有找到匹配的规则，则决定基于隐式默认目标。这个隐式默认为阻止设备，直到用户做出决定。

规则语言 grammar 如下：

```
rule ::= target device_id device_attributes conditions.

target ::= "allow" | "block" | "reject".

device_id ::= "*:*" | vendor_id ":*" | vendor_id ":" product_id.

device_attributes ::= device_attributes | attribute.
device_attributes ::= .

conditions ::= conditions | condition.
conditions ::= .
```

有关规则语言（如 target、设备规格或设备属性）的详情，请查看 usbguard-rules.conf (5) 手册页。

例 4.6. usbguard 示例策略

允许 USB mass 存储设备以及阻止所有其他设备

这个策略会阻止任何不仅仅是一个大容量存储设备的设备。USB 闪存磁盘中带有隐藏键盘接口的设备被阻止。只有具有单个大容量存储接口的设备才能与操作系统交互。该策略由一个规则组成：

```
allow with-interface equals { 08:*:* }
```

阻塞是隐式的，因为没有块规则。隐式阻止对桌面用户很有用，因为侦听 USBGuard 事件的桌面小程序可以询问用户是否为设备选择了隐式目标。

允许通过特定端口连接特定的 Yubikey 设备

拒绝该端口上的所有其他操作。

```
allow 1050:0011 name "Yubico Yubikey II" serial "0001234567" via-port "1-2" hash
"044b5e168d40ee0245478416caf3d998"
reject via-port "1-2"
```

拒绝具有可疑接口组合的设备

实施键盘或网络接口的 USB 闪存磁盘非常可疑。以下一组规则形成了一个策略，它允许 USB 闪存磁盘，并使用额外的可疑接口明确拒绝设备。

```
allow with-interface equals { 08:*:* }
reject with-interface all-of { 08:*:* 03:00:* }
reject with-interface all-of { 08:*:* 03:01:* }
reject with-interface all-of { 08:*:* e0:*:* }
reject with-interface all-of { 08:*:* 02:*:* }
```

注意

黑名单是错误的方法，您不应该只将一组设备列入黑名单，并允许其余设备。上面的策略假定阻止是隐式默认值。拒绝一组设备被视为"bad"是一种良好的方法，如何将系统暴露限制尽可能多。

允许只使用键盘的 USB 设备

只有已经允许使用键盘接口的 USB 设备时，以下规则才允许键盘 USB 设备。

```
allow with-interface one-of { 03:00:01 03:01:01 } if !allowed-matches(with-interface one-of {
03:00:01 03:01:01 })
```

使用 **usbguard generate-policy** 命令生成初始策略后，编辑 **/etc/usbguard/rules.conf** 来自定义 USBGuard 策略规则。

```
~]$ usbguard generate-policy > rules.conf
~]$ vim rules.conf
```

要安装更新的策略并使您的更改有效，请使用以下命令：

```
~]# install -m 0600 -o root -g root rules.conf /etc/usbguard/rules.conf
```

4.12.4. 其它资源

有关 USBGuard 的更多信息，请参阅以下文档：

- usbguard (1) 手册页
- usbguard-rules.conf(5) man page
- usbguard-daemon (8) 手册页
- usbguard-daemon.conf(5) man page
- USBGuard 主页

4.13. 强化 TLS 配置

TLS (传输层安全)是用于保护网络通信的加密协议。当通过配置首选 密钥交换协议、身份验证方法 和加密算法 来强化系统安全设置时，需要注意支持的客户端的范围越大，生成的安全性较低。相反，严格的安全设置会导致与客户端的兼容性受限，这可能导致某些用户被锁定在系统之外。请确保以最严格的可用配置为目标，并且仅在出于兼容性原因需要时才放宽配置。

请注意，Red Hat Enterprise Linux 7 中包含的库提供的默认设置对于大多数部署来说都足够安全。TLS 实现尽可能使用安全算法，而不阻止来自或到旧客户端或服务器的连接。在满足严格的安全要求的环境中应用此部分中描述的强化设置，其中不支持安全算法或协议的旧客户端或服务器无法连接或允许连接。

4.13.1. 选择启用算法

需要选择和配置几个组件。以下每个都直接影响生成的配置（以及客户端中的支持级别）或解决方案在系统上拥有的计算需求。

协议版本

TLS 的最新版本提供最佳安全机制。除非有充分的理由包含对旧版本的 TLS （甚至 SSL）的支持，否则允许您的系统只使用最新版本的 TLS 来协商连接。

不允许使用 SSL 版本 2 或 3 协商。两个版本都有严重的安全漏洞。只允许使用 TLS 版本 1.0 或更高

版本的协商。当前版本的 TLS 1.2 应始终是首选的。

注意

请注意，当前所有 TLS 版本的安全性取决于使用 TLS 扩展、特定密码（请参阅以下）和其他临时解决方案。所有 TLS 连接对等点都需要实施安全重新协商指示(RFC 5746)，且必须对 CBC-mode 密码(Lucky Thirteen 攻击)实施缓解时间攻击。TLS 1.0 客户端还需要额外实施记录分割（针对EAST 攻击的临时解决方案）。TLS 1.2 支持通过关联数据 (AEAD)模式密码进行身份验证加密，如 AES-GCM、AES-CCM 或 Camellia-GCM，它们没有已知的问题。所有上述缓解方案均在 Red Hat Enterprise Linux 中包含的加密库中实现。

有关协议版本和推荐使用的快速概述，请参阅 表 4.6 “协议版本”。

表 4.6. 协议版本

协议版本	使用建议
SSL v2	不要使用。具有严重的安全漏洞。
SSL v3	不要使用。具有严重的安全漏洞。
TLS 1.0	在需要时用于互操作性.已知的无法以保证互操作性的方式缓解的问题，因此不会默认启用缓解方案。不支持现代加密套件。
TLS 1.1	在需要时用于互操作性.没有已知问题，但依赖于 Red Hat Enterprise Linux 中的所有 TLS 实现中包含的协议修复。不支持现代加密套件。

协议版本	使用建议
TLS 1.2	推荐的版本。支持现代 AEAD 密码套件。

Red Hat Enterprise Linux 中的一些组件被配置为使用 TLS 1.0，即使它们支持 TLS 1.1 甚至 1.2。这是尝试实现最高级别的互操作性和外部服务（可能不支持最新版本的 TLS ）的动机。根据您的互操作性要求，启用最高可用的 TLS 版本。

重要

不建议使用 SSL v3。但是，如果事实上，虽然它被视为不安全且不适合用于常规用途，但绝对必须保持 SSL v3 启用。有关如何使用 stunnel 安全地加密通信的说明，即使使用了不支持加密的服务，或者只能使用过时的、不安全的加密模式。第 4.8 节 “使用 stunnel”

密码套件

现代、更安全 的密码套件应该优先于旧的、不安全的密码套件。一直禁止 eNULL 和 aNULL 密码套件的使用，它们根本不提供任何加密或身份验证。如果可能，基于 RC4 或 HMAC-MD5 的密码套件也应被禁用。这同样适用于所谓的 导出 密码套件，它们被有意较弱，因此很容易中断。

虽然不能立即不安全，但提供超过 128 位安全性的密码套件不应被视为其简短的有用生命周期。使用 128 位或更高安全性的算法可以预期在至少数年内不破坏，因此强烈建议这样做。请注意，虽然 3DES 密码公告使用 168 位，但它们实际上提供了 112 位的安全性。

始终优先选择支持 （完美）转发保密 (PFS)的密码套件，这样可确保在服务器密钥泄露时加密数据的机密性。这个规则排除了快速 RSA 密钥交换，但允许使用 ECDHE 和 DHE。在两者中，ECDHE 速度更快，因此首选。

您还应该在 CBC-mode 密码之前优先使用 AEAD 密码，如 AES-GCM，因为它们不会受到 padding oracle 攻击的影响。另外，在很多情况下，AES-GCM 比 CBC 模式的 AES 快，特别是在硬件具有 AES 加密加速器时。

另请注意，当使用带有 ECDSA 证书的 ECDHE 密钥交换时，事务的速度甚至比纯 RSA 密钥交换要快。要支持旧客户端，您可以在服务器上安装两对证书和密钥：一对带有 ECDSA 密钥（用于新客户

端），以及一个带有 RSA 密钥（用于旧密钥）。

公钥长度

当使用 RSA 密钥时，总是首选使用至少由 SHA-256 签名的 3072 位的密钥长度，对于真实的 128 位安全性来说，这个密钥长度足够大。

警告

请记住，您的系统安全性仅与链中最弱的链接一样强大。例如，只是一个强大的密码不能保证良好安全性。密钥和证书以及 认证机构 (CA)用来签署您的密钥的哈希功能和密钥同样重要。

4.13.2. 使用 TLS 的实现

Red Hat Enterprise Linux 7 提供了多个功能全面的 TLS 实现。本节描述了 OpenSSL 和 GnuTLS 的配置。有关如何在独立应用程序中配置 TLS 支持的说明，请参阅 第 4.13.3 节 “配置特定应用程序”。

可用的 TLS 实现支持各种 密码套件，用于定义建立和使用 TLS-secured 通信时附带的所有元素。

在考虑 第 4.13.1 节 “选择启用算法” 中概述的建议时，使用不同实现中的工具列出并指定为您的用例提供最佳安全性的密码套件。然后，生成的密码套件可用于配置独立应用程序协商和安全连接的方式。

重要

请确定在每次更新或升级您使用的 TLS 实现或升级使用该实现的应用程序后检查您的设置。新版本可能会引入您不希望启用的新密码套件，并且当前配置没有禁用。

4.13.2.1. 在 OpenSSL 中使用 Cipher Suites

OpenSSL 是一个工具包和一个加密库，它支持 SSL 和 TLS 协议。在 Red Hat Enterprise Linux 7 中，配置文件在 /etc/pki/tls/openssl.cnf 中提供。这个配置文件的格式在 config(1) 中进行了描述。另请参阅 第 4.7.9 节 “配置 OpenSSL”。

要获取安装 OpenSSL 支持的所有密码套件的列表，请使用 openssl 命令和 password 子命令，如下

所示：

```
~]$ openssl ciphers -v 'ALL:COMPLEMENTOFALL'
```

将其他参数（称为 OpenSSL 文档中的 密码字符串和 关键字 ）传给 password 子命令，以缩小输出范围。 特殊关键字可用于列出满足特定条件的套件。例如，要只列出定义为 HIGH 组的套件，请使用以下命令：

```
~]$ openssl ciphers -v 'HIGH'
```

有关可用关键字和密码字符串的列表，请参阅 ciphers(1) 手册页。

要获得满足 第 4.13.1 节 “选择启用算法” 中推荐的密码套件列表，请使用类似如下的命令：

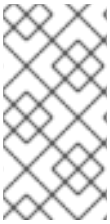
```
~]$ openssl ciphers -v 'kEECDH+aECDSA+AES:kEECDH+AES+aRSA:kEDH+aRSA+AES' |
column -t
ECDHE-ECDSA-AES256-GCM-SHA384  TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AESGCM(256)
Mac=AEAD
ECDHE-ECDSA-AES256-SHA384      TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AES(256)
Mac=SHA384
ECDHE-ECDSA-AES256-SHA         SSLv3    Kx=ECDH  Au=ECDSA  Enc=AES(256)    Mac=SHA1
ECDHE-ECDSA-AES128-GCM-SHA256  TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AESGCM(128)
Mac=AEAD
ECDHE-ECDSA-AES128-SHA256      TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AES(128)
Mac=SHA256
ECDHE-ECDSA-AES128-SHA         SSLv3    Kx=ECDH  Au=ECDSA  Enc=AES(128)    Mac=SHA1
ECDHE-RSA-AES256-GCM-SHA384    TLSv1.2  Kx=ECDH  Au=RSA    Enc=AESGCM(256)
Mac=AEAD
ECDHE-RSA-AES256-SHA384        TLSv1.2  Kx=ECDH  Au=RSA    Enc=AES(256)    Mac=SHA384
ECDHE-RSA-AES256-SHA           SSLv3    Kx=ECDH  Au=RSA    Enc=AES(256)    Mac=SHA1
ECDHE-RSA-AES128-GCM-SHA256    TLSv1.2  Kx=ECDH  Au=RSA    Enc=AESGCM(128)
Mac=AEAD
ECDHE-RSA-AES128-SHA256        TLSv1.2  Kx=ECDH  Au=RSA    Enc=AES(128)    Mac=SHA256
ECDHE-RSA-AES128-SHA           SSLv3    Kx=ECDH  Au=RSA    Enc=AES(128)    Mac=SHA1
DHE-RSA-AES256-GCM-SHA384      TLSv1.2  Kx=DH    Au=RSA    Enc=AESGCM(256)
Mac=AEAD
DHE-RSA-AES256-SHA256          TLSv1.2  Kx=DH    Au=RSA    Enc=AES(256)    Mac=SHA256
DHE-RSA-AES256-SHA             SSLv3    Kx=DH    Au=RSA    Enc=AES(256)    Mac=SHA1
DHE-RSA-AES128-GCM-SHA256      TLSv1.2  Kx=DH    Au=RSA    Enc=AESGCM(128)
Mac=AEAD
DHE-RSA-AES128-SHA256          TLSv1.2  Kx=DH    Au=RSA    Enc=AES(128)    Mac=SHA256
DHE-RSA-AES128-SHA             SSLv3    Kx=DH    Au=RSA    Enc=AES(128)    Mac=SHA1
```

以上命令省略了所有不安全的密码，它提供了 临时的 elliptic curve Diffie-Hellman 密钥交换和 ECDSA 密码，并省略 RSA 密钥交换（确保 完美的转发保密）。

请注意，这是一个严格的配置，可能需要放宽真实场景中的条件，以允许与更广泛的客户端兼容。

4.13.2.2. 在 GnuTLS 中使用 Cipher Suites

gnutls 是一个实现 SSL 和 TLS 协议和相关技术的通信库。



注意

Red Hat Enterprise Linux 7 上的 GnuTLS 安装提供了最佳默认配置值，为大多数用例提供足够的安全性。除非需要满足特殊安全要求，否则建议使用提供的默认值。

使用带有 **-l**（或 **--list**）选项的 **gnutls-cli** 命令列出所有支持的密码套件：

```
~]$ gnutls-cli -l
```

要缩小 **-l** 选项显示的密码套件列表，请将一个或多个参数（称为 GnuTLS 文档中的 优先级字符串 和关键字）传给 **--priority** 选项。有关所有可用优先级字符串的列表，请参阅 http://www.gnutls.org/manual/gnutls.html#Priority-Strings 中的 GnuTLS 文档。例如，运行以下命令来获取提供至少 128 位安全性的密码套件列表：

```
~]$ gnutls-cli --priority SECURE128 -l
```

要获得满足 第 4.13.1 节 “选择启用算法” 中推荐的密码套件列表，请使用类似如下的命令：

```
~]$ gnutls-cli --priority SECURE256:+SECURE128:-VERS-TLS-ALL:+VERS-TLS1.2:-RSA:-DHE-
DSS:-CAMELLIA-128-CBC:-CAMELLIA-256-CBC -l
Cipher suites for SECURE256:+SECURE128:-VERS-TLS-ALL:+VERS-TLS1.2:-RSA:-DHE-DSS:-
CAMELLIA-128-CBC:-CAMELLIA-256-CBC
TLS_ECDHE_ECDSA_AES_256_GCM_SHA384                0xc0, 0x2c      TLS1.2
TLS_ECDHE_ECDSA_AES_256_CBC_SHA384                0xc0, 0x24      TLS1.2
TLS_ECDHE_ECDSA_AES_256_CBC_SHA1                  0xc0, 0x0a      SSL3.0
TLS_ECDHE_ECDSA_AES_128_GCM_SHA256                0xc0, 0x2b      TLS1.2
TLS_ECDHE_ECDSA_AES_128_CBC_SHA256                0xc0, 0x23      TLS1.2
TLS_ECDHE_ECDSA_AES_128_CBC_SHA1                  0xc0, 0x09      SSL3.0
TLS_ECDHE_RSA_AES_256_GCM_SHA384                  0xc0, 0x30      TLS1.2
TLS_ECDHE_RSA_AES_256_CBC_SHA1                    0xc0, 0x14      SSL3.0
TLS_ECDHE_RSA_AES_128_GCM_SHA256                  0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                  0xc0, 0x27      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA1                    0xc0, 0x13      SSL3.0
TLS_DHE_RSA_AES_256_CBC_SHA256                    0x00, 0x6b      TLS1.2
TLS_DHE_RSA_AES_256_CBC_SHA1                      0x00, 0x39      SSL3.0
TLS_DHE_RSA_AES_128_GCM_SHA256                    0x00, 0x9e      TLS1.2
```

```
TLS_DHE_RSA_AES_128_CBC_SHA256                0x00, 0x67      TLS1.2
TLS_DHE_RSA_AES_128_CBC_SHA1                  0x00, 0x33      SSL3.0

Certificate types: CTYPE-X.509
Protocols: VERS-TLS1.2
Compression: COMP-NULL
Elliptic curves: CURVE-SECP384R1, CURVE-SECP521R1, CURVE-SECP256R1
PK-signatures: SIGN-RSA-SHA384, SIGN-ECDSA-SHA384, SIGN-RSA-SHA512, SIGN-ECDSA-
SHA512, SIGN-RSA-SHA256, SIGN-DSA-SHA256, SIGN-ECDSA-SHA256
```

以上命令将输出限制为至少 128 位安全性的密码，同时优先选择更强大的密码。它还禁止 RSA 密钥交换和 DSS 身份验证。

请注意，这是一个严格的配置，可能需要放宽真实场景中的条件，以允许与更广泛的客户端兼容。

4.13.3. 配置特定应用程序

不同的应用为 TLS 提供自己的配置机制。本节介绍了最常用的服务器应用程序所使用的与 TLS相关的配置文件，并提供典型配置示例。

无论您选择使用什么配置，始终确保您的服务器应用程序强制实施 服务器端密码顺序，以便使用的密码套件由您配置的顺序决定。

4.13.3.1. 配置 Apache HTTP 服务器

Apache HTTP 服务器 可以使用 OpenSSL 和 NSS 库来满足其 TLS 的需求。根据您选择的 TLS 库，您需要安装 mod_ssl 或 mod_nss 模块（由 eponymous 软件包提供）。例如，要安装提供 OpenSSL mod_ssl 模块的软件包，请以 root 身份运行以下命令：

```
~]# yum install mod_ssl
```

mod_ssl 软件包安装 /etc/httpd/conf.d/ssl.conf 配置文件，该文件可用于修改 Apache HTTP Server 的与 TLS相关的设置。同样，mod_nss 软件包会安装 /etc/httpd/conf.d/nss.conf 配置文件。

安装 httpd-manual 软件包以获取 Apache HTTP 服务器 的完整文档，包括 TLS 配置。/etc/httpd/conf.d/ssl.conf 配置文件中的指令在 /usr/share/httpd/manual/mod_ssl.html 中进行了详细介绍。各种设置示例位于 /usr/share/httpd/manual/ssl/ssl_howto.html。

修改 /etc/httpd/conf.d/ssl.conf 配置文件中的设置时，请确保至少考虑以下三个指令：

SSLProtocol

使用这个指令指定您要允许的 TLS 版本（或 SSL）。

SSLCipherSuite

使用这个指令来指定您首选的密码套件或禁用您要禁止的密码套件。

SSLHonorCipherOrder

取消注释并将此指令设置为 on，以确保连接的客户端遵循您指定的密码顺序。

例如：

```
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite HIGH:!aNULL:!MD5
SSLHonorCipherOrder on
```

请注意，上述配置最小是裸机，可以根据 第 4.13.1 节 “选择启用算法” 中概述的建议来显著强化。

要配置和使用 mod_nss 模块，请修改 /etc/httpd/conf.d/nss.conf 配置文件。mod_nss 模块派生自 mod_ssl，因此它与它共享许多功能，而不用配置文件结构以及可用的指令共享。请注意，mod_nss 指令的前缀为 NSS 而不是 SSL。有关 mod_nss 的信息的概述信息，包括不适用于 mod_nss 的 mod_ssl 配置指令列表。 https://git.fedorahosted.org/cgit/mod_nss.git/plain/docs/mod_nss.html

4.13.3.2. 配置 Dovecot 邮件服务器

要将 Dovecot 邮件服务器的安装配置为使用 TLS，请修改 /etc/dovecot/conf.d/10-ssl.conf 配置文件。您可以在 /usr/share/doc/dovecot-2.2.10/wiki/SSL.DovecotConfiguration.txt 中找到该文件中一些基本配置指令的说明（此帮助文件与 Dovecot的标准安装一起安装）。

修改 /etc/dovecot/conf.d/10-ssl.conf 配置文件中的设置时，请确保至少考虑以下三个指令：

ssl_protocols

使用这个指令指定您要允许的 TLS 版本（或 SSL）。

ssl_cipher_list

使用这个指令指定您首选的密码套件或禁用您要禁止的密码套件。

ssl_prefer_server_ciphers

取消注释并将此指令设置为 **yes**，以确保连接的客户端遵循您指定的密码顺序。

例如：

```
ssl_protocols = !SSLv2 !SSLv3
ssl_cipher_list = HIGH:!aNULL:!MD5
ssl_prefer_server_ciphers = yes
```

请注意，上述配置最小是裸机，可以根据 第 4.13.1 节 “选择启用算法” 中概述的建议来显著强化。

4.13.4. 其它信息

有关 TLS 配置和相关主题的更多信息，请参阅以下列出的资源。

安装的文档

- **config(1) - 描述 /etc/ssl/openssl.conf 配置文件的格式。**
- **ciphers(1) - 包含可用 OpenSSL 关键字和密码字符串的列表。**
- **/usr/share/httpd/manual/mod_ssl.html - 包含对 Apache HTTP 服务器 使用的 mod_ ssl 模块使用的 /etc/httpd/conf.d/ ssl.conf 配置文件中提供的指令的详细描述。**
- **/usr/share/httpd/manual/ssl/ssl_howto.html - 在 Apache HTTP 服务器的 mod_ ssl /conf.d/ ssl.conf 配置文件中包含实际设置的实际示例。**
- **/usr/share/doc/dovecot-2.2.10/wiki/SSL.DovecotConfiguration.txt - 解释 Dovecot 邮件服务器使用的 /etc/dovecot/conf.d/10-ssl.conf 配置文件中提供的一些基本配置指令。**

在线文档

-

Red Hat Enterprise Linux 7 SELinux 用户和管理员指南 - Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南 介绍了 SELinux 的基本原则，以及如何配置和使用各种服务的 SELinux，如 Apache HTTP 服务器。

- http://tools.ietf.org/html/draft-ietf-uta-tls-bcp-00 - 安全使用 TLS 和 DTLS 的建议。

另请参阅

- 第 A.2.4 节 “SSL/TLS” 提供 SSL 和 TLS 协议的简要描述。

- 第 4.7 节 “使用 OpenSSL” 除其他方面，如何使用 OpenSSL 创建和管理密钥、生成证书以及加密和解密文件。

4.14. 使用共享系统证书

共享系统证书存储允许 NSS、GnuTLS、OpenSSL 和 Java 共享用于检索系统证书定位符和黑色列表信息的默认源。默认情况下，信任存储包含 Mozilla CA 列表，包括正和负信任。系统允许更新核心 Mozilla CA 列表或选择其他证书列表。

4.14.1. 使用系统范围的信任存储

在 Red Hat Enterprise Linux 7 中，整合的系统范围的信任存储位于 /etc/pki/ca-trust/ 和 /usr/share/pki/ca-trust-source/ 目录中。对 /usr/share/pki/ca-trust-source/ 中信任设置的优先级的处理低于/etc/pki/ca-trust/ 中的设置。

证书文件根据它们所安装到的子目录处理：

- /usr/share/pki/ca-trust-source/anchors/ 或 /etc/pki/ca-trust/source/anchors/ - 用于信任锚。请参阅 第 4.5.6 节 “了解信任 Anchors”。

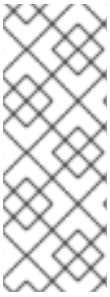
- /usr/share/pki/ca-trust-source/blacklist/ 或 /etc/pki/ca-trust/source/blacklist/ - 用于不受信任的证书。

- /usr/share/pki/ca-trust-source/ 或 /etc/pki/ca-trust/source/ - 用于扩展 BEGIN TRUSTED 文件格式的证书。

4.14.2. 添加新证书

要将简单 PEM 或 DER 文件格式的证书添加到系统上信任的 CA 列表中，请将证书文件复制到 /usr/share/pki/ca-trust-source/anchors/ 或 /etc/pki/ca-trust/source/anchors/ 目录中。要更新系统范围的信任存储配置，请使用 update-ca-trust 命令，例如：

```
# cp ~/certificate-trust-examples/Cert-trust-test-ca.pem /usr/share/pki/ca-trust-source/anchors/
# update-ca-trust
```



注意

虽然 Firefox 浏览器可以使用添加的证书而无需执行 update-ca-trust，但建议在 CA 更改后运行 update-ca-trust。另请注意，浏览器，如 Firefox、Epiphany 或 Chromium、缓存文件，您可能需要清除浏览器的缓存或重新启动浏览器来加载当前的系统证书配置。

4.14.3. 管理受信任的系统证书

要列出、提取、添加、删除或修改信任锚，请使用 trust 命令。要查看这个命令的内置帮助信息，请不要输入任何参数，或使用 --help 指令：

```
$ trust
usage: trust command <args>...

Common trust commands are:
  list            List trust or certificates
  extract         Extract certificates and trust
  extract-compat  Extract trust compatibility bundles
  anchor          Add, remove, change trust anchors
  dump            Dump trust objects in internal format

See 'trust <command> --help' for more information
```

要列出所有系统信任锚和证书，请使用 trust list 命令：

```
$ trust list
pkcs11:id=%d2%87%b4%e3%df%37%27%93%55%f6%56%ea%81%e5%36%cc%8c%1e%3f%bd;ty
pe=cert
    type: certificate
    label: ACCVRAIZ1
    trust: anchor
    category: authority

pkcs11:id=%a6%b3%e1%2b%2b%49%b6%d7%73%a1%aa%94%f5%01%e7%73%65%4c%ac%50;t
ype=cert
    type: certificate
```

```
    label: ACEDICOM Root
    trust: anchor
    category: authority
...
[output has been truncated]
```

trust 命令的所有子命令都提供了详细的内置帮助，例如。

```
$ trust list --help
usage: trust list --filter=<what>

  --filter=<what>     filter of what to export
                      ca-anchors        certificate anchors
                      blacklist         blacklisted certificates
                      trust-policy      anchors and blacklist (default)
                      certificates      all certificates
                      pkcs11:object=xx  a PKCS#11 URI
  --purpose=<usage>   limit to certificates usable for the purpose
                      server-auth       for authenticating servers
                      client-auth       for authenticating clients
                      email             for email protection
                      code-signing      for authenticating signed code
                      1.2.3.4.5...      an arbitrary object id
  -v, --verbose       show verbose debug output
  -q, --quiet         suppress command output
```

要将信任锚存储在系统范围的信任存储中，请使用 trust anchor 子命令，并指定 path.to a certificate，例如：

```
# trust anchor path.to/certificate.crt
```

要删除证书，请使用 path. 到 证书或证书的 ID：

```
# trust anchor --remove path.to/certificate.crt
# trust anchor --remove "pkcs11:id=%AA%BB%CC%DD%EE;type=cert"
```

4.14.4. 其它资源

如需更多信息，请参阅以下手册页：

- **update-ca-trust(8)**

- trust(1)

4.15. 使用 MACSEC

媒体访问控制安全性 (MACsec、IEEE 802.1AE)使用 GCM-AES-128 算法加密并验证 LAN 中的所有流量。MACsec 不仅可以保护 IP，还可以保护地址解析协议(ARP)、邻居发现(ND)或 DHCP。虽然 IPsec 在网络层（层 3）和 SSL 或 TLS 上运行，但 MACsec 在应用程序层（层 7）上运行，但 MACsec 在数据链路层（层 2）中运行。将 MACsec 与其它网络层的安全协议相结合，以利用这些标准提供的不同安全功能。

有关 MACsec 网络架构、用例场景和配置示例的更多信息，请参阅 MACsec: 不同的解决方案来加密网络流量。

有关如何使用 wpa_supplicant 和 NetworkManager 配置 MACsec 的示例，请参阅 Red Hat Enterprise Linux 7 网络指南。

4.16. 使用 清理安全地删除数据

scrub 实用程序设置特殊文件或磁盘设备上的模式，以便更难以检索数据。使用 清理 比在磁盘上写入随机数据要快。此过程提供了高可用性、可靠性和数据保护。

要使用 scrub 命令启动，请安装 scrub 软件包：

```
~]# yum install scrub
```

scrub 工具以以下基本模式之一运行：

字符或块设备

与整个磁盘对应的 特殊文件 会被清理，并销毁它上的所有数据。这是最有效的方法。

```
scrub [OPTIONS] special file
```

File

常规文件会被清理，且仅销毁文件中的数据。

```
scrub [OPTIONS] file
```

目录

使用 -X 选项时，会创建目录并填充文件，直到文件系统已满为止。然后，文件在文件模式中清理为。

```
scrub -X [OPTIONS] directory
```

例 4.7. 清理原始设备

要清理 带有默认国家安全管理(NNSA)模式的原始设备 /dev/sdf1，请输入以下命令：

```
~]# scrub /dev/sdf1
scrub: using NNSA NAP-14.1-C patterns
scrub: please verify that device size below is correct!
scrub: scrubbing /dev/sdf1 1995650048 bytes (~1GB)
scrub: random  |.....................................................|
scrub: random  |.....................................................|
scrub: 0x00    |.....................................................|
scrub: verify  |.....................................................|
```

例 4.8. 清理文件

1. *创建一个 1MB 文件：*

```
~]$ base64 /dev/urandom | head -c $[ 1024*1024 ] > file.txt
```

2. *显示文件大小：*

```
~]$ ls -lh
total 1.0M
-rw-rw-r--. 1 username username 1.0M Sep  8 15:23 file.txt
```

3. *显示文件的内容：*

```
~]$ head -1 file.txt
JnNpaTEveB/IYsbM9lhuJdw+0jKhwCIBUsxLXLAyB8ultotUlNHKKUeS/7bCRKDogE
P+yJm8VQkL
```

4. 清理文件：

```
~]$ scrub file.txt
scrub: using NNSA NAP-14.1-C patterns
scrub: scrubbing file.txt 1048576 bytes (~1024KB)
scrub: random  |.............................................|
scrub: random  |.............................................|
scrub: 0x00    |.............................................|
scrub: verify  |.............................................|
```

5. 验证文件内容是否已清理：

```
~]$ cat file.txt
SCRUBBED!
```

6. 验证文件大小是否保持不变：

```
~]$ ls -lh
total 1.0M
-rw-rw-r--. 1 username username 1.0M Sep  8 15:24 file.txt
```

有关 清理 模式、选项、方法和注意事项的详情，请查看 scrub(1) man page。

第 5 章 使用防火墙

5.1. FIREWALLD入门

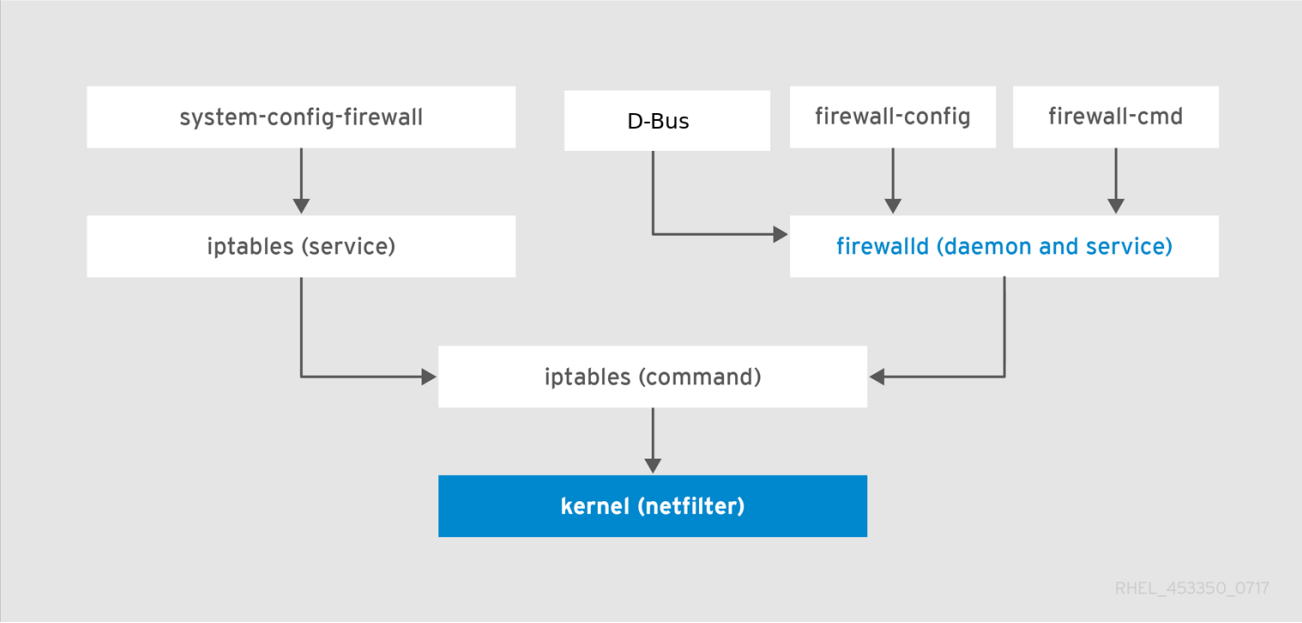
防火墙是保护机器不受来自外部的、不需要的网络数据的一种方式。它允许用户通过定义一组防火墙规则来控制主机上的入站网络流量。这些规则用于对进入的流量进行排序，并可以阻断或允许流量。

firewalld 是一个防火墙服务守护进程，通过 D-Bus 接口提供动态可自定义的基于主机的防火墙。如果是动态的，它可在每次修改规则时启用、修改和删除规则，而不需要在每次修改规则时重启防火墙守护进程。

firewalld 使用 区域和服务 的概念来简化流量管理。zones 是预定义的规则集。网络接口和源可以分配给区。允许的流量取决于您计算机连接到的网络，并分配了这个网络的安全级别。防火墙服务是预定义的规则，覆盖了允许特定服务进入流量的所有必要设置，并在区中应用。

服务使用一个或多个 端口或 地址进行 网络通信。防火墙会根据端口过滤通讯。要允许服务的网络流量，必须打开 其端口。firewalld 会阻止未明确设置为打开的端口上的所有流量。一些区（如 可信 区）默认允许所有流量。

图 5.1. 防火墙堆栈



[D]

5.1.1. Zones

可以根据用户对该网络中的接口和流量设置的信任程度，使用 firewalld 来将网络划分为不同的区。一个连接只能是一个区的一部分，但一个区可以被用来进行很多网络连接。

NetworkManager 通知接口区的 firewalld。您可以使用 firewall-config 工具或 firewall-cmd 命令行工具为 NetworkManager 分配区域。后两个只编辑适当的 NetworkManager 配置文件。如果您使用 firewall-cmd 或 firewall-config 更改接口区，则请求会转发到 NetworkManager，且不会由firewalld 处理。

预定义的区存储在 /usr/lib/firewalld/zones/ 目录中，并可立即应用到任何可用的网络接口。只有在修改后，这些文件才会被拷贝到 /etc/firewalld/zones/ 目录中。下表描述了预定义区的默认设置：

block

任何传入的网络连接都会被拒绝，并显示 IPv4 的 icmp-host-prohibited 消息，对于 IPv6 的 icmp6-adm-prohibited 消息。只有从系统启动的网络连接才能进行。

dmz

对于您的非企业化区里的计算机来说，这些计算机可以被公开访问，且有限访问您的内部网络。只接受所选的入站连接。

drop

所有传入的网络数据包都会丢失，没有任何通知。只有外发网络连接也是可行的。

external

适用于启用了伪装的外部网络，特别是路由器。您不信任网络中的其他计算机不会损害您的计算机。只接受所选的入站连接。

home

用于家用，因为您可以信任其他计算机。只接受所选的入站连接。

internal

当您主要信任网络中的其他计算机时，供内部网络使用。只接受所选的入站连接。

public

可用于您不信任网络中其他计算机的公共区域。只接受所选的入站连接。

trusted

所有网络连接都被接受。

work

可用于您主要信任网络中其他计算机的工作。只接受所选的入站连接。

这些区中的一个被设置为 **default** 区。当接口连接被添加到 **NetworkManager** 中时，它们会被分配到默认区。安装时，**firewalld** 中的默认区被设为 **public** 区。默认区可以被修改。

注意

网络区名称已被选择进行自我解释，并允许用户快速做出合理的决定。要避免安全问题，请查看默认区配置并根据您的需要和风险禁用任何不必要的服务。

5.1.2. 预定义的服务

服务可以是本地端口、协议、源端口和目的地列表，并在启用了服务时自动载入防火墙帮助程序模块列表。使用服务可节省用户时间，因为它们可以完成一些任务，如打开端口、定义协议、启用数据包转发等等，而不必在另外的步骤中设置所有任务。

服务配置选项和通用文件信息在 **firewalld.service (5)** 手册页中进行了描述。服务通过单独的 **XML** 配置文件来指定，这些文件采用以下格式命名：**service-name.xml** 。协议名称优先于 **firewalld** 中的服务或应用程序名称。

5.1.3. 运行时和永久设置

仅在 运行时 模式中提交的任何更改才会在 firewalld 运行时应用。当 firewalld 重启时，设置会恢复到其 永久 值。

要使更改在重启后持久保留，请使用 --permanent 选项再次应用它们。或者，若要在 firewalld 运行时保留更改，请使用 --runtime-to-permanent firewall-cmd 选项。

如果您在 firewalld 只使用 --permanent 选项运行时设置规则，则在重启 firewalld 前它们不会生效。但是，重启 firewalld 会关闭所有打开的端口，并停止网络流量。

5.1.4. 使用 CLI 修改运行时和永久配置中的设置

使用 CLI，您不会同时修改这两种模式的防火墙设置。您只能修改运行时模式或永久模式。要在永久模式下修改防火墙设置，请在 firewall-cmd 命令中使用 --permanent 选项。

```
~]# firewall-cmd --permanent <other options>
```

如果没有这个选项，命令将修改运行时模式。

要更改这两种模式的设置，您可以使用以下两种方法：

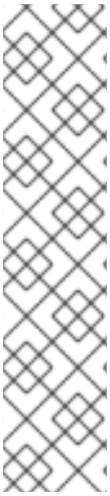
1. 更改运行时设置，然后将其持久化，如下：

```
~]# firewall-cmd <other options>
~]# firewall-cmd --runtime-to-permanent
```

2. 设置永久性设置并将设置重新载入运行时模式：

```
~]# firewall-cmd --permanent <other options>
~]# firewall-cmd --reload
```

第一种方法允许您在将设置应用到永久模式前测试这些设置。



注意

特别是在远程系统中，不正确的设置可能会导致用户锁定其自身的机器。要防止这种情况，请使用 **--timeout** 选项。在指定时间后，任何更改都会恢复到之前的状态。使用此选项排除 **--permanent** 选项。

例如，将 SSH 服务添加 15 分钟：

```
~]# firewall-cmd --add-service=ssh --timeout 15m
```

5.2. 安装 FIREWALL-CONFIG GUI 配置工具

要使用 firewall-config GUI 配置工具，以 root 用户身份安装 firewall-config 软件包：

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

或者，在 GNOME 中，使用 Super 键并输入 Software 来启动 软件源 应用程序。在搜索框中输入 firewall，在右上角选择搜索按钮后会出现。从搜索结果中选择 Firewall 项，然后点 Install 按钮。

要运行 firewall-config，请使用 firewall-config 命令，或者按 Super 键进入 活动概览，输入 firewall，然后按 Enter。

5.3. 查看 FIREWALLD的当前状态和设置

5.3.1. 查看 firewalld的当前状态

默认情况下，防火墙服务 firewalld 会在系统上安装。使用 firewalld CLI 接口来检查该服务是否正在运行。

查看服务的状态：

```
~]# firewall-cmd --state
```

如需有关服务状态的更多信息，请使用 systemctl status 子命令：

```
~]# systemctl status firewalld
```

```
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor pr
   Active: active (running) since Mon 2017-12-18 16:05:15 CET; 50min ago
     Docs: man:firewalld(1)
 Main PID: 705 (firewalld)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/firewalld.service
           └─705 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid
```

另外，在尝试编辑设置前，务必要知道如何设置 firewalld 以及哪些规则被强制使用。要显示防火墙设置，请查看 第 5.3.2 节 “查看当前的 firewalld 设置”

5.3.2. 查看当前的 firewalld 设置

5.3.2.1. 使用 GUI 查看允许的服务

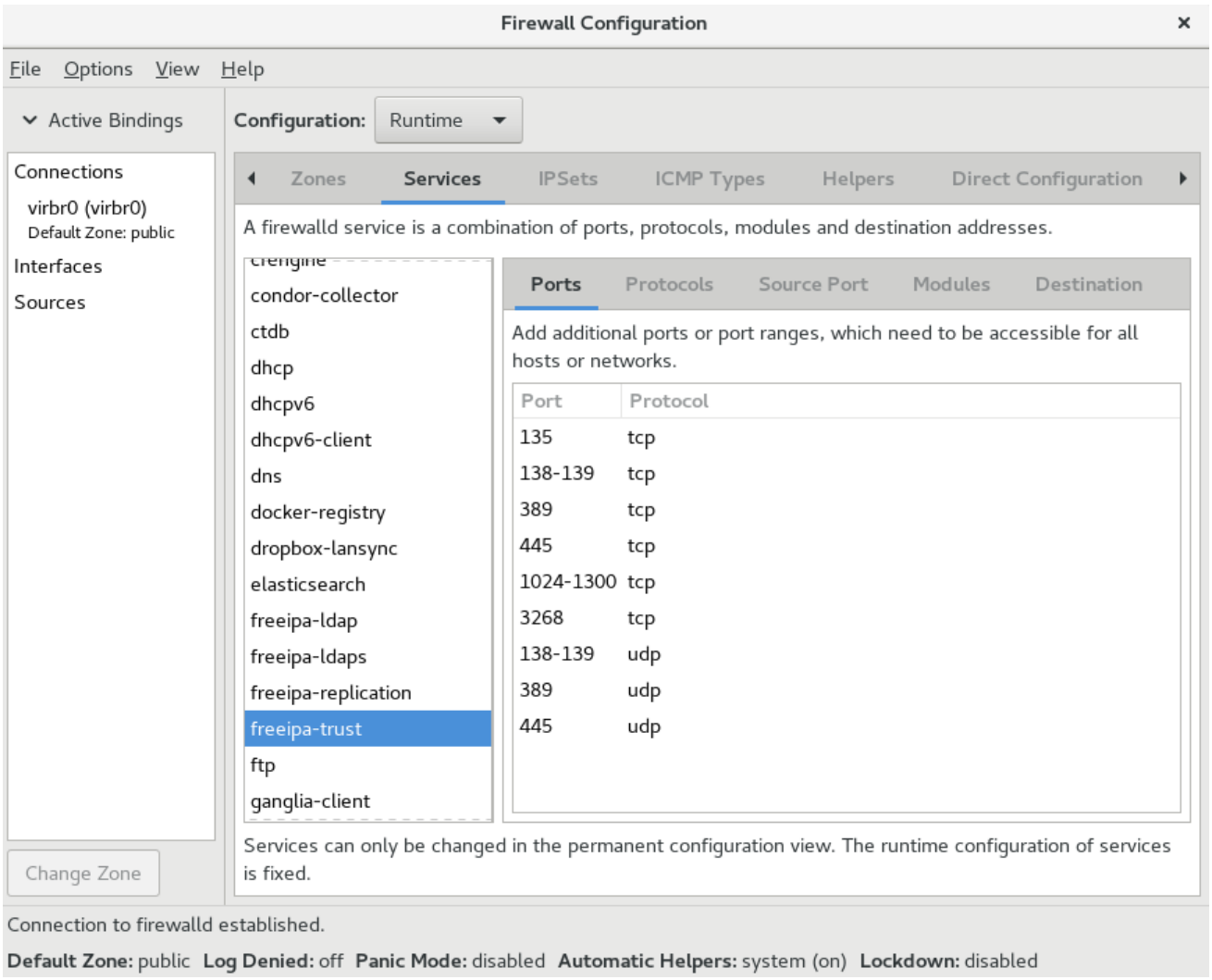
要使用图形化的 firewall-config 工具来查看服务列表，请按 Super 键进入到活动概览，输入 firewall，然后按 Enter 。firewall-config 工具会出现。现在，您可以在 Services 选项卡下查看服务列表。

另外，要用命令行启动图形防火墙配置工具，请输入以下命令：

```
~]$ firewall-config
```

此时会打开 Firewall Configuration 窗口。请注意，这个命令可以以普通用户身份运行，但偶尔会提示您输入管理员密码。

图 5.2. firewall-config 中的 Services 选项卡



[D]

5.3.2.2. 使用 CLI 查看 firewalld 设置

使用 CLI 客户端可能会对当前防火墙设置有不同的视图。list-all 选项显示 firewalld 设置的完整概述。

Firewalld 使用区来管理流量。如果没有通过 --zone 选项指定区，则该命令在分配给活跃网络接口和连接的默认区中有效。

要列出默认区的所有相关信息：

```
~]# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
```

```
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

注意

要指定显示其设置的区，请在 firewall-cmd --list-all 命令中添加 --zone=zone-name 参数，例如：

```
~]# firewall-cmd --list-all --zone=home
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
... [output truncated]
```

要查看特定信息（如服务或端口）的设置，请使用特定选项。使用命令帮助来查看 firewalld 手册页或获取选项列表：

```
~]# firewall-cmd --help

Usage: firewall-cmd [OPTIONS...]

General Options
  -h, --help          Prints a short help text and exists
  -V, --version       Print the version string of firewalld
  -q, --quiet         Do not print status messages

Status Options
  --state             Return and print firewalld state
  --reload            Reload firewall and keep state information
... [output truncated]
```

例如：查看当前区中允许哪些服务：

```
~]# firewall-cmd --list-services
ssh dhcpv6-client
```

使用 CLI 工具列出某个子部分的设置有时会比较困难。例如，允许 SSH 服务，firewalld 为该服务打

开所需的端口(22)。之后，如果您列出允许的服务，列表会显示 SSH 服务，但如果列出开放的端口，则不会显示任何内容。因此，建议您使用 --list-all 选项来确保您收到完整的信息。

5.4. 启动 FIREWALLD

要启动 firewalld，请以 root 用户身份输入以下命令：

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

要确保 firewalld 在系统启动时自动启动，请以 root 用户身份输入以下命令：

```
~]# systemctl enable firewalld
```

5.5. 停止 FIREWALLD

要停止 firewalld，请以 root 用户身份输入以下命令：

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

要防止 firewalld 在系统启动时自动启动，请以 root 用户身份输入以下命令：

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

要通过访问 firewalld D-Bus 接口以及其他服务需要 firewalld 来确保 firewalld 没有启动，请以 root 用户身份输入以下命令：

```
~]# systemctl mask firewalld
```

5.6. 控制流量

5.6.1. 预定义的服务

可使用图形化的 firewall-config 工具、firewall-cmd 和 firewall-offline-cmd 来添加和删除服务。

或者，您可以编辑 /etc/firewalld/services/ 目录中的 XML 文件。如果用户没有添加或更改服务，则在 /etc/firewalld/services/ 中找不到相应的 XML 文件。如果要添加或更改服务，则

/usr/lib/firewalld/services/ 目录中的文件可用作模板。

5.6.2. 使用 CLI 禁用发生时的所有流量

在紧急情况下，如系统攻击，可以禁用所有网络流量并关闭攻击者。

要立即禁用网络流量，请切换 panic 模式：

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

关闭 panic 模式会使防火墙恢复到其永久设置。关闭 panic 模式：

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

要查看是否打开或关闭 panic 模式，请使用：

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

5.6.3. 使用 CLI 使用预定义的服务控制流量

控制流量的最简单的方法是在 firewalld 中添加预定义的服务。这会打开所有必需的端口并根据 服务定义文件 修改其他设置。

1. 检查该服务是否还未被允许：

```
~]# firewall-cmd --list-services
ssh dhcpv6-client
```

2. 列出所有预定义的服务：

```
~]# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bitcoin bitcoin-rpc
bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6
```

```
dhcpv6-client dns docker-registry ...
[output truncated]
```

3. 在允许的服务中添加服务：

```
~]# firewall-cmd --add-service=<service-name>
```

4. 使新设置具有持久性：

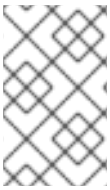
```
~]# firewall-cmd --runtime-to-permanent
```

5.6.4. 使用 GUI 使用预定义服务控制流量

要启用或禁用预定义或自定义服务，请启动 **firewall-config** 工具并选择要配置其服务的网络区。选择 **Services** 选项卡，再选中您要信任的每种服务类型的复选框。清除复选框以阻止服务。

要编辑服务，请启动 **firewall-config** 工具，然后从标记为 **Configuration** 的菜单中选择 **Permanent**。其它图标和菜单按钮会出现在服务窗口底部。选择您要配置的服务。

Ports、**Protocols** 和 **Source Port** 选项卡支持添加、更改和删除所选服务的端口、协议和源端口。模块选项卡是用于配置 **Netfilter** 助手模块的。**Destination** 选项卡允许将流量限制到特定的目标地址和互联网协议（**IPv4** 或 **IPv6**）。



注意

在 **Runtime** 模式下无法更改服务设置。

5.6.5. 添加新服务

可使用图形化的 **firewall-config** 工具、**firewall-cmd** 和 **firewall-offline-cmd** 来添加和删除服务。或者，您可以编辑 **/etc/firewalld/services/** 中的 **XML** 文件。如果用户没有添加或更改服务，那么 **/etc/firewalld/services/** 中没有相应的 **XML** 文件。如果要添加或更改服务，则可以使用 **/usr/lib/firewalld/services/** 文件作为模板。

要在终端中添加新服务，请在未激活 **firewalld** 的情况下使用 **firewall-cmd** 或 **firewall-offline-cmd**。输入以下命令来添加新的和空服务：

```
~]$ firewall-cmd --new-service=service-name
```

要使用本地文件添加新服务，请使用以下命令：

```
~]$ firewall-cmd --new-service-from-file=service-name.xml
```

您可以使用额外的 **--name=service-name** 选项来更改服务名称。

更改服务设置后，服务的更新副本放在 **/etc/firewalld/services/** 中。

作为 **root** 用户，您可以输入以下命令来手动复制服务：

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

firewalld 首先从 **/usr/lib/firewalld/services** 加载文件。如果文件放在 **/etc/firewalld/services** 中，并且有效，则这些文件将覆盖 **/usr/lib/firewalld/services** 中的匹配文件。在 **/etc/firewalld/services** 中的匹配文件已被删除，或者要求 **firewalld** 加载服务的默认值时，就会使用 **/usr/lib/firewalld/services** 中的 **overriden** 文件。这只适用于永久性环境。要在运行时环境中获取这些回退，则需要重新载入。

5.6.6. 使用 CLI 控制端口

端口是可让操作系统接收和区分网络流量并将其转发到系统服务的逻辑设备。它们通常由侦听端口的守护进程来表示，它会等待到达这个端口的任何流量。

通常，系统服务侦听为它们保留的标准端口。例如，**httpd** 守护进程监听 **80** 端口。但默认情况下，系统管理员会将守护进程配置为在不同端口上侦听以便增强安全性或出于其他原因。

打开端口

通过打开端口，系统可从外部访问，这代表了安全风险。通常，让端口保持关闭，且只在某些服务需要时才打开。

要获得当前区的打开端口列表：

1. 列出所有允许的端口：

```
~]# firewall-cmd --list-ports
```

2. 在允许的端口中添加一个端口，以便为入站流量打开这个端口：

```
~]# firewall-cmd --add-port=port-number/port-type
```

3. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

端口类型为 **tcp**、**udp**、**sctp** 或 **dccp**。这个类型必须与网络通信的类型匹配。

关闭端口

当打开的端口不再需要时，在 **firewalld** 中关闭此端口。强烈建议您尽快关闭所有不必要的端口，因为端口处于打开状态会存在安全隐患。

要关闭某个端口，请将其从允许的端口列表中删除：

1. 列出所有允许的端口：

```
~]# firewall-cmd --list-ports
[WARNING]
====
This command will only give you a list of ports that have been opened as ports. You will not be able to see any open ports that have been opened as a service. Therefore, you should consider using the --list-all option instead of --list-ports.
====
```

2. 从允许的端口中删除端口，以便对传入的流量关闭：

```
~]# firewall-cmd --remove-port=port-number/port-type
```

3. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

5.6.7. 使用 GUI 打开端口

要允许流量通过防火墙到某个端口，请启动 firewall-config 工具并选择您要更改的网络区。选择 Ports 选项卡，然后点击右侧的 Add 按钮。此时会打开 端口和协议 窗口。

输入要允许的端口号或者端口范围。从列表中选择 tcp 或 udp。

5.6.8. 使用 GUI 控制协议的流量

要允许使用特定协议通过防火墙的流量，请启动 firewall-config 工具并选择您要更改的网络区。选择 Protocols 选项卡，然后点击右侧的 Add 按钮。此时会打开 协议 窗口。

从列表中选择协议，或者选择 Other Protocol 复选框，并在字段中输入协议。

5.6.9. 使用 GUI 打开源端口

要允许来自某个端口的流量通过防火墙，请启动 firewall-config 工具并选择您要更改的网络区。选择 Source Port 选项卡，然后点击右侧的 Add 按钮。源端口 窗口将打开。

输入要允许的端口号或者端口范围。从列表中选择 tcp 或 udp。

5.7. 使用区域

zones 代表一种更透明管理传入流量的概念。这些区域连接到联网接口或者分配一系列源地址。您可以独立为每个区管理防火墙规则，这样就可以定义复杂的防火墙设置并将其应用到流量。

5.7.1. 列出区域

查看系统中有哪些可用区：

```
~]# firewall-cmd --get-zones
```

firewall-cmd --get-zones 命令显示系统上所有可用的区，但不显示特定区的详情。

查看所有区的详细信息：

```
~]# firewall-cmd --list-all-zones
```

查看特定区的详细信息：

```
~]# firewall-cmd --zone=zone-name --list-all
```

5.7.2. 为 Certain Zone 修改 firewalld 设置

第 5.6.3 节 “使用 CLI 使用预定义的服务控制流量” 和 第 5.6.6 节 “使用 CLI 控制端口” 解释了如何在当前工作区范围内添加服务或修改端口。有时，需要在不同区内设置规则。

要在其他区域中工作，请使用 **--zone=zone-name** 选项。例如，要允许在区 **public** 中使用 **SSH** 服务：

```
~]# firewall-cmd --add-service=ssh --zone=public
```

5.7.3. 更改默认区域

系统管理员在其配置文件中为网络接口分配区域。如果接口没有被分配给指定区，它将被分配给默认区。每次重启 **firewalld** 服务后，**firewalld** 会加载默认区的设置，并使其处于活动状态。

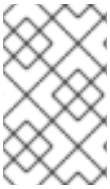
设置默认区：

1. 显示当前的默认区：

   ```
   ~]# firewall-cmd --get-default-zone
   ```

2. 设置新的默认区：

   ```
   ~]# firewall-cmd --set-default-zone zone-name
   ```



注意

按照此流程，设置是一个永久设置，即使没有 **--permanent** 选项。

5.7.4. 将网络接口分配给区

可以为不同区定义不同的规则集，然后通过更改所使用的接口的区来快速改变设置。使用多个接口，可以为每个具体区设置一个区来区分通过它们的网络流量。

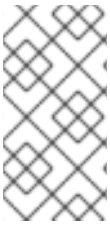
要将区分配给特定的接口：

1. 列出活跃区以及分配给它们的接口：

```
~]# firewall-cmd --get-active-zones
```

2. 为不同的区分配接口：

```
~]# firewall-cmd --zone=zone-name --change-interface=<interface-name>
```



注意

您不必使用 **--permanent** 选项在重启后保留设置。如果您设置了新的默认区，则设置将变为 **permanent**。

5.7.5. 将默认区域分配给网络连接

当连接由 **NetworkManager** 管理时，必须了解它使用的区。为每个网络连接指定区域，根据计算机有可移植设备的位置提供各种防火墙设置的灵活性。因此，可以为不同的位置（如公司或家）指定区域和设置。

要为互联网连接设置默认区，请使用 **NetworkManager GUI** 或编辑 **/etc/sysconfig/network-scripts/ifcfg-connection-name** 文件，并添加一个将区分配给此连接的行：

```
ZONE=zone-name
```

5.7.6. 创建新区域

要使用自定义区，创建一个新的区并使用它像预定义区一样。

注意

新区需要 **--permanent** 选项，否则命令无法工作。

1. 创建一个新区：

```
~]# firewall-cmd --permanent --new-zone=zone-name
```

2. 重新载入新区：

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

3. 检查是否在您的永久设置中添加了新的区：

```
~]# firewall-cmd --get-zones
```

4. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

5.7.7. 使用配置文件创建新区域

区也可以通过区配置文件创建。如果您需要创建新区，但想从不同区重复使用设置，这种方法就很有用了。

firewalld 区配置文件包含区的信息。这些区描述、服务、端口、协议、**icmp-blocks**、**masquerade**、**forward-ports** 和丰富的语言规则采用 **XML** 文件格式。文件名必须是 **zone-name.xml**，其中 **zone-name** 的长度限制为 **17** 个字符。区域配置文件位于 **/usr/lib/firewalld/zones/** 和 **/etc/firewalld/zones/** 目录中。

以下示例显示了允许 **TCP** 和 **UDP** 协议的一个服务(**SSH**)和一个端口范围的配置：

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>My zone</short>
  <description>Here you can describe the characteristic features of the zone.</description>
  <service name="ssh"/>
  <port port="1025-65535" protocol="tcp"/>
  <port port="1025-65535" protocol="udp"/>
</zone>
```

要更改那个区的设置，请添加或者删除相关的部分来添加端口、转发端口、服务等等。如需更多信息，请参阅 **firewalld.zone** 手册页。

5.7.8. 使用区目标为 Incoming 流量设置默认行为

对于每个区，您可以设置一种处理尚未进一步指定的传入流量的默认行为。这种行为是通过设置区目标来定义的。有三个选项 - 默认、**ACCEPT**、**REJECT** 和 **DROP**。通过将目标设置为 **ACCEPT**，您可以接受所有传入的数据包，除了特定规则禁用的那些数据包。如果将目标设置为 **REJECT** 或 **DROP**，您将禁用所有传入的数据包，除了您在特定规则中允许的数据包。拒绝数据包时，会通知源机器，但丢弃数据包时不会发送任何信息。

为区设置目标：

1. 列出特定区的信息以查看默认目标：

```
~]$ firewall-cmd --zone=zone-name --list-all
```

2. 在区中设置一个新目标：

```
~]# firewall-cmd --zone=zone-name --set-target=<default|ACCEPT|REJECT|DROP>
```

5.8. 使用区域管理流量取决于源

您可以使用区管理传入的流量，根据其源管理传入的流量。这可让您对进入的流量进行排序，并将其路由到不同的区，以允许或禁止该流量可访问的服务。

如果您给区添加一个源，区就会成为活跃的，来自该源的所有进入流量都会被定向到它。您可以为每个区指定不同的设置，这些设置相应地应用于来自给定源的网络流量。即使只有一个网络接口，您可以使用更多区域。

5.8.1. 添加源

要将传入的流量路由到特定源，请将源添加到那个区。源可以是 CIDR 格式的 IP 地址或 IP 掩码。

1. 在当前区中设置源：

```
~]# firewall-cmd --add-source=<source>
```

2. 要为特定区设置源 IP 地址：

```
~]# firewall-cmd --zone=zone-name --add-source=<source>
```

以下流程允许来自 受信任 区中 192.168.2.15 的所有传入的流量：

1. 列出所有可用区：

```
~]# firewall-cmd --get-zones
```

2. 将源 IP 添加到持久性模式的信任区中：

```
~]# firewall-cmd --zone=trusted --add-source=192.168.2.15
```

3. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

5.8.2. 删除源

从区中删除源会关闭来自它的网络流量。

1. 列出所需区的允许源：

```
~]# firewall-cmd --zone=zone-name --list-sources
```

2. 从区永久删除源：

```
~]# firewall-cmd --zone=zone-name --remove-source=<source>
```

3. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

5.8.3. 添加源端口

要启用根据原始端口对流量进行排序，请使用 **--add-source-port** 选项指定源端口。您还可以将此与 **--add-source** 选项结合使用，将流量限制在特定的 IP 地址或 IP 范围。

添加源端口：

```
~]# firewall-cmd --zone=zone-name --add-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

5.8.4. 删除源端口

通过删除源端口，您可以根据原始端口禁用对流量排序。

要删除源端口：

```
~]# firewall-cmd --zone=zone-name --remove-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

5.8.5. 使用 Zones 和 Sources 只允许服务只用于特定域

要允许特定网络的流量在机器上使用服务，请使用区和源。以下流程只允许来自 192.0.2.0/24 网络的 HTTP 流量，而阻止其他任何流量。

警告

当您配置此场景时，请使用具有 default 目标的区。使用目标设为 ACCEPT 的区存在安全风险，因为对于来自 192.0.2.0/24 的流量，所有网络连接都将被接受。

1. **列出所有可用区：**

```
~]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

2. **将 IP 范围添加到 internal 区，来将来自源的流量通过区：**

```
~]# firewall-cmd --zone=internal --add-source=192.0.2.0/24
```

3. **在 internal 区中添加 http 服务：**

```
~]# firewall-cmd --zone=internal --add-service=http
```

4. **使新设置具有持久性：**

```
~]# firewall-cmd --runtime-to-permanent
```

5. **检查 internal 区是否活跃，以及该区中服务是否被允许：**

```
~]# firewall-cmd --zone=internal --list-all
internal (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources: 192.0.2.0/24
  services: dhcpv6-client mdns samba-client ssh http
  ...
```

5.8.6. 配置基于区接受的流量

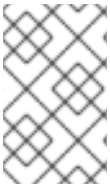
您可以根据协议允许区接受传入的流量。所有使用指定协议的流量都会被区接受，您可以在其中应用进一步的规则和过滤。

在区中添加协议

通过在某个区中添加协议，您可以允许这个区接受使用这个协议的所有流量。

在区中添加协议：

```
~]# firewall-cmd --zone=zone-name --add-protocol=port-name/tcp|udp|sctp|dccp|igmp
```



注意

要接收多播流量，请使用带有 **--add-protocol** 选项的 **igmp** 值。

从区中删除协议

从某个区中删除协议，您可以停止接受区基于这个协议的所有流量。

从区中删除协议：

```
~]# firewall-cmd --zone=zone-name --remove-protocol=port-name/tcp|udp|sctp|dccp|igmp
```

5.9. 端口转发

使用 **firewalld**，您可以设置端口重定向，以便到达系统上某个端口的任何传入的流量都被传送到您选择的其他内部端口或另一台计算机上的外部端口。

5.9.1. 向重定向添加端口

在您将一个端口的流量重定向到另一个端口或另一个地址之前，您需要了解 **3** 个问题：数据包到达哪个端口、使用哪个协议，以及您要重定向它们的位置。

将端口重新指向另一个端口：

```
~]# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp|sctp|dccp:toport=port-number
```

将端口重定向到不同 IP 地址的另一个端口：

1. **添加要转发的端口：**

```
~]# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp:toport=port-number:toaddr=IP
```

2. **启用伪装：**

```
~]# firewall-cmd --add-masquerade
```

例 5.1. 将 TCP 端口 80 重定向到同一计算机上的端口 88

重定向端口：

1. **将端口 80 重定向到 TCP 流量的端口 88:**

```
~]# firewall-cmd --add-forward-port=port=80:proto=tcp:toport=88
```

2. **使新设置具有持久性：**

```
~]# firewall-cmd --runtime-to-permanent
```

3. **检查是否重定向了端口：**

```
~]# firewall-cmd --list-all
```

5.9.2. 删除重定向的端口

要删除重定向的端口：

```
~]# firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
```

要删除重定向到不同地址的转发端口：

1. **删除转发的端口：**

```
~]# firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
```

2. **禁用伪装：**

```
~]# firewall-cmd --remove-masquerade
```

注意

使用此方法重定向端口只可用于基于 IPv4 的流量。对于 IPv6 重定向设置，您需要使用丰富的规则。如需更多信息，请参阅 第 5.15 节 “使用"Rich Language"语法配置复杂防火墙规则”。

要重定向到外部系统，需要启用伪装。如需更多信息，请参阅 第 5.10 节 “配置 IP 地址伪装”。

例 5.2. 删除在同一机器上将 TCP 端口 80 转发到端口 88

删除端口重定向：

1. **列出重定向的端口：**

```
~]# firewall-cmd --list-forward-ports
port=80:proto=tcp:toport=88:toaddr=
```

2. **从防火墙中删除重定向的端口：**

```
~]# firewall-cmd --remove-forward-port=port=80:proto=tcp:toport=88:toaddr=
```

3. **使新设置具有持久性：**

```
~]# firewall-cmd --runtime-to-permanent
```

5.10. 配置 IP 地址伪装

IP 伪装是一台计算机充当网络的 IP 网关的进程。对于伪装，网关会始终动态查找传出接口的 IP，并将数据包中的源地址替换为这个地址。

如果传出接口的 IP 可以更改，您可以使用伪装。伪装的典型用例是，如果路由器将没有在互联网上路由的专用 IP 地址替换为路由器上传出接口的公共动态 IP 地址。

要检查是否启用了 IP 伪装（例如，对于 external 区），以 root 用户身份输入以下命令：

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

如果已启用，命令将会打印 yes，且退出状态为 0。否则，将打印 no，且退出状态为 1。如果省略了 zone，则将使用默认区。

要启用 IP 伪装，请以 root 用户身份输入以下命令：

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

要使此设置持久，请重复添加 **--permanent** 选项的命令。

要禁用 IP 伪装，请以 root 身份输入以下命令：

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

要使此设置持久，请重复添加 **--permanent** 选项的命令。

如需更多信息，请参阅：

- 第 6.3.1 节 “不同的 NAT 类型： masquerading、source NAT、destination NAT 和 redirect”

- 第 6.3.2 节 “使用 nftables 配置伪装”

5.11. 管理 ICMP 请求

Internet 控制消息协议 (ICMP)是一种支持协议，供各种网络设备用来发送错误消息和表示连接问题的操作信息，例如，请求的服务不可用。ICMP 与 TCP 和 UDP 等传输协议不同，因为它不用于在系统之间交换数据。

不幸的是，可以使用 ICMP 消息（特别是 echo-request 和 echo-reply ）来揭示关于您网络的信息，并将这些信息滥用于各种欺诈活动。因此，firewalld 允许阻止 ICMP 请求，来保护您的网络信息。

5.11.1. 列出 ICMP 请求

ICMP 请求在 /usr/lib/firewalld/icmptypes/ 目录中的单个 XML 文件中进行了描述。您可以阅读这些文件来查看请求的描述。firewall-cmd 命令控制 ICMP 请求操作。

要列出所有可用的 ICMP 类型：

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

IPv4、IPv6 或两个协议都可以使用 ICMP 请求。要查看使用 ICMP 请求的协议：

```
~]# firewall-cmd --info-icmptype=<icmptype>
```

如果请求当前被阻止，则 ICMP 请求显示 yes 或如果请求没被阻止，则显示 no。要查看 ICMP 请求当前是否被阻止：

```
~]# firewall-cmd --query-icmp-block=<icmptype>
```

5.11.2. 阻止或取消阻止 ICMP 请求

当您的服务器阻止了 ICMP 请求时，它不会提供通常应该提供的信息。但这并不意味着根本不给出任何信息。客户端会收到特定 ICMP 请求被阻止（拒绝）的信息。应仔细考虑阻止 ICMP 请求，因为它可能会造成通信问题，特别是 IPv6 流量。

要查看 ICMP 请求当前是否被阻止：

```
~]# firewall-cmd --query-icmp-block=<icmptype>
```

要阻止 ICMP 请求：

```
~]# firewall-cmd --add-icmp-block=<icmptype>
```

要删除 ICMP 请求的块：

```
~]# firewall-cmd --remove-icmp-block=<icmptype>
```

5.11.3. 在没有任何信息的情况下阻止 ICMP 请求

通常，如果您阻止了 ICMP 请求，客户端会知道您正在阻止它。这样潜在的攻击者仍然可以看到您的 IP 地址在线。要完全隐藏此信息，您必须丢弃所有 ICMP 请求。

要阻止和丢弃所有 ICMP 请求：

1. 将区的目标设为 DROP ：

   ```
   ~]# firewall-cmd --set-target=DROP
   ```

2. 使新设置具有持久性：

   ```
   ~]# firewall-cmd --runtime-to-permanent
   ```

现在，所有流量（包括 ICMP 请求）都将被丢弃，除了您明确允许的流量外。

要阻止和丢弃某些 ICMP 请求，而允许其他请求：

1. 将区的目标设为 DROP ：

```
~]# firewall-cmd --set-target=DROP
```

2. 添加 ICMP 块反转以一次性阻止所有 ICMP 请求：

```
~]# firewall-cmd --add-icmp-block-inversion
```

3. 为这些您要允许的 ICMP 请求添加 ICMP 块：

```
~]# firewall-cmd --add-icmp-block=<icmptype>
```

4. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

块反转会 反转 ICMP 请求块的设置，因此所有之前没有被阻止的请求都会被阻止。被阻止的那些不会被阻止。这意味着，如果您需要取消阻塞请求，则必须使用 blocking 命令。

把它恢复到完全 permissive 设置：

1. 将区的目标设为 default 或 ACCEPT ：

```
~]# firewall-cmd --set-target=default
```

2. 删除 ICMP 请求的所有添加的块：

```
~]# firewall-cmd --remove-icmp-block=<icmptype>
```

3. 删除 ICMP 块反转:

```
~]# firewall-cmd --remove-icmp-block-inversion
```

4. 使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

5.11.4. 使用 GUI 配置 ICMP 过滤器

要启用或禁用 ICMP 过滤器，请启动 **firewall-config** 工具，并选择其信息要被过滤的网络区。选择 **ICMP Filter** 选项卡，然后选中您要过滤的每种 ICMP 消息类型的复选框。清除复选框以禁用过滤器。这个设置按方向设置，默认允许所有操作。

要启用反转 **ICMP Filter**，请单击右侧的 **Invert Filter** 复选框。现在只接受标记的 ICMP 类型，所有其他类型都被拒绝。在使用 **DROP** 目标的区域里它们会被丢弃。

5.12. 使用 FIREWALLD设置和控制 IP 集

要查看 **firewalld** 所支持的 IP 集设置类型列表，请以 **root** 用户身份输入以下命令。

```
~]# firewall-cmd --get-ipset-types
hash:ip hash:ip,mark hash:ip,port hash:ip,port,ip hash:ip,port,net hash:mac hash:net hash:net,iface
hash:net,net hash:net,port hash:net,port,net
```

5.12.1. 使用命令行客户端配置 IP 设置选项

IP 集可以在 **firewalld** 区中用作源，也可以用作富规则中的源。在 **Red Hat Enterprise Linux 7** 中，首选的方法是使用在直接规则中使用 **firewalld** 创建的 IP 集。

要列出 **permanent** 环境中 **firewalld** 已知的 IP 集，请以 **root** 用户身份运行以下命令：

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

要添加新的 IP 集，请以 **root** 用户身份使用 **permanent** 环境来运行以下命令：

```
~]# firewall-cmd --permanent --new-ipset=test --type=hash:net
success
```

上面的命令为 **IPv4** 创建了一个名为 **test** 和 **hash:net** 类型的新 IP 集。要创建用于 **IPv6** 的 IP 集，请添加 **--option=family=inet6** 选项。要使新设置在运行时环境中有效，请重新加载 **firewalld**。以 **root** 身份使用以下命令列出新 IP 集：

```
~]# firewall-cmd --permanent --get-ipsets
test
```

要获得有关 IP 集的更多信息，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --info-ipset=test
test
type: hash:net
options:
entries:
```

请注意，IP 集目前没有任何条目。要向 test IP 集中添加一个条目，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
success
```

前面的命令将 IP 地址 192.168.0.1 添加到 IP 集合中。要获取 IP 集合中当前条目的列表，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

生成包含 IP 地址列表的文件，例如：

```
~]# cat > iplist.txt <<EOL
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
EOL
```

包含 IP 集合 IP 地址列表的文件应该每行包含一个条目。以 hash、分号或空行开头的行将被忽略。

要从 iplist.txt 文件中添加地址，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --add-entries-from-file=iplist.txt
success
```

要查看 IP 集合的扩展条目列表，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
192.168.0.2
```

```
192.168.0.3
192.168.1.0/24
192.168.2.254
```

要从 IP 集合中删除地址并检查更新的条目列表，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --remove-entries-from-file=iplist.txt
success
~]# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

您可以将 IP 集合作为一个源添加到区，以便处理所有来自 IP 集合中列出的任意地址的网络流量。例如，要将 test IP 集作为源添加到 drop 区来丢弃来自 test IP 集中列出的所有条目的所有数据包，请以 root 身份运行以下命令：

```
~]# firewall-cmd --permanent --zone=drop --add-source=ipset:test
success
```

源中的 ipset: 前缀向 firewalld 表示，源是一个 IP 集，而不是一个 IP 地址或一个地址范围。

创建和删除 IP 集合仅限于永久环境，其它 IP 设置选项也可以用于运行时环境中，而无需 --permanent 选项。

5.12.2. 为 IP 集配置自定义服务

将自定义服务配置为在 firewalld 启动前创建和加载 IP 设置结构：

1. 以 root 用户身份运行的编辑器，按如下所示创建一个文件：

```
~]# vi /etc/systemd/system/ipset_name.service
[Unit]
Description=ipset_name
Before=firewalld.service

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/bin/ipset_name.sh start
ExecStop=/usr/local/bin/ipset_name.sh stop

[Install]
WantedBy=basic.target
```

2.

在 firewalld 中永久使用设置的 IP：

```
~]# vi /etc/firewalld/direct.xml
<?xml version="1.0" encoding="utf-8"?>
<direct>
 <rule ipv="ipv4" table="filter" chain="INPUT" priority="0">-m set --match-set
<replaceable>ipset_name</replaceable> src -j DROP</rule>
</direct>
```

3.

需要 firewalld 重新加载才能激活更改：

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

这会重新加载防火墙，而不丢失状态信息(TCP 会话不会终止)，但在重新加载期间可能会中断服务。

警告

红帽不推荐使用不是通过 firewalld 管理的 IP 集。要使用这样的 IP 组，需要一个永久直接规则来引用集合，且必须添加自定义服务来创建这些 IP 组件。这个服务需要在 firewalld 启动前启动，否则 firewalld 无法使用这些集合添加直接规则。您可以使用 /etc/firewalld/direct.xml 文件来添加永久的直接规则。

5.13. 使用 IPTABLES设置和控制 IP 集

firewalld 和 iptables （和 ip6tables）服务之间的基本区别是：

- iptables 服务将 配置存储在 /etc/sysconfig/iptables 和 /etc/sysconfig/ip6tables 中，而 firewalld 将其存储在 /usr/lib/firewalld/ 和 / etc/firewalld/ 的不同 XML 文件中。请注意，/etc/sysconfig/iptables 文件不存在，因为默认情况下在 Red Hat Enterprise Linux 中安装 firewalld。

- 使用 iptables 服务 时，每个更改都意味着清除所有旧规则，并从 /etc/sysconfig/iptables 读取所有新规则，而 firewalld 不会重新创建所有规则。仅应用不同之处。因此，firewalld 可以在运行时更改设置，而不会丢失现有连接。

两者都使用 iptables 工具 与内核数据包过滤。

要使用 iptables 和 ip6tables 服务而不是 firewalld，首先以 root 用户身份运行以下命令来禁用 firewalld ：

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

然后以 root 用户身份输入以下命令安装 iptables-services 软件包：

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

iptables-services 软件包包含 iptables 服务和 ip6tables 服务。

然后，要启动 iptables 和 ip6tables 服务，请以 root 用户身份输入以下命令：

```
~]# systemctl start iptables
~]# systemctl start ip6tables
```

要启用服务在每次系统启动时启动，请输入以下命令：

```
~]# systemctl enable iptables
~]# systemctl enable ip6tables
```

ipset 工具用于管理 Linux 内核中的 IP 集。IP 集是用于存储 IP 地址、端口号、IP 和 MAC 地址对的框架，或者 IP 地址和端口号对。集合的索引方式可以针对集合进行快速匹配，即使集合非常大。IP 集启用更简单且更易于管理的配置，以及使用 iptables 时提供性能优势。iptables 匹配和目标，创建保护内核中给定集合的引用。当存在指向它的单一引用时，无法销毁集合。

使用 ipset 可启用 iptables 命令（如下面的命令）被一个集合替代：

-

```
~]# iptables -A INPUT -s 10.0.0.0/8 -j DROP
~]# iptables -A INPUT -s 172.16.0.0/12 -j DROP
~]# iptables -A INPUT -s 192.168.0.0/16 -j DROP
```

该集合创建如下：

```
~]# ipset create my-block-set hash:net
~]# ipset add my-block-set 10.0.0.0/8
~]# ipset add my-block-set 172.16.0.0/12
~]# ipset add my-block-set 192.168.0.0/16
```

然后，在 iptables 命令中引用该集合，如下所示：

```
~]# iptables -A INPUT -m set --set my-block-set src -j DROP
```

如果设置被多次使用，则进行保存配置时间。如果集合包含多个在处理时间保存的条目。

5.14. 使用直接接口

通过将 --direct 选项与 firewall-cmd 工具一起使用，可以在运行时添加和删除链。这里提供了几个示例。详情请查看 firewall-cmd (1) 手册页。

如果您不非常熟悉 iptables，则使用直接接口是危险的，因为您可能会意外导致防火墙出现问题。

直接接口模式用于服务或应用程序，以便在运行时添加特定的防火墙规则。可以使用 firewall-cmd --permanent --direct 命令或修改 /etc/firewalld/direct.xml 添加 --permanent 选项，使规则永久生效。有关 /etc/firewalld/direct.xml 文件的信息，请参阅 man firewalld.direct (5)。

5.14.1. 使用直接接口添加规则

要为 “IN_public_allow” 链添加规则，请以 root 用户身份输入以下命令：

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

添加 --permanent 选项，使设置持久。

5.14.2. 使用直接接口删除规则

要从 “IN_public_allow” 链中删除规则，请以 root 用户身份输入以下命令：

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

添加 **--permanent** 选项，使设置持久。

5.14.3. 使用直接接口列出规则

要列出 “IN_public_allow” 链中的规则，请以 root 用户身份输入以下命令：

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

请注意，这个命令(**--get-rules** 选项)只列出之前使用 **--add-rule** 选项添加的规则。它不会列出其他方法添加的现有 **iptables** 规则。

5.15. 使用"RICH LANGUAGE"语法配置复杂防火墙规则

通过 “丰富的语言” 语法，可以创建复杂的防火墙规则，比直接接口方法更容易理解。此外，可以永久设置。语言使用带有值的关键字，是 **iptables** 规则的一个抽象表示。可以使用此语言配置区域；当前配置方法仍被支持。

5.15.1. Rich Language 命令的格式

本节中的所有命令都需要以 root 用户身份运行。添加规则的命令格式如下：

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

这将为区域 区域 添加丰富的语言 规则。这个选项可多次指定。如果省略了区，则使用默认区。如果提供了超时，规则或规则只在指定的时间内保持活动状态，之后会自动删除。时间值可以跟随 **s** （秒）、 **m** （分钟）或 **h** （小时）来指定时间单位。默认值为 秒。

删除规则：

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

这将删除区域 区域 的丰富语言 规则。这个选项可多次指定。如果省略了区，则使用默认区。

检查是否存在规则：

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

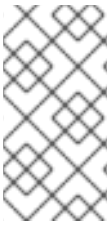
这将返回是否为区域 区域 添加丰富的语言 规则规则。如果已启用，命令将会打印 yes，且退出状态为 0。否则，将打印 no，且退出状态为 1。如果省略了区，则使用默认区。

有关区配置文件中使用的丰富的语言表示的详情，请查看 firewalld.zone(5) man page。

5.15.2. 了解 Rich Rule 结构

rich rule 命令的格式或结构如下：

```
rule [family="rule family"]
  [ source [NOT] [address="address"] [mac="mac-address"] [ipset="ipset"] ]
  [ destination [NOT] address="address" ]
  [ element ]
  [ log [prefix="prefix text"] [level="log level"] [limit value="rate/duration"] ]
  [ audit ]
  [ action ]
```



注意

文件中富规则的结构使用 NOT 关键字来反转源地址命令的含义，但命令行使用 invert="true" 选项。

规则与特定区域关联。个区域可以有多个规则。如果某些规则交互或字典，则应用第一个与数据包匹配的规则。

5.15.3. 了解 Rich Rule 命令选项

系列

如果提供了规则系列，可以是 ipv4 或 ipv6，它将规则分别限制为 IPv4 或 IPv6。如果没有提供规则系列，则会为 IPv4 和 IPv6 添加该规则。如果在规则中使用源或目标地址，则需要提供规则系

列。端口转发也是端口转发的情况。

源和目标地址

source

通过指定源地址，连接尝试的来源可以限制为源地址。源地址或地址范围是 IP 地址或网络 IP 地址，其掩码为 IPv4 或 IPv6。对于 IPv4，掩码可以是网络掩码或纯文本。对于 IPv6，掩码是纯数字。不支持使用主机名。通过添加 NOT 关键字；除提供的地址匹配，可以反转源 address 命令的意义。

如果该规则没有指定 系列，则可以为 IPv4 和 IPv6 添加类型为 hash:mac 的 MAC 地址和 IP 集。其他 IP 集需要与规则的 family 设置匹配。

目的地

通过指定目标地址，目标可以限制为目标地址。目标地址使用与 IP 地址或地址范围的源地址相同的语法。源和目标地址的使用是可选的，所有元素都不能使用目标地址。这取决于目标地址的使用，例如在服务条目中。您可以组合 目的地和操作 。

元素

该元素 只能是以下元素类型之一 ： service,port,protocol,masquerade,icmp-block,forward-port, 和 source-port。

service

service 元素是 firewalld 提供的服务之一。要获取预定义服务列表，请输入以下命令：

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

如果服务提供目标地址，它将与规则中的目标地址冲突，并会导致错误。在内部使用目标地址的服务主要是使用多播的服务。该命令采用以下格式：

```
service name=service_name
```

port

port 元素可以是单个端口号或端口范围，例如 5060-5062，后跟协议，可以是 tcp 或 udp。该命令采用以下格式：

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

protocol

protocol 值可以是协议 ID 号或协议名称。有关允许 的协议 条目，请参阅 /etc/protocols。该命令采用以下格式：

```
protocol value=protocol_name_or_ID
```

icmp-block

使用此命令阻止一个或多个 ICMP 类型。ICMP 类型是 firewalld 支持的 ICMP 类型之一。要获取支持的 ICMP 类型列表，请输入以下命令：

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

此处不允许指定操作。ICMP-block 在内部使用操作 拒绝。该命令采用以下格式：

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

masquerade

在规则中打开 IP 伪装。可以提供源地址以限制伪装到此区域，但不能提供目标地址。此处不允许指定操作。

forward-port

使用指定为 tcp 或 udp 的协议从本地端口转发数据包到本地的其它端口，或转发到另一台计算机上的其他端口。port 和 to-port 可以是单个端口号或端口范围。目标地址是一个简单的 IP 地址。此处不允许指定操作。forward-port 命令使用操作在内部 接受。该命令采用以下格式：

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

source-port

匹配数据包的源端口 - 连接尝试的源端口。要匹配当前计算机上的端口，请使用 port 元素。source-port 元素可以是单个端口号或端口范围（例如 5060-5062），后跟协议为 tcp 或 udp。该命令采用以下格式：

```
source-port port=number_or_range protocol=protocol
```

日志记录

log

使用内核日志记录记录新连接尝试规则，例如在 **syslog** 中。您可以定义将作为前缀添加到日志消息中的前缀文本。日志级别可以是 **emerg**、**alert**、**crit**、**error**、**warning**、**notice**、**info** 或 **debug** 之一。使用日志是可选的。可以按如下所示限制日志记录：

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

速率是一个自然正数 [1, ..]，持续时间为 **s,m,h,d**。**s** 表示秒，**m** 表示分钟，**h** 表示小时和 **d** 天。最大限制值为 **1/d**，这表示每天最多一个日志条目。

audit

Audit 提供了使用发送到 **service auditd** 的审计记录的日志的替代方法。**audit** 类型可以是 **ACCEPT**、**REJECT** 或 **DROP** 之一，但在命令 审计 后未指定，因为审计类型将从规则操作中自动收集。审计没有自己的参数，但可以选择性地添加限制。审计的使用是可选的。

操作

accept|reject|drop|mark

一个操作可以是 接受 之一、拒绝、丢弃 或 标记。该规则只能包含元素或源。如果规则包含一个元素，则与该元素匹配的新连接将使用该操作进行处理。如果规则包含源，则来自源地址的所有内容都将使用指定的操作进行处理。

```
accept | reject [type=reject type] | drop | mark set="mark[/mask]"
```

使用 接受 时，将授予所有新的连接尝试。如果 拒绝，则它们的源将被拒绝，并且其源将收到拒绝消息。**reject** 类型可以设置为使用另一个值。使用 **drop** 时，所有数据包将立即丢弃，且不会向源发送任何信息。使用 标记 所有数据包时，将使用给定的 标记和可选 掩码 标记。

5.15.4. 使用 Rich Rule Log 命令

可以使用 **Netfilter** 日志目标以及 **audit** 目标来完成日志记录。向所有区域都添加了一个新链，其格式为 “**zone_log**”，其中 **zone** 是区域名称。这会在 拒绝 链之前进行处理，以便正确排序。根据规则的操作，它们的规则或部分放置在单独的链中，如下所示：

```
zone_log
  zone_deny
  zone_allow
```

所有日志记录规则都将放在 “区域_log” 链中，首先解析这些规则。所有 拒绝 和丢弃 规则将放置在 “区域_deny” 链中，这些规则将在日志链后解析。所有 接受 规则将放置在 “区域_allow” 链中，这将在 拒绝 链后解析。如果规则包含 日志 以及 拒绝或允许 操作，则指定这些操作的规则部分放置在匹配的链中。

5.15.4.1. 使用 Rich Rule Log 命令示例 1

为身份验证标头协议 AH 启用新的 IPv4 和 IPv6 连接：

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

5.15.4.2. 使用 Rich Rule Log Command Example 2

允许协议 FTP 的新 IPv4 和 IPv6 连接，并使用审计每分钟记录 1 个：

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

5.15.4.3. 使用 Rich Rule Log 命令示例 3

对于协议 TFTP 允许从地址 192.168.0.0/24 进行新的 IPv4 连接，并使用 syslog 每分钟记录 1 个：

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

5.15.4.4. 使用 Rich Rule Log Command Example 4

从 1:2:3:4:6:: 用于协议 RADIUS 的新 IPv6 连接都会被拒绝，并记录每分钟的 3 个速率。可接受来自其他源的新 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
```

5.15.4.5. 使用 Rich Rule Log 命令示例 5

在端口 4011 上将从 1:2:3:4:6:: 接收的 IPv6 数据包转发到端口 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"
```

5.15.4.6. 使用 Rich Rule Log Command Example 6

将源地址列入白名单以允许来自此源的所有连接。

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

有关更多示例，请参阅 firewalld.richlanguage (5) 手册页。

5.16. 配置防火墙锁定

如果本地应用程序或服务以 root 身份运行（例如 libvirt），则可以更改防火墙配置。使用这个特性，管理员可以锁定防火墙配置，从而达到没有应用程序或只有添加到锁定白名单中的应用程序可以请求防火墙更改的目的。锁定设置默认会被禁用。如果启用，用户就可以确定，防火墙没有被本地的应用程序或服务进行了不必要的配置更改。

5.16.1. 使用命令行客户端配置锁定

要查询是否启用了锁定，请以 root 身份运行以下命令：

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

如果启用了锁定，该命令会打印 yes，且退出状态为 0。否则，将打印 no，且退出状态为 1。

要启用锁定，请以 root 身份输入以下命令：

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

要禁用锁定，请以 root 身份使用以下命令：

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

5.16.2. 使用命令行客户端配置锁定白名单选项

锁定白名单中可以包含命令、安全上下文、用户和用户 ID。如果白名单中的命令条目以星号 "*" 结尾，则所有以该命令开头的命令行都将匹配。如果没有 "*"，则包括参数的绝对命令必须匹配。

上下文是正在运行的应用程序或服务的安全（SELinux）上下文。要获得正在运行的应用程序的上下文，请使用以下命令：

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

该命令返回所有正在运行的应用程序。通过 grep 工具将结果进行管道输出以获取感兴趣的应用程序。例如：

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

要列出白名单中的所有命令行，请以 root 用户身份输入以下命令：

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

要在白名单中添加命令 command，请以 root 用户身份输入以下命令：

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

要从白名单中删除命令 command，请以 root 用户身份输入以下命令：

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

要查询 command 命令是否在白名单中，请以 root 用户身份输入以下命令：

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

如果为 true，命令会打印 yes，且退出状态为 0。否则，将打印 no，且退出状态为 1。

要列出白名单中的所有安全上下文，请以 root 用户身份输入以下命令：

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

要在白名单中添加上下文 context，请以 root 用户身份输入以下命令：

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

要从白名单中删除上下文 context，请以 root 身份输入以下命令：

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

要查询上下文 context 是否在白名单中，请以 root 用户身份输入以下命令：

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

如果为 true，则打印 yes，退出状态为 0，否则打印 no，退出状态为 1。

要列出白名单中的所有用户 ID，请以 root 用户身份输入以下命令：

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

要在白名单中添加用户 ID uid，请以 root 用户身份输入以下命令：

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

要从白名单中删除用户 ID uid，请以 root 用户身份输入以下命令：

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

要查询用户 ID uid 是否在白名单中，请输入以下命令：

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

如果为 true，则打印 yes，退出状态为 0，否则打印 no，退出状态为 1。

要列出白名单中的所有用户名，请以 root 用户身份输入以下命令：

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

要在白名单中添加用户名 user，请以 root 用户身份输入以下命令：

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

要从白名单中删除用户名 user，请以 root 用户身份输入以下命令：

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

要查询用户名 user 是否在白名单中，请输入以下命令：

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

如果为 true，则打印 yes，退出状态为 0，否则打印 no，退出状态为 1。

5.16.3. 使用配置文件配置锁定白名单选项

默认白名单配置文件包含 NetworkManager 上下文和 libvirt 的默认上下文。用户 ID 0 也位于列表中。

```
<?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>
```

以下是一个白名单配置文件示例，为 firewall-cmd 工具启用所有命令，对于名为 user 的用户，其用户 ID 为 815：

```
<?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>
```

此示例展示了 user id 和 user name，但只需要其中一个选项。Python 是程序解释器，它位于命令行的前面。您还可以使用特定的命令，例如：

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

。在这个示例中，只允许 --lockdown-on 命令。

注意

在 Red Hat Enterprise Linux 7 中，所有工具都放在 /usr/bin/ 目录中，/bin/ 目录被符号链接到 /usr/bin/ 目录。换句话说，虽然以 root 用户身份运行 firewall-cmd 的路径可能会解析为 /bin/firewall-cmd，但现在可以使用/usr/bin/firewall-cmd。所有新脚本都应该使用新位置。但请注意，如果以 root 身份运行的脚本已被写为使用 /bin/firewall-cmd 路径，那么除了通常为非root 用户使用的 /usr/bin/firewall-cmd 路径外，还必须将该命令路径列入白名单。

命令的 name 属性末尾的 “*” 表示所有以这个字符串开头的命令都将匹配。如果没有 “*”，则包括参数的绝对命令必须匹配。

5.17. 为 DENIED PACKETS 配置日志记录

使用 firewalld 中的 LogDenied 选项，可以为拒绝的数据包添加一个简单的日志记录机制。这些是被拒绝或丢弃的数据包。要更改日志的设置，请编辑 /etc/firewalld/firewalld.conf 文件，或者使用命令行或 GUI 配置工具。

如果启用了 LogDenied，则会在 INPUT、FORWARD 和 OUTPUT 链中的 reject 和 drop 规则之前添加日志规则，以及区域中的最终拒绝和丢弃规则。此设置可能的值有：
all、unicast、broadcast、multicast 和 off。默认设置为 off。使用 单播、广播 和多播 设置时，pkttype 匹配用于匹配链路层数据包类型。使用 所有 时，所有数据包都会记录。

要使用 firewall-cmd 列出实际的 LogDenied 设置，请以 root 用户身份运行以下命令：

```
~]# firewall-cmd --get-log-denied
off
```

要更改 LogDenied 设置，请以 root 用户身份运行以下命令：

```
~]# firewall-cmd --set-log-denied=all
success
```

要使用 firewalld GUI 配置工具更改 LogDenied 设置，请启动 firewall-config，点 Options 菜单并选择 Change Log Denied。此时会出现 LogDenied 窗口。从菜单中选择新的 LogDenied 设置，然后单击 OK。

5.18. 其它资源

以下信息源提供有关 firewalld 的其他资源。

5.18.1. 安装的文档

- firewalld (1) 手册页 - 描述 firewalld 的命令选项。
- firewalld.conf (5) 手册页 - 包含用于配置 firewalld 的信息。
- firewall-cmd (1) 手册页 - 描述 firewalld 命令行客户端的命令行选项。
- firewall-config (1) 手册页 - 描述 firewall-config 工具的设置。
- firewall-offline-cmd (1) 手册页 - 描述 firewalld 离线命令行客户端的命令选项。
- firewalld.icmptype (5) 手册页 - 描述用于 ICMP 过滤的 XML 配置文件。
- firewalld.ipset (5) 手册页 - 描述 firewalld IP 集的 XML 配置文件。
- firewalld.service (5) 手册页 - 描述 firewalld 服务的 XML 配置文件。
- firewalld.zone (5) 手册页 - 描述 firewalld 区配置的 XML 配置文件。

- *firewalld.direct (5) 手册页 - 描述 firewalld 直接接口配置文件。*
- *firewalld.lockdown-whitelist (5) 手册页 - 描述 firewalld 锁定白名单配置文件。*
- *firewalld.richlanguage (5) 手册页 - 描述 firewalld 丰富的语言规则语法。*
- *firewalld.zones (5) 手册页 - 哪些区的一般描述以及如何配置它们。*
- *firewalld.dbus (5) 手册页 - 描述 firewalld 的 D-Bus 接口。*

5.18.2. 在线文档

- *http://www.firewalld.org/ - firewalld 主页。*

第 6 章 NFTABLES 入门

nftables 框架提供数据包分类工具，它是 iptables、ip6tables、arptables、ebtables 和 ipset 工具的指定成功者。与之前的数据包过滤工具相比，它在方便、特性和性能方面提供了大量改进，最重要的是：

- 内置查找表而不是线性处理
- IPv4 和 IPv6 协议的单一框架
- 规则会以一个整体被应用，而不是分为抓取、更新和存储完整的规则集的步骤
- 支持在规则集(nftrace)和监控追踪事件(nft)中调试和追踪
- 更加一致和压缩的语法，没有特定协议的扩展
- 用于第三方应用程序的 Netlink API

与 iptables 类似，nftables 使用表来存储链。链包含执行动作的独立规则。nft 工具替换了之前数据包过滤框架中的所有工具。libnftnl 库可用于通过 libmnl 库与 nftables Netlink API 进行低级交互。

要显示规则集变化的影响，请使用 nft list ruleset 命令。由于这些工具将表、链、规则、集合和其他对象添加到 nftables 规则集，请注意 nftables 规则集操作（如 nft flush ruleset 命令）可能会影响使用之前独立的旧命令安装的规则集。

何时使用 FIREWALLD 或 NFTABLES

- firewalld ：将 firewalld 工具用于简单的防火墙用例。此工具易于使用，并涵盖了这些场景的典型用例。
- nftables ：使用 nftables 工具来设置复杂和性能关键的防火墙，如用于整个网络。

重要

要避免不同的防火墙服务相互影响，在 RHEL 主机中只有一个服务，并禁用其他服务。

6.1. 编写和执行 NFTABLES 脚本

nftables 框架提供了一个原生脚本环境，它比使用 shell 脚本维护 防火墙规则 提供了主要优势：执行脚本是原子的。这意味着，系统会应用整个脚本，或者在出现错误时防止执行。这样可保证防火墙始终处于一致状态。

另外，nftables 脚本环境使管理员能够：

- 添加评论
- 定义变量
- 包含其他规则集文件

本节介绍如何使用这些功能，以及创建和执行 nftables 脚本。

当您安装 nftables 软件包时，Red Hat Enterprise Linux 会在 /etc/nftables/ 目录中自动创建 lf nft 脚本。这些脚本包含为不同目的创建表和空链的命令。

6.1.1. 支持的 nftables 脚本格式

nftables 脚本环境支持以下格式的脚本：

- 您可以以与 nft list ruleset 命令相同的格式编写脚本，显示规则集：

```
#!/usr/sbin/nft -f

# Flush the rule set
flush ruleset

table inet example_table {
```

```
  chain example_chain {
    # Chain for incoming packets that drops all packets that
    # are not explicitly allowed by any rule in this chain
    type filter hook input priority 0; policy drop;

    # Accept connections to port 22 (ssh)
    tcp dport ssh accept
  }
}
```

- **您可以使用与 nft 命令相同的语法：**

```
#!/usr/sbin/nft -f

# Flush the rule set
flush ruleset

# Create a table
add table inet example_table

# Create a chain for incoming packets that drops all packets
# that are not explicitly allowed by any rule in this chain
add chain inet example_table example_chain { type filter hook input priority 0 ; policy drop ; }

# Add a rule that accepts connections to port 22 (ssh)
add rule inet example_table example_chain tcp dport ssh accept
```

6.1.2. 运行 nftables 脚本

您可以通过将脚本传递给 nft 工具或直接执行脚本来运行 nftables 脚本。

先决条件

- **本节的流程假设您在 /etc/ nftables /example_firewall.nft 文件中存储了一个 nftables 脚本。**

过程 6.1. 使用 nft 工具运行 nftables 脚本

- **要通过将其传给 nft 工具来运行 nftables 脚本，请输入：**

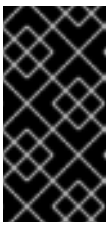
```
# nft -f /etc/nftables/example_firewall.nft
```

过程 6.2. 直接运行 nftables 脚本：

1. 只需要执行一次的步骤：

   1. 确保脚本以以下 shebang 序列开头：

      ```
      #!/usr/sbin/nft -f
      ```

      重要

      如果省略 -f 参数，nft 工具不会读取脚本，并显示 Error: syntax error, unexpected newline, expecting string。

   2. 可选：将脚本的所有者设置为 root ：

      ```
      # chown root /etc/nftables/example_firewall.nft
      ```

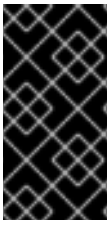
   3. 使脚本可以被其所有者执行：

      ```
      # chmod u+x /etc/nftables/example_firewall.nft
      ```

2. 运行脚本：

   ```
   # /etc/nftables/example_firewall.nft
   ```

   如果没有输出结果，系统将成功执行该脚本。



重要

即使 nft 成功地执行了脚本，在脚本中错误放置的规则、缺失的参数或其他问题都可能会导致防火墙的行为不符合预期。

其他资源

- 有关设置文件所有者的详情，请查看 chown (1) 手册页。

- 有关设置文件权限的详情，请查看 **chmod (1)** 手册页。

- 有关使用系统引导载入 **nftables** 规则的更多信息，请参阅 第 6.1.6 节 “系统引导时自动载入 nftables 规则”

6.1.3. 使用 nftables 脚本中的注释

nftables 脚本环境将 **#** 字符右侧的所有内容解释为注释。

> **例 6.1. nftables 脚本中的注释**
>
> 注释可在一行的开始，也可以在命令后：
>
> ```
> ...
> # Flush the rule set
> flush ruleset
>
> add table inet example_table  # Create a table
> ...
> ```

6.1.4. 使用 nftables 脚本中的变量

要在 **nftables** 脚本中定义一个变量，请使用 **define** 关键字。您可以在变量中存储单个值和匿名集合。对于更复杂的场景，请使用命名集或 **verdict** 映射。

只有一个值的变量

以下示例定义了名为 **INET_DEV** 的变量，其值为 **enp1s0** ：

```
define INET_DEV = enp1s0
```

您可以通过在 **$** 符号后跟变量名称来在脚本中使用变量：

```
...
add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept
...
```

包含匿名集合的变量

以下示例定义了一个包含匿名集合的变量：

```
define DNS_SERVERS = { 192.0.2.1, 192.0.2.2 }
```

您可以通过在 $ 符号后跟变量名称来在脚本中使用变量：

```
add rule inet example_table example_chain ip daddr $DNS_SERVERS accept
```

注意

请注意，在规则中使用大括号时具有特殊的意义，因为它们表示变量代表一个集合。

其他资源

- 有关集合的详情请参考 第 6.4 节 “使用 nftables 命令中的设置”。
- 有关验证映射的详情，请参考 第 6.5 节 “在 nftables 命令中使用 verdict 映射”。

6.1.5. 在 nftables 脚本中包含文件

nftables 脚本环境可让管理员使用 include 语句包含其他脚本。

如果您只指定了文件名，而没有绝对路径或相对路径，那么 nftables 将包含默认搜索路径中的文件，在 Red Hat Enterprise Linux 上，该路径设为 /etc。

例 6.2. 包含默认搜索目录中的文件

从默认搜索目录中包含一个文件：

```
include "example.nft"
```

例 6.3. 包括一个 目录中的所有 lfnft 文件

要包含在 /etc/nftables/rulesets/ 目录中以 lf nft 结尾的所有文件：

```
include "/etc/nftables/rulesets/*.nft"
```

请注意，include 语句不匹配以点开头的文件。

其他资源

- 详情请查看 nft (8) 手册页中的 Include files 部分。

6.1.6. 系统引导时自动载入 nftables 规则

nftables systemd 服务加载包含在 /etc/sysconfig/nftables.conf 文件中的防火墙脚本。这部分论述了如何在系统引导时载入防火墙规则。

先决条件

- nftables 脚本存储在 /etc/nftables/ 目录中。

过程 6.3. 系统引导时自动载入 nftables 规则

1. 编辑 /etc/sysconfig/nftables.conf 文件。

   - 如果您在安装 nftables 软件包时增强了在 /etc/nftables/ 中创建的 lf nft 脚本，请取消对这些脚本的 include 语句的注释。

   - 如果您从头编写脚本，请添加 include 语句来包括这些脚本。例如，要在 nftables 服务启动时载入 /etc/nftables/example.nft 脚本，请添加：

     ```
     include "/etc/nftables/example.nft"
     ```

2. （可选）启动 nftables 服务来加载防火墙规则，而不重启系统：

   ```
   # systemctl start nftables
   ```

3. 启用 nftables 服务。

```
# systemctl enable nftables
```

其他资源

- 如需更多信息，请参阅 第 6.1.1 节 “支持的 nftables 脚本格式”。

6.2. 创建和管理 NFTABLES 表、链和规则

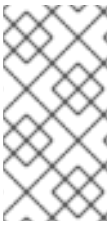
本节介绍如何显示 nftables 规则集以及如何管理它。

6.2.1. 显示 nftables 规则集

nftables 的规则集包含表、链和规则。本节介绍如何显示此规则集。

要显示所有规则集，请输入：

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport http accept
    tcp dport ssh accept
  }
}
```



注意

默认情况下，nftables 不预先创建表。因此，在没有表的情况下显示主机上设置的规则，nft list ruleset 命令不会显示任何输出。

6.2.2. 创建 nftables 表

nftables 中的表是包含链、规则、集合和其他对象集合的名字空间。本节介绍如何创建表。

每个表都必须定义一个地址系列。表的地址系列定义了表进程的类型。在创建表时，您可以设置以下地址系列之一：

- **ip** ：仅匹配 IPv4 数据包。如果没有指定地址系列，这是默认设置。
- **ip6**: 仅匹配 IPv6 数据包。
- **inet**: 匹配 IPv4 和 IPv6 数据包。
- **arp**: 匹配 IPv4 地址解析协议(ARP)数据包。
- **网桥** ：匹配遍历网桥设备的数据包。
- **netdev** ：匹配来自 ingress 的数据包。

过程 6.4. 创建 nftables 表

1. 使用 **nft add table** 命令来创建新表。例如，要创建一个名为 **example_table** 的表，用于处理 IPv4 和 IPv6 数据包：

```
# nft add table inet example_table
```

2. 另外，还可列出规则集中的所有表：

```
# nft list tables
table inet example_table
```

其他资源

- 有关地址系列的详情，请查看 **nft (8)** 手册页中的 **Address families** 部分。
- 有关您可以在表中运行的其他操作的详情，请查看 **nft (8)** 手册页中的 **Tables** 部分。

6.2.3. 创建 nftables 链

chains 是规则的容器。存在以下两种规则类型：

- 基本链：您可以使用基础链作为来自网络堆栈的数据包的入口点。

- 常规链：您可以使用常规链作为 跳过 目标，并更好地组织规则。

这个步骤描述了如何在现有表中添加基本链。

先决条件

- 已存在您要添加新链的表。

过程 6.5. 创建 nftables 链

1. 使用 nft add chain 命令来创建新链。例如，要在 example_table 中创建一个名为 example_chain 的链：

```
# nft add chain inet example_table example_chain '{ type filter hook input priority 0 ; policy accept ; }'
```

重要

要避免 shell 认为分号作为命令结尾，您必须用反斜杠转义分号。此外，一些 shell 也解译大括号，因此请使用 ticks (1)将大括号和它们内的任何内容引用。

这个链过滤传入的数据包。priority 参数指定 nftables 进程处理具有相同 hook 值的链的顺序。较低优先级的值优先于优先级更高的值。policy 参数为此链中的规则设置默认操作。请注意，如果您远程登录到服务器，并将默认策略设置为 drop，如果没有其他规则允许远程访问，则会立即断开连接。

2. 另外，还可以显示所有链：

```
# nft list chains
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
  }
}
```

其他资源

- 有关地址系列的详情，请查看 **nft (8)** 手册页中的 **Address families** 部分。

- 有关您可以在链上运行的其他操作的详情，请查看 **nft (8)** 手册页中的 **链** 部分。

6.2.4. 将规则附加到 nftables 链的末尾

本节介绍如何在现有 **nftables** 链的末尾附加规则。

先决条件

- 您要添加该规则的链已存在。

过程 6.6. 将规则附加到 nftables 链的末尾

1. 要添加新规则，请使用 **nft add rule** 命令。例如，要在 **example_table** 中的 **example_chain** 中添加一条规则，以允许端口 **22** 上的 **TCP** 流量：

```
# nft add rule inet example_table example_chain tcp dport 22 accept
```

您还可以指定服务名称而不是端口号。在示例中，您可以使用 **ssh** 而不是端口号 **22**。请注意，服务名称根据 **/etc/services** 文件中的条目解析为端口号。

2. 另外，还可在 **example_table** 中显示所有链及其规则：

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    ...
    tcp dport ssh accept
  }
}
```

其他资源

- 有关地址系列的详情，请查看 **nft (8)** 手册页中的 **Address families** 部分。

- 有关您可以在链中运行的其他操作的详情，请查看 **nft (8)** 手册页中的 **Rules** 部分。

6.2.5. 在 nftables 链的开头插入一条规则

本节介绍如何在现有 **nftables** 链的开头插入规则。

先决条件

- 您要添加该规则的链已存在。

过程 6.7. 在 nftables 链的开头插入一条规则

1. 要插入新规则，请使用 **nft insert rule** 命令。例如，要在 **example_table** 中的 **example_chain** 插入一条规则，以允许端口 **22** 上的 **TCP** 流量：

```
# nft insert rule inet example_table example_chain tcp dport 22 accept
```

您还可以指定服务名称而不是端口号。在示例中，您可以使用 **ssh** 而不是端口号 **22**。请注意，服务名称根据 **/etc/services** 文件中的条目解析为端口号。

2. 另外，还可在 **example_table** 中显示所有链及其规则：

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept
    ...
  }
}
```

其他资源

- 有关地址系列的详情，请查看 **nft (8)** 手册页中的 **Address families** 部分。

- 有关您可以在链中运行的其他操作的详情，请查看 **nft (8)** 手册页中的 **Rules** 部分。

6.2.6. 在 nftables 链的特定位置插入一条规则

本节解释了如何在 nftables 链中的现有规则前后插入规则。这样，您可以将新规则放在正确的位置上。

先决条件

- 您要添加该规则的链已存在。

过程 6.8. 在 nftables 链的特定位置插入一条规则

1. 使用 nft -a list ruleset 命令显示 example_table 中的所有链及其规则，包括其句柄：

```
# nft -a list table inet example_table
table inet example_table { # handle 1
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport 22 accept # handle 2
    tcp dport 443 accept # handle 3
    tcp dport 389 accept # handle 4
  }
}
```

使用 -a 显示句柄。您需要此信息才能在后续步骤中定位新规则。

2. 将新规则插入到 example_table 中的 example_chain 链中：

- 要在句柄 3 前插入一条允许端口 636 上 TCP 流量的规则，请输入：

```
# nft insert rule inet example_table example_chain position 3 tcp dport 636 accept
```

- 要添加一条规则，在句柄 3 后允许端口 80 上的 TCP 流量，请输入：

```
# nft add rule inet example_table example_chain position 3 tcp dport 80 accept
```

3. 另外，还可在 example_table 中显示所有链及其规则：

```
# nft -a list table inet example_table
table inet example_table { # handle 1
  chain example_chain { # handle 1
```

```
    type filter hook input priority filter; policy accept;
    tcp dport 22 accept # handle 2
    tcp dport 636 accept # handle 5
    tcp dport 443 accept # handle 3
    tcp dport 80 accept # handle 6
    tcp dport 389 accept # handle 4
  }
}
```

其他资源

- **有关地址系列的详情，请查看 nft (8) 手册页中的 Address families 部分。**
- **有关您可以在链中运行的其他操作的详情，请查看 nft (8) 手册页中的 Rules 部分。**

6.3. 使用 NFTABLES 配置 NAT

使用 nftables，您可以配置以下网络地址转换(NAT)类型：

- **伪装**
- **源 NAT (SNAT)**
- **目标 NAT (DNAT)**
- **重定向**

6.3.1. 不同的 NAT 类型： masquerading、source NAT、destination NAT 和 redirect

这些是不同的网络地址转换(NAT)类型：

伪装和源 NAT（SNAT）

使用这些 NAT 类型之一更改数据包的源 IP 地址。例如，互联网服务提供商不路由私有 IP 范围，如 10.0.0.0/8。如果您在网络中使用私有 IP 范围，并且用户应该能够访问 Internet 上的服务器，请将这些范围内的数据包的源 IP 地址映射到公共 IP 地址。

伪装和 SNAT 非常相似。不同之处是：

- 伪装自动使用传出接口的 IP 地址。因此，如果传出接口使用了动态 IP 地址，则使用伪装。
- SNAT 将数据包的源 IP 地址设置为指定 IP，且不会动态查找传出接口的 IP 地址。因此，SNAT 比伪装更快。如果传出接口使用了固定 IP 地址，则使用 SNAT。

目标 NAT（DNAT）

使用此 NAT 类型将传入的流量路由到不同主机。例如，如果您的 web 服务器使用保留 IP 范围内的 IP 地址，因此无法直接从互联网访问，您可以在路由器上设置 DNAT 规则，以将传入的流量重定向到这个服务器。

重定向

这个类型是 IDT 的特殊示例，它根据链 hook 将数据包重定向到本地机器。例如，如果服务运行在与其标准端口不同的端口上，您可以将传入的流量从标准端口重定向到此特定端口。

6.3.2. 使用 nftables 配置伪装

伪装使路由器动态地更改通过接口到接口 IP 地址发送的数据包的源 IP。这意味着，如果接口被分配了一个新 IP，nftables 会在替换源 IP 时自动使用新的 IP。

以下流程描述了如何将通过 ens3 接口离开主机的数据包的源 IP 替换为 ens3 上设置的 IP。

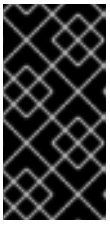
过程 6.9. 使用 nftables 配置伪装

1. 创建一个表：

```
# nft add table nat
```

2. 向表中添加 prerouting 和 postrouting 链：

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```



重要

即使您没有向 prerouting 链中添加规则，nftables 框架也会要求此链与传入的数据包回复匹配。

请注意，您必须将 -- 选项传递给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选项。

3. 向 postrouting 链中添加一条规则，来匹配 ens3 接口上传出的数据包：

```
# nft add rule nat postrouting oifname "ens3" masquerade
```

6.3.3. 使用 nftables 配置源 NAT

在路由器上，源 NAT (SNAT)允许您将通过接口发送的数据包 IP 更改为特定的 IP 地址。

以下流程描述了如何替换数据包的源 IP，使其通过 ens3 接口离开路由器到达192.0.2.1。

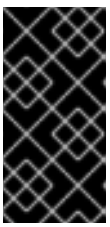
过程 6.10. 使用 nftables 配置源 NAT

1. 创建一个表：

```
# nft add table nat
```

2. 向表中添加 prerouting 和 postrouting 链：

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```



重要

即使您没有向 prerouting 链添加规则，nftables 框架也要求此链与传出数据包回复匹配。

请注意，您必须将 -- 选项传递给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选

项。

3. 向 postrouting 链中添加一条规则，该规则将使用 192.0.2.1 替换通过ens3 的传出数据包的源 IP ：

```
# nft add rule nat postrouting oifname "ens3" snat to 192.0.2.1
```

其他资源

- 如需更多信息，请参阅 第 6.6.2 节 “将特定本地端口上传入的数据包转发到不同主机”。

6.3.4. 使用 nftables 配置目标 NAT

目标 NAT 可让您将路由器上的流量重定向到无法直接从互联网访问的主机。

以下流程描述了如何将发送到路由器端口 80 和 443 的传入流量重定向到 IP 地址为 192.0.2.1 的主机。

过程 6.11. 使用 nftables 配置目标 NAT

1. 创建一个表：

```
# nft add table nat
```

2. 向表中添加 prerouting 和 postrouting 链：

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

重要

即使您没有向 postrouting 链添加规则，nftables 框架也要求此链与传出数据包回复匹配。

请注意，您必须将 -- 选项传递给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选项。

3. 向 prerouting 链中添加一条规则，将发送到端口 80 和 443 的 ens3 接口上的传入流量重定向到 IP 为 192.0.2.1 的主机：

```
# nft add rule nat prerouting iifname ens3 tcp dport { 80, 443 } dnat to 192.0.2.1
```

4. 根据您的环境，添加 SNAT 或伪装规则以更改源地址：

   1. 如果 ens3 接口使用动态 IP 地址，请添加一条伪装规则：

   ```
   # nft add rule nat postrouting oifname "ens3" masquerade
   ```

   2. 如果 ens3 接口使用静态 IP 地址，请添加 SNAT 规则。例如，如果 ens3 使用 198.51.100.1 IP 地址：

   ```
   # nft add rule nat postrouting oifname "ens3" snat to 198.51.100.1
   ```

其他资源

- 如需更多信息，请参阅 第 6.3.1 节 “不同的 NAT 类型： masquerading、source NAT、destination NAT 和 redirect”。

6.3.5. 使用 nftables 配置重定向

重定向 功能是目标网络地址转换(DNAT)的一种特殊情况，它根据链 hook 将数据包重定向到本地计算机。

以下流程描述了如何将发送到本地主机端口 22 的流量重定向到端口 2222。

过程 6.12. 使用 nftables 配置重定向

1. 创建一个表：

```
# nft add table nat
```

2. 在表中添加 prerouting 链：

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
```

请注意，您必须将 -- 选项传递给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选项。

3. 向 prerouting 链中添加一条规则，其将端口 22 上的传入流量重定向到端口 2222 ：

```
# nft add rule nat prerouting tcp dport 22 redirect to 2222
```

其他资源

- 如需更多信息，请参阅 第 6.3.1 节 “不同的 NAT 类型： masquerading、source NAT、destination NAT 和 redirect”。

6.4. 使用 NFTABLES 命令中的设置

nftables 框架原生支持集合。您可以使用一个集合，例如，规则匹配多个 IP 地址、端口号、接口或其他匹配标准。

6.4.1. 在 nftables 中使用匿名集合

匿名集合包含用逗号分开的值，如 { 22, 80, 443 }，您直接在规则中使用。您还可以将匿名集合用于 IP 地址或其他匹配标准。

匿名集合的缺陷是，如果要更改集合，则需要替换规则。对于动态解决方案，请使用命名集，如 第 6.4.2 节 “在 nftables 中使用命名集” 所述。

先决条件

- inet 系列中的 example_chain 链和 example_table 表存在。

过程 6.13. 在 nftables 中使用匿名集合

1. 例如，要向 example_table 中的 example_chain 添加一条规则，其允许传入流量到端口 22、80 和 443 ：

```
# nft add rule inet example_table example_chain tcp dport { 22, 80, 443 } accept
```

2. 另外，还可在 example_table 中显示所有链及其规则：

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport { ssh, http, https } accept
  }
}
```

6.4.2. 在 nftables 中使用命名集

nftables 框架支持可变命名集合。命名集是一个列表或一组元素，您可以在表中的多个规则中使用。匿名集合的另外一个好处在于，您可以更新命名的集合而不必替换使用集合的规则。

当您创建一个命名集时，必须指定集合包含的元素类型。您可以设置以下类型：

- 包含 IPv4 地址或范围的集合的 ipv4_addr，如 192.0.2.1 或 192.0.2.0/24。
- 包含 IPv6 地址或范围的集合的 ipv6_addr，如 2001:db8:1::1 或 2001:db8:1::1/64。
- 包含介质访问控制(MAC)地址列表的集合的 ether_addr，如 52:54:00:6b:66:42。
- 包含互联网协议类型列表的集合的 inet_proto，如 tcp。
- 包含互联网服务列表的集合的 inet_service，如 ssh。
- 包含数据包标记列表的集合的 mark。数据包标记可以是任意正 32 位整数值(0 到 2147483647)。

先决条件

- example_chain 链和 example_table 表存在。

过程 6.14. 在 nftables 中使用命名集

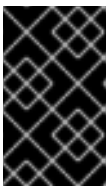
1. 创建一个空集。以下示例为 IPv4 地址创建一个集合：

   a. 要创建可存储多个独立 IPv4 地址的集合：

   ```
   # nft add set inet example_table example_set { type ipv4_addr \; }
   ```

   b. 要创建可存储 IPv4 地址范围的集合：

   ```
   # nft add set inet example_table example_set { type ipv4_addr \; flags interval \; }
   ```

   重要

   要避免 shell 认为分号作为命令结尾，您必须用反斜杠转义分号。

2. 另外，还可创建使用该集合的规则。例如，以下命令向 example_table 中的 example_chain 添加一条规则，该规则将丢弃 example_set 中来自 IPv4 地址的所有数据包。

   ```
   # nft add rule inet example_table example_chain ip saddr @example_set drop
   ```

   由于 example_set 仍为空，所以该规则目前不起作用。

3. 向 example_set 中添加 IPv4 地址：

   a. 如果您创建存储单个 IPv4 地址的集合，请输入：

   ```
   # nft add element inet example_table example_set { 192.0.2.1, 192.0.2.2 }
   ```

   b. 如果您创建存储 IPv4 范围的集合，请输入：

   ```
   # nft add element inet example_table example_set { 192.0.2.0-192.0.2.255 }
   ```

   当您指定 IP 地址范围时，您也可以使用无类别域间路由(CIDR)标记，如上例中的 192.0.2.0/24。

6.4.3. 相关信息

有关集合的详情，请查看 nft (8) 手册页中的 Sets 部分。

6.5. 在 NFTABLES 命令中使用 VERDICT 映射

判决映射（也称为字典），使 nft 能够通过将匹配条件映射到某个操作来根据数据包信息执行操作。

6.5.1. 在 nftables 中使用匿名映射

匿名映射是您直接在规则中使用的 { match_criteria : action } 语句。这个语句可以包含多个用逗号分开的映射。

匿名映射的缺点是，如果要修改映射，则必须替换规则。对于动态解决方案，请使用命名映射，如第 6.5.2 节 “在 nftables 中使用命名映射”所述。

这个示例描述了如何使用匿名映射将 IPv4 和 IPv6 协议的 TCP 和 UDP 数据包路由到不同的链，以分别计算传入的 TCP 和 UDP 数据包。

过程 6.15. 在 nftables 中使用匿名映射

1. 创建 example_table ：

```
# nft add table inet example_table
```

2. 在 example_table 中创建 tcp_packets 链：

```
# nft add chain inet example_table tcp_packets
```

3. 向统计此链中流量的 tcp_packets 中添加一条规则：

```
# nft add rule inet example_table tcp_packets counter
```

4. 在 example_table 中创建 udp_packets 链：

```
# nft add chain inet example_table udp_packets
```

5. 向统计此链中流量的 udp_packets 中添加一条规则：

```
# nft add rule inet example_table udp_packets counter
```

6. 为传入的流量创建一个链。例如，要在过滤传入的流量的 example_table 中创建一个名为 incoming_traffic 的链：

```
# nft add chain inet example_table incoming_traffic { type filter hook input priority 0 \; }
```

7. 添加一条带有到 incoming_traffic 匿名映射的规则 ：

```
# nft add rule inet example_table incoming_traffic ip protocol vmap { tcp : jump tcp_packets,
udp : jump udp_packets }
```

匿名映射区分数据包，并根据它们的协议将它们发送到不同的计数链。

8. 要列出流量计数器，请显示 example_table ：

```
# nft list table inet example_table
table inet example_table {
  chain tcp_packets {
    counter packets 36379 bytes 2103816
  }

  chain udp_packets {
    counter packets 10 bytes 1559
  }

  chain incoming_traffic {
    type filter hook input priority filter; policy accept;
    ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
  }
}
```

tcp_packets 和 udp_packets 链中的计数器显示两者接收的数据包和字节数。

6.5.2. 在 nftables 中使用命名映射

nftables 框架支持命名映射。您可以在表中的多个规则中使用这些映射。匿名映射的另一个好处在于，您可以更新命名映射而不比替换使用它的规则。

在创建命名映射时，您必须指定元素的类型：

- 匹配部分包含 IPv4 地址的映射的 ipv4_addr，如 192.0.2.1。
- 匹配部分包含 IPv6 地址的映射的 ipv6_addr，如 2001:db8:1::1。
- 匹配部分包含介质访问控制(MAC)地址的映射的 ether_addr，如 52:54:00:6b:66:42。
- 匹配部分包含互联网协议类型的映射的 inet_proto ，如 tcp。
- 匹配部分包含互联网服务名称端口号的映射的inet_service ，如 ssh 或 22。
- 匹配部分包含数据包的映射的 mark 。数据包标记可以是任意正 32 位整数值(0 到 2147483647)。
- 匹配部分包含计数器值的映射的 counter。计数器值可以是任意正 64 位整数值。
- 匹配部分包含配额值的映射的 quota。配额值可以是任意正 64 位整数值。

这个示例论述了如何根据源 IP 地址允许或丢弃传入的数据包。使用命名映射时，您只需要一条规则来配置这种场景，而 IP 地址和操作被动态存储在映射中。此流程还描述了如何从映射中添加和删除条目。

过程 6.16. 在 nftables 中使用命名映射

1. 创建表。例如，要创建一个名为 example_table 的表来处理 IPv4 数据包：

```
# nft add table ip example_table
```

2. 创建链。例如，要在 example_table 中创建一个名为 example_chain 的链：

```
# nft add chain ip example_table example_chain { type filter hook input priority 0 \; }
```

重要

要避免 shell 认为分号作为命令结尾，您必须用反斜杠转义分号。

3. **创建一个空的映射。例如，要为 IPv4 地址创建映射：**

```
# nft add map ip example_table example_map { type ipv4_addr : verdict \; }
```

4. **创建使用该映射的规则。例如，以下命令向 example_table 中的 example_chain 添加一条规则，该规则将操作应用到 example_map 中定义的 IPv4 地址：**

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

5. **向 example_map 添加 IPv4 地址和对应操作：**

```
# nft add element ip example_table example_map { 192.0.2.1 : accept, 192.0.2.2 : drop }
```

这个示例定义了 IPv4 地址到操作的映射。与以上创建的规则相结合，防火墙接受来自 192.0.2.1 的数据包，丢弃来自 192.0.2.2 的数据包。

6. **另外，还可添加另一个 IP 地址和 action 语句来增强映射：**

```
# nft add element ip example_table example_map { 192.0.2.3 : accept }
```

7. **（可选）从映射中删除条目：**

```
# nft delete element ip example_table example_map { 192.0.2.1 }
```

8. **另外，还可显示规则集：**

```
# nft list ruleset
table ip example_table {
  map example_map {
    type ipv4_addr : verdict
    elements = { 192.0.2.2 : drop, 192.0.2.3 : accept }
  }
```

```
  chain example_chain {
    type filter hook input priority filter; policy accept;
    ip saddr vmap @example_map
  }
}
```

6.5.3. 相关信息

有关 verdict 映射的详情，请查看 nft (8) 手册页中的 Maps 部分。

6.6. 使用 NFTABLES 配置端口转发

端口转发可让管理员将发送到特定目的端口的数据包转发到不同的本地或者远程端口。

例如，如果您的 web 服务器没有公共 IP 地址，您可以在防火墙上设置一条端口转发规则，将防火墙上端口 80 和 443 上的传入数据包转发到 web 服务器。使用这个防火墙规则，互联网中的用户可以使用防火墙的 IP 或主机名访问网页服务器。

6.6.1. 将传入的数据包转发到不同的本地端口

这部分描述了如何将端口 8022 上的传入 IPv4 数据包转发到本地系统的端口 22 。

过程 6.17. 将传入的数据包转发到不同的本地端口

1. 使用 ip 地址系列创建一个名为 nat 的表：

```
# nft add table ip nat
```

2. 向表中添加 prerouting 和 postrouting 链：

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
```

注意

将 -- 选项传给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选项。

3. 向 prerouting 链中添加一条规则，将端口 8022 上的传入数据包重定向到本地端口 22 ：

```
# nft add rule ip nat prerouting tcp dport 8022 redirect to :22
```

6.6.2. 将特定本地端口上传入的数据包转发到不同主机

您可以使用目标网络地址转换(DNAT)规则将本地端口上传入的数据包转发到远程主机。这可让互联网中的用户访问使用专用 IP 地址在主机上运行的服务。

这个流程描述了如何将本地端口 443 上的传入 IPv4 数据包转发到 IP 地址为 192.0.2.1 的远程系统上的同一端口。

前提条件

- 您以 root 用户身份登录应该转发数据包的系统上。

过程 6.18. 将特定本地端口上传入的数据包转发到不同主机

1. 使用 ip 地址系列创建一个名为 nat 的表：

```
# nft add table ip nat
```

2. 向表中添加 prerouting 和 postrouting 链：

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
```

注意

将 -- 选项传给 nft 命令，以避免 shell 将负优先级值解析为 nft 命令的选项。

3. 向 prerouting 链添加一条规则，该规则将端口 443 上的传入数据包重定向到 192.0.2.1 上的同一端口：

```
# nft add rule ip nat prerouting tcp dport 443 dnat to 192.0.2.1
```

4. 向 postrouting 链中添加一条规则来伪装出站流量：

```
# nft add rule ip nat postrouting ip daddr 192.0.2.1 masquerade
```

5. 启用数据包转发：

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

6.7. 使用 NFTABLES 来限制连接数量

您可以使用 nftables 来限制连接数或限制到建立给定数量连接的块 IP 地址，以防止它们使用太多的系统资源。

6.7.1. 使用 nftables 限制连接数量

nft 工具的 ct count 参数可让管理员限制连接数量。这个步骤描述了如何限制进入的连接的基本示例。

先决条件

- example_table 中的基础example_chain 存在。

过程 6.19. 使用 nftables 限制连接数量

1. 添加一条规则，仅允许从 IPv4 地址同时连接到 SSH 端口(22)，并从同一 IP 拒绝所有进一步连接：

```
# nft add rule ip example_table example_chain tcp dport ssh meter
example_meter { ip saddr ct count over 2 } counter reject
```

2. 另外，还可以显示上一步中创建的 meter：

```
# nft list meter ip example_table example_meter
table ip example_table {
  meter example_meter {
    type ipv4_addr
    size 65535
```

```
    elements = { 192.0.2.1 : ct count over 2 , 192.0.2.2 : ct count over 2  }
  }
}
```

elements 条目显示当前与该规则匹配的地址。在这个示例中，elements 列出已活跃连接到 SSH 端口的 IP 地址。请注意，输出不会显示活跃连接的数量，或者连接是否被拒绝。

6.7.2. 在一分钟内尝试超过十个进入的 TCP 连接的 IP 地址

nftables 框架可让管理员动态更新集合。本节解释了如何使用这个功能临时阻止在一分钟内建立十个 IPv4 TCP 连接的主机。五分钟后，nftables 会自动从拒绝列表中删除 IP 地址。

过程 6.20. 在一分钟内尝试超过十个进入的 TCP 连接的 IP 地址

1. **使用 ip 地址系列创建 filter 表：**

```
# nft add table ip filter
```

2. **在 filter 表中添加输入链：**

```
# nft add chain ip filter input { type filter hook input priority 0 \; }
```

3. **在 filter 表中添加名为 denylist 的集合：**

```
# nft add set ip filter denylist { type ipv4_addr \; flags dynamic, timeout \; timeout 5m \; }
```

这个命令为 IPv4 地址创建动态设置。timeout 5m 参数定义 nftables 在 5 分钟后自动删除集合中的条目。

4. **添加一条规则，该规则会在一分钟内尝试建立十个新的 TCP 连接的主机源 IP 地址添加到 denylist 集：**

```
# nft add rule ip filter input ip protocol tcp ct state new, untracked limit rate over 10/minute
add @denylist { ip saddr }
```

5. **添加一条规则，该规则丢弃来自 denylist 集合中 IP 地址的所有连接：**

```
# nft add rule ip filter input ip saddr @denylist drop
```

6.7.3. 其他资源

- 如需更多信息，请参阅 第 6.4.2 节 “在 nftables 中使用命名集”。

6.8. 调试 NFTABLES 规则

nftables 框架为管理员提供了不同的选项来调试规则，以及数据包是否匹配。本节描述了这些选项。

6.8.1. 创建带有计数器的规则

在识别规则是否匹配时，可以使用计数器。本节描述了如何创建带有计数器的新规则。

有关在现有规则中添加计数器的步骤，请参阅 第 6.8.2 节 “在现有规则中添加计数器”。

先决条件

- 您要添加该规则的链已存在。

过程 6.21. 创建带有计数器的规则

1. 在链中添加带有 **counter** 参数的新规则。以下示例添加了一个带有计数器的规则，允许端口 22 上的 TCP 流量，并统计与此规则匹配的数据包和流量：

```
# nft add rule inet example_table example_chain tcp dport 22 counter accept
```

2. 显示计数器值：

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

6.8.2. 在现有规则中添加计数器

在识别规则是否匹配时，可以使用计数器。本节论述了如何在现有规则中添加计数器。

有关使用计数器添加新规则的步骤，请参阅 第 6.8.1 节 “创建带有计数器的规则”。

先决条件

- 您要添加计数器的规则已存在。

过程 6.22. 在现有规则中添加计数器

1. 在链中显示规则及其句柄：

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. 通过将规则替换为 **counter** 参数来添加计数器。以下示例替换了上一步中显示的规则并添加计数器：

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 counter accept
```

3. 显示计数器值：

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

6.8.3. 监控与现有规则匹配的数据包

nftables 中的追踪功能与 **nft monitor** 命令相结合，使管理员能够显示与某一规则匹配的数据包。该流程描述了如何为规则启用追踪以及与本规则匹配的监控数据包。

先决条件

- 您要添加计数器的规则已存在。

过程 6.23. 监控与现有规则匹配的数据包

1. 在链中显示规则及其句柄：

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. 通过使用 **meta nftrace set** 参数替换规则来添加追踪功能。1以下示例替换了上一步中显示的规则并启用追踪：

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 meta nftrace set 1 accept
```

3. 使用 **nft monitor** 命令来显示追踪。以下示例过滤了命令的输出，来只显示包含 **inet example_table example_chain** 的条目：

```
# nft monitor | grep "inet example_table example_chain"
trace id 3c5eb15e inet example_table example_chain packet: iif "enp1s0" ether saddr 52:54:00:17:ff:e4 ether daddr 52:54:00:72:2f:6e ip saddr 192.0.2.1 ip daddr 192.0.2.2 ip dscp cs0 ip ecn not-ect ip ttl 64 ip id 49710 ip protocol tcp ip length 60 tcp sport 56728 tcp dport ssh tcp flags == syn tcp window 64240
trace id 3c5eb15e inet example_table example_chain rule tcp dport ssh nftrace set 1 accept (verdict accept)
    ...
```

警告

根据启用了追踪的规则数量以及匹配流量的数量，**nft monitor** 命令可以显示很多输出。使用 **grep** 或其他工具来过滤输出。

第 7 章 系统审计

Linux 审计系统提供了一种方式来跟踪系统上与安全相关的信息。根据预配置的规则，审计会生成日志条目，来尽可能多地记录系统上所发生的事件的相关信息。对于关键任务环境而言至关重要，可用来确定安全策略的违反者及其所执行的操作。审计不会为您的系统提供额外的安全，而是用于发现系统上使用的安全策略的违规。可以通过其他安全措施(如 SELinux)进一步防止这些违规。

以下列表总结了审计可以在其日志文件中记录的一些信息：

- 事件的日期、时间、类型和结果.
- 主题和对象的敏感度标签。
- 事件与触发事件的用户身份的关联。
- 对审计配置的所有修改，以及对访问审计日志文件的尝试。
- 所有身份验证机制的使用，如 SSH 和 Kerberos 等。
- 对任何受信任数据库的修改，如 /etc/passwd。
- 尝试将信息导入系统或从系统导出。
- 根据用户身份、主题和对象标签以及其他属性包含或排除事件。

审计系统的使用也是许多安全相关认证的一项要求。审计旨在满足或超出以下认证或合规指南的要求：

- 受控访问保护配置文件(CAPP)
- 标记的安全保护配置文件(LSPP)

- 规则集基本访问控制(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)

审计还包括：

- 由国家信息保障合作伙伴(NIAP)和最佳安全行业(BSI)评估。
- Red Hat Enterprise Linux 5 上的 LSPP/CAPP/RSBAC/EAL4+ 认证.
- Red Hat Enterprise Linux 6 上的操作系统保护配置文件/评估保障级别 4+(OSPP/EAL4+)认证.

使用案例

监视文件访问

审计可以跟踪文件或目录是否已被访问、修改、执行或者文件的属性是否已改变。例如，这有助于检测对重要文件的访问，并在其中一个文件损坏时提供审计跟踪。

监控系统调用

可将审计配置为在每次使用特定系统调用时生成日志条目。例如，这可用于通过监控 settimeofday、clock_adjtime 和其他与时间相关的系统调用来跟踪对系统时间的修改。

记录用户运行的命令

审计可以跟踪文件是否已被执行，因此可以定义一个规则以记录每次特定命令的执行。例如，可以对 /bin 目录中的每个可执行文件定义一个规则。然后，可以按用户 ID 搜索生成的日志条目，以生成

每个用户所执行的命令的审计跟踪。

记录系统路径名称的执行

除了观察在规则调用时将路径转换为 inode 的文件访问之外，审计现在还可以观察路径的执行，即使路径在规则调用中不存在，或者在规则调用后文件被替换了。这允许规则在升级程序可执行文件后或甚至在其安装之前继续运行。

记录安全事件

pam_faillock 认证模块能够记录失败的登录尝试。也可以将审计设置为记录失败的登录尝试，并提供有关试图登录的用户的附加信息。

搜索事件

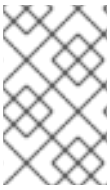
审计提供了 ausearch 工具，可用于过滤日志条目，并根据多个条件提供完整的审计跟踪。

运行总结报告

aureport 实用程序可用于生成记录事件的日常报告等。然后，系统管理员可以分析这些报告，并进一步调查可疑的活动。

监控网络访问

iptables 和 ebtables 工具可以配置为触发审计事件，允许系统管理员监控网络访问。



注意

系统性能可能会受到影响，具体取决于审计所收集的信息量。

7.1. 审计系统架构

审计系统由两个主要部分组成：用户空间应用程序和工具，以及内核端系统调用处理。内核组件接收用户空间应用程序的系统调用，并通过以下过滤器对其进行过滤：user、task、fstype 或 exit。

系统调用通过 exclude 过滤器后，它将通过上述其中一个过滤器发送，该过滤器根据审计规则配置将其发送到审计守护进程，以进行进一步处理。

用户空间审计守护进程从内核收集信息，并在日志文件中创建条目。其他审计用户空间工具与审计守护进程、内核审计组件或审计日志文件进行交互：

- audisp - Audit 分配程序守护进程与 Audit 守护进程交互，并将事件发送到其他应用程序，以便进一步处理。此守护进程的目的是提供插件机制，以便实时分析程序可以与审计事件交互。
- auditctl - 审计控制实用程序与内核审计组件交互，以管理规则并控制事件生成进程的多个设置和参数。
- 其余的审计工具会将审计日志文件的内容作为输入，并根据用户的要求生成输出。例如，aureport 工具生成所有记录的事件的报告。

7.2. 安装 AUDIT 软件包

要使用审计系统，必须在系统中安装了 audit 软件包。audit 软件包(audit 和 audit-libs)默认安装在 Red Hat Enterprise Linux 7 中。如果您没有安装这些软件包，请以 root 用户身份执行以下命令来安装审计和依赖项：

```
~]# yum install audit
```

7.3. 配置 审计 服务

Audit 守护进程可以在 /etc/audit/auditd.conf 文件中配置。此文件由修改审计守护进程行为的配置参数组成。哈希符号(#)后面的空行和文本将被忽略。详情请查看 auditd.conf(5) man page。

7.3.1. 为安全环境配置 auditd

默认的 auditd 配置应该适合于大多数环境。但是，如果您的环境必须满足严格的安全策略，建议对 /etc/audit/auditd.conf 文件中的审计守护进程配置进行以下设置：

log_file

包含审计日志文件的目录（通常为 /var/log/audit/）应位于单独的挂载点上。这可以防止其他进程消耗此目录的空间，并为审计守护进程提供准确的剩余空间检测。

max_log_file

指定单个审计日志文件的最大大小，必须设置为充分利用保存审计日志文件的分区上的可用空间。

max_log_file 参数指定最大文件大小（以 MB 为单位）。给出的值必须是数字。

max_log_file_action

在达到 max_log_file 中设置的限制后，决定要采取什么操作，应设置为 keep_logs 以防止审计日志文件被覆盖。

space_left

指定磁盘上剩余的可用空间量，该磁盘中触发 space_left_action 参数中设置的操作。必须设置一个数字，让管理员有足够的时间来响应，并释放磁盘空间。space_left 值取决于生成审计日志文件的速率。

如果 space_left 的值指定为整数，它将解释为绝对大小(MiB)。如果该值指定为 1 到 99 之间的数字，后跟一个百分比符号（例如 5%），则审计守护进程会根据包含 log_file 的文件系统的大小计算绝对大小（以 MB 为单位）。

space_left_action

建议您使用适当的通知方法将 space_left_action 参数设置为 email 或 exec。

admin_space_left

指定触发 admin_space_left_action 参数中设置的操作的绝对最小可用空间量，必须将其设置为一个值，以便有足够的空间来记录管理员执行的操作。

此参数的数字值应小于 space_left 的数字值。您还可以在数字后面附加一个百分比符号（例如 1%），以便审计守护进程根据磁盘分区计算数值。

admin_space_left_action

应设置为 single 来将系统置于单用户模式，并允许管理员释放一些磁盘空间。

disk_full_action

指定当保存审计日志文件的分区上没有可用空间时触发的操作，必须设置为 halt 或 single。当审计无法记录事件时，这可确保系统关闭或以单用户模式运行。

disk_error_action

指定当在包含审计日志文件的分区上检测到错误时触发的操作，必须设置为 **syslog**、**single** 或**halt**，具体取决于您处理硬件故障的本地安全策略。

flush

应设置为 **incremental_async**。它与 **freq** 参数结合使用，该参数决定了在强制与硬盘进行硬盘同步前可以将多少条记录发送到磁盘。**freq** 参数应设置为 **100**。这些参数可确保审计事件数据与磁盘上的日志文件同步，同时保持良好的活动性能。

其余配置选项应根据您的本地安全策略来设置。

7.4. 启动 审计 服务

配置了 **auditd** 后，启动服务以收集审计信息并将其存储在日志文件中。以 **root** 用户身份运行以下命令来启动 **auditd** ：

```
~]# service auditd start
```

注意

service命令是与 **auditd** 守护进程正确交互的唯一方法。您需要使用 **service** 命令，以便正确记录 **auid** 值。您只将 **systemctl** 命令用于两个操作： **enable** 和 **status**。

将 **auditd** 配置为在引导时启动：

```
~]# systemctl enable auditd
```

可以使用 **service auditd action** 命令对 **auditd** 执行许多其他操作，其中 **action** 可以是以下之一：

stop

停止 **auditd**。

restart

重新启动 auditd。

reload 或 force-reload

重新加载 /etc/audit/auditd.conf 文件中 auditd 的配置。

rotate

轮转 /var/log/audit/ 目录中的日志文件。

resume

在其之前被暂停后重新恢复审计事件记录，例如，当保存审计日志文件的磁盘分区中没有足够的可用空间时。

condrestart 或 try-restart

只有当 auditd 运行时才重新启动它。

status

显示 auditd 的运行状态。

7.5. 定义审计规则

审计系统对一组规则进行操作，这些规则定义日志文件中要捕获的内容。可以指定以下类型的审计规则：

控制规则

允许修改 Audit 系统的行为及其某些配置。

文件系统规则

也称为文件监视，允许审核对特定文件或目录的访问。

系统调用规则

允许记录任何指定程序进行的系统调用。

可以设置审计规则：

- 在命令行中使用 auditctl 工具。请注意，这些规则在重启后不会保留。详情请查看 第 7.5.1 节 “使用 auditctl定义审计规则”
- 在 /etc/audit/audit.rules 文件中。详情请查看 第 7.5.3 节 “在 /etc/audit/audit.rules 文件中定义持久性审计规则和控制”

7.5.1. 使用 auditctl定义审计规则

auditctl 命令允许您控制审计系统的基本功能，并定义决定记录哪些审计事件的规则。

注意

与 Audit 服务和审计日志文件交互的所有命令都需要 root 特权。确保您以 root 用户身份执行这些命令。另外，需要 CAP_AUDIT_CONTROL 功能来设置审计服务，以及记录用户消息所需的 CAP_AUDIT_WRITE 功能。

定义控制规则

以下是允许您修改审计系统行为的一些控制规则：

-b

在内核中设置现有审计缓冲区的最大数量，例如：

```
~]# auditctl -b 8192
```

-f

设置在检测到关键错误时执行的操作，例如：

```
~]# auditctl -f 2
```

以上配置会在出现严重错误时触发内核 panic。

-e

启用和禁用审计系统或锁定其配置，例如：

```
~]# auditctl -e 2
```

以上命令锁定 Audit 配置。

-r

设置每秒生成的消息率，例如：

```
~]# auditctl -r 0
```

以上配置对生成的消息没有设置速率限制。

-s

报告审计系统状态，例如：

```
~]# auditctl -s
AUDIT_STATUS: enabled=1 flag=2 pid=0 rate_limit=0 backlog_limit=8192 lost=259 backlog=0
```

-l

列出所有当前载入的审计规则，例如：

```
~]# auditctl -l
-w /etc/passwd -p wa -k passwd_changes
-w /etc/selinux -p wa -k selinux_changes
-w /sbin/insmod -p x -k module_insertion
⋮
```

-D

删除所有当前载入的审计规则，例如：

```
~]# auditctl -D
No rules
```

定义文件系统规则

要定义文件系统规则，请使用以下语法：

```
auditctl -w path_to_file -p permissions -k key_name
```

其中：

- path_to_file 是审计的文件或目录。
- 权限 是日志记录的权限：
  - r - 对文件或目录的读访问权限。
  - w - 对文件或目录的写入访问权限。
  - X - 执行对文件或目录的访问。
  - A - 更改文件或目录的属性。
- key_name 是一个可选字符串，可帮助您识别哪个规则或一组规则生成特定的日志条目。

例 7.1. 文件系统规则

要定义一条规则，记录对 /etc/passwd 文件的所有写访问以及 /etc/passwd 文件的每个属性，请执行以下命令：

```
~]# auditctl -w /etc/passwd -p wa -k passwd_changes
```

请注意，-k 选项后面的字符串是任意字符串。

要定义一条规则，记录对 /etc/selinux/ 目录中的所有文件的所有写入访问以及 /etc/selinux/ 目录中的所有文件，请执行以下命令：

```
~]# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

要定义一条规则，记录 /sbin/insmod 命令的执行，该命令可将模块插入到 Linux 内核中，请执行以下命令：

```
~]# auditctl -w /sbin/insmod -p x -k module_insertion
```

定义系统调用规则

要定义系统调用规则，请使用以下语法：

```
auditctl -a action,filter -S system_call -F field=value -k key_name
```

其中：

- action 和 filter 指定何时记录特定事件。操作 可以是 always 或 never。filter 指定将哪个内核规则匹配过滤器应用到事件。rule-matching 过滤器可以是以下之一：任务、退出、用户，以及 排除。有关这些过滤器的详情请参考开始 第 7.1 节 “审计系统架构”。
- system_call 按名称指定系统调用。所有系统调用的列表可在 /usr/include/asm/unistd_64.h 文件中找到。多个系统调用可以分组到一个规则中，每个规则都在其自身 -S 选项后指定。
- field=value 指定附加选项，进一步根据指定的架构、组 ID、进程 ID 等进一步修改规则来匹配事件。有关所有可用字段类型及其值的完整列表，请查看 auditctl(8) man page。
- key_name 是一个可选字符串，可帮助您识别哪个规则或一组规则生成特定的日志条目。

例 7.2. 系统调用规则

要定义一个规则，当程序每次使用 adjtimex 或 settimeofday 系统调用时创建日志条目，系统会使用 64 位架构，请使用以下命令：

```
~]# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
```

要定义一个规则，当 ID 为 1000 或更大的系统用户每次删除或重命名文件时，使用以下命令：

```
~]# auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F
auid!=4294967295 -k delete
```

请注意，-F auid!=4294967295 选项用于排除未设置登录 UID 的用户。

也可以使用系统调用规则语法定义文件系统规则。以下命令为系统调用创建一个类似于 -w /etc/shadow -p wa 文件系统规则的规则：

```
~]# auditctl -a always,exit -F path=/etc/shadow -F perm=wa
```

7.5.2. 定义可执行文件规则

要定义可执行文件规则，请使用以下语法：

```
auditctl  -a action,filter [ -F arch=cpu -S system_call] -F exe=path_to_executable_file -k key_name
```

其中：

- action 和 filter 指定何时记录特定事件。操作 可以是 always 或 never。filter 指定将哪个内核规则匹配过滤器应用到事件。rule-matching 过滤器可以是以下之一：任务、退出、用户，以及 排除。有关这些过滤器的详情请参考开始 第 7.1 节 “审计系统架构”。
- system_call 按名称指定系统调用。所有系统调用的列表可在 /usr/include/asm/unistd_64.h 文件中找到。多个系统调用可以分组到一个规则中，每个规则都在其自身 -S 选项后指定。
- path_to_executable_file 是审计的可执行文件的绝对路径。
- key_name 是一个可选字符串，可帮助您识别哪个规则或一组规则生成特定的日志条目。

例 7.3. 可执行文件规则

要定义一条规则，记录所有 /bin/id 程序的执行，请执行以下命令：

```
~]# auditctl -a always,exit -F exe=/bin/id -F arch=b64 -S execve -k execution_bin_id
```

7.5.3. 在 /etc/audit/audit.rules 文件中定义持久性审计规则和控制

要定义重启后保留的审计规则，您必须直接将其包含在 /etc/audit/audit.rules 文件中，或使用 augenrules 程序读取位于 /etc/audit/rules.d/ 目录中的规则。/etc/audit/audit.rules 文件使用相同的 auditctl 命令行语法来指定规则。哈希符号(#)后面的空行和文本将被忽略。

auditctl 命令也可用于使用 -R 选项从指定的文件中读取规则，例如：

```
~]# auditctl -R /usr/share/doc/audit/rules/30-stig.rules
```

定义控制规则

文件只能包含以下修改 Audit 系统行为的控制规则：-b、-D、-e、-f、--r、--loginuid-immutable 和 --backlog_wait_time。有关这些选项的详情请参考 “定义控制规则”一节。

例 7.4. audit.rules中的控制规则

```
# 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

# Make login UID immutable once it is set (may break containers)
--loginuid-immutable 1
```

定义文件系统和系统调用规则

文件系统和系统调用规则使用 auditctl 语法来定义。第 7.5.1 节 “使用 auditctl定义审计规则” 中的示

例可使用以下规则文件表示：

例 7.5. 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>=1000 -F auid!=4294967295 -k delete
```

预配置的规则文件

在 /usr/share/doc/audit/rules/ 目录中，audit 软件包会根据各种认证标准提供一组预配置的规则文件：

- 30-NISPOM.rules - 满足国家工业安全计划操作手册中信息系统安全章节中指定的要求的审计规则配置。
- 30-PCI-dss-v31.rules - 满足支付卡行业数据安全标准(PCI DSS) v3.1 要求的审计规则配置。
- 30-STIG.rules - 满足安全技术实施指南(STIG)设置的要求的审计规则配置。

要使用这些配置文件，请创建一个原始 /etc/audit/audit.rules 文件的备份，并在 /etc/audit/audit.rules 文件中复制您选择的配置文件：

```
~]# cp /etc/audit/audit.rules /etc/audit/audit.rules_backup
~]# cp /usr/share/doc/audit/rules/30-stig.rules /etc/audit/audit.rules
```

注意

审计规则有一个编号方案，允许排序它们。要了解更多有关命名方案的信息，请参阅 /usr/share/doc/audit/rules/README-rules 文件。

使用 augenrules 定义持久性规则

augenrules脚本读取位于/etc/audit/rules.d/目录下的规则，并将它们编译成audit.rures文件。此脚本会根据它们的自然排序顺序，以特定顺序处理以 .rules 结尾的所有文件。这个目录中的文件被组织成组，

其含义如下：

- 10 - 内核和 auditctl 配置
- 20 - 可与常规规则匹配但您希望不同匹配的规则
- 30 - 主规则
- 40 - 可选规则
- 50 - 特定于服务器的规则
- 70 - 系统本地规则
- 90 - 定稿（不可变）

规则并非是一次全部使用。它们是策略的一部分，应仔细考虑，并将单个文件复制到 /etc/audit/rules.d/。例如，要在 STIG 配置中设置系统，请复制规则 10-base-config、30-stig、31-privileged 和 99-finalize。

在 /etc/audit/rules.d/ 目录中有了规则之后，运行带有 --load 参数的 augenrules 脚本来加载它们：

```
~]# augenrules --load
augenrules --load No rules
enabled 1
failure 1
pid 634
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
enabled 1
failure 1
pid 634
```

```
rate_limit 0
backlog_limit 8192
lost 0
backlog 1
```

有关审计规则和 augenrules 脚本的更多信息，请参阅 audit.rules (8) 和 augenrules (8) 手册页。

7.6. 了解审计日志文件

默认情况下，审计系统将日志条目存储在 /var/log/audit/audit.log 文件中；如果启用了日志轮转，则轮转的 audit.log 文件也在存储同一个目录中。

以下审计规则记录每次尝试读取或修改 /etc/ssh/sshd_config 文件：

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

如果 auditd 守护进程正在运行，使用以下命令在审计日志文件中创建新事件，例如：

```
~]$ cat /etc/ssh/sshd_config
```

audit.log 文件中的该事件如下。

```
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=1000 uid=1000
gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 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
objtype=NORMAL cap_fp=none cap_fi=none cap_fe=0 cap_fver=0
type=PROCTITLE msg=audit(1364481363.243:24287) :
proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

以上事件由四个记录组成，它们共享相同的时间戳和序列号。记录始终以 type= 关键字开头。每个记录由多个 name=value 对组成，它们之间由空格或逗号分开。对上述事件的详细分析如下：

第一条记录

type=SYSCALL

type 字段包含记录的类型。在本例中，SYSCALL 值指定此记录是由对内核的系统调用触发的。

有关所有可能类型值及其解释的列表，请参阅 审计记录类型。

msg=audit(1364481363.243:24287):

msg 字段记录：

- 记录的时间戳和唯一 ID，格式为 audit(time_stamp:ID)。如果多个记录是作为同一审计事件的一部分而产生的，则它们共享相同的时间戳和 ID。时间戳使用 Unix 时间格式 - 自 1970 年 1 月 1 日 00:00:00 UTC 以来的秒数。
- 各种特定于事件的 name= 由内核或用户空间应用程序提供的值对。

arch=c000003e

arch 字段包含系统的 CPU 架构信息。该值 c000003e 以十六进制表示法编码。当使用 ausearch 命令搜索审计记录时，请使用 -i 或 --interpret 选项来自动将十六进制值转换成人类可读的等效值。c000003e 值被解释为 x86_64。

syscall=2

syscall字段记录了发送到内核的系统调用的类型。值 2 可以与 /usr/include/asm/unistd_64.h 文件中人类可读的等效值匹配。在本例中，2 是 打开 系统调用。请注意，ausyscall 工具允许您将系统调用号转换为人类可读的等效值。使用 ausyscall --dump 命令显示所有系统调用及其编号的列表。如需更多信息，请参阅 ausyscall(8) man page。

success=no

success 字段记录了该特定事件中记录的系统调用是成功还是失败。在这种情况下，调用不成功。

exit=-13

exit 字段包含一个值，指定系统调用返回的退出码。这个值因不同的系统调用而异。您可以使用以下命令将值解释成人类可读的等效值：

```
~]# ausearch --interpret --exit -13
```

请注意，上例假定您的审计日志包含一个失败的事件，其退出码为 -13 。

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

a0至a3字段记录了该事件中系统调用的前四个参数，用十六进制符号编码。这些参数取决于使用的系统调用，可以通过 ausearch 工具来解释它们。

items=1

items 字段包含系统调用记录后面的 PATH 辅助记录的数量。

ppid=2686

ppid 字段记录了父进程ID（PPID）。在这种情况下，2686 是父进程（如 bash）的 PPID 。

pid=3538

pid 字段记录了进程 ID（PID）。在本例中，3538 是 cat 进程的 PID。

auid=1000

auid字段记录了审计用户 ID，即loginuid。此 ID 在登录时分配给用户，并被每个进程继承，即使用户的身份有变化，例如，使用 su - john 命令切换用户帐户。

uid=1000

uid 字段记录了启动分析过程的用户的用户 ID。使用以下命令可以将用户 ID 解释成用户名：ausearch -i --uid UID。

gid=1000

gid 字段记录了启动分析过程的用户的组 ID。

euid=1000

euid 字段记录了启动分析过程的用户的有效用户 ID。

suid=1000

suid 字段记录了启动分析过程的用户的设置用户 ID。

fsuid=1000

fsuid 字段记录了启动分析进程的用户的文件系统用户 ID。

egid=1000

egid 字段记录了启动分析过程的用户的有效组 ID。

sgid=1000

sgid 字段记录了启动分析过程的用户的组 ID。

fsgid=1000

fsgid 字段记录了启动分析进程的用户的文件系统组 ID。

tty=pts0

tty 字段记录了分析过程被调用的终端。

ses=1

ses 字段记录了分析过程被调用的会话的会话 ID。

comm="cat"

comm 字段记录了用于调用分析过程的命令行名称。在本例中，cat 命令用于触发此审计事件。

exe="/bin/cat"

exe 字段记录了用于调用分析过程的可执行文件的路径。

subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

subj 字段记录了被分析的进程在执行时被标记的 SELinux 上下文。

key="sshd_config"

key 记录了与在审计日志中生成该事件的规则相关联的管理员定义的字符串。

第二条记录

type=CWD

在第二条记录中，type 字段值为 CWD - 当前工作目录。此类型用于记录从中调用第一条记录中指定的系统调用的进程的工作目录。

此记录的目的是记录当前进程的位置，以防在关联的 PATH 记录中捕获相对路径。这样，就可以重建绝对路径。

msg=audit(1364481363.243:24287)

msg 字段持有与第一条记录中的值相同的时间戳和 ID 值。时间戳使用 Unix 时间格式 - 自 1970 年 1 月 1 日 00:00:00 UTC 以来的秒数。

cwd="/home/user_name"

cwd 字段包含系统调用所在目录的路径。

第三条记录

type=PATH

在第三条记录中，type 字段值为 PATH。审计事件包含作为参数传递给系统调用的每个路径的 PATH 类型记录。在这个审计事件中，只有一个路径(/etc/ssh/sshd_config) 被用作参数。

msg=audit(1364481363.243:24287):

msg 字段拥有与第一和第二条记录中的值相同的时间戳和 ID 值。

item=0

item 字段表示在 SYSCALL 类型记录所引用的项目总数中，当前记录是哪个项目。这个数是以零为基础的；值为 0 表示它是第一项。

name="/etc/ssh/sshd_config"

name 字段记录了作为参数传递给系统调用的文件或目录的路径。在本例中，它是

/etc/ssh/sshd_config 文件。

inode=409248

inode 字段包含与该事件中记录的文件或目录相关联的 inode 号。以下命令显示与 409248 inode 号相关联的文件或目录：

```
~]# find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00

dev 字段指定了包含该事件中记录的文件或目录的设备的次要和主要 ID。在本例中，值表示 /dev/fd/0 设备。

mode=0100600

mode 字段记录文件或目录权限，由数字标记。它是 st_mode 字段中的 stat 命令返回。如需更多信息，请参阅 stat(2) 手册页。在这种情况下，0100600 可以解释为 -rw-------，这意味着只有 root 用户对 /etc/ssh/sshd_config 文件具有读和写的权限。

ouid=0

ouid 字段记录了对象所有者的用户 ID。

ogid=0

ogid 字段记录了对象所有者的组 ID。

rdev=00:00

rdev 字段包含一个记录的设备标识符，仅用于特殊文件。在这种情况下，不会使用它，因为记录的文件是一个常规文件。

obj=system_u:object_r:etc_t:s0

obj 字段记录了 SELinux 上下文，在执行时，记录的文件或目录被贴上了标签。

objtype=NORMAL

objtype 字段记录了给定系统调用的上下文中每个路径记录操作的意图。

cap_fp=none

cap_fp 字段记录了与设置文件或目录对象的基于文件系统的允许能力有关的数据。

cap_fi=none

cap_fi 字段记录了与文件或目录对象的基于继承文件系统的能力设置有关的数据。

cap_fe=0

cap_fe 字段记录了文件或目录对象基于文件系统能力的有效位的设置。

cap_fver=0

cap_fver 字段记录了文件或目录对象基于文件系统能力的版本。

第四条记录

type=PROCTITLE

type 字段包含记录的类型。在本例中，PROCTITLE 值指定此记录提供触发此审计事件的完整命令行，该事件是由对内核的系统调用触发的。

proctitle=636174002F6574632F7373682F737368645F636F6E666967

proctitle 字段记录了用于调用分析过程的命令的完整命令行。该字段采用十六进制表示法编码，不允许用户影响审计日志解析器。对触发此审计事件的命令进行文本解码。当使用 ausearch 命令搜索审计记录时，请使用 -i 或 --interpret 选项来自动将十六进制值转换成人类可读的等效值。636174002F6574632F7373682F737368645F636F6E666967 值解释为 cat /etc/ssh/sshd_config。

以上分析的审计事件仅包含事件可以包含的所有可能字段的子集。有关所有事件字段及其解释的列表，请参阅 审计事件字段。有关所有事件类型及其解释的列表，请参阅 审计记录类型。

例 7.6. 其他 audit.log 事件

以下审计事件记录了成功启动 auditd 守护进程。ver 字段显示启动的审计守护进程的版本。

```
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=1000 pid=4979 subj=unconfined_u:system_r:auditd_t:s0
res=success
```

以下审计事件记录了 UID 为 1000 的失败尝试，以 root 用户身份登录。

```
type=USER_AUTH msg=audit(1364475353.159:24270): user pid=3280 uid=1000 auid=1000
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'
```

7.7. 搜索审计日志文件

ausearch 工具允许您为特定事件搜索审计日志文件。默认情况下，ausearch 搜索 /var/log/audit/audit.log 文件。您可以使用 ausearch options -if file_name 命令指定不同的文件。在一个 ausearch 命令中提供多个选项等同于在字段类型和相同字段类型的多个实例间使用 AND 运算符。

例 7.7. 使用 ausearch 搜索审计日志文件

要搜索 /var/log/audit/audit.log 文件以失败的登录尝试，请使用以下命令：

```
~]# ausearch --message USER_LOGIN --success no --interpret
```

要搜索所有帐户、组和角色更改，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
```

要使用用户的登录 ID (uid)搜索特定用户执行的所有日志记录操作，请使用以下命令：

```
~]# ausearch -ua 1000 -i
```

要在现在前从 yesterday 搜索所有失败的系统调用，请使用以下命令：

```
~]# ausearch --start yesterday --end now -m SYSCALL -sv no -i
```

有关所有 ausearch 选项的完整列表，请查看 ausearch(8) man page。

7.8. 创建审计报告

aureport 工具允许您生成关于审计日志文件中记录的事件的摘要和列表报告。默认情况下，将查询 /var/log/audit/ 目录中的所有 audit.log 文件来创建报告。您可以使用 aureport options -if file_name 命令指定要针对运行报告的不同文件。

例 7.8. 使用 aureport 生成审计报告

要为过去 3 天（不包括当前 example 天）中的日志事件生成报告，请使用以下命令：

```
~]# aureport --start 04/08/2013 00:00:00 --end 04/11/2013 00:00:00
```

要生成所有可执行文件事件的报告，请使用以下命令：

```
~]# aureport -x
```

要生成上述可执行文件事件报告的摘要，请使用以下命令：

```
~]# aureport -x --summary
```

要为所有用户生成失败事件的摘要报告，请使用以下命令：

```
~]# aureport -u --failed --summary -i
```

要为每个系统用户生成所有失败的登录尝试的摘要报告，请使用以下命令：

```
~]# aureport --login --summary -i
```

要从 ausearch 查询生成报告，用于搜索用户 ID 1000 的所有文件访问事件，请使用以下命令：

```
~]# ausearch --start today --loginuid 1000 --raw | aureport -f --summary
```

要生成所有正在查询的审计文件的报告及其包含的事件范围，请使用以下命令：

```
~]# aureport -t
```

有关所有 aureport 选项的完整列表，请查看 aureport(8) man page。

7.9. 其它资源

有关审计系统的更多信息，请参阅以下源。

在线源

- RHEL Audit 系统 Refence: https://access.redhat.com/articles/4409591.
- Linux Audit 文档项目页面： https://github.com/linux-audit/audit-documentation/wiki

安装的文档

audit 软件包提供的文档可在 /usr/share/doc/audit/ 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 8 章 扫描系统以了解配置合规和漏洞

合规审计是一个确定给定对象是否遵循合规策略中指定的所有规则的流程。合规策略由安全专业人员定义的，他们通常以检查清单的形式指定计算环境应使用的必要设置。

跨组织甚至同一组织内不同系统之间的合规政策可能有很大差异。这些政策之间的差异取决于每个系统的用途及其对组织的重要性。自定义软件设置和部署特征也需要自定义策略检查表。

8.1. RHEL 中的配置合规工具

Red Hat Enterprise Linux 提供了可让您执行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，专为自动定制合规策略而设计。

- SCAP Workbench - scap-workbench 图形工具旨在对单个本地或远程系统执行配置和漏洞扫描。您还可以根据这些扫描和评估，使用它来生成安全报告。
- OpenSCAP - OpenSCAP 库以及附带的 oscap 命令行工具，旨在对本地系统执行配置和漏洞扫描，验证配置合规性内容，并根据这些扫描和评估生成报告和指南。
- SCAP 安全指南(SSG) - scap-security-guide 软件包为 Linux 系统提供最新的安全策略集合。该指南包括一个实用强化建议目录，在适用的情况下与政府的要求相关联。该项目弥补了一般性政策要求和具体实施指南间的差距。
- 脚本检查引擎(SCE) - SCE 是 SCAP 协议的扩展，可供管理员使用脚本语言（如 Bash、Python 和 Ruby）编写安全内容。SCE 扩展在 openscap-engine-sce 软件包中提供。SCE 本身不是 SCAP 环境的一部分。

要在多个系统上远程执行自动合规审计，您可以使用 Red Hat Satellite 的 OpenSCAP 解决方案。

其它资源

- oscap (8) - oscap 命令行工具的手册页提供了可用选项及其用法的完整列表。
- 红帽安全演示：创建自定义安全策略内容以自动化安全合规 - 一个动手实验室，使用 Red Hat Enterprise Linux 中包含的工具来获取自动化安全合规的初始经验，以符合行业标准安全策略和

自定义安全策略。如果您希望为您的团队提供培训或访问这些实验室练习，请联系您的红帽客户团队以了解更多详细信息。

- 红帽安全演示：使用 RHEL 安全技术保护自己 - 一个动手实验室，了解如何使用 Red Hat Enterprise Linux 中可用的关键安全技术（包括 OpenSCAP）在所有 RHEL 系统级别实施安全性。如果您希望为您的团队提供培训或访问这些实验室练习，请联系您的红帽客户团队以了解更多详细信息。

- scap-workbench (8) - SCAP Workbench 应用的手册页提供了有关应用程序的基本信息，以及 SCAP 内容潜在来源的链接。

- scap-security-guide (8) - scap-security-guide 项目的手册页提供了更多有关各种可用的 SCAP 安全配置集的文档。它还包含使用 OpenSCAP 工具提供的基准的示例。

- 管理 Red Hat Satellite 指南中的安全合规管理提供了有关将 OpenSCAP 与 Red Hat Satellite 搭配使用的更多详情。

8.2. 漏洞扫描

8.2.1. 红帽安全公告 OVAL Feed

Red Hat Enterprise Linux 安全审计功能基于安全内容自动化协议(SCAP)标准。SCAP 是一种多用途规格框架，支持自动化配置、漏洞和补丁检查、技术控制合规性活动和安全衡量。

SCAP 规范创建一个生态系统，其中安全内容的格式是众所周知的且标准化的，尽管扫描程序或策略编辑器的实现并不是强制性的。这使得组织能够一次性构建它们的安全策略（SCAP 内容），无论他们使用了多少家安全供应商。

开放式漏洞评估语言(OVAL)是 SCAP 最基本、最古老的组件。与其他工具和自定义脚本不同，OVAL 以声明式方法描述资源的必需状态。OVAL 代码从不直接执行，而是使用称为扫描器的 OVAL 解释器工具。OVAL 的声明性质可确保评估的系统状态不会被意外修改。

与所有其他 SCAP 组件一样，OVAL 也是基于 XML。SCAP 标准定义了多个文档格式。每一个都包括一种不同的信息，用于不同的目的。

红帽产品安全团队通过跟踪和调查影响红帽客户的所有安全问题，帮助客户评估和管理风险。它在红帽客户门户网站中提供及时、简洁的补丁和安全公告。红帽创建和支持 OVAL 补丁定义，提供机器可读

的安全公告版本。

由于平台、版本和其他因素之间的差异，红帽产品安全团队的严重性评级 不会直接与第三方提供的通用漏洞评分系统(CVSS)基准评级一致。因此，我们建议您使用 RHSA OVAL 定义，而不是第三方提供的定义。

RHSA OVAL 定义 单独提供，并作为一个完整的软件包提供，并在红帽客户门户网站上提供新安全公告的一小时内进行更新。

每个 OVAL 补丁定义将一对一映射到红帽安全公告(RHSA)。由于 RHSA 可以包含对多个漏洞的修复，因此每个漏洞都通过其通用漏洞和风险(CVE)名称单独列出，并在我们的公共 bug 数据库中有一个指向其条目的链接。

RHSA OVAL 定义旨在检查系统上安装的 RPM 软件包是否存易受攻击的版本。可以扩展这些定义以包括进一步的检查，例如，查找软件包是否在易受攻击的配置中被使用。这些定义旨在涵盖红帽提供的软件和更新。需要其他定义来检测第三方软件的补丁状态。

注意

要扫描容器或容器镜像以了解安全漏洞，请参阅 第 8.9 节 “扫描容器和容器镜像中的漏洞”。

其它资源

- 红帽和 OVAL 兼容性
- 红帽和 CVE 兼容性
- 产品安全概述中的通知和建议
- 安全数据指标
- 第 8.9 节 “扫描容器和容器镜像中的漏洞”

8.2.2. 扫描系统是否存在漏洞

oscap命令行实用程序使您能够扫描本地系统，验证配置合规性内容，并根据这些扫描和评估生成报告和指南。此工具充当 OpenSCAP 库的前端，并根据它所处理的 SCAP 内容类型将其功能分组到模块（子命令）。

流程

1. 安装 openscap-scanner 和 bzip2 软件包：

```
~]# yum install openscap-scanner bzip2
```

2. 下载系统的最新 RHSA OVAL 定义，例如：

```
~]# wget -O - https://www.redhat.com/security/data/oval/v2/RHEL7/rhel-7.oval.xml.bz2 |
bzip2 --decompress > rhel-7.oval.xml
```

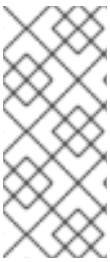
3. 扫描系统漏洞并将结果保存到 vulnerability.html 文件中：

```
~]# oscap oval eval --report vulnerability.html rhel-7.oval.xml
```

验证

1. 在您选择的浏览器中检查结果，例如：

```
~]$ firefox vulnerability.html &
```



注意

CVE OVAL 检查会搜索漏洞。因此，结果"True"意味着系统存在安全漏洞，而 "False" 表示扫描找不到任何漏洞。在 HTML 报告中，这可以通过结果行的颜色进一步区分。

其它资源

- oscap (8) 手册页。

- Red Hat OVAL 定义 列表。

8.2.3. 扫描远程系统中的漏洞

您还可以使用通过 SSH 协议的 oscap-ssh 工具，使用 OpenSCAP 扫描程序来检查远程系统的漏洞。

先决条件

- openscap-scanner 软件包安装在远程系统上。
- SSH 服务器在远程系统上运行。

流程

1. 安装 openscap-utils 和 bzip2 软件包：

```
~]# yum install openscap-utils bzip2
```

2. 下载系统的最新 RHSA OVAL 定义：

```
~]# wget -O - https://www.redhat.com/security/data/oval/v2/RHEL7/rhel-7.oval.xml.bz2 |
bzip2 --decompress > rhel-7.oval.xml
```

3. 扫描 SSH 在端口 22 上运行、用户名为 joesec、主机名为 machine1 的远程系统上的漏洞，并将结果保存到 remote-vulnerability.html 文件中：

```
~]# oscap-ssh joesec@machine1 22 oval eval --report remote-vulnerability.html rhel-
7.oval.xml
```

其它资源

- oscap-ssh (8) 手册页。
- Red Hat OVAL 定义 列表。

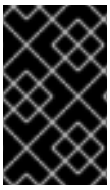
8.3. 配置合规性扫描

8.3.1. RHEL 7 中的配置合规性

您可以使用配置合规性扫描来遵循特定组织定义的基准。例如，如果您与美国政府合作，您可能需要遵守操作系统保护配置文件(OSPP)，如果您是一个支付处理商，您可能必须遵循支付卡行业数据安全标准(PCI-DSS)。您还可以执行配置合规性扫描来强化您的系统安全。

红帽建议您遵循 SCAP 安全指南软件包中提供的安全内容自动化协议(SCAP)内容，因为它符合红帽针对受影响组件的最佳实践。

SCAP 安全指南软件包提供了符合 SCAP 1.2 和 SCAP 1.3 标准的内容。openscap 扫描器实用程序与SCAP安全指南包中提供的SCAP 1.2和SCAP 1.3内容兼容。



重要

执行配置合规性扫描不能保证系统是合规的。

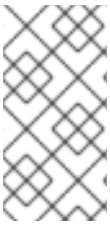
SCAP 安全指南套件以数据流文档的形式为多个平台提供配置文件。数据流是包含定义、基准、配置文件和单个规则的文件。每条规则都规定了合规的适用性和要求。RHEL 7 提供多个配置文件来满足安全策略要求。除了行业标准之外，红帽数据流还包含用于修复失败规则的信息。

合规性扫描资源的结构

```
Data stream
  ├── xccdf
  |     ├── benchmark
  |         ├── profile
  |           ├──rule
  |             ├── xccdf
  |               ├── oval reference
  ├── oval            ├── ocil reference
  ├── ocil            ├── cpe reference
  └── cpe             └── remediation
```

配置文件是基于安全策略的一组规则，如操作系统保护配置文件(OSPP)或支付卡行业数据安全标准(PCI-DSS)。这可让您以自动化的方式审核系统，以符合安全标准。

您可以修改（定制）配置文件来自定义某些规则，例如密码长度。有关配置集定制的更多信息，请参阅 第 8.7.2 节 “使用 SCAP Workbench 自定义安全配置文件”



注意

要扫描容器或容器镜像以了解配置合规性，请参阅 第 8.9 节 “扫描容器和容器镜像中的漏洞”

8.3.2. OpenSCAP 扫描的可能结果

根据您的系统的不同属性以及应用于 OpenSCAP 扫描的数据流和配置文件，每个规则可能会产生特定的结果。这是一个可能的结果列表，并简要解释了它们的含义。

表 8.1. OpenSCAP 扫描的可能结果

结果	介绍
Pass	扫描没有发现与此规则有任何冲突。
Fail	扫描发现与此规则有冲突。
Not checked	OpenSCAP 对此规则不执行自动评估。手动检查您的系统是否符合此规则。
Not applicable	此规则不适用于当前配置。
Not selected	此规则不是配置文件的一部分。OpenSCAP 不评估此规则，也不会在结果中显示这些规则。
Error	扫描遇到了错误。如需更多信息，您可以输入带有 **-verbose DEVEL** 选项的 **oscap-scanner** 命令。考虑打开 bug 报告。
Unknown	扫描遇到了意外情况。如需更多信息，您可以输入带有 **-verbose DEVEL** 选项的 **oscap-scanner** 命令。考虑打开 bug 报告。

8.3.3. 查看配置合规性的配置集

在决定使用配置文件进行扫描或修复前，您可以使用 **oscap info** 子命令列出它们并检查其详细描述。

先决条件

- 已安装 **openscap-scanner** 和 **scap-security-guide** 软件包。

流程

1. 列出 SCAP 安全指南项目提供的带有配置合规配置文件的所有可用文件：

```
~]$ ls /usr/share/xml/scap/ssg/content/
ssg-firefox-cpe-dictionary.xml  ssg-rhel6-ocil.xml
ssg-firefox-cpe-oval.xml        ssg-rhel6-oval.xml
...
ssg-rhel6-ds-1.2.xml            ssg-rhel8-xccdf.xml
ssg-rhel6-ds.xml
...
```

2. 使用 **oscap info** 子命令显示关于所选数据流的详细信息。包含数据流的 **XML** 文件由其名称中的 **-ds** 字符串表示。在 **Profiles** 部分，您可以找到可用的配置文件及其 **ID** 列表：

```
~]$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
...
Profiles:
 Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 7
  Id: xccdf_org.ssgproject.content_profile_pci-dss
 Title: OSPP - Protection Profile for General Purpose Operating Systems v. 4.2.1
  Id: xccdf_org.ssgproject.content_profile_ospp
...
```

3. 从数据流文件中选择一个配置文件，并显示所选配置文件的更多详情。为此，请使用带有 **--profile** 选项的 **oscap info**，后跟上一命令输出中显示的 **ID** 的后缀。例如，**PCI-DSS** 配置集的 **ID** 为：**xccdf_org.ssgproject.content_profile_pci-dss**，**--profile** 选项的值可以是 **_pci-dss**：

```
~]$ oscap info --profile _pci-dss /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
...
Profile
 Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 7
 Id: xccdf_org.ssgproject.content_profile_pci-dss

 Description: Ensures PCI-DSS v3.2.1 related security configuration settings are applied.
...
```

4. 另外，在使用 **GUI** 时，安装 **scap-security-guide-doc** 软件包并在网页浏览器中打开 **file:///usr/share/doc/scap-security-guide-doc-0.1.46/ssg-rhel7-guide-index.html** 文件。在 指南的右上方选择 **Red Hat Enterprise Linux 7** 安全配置 文档中的所需配置文件，您可以为后续评估在相关命令中看到已包含的 **ID**。

其它资源

- scap-security-guide (8) 手册页还包含配置集列表。

8.3.4. 使用特定 Baseline 评估配置合规

要确定您的系统是否符合特定基准，请按照以下步骤操作：

先决条件

- 已安装 openscap-scanner 和 scap-security-guide 软件包。
- 您知道系统应遵守的基准中的配置文件的 ID。要查找 ID，请参阅 第 8.3.3 节 “查看配置合规性的配置集”。

流程

1. 使用所选配置文件评估系统的合规性，并将扫描结果保存在 report.html HTML 文件中，例如：

```
~]$ sudo oscap xccdf eval --report report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

2. 可选：扫描带有 machine1 主机名、在端口 22 上运行的 SSH 的远程系统，以及 joesec 用户名中的漏洞，并将结果保存到 remote-report.html 文件中：

```
~]$ oscap-ssh joesec@machine1 22 xccdf eval --report remote_report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

其它资源

- scap-security-guide(8) man page
- SCAP 安全指南 文档安装在 file:///usr/share/doc/scap-security-guide-doc-0.1.46/ 目录中。
- 安装了 scap-security-guide-doc 软件包的 Red Hat Enterprise Linux 7 安全配置指南。

8.4. 使用特定基本线将系统修复到 ALIGN

使用这个流程修复 RHEL 7 系统，使其与特定基准一致。这个示例使用保护配置文件进行通用目的操作系统(OSPP)。

警告

如果不小心使用，在启用了 Remediate 选项的情况下运行系统评估可能会导致系统无法正常工作。红帽不提供任何自动的方法来恢复由安全强化补救所做的更改。默认配置的 RHEL 系统支持自动安全补救功能。如果在安装后更改了您的系统，运行补救可能无法使其与所需安全配置兼容。

先决条件

- scap-security-guide 软件包安装在 RHEL 7 系统中。

流程

1. 使用带有 --remediate 选项的 oscap 命令：

```
~]$ sudo oscap xccdf eval --profile ospp --remediate /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

2. 重启您的系统。

验证

1. 使用 OSPP 配置集评估系统的合规性，并将扫描结果保存到 ospp_report.html 文件中：

```
~]$ oscap xccdf eval --report ospp_report.html --profile ospp /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

其它资源

- scap-security-guide(8) 和 oscap(8) 手册页

8.5. 使用 SSG ANSIBLE PLAYBOOK 修复系统以使用特定基础行 ALIGN

使用 SCAP 安全指南项目中的 Ansible playbook 文件，使用此流程使用特定基准修复您的系统。这个示例使用保护配置文件进行通用目的操作系统(OSPP)。

警告

如果不小心使用，在启用了 Remediate 选项的情况下运行系统评估可能会导致系统无法正常工作。红帽不提供任何自动的方法来恢复由安全强化补救所做的更改。默认配置的 RHEL 系统支持自动安全补救功能。如果在安装后更改了您的系统，运行补救可能无法使其与所需安全配置兼容。

先决条件

- scap-security-guide 软件包安装在 RHEL 7 系统中。
- 已安装 ansible 软件包。如需更多信息，请参阅 Ansible 安装指南。

流程

1. 使用 Ansible 修复您的系统，使其与 OSPP 一致：

```
~]# ansible-playbook -i localhost, -c local /usr/share/scap-security-guide/ansible/ssg-rhel7-role-ospp.yml
```

2. 重新启动系统。

验证

1. 使用 OSPP 配置集评估系统的合规性，并将扫描结果保存到 ospp_report.html 文件中：

```
~]# oscap xccdf eval --profile ospp --report ospp_report.html /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

其它资源

- *scap-security-guide(8) 和 oscap(8) 手册页*
- *Ansible 文档*

8.6. 创建修复 ANSIBLE PLAYBOOK 以选择具有特定基础的系统

使用这个流程创建一个 Ansible playbook，它只包含使您的系统与特定基准保持一致所需的补救。这个示例使用保护配置文件进行通用目的操作系统(OSPP)。通过这个过程，您可以创建一个较小的 playbook，其不包括已经满足的需求。按照以下步骤，您不需要以任何方式修改您的系统，您只需为后续应用程序准备一个文件。

先决条件

- *scap-security-guide 软件包安装在您的系统中。*

流程

1. *扫描系统并保存结果：*

```
~]# oscap xccdf eval --profile ospp --results ospp-results.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

2. *根据上一步中生成的文件生成一个 Ansible playbook：*

```
~]# oscap xccdf generate fix --fix-type ansible --profile ospp --output ospp-remediations.yml
ospp-results.xml
```

3. *ospp-remediations.yml 文件包含对在第 1 步中执行扫描过程中失败的规则的 Ansible 修复。查看生成的文件后，您可以使用 ansible-playbook ospp-remediations.yml 命令应用该文件。*

验证

1. *在您选择的文本编辑器中，检查 ospp-remediations.yml 文件是否包含在第 1 步中执行的扫描中失败的规则。*

其它资源

- *scap-security-guide(8) 和 oscap(8) 手册页*
- *Ansible 文档*

8.7. 使用 SCAP WORKBENCH 使用自定义配置文件扫描系统

SCAP Workbench 是一个图形实用程序，可让您在单个本地或远程系统上执行配置扫描，对系统执行修复，并根据扫描评估生成报告。请注意，与 oscap 命令行工具相比，SCAP Workbench 的功能有限。SCAP Workbench 以数据流文件的形式处理安全内容。

8.7.1. 使用 SCAP Workbench 扫描和补救系统

要针对所选的安全策略评估您的系统，请使用以下流程。

先决条件

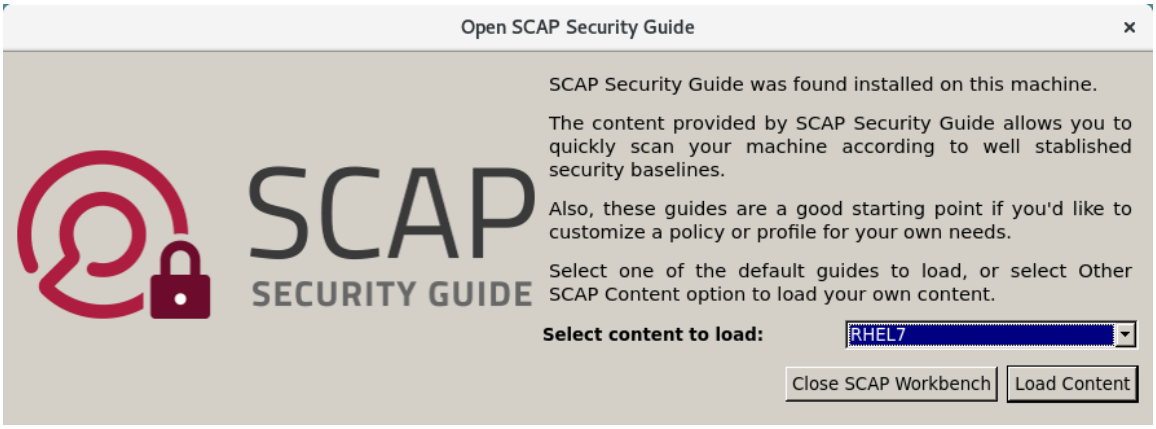
- *scap-workbench 软件包安装在您的系统中。*

流程

1. *要从 GNOME Classic 桌面环境运行 SCAP Workbench，请按 Super 键进入 Activities Overview，输入 scap-workbench，然后按 Enter。或者，使用：*

```
~]$ scap-workbench &
```

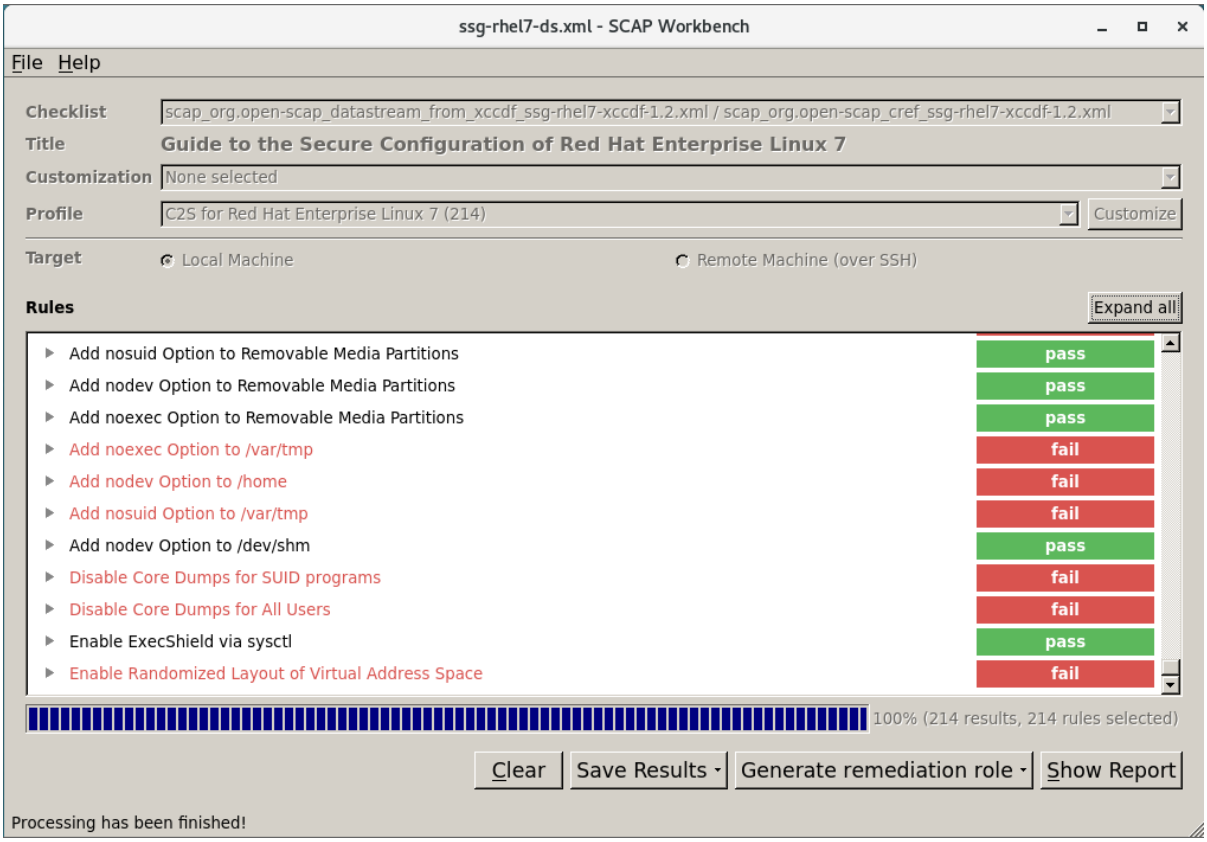
2. *使用以下任一选项选择安全策略：*
   - *开始窗口中的 Load Content 按钮*
   - *打开 SCAP 安全指南中的内容*
   - *在 File 中打开 Other Content，搜索相关的 XCCDF、SCAP RPM 或数据流文件。*



3. *您可以选择 **Remediate** 复选框来启用系统配置自动修正。启用此选项后，**SCAP Workbench** 会尝试根据策略所应用的安全规则来修改系统配置。这个过程会尝试修复系统扫描过程中失败的相关检查。*

> **警告**
>
> *如果不小心使用，在启用了 **Remediate** 选项的情况下运行系统评估可能会导致系统无法正常工作。红帽不提供任何自动的方法来恢复由安全强化补救所做的更改。默认配置的 **RHEL** 系统支持自动安全补救功能。如果在安装后更改了您的系统，运行补救可能无法使其与所需安全配置兼容。*

4. *单击**Scan**按钮，使用所选配置文件扫描您的系统。*



5. *要以 XCCDF、ARF 或 HTML 文件的形式保存扫描结果，请点击 Save Results 组合框。选择 HTML Report 选项，以人类可读的格式生成扫描报告。XCCDF 和 ARF（数据流）格式适合进一步自动处理。您可以重复选择所有三个选项。*

6. *要将基于结果的补救导出到文件，请使用 Generate remediation role 弹出菜单。*

8.7.2. 使用 SCAP Workbench 自定义安全配置文件

您可以通过更改某些规则中的参数（如最小密码长度）、删除以不同方式涵盖的规则，并选择额外的规则来自定义安全配置文件，以实现内部策略。您不能通过自定义配置文件来定义新规则。

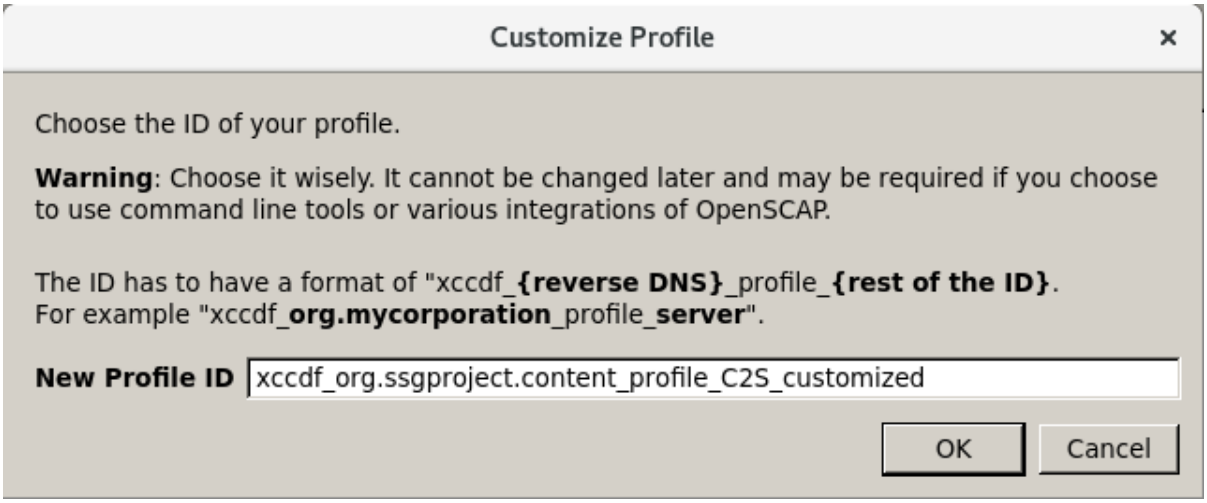
以下流程演示了如何使用 SCAP Workbench 来自定义（定制）配置文件。您还可以保存定制的配置文件，以便在 oscap 命令行工具中使用。。

流程

1. *运行 SCAP Workbench，然后使用 Open content from SCAP Security Guide 或 Open Other Content in the File 菜单中选择您要自定义的配置集。*

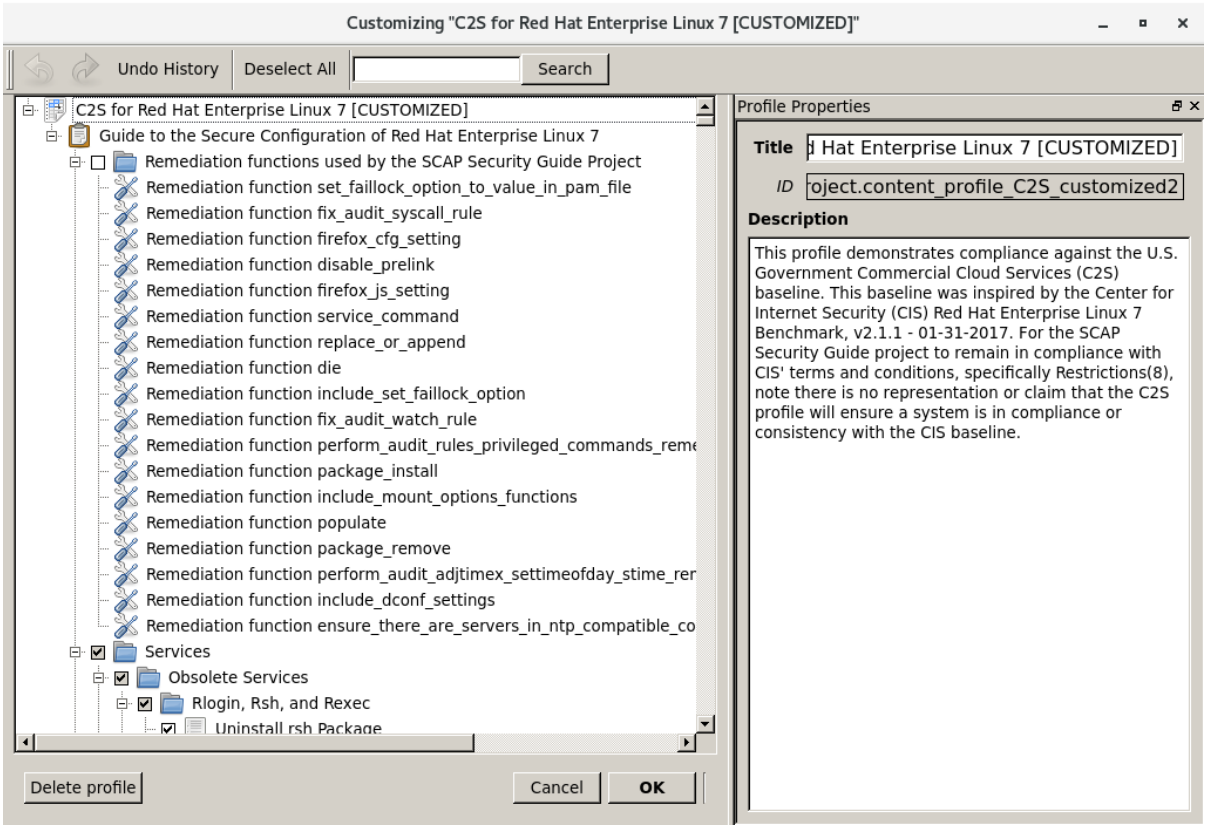
2. *要根据您的需要调整所选的安全配置文件，请点击 **Customize** 按钮。*

*这会打开新的 **Customization** 窗口，允许您在不更改原始 **XCCDF** 文件的情况下修改当前选择的 **XCCDF** 配置集。选择新的配置文件 **ID**。*



3. *使用将规则组织成逻辑组的树结构或 **Search** 字段查找要修改的规则。*

4. *使用树结构中的复选框来包含或排除规则，或者在适用情况下修改规则中的值。*



5. 点击 OK 按钮以确认修改。

6. 要永久存储您的修改，请使用以下选项之一：

   - 使用 File 菜单中的 Save Customization Only 分别保存自定义文件。
   - 使用 File 菜单中的 Save All 一次保存所有安全内容。

   如果您选择了 Into a directory 选项，SCAP Workbench 会将 XCCDF 或数据流文件以及自定义文件保存到指定位置。您可以使用它作为备份解决方案。

   通过选择 As RPM 选项，您可以指示 SCAP Workbench 创建包含数据流文件和自定义文件的 RPM 软件包。这对于将安全内容分发到无法远程扫描的系统以及交付内容以供进一步处理非常有用。

注意

因为 SCAP Workbench 不支持对定制配置文件的基于结果的补救，所以请使用 oscap 命令行工具导出的补救。

8.7.3. 相关信息

- scap-workbench(8) 手册页
- SCAP Workbench 用户手册
- 使用 Satellite 6.x 部署自定义 SCAP 策略 - 关于定制脚本的知识库文章

8.8. 在安装后使用安全配置文件 IMMEDIATELY 部署 ARE COMPLIANT 的系统

您可以在安装过程后立即使用 OpenSCAP 套件部署符合安全配置集（如 OSPP 或 PCI-DSS）的 RHEL 系统。使用此部署方法，您可以应用以后无法使用修复脚本应用的特定规则，例如，密码强度和分区规则。

8.8.1. 使用图形安装部署 Baseline-Compliant RHEL 系统

使用此流程部署与特定基准兼容的 RHEL 系统。这个示例为常规目的的操作系统(OSPP)使用保护配置集。

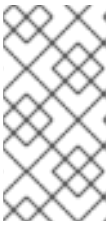
先决条件

- 您已引导到 图形化 安装程序。请注意，OSCAP Anaconda 附加组件 不支持纯文本安装。
- 您已访问 安装概述 窗口。

流程

1. 在 安装概述 窗口中点击 软件选择。此时会打开 软件选择窗口。
2. 在 Base Environment 窗格中选择 服务器 环境。您只能选择一个基本环境。
3. 点击 完成 应用设置并返回 安装概述 窗口。
4. 点击 安全策略。此时会打开 Security Policy 窗口。
5. 要在系统中启用安全策略，将Apply security policy 切换为 ON。
6. 从配置集栏中选择 Protection Profile for General Purpose Operating Systems.
7. 点 Select Profile 来确认选择。
8. 确认在窗口底部显示 Changes that were done or need to be done。完成所有剩余的手动更改。
9. 因为 OSPP 有必须满足的严格的分区要求，所以可以为 /boot、/home、/var、/var/log、/var/tmp 和 /var/log/audit 创建单独的分区。

10. 完成图形安装过程。



注意

图形安装程序在安装成功后自动创建对应的 Kickstart 文件。您可以使用 /root/anaconda-ks.cfg 文件自动安装兼容 OSPP 的系统。

验证

1. 要在安装完成后检查系统当前的状态,请重启系统并启动新的扫描：

```
~]# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

其它资源

- 有关分区的详情，请参阅 配置手动分区。

8.8.2. 使用 Kickstart 部署 Baseline-Compliant RHEL 系统

使用此流程部署符合特定基准的 RHEL 系统。这个示例为常规目的的操作系统(OSPP)使用保护配置集。

先决条件

- scap-security-guide 软件包安装在您的系统中。

流程

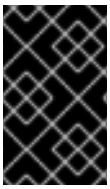
1. 在您选择的编辑器中打开 /usr/share/scap-security-guide/kickstart/ssg-rhel7-ospp-ks.cfg Kickstart 文件。

2. 更新分区方案以符合您的配置要求。对于 OSPP 合规性，必须保留 /boot、/home、/var /log、/var/log、/var/tmp 和 /var/log/audit 的独立分区，但您可以更改这些分区的大小。

警告

因为 OSCAP Anaconda 附加组件 不支持只使用文本安装，请不要在 Kickstart 文件中使用 text 选项。如需更多信息，请参阅 RHBZmvapich4001。

3. 按照 使用 Kickstart 执行自动安装 中所述来开始 Kickstart 安装。



重要

使用哈希格式的密码无法检测 OSPP 要求。

验证

1. 要在安装完成后检查系统当前的状态,请重启系统并启动新的扫描：

```
~]# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

其它资源

- 详情请查看 OSCAP Anaconda Add-on 项目页面。

8.9. 扫描容器和容器镜像中的漏洞

使用以下步骤查找容器或容器镜像中的安全漏洞。

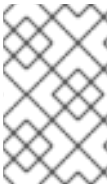
您可以使用 oscap-docker 命令行工具或 atomic 扫描 命令行实用程序来查找容器或容器镜像中的安全漏洞。

使用 oscap-docker 时，您可以使用 oscap 程序扫描容器镜像和容器。

通过 原子扫描，您可以使用 OpenSCAP 扫描功能来扫描系统上的容器镜像和容器。您可以扫描已知 CVE 漏洞以及配置合规性。另外，您还可以将容器镜像修复到指定的策略。

8.9.1. 使用 oscap-docker扫描容器镜像和容器中的漏洞

您可以使用 oscap-docker 工具扫描容器和容器镜像。



注意

oscap-docker 命令需要 root 特权，容器的 ID 是第二个参数。

先决条件

- 已安装 openscap-containers 软件包。

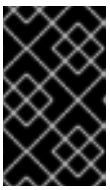
流程

1. 查找容器或容器镜像的 ID，例如：

```
~]# docker images
REPOSITORY                          TAG      IMAGE ID       CREATED       SIZE
registry.access.redhat.com/ubi7/ubi   latest   096cae65a207   7 weeks ago   239 MB
```

2. 扫描容器或容器镜像的漏洞，并将结果保存到 vulnerability.html 文件中：

```
~]# oscap-docker image-cve 096cae65a207 --report vulnerability.html
```



重要

要扫描容器，请将 image-cve 参数替换为 container-cve。

验证

1. 在您选择的浏览器中检查结果，例如：

```
~]$ firefox vulnerability.html &
```

其它资源

- *如需更多信息，请参阅 oscap-docker (8) 和 oscap (8) 手册页。*

8.9.2. 使用 原子扫描扫描容器镜像和容器中的漏洞

使用 atomic 扫描 实用程序，您可以扫描容器和容器镜像，以了解红帽发布的 CVE OVAL 定义中所定义的已知安全漏洞。atomic scan 命令的格式如下：

```
~]# atomic scan [OPTIONS] [ID]
```

其中 ID 是您要扫描的容器镜像或容器的 ID。

警告

atomic 扫描 功能已弃用，OpenSCAP 容器镜像不再针对新的漏洞更新。因此，首选 oscap-docker 工具进行漏洞扫描。

使用案例

- *要扫描所有容器镜像，请使用 --images 指令。*
- *要扫描所有容器，请使用 --containers 指令。*
- *要扫描这两种类型，请使用 --all 指令。*
- *若要列出所有可用的命令行选项，可使用 atomic scan --help 命令。*

atomic scan 命令的默认扫描类型是 CVE 扫描。使用它来检查 红帽发布的 CVE OVAL 定义 中定义的已知安全漏洞的目标。

先决条件

- 您已使用 atomic install rhel7/openscap 命令，从 红帽容器目录(RHCC) 下载并安装 OpenSCAP 容器镜像。

流程

1. 验证您是否具有最新的 OpenSCAP 容器镜像，以确保定义是最新的：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap | grep version
```

2. 扫描带有几个已知的安全漏洞的 RHEL 7.2 容器镜像：

```
~]# atomic scan registry.access.redhat.com/rhel7:7.2
docker run -t --rm -v /etc/localtime:/etc/localtime -v /run/atomic/2017-11-01-14-49-36-614281:/scanin -v /var/lib/atomic/openscap/2017-11-01-14-49-36-614281:/scanout:rw,Z -v /etc/oscapd:/etc/oscapd:ro registry.access.redhat.com/rhel7/openscap oscapd-evaluate scan --no-standard-compliance --targets chroots-in-dir:///scanin --output /scanout

registry.access.redhat.com/rhel7:7.2 (98a88a8b722a718)

The following issues were found:

 RHSA-2017:2832: nss security update (Important)
 Severity: Important
  RHSA URL: https://access.redhat.com/errata/RHSA-2017:2832
  RHSA ID: RHSA-2017:2832-01
  Associated CVEs:
    CVE ID: CVE-2017-7805
    CVE URL: https://access.redhat.com/security/cve/CVE-2017-7805
...
```

其它资源

- Red Hat Enterprise Linux Atomic Host 产品文档包含 atomic 命令用法和容器的详细描述。

- 红帽客户门户提供了 Atomic 命令行界面(CLI)的指南。

8.10. 评估容器或带有特定基本行的容器镜像的配置合规性

按照以下步骤，使用特定安全基线评估容器或容器镜像的合规性，如操作系统保护配置文件(OSPP)或

支付卡行业数据安全标准(PCI-DSS)。

先决条件

- 已安装 openscap-utils 和 scap-security-guide 软件包。

流程

1. 查找容器或容器镜像的 ID，例如：

```
~]# docker images
REPOSITORY                          TAG      IMAGE ID       CREATED       SIZE
registry.access.redhat.com/ubi7/ubi   latest   096cae65a207   7 weeks ago   239 MB
```

2. 使用 OSPP 配置集评估容器镜像的合规性，并将扫描结果保存到 report.html HTML 文件中。

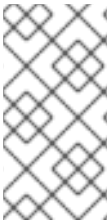
```
~]$ sudo oscap-docker 096cae65a207 xccdf eval --report report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

如果您评估配置符合 PCI-DSS 基准，请将 096cae65a207 替换为您的容器镜像 ID，将 ospp 值替换为 pci-dss。

验证

1. 在您选择的浏览器中检查结果，例如：

```
~]$ firefox report.html &
```



注意

标记为 notapplicable 的规则是不适用于容器化系统的规则。这些规则仅适用于裸机或虚拟化系统。

其它资源

- 如需更多信息，请参阅 oscap-docker (8) 和 scap-security-guide (8) 手册页。

- SCAP 安全指南 文档安装在 file:///usr/share/doc/scap-security-guide-doc-0.1.46/ 目录中。

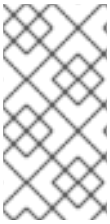
8.11. 使用 原子扫描扫描对容器镜像和容器进行扫描和补救配置合规性

8.11.1. 使用 原子扫描扫描来扫描容器镜像和容器的配置合规性

使用这种类型的扫描来评估基于 Red Hat Enterprise Linux 的容器镜像和容器，以及 SCAP 安全指南(SSG)在 OpenSCAP 容器镜像中提供的 SCAP 内容。这可启用对 SCAP 安全指南提供的任何配置集进行扫描。

警告

atomic 扫描 功能已弃用，OpenSCAP 容器镜像不再使用新的安全合规内容更新。因此，首选 oscap-docker 工具用于安全合规性扫描。



注意

有关使用 atomic 命令和容器的详情，请查看 Red Hat Enterprise Linux Atomic Host 7 产品文档。红帽客户门户网站还为 atomic 命令行界面(CLI)提供指南。

先决条件

- 您已使用 atomic install rhel7/openscap 命令，从 红帽容器目录(RHCC) 下载并安装 OpenSCAP 容器镜像。

流程

1. 列出 OpenSCAP 镜像为 configuration_compliance 扫描提供的 SCAP 内容：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap
```

使用 Defense Information Systems Agency Security Technical Implementation Guide (DISA STIG)策略验证最新 Red Hat Enterprise Linux 7 容器镜像合规性，并从扫描中生成 HTML 报告：

```
~]# atomic scan --scan_type configuration_compliance --scanner_args xccdf-
id=scap_org.open-scap_cref_ssg-rhel7-xccdf-
1.2.xml,profile=xccdf_org.ssgproject.content_profile_stig-rhel7-disa,report
registry.access.redhat.com/rhel7:latest
```

以上命令的输出包含有关末尾扫描关联的文件的信息：

```
.....

Files associated with this scan are in /var/lib/atomic/openscap/2017-11-03-13-35-34-296606.

~]# tree /var/lib/atomic/openscap/2017-11-03-13-35-34-296606
/var/lib/atomic/openscap/2017-11-03-13-35-34-296606
├── db7a70a0414e589d7c8c162712b329d4fc670fa47ddde721250fb9fcdbed9cc2
│   ├── arf.xml
│   ├── fix.sh
│   ├── json
│   └── report.html
└── environment.json

1 directory, 5 files
```

atomic 扫描生成含有所有结果的子目录，并从 **/var/lib/atomic/openscap/** 目录中的扫描报告。每次扫描配置合规性时都会生成带有结果的 **arf.xml** 文件。要生成人类可读的 **HTML** 报告文件，请将 **报告** 子选项添加到 **--scanner_args** 选项。

2. 可选： 要生成由 **DISA STIG Viewer** 读取的 **XCCDF** 结果，请将 **stig-viewer** 子选项添加到 **--scanner_args** 选项中。结果放置在 **stig.xml** 中。

注意

当省略 **--scanner_args** 选项的 **xccdf-id** 子选项时，扫描程序会在所选数据流文件的第一个 **XCCDF** 组件中搜索配置集。有关数据流文件的详情，请参考 第 8.3.1 节 “RHEL 7 中的配置合规性”。

8.11.2. 使用 原子扫描修复容器镜像和容器的配置合规性

您可以针对原始容器镜像运行配置合规性扫描，以检查其与 **DISA STIG** 策略的合规性。根据扫描结果，会生成包含失败扫描结果的 **bash** 补救的修复脚本。然后，修复脚本会应用到原始容器镜像 **-** 这称为补救。补救会导致容器镜像具有更改的配置，该配置在原始容器镜像之上作为新层添加。

重要

请注意，原始容器镜像保持不变，且仅在其之上创建一个新层。补救过程会构建包含所有配置改进的新容器镜像。此层的内容由扫描的安全策略定义，在上例中是 DISA STIG 策略。这也意味着修复的容器镜像不再由红帽签名，因为它与包含修复层的原始容器镜像不同。

警告

atomic 扫描 功能已弃用，OpenSCAP 容器镜像不再使用新的安全合规内容更新。因此，首选 oscap-docker 工具用于安全合规性扫描。

先决条件

- 您已使用 atomic install rhel7/openscap 命令，从 红帽容器目录(RHCC) 下载并安装 OpenSCAP 容器镜像。

流程

1. 列出 OpenSCAP 镜像为 configuration_compliance 扫描提供的 SCAP 内容：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap
```

2. 要将容器镜像修复到指定的策略中，请在扫描配置合规性时将 --remediate 选项添加到 atomic scan 命令中。以下命令构建与 Red Hat Enterprise Linux 7 容器镜像中的 DISA STIG 策略兼容的新修复的容器镜像：

```
~]# atomic scan --remediate --scan_type configuration_compliance --scanner_args
profile=xccdf_org.ssgproject.content_profile_stig-rhel7-disa,report
registry.access.redhat.com/rhel7:latest

registry.access.redhat.com/rhel7:latest (db7a70a0414e589)

The following issues were found:
.....
  Configure Time Service Maxpoll Interval
  Severity: Low
   XCCDF result: fail

  Configure LDAP Client to Use TLS For All Transactions
```

```
  Severity: Moderate
   XCCDF result: fail
.............
Remediating rule 43/44: 'xccdf_org.ssgproject.content_rule_chronyd_or_ntpd_set_maxpoll'
Remediating rule 44/44: 'xccdf_org.ssgproject.content_rule_ldap_client_start_tls'

Successfully built 9bbc7083760e
Successfully built remediated image 9bbc7083760e from
db7a70a0414e589d7c8c162712b329d4fc670fa47ddde721250fb9fcdbed9cc2.

Files associated with this scan are in /var/lib/atomic/openscap/2017-11-06-13-01-42-785000.
```

3. *可选：* ***atomic scan*** *命令的输出报告修复的镜像* ***ID****。要方便记住镜像，请使用一些名称进行标记，例如：*

```
~]# docker tag 9bbc7083760e rhel7_disa_stig
```

8.12. RHEL 7 支持的 SCAP 安全指南配置文件

只使用 RHEL 的特定次要版本中提供的 SCAP 内容。这是因为参与强化的组件会定期使用新功能更新。修改 SCAP 内容来反映这些更新，但并不总是向后兼容的。

在下表中，您可以找到每个 RHEL 次要版本中提供的配置文件，以及配置文件所对应的策略版本。

表 8.2. RHEL 7.9 支持的 SCAP 安全指南配置集

配置文件名称	配置文件 ID	策略版本
第 2 级 CIS Red Hat Enterprise Linux 7 基准 - 服务器	**xccdf_org.ssgproject.content_profile_cis**	RHEL 7.9.9 及更早版本：2.2.0 RHEL 7.9.10 到 RHEL 7.9.29:3.1.1 RHEL 7.9.30 及更新版本：4.0.0
第 1 级 CIS Red Hat Enterprise Linux 7 基准 - 服务器	**xccdf_org.ssgproject.content_profile_cis_server_l1**	RHEL 7.9.10 到 RHEL 7.9.29:3.1.1 RHEL 7.9.30 及更新版本：4.0.0
第 1 级 CIS Red Hat Enterprise Linux 7 基准 - 工作站	**xccdf_org.ssgproject.content_profile_cis_workstation_l1**	RHEL 7.9.10 到 RHEL 7.9.29:3.1.1 RHEL 7.9.30 及更新版本：4.0.0
第 2 级 CIS Red Hat Enterprise Linux 7 基准 - 工作站	**xccdf_org.ssgproject.content_profile_cis_workstation_l2**	RHEL 7.9.10 到 RHEL 7.9.29:3.1.1 RHEL 7.9.30 及更新版本：4.0.0

配置文件名称	配置文件 ID	策略版本
法国信息系统安全局(ANSSI)BP-028 增强级	**xccdf_org.ssgproject.content_profile_**anssi_nt28_enhanced	RHEL 7.9.4 及更早版本：draft RHEL 7.9.5 到 RHEL 7.9.24:1.2 RHEL 7.9.25 及更新版本：2.0
法国信息系统安全部(ANSSI)BP-028 高级别	**xccdf_org.ssgproject.content_profile_**anssi_nt28_high	RHEL 7.9.6 及更早版本：draft RHEL 7.9.7 到 RHEL 7.9.24:1.2 RHEL 7.9.25 及更新版本：2.0
法国信息系统安全局(ANSSI)BP-028 中级	**xccdf_org.ssgproject.content_profile_**anssi_nt28_intermediary	RHEL 7.9.4 及更早版本：草案 RHEL 7.9.5 到 RHEL 7.9.24:1.2 RHEL 7.9.25 及更新版本：2.0
法国信息系统安全局(ANSSI)BP-028 最低级	**xccdf_org.ssgproject.content_profile_**anssi_nt28_minimal	RHEL 7.9.4 及更早版本：draft RHEL 7.9.5 到 RHEL 7.9.24:1.2 RHEL 7.9.25 及更新版本：2.0
C2S for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_**C2S	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_**cjis	5.4
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_**cui	r1
Australian Cyber Security Centre (ACSC) Essential Eight	**xccdf_org.ssgproject.content_profile_**e8	未版本化
健康保险可移植性和责任法案(HIPAA)	**xccdf_org.ssgproject.content_profile_**hipaa	未版本化
NIST 国家检查计划安全指南	**xccdf_org.ssgproject.content_profile_**ncp	未版本化
OSPP - 常规目的操作系统 v4.2.1 的保护配置文件	**xccdf_org.ssgproject.content_profile_**ospp	4.2.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_**pci-dss_centric	RHEL 7.9.12 及更早版本： 3.2.1 在 7.9.13 及更新的版本中删除。如需更多信息，请参阅 RHBZ#2038165

配置文件名称	配置文件 ID	策略版本
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_**pci-dss	RHEL 7.9.0 到 RHEL 7.9.29:3.2.1 RHEL 7.9.30 及更新版本：4.0
[DRAFT] DISA STIG for Red Hat Enterprise Linux Virtualization Host (RHELH)	**xccdf_org.ssgproject.content_profile_**rhelh-stig	草案
VPP - 虚拟化 v 的保护配置文件.1.0 for Red Hat Enterprise Linux Hypervisor (RHELH)	**xccdf_org.ssgproject.content_profile_**rhelh-vpp	1.0
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_**rht-ccp	未版本化
Red Hat Enterprise Linux 7 的标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_**standard	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_**stig	RHEL 7.9.0 和 7.9.1: 1.4 RHEL 7.9.2 到 7.9.4: V3R1 RHEL 7.9.5 和 7.9.6:V3R2 RHEL 7.9.7 到 RHEL 7.9.9:V3R3 RHEL 7.9.10 和 RHEL 7.9.11:V3R5 RHEL 7.9.12 和 RHEL 7.9.13:V3R6 RHEL 7.9.14 到 RHEL 7.9.16:V3R7 RHEL 7.9.17 到 RHEL 7.9.20:V3R8 RHEL 7.9.21 到 RHEL 7.9.24:V3R10 RHEL 7.9.25 到 RHEL 7.9.29:V3R12 RHEL 7.9.30 及更新版本：V3R14

配置文件名称	配置文件 ID	策略版本
Red Hat Enterprise Linux 7 的 DISA STIG with GUI	**xccdf_org.ssgproject.content_profile_stig_gui**	RHEL 7.9.7 到 RHEL 7.9.9:V3R3 RHEL 7.9.10 和 RHEL 7.9.11:V3R5 RHEL 7.9.12 和 RHEL 7.9.13:V3R6 RHEL 7.9.14 到 RHEL 7.9.16:V3R7 RHEL 7.9.17 到 RHEL 7.9.20:V3R8 RHEL 7.9.21 到 RHEL 7.9.24:V3R10 RHEL 7.9.25 到 RHEL 7.9.29:V3R12 RHEL 7.9.30 及更新版本：V3R14

表 8.3. RHEL 7.8 支持 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
DRAFT - ANSSI DAT-NT28 (enhanced)	**xccdf_org.ssgproject.content_profile_anssi_nt28_enhanced**	草案
DRAFT - ANSSI DAT-NT28 (high)	**xccdf_org.ssgproject.content_profile_anssi_nt28_high**	草案
DRAFT - ANSSI DAT-NT28 (intermediary)	**xccdf_org.ssgproject.content_profile_anssi_nt28_intermediary**	草案
DRAFT - ANSSI DAT-NT28 (minimal)	**xccdf_org.ssgproject.content_profile_anssi_nt28_minimal**	草案
C2S for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_C2S**	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_cjis**	5.4
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_cui**	r1
Australian Cyber Security Centre (ACSC) Essential Eight	**xccdf_org.ssgproject.content_profile_e8**	未版本化
健康保险可移植性和责任法案 (HIPAA)	**xccdf_org.ssgproject.content_profile_hipaa**	未版本化

配置文件名称	配置文件 ID	策略版本
NIST 国家检查计划安全指南	**xccdf_org.ssgproject.content_profile_ncp**	未版本化
OSPP - 常规目的操作系统 v4.2.1 的保护配置文件	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_pci-dss_centric**	3.2.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1
[DRAFT] DISA STIG for Red Hat Enterprise Linux Virtualization Host (RHELH)	**xccdf_org.ssgproject.content_profile_rhelh-stig**	草案
VPP - 虚拟化 v 的保护配置文件.1.0 for Red Hat Enterprise Linux Hypervisor (RHELH)	**xccdf_org.ssgproject.content_profile_rhelh-vpp**	1.0
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化
Red Hat Enterprise Linux 7 的标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_standard**	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_stig**	1.4

表 8.4. RHEL 7.7 中支持的 SCAP 安全指南配置集

配置文件名称	配置文件 ID	策略版本
C2S for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_C2S**	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_cjis**	5.4
健康保险可移植性和责任法案 (HIPAA)	**xccdf_org.ssgproject.content_profile_hipaa**	未版本化
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_nist-800-171-cui**	r1

配置文件名称	配置文件 ID	策略版本
OSPP - 常规目的操作系统 v 的保护配置文件。4.2	**xccdf_org.ssgproject.content_profile_**ospp42	4.2
美国政府配置基线	**xccdf_org.ssgproject.content_profile_**ospp	3.9
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_**pci-dss_centric	3.2.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 控制基准	**xccdf_org.ssgproject.content_profile_**pci-dss	3.2.1
VPP - 虚拟化 v 的保护配置文件.1.0 for Red Hat Enterprise Linux Hypervisor (RHELH)	**xccdf_org.ssgproject.content_profile_**rhelh-vpp	1.0
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_**rht-ccp	未版本化
Red Hat Enterprise Linux 7 的标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_**standard	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_**stig-rhel7-disa	1.4

表 8.5. RHEL 7.6 中支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
C2S for Red Hat Enterprise Linux 7	xccdf_org.ssgproject.content_profile_**C2S**	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_**cjis	5.4
健康保险可移植性和责任法案 (HIPAA)	**xccdf_org.ssgproject.content_profile_**hipaa	未版本化
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_**nist-800-171-cui	r1
OSPP - 常规目的操作系统 v 的保护配置文件。4.2	**xccdf_org.ssgproject.content_profile_**ospp42	4.2
美国政府配置基线	**xccdf_org.ssgproject.content_profile_**ospp	3.9

配置文件名称	配置文件 ID	策略版本
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss_centric**	3.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss**	3.1
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化
Red Hat Enterprise Linux 7 的标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_standard**	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_stig-rhel7-disa**	1.4

表 8.6. RHEL 7.5 中支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
C2S for Red Hat Enterprise Linux	**xccdf_org.ssgproject.content_profile_C2S**	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_cjis-rhel7-server**	5.4
General-Purpose 系统的通用配置文件	**xccdf_org.ssgproject.content_profile_common**	未版本化
标准 Docker 主机安全配置文件	**xccdf_org.ssgproject.content_profile_docker-host**	未版本化
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_nist-800-171-cui**	r1
United States Government Configuration Baseline (USGCB / STIG)- DRAFT	**xccdf_org.ssgproject.content_profile_ospp-rhel7**	3.9
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss_centric**	3.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss**	3.1

配置文件名称	配置文件 ID	策略版本
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化
标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_standard**	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_stig-rhel7-disa**	1.4
STIG for Red Hat Virtualization Hypervisor	**xccdf_org.ssgproject.content_profile_stig-rhevh-upstream**	1.4

表 8.7. RHEL 7.4 中支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
C2S for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_C2S**	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_cjis-rhel7-server**	5.4
General-Purpose 系统的通用配置文件	**xccdf_org.ssgproject.content_profile_common**	未版本化
标准 Docker 主机安全配置文件	**xccdf_org.ssgproject.content_profile_docker-host**	未版本化
非联邦信息系统和组织中的非保密信息(NIST 800-171)	**xccdf_org.ssgproject.content_profile_nist-800-171-cui**	r1
United States Government Configuration Baseline (USGCB / STIG)- DRAFT	**xccdf_org.ssgproject.content_profile_ospp-rhel7**	3.9
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss_centric**	3.1
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss**	3.1
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化

配置文件名称	配置文件 ID	策略版本
标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_**standard	未版本化
Red Hat Enterprise Linux 7 的 DISA STIG	**xccdf_org.ssgproject.content_profile_**stig-rhel7-disa	1.4
STIG for Red Hat Virtualization Hypervisor	**xccdf_org.ssgproject.content_profile_**stig-rhevh-upstream	

表 8.8. RHEL 7.3 中支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
C2S for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_**C2S	未版本化
criminal Justice Information Services (CJIS)安全策略	**xccdf_org.ssgproject.content_profile_**cjis-rhel7-server	5.4
General-Purpose 系统的通用配置文件	**xccdf_org.ssgproject.content_profile_**common	未版本化
CNSSI 1253 Low/Low/Low Control Baseline for Red Hat Enterprise Linux 7	**xccdf_org.ssgproject.content_profile_**nist-cl-il-al	未版本化
美国政府配置基线(USGCB / STIG)	**xccdf_org.ssgproject.content_profile_**ospp-rhel7-server	未版本化
适用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_**pci-dss	3.1
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_**rht-ccp	未版本化
标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_**standard	未版本化
STIG for Red Hat Enterprise Linux 7 Server Running GUIs	**xccdf_org.ssgproject.content_profile_**stig-rhel7-server-gui-upstream	1.4
STIG for Red Hat Enterprise Linux 7 Server	**xccdf_org.ssgproject.content_profile_**stig-rhel7-server-upstream	1.4

配置文件名称	配置文件 ID	策略版本
STIG for Red Hat Enterprise Linux 7 Workstation	**xccdf_org.ssgproject.content_profile_stig-rhel7-workstation-upstream**	1.4

表 8.9. RHEL 7.2 支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
General-Purpose 系统的通用配置文件	**xccdf_org.ssgproject.content_profile_common**	未版本化
草案用于 Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	**xccdf_org.ssgproject.content_profile_pci-dss**	草案
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化
标准系统安全配置文件	**xccdf_org.ssgproject.content_profile_standard**	未版本化
Red Hat Enterprise Linux 7 Server 的预发布 Draft STIG	**xccdf_org.ssgproject.content_profile_stig-rhel7-server-upstream**	草案

表 8.10. RHEL 7.1 中支持的 SCAP 安全指南配置文件

配置文件名称	配置文件 ID	策略版本
Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)	**xccdf_org.ssgproject.content_profile_rht-ccp**	未版本化

其它资源

- *有关 RHEL 8 中配置集的详情，请参考 RHEL 8 支持的 SCAP 安全指南配置集*

8.13. 相关信息

- *支持的 SCAP 安全指南版本 - 文章列出了不同 RHEL 版本中支持的 SCAP 安全指南版本。*
-

OpenSCAP 项目页面 - OpenSCAP项目的主页提供了关于 oscap 实用程序和其他与SCAP 相关的组件和项目的详细信息。

- *SCAP Workbench 项目页面 - SCAP Workbench 项目的主页提供了有关 scap-workbench 应用的详细信息。*
- *SCAP 安全指南(SSG)项目页面 - 为 Red Hat Enterprise Linux 提供最新安全内容的 SSG 项目的主页。*
- *红帽安全演示：创建自定义安全策略内容以自动化安全合规 - 一个动手实验室，使用 Red Hat Enterprise Linux 中包含的工具来获取自动化安全合规的初始经验，以符合行业标准安全策略和自定义安全策略。如果您希望为您的团队提供培训或访问这些实验室练习，请联系您的红帽客户团队以了解更多详细信息。*
- *红帽安全演示：使用 RHEL 安全技术保护自己 - 一个动手实验室，了解如何使用 Red Hat Enterprise Linux 中可用的关键安全技术（包括 OpenSCAP）在所有 RHEL 系统级别实施安全性。如果您希望为您的团队提供培训或访问这些实验室练习，请联系您的红帽客户团队以了解更多详细信息。*
- *美国国家标准与技术研究院(NIST) SCAP 页面 - 此页面包含了大量与 SCAP 相关的材料，包括 SCAP 出版物、规范和 SCAP 验证计划。*
- *国家漏洞数据库(NVD) - 此页面代表了 SCAP 内容和其他基于 SCAP 标准漏洞管理数据的最大存储库。*
- *Red Hat OVAL 内容存储库 - 这是一个包含 Red Hat Enterprise Linux 系统漏洞的 OVAL 定义的软件仓库。这是推荐的漏洞内容来源。*
- *MITRE CVE - 这是一个由 MITRE 公司提供的已知安全漏洞的数据库。对于 RHEL，建议使用红帽提供的 OVAL CVE 内容。*
- *MITRE OVAL - 本页代表了 MITRE 公司提供的与 OVAL 相关的项目。除了其他 OVAL 相关信息，这些页面还包含 OVAL 语言的最新版本以及具有数千个 OVAL 定义的 OVAL 内容存储库。请注意，为了扫描 RHEL，建议使用红帽提供的 OVAL CVE 内容。*
- *Red Hat Satellite 文档 - 这组指南还介绍了其他主题，以及如何使用 OpenSCAP 在多个系统上维护系统安全性。*

第 9 章 联邦标准和 REGULATIONS

为了保持安全级别，您的组织可以努力遵守联邦和行业安全规格、标准及法规。本章论述了其中的一些标准和规范。

9.1. 联邦信息处理标准(FIPS)

联邦信息处理标准(FIPS)出版物 140-2 是美国开发的计算机安全标准。政府和行业工作组来验证加密模块的质量。请参阅 NIST 计算机安全资源中心 上的官方 FIPS 出版物。

FIPS 140-2 标准可确保加密工具正确实施其算法。有关这些级别和其他 FIPS 标准规格的详情，请查看 http://dx.doi.org/10.6028/NIST.FIPS.140-2 的完整 FIPS 140-2 标准。

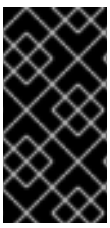
要了解合规要求，请参阅红帽政府标准页面。

9.1.1. 启用 FIPS 模式

要使 Red Hat Enterprise Linux 与联邦信息处理标准(FIPS)出版物 140-2 兼容，您需要进行一些更改以确保使用认证加密模块。您可以在系统安装过程中或之后启用 FIPS 模式。

在系统安装过程中

要跟踪严格的 FIPS 140-2 合规性，请在系统安装过程中将 fips=1 内核选项添加到内核命令行中。使用此选项时，所有密钥的生成都是使用 FIPS 批准的算法和持续监控测试进行的。安装后，系统被配置为自动引导至 FIPS 模式。



重要

通过移动鼠标或按许多击键操作，确保系统在安装过程中有大量熵。推荐的击键次数为 256 等等。少于 256 个击键操作可能会生成非唯一密钥。

系统安装后

要在安装后将系统的内核空间和用户空间变为 FIPS 模式，请按照以下步骤执行：

1. 安装 dracut-fips 软件包：

```
~]# yum install dracut-fips
```

对于带有 AES 新指令(AES-NI)支持的 CPU，还要安装 dracut-fips-aesni 软件包：

```
~]# yum install dracut-fips-aesni
```

2. 重新生成 initramfs 文件：

```
~]# dracut -v -f
```

要启用模块完整性验证，且内核启动过程中存在所有必需的模块，必须重新生成 initramfs 文件。

警告

此操作将覆盖现有的 initramfs 文件。

3. 修改引导装载程序配置。

要引导至 FIPS 模式，请在引导装载程序的内核命令行中添加 fips=1 选项。如果您的 /boot 分区位于独立分区中，请将 boot= <partition> （其中 <partition > 代表 /boot）参数添加到内核命令行中。

要识别引导分区，请输入以下命令：

```
~]$ df /boot
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1               495844     53780    416464  12% /boot
```

为确保 boot= 配置选项即使引导之间的设备命名更改也可以正常工作，请运行以下命令来识别分区的通用唯一识别符(UUID)：

```
~]$ blkid /dev/sda1
/dev/sda1: UUID="05c000f1-f899-467b-a4d9-d5ca4424c797" TYPE="ext4"
```

在内核命令行中附加 UUID：

```
boot=UUID=05c000f1-f899-467b-a4d9-d5ca4424c797
```

根据您的引导装载程序，进行以下更改：

- **GRUB 2**

  将 /boot> 选项的 fips=1 和 boot=<partition 添加到 /etc/default/grub 文件中的 GRUB_CMDLINE_LINUX 键中。要将更改应用到 /etc/default/grub，请重新构建 grub.cfg 文件，如下所示：

  - **在基于 BIOS 的机器上，以 root 用户身份输入以下命令：**

    ```
    ~]# grub2-mkconfig -o /etc/grub2.cfg
    ```

  - **在基于 UEFI 的机器上，以 root 用户身份输入以下命令：**

    ```
    ~]# grub2-mkconfig -o /etc/grub2-efi.cfg
    ```

- **zipl （仅适用于 IBM Z 系统架构）**

  将 /boot> 选项的 fips=1 和 boot=<partition 添加到 /etc/zipl.conf 中，输入以下内容来应用更改：

  ```
  ~]# zipl
  ```

4. 确保禁用预链接。

   要正确操作模块完整性验证，必须禁用对库和二进制文件的预链接。预链接由 prelink 软件包完成，该软件包默认不会安装。除非已安装 prelink，否则不需要这一步。要禁用预链接，请在 /etc/sysconfig/prelink 配置文件中设置 PRELINKING=no 选项。要禁用所有系统文件的现有预链接，请使用 prelink -u -a 命令。

5. 重启您的系统。

在容器中启用 FIPS 模式

如果主机也在 FIPS140-2 模式中设置，则可以将容器切换到 FIPS140-2 模式，并满足以下要求之一：

- dracut-fips 软件包安装在容器中。
- /etc/system-fips 文件从主机挂载到容器上。

9.2. 国家工业安全计划操作手册(NISPOM)

NISPOM （也称为 DoD 5220.22-M）作为国家工业安全计划(NISP)的组件，为所有政府承包信息建立了一系列程序和要求。当前的 NISPOM 日期为 2006 年 2 月 28 日，并纳入了 2013 年 3 月 28 日的主要变化。NISPOM 文档可从以下 URL 下载：

9.3. 支付卡行业数据安全标准(PCI DSS)

https://www.pcisecuritystandards.org/about/index.shtml：PCI 安全标准议会是 2006 年启动的开放性论坛，负责 PCI 安全标准的开发、管理、教育和认知，包括数据安全标准(DSS)。

您可以从 https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml 下载 PCI DSS 标准。

9.4. 安全技术实施指南

安全技术实施指南(STIG)是标准化安全安装和维护计算机软件和硬件的方法。

*有关 **STIG** 的更多信息，请参见以下 **URL**： **https://public.cyber.mil/stigs/**。*

附录 A. 加密标准

A.1. 同步加密

A.1.1. 高级加密标准 - AES

在加密中，高级加密标准(AES)是美国所采用的加密标准。政府.标准包含三个块密码：AES-128、AES-192 和 AES-256，从最初发布为 Rijndael 的大型集合中采用。每个 AES 密码都有一个 128 位块大小，密钥大小为 128, 192 和 256 位。AES 密码已被广泛分析，现在全球使用，就像数据加密标准(DES)的前身一样。[3]

A.1.1.1. AES History

美国国家标准与技术研究所(NIST)发布了 AES。在 5 年的标准化过程后，在 2001 年 11 月 26 日的 FIPS PUB 197 (FIPS 197)。在选择 Rijndael 作为最合适的之前，出示和评估了十五竞争设计。它作为标准 2002 年 5 月 26 日生效。它可用于许多不同的加密软件包。AES 是第一个公开访问，打开由 NSA 批准的顶级机密信息的密码（请参阅 AES 上 Wikipedia 文章中的 Security 部分）。[4]

Rijndael 密码是由两个 Belgian cryptographers、Joan Daemen 和 Vincent Rijmen 开发的，并由它们提交至 AES 选择过程。Rijndael 是两个清单的名称的一个 portmanteau。[5]

A.1.2. 数据加密标准 - DES

数据加密标准(DES)是一个块密码（共享机密加密形式），它被国家标准 Bureau 选为 1976 年美国的官方联邦信息处理标准(FIPS)，然后在国际范围内享受广泛使用。它基于使用 56 位密钥的对称密钥算法。该算法最初与分类设计元素、相对较短的密钥长度和有关国家安全局(NSA)后台的可疑性方面存在。因此，DES 变得非常激烈，它推动了对块密码的现代了解及其加密分析。[6]

A.1.2.1. DES History

对于很多应用程序，DES 现在被视为不安全。这主要是由于 56 位密钥规模太小；在 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年 1999 年，Eectronic Frontier 合作为 22 小时和 15 分钟的公开划分了 DES 密钥。有些分析结果也展示了密码中的理论弱点，尽管在实践中无法挂载它们。该算法以 Triple DES 的形式实际安全，尽管理论存在攻击。近年来，密码已被高级加密标准(AES)取代。[7]

在某些文档中，在 DES 作为标准和 DES 之间区分，该算法被称为 DEA （数据加密算法）。[8]

A.2. 公钥加密

公钥加密是一种加密方法，由许多加密算法和加密系统使用，其区分特征是使用非对称密钥算法，而不是或除了对称密钥算法之外。使用公钥加密的技术，许多保护通信或身份验证消息的方法已变得可行。它们不需要一个或多个密钥的安全初始交换，在使用对称密钥算法时是必需的。它还可用于创建数字签名。[9]

公钥加密是世界各地的基础和广泛使用的技术，它遵循互联网标准作为传输层安全(TLS)（成功到 SSL）、PGP 和 GPG 等方法。[10]

区分在公钥加密中使用的技术是使用非对称密钥算法，其中用于加密消息的密钥与用于解密它的密钥不同。每个用户都有一对加密密钥 - 公钥和私钥。私钥保管了机密，而由于公钥可能被广泛分发。消息使用接收者的公钥加密，且只能使用对应的私钥解密。密钥以数学方式相关，但私钥不可取（例如，在实际或投射实践中）从公钥衍生而来。它是此类算法的发现，其致使在 1970 年中开始的加密实践。[11]

与之相反，Symmetric-key 算法已经用于几万年，使用由发送方和接收器共享的单个 secret 密钥（也必须保持私有，因此会计用于加密和解密的常见术语）。要使用对称加密方案，发件人和接收器必须预先安全地共享密钥。[12]

由于对称密钥算法几乎比较低，因此通常使用密钥交换算法交换密钥，并使用该密钥和对称密钥算法传输数据。例如，PGP 和 SSL/TLS 方案系列执行此操作，例如，称为混合加密系统。[13]

A.2.1. Diffie-Hellman

Diffie-Hellman 密钥交换(D-H)是一种加密协议，允许双方之前没有预先了解通过不安全的通信通道建立共享 secret 密钥。然后，这个密钥可用于使用对称密钥密码加密后续通信。[14]

A.2.1.1. Diffie-Hellman History

该方案最初由 1976 年的 Whitfield Diffie 和 Martin Hellman 发布了，但之后它被 GCHQ 中单独发明了几年，British 信号智能机构由 Malcolm J. criuon 进行交流，但被归类。2002 年，Hellman 建议算法被称为 Diffie-Hellman-Merkle 密钥交换，以认可 Ralph Merkle 对公钥加密(Hellman、2002)的贡献。[15]

虽然 Diffie-Hellman 密钥协议本身是一个匿名（非验证的）密钥协议，但它为各种经过身份验证的协议提供了基础，并在传输层安全的临时模式中提供完美的转发保密（称为 EDH 或 DHE）。[16]

U.S.现在，Chight 4,200,770 已过期，描述算法和信用 Hellman、Diffie 和 Merkle 作为发明商。[17]

A.2.2. RSA

在加密中，RSA （代表 Rivest、Shadr 和 Adleman 是首次公开描述它的算法）是公钥加密的算法。它是已知适合签名和加密的第一个算法，是公钥加密的第一大进步之一。RSA 在电子商业协议中广泛使用，它被认为是安全的，因为密钥足够长且使用最新实施。

A.2.3. DSA

DSA (Digital Signature Algorithm)是数字签名的标准，这是用于数字签名的美国联邦政府标准。DSA 仅用于签名，不是加密算法。[18]

A.2.4. SSL/TLS

传输层安全性(TLS)及其前身安全套接字层(SSL)是加密协议，提供通过互联网等网络进行通信的加密协议。TLS 和 SSL 加密传输层端到端网络连接的片段。

多种版本的协议在 Web 浏览、电子邮件、互联网传真、即时消息和语音无线 IP (VoIP)等应用程序中广泛使用。[19]

A.2.5. Cramer-Shoup Cryptosystem

Cramer-Shoup 系统是一种非对称密钥加密算法，它是验证使用标准加密假设的自适应选择密码文本攻击的第一个有效的方案。其安全性基于决策 Diffie-Hellman 假定的计算不易性（而非被证明）。由 Ronald Cramer 和 Victor Shoup 开发，它是 ElGamal cryptosystem 的扩展。与 ElGamal 不同，这非常易变，Cramer-Shoup 增加了额外的元素，以确保对资源攻击也是如此。这种不可适应性是通过使用冲突的哈希函数和其他计算来实现的，从而导致密码文本与 ElGamal 相同。[20]

A.2.6. ElGamal Encryption

在加密中，ElGamal 加密系统是公钥加密的非对称密钥加密算法，它基于 Diffie-Hellman 密钥协议。它由 1985 年 Taher ElGamal 描述。ElGamal 加密用于免费 GNU Privacy Guard 软件、最新版本的 PGP 和其他加密系统。[21]

---

[3]

"高级加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[4] "高级加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[5] "高级加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[6] "数据加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[7] "数据加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[8] "数据加密标准"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[9] "公钥加密"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[10] "公钥加密"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[11] "公钥加密"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[12] "公钥加密"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[13] "公钥加密"。 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[14] "Diffie-Hellman." 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[15] "Diffie-Hellman." 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[16]
"Diffie-Hellman." 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[17]
"Diffie-Hellman." 维基百科.14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[18]
"DSA." 维基百科.24 February 2010 http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[19]
"TLS/SSL." 维基百科.24 February 2010 http://en.wikipedia.org/wiki/Transport_Layer_Security

[20]
"Cramer-Shoup cryptosystem"。 维基百科.24 February 2010 http://en.wikipedia.org/wiki/Cramer–Shoup_cryptosystem

[21]
"ElGamal encryption" Wikipedia.24 February 2010 http://en.wikipedia.org/wiki/ElGamal_encryption

附录 B. 修订历史记录

修订 1-43 Fri Feb 7 2020 Jan Fiala
使用合规性和漏洞扫描章节的更新同步发行版本。

修订 1-42 Fri Aug 9 2019 Mirek Jahoda
7.7 GA 发布版本。

修订 1-41 Sat Oct 20 2018 Mirek Jahoda
7.6 GA 发布版本。

修订 1-32 Wed Apr 4 2018 Mirek Jahoda
7.5 GA 出版物的版本。

修订 1-30 Thu Jul 27 2017 Mirek Jahoda
7.4 GA 发布版本。

修订 1-24 Mon Feb 6 2017 Mirek Jahoda
带有 misc. updates 的 async 发行版本，特别是在 firewalld 部分。

修订 1-23 Tue Nov 1 2016 Mirek Jahoda
7.3 GA 出版物版本。

修订 1-19 Mon Jul 18 2016 Mirek Jahoda
智能卡部分添加。

修订 1-18 Mon Jun 27 2016 Mirek Jahoda
添加了 OpenSCAP-daemon 和 Atomic Scan 部分。

修订 1-17 Fri Jun 3 2016 Mirek Jahoda
带有 misc. 更新的 async 版本。

修订 1-16 Tue Jan 5 2016 Robert Krátký
后 7.2 GA 修复。

修订 1-15 Tue Nov 10 2015 Robert Krátký
7.2 GA 版本。

修订 1-14.18 Mon Nov 09 2015 Robert Krátký
带有 misc. 更新的 async 版本。

修订 1-14.17 Wed Feb 18 2015 Robert Krátký
7.1 GA 版本。

修订 1-14.15 Fri Dec 06 2014 Robert Krátký
更新以在红帽客户门户网站上排序顺序。

修订 1-14.13 Thu Nov 27 2014 Robert Krátký
反映 POODLE vuln 的更新。

修订 1-14.12 Tue Jun 03 2014 Tomáš Čapek
7.0 GA 版本。