



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

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

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Security_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本书帮助用户和管理员学习保护工作站和服务器的本地和远程入侵、利用和恶意活动的流程和实践。本文档侧重于 Red Hat Enterprise Linux，但其中介绍的概念和技术也适用于所有 Linux 系统。本指南详细介绍了为数据中心、企业及家庭创建安全计算环境的规划及工具。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	13
1.1. 什么是计算机安全性？	13
1.1.1. 标准化安全性	13
1.1.2. 加密软件和认证	13
1.2. 安全控制	13
1.2.1. 物理控制	14
1.2.2. 技术控制	14
1.2.3. 管理控制	14
1.3. 漏洞评估	14
1.3.1. 定义评估和测试	15
1.3.2. 建立漏洞评估方法	16
1.3.3. 漏洞评估工具	16
1.3.3.1. 使用 Nmap 扫描主机	16
1.3.3.1.1. 使用 Nmap	17
1.3.3.2. Nessus	17
1.3.3.3. openVAS	17
1.3.3.4. Nikto	18
1.4. 安全错误	18
1.4.1. 网络安全隐患	18
不安全的架构	18
广播网络	18
集中式服务器	18
1.4.2. 服务器安全隐患	18
未使用的服务和开放端口	18
未修补的服务	19
不小心管理	19
本质上不安全服务	19
1.4.3. 工作站和主页 PC 安全威胁	19
错误密码	20
存在安全漏洞的客户端应用程序	20
1.5. 常见的 EXPLOITS 和 ATTACKS	20
第 2 章 安装的安全提示	23
2.1. 保护 BIOS	23
2.1.1. BIOS 密码	23
2.1.1.1. 保护基于非 BIOS 的系统	23
2.2. 分区磁盘	23
2.3. 安装所需的最小软件包量	24
2.4. 在安装过程中限制网络连接	24
2.5. 安装后流程	25
2.6. 其它资源	25
第 3 章 保持系统正常运行	27
3.1. 维护安装的软件	27
3.1.1. 规划和配置安全更新	27
3.1.1.1. 使用 Yum 的安全功能	27
3.1.2. 更新和安装软件包	28
3.1.2.1. 验证签名的软件包	29
3.1.2.2. 安装签名的软件包	29
3.1.3. 应用由已安装更新引入的更改	29
3.2. 使用红帽客户门户网站	32

3.2.1. 在客户门户网站中查看安全公告	32
3.2.2. CVE 客户门户页面导航	32
3.2.3. 了解问题严重分级	32
3.3. 其它资源	33
安装的文档	33
在线文档	33
红帽客户门户网站	33
另请参阅	33
第 4 章 使用工具和服务强化您的系统	35
4.1. 桌面安全	35
4.1.1. 密码安全性	35
4.1.1.1. 创建 Strong 密码	36
4.1.1.2. 强制使用 Strong 密码	37
4.1.1.3. 配置密码期限	38
4.1.2. 帐户锁定	40
使用 authconfig 保留自定义设置	42
删除 nullok 选项	44
4.1.3. 会话锁定	44
4.1.3.1. 使用 vlock 锁定虚拟控制台	44
4.1.4. 强制只手动挂载可移动介质	45
使用 blockdev 强制只读挂载可移动介质	45
应用新的 udev 设置	45
4.2. 控制根访问	46
4.2.1. 禁止 Root 访问	46
4.2.2. 允许 Root 访问	53
4.2.3. 限制 Root 访问	53
4.2.4. 启用自动注销	53
4.2.5. 保护引导装载程序	54
4.2.5.1. 禁用交互式启动	55
4.2.6. 保护硬链接和符号链接	55
4.3. 保护服务	56
4.3.1. 服务风险	56
4.3.2. 识别和配置服务	57
4.3.3. 不安全的服务	58
4.3.4. 保护 rpcbind	59
4.3.4.1. 使用 TCP wrappers 保护 rpcbind	59
4.3.4.2. 使用 firewalld 保护 rpcbind	60
4.3.5. 保护 rpc.mountd	60
4.3.5.1. 使用 TCP wrappers 保护 rpc.mountd	60
4.3.5.2. 使用 firewalld 保护 rpc.mountd	61
4.3.6. 保护 NIS	61
4.3.6.1. 仔细规划网络	62
4.3.6.2. 使用类似密码 NIS 域名和主机名	62
4.3.6.3. 编辑 /var/yp/securenets 文件	62
4.3.6.4. 分配静态端口和使用丰富语言规则	63
4.3.6.5. 使用 Kerberos 身份验证	63
4.3.7. 保护 NFS	64
4.3.7.1. 仔细规划网络	64
4.3.7.2. 保护 NFS 挂载选项	64
4.3.7.2.1. 查看 NFS 服务器	64
4.3.7.2.2. 查看 NFS 客户端	65
4.3.7.3. 留意语法错误	66

4.3.7.4. 不要使用 no_root_squash 选项	67
4.3.7.5. NFS 防火墙配置	67
为 NFSv3 配置端口	67
4.3.7.6. 使用红帽身份管理保护 NFS	68
4.3.8. 保护 HTTP 服务器	68
4.3.8.1. 保护 Apache HTTP 服务器	68
删除 httpd 模块	70
httpd 和 SELinux	70
4.3.8.2. 保护 NGINX	70
禁用版本字符串	70
包括其他与安全相关的标头	71
禁用 Potentially Harmful HTTP 方法	71
配置 SSL	71
4.3.9. 保护 FTP	71
4.3.9.1. FTP Greeting Banner	72
4.3.9.2. 匿名访问	73
4.3.9.2.1. 匿名上传	73
4.3.9.3. 用户帐户	74
4.3.9.3.1. 限制用户帐户	74
4.3.9.4. 使用 TCP wrapper 控制访问	74
4.3.10. 保护 Postfix	74
4.3.10.1. 限制服务攻击(Denial of Service Attack)	75
4.3.10.2. NFS 和 Postfix	76
4.3.10.3. 仅邮件用户	76
4.3.10.4. 禁用 Postfix 网络监听	76
4.3.10.5. 将 Postfix 配置为使用 SASL	77
设置 Dovecot	77
设置 Postfix	78
其它资源	79
4.3.11. 保护 SSH	79
4.3.11.1. 加密登录	79
4.3.11.2. 多种身份验证方法	80
4.3.11.3. 保护 SSH 的其他方式	80
协议版本	81
密钥类型	81
非默认端口	81
没有根登录	81
使用 X 安全扩展	81
4.3.12. Securing PostgreSQL	82
4.3.13. 保护 Docker	83
4.3.14. 针对 DDoS Attacks 保护 memcached	83
Memcached Vulnerabilities	83
强化 memcached	83
4.4. 保护网络访问	85
4.4.1. 使用 TCP wrapper 和 xinetd 保护服务	85
4.4.1.1. TCP 封装器和连接程序	85
4.4.1.2. TCP wrapper 和 Attack Warning	86
4.4.1.3. TCP wrappers 和增强的日志记录	86
4.4.2. 验证正在侦听哪些端口	87
使用 netstat 进行开放端口扫描	87
使用 ss 进行开放端口扫描	88
使用 netstat 和 ss 为 Open SCTP 端口扫描	89
4.4.3. 禁用源路由	90

4.4.3.1. 反向路径转发	92
启用数据包转发	94
4.4.3.2. 其它资源	94
4.5. 使用 DNSSEC 保护 DNS 流量	95
4.5.1. DNSSEC 简介	95
4.5.2. 了解 DNSSEC	95
了解 Hotspot 问题	95
选择 DNSSEC Capable Recursive Resolver	96
4.5.3. 了解 Dnssec-trigger	96
4.5.4. VPN 提供的域和名称服务器	97
4.5.5. 推荐的命名实践	97
4.5.6. 了解信任 Anchors	97
4.5.7. 安装 DNSSEC	98
4.5.7.1. 安装 unbound	98
4.5.7.2. 检查 unbound 是否正在运行	98
4.5.7.3. 启动 unbound	98
4.5.7.4. 安装 Dnssec-trigger	99
4.5.7.5. 检查 Dnssec-trigger 守护进程是否正在运行	99
4.5.8. 使用 Dnssec-trigger	100
4.5.9. 在 DNSSEC 中使用 dig	100
4.5.10. 为 Dnssec-trigger 设置 Hotspot 检测基础架构	103
4.5.11. 为连接提供域配置 DNSSEC 验证	103
4.5.11.1. 为 Wi-Fi 提供的域配置 DNSSEC 验证	104
4.5.12. 其它资源	104
4.5.12.1. 安装的文档	104
4.5.12.2. 在线文档	105
4.6. 使用 LIBRESWAN 保护虚拟网络(VPN)	106
4.6.1. 安装 Libreswan	106
4.6.2. 使用 Libreswan 创建 VPN 配置	108
4.6.3. 使用 Libreswan 创建主机至主机 VPN	109
4.6.3.1. 使用 Libreswan 验证主机至主机 VPN	111
4.6.4. 使用 Libreswan 配置站点 VPN	112
4.6.4.1. 使用 Libreswan 验证站点到站点的 VPN	113
4.6.5. 使用 Libreswan 配置 Site-to-Site Single Tunnel VPN	113
4.6.6. 使用 Libreswan 配置子网扩展	114
4.6.7. 配置 IKEv2 远程访问 VPN Libreswan	115
4.6.8. 使用 X.509 配置 IKEv1 远程访问 VPN Libreswan 和 XAUTH	117
其它资源	119
4.6.9. 使用对 Quantum Computers 的保护	119
4.6.10. 其它资源	120
4.6.10.1. 安装的文档	120
4.6.10.2. 在线文档	120
4.7. 使用 OPENSLL	121
4.7.1. 创建和管理加密密钥	121
4.7.2. 生成证书	122
4.7.2.1. 创建证书签名请求	122
4.7.2.2. 创建自签名证书	123
4.7.2.3. 使用 Makefile 创建证书	123
4.7.3. 验证证书	124
4.7.4. 加密和解密文件	124
使用 RSA 密钥	125
使用对称算法	125
4.7.5. 生成消息摘要	126

4.7.6. 生成密码哈希	127
4.7.7. 生成随机数据	127
4.7.8. 对您的系统进行基准测试	128
4.7.9. 配置 OpenSSL	128
4.8. 使用 STUNNEL	128
4.8.1. 安装 stunnel	129
4.8.2. 将 stunnel 配置为 TLS wrapper	129
4.8.3. 启动、停止和重启 stunnel	132
4.9. 加密	132
4.9.1. 使用 LUKS 磁盘加密	132
LUKS 概述	132
4.9.1.1. Red Hat Enterprise Linux 中的 LUKS 实施	133
4.9.1.2. 手动加密目录	134
4.9.1.3. 添加新密码到现有设备	136
4.9.1.4. 从现有设备中删除密码	136
4.9.1.5. 在 Anaconda 中创建加密的块设备	137
4.9.1.6. 其它资源	137
4.9.2. 创建 GPG 密钥	137
4.9.2.1. 在 GNOME 中创建 GPG 密钥	138
4.9.2.2. 在 KDE 中创建 GPG 密钥.	138
4.9.2.3. 使用命令行创建 GPG 密钥	139
4.9.2.4. 关于公钥加密	142
4.9.3. 将 openCryptoki 用于公共加密	142
4.9.3.1. 安装 openCryptoki 和 Starting Service	142
4.9.3.2. 配置和使用 openCryptoki	143
4.9.4. 使用智能卡向 OpenSSH 提供凭证	143
4.9.4.1. 从卡检索公钥	143
4.9.4.2. 在服务器上存储公钥	144
4.9.4.3. 在智能卡中使用密钥向服务器进行身份验证	144
4.9.4.4. 使用 ssh-agent 自动登录 PIN Logging	145
4.9.4.5. 其它资源	145
4.9.5. 可信和加密的密钥	146
4.9.5.1. 使用密钥	146
4.9.5.2. 其它资源	148
安装的文档	149
在线文档	149
另请参阅	149
4.9.6. 使用随机数字生成器	149
4.10. 使用基于策略的解密配置自动解锁加密卷	153
4.10.1. Network-Bound Disk Encryption	153
4.10.2. 安装加密客户端 - Clevis	154
4.10.3. 在强制模式中使用 SELinux 部署 Tang 服务器	156
先决条件	156
流程	156
4.10.3.1. 部署高可用性系统	158
4.10.4. 使用 Tang 为 NBDE 系统部署加密客户端	159
先决条件	159
流程	159
4.10.5. 使用 TPM 2.0 策略部署加密客户端	160
4.10.6. 配置手动注册卷	161
4.10.7. 使用 Kickstart 配置自动注册	163
4.10.8. 配置可移动存储设备的自动解锁	164
4.10.9. 在引导时配置非 root 卷的自动解锁	165

4.10.10. 在 NBDE 网络中部署虚拟机	165
4.10.11. 使用 NBDE 为云环境构建可自动滚动的虚拟机镜像	166
4.10.12. 其它资源	166
4.11. 使用 AIDE 检查完整性	167
4.11.1. 安装 AIDE	167
4.11.2. 执行完整性检查	168
4.11.3. 更新 AIDE 数据库	169
4.11.4. 其它资源	169
4.12. 使用 USBGUARD	169
4.12.1. 安装 USBGuard	170
4.12.2. 创建白名单和黑名单	171
4.12.3. 使用规则语言创建您的策略	174
4.12.4. 其它资源	176
4.13. 强化 TLS 配置	176
4.13.1. 选择 Algorithms 来启用	176
协议版本	177
密码套件	178
公钥长度	179
4.13.2. 使用 TLS 的实现	179
4.13.2.1. 在 OpenSSL 中使用 Cipher Suite	179
4.13.2.2. 在 GnuTLS 中使用 Cipher Suite	181
4.13.3. 配置特定应用程序	182
4.13.3.1. 配置 Apache HTTP 服务器	182
4.13.3.2. 配置 Dovecot 邮件服务器	183
4.13.4. 附加信息	184
安装的文档	184
在线文档	185
另请参阅	185
4.14. 使用共享系统证书	185
4.14.1. 使用系统范围的 Trust Store	185
4.14.2. 添加新证书	186
4.14.3. 管理可信系统证书	186
4.14.4. 其它资源	188
4.15. 使用 MACSEC	188
4.16. 使用清理安全地删除数据	188
第 5 章 使用防火墙	192
5.1. FIREWALLD 入门	192
5.1.1. Zones	192
5.1.2. 预定义的服务	194
5.1.3. 运行时和永久设置	194
5.1.4. 使用 CLI 修改运行时和永久配置中的设置	195
5.2. 安装 FIREWALL-CONFIG GUI 配置工具	196
5.3. 查看 FIREWALLD 的当前状态和设置	196
5.3.1. 查看 firewalld 的当前状态	196
5.3.2. 查看当前 firewalld 设置	197
5.3.2.1. 使用 GUI 查看允许的服务	197
5.3.2.2. 使用 CLI 查看 firewalld 设置	198
5.4. 启动 FIREWALLD	200
5.5. 停止 FIREWALLD	200
5.6. 控制流量	200
5.6.1. 预定义的服务	200
5.6.2. 使用 CLI 在紧急情况时禁用所有流量	201

5.6.3. 使用 CLI 使用预定义服务控制流量	201
5.6.4. 使用 GUI 使用预定义服务控制流量	202
5.6.5. 添加新服务	202
5.6.6. 使用 CLI 控制端口	203
打开端口	203
关闭端口	204
5.6.7. 使用 GUI 打开端口	205
5.6.8. 使用 GUI 控制协议的流量	205
5.6.9. 使用 GUI 打开源端口	205
5.7. 使用区域	205
5.7.1. 列出区域	205
5.7.2. 修改 Certain 区的 firewalld 设置	206
5.7.3. 更改默认区	206
5.7.4. 将网络接口分配给区	207
5.7.5. 为网络连接分配默认区	207
5.7.6. 创建新区域	207
5.7.7. 使用配置文件创建新区域	208
5.7.8. 使用区域目标设置传入流量的默认行为	209
5.8. 使用区管理传入的流量依赖源	209
5.8.1. 添加源	209
5.8.2. 删除源	210
5.8.3. 添加源端口	211
5.8.4. 删除源端口	211
5.8.5. 使用区和源允许服务仅用于特定域	211
5.8.6. 配置受基于协议的区域接受的流量	212
在区中添加协议	212
从区中删除协议	213
5.9. 端口转发	213
5.9.1. 添加端口到重定向	213
5.9.2. 删除重定向的端口	214
5.10. 配置 IP 地址伪装	215
5.11. 管理 ICMP 请求	216
5.11.1. 列出 ICMP 请求	217
5.11.2. 阻塞或取消阻塞 ICMP 请求	217
5.11.3. 在不提供任何信息的情况下阻止 ICMP 请求	217
5.11.4. 使用 GUI 配置 ICMP Filter	219
5.12. 使用 FIREWALLD 设置和控制 IP 集	219
5.12.1. 使用命令行客户端配置 IP 设置选项	220
5.12.2. 为 IP 集合配置自定义服务	222
5.13. 使用 IPTABLES 设置和控制 IP 集	223
5.14. 使用直接接口	224
5.14.1. 使用直接接口添加规则	225
5.14.2. 使用直接接口删除规则	225
5.14.3. 使用直接接口列出规则	225
5.15. 使用 "RICH LANGUAGE" 语法配置复杂防火墙规则	226
5.15.1. Rich Language 命令格式化	226
5.15.2. 了解 Rich 规则结构	226
5.15.3. 了解 Rich Rule 命令选项	227
源和目标地址	227
元素	228
日志	229
操作	230
5.15.4. 使用 Rich Rule Log 命令	230

5.15.4.1. 使用 Rich 规则日志命令示例 1	230
5.15.4.2. 使用 Rich Rule 日志命令示例 2	230
5.15.4.3. 使用 Rich Rule 日志命令示例 3	231
5.15.4.4. 使用 Rich Rule 日志命令示例 4	231
5.15.4.5. 使用 Rich Rule 日志命令示例 5	231
5.15.4.6. 使用 Rich Rule 日志命令示例 6	231
5.16. 配置防火墙锁定	232
5.16.1. 使用命令行客户端配置锁定	232
5.16.2. 使用命令行客户端配置锁定白名单选项	232
5.16.3. 使用配置文件配置锁定白名单选项	235
5.17. 为拒绝数据包配置日志记录	236
5.18. 其它资源	237
5.18.1. 安装的文档	237
5.18.2. 在线文档	238
第 6 章 NFTABLES 入门	239
使用 FIREWALLD 或 NFTABLES 时	239
6.1. 编写和执行 NFTABLES 脚本	240
6.1.1. 支持的 nftables 脚本格式	240
6.1.2. 运行 nftables 脚本	241
先决条件	241
其它资源	242
6.1.3. 使用 nftables 脚本中的注释	243
6.1.4. 使用 nftables 脚本中的变量	243
只有一个值的变量	243
包含匿名集合的变量	243
其它资源	244
6.1.5. 在 nftables 脚本中包含文件	244
其它资源	245
6.1.6. 系统引导时自动载入 nftables 规则	245
先决条件	245
其它资源	245
6.2. 创建和管理 NFTABLES 表、链和规则	246
6.2.1. 显示 nftables 规则集	246
6.2.2. 创建 nftables 表	246
其它资源	247
6.2.3. 创建 nftables 链	247
先决条件	248
其它资源	248
6.2.4. 在 nftables 链末尾附加规则	249
先决条件	249
其它资源	249
6.2.5. 在 nftables 链的开头插入规则	249
先决条件	250
其它资源	250
6.2.6. 在 nftables 链的特定位置插入规则	250
先决条件	250
其它资源	251
6.3. 使用 NFTABLES 配置 NAT	252
6.3.1. 不同的 NAT 类型：masquerading、source NAT、destination NAT 和 redirect	252
伪装和源 NAT（SNAT）	252
目标 NAT（DNAT）	253
重定向	253

6.3.2. 使用 nftables 配置伪装	253
6.3.3. 使用 nftables 配置源 NAT	254
其它资源	254
6.3.4. 使用 nftables 配置目标 NAT	255
其它资源	256
6.3.5. 使用 nftables 配置重定向	256
其它资源	257
6.4. 使用 NFTABLES 命令中的设置	257
6.4.1. 在 nftables 中使用匿名集合	257
先决条件	257
6.4.2. 在 nftables 中使用命名集	258
先决条件	258
6.4.3. 相关信息	259
6.5. 在 NFTABLES 命令中使用 VERDICT 映射	259
6.5.1. 在 nftables 中使用匿名映射	259
6.5.2. 在 nftables 中使用命名映射	261
6.5.3. 相关信息	264
6.6. 使用 NFTABLES 配置端口转发	264
6.6.1. 将传入的数据包转发到不同的本地端口	264
6.6.2. 将特定本地端口上传入的数据包转发到不同主机	265
先决条件	265
6.7. 使用 NFTABLES 来限制连接数量	266
6.7.1. 使用 nftables 限制连接数量	266
先决条件	266
6.7.2. 在一分钟内尝试超过十个进入的 TCP 连接的 IP 地址	267
6.7.3. 其它资源	267
6.8. 调试 NFTABLES 规则	268
6.8.1. 创建带有计数器的规则	268
先决条件	268
6.8.2. 在现有规则中添加计数器	268
先决条件	269
6.8.3. 监控与现有规则匹配的数据包	269
先决条件	269
第 7 章 系统审核	271
使用案例	272
7.1. AUDIT 系统架构	273
7.2. 安装 AUDIT 软件包	274
7.3. 配置审计服务	274
7.3.1. 为安全环境配置 auditd	274
7.4. 启动审计服务	276
7.5. 定义审计规则	277
7.5.1. 使用 auditctl 定义审计规则	278
定义控制规则	278
定义文件系统规则	279
定义系统调用规则	280
7.5.2. 定义可执行文件规则	282
7.5.3. 在 /etc/audit/audit.rules 文件中定义持久性审计规则和控制	282
定义控制规则	283
定义文件系统和系统调用规则	283
预配置规则文件	283
使用 augenrules 定义持久性规则	284
7.6. 了解 AUDIT 日志文件	285

第一次记录	286
第二记录	289
第三个记录	290
第四个记录	292
7.7. 搜索 AUDIT 日志文件	292
7.8. 创建审计报告	293
7.9. 其它资源	294
在线源	294
安装的文档	295
手动页面	295
第 8 章 扫描系统以了解配置合规性和漏洞	297
8.1. RHEL 中的配置合规工具	297
8.2. 漏洞扫描	298
8.2.1. 红帽安全公告 OVAL Feed	298
8.2.2. 扫描系统是否有漏洞	299
8.2.3. 扫描远程系统是否有漏洞	301
8.3. 配置合规性扫描	301
8.3.1. RHEL 7 中的配置合规性	302
合规性扫描资源的结构	302
8.3.2. OpenSCAP 扫描的可能结果	303
8.3.3. 查看配置合规性配置集	303
8.3.4. 使用特定基行评估配置合规性	305
8.4. 使用特定基线将系统修复为强制	305
8.5. 使用 SSG ANSIBLE PLAYBOOK, 使用特定基线将系统修复为强制	306
8.6. 创建修复 ANSIBLE PLAYBOOK 以使用特定基本行的系统	308
8.7. 使用 SCAP WORKBENCH 使用自定义配置集扫描系统	309
8.7.1. 使用 SCAP Workbench 扫描和修复系统	309
8.7.2. 使用 SCAP Workbench 自定义安全配置集	311
8.7.3. 相关信息	313
8.8. 在安装后, 使用安全配置集部署与安全配置集兼容的系统	313
8.8.1. 使用图形安装部署 Baseline-Compliant RHEL 系统	314
8.8.2. 使用 Kickstart 部署 Baseline-Compliant RHEL 系统	315
8.9. 针对漏洞扫描容器和容器镜像	316
8.9.1. 使用oscap-docker扫描容器镜像和容器的漏洞	317
8.9.2. 使用原子扫描扫描容器镜像和容器以了解漏洞	318
8.10. 使用特定基础镜像评估容器或容器镜像的配置合规性	319
8.11. 使用原子扫描扫描并修复容器镜像和容器的配置合规性	321
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	321
8.11.2. 使用原子扫描修复容器镜像和容器的配置合规性	322
8.12. RHEL 7 中支持的 SCAP 安全指南配置集	323
8.13. 相关信息	332
第 9 章 联邦标准和强制	334
9.1. 联邦信息处理标准(FIPS)	334
9.1.1. 启用 FIPS 模式	334
系统安装过程中	334
系统安装后	334
在容器中启用 FIPS 模式	337
9.2. 国家工业安全计划操作手册(NISPOM)	337
9.3. 支付卡行业数据安全标准(PCI DSS)	337
9.4. 安全技术实施指南	337
附录 A. 加密标准	339

A.1. 同步加密	339
A.1.1. 高级加密标准 - AES	339
A.1.1.1. AES 历史记录	339
A.1.2. 数据加密标准 - DES	339
A.1.2.1. DES History	339
A.2. 公钥加密	339
A.2.1. Diffie-Hellman	340
A.2.1.1. Diffie-Hellman History	340
A.2.2. RSA	341
A.2.3. DSA	341
A.2.4. SSL/TLS	341
A.2.5. cramer-Shoup Cryptosystem	341
A.2.6. Elgamal Encryption	341
附录 B. 修订历史记录	344

第 1 章 安全主题概述

由于日益依赖强大的网络计算机来帮助经营业务并跟踪个人信息，整个行业围绕网络和计算机安全性建立起来。企业已请求安全专家的知识和技能正确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）都认为安全性是自带的，被人忽略的是为了提高功能、生产率、便利性、易用性和预算问题。通常会在发生了未经授权的入侵后，进行适当的安全实施。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` 目录中的文件进行多个引用。使用 64 位系统时，上面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。些术语和指标已进入我们的日常业务词汇，如总拥有成本(TCO)、投资回报(ROI)和服务质量(QoS)。借助这些指标，行业可以将数据完整性和高可用性(HA)等因素作为规划和流程管理成本的一部分进行计算。在电子商业等行业中，数据的可用性和可信性意味着成功和失败之间的区别。

1.1.1. 标准化安全性

每个行业的企业依赖制定标准机构（如美国医疗协会(AMA)或电气与电子工程师协会(IEEE)）设定的法规和规则。对于信息安全性，同样也是一样的理想选择。许多安全顾问和供应商都同意了称为 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传输和使用。例如，信息的机密性确保了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（如身份盗窃或信用欺诈）获得。
- 完整性 - 不应以导致信息不完整或不正确的方式更改信息。未授权用户应受限与修改或销毁敏感信息的能力。
- 可用性 - 授权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，并同意在网络服务提供商及其企业客户的服务级别协议(SLA)中正式制定。

1.1.2. 加密软件和认证

以下红帽知识库文章提供了有关 Red Hat Enterprise Linux 核心加密组件的概述，记录这些加密组件是什么，它们的选择方式，如何集成到操作系统中，它们如何支持硬件安全模块和智能卡，以及如何对他们应用加密认证。

- [RHEL7 Core Crypto 组件](#)

1.2. 安全控制

计算机安全性通常分为三种不同的 master 类别，通常称为控制：

- 物理
- 技术
- 管理

这三大类别定义了适当安全实施的主要目标。这些控件内是子类别，进一步详细说明控件及其实施方法。

1.2.1. 物理控制

物理控制是在定义的结构中实施安全措施，用于阻止或阻止对敏感材料进行未经授权的访问。物理控制示例如下：

- 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控制物理结构和网络上敏感数据访问和使用的基础。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。它们涉及机构内所有级别的人员，并确定哪些用户有权访问哪些资源和信息，例如：

- 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

根据时间、资源和动机，攻击者几乎可以进入任何系统。当前提供的所有安全程序和技术都无法保证所有系统完全不受入侵。路由器有助于保护到互联网的网关的安全。防火墙有助于保护网络边缘。虚拟专用网络在加密流中安全地传递数据。入侵检测系统提醒您进行恶意活动。但是，这些技术能否成功取决于多个变量，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。
- 负责人员在网络上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。尽管在信息安全的许多领域具有丰富的人员知识仍然有可能，但很难保留在多个主题领域的专家。这主要是因为信息安全的每个主题领域都需要持续关注。信息安全并不存在。

漏洞评估是对您的网络和系统安全性的内部审计；其结果表明网络的机密性、完整性和可用性（如第 1.1.1 节“标准化安全性”中所述）。通常，漏洞评估从分析阶段开始，在该阶段收集有关目标系统和资源的重要数据。此阶段会导致系统就绪阶段，将基本检查所有已知的漏洞目标。报告阶段结束，调查结果分为高、中度和低风险类别；并讨论提高目标安全（或降低漏洞风险）的方法。

如果您要对家进行漏洞评估，您可能会检查家中的每个门，看看它们是否被关闭和锁定。您还要检查每个窗口，确保它们完全关闭且正确。同样的概念也适用于系统、网络和电子数据。恶意用户是您的数据的无处不在和篡改。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

漏洞评估可分为两种类型之一：*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。成为外部您的公司，为您提供攻击者的观点。您会看到攻击者可以看到的內容 - 可公开路由的 IP 地址、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 Internet 流量可访问的设备，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，而且您的状态已提升为可信。这是您和同事登录系统之后有了自己的观点。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。确保组织内部几乎很少的操作（如部门防火墙、用户级访问控制和内部资源的身份验证程序）。通常，内部查看时会有更多资源，因为大多数系统都是公司内部系统。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受限。

漏洞评估和渗透测试之间有区别。可将漏洞评估作为入渗透测试的第一步。从评估中获得的信息用于测试。评估会检查漏洞和潜在漏洞，而渗透测试实际上会尝试利用发现。

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。

以下列表检查了执行漏洞的一些益处。

- 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. 建立漏洞评估方法

为了协助选择用于漏洞评估的工具，建立漏洞评估方法很有帮助。遗憾的是，目前还没有预定义或行业批准的方法；但是，常识和最佳实践可以作为足够的指南。

*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要选择的工具，而且有助于确定它们的使用方式。

要了解更多有关建立方法的信息，请参阅以下网站：

- <https://www.owasp.org/> - 开放 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，请先映射布局，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。

正如在日常生活的任何方面一样，也有许多不同的工具来执行相同的工作。这个概念也适用于执行漏洞评估。有一些特定于操作系统、应用程序甚至网络的工具（基于所用的协议）。些工具是免费的，另一些则没有。些工具直观且易于使用，而其他工具则加密且记录不准确，但具有其他工具所不具备的功能。

找到正确的工具可能是一项艰巨的任务，最后还要计算经验计数。如果可能，设置测试实验室并尝试尽可能多的工具，注意每个测试的优缺点和缺点。查看工具的README 文件或 man page。此外，请参阅 Internet 获取更多信息，如文章、逐步指南，甚至邮寄特定于工具的列表。

下面讨论的工具仅仅是可用工具的一小部分。

1.3.3.1. 使用 Nmap 扫描主机

nmap 是一个流行工具，可用于确定网络布局。**nmap** 已存在多年，可能是收集信息时最常用的工具。其中包含了一个优秀的 man page，其中提供了其选项和用法的详细描述。管理员可以使用网络上的 **Nmap** 来查找主机系统和这些系统上的开放端口。

nmap 是漏洞评估的一个坚实的第一步。您可以映射您的网络中的所有主机，甚至传递一个选项，让 **Nmap** 能够尝试识别特定主机上运行的操作系统。**nmap** 是建立使用安全服务和限制未使用的服务的政策的良好基础。

要安装 **Nmap**，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 提示符下运行，只需键入 **nmap** 命令，后跟要扫描的主机的主机名或 **IP 地址**：

```
nmap <hostname>
```

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

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

基本扫描的结果（可能最多需要几分钟，具体取决于主机所处的位置和其他网络条件），如下所示：

```
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的网络通信端口。该知识对希望关闭不必要的或未使用的服务的管理员很有帮助。

有关使用 **Nmap** 的更多信息，请参阅以下 URL 中的官方主页：

<http://www.insecure.org/>

1.3.3.2. Nessus

Nessus 是一款全面服务的安全扫描器。**Nessus** 的插件架构允许用户为其系统和网络进行自定义。与任何扫描器一样，**Nessus** 的效果只能是它所依赖的签名数据库。幸运的是，**Nessus** 经常被更新，具有完整报告、主机扫描和实时漏洞搜索的功能。请记住，即使工具功能强大且经常更新为 **Nessus**，也可能出现假的正状态和假的负数。



注意

Nessus 客户端和服务端软件需要使用订阅。本文档中已包含它，以作为可能有兴趣使用此流行应用的用户的参考。

有关 **Nessus** 的更多信息，请参阅官方网站，网址为以下 URL：

<http://www.nessus.org/>

1.3.3.3. openVAS

OpenVAS（开放式漏洞评估系统） 是一组可用于扫描漏洞和全面的漏洞管理的工具和服务。**OpenVAS** 框架提供多个基于 Web 的桌面和命令行工具来控制解决方案的各个组件。**OpenVAS** 的核心功能由安全扫描器提供，利用了超过 3,300 万次每天更新的网络漏洞测试(NVT)。与 **Nessus**（请参阅第 1.3.3.2 节

“Nessus”) 不同, OpenVAS 不需要任何订阅。

有关 OpenVAS 的更多信息, 请参阅官方网站, 网址为以下 URL :

<http://www.openvas.org/>

1.3.3.4. Nikto

Nikto 是卓越的通用网关接口(CGI)脚本扫描程序。Nikto 不仅会检查 CGI 漏洞, 而且会以侵入方式进行检查, 从而避免入侵检测系统。它附带了完整的文档, 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 将是检查这些服务器的安全性的绝佳资源。

有关 Nikto 的更多信息, 请访问以下 URL :

<http://cirt.net/nikto2>

1.4. 安全错误

1.4.1. 网络安全隐患

配置网络的以下方面时的错误做法可能会增加攻击的风险。

不安全的架构

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

系统管理员往往无法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广播, 直到接收节点接收和处理数据。此方法最易受地址解析协议(ARP)或介质访问控制(MAC)地址欺骗, 它受到本地主机上的入侵者和未经授权的用户欺骗。

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, 因为管理和成本比多服务器配置要低得多。但是, 集中式服务器在网络上引入单点故障。如果中央服务器泄露, 可能会导致网络完全不可用或更加糟糕, 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大量组织的重要信息。如果服务器被入侵, 则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 1000 多个应用程序和库软件包。但是, 大多数服务器管理员不选择在发行版中安装每一个软件包, 而更喜欢安装软件包的基本安装, 包括多个服务器应用。有关限制安装的软件包数量以及其它资源的原因, 请查看 [第 2.3 节“安装所需的最小软件包量”](#)。

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, 使用默认设置进行配置, 并且可能开启。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, 而管理员未意识到它, 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, 请查看 [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、经过全面测试的软件。多年来一直在生产环境中使用，其代码得到了全面优化，发现并修复了许多漏洞。

然而，没有像完美软件这样的事情，也总有进一步解决的空间。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在服务器应用程序中找到可利用的错误，并在错误跟踪和安全相关网站上发布相关信息，如 Bugtraq 邮件列表(<http://www.securityfocus.com>)或计算机应急响应团队(CERT)网站(<http://www.cert.org>)。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。这一点尤为正确，因为攻击者可以访问这些相同的漏洞跟踪服务，并将在可能时利用信息破解未修补的系统。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详情，请参考 [第3章 保持系统正常运行](#)。

不小心管理

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) 的调查，计算机安全漏洞的主要原因是“指派未经培训的人员维护安全性、提供培训或提供培训或时间来学习和完成该工作”。^[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 workstation 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，一些数据库具有默认的管理密码，因为数据库开发人员假定系统管理员在安装后立即更改这些密码。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。

本质上不安全服务

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用户名和密码进行身份验证的服务。Telnet 和 FTP 是两个这样的服务。如果数据包嗅探软件正在监控远程用户和此类服务用户名和密码之间的流量，则很容易截获。

从本质上讲，这些服务也更容易成为安全行业所说的中间人攻击的先驱。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。旦有人打开到服务器的远程会话，攻击者的计算机将充当不可见的渠道，在远程服务和不指定用户捕获信息之间保持静默。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。

另一类不安全的网络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 还具有网络中的每个计算机都必须在纯文本 ASCII 或 DBM（ASCII 派生）数据库中识别的重要信息，包括密码和文件权限。获得此数据库访问权限的攻击者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，发布 Red Hat Enterprise Linux 7 时关闭所有此类服务。但是，由于管理员通常发现自己强制使用这些服务，因此仔细的配置非常重要。有关使用安全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. 工作站和主页 PC 安全威胁

工作站和家庭 PC 可能不会受到网络或服务器的攻击，而是因为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。工作站也可以在没有用户知识的情况下使用，攻击者将工作站用作协调攻击中的“从属”机器。因此，了解工作站的漏洞可让用户避免重新安装操作系统的麻烦，或者更糟糕地从数据

失窃中恢复。

错误密码

错误密码是攻击者获得系统访问权限最简单的方法之一。有关如何在创建密码时避免出现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修补的服务器，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，攻击者可以在通过网络时捕获纯文本用户名和密码，然后使用帐户信息来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 客户端容易受到来自恶意 SSH 服务器的 X 转发攻击。连接到服务器后，攻击者可以静默捕获客户端通过网络执行的任何击键操作和鼠标单击操作。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”](#) 更详细地讨论管理员和家庭用户应该执行哪些步骤来限制计算机工作站的漏洞。

1.5. 常见的 EXPLOITS 和 ATTACKS

[表 1.1 “常见 Exploits”](#) 详细说明入侵者用于访问组织网络资源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如何执行它们以及管理员如何正确地保护其网络免受此类攻击。

表 1.1. 常见 Exploits

漏洞	描述	备注
null 或默认密码	将管理密码留空，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（尽管红帽企业 Linux 7 不附带它们）。	<p>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。</p> <p>在很多传统操作系统中常见，尤其是捆绑服务（如 UNIX 和 Windows）的操作系统。</p> <p>管理员有时会在崩溃中创建特权用户帐户，并将密码保留为空，从而为发现该帐户的恶意用户创建完美入口点。</p>
默认共享密钥	有时，安全服务会打包用于开发或评估测试目的默认安全密钥。如果这些密钥保持不变，并放置在互联网上的生产环境中，则具有相同默认密钥的所有用户都可以访问该共享密钥资源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，找到您的服务器的漏洞，并安装一个后门程序或 Trojan horse 来控制您的网络资源。	<p>欺骗比较困难，因为它涉及到攻击者预测 TCP/IP 序列号以协调与目标系统连接的攻击者，但有多种工具都可用于协助攻击者执行此类漏洞。</p> <p>具体取决于使用基于源的身份验证技术的目标系统运行服务（如 rsh、telnet、FTP 等），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。</p>

漏洞	描述	备注
Seavesdropping	通过窃听两个节点之间的连接，在网络上的两个活跃节点之间传递数据。	<p>这种类型的攻击主要适用于 Telnet、FTP 和 HTTP 传输等纯文本传输协议。</p> <p>远程攻击者必须有权访问 LAN 上的已入侵系统才能执行此类攻击；通常攻击者已使用主动攻击（如 IP 欺骗或中间人）破坏了 LAN 上的系统。</p> <p>安全措施包括带有加密密钥交换的服务、一次性密码或加密身份验证以防止密码嗅探；还建议在传输期间进行强大的加密。</p>
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可能保存的任何数据，并可能破坏网络中的其他系统。	<p>基于 HTTP 的服务（如 CGI）容易受到远程命令执行甚至交互式 shell 访问的影响。即使 HTTP 服务作为非特权用户（如 "nobody"）运行，可以读取配置文件和网络映射等信息，或者攻击者也可以启动拒绝服务攻击，从而排空系统资源或使其无法被其他用户访问。</p> <p><i>服务有时可能会有在开发和测试过程中没有被注意的漏洞；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，从而给予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。</i></p> <p>管理员应确保服务不以 root 用户身份运行，并且应保持对来自供应商或安全组织（如 CERT 和 CVE）的应用程序的补丁和勘误表更新。</p>
应用程序漏洞	攻击者在桌面和 workstation 应用程序（如电子邮件客户端）中发现故障，执行任意代码，为将来的入侵或崩溃系统而模仿 Trojan 马车。如果被入侵的工作站在网络其余部分上具有管理特权，则可能会进一步利用。	<p>workstation 和桌面更易被利用，因为工作者不具备防止或检测到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开放非请求电子邮件附件时所承担的风险。</p> <p>可以实施保护，使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 workstation 软件；或其他系统管理服务可以减轻多套安全部署的负担。</p>
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 workstation）发送未经授权的数据包，针对组织的网络或服务器资源进行协调。这会强制资源对合法用户不可用。	<p>美国报告的最新 DoS 问题单在 2000 年发生。使用具有高带宽连接的几个具有高带宽连接的系统作为僵停或重定向广播节点，一个协调的 ping 攻击使一些高流量商业和政府站点不可用。</p> <p>源数据包通常会伪造（以及重播），从而使调查攻击的真正来源变得困难。</p> <p>使用 iptables 和 Network Intrusion Detection 系统（如 snort）帮助管理员跟踪并防止分布式 DoS 攻击，从而在入口过滤(IETF rfc2267)中进行入口过滤(IETF rfc2267)。</p>

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

第 2 章 安装的安全提示

安全性从首次将该 CD 或 DVD 放入磁盘驱动器以安装 Red Hat Enterprise Linux 7 时开始。从开始便安全地配置您的系统，以后更轻松的实施额外的安全设置。

2.1. 保护 BIOS

对 BIOS（或 BIOS 等效）和启动加载程序的密码保护可防止对系统具有物理访问权限的未授权用户使用可移动介质启动，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和计算机位置的敏感程度。

例如，如果一台计算机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，如果员工的笔记本电脑中对公司网络的未加密 SSH 密钥仍保持不变，则可能导致整个公司出现重大安全漏洞。

但是，如果工作站位于只有授权人或可信人员有权访问的地方，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. 防止更改 BIOS 设置 - 如果入侵者有权访问 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，从而让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 - 某些 BIOS 允许对引导过程进行密码保护。激活后，攻击者必须在 BIOS 启动加载器启动前输入密码。

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了解具体说明。

如果您忘记 BIOS 密码，可以通过主板上的跳转器重置，也可以通过断开 CMOS 电池来重置。因此，最好尽可能锁定计算机案例。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。

有关保护类似 BIOS 程序的密码的说明，请查看制造商的说明。

2.2. 分区磁盘

红帽建议为 /boot、/home、/tmp 和 /var/tmp / 目录创建单独的分区。每种分区的原因各不相同，我们将探讨每个分区。

/boot

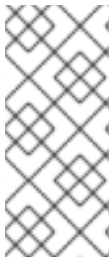
这个分区是系统在启动过程中读取的第一个分区。用于将系统引导至 Red Hat Enterprise Linux 7 的引导装载程序和内核映像存储在这个分区中。此分区不应加密。如果此分区包含在 / 中，并且该分区已加密或者不可用，那么您的系统将无法引导。

/home

当用户数据(/home)存储在 / 中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 的下一个版本时，当您可以在 /home 分区中保存数据时会更加容易，因为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，可以稍微多一点地保护数据丢失。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，则它可以消耗您的所有存储空间。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，将这些目录移动到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. 安装所需的最小软件包量

最好仅安装您将使用的软件包，因为计算机上的每一款软件可能包含漏洞。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。

有关安装最小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。Kickstart 文件也可以使用 `--nobase` 选项执行最小安装。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/en-US/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，它可能不是最新的安全修复程序，而且可能会受到仅在安装介质提供的系统发布后解决的某些问题的影响。

安装有潜在漏洞的操作系统时，始终只限制对最接近的必要网络区域的影响。最安全的选择是“无网络”区域，这意味着在安装过程中保持计算机断开连接。在某些情况下，LAN 或 Intranet 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，请在从网络安装 Red Hat Enterprise Linux 时，选择带有您的存储库的最接近的区域。

有关配置网络连接的更多信息，请参阅 Red Hat Enterprise Linux 7 [安装指南中的网络与主机名章节](#)。

2.5. 安装后流程

以下步骤是在安装 Red Hat Enterprise Linux 后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. 尽管随着安装 Red Hat Enterprise Linux 会自动启用防火墙服务 `firewalld`，但在某些情况下，它可能会明确禁用，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`，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，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
```

要查看活跃的服务，请输入以下命令：

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

2.6. 其它资源

有关一般安装的详情，请查看 [Red Hat Enterprise Linux 7 安装指南](#)。

[2]

由于系统 **BIOS** 在制造商之间有所不同，一些可能不支持任一类型的密码保护，另一些则可能支持一种类型，但不支持另一种类型。

第 3 章 保持系统正常运行

本章论述了保持您的系统最新状态的过程，其中包括规划和配置安全更新的安装方式、应用新更新软件包引入的更改，以及使用红帽客户门户网站跟踪安全公告。

3.1. 维护安装的软件

随着安全漏洞的发现，必须更新受影响的软件，以限制任何潜在的安全风险。如果该软件是目前支持的 Red Hat Enterprise Linux 发行包的一部分，红帽会致力于发布更新的软件包，以便尽快修复漏洞。

通常，有关给定安全漏洞的公告附带了修复该问题的补丁（或源代码）。然后，这个补丁会应用到 Red Hat Enterprise Linux 软件包，并作为勘误更新进行测试和发布。但是，如果公告中不包含补丁，红帽开发人员首先与软件维护人员合作以解决问题。问题解决后，软件包会作为勘误更新进行测试并发布。

如果为您的系统中使用的软件发布了勘误更新，强烈建议您尽快更新受影响的软件包，以最小化系统可能受到攻击的时间。

3.1.1. 规划和配置安全更新

所有软件都包含错误。通常，这些漏洞可能会导致一个漏洞，将您的系统暴露给恶意用户。尚未更新的软件包是导致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，从而无法利用它们。

测试安全更新可用时，并将它们调度安装。需要使用其他控件来保护系统在更新发布和系统中安装它之间的时间。这些控件取决于确切的漏洞，但可能包括额外的防火墙规则、使用外部防火墙或在软件设置中更改。

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，并附带对特定勘误处理的问题的简短说明。所有勘误表都通过红帽订阅管理服务分发给具有有效订阅的客户。解决安全问题的勘误称为红帽安全公告。

有关使用安全勘误的详情请参考第 3.2.1 节“[在客户门户网站中查看安全公告](#)”。有关 Red Hat Subscription Management 服务的详细信息，包括如何从 RHN Classic 迁移的说明，请参见与该服务相关的文档：[Red Hat Subscription Management](#)。

3.1.1.1. 使用 Yum 的安全功能

Yum 软件包管理器包含多个与安全相关的功能，可用于搜索、列出、显示和安装安全勘误表。这些功能还使得可以使用 Yum 进行仅安装安全更新。

要检查您的系统可用的安全相关更新，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，因此可在脚本中使用该命令自动检查是否有可用的更新。当有任何可用的安全更新时，该命令会返回 100 的退出值，如果没有任何安全更新，则返回 0。在遇到错误时，它会返回 1。

同样，使用以下命令仅安装与安全相关的更新：

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

使用 **updateinfo** 子命令显示或操作存储库提供的有关可用更新的信息。**updateinfo** 子命令本身接受多个命令，其中一些与安全性相关的用途相关。有关这些命令的概述请查看 [表 3.1 “yum updateinfo 可用的与安全相关的命令”](#)。

表 3.1. yum updateinfo 可用的与安全相关的命令

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 advisories。
CVE	显示与 CVE 相关的信息子集（常见漏洞和风险）。
security 或 sec	显示所有与安全相关的信息。
[严重性_level 或 sev [severity_level]]	显示有关所提供严重性_级别的安全相关软件包的信息。

3.1.2. 更新和安装软件包

在系统上更新软件时，务必要从可信来源下载更新。攻击者可以轻松重建一个版本号与应该解决这个问题但会存在不同安全漏洞并在互联网上发布的软件包。如果发生这种情况，使用安全措施（例如针对原始 RPM 验证文件）不会检测到漏洞。因此，务必要仅从可信来源（如红帽）下载 RPM，并检查软件包签名以验证其完整性。

如需了解有关如何使用 **Yum** 软件包管理器的详细信息，请参见《红帽企业 Linux 7 系统管理员指南》中的 **Yum** 一章。

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，GnuPG 是用于确保分布式文件的真实性的免费软件包。如果对软件包签名的验证失败，软件包可能会被更改，因此不可信任。

Yum 软件包管理器允许自动验证安装或升级的所有软件包。此功能默认为启用。要在您的系统上配置这个选项，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上的[产品签名\(GPG\)密钥文章](#)。

3.1.2.2. 安装签名的软件包

要从您的文件系统安装已验证的软件包（有关如何验证软件包的信息），以 root 用户身份使用 `yum install` 命令，如下所示：[第 3.1.2.1 节“验证签名的软件包”](#)

```
yum install package_file.rpm
```

使用 `shell glob` 同时安装多个软件包。例如，以下命令会在当前目录中安装 `all.rpm` 软件包：

```
yum install *.rpm
```

重要

在安装任何安全勘误之前，请务必阅读勘误表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常规说明，请参阅[第 3.1.3 节“应用由已安装更新引入的更改”](#)。

3.1.3. 应用由已安装更新引入的更改

下载并安装安全勘误和更新后，必须停止旧软件的使用，再开始使用新软件。具体操作方式取决于已更新的软件的类型。以下列表列出了软件的一般类别，并提供软件包升级后使用更新版本的说明。



注意

通常，重新启动系统是确保使用最新版本的软件包的最可靠方法；但是，此选项并非始终必需，系统管理员也不始终可用。

应用程序

用户空间应用是可由用户发起的任何程序。通常，此类应用仅在用户、脚本或自动任务实用程序启动时才使用。

更新此类用户空间应用后，在系统上停止应用的任何实例，然后再次启动该程序以使用更新的版本。

内核

内核是红帽企业 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。

由于它的核心角色，如果不同时重新启动计算机，内核就无法重新启动。因此，重启系统之前，无法使用内核的更新版本。

KVM

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

使用 `lsmod` 命令确定加载了以下哪些模块：`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 **glibc** 等代码单元，供多个应用和服务使用。利用共享库的应用程序通常会在应用初始化时加载共享代码，因此必须停止并重新启动使用更新的库的任何应用程序。

要确定针对特定库运行的应用程序链接，请使用 **lsuf** 命令：

lsuf library

例如，要确定在 **libwrap.so.0** 库上运行哪些应用程序链接，请输入：

```
~]# lsuf /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
```

此命令返回使用 **TCP** 打包程序进行主机访问控制的所有正在运行的程序的列表。因此，在更新 **tcp_wrappers** 软件包时，必须停止并重新启动所有列出的程序。

systemd 服务

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

由于这些程序通常在内存中保留，只要计算机正在运行，必须停止每个更新的 **systemd** 服务并在其软件包升级后重新启动。这可以作为 **root** 用户使用 **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

按照下面链接的资源中所述，正确更新以下应用程序。

- **Red Hat Directory Server** - 请参阅有关红帽目录服务器版本的发行说明：
https://access.redhat.com/documentation/en-US/Red_Hat_Directory_Server/
-

Red Hat Enterprise Virtualization Manager - 请参阅 Red Hat Enterprise Virtualization 版本的 安装指南 (https://access.redhat.com/documentation/en-US/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。您可以使用它查找文档、管理订阅、下载产品和更新、打开支持案例以及了解安全更新。

3.2.1. 在客户门户网站中查看安全公告

要查看与您有效订阅的系统相关的安全公告（勘误），请访问 <https://access.redhat.com/> 并点击主页面上的 **Download Products & Updates** 按钮。当您进入 **Software & Download Center** 页面时，请点击 **Errata** 按钮来查看与您注册的系统相关的公告列表。

要浏览所有有效红帽产品的所有安全更新列表，请访问页面顶部的导航菜单，进入安全 安全更新活动 产品。

点击表左侧的勘误代码显示有关单个公告的更多详细信息。下一页不仅包含给定勘误的描述，包括其原因、结果和所需修复，还包含特定勘误更新的所有软件包的列表，以及关于如何应用更新的说明。该页面还包括相关参考链接，如相关的 CVE。

3.2.2. CVE 客户门户网站页面导航

CVE（通用漏洞和风险）项目由 MITRE 公司维护，是漏洞和安全风险的标准化名称列表。要在客户门户网站中浏览与红帽产品相关的 CVE 列表，请登录 [https://access.redhat.com/ Security](https://access.redhat.com/Security) → **Resources** → **CVE Database**。

单击表格左侧的 CVE 代码，以显示有关各个漏洞的更多详细信息。下一页不仅包含给定 CVE 的说明，还包括受影响红帽产品的列表以及相关红帽勘误的链接。

3.2.3. 了解问题严重分级

根据问题的严重性，红帽产品安全团队会为红帽产品安全团队分配影响等级。四点评级由以下级别组成：Low、Moderate、Important 和 Critical。此外，每个安全问题都使用通用漏洞评分系统 (CVSS) 基本评分。

这些评级可帮助您了解安全问题的影响，让您能够为系统排程和优先升级策略。请注意，评级反映了给定漏洞的潜在风险，此漏洞基于对漏洞的技术分析，而非当前的威胁级别。这意味着，如果针对特定漏洞发布了漏洞漏洞，则安全影响评级不会改变。

要查看客户门户中个别严重级别等级的详细描述，请访问 [Severity Ratings](#) 页面。

3.3. 其它资源

有关安全更新、应用方法、红帽客户门户和相关主题的更多信息，请参见以下列出的资源。

安装的文档

- [yum\(8\)](#) - Yum 软件包管理器的 man page 提供了有关 Yum 可用于在您的系统上安装、更新和删除软件包的方式的信息。
- [rpmkeys\(8\)](#) - rpmkeys 实用工具的 man page 描述了这个程序可以用来验证下载软件包的真实方法。

在线文档

- [红帽企业 Linux 7 系统管理员指南](#) - 红帽企业 Linux 7 的系统管理员指南介绍了用于在红帽企业 Linux 7 系统上安装、更新和删除软件包的 Yum 和 rpm 命令的使用。
- [Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide](#) - 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

- [红帽客户门户（安全）](#) - 客户门户网站的安全部分包含最重要的资源（包括红帽 CVE 数据库和红帽产品安全团队的联系人）。
- [红帽安全博客](#) - 红帽安全专家提供的最新安全相关问题文章。

另请参阅

- [第 2 章 安装的安全提示](#) 描述如何从一开始安全地配置您的系统，以便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 密钥来验证您的通信。

第 4 章 使用工具和服务强化您的系统

4.1. 桌面安全

红帽企业 Linux 7 提供了多种方法来强化桌面免受攻击和防止未经授权的访问。本节论述了有关用户密码、会话和帐户锁定以及可移动介质安全处理的建议做法。

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。这就是密码安全性对于保护用户、工作站和网络非常重要的原因。

出于安全考虑，安装程序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。

如果在安装期间取消选择影子密码，则所有密码都作为单向哈希存储在全局可读的 `/etc/passwd` 文件中，这使得系统易受离线密码破解攻击。如果入侵者能够以普通用户身份访问计算机，他可以将 `/etc/passwd` 文件复制到自己的计算机上，并针对它运行任意数量的密码破解程序。如果文件中存在不安全的密码，则仅需等待密码破解程序发现密码。

影子密码通过将密码哈希存储在文件 `/etc/shadow` 中（仅可由 root 用户读取）来消除这种类型的攻击。

这强制潜在攻击者通过登录计算机上的网络服务（如 SSH 或 FTP）来远程尝试窃取密码。这种暴力攻击的速度非常慢，随着数百次登录尝试写入到系统文件，会出现明显的跟踪。当然，如果攻击者在密码较弱的系统上夜间开始攻击，破解者在进入并编辑日志文件以覆盖他的记录之前便获得了访问权限。

除了格式和存储注意事项之外，还需要考虑内容问题。为了防止其帐户遭到密码破解攻击，用户可以执行的一项最重要的事情就是创建强大的密码。



注意

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

- [Red Hat Identity Management 简介](#)
- [定义密码策略](#)

4.1.1.1. 创建 Strong 密码

在创建安全密码时，用户必须记住，很长的密码比短而复杂的密码强大。创建仅包含八个字符的密码并不理想，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，因此个人很难记住这些密码。

在信息理论中，熵是与随机变量相关的不确定性级别，以位数表示。熵值越大，密码安全性越高。根据 NIST SP 800-63-1，未在由 50000 个常用选择密码组成的字典中不存在的密码应至少具有 10 位熵。因此，由四个随机单词组成的密码包含大约 40 位熵。包含多个用于添加安全性的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简单修改，例如，将第一个字符更改为大写字母并附加"1!"。请注意，此类修改不会显著提高密码的安全性。

自我创建密码的另一种方法是使用密码生成器。Thepwmake 是用于生成随机密码的命令行工具，由全部四组字符组成，即大写、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 中拉取。您可以指定的最少位数是 56，这足以满足暴力攻击的系统和服务的密码。64 位适合攻击者无法直接访问密码哈希文件的应用程序。对于攻击者可能获得对密码哈希的直接访问权限或将密码用作加密密钥的情况，应该使用 80 到 128 位。如果您指定了无效的熵位数，pwmake 将使用默认位数。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有所不同，但请务必避免以下错误做法：

- 使用单个字典单词、外部语言中的词语、颠倒的词语或仅数字。

- 将少于 10 个字符用作密码或密码短语。
- 使用键盘布局中的一系列键。
- 写下您的密码。
- 使用密码中的个人信息，如生日、横线、成员姓名或宠物名称。
- 在多台计算机上使用相同的密语或密码。

在创建安全密码的同时，务必要对它们进行正确管理，特别是对于大型企业内的系统管理员而言。以下小节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. 强制使用 Strong 密码

如果组织拥有大量用户，系统管理员可以通过两个基本选项来强制使用强密码：他们可以为用户创建密码，也可以让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确，但随着组织的扩展，它是一项艰巨的任务。它还会增加用户写密码的风险，从而公开密码。

因此，大多数系统管理员更喜欢让用户创建自己的密码，但会主动验证这些密码是否足够强大。在某些情况下，管理员可以强制用户通过密码有效期定期更改其密码。

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



注意

在红帽企业 Linux 7 中，`pam_pwquality` PAM 模块取代 `pam_cracklib`，该模块在红帽企业 Linux 6 中用作密码质量检查的默认模块。它使用与 `pam_cracklib` 相同的后端。

pam_pwquality 模块用于根据一组规则检查密码的强度。其过程由两个步骤组成：首先检查提供的密码是否在字典中找到。如果没有，它会继续执行几个附加检查。**pam_pwquality** 与 `/etc/pam.d/passwd` 文件的密码组件中的其他 PAM 模块一起堆叠，自定义规则集在 `/etc/security/pwquality.conf` 配置文件中指定。有关这些检查的完整列表，请查看 `pwquality.conf(8)` 手册页。

例 4.1. 配置密码强度检查 `inpwquality.conf`

要使用 **pam_quality** 启用，请在 `/etc/pam.d/passwd` 文件中的密码堆栈中添加以下行：

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

检查的选项每行指定一个。例如，若要要求密码长度最少为 8 个字符（包括所有四类字符），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连续字符设置密码强度检查，请在 `/etc/security/pwquality.conf` 中添加以下行：

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

在这个示例中，输入的密码在单例序列中不能包含 3 个以上字符，如 `abcd`，以及 3 个以上的连续字符，如 `1111`。

注意

由于 **root** 用户是强制执行密码创建规则的用户，他们可以为自己或普通用户设置任何密码，尽管有警告消息。

4.1.1.3. 配置密码期限

密码期限是系统管理员用来防止组织内错误密码的另一种技巧。密码过期意味着，在指定期间（通常为 90 天）后，会提示用户创建新密码。其背后的理论是，如果强制用户定期更改其密码，破解的密码仅在有限时间内对入侵者有用。但是，密码过期的缺点是用户更有可能写下密码。

要在 Red Hat Enterprise Linux 7 中指定密码期限，请使用 `chage` 命令。



重要

在 Red Hat Enterprise Linux 7 中，默认启用 shadow 密码。如需更多信息，请参阅《红帽企业 Linux 7 系统管理员指南》。

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

```
chage -M 90 username
```

在上述命令中，将 `username` 替换为用户名称。要禁用密码过期，请在 `-M` 选项后使用 `-1` 值。

有关 `chage` 命令可用选项的更多信息，请参考下表。

表 4.1. `chage` 命令行选项

选项	描述
<code>-d days</code>	指定自 1 月 1 日以来的天数，即 1970 年更改密码的天数。
<code>-e date</code>	以 YYYY-MM-DD 格式指定帐户锁定的日期。也可以使用 1970 年 1 月 1 日起的天数，而不是日期。
<code>-I days</code>	指定在锁定帐户之前密码过期后的非活动天数。如果值为 <code>0</code> ，则帐户在密码过期后不会被锁定。
<code>-l</code>	列出当前的帐户过期设置。
<code>-m days</code>	指定用户必须更改密码的最小天数。如果值为 <code>0</code> ，则密码不会过期。
<code>-M days</code>	指定密码有效的最长天数。如果此选项指定的天数加上通过 <code>-d</code> 选项指定的天数小于当前日期，用户必须在使用该帐户之前更改密码。
<code>-W days</code>	指定密码到期日期前的天数，以提醒用户。

您还可以在交互模式中使用 `chage` 命令修改多个密码期限和帐户详细信息。使用以下命令进入互动模式：

```
chage <username>
```

以下是使用这个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第一次登录时过期。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 实用程序具有可设置空密码的选项。使用空密码时，使用空密码是一种高度不安全的做法，因为任何第三方都可以使用不安全的用户名登录和访问系统。尽可能避免使用空密码。如果不可能，请务必确保用户已准备好登录，然后再解锁具有空密码的帐户。

2.

以 root 用户身份运行以下命令强制立即过期密码：

```
chage -d 0 username
```

此命令将上次更改为 **epoch** 的日期的值设置为 **epoch**(January 1, 1970)。此值会强制立即过期密码，无论是否有密码过期策略（若有）。

首次登录后，系统将提示用户输入新密码。

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的尝试失败后锁定用户帐户。限制用户登录尝试主要是一种安全措施，旨在防止可能针对获取用户帐户密码的暴力攻击。

使用 `pam_faillock` 模块时，失败的登录尝试会存储在 `/var/run/faillock` 目录中每个用户的单独文件中。



注意

失败尝试日志文件中的行顺序非常重要。此顺序的任何更改都可锁定所有用户帐户，包括使用 `even_deny_root` 选项时 `root` 用户帐户。

按照以下步骤配置帐户锁定：

1.

要在 10 分钟后尝试三次并解锁该用户后锁定任何非 `root` 用户，请在 `/etc/pam.d/system-auth` 和 `/etc/pam.d/password-auth` 文件的 `auth` 部分添加两行。编辑后，这两个文件中的整个 `auth` 部分应如下所示：

```
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.

在上一步中指定两个文件的 `account` 部分中添加以下行：

```
account required pam_faillock.so
```

3.

要为 `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** 在无法登录三次后尝试登录四次时，在第一次尝试时会锁定他的帐户：

```
~J$ 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，请在首次在 `/etc/pam.d/system-auth` 和 `/etc/pam.d/password-auth` 中第一次调用 `pam_faillock` 的行上方添加以下行。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
```

要查看每位用户失败的尝试次数，以 **root** 用户身份运行以下命令：

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

要解锁用户帐户，以 **root** 用户身份运行以下命令：

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

重要

运行 **cron** 作业会重置正在运行 **cron** 作业的 `pam_faillock` 的失败计数器，因此不应为 **cron** 配置 `pam_faillock`。如需更多信息，请参阅[知识中心支持\(KCS\)解决方案](#)。

使用 `authconfig` 保留自定义设置

使用 `authconfig` 实用程序修改身份验证配置时，`system-auth` 和 `password-auth` 文件会被 `authconfig` 实用程序的设置覆盖。这可以通过创建符号链接来代替配置文件（`authconfig` 可识别且不会覆盖这些文件）来避免这种情况。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` 文件是否指向 `system-auth-ac` 和 `password-auth-ac`（这是系统默认）的符号链接：

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

如果输出结果类似如下，符号链接会就位，您可以跳过第 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。

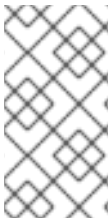
删除 nullok 选项

`nullok` 选项允许用户使用空白密码登录，如果 `/etc/shadow` 文件中的密码字段为空，则默认启用。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`。

请参见 [Will 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

由于日常操作期间的许多原因，用户可能需要无人值守的工作站。这为攻击者提供了物理访问计算机的机会，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”](#)）。由于笔记本电脑的移动干扰物理安全性，因此这些笔记本电脑会特别暴露。您可以使用会话锁定功能来缓解这些风险，该功能阻止访问系统，直到输入了正确的密码。



注意

锁定屏幕而非注销的主要优点在于锁定允许用户的进程（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

用户可能还需要锁定虚拟控制台。可以使用名为 `vlock` 的实用程序完成此操作。要安装这个工具，以 `root` 用户身份执行以下命令：

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


安装后，可以使用 `vlock` 命令锁定任何控制台会话，而无需任何附加参数。这会锁定当前活动的虚拟控制台会话，同时仍允许访问其他虚拟控制台。要阻止访问工作站中的所有虚拟控制台，请执行以下操作：

```
vlock -a
```

在这种情况下，`vlock` 会锁定当前活动的控制台，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. 强制只手动挂载可移动介质

要强制以只读方式挂载可移动介质（如 USB 闪存磁盘），管理员可以使用 `udev` 规则检测可移动介质并使用 `blockdev` 实用程序将其配置为只读挂载。这足以强制以只读方式挂载物理介质。

使用 `blockdev` 强制只读挂载可移动介质

要强制以只读方式挂载所有可移动介质，请使用以下内容创建一个新的 `udev` 配置文件，例如：`/etc/udev/rules.d/` 目录中的 `80-readonly-removables.rules`：

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

以上 `udev` 规则确保使用 `blockdev` 实用程序自动将任何新连接的可移动块（存储）设备配置为只读。

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，但不会将新设置应用到现有的设备。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，以确保新设置在下次插入时应用到它们。

要强制 `udev` 重新为现有设备应用所有规则，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已经挂载的任何存储设备。

要强制 **udev** 重新载入所有规则（如果由于某种原因无法自动检测到新规则），请使用以下命令：

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

4.2. 控制根访问

在管理主计算机时，用户必须以 **root** 用户身份执行某些任务，或使用 **setuid** 程序（如 **sudo** 或 **su**）获得有效的 **root** 权限。**setuid** 程序是使用程序所有者而不是操作程序的用户的用户 ID (UID) 运行的程序。此类程序由长格式列表的 **owner** 部分中的 **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 可以是大写或小写。如果它显示为大写，则表示尚未设置底层权限位。

但是，对于组织的系统管理员而言，必须选择组织内的管理访问用户应对其计算机进行多少管理访问。通过名为 **pam_console.so** 的 PAM 模块，对于在物理控制台中登录的第一个用户，通常仅为 **root** 用户保留的某些活动，如重新引导和挂载可移动介质等。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其网络上的用户应获得的访问量。

4.2.1. 禁止 Root 访问

如果管理员由于这些或其他原因而允许用户以 **root** 身份登录，则 **root** 密码应保持机密，并且应当禁止通过引导装载程序密码保护访问运行级别一个或多个用户模式（请参阅 [第 4.2.5 节“保护引导装载程序”](#)）。

以下是管理员可以进一步确保不允许 **root** 登录的四种不同方式：

更改 root shell

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

表 4.2. 禁用 Root Shell

影响	不受影响
<p>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：</p> <ul style="list-style-type: none">logingdmkdmxdmsusshscpsftp	<p>不需要 shell 的程序，如 FTP 客户端、邮件客户端和多个 setuid 程序。以下程序没有阻止访问 root 帐户：</p> <ul style="list-style-type: none">sudoFTP 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，管理员可以编辑 **/etc/securetty** 文件，在控制台上禁用 **root** 登录。此文件列出了 **root** 用户被允许登录的所有设备。如果文件完全不存在，**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** 用户身份登录其计算机，它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** 用户在实际连接到计算机的控制台中进行登录。要防止 **root** 用户以 **root** 身份登录，请以 **root** 用户身份在 **shell** 提示符后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
```

下面列出的文件：

- **/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` 工具套件远程登录，因为在验证后才会打开控制台。

表 4.3. 禁用根登录

影响	不受影响
<p>防止使用控制台或网络访问 <code>root</code> 帐户。以下程序无法访问 <code>root</code> 帐户：</p> <ul style="list-style-type: none"><code>login</code><code>gdm</code><code>kdm</code><code>xdm</code>打开 <code>tty</code> 的其他网络服务	<p>不以 <code>root</code> 身份登录的程序，而是通过 <code>setuid</code> 或其他机制执行管理任务.以下程序没有阻止访问 <code>root</code> 帐户：</p> <ul style="list-style-type: none"><code>su</code><code>sudo</code><code>ssh</code><code>scp</code><code>sftp</code>

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`，并更改以下行：

#PermitRootLogin yes

内容如下：

PermitRootLogin no

表 4.4. 禁用 Root SSH 登录

影响	不受影响
<p>使用 OpenSSH 工具套件防止 root 访问。以下程序无法访问 root 帐户：</p> <ul style="list-style-type: none">sshscpsftp	<p>不属于 OpenSSH 工具套件的程序。</p>

使用 PAM 限制对服务的 root 访问权限

PAM 通过 /lib/security/pam_listfile.so 模块，提供了极大的灵活性来拒绝特定帐户。管理员可以使用此模块引用不允许登录的用户列表。若要限制对系统服务的 root 访问权限，请在 /etc/pam.d/ 目录中编辑目标服务的文件，并确保需要 pam_listfile.so 模块进行身份验证。

以下是 /etc/pam.d/vsftpd PAM 配置文件（如果指令位于一行中，则不需要第一行末尾的 \ 字符）：

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

■

这将指示 PAM 查阅 `/etc/vsftpd.ftpusers` 文件，并拒绝任何列出的用户对该服务的访问。管理员可以更改此文件的名称，并且可以为各个服务保留单独的列表，或者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访问多个服务，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`，该指南位于 `/usr/share/doc/pam-<version>/html/` 目录中。

表 4.5. 使用 PAM 禁用 Root

影响	不受影响
----	------

影响	不受影响
<p>阻止对 PAM 可识别的网络服务的根访问。以下服务无法访问 root 帐户：</p> <ul style="list-style-type: none">logingdmkdmxdmsshscpsftpFTP 客户端电子邮件客户端任何 PAM 识别服务	<p>PAM 可识别的程序和服务。</p>

影响

不受影响

4.2.2. 允许 Root 访问

如果组织内的用户是值得信任且计算机拆分的，那么允许他们进行 root 访问可能不是问题。允许用户进行 root 访问意味着次要活动（如添加设备或配置网络接口）可由个人用户处理，使系统管理员可以自由地处理网络安全性和其他重要问题。

另一方面，为单个用户授予 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配置其计算机并需要帮助来解决问题。更糟糕的是，他们可能会在不了解的情况下打开安全漏洞。
- 运行不安全服务** - 具有 root 访问权限的用户可能会在其计算机上运行不安全服务器，如 FTP 或 Telnet，可能会使用户名和密码面临风险。这些服务以纯文本形式通过网络传输此信息。
- 将电子邮件附件作为根运行** - 虽然很少，但影响 Linux 的电子邮件病毒确实存在。恶意程序在由 root 用户运行时构成最大的威胁。
- 保持审核跟踪完整** - 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中的哪个用户在给定时间是 root 用户。当使用单独的登录时，用户使用登录的帐户以及用于会话跟踪的唯一编号将放入任务结构中，由用户启动的每个进程继承。使用并发登录时，唯一数字可用于跟踪到特定登录的操作。当操作生成审计事件时，它将使用登录帐户和与该唯一编号关联的会话进行记录。使用 `ausearch` 命令查看这些登录和会话。可使用 `ausearch` 命令的 `--proof` 选项建议特定的 `ausearch` 查询隔离特定会话生成的可审计事件。有关 Audit 系统的详情请参考 [第 7 章 系统审核](#)。

4.2.3. 限制 Root 访问

管理员可能希望仅通过 `setuid` 程序（如 `su` 或 `sudo`）来允许对 root 用户的访问，而不是完全拒绝访问。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)`man page。

4.2.4. 启用自动注销

当用户以 root 身份登录时，无人值守的登录会话可能会带来重大的安全风险。要降低此风险，您可以将系统配置为在固定的时间段后自动注销空闲用户。

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

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

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

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

如果没有指定秒数的活动（上例中的 设置为 120），**TMOUT** 变量将终止 **shell**。您可以根据特定安装的需求更改限制。

4.2.5. 保护引导装载程序

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 - 如果攻击者可以引导系统进入单用户模式，则它们会自动以 **root** 身份登录，而无需提示输入 **root** 密码。



警告

不建议通过编辑 **/etc/sysconfig/init** 文件中的 **SINGLE** 参数来使用密码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。建议您对 **GRUB 2** 引导加载程序进行密码保护，如《[Red Hat Enterprise Linux 7 系统管理员指南](#)》使用密码保护 **GRUB 2** 中所述。

2. 防止访问 **GRUB 2** 控制台 - 如果计算机使用 **GRUB 2** 作为启动加载器，攻击者可以使用 **GRUB 2** 编辑器界面更改其配置或使用 **cat** 命令收集信息。
3. 防止访问 **Insecure** 操作系统 - 如果是双引导系统，攻击者可在引导时选择操作系统，例如 **DOS**，它忽略访问控制和文件权限。

Red Hat Enterprise Linux 7 在 Intel 64 和 AMD64 平台上包含 GRUB 2 引导装载程序。有关 GRUB 2 的详细介绍，请参阅《Red Hat Enterprise Linux 7 系统管理员指南》的使用 [GRUB 2 引导加载器](#) 章节。

4.2.5.1. 禁用交互式启动

通过在启动序列开始时按 I 键，您可以以交互方式启动您的系统。在交互式启动期间，系统会提示您逐个启动每个服务。但是，这可以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获得系统访问权限。

要防止用户以 root 身份以互动方式启动系统，请在 `/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受保护的硬链接和符号链接导致的潜在漏洞，Red Hat Enterprise Linux 7 包含了只允许创建或遵循链接的功能，只要满足某些条件。

如果有硬链接，需要满足以下条件之一：

- 用户拥有他们链接到的文件。
- 用户已对其链接的文件具有读写访问权限。

如果出现符号链接，仅在带有粘滞位的世界可写入目录之外或以下其中一个需要满足以下条件时，才允许进程跟踪链接：

- 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。

默认情况下将打开此保护。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，请使用以下内容创建一个名为 `51-no-protect-links.conf` 的新配置文件，例如：`/etc/sysctl.d/` 目录中的 `51-no-protect-links.conf`：

```
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，而且需要在默认系统文件后读取该文件（这些文件按字典顺序读取，因此优先使用文件名开头带有更高数字的文件中包含的设置）。

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

虽然用户对管理控制的访问权限对企业内的系统管理员而言是一个重要问题，但监控哪个网络服务处于活动状态对任何管理和运行 Linux 系统的人员而言至关重要。

红帽企业 Linux 7 中的许多服务都是网络服务器。如果网络服务正在计算机上运行，则服务器应用程序（称为守护进程）正在侦听一个或多个网络端口上的连接。其中每个服务器都应被视为潜在的攻击途径。

4.3.1. 服务风险

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：

- **拒绝服务攻击(DoS)** - 通过请求占用服务时，拒绝服务攻击可能会导致系统无法使用，因为它尝试记录并回答每个请求。
- **分布式拒绝服务攻击(DDoS)** - 一种 DoS 攻击类型，它使用多台受入侵机器（通常数以千计或更长时间）来指示对服务的协同攻击，将其与请求充满，并使其无法使用。
- **脚本漏洞攻击** - 如果服务器使用脚本执行服务器端操作，如 Web 服务器通常执行，攻击者可

以将编写错误的脚本作为目标。这些脚本漏洞攻击可能会导致缓冲区溢出状况，或允许攻击者更改系统上的文件。

•

缓冲区溢出攻击 - 希望侦听端口 1 到 1023 的服务必须以管理权限启动，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该端口后，通常会丢弃特权或功能。如果没有丢弃特权或功能，并且应用具有可利用的缓冲区溢出，攻击者能够以运行守护进程的用户身份获取系统访问权限。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，一旦获得访问权限，他们便使用自动化的根基片来维护其对系统的访问权限。

注意

在红帽企业 Linux 7 中，缓冲区溢出漏洞的风险由 ExecShield 缓解，这是受 x86 兼容单处理器和多处理器内核支持的可执行内存分段和保护技术。ExecShield 通过将虚拟内存划分为可执行和非可执行文件片段来降低缓冲区溢出的风险。任何试图在可执行段外执行的程序代码（如从缓冲区溢出利用而注入的恶意代码）都会触发分段错误并终止。

ExecShield 还包括对 AMD64 平台和 Intel® 64 系统中无 eXecute (NX) 技术的支持。这些技术可与 ExecShield 配合使用，防止恶意代码以细粒度为 4KB 的可执行文件代码在虚拟内存的可执行部分运行，从而降低受缓冲区溢出漏洞攻击攻击的风险。

重要

为限制网络受到攻击的风险，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，随红帽企业 Linux 7 一起安装的大多数网络服务都默认处于关闭状态。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 - 替代打印服务器。

•

xinetd - 控制到一系列从属服务器（如 gssftp 和 telnet）的连接的超级服务器。

-

sshd - OpenSSH 服务器，这是 Telnet 的安全替代品。

在确定是否让这些服务运行时，最好使用常识并避免承担任何风险。例如，如果打印机不可用，则不要让 cups 保持运行。portreserve 也是如此。如果您不挂载 NFSv3 卷或使用 NIS (ypbind 服务)，则应禁用 rpcbind。检查哪些网络服务可以在引导时启动是不够的。建议您还要检查哪些端口已经打开并正在侦听。如需更多信息，请参阅 [第 4.4.2 节“验证正在侦听哪些端口”](#)。

4.3.3. 不安全的服务

或许，任何网络服务都不安全。这就是为什么关闭未使用的服务非常重要。服务漏洞定期被发现和修补，这使得定期更新与任何网络服务相关的软件包非常重要。如需更多信息，请参阅 [第 3 章 保持系统正常运行](#)。

些网络协议本质上比其他协议更加不安全。这包括以下任何服务：

-

通过未加密的网络传输用户名和密码 - 许多较旧的协议（如 Telnet 和 FTP）不加密身份验证会话，应该尽可能避免。

-

通过网络未加密传输敏感数据 - 许多协议通过网络未加密传输数据。这些协议包括 Telnet、FTP、HTTP 和 SMTP。NFS 和 SMB 等许多网络文件系统也通过网络未加密传输信息。使用这些协议限制传输的数据类型时，用户的责任。

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 的更多信息，请参阅 [第 4.3.11 节“保护 SSH”](#)。

FTP 不像远程 shell 一样对系统安全性具有固有危险，但必须仔细配置和监控 FTP 服务器以避免出现问题。有关保护 FTP 服务器的详情，请参考 [第 4.3.9 节“保护 FTP”](#)。

应谨慎实施的服务和防火墙后的服务包括：

-

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

Therpcbind 服务是用于 RPC 服务（如 NIS 和 NFS）的动态端口分配守护进程。它具有较弱的身份验证机制，能够为其控制的服务分配广泛的端口。由于这些原因，很难保证。



注意

Securingrpcbind 仅影响 NFSv2 和 NFSv3 实施，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，并应用以下部分：

如果运行 RPC 服务，请遵循这些基本规则。

4.3.4.1. 使用 TCP wrappers 保护 rpcbind

使用 TCP wrappers 来限制哪些网络或主机有权访问 rpcbind 服务非常重要，因为它没有内置的身份验证形式。

此外，仅在限制对服务的访问时使用 IP 地址。避免使用主机名，因为可以通过 DNS 投毒和其他方法对其进行伪造。

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。

以下是两个 `firewalld` 富语言命令示例：第一种命令允许 TCP 从 192.168.0.0/24 网络连接到端口 111（由 `rpcbind` 服务使用）。第二个命令允许 TCP 从本地主机连接到同一端口。所有其他数据包都会丢弃。

```
~]# 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` 富语言命令，使设置永久保留。有关实现防火墙的更多信息，请参阅 [第 5 章 使用防火墙](#)。

4.3.5. 保护 rpc.mountd

The `rpc.mountd` 守护进程实施 NFS MOUNT 协议的服务器端，这是 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，请遵循这些基本规则。

4.3.5.1. 使用 TCP wrappers 保护 rpc.mountd

使用 `TCP wrappers` 来限制哪些网络或主机有权访问 `rpc.mountd` 服务非常重要，因为它没有内置的身份验证形式。

此外，仅在限制对服务的访问时使用 IP 地址。避免使用主机名，因为可以通过 DNS 投毒和其他方法对其进行伪造。

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，请将 `firewalld` 丰富的语言规则添加到服务器并限制对特定网络的访问。

以下是两个 `firewalld` 富语言命令示例：第一种命令允许从 `192.168.0.0/24` 网络 挂载的连接。第二个命令允许从本地主机挂载的连接。所有其他数据包将被丢弃。

```
~]# 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` 富语言命令，使设置永久保留。有关实现防火墙的更多信息，请参阅 [第 5 章 使用防火墙](#)。

4.3.6. 保护 NIS

网络信息服务(NIS)是一种 RPC 服务，称为 `ypserv`，它与 `rpcbind` 和其他相关服务一起使用，以向声称在其域中的任何计算机分发用户名、密码和其他敏感信息的映射。

NIS 服务器由多个应用程序组成。它们包括以下几项：

- `/usr/sbin/rpc.yppasswdd` - 也称为 `yppasswdd` 服务，此后台程序允许用户更改其 NIS 密码。
- `/usr/sbin/rpc.ypxfrd` - 也称为 `ypxfrd` 服务，此后台程序负责通过网络进行 NIS 映射传输。
- `/usr/sbin/ypserv` - 这是 NIS 服务器守护进程。

根据当今的标准，NIS 稍微不安全。它没有主机身份验证机制，并且没有通过网络未加密传输其所有信息，包括密码哈希。因此，设置使用 NIS 的网络时必须特别小心。这因 NIS 默认配置本质上不安全这一事实而变得更加复杂。

建议计划实施 NIS 服务器的用户首先保护 [第 4.3.4 节“保护 rpcbind”](#) 中概述的 rpcbind 服务，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 会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、分段和安全的网络上运行。每当通过不安全的网络传输 NIS 信息时，可能会被截获。谨慎的网络设计可帮助防止严重安全漏洞。

4.3.6.2. 使用类似密码 NIS 域名和主机名

只要用户知道 NIS 服务器的 DNS 主机名和 NIS 域名，则 NIS 域内的任何计算机都可以使用命令从服务器中提取信息，而无需验证。

例如，如果有人将笔记本电脑连接到网络或者从外部连接到网络（并管理欺骗内部 IP 地址），以下命令会显示 `/etc/passwd` 映射：

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

如果这个攻击者是 root 用户，则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 的访问更难以攻击者，请为 DNS 主机名创建一个随机字符串，如 `o7hfawtgmhgw.domain.com`。同样，创建不同的随机 NIS 域名。这使得攻击者更难访问 NIS 服务器。

4.3.6.3. 编辑 `/var/yp/securenets` 文件

如果 `/var/yp/securenets` 文件为空白或不存在（如同默认安装后的情况），NIS 会监听所有网络。首先需要将子网掩码/网络对放入文件中，以便 `ypserv` 仅响应来自相应网络的请求。

以下是 `/var/yp/securenets` 文件中的条目示例：

255.255.255.0 192.168.0.0

**警告**

不得首次启动 NIS 服务器，而不创建 `/var/yp/securenets` 文件。

该技术不会提供 IP 欺骗攻击的保护，但它至少对 NIS 服务器服务的网络施加限制。

4.3.6.4. 分配静态端口和使用丰富语言规则

可以为所有与 NIS 相关的服务器分配除 `forrpc.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以进一步保护 NIS 服务器守护进程免受入侵者的影响。

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

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

然后，可以使用以下丰富的语言 `firewalld` 规则来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，则服务器仅允许连接端口 `834` 和 `835`。第一条规则适用于 TCP，第二个规则用于 UDP。

**注意**

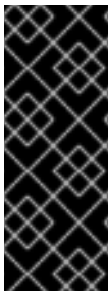
有关使用 `iptables` 命令实现防火墙的详情，请查看 [第 5 章 使用防火墙](#)。

4.3.6.5. 使用 Kerberos 身份验证

当使用 NIS 进行验证时需要考虑的问题之一是：每当用户登录计算机时，会通过网络发送来自 `/etc/shadow` 映射的密码哈希。如果入侵者获得 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。在足够时间内，密码破解程序可能会猜测弱密码，攻击者可以访问网络上的有效帐户。

由于 Kerberos 使用密钥加密，因此永远不会通过网络发送密码哈希，从而使系统更加安全。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 IdM 部分](#)。

4.3.7. 保护 NFS



重要

NFS 流量可以在所有版本中使用 TCP 发送，应与 NFSv3 而不是 UDP 一起使用，在使用 NFSv4 时需要该通信。所有版本的 NFS 支持 Kerberos 用户和组身份验证，作为 `RPCSEC_GSS` 内核模块的一部分。有关 `rpcbind` 的信息仍然包含在内，因为 Red Hat Enterprise Linux 7 支持使用 `rpcbind` 的 NFSv3。

4.3.7.1. 仔细规划网络

NFSv2 和 NFSv3 传统上不安全地传递数据。现在，所有版本的 NFS 都可以使用 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，文件锁定和挂载仍不使用 Kerberos。使用 NFSv4.0 时，如果客户端位于 NAT 后面或防火墙，则可能会关闭助理。有关使用 NFSv4.1 允许委派通过 NAT 和防火墙操作的详情，请参考 Red Hat Enterprise Linux 7 存储管理指南中的 [pNFS](#) 部分。

4.3.7.2. 保护 NFS 挂载选项

有关在 `/etc/fstab` 文件中使用 `mount` 命令的信息，请参见 [《Red Hat Enterprise Linux 7 存储管理指南》的使用挂载命令章节](#)。从安全管理角度来看，需要注意也可以在 `/etc/nfsmount.conf` 中指定 NFS 挂载选项，该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。在某些情况下，客户端可能会“破坏”文件系统导出的部分，并进入未导出的部分（请参阅 `exports(5)` `man page` 中关于子树检查的章节）。

尽可能使用 `ro` 选项将文件系统导出为只读，以减少能够写入挂载的文件系统的用户数量。仅在需要时才使用 `rw` 选项。如需更多信息，请参阅 `man exports(5)` 页。例如，允许写入访问会增加 `symlink` 攻击的风险。这包括 `/tmp` 和 `/usr/tmp` 等临时目录。

当必须使用 `rw` 选项挂载目录时，可以避免尽可能使目录全局可写入，以降低风险。导出主目录也被视为风险，因为某些应用将密码以明文或加密方式存储。随着应用程序代码的审核和改进，这种风险得以降低。有些用户不在其 `SSH` 密钥上设置密码，这也意味着主目录存在风险。强制使用密码或使用 `Kerberos` 将降低这一风险。

仅将导出限制为需要访问的客户端。在 `NFS` 服务器上使用 `showmount -e` 命令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它没有被使用。如需更多信息，请参阅第 4.3.7.4 节“不要使用 `no_root_squash` 选项”。

安全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，服务器仅允许预留端口”（编号小于 1024 “的端口）进行通信，因为传统客户端仅允许受信任的代码”（如内核内 `NFS` 客户端）使用这些端口。但是，在许多网络上，任何人都很难在某些客户端上成为 `root` 用户，因此，服务器几乎无法假定来自保留端口的通信具有特权。因此，对保留端口的限制值有限；最好依赖 `Kerberos`、防火墙和对特定客户端导出的限制。

大多数客户端仍会尽可能使用保留的端口。但是，保留端口是一个有限资源，因此客户端（特别是具有大量 `NFS` 挂载的客户端）也可以选择使用数值较高的端口。`Linux` 客户端可以使用“`noresvport`”挂载选项执行此操作。“如果要在导出中允许此操作，您可以使用不安全的导出选项这样做”。

最好不要允许用户登录到服务器。在查看 `NFS` 服务器上的上述设置时，将检查谁以及可以访问该服务器的内容。

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。使用此命令可防止用户无意执行放置在文件系统文件。`nosuid` 和 `noexec` 选项是大多数（如果不是全部）文件系统的标准选项。

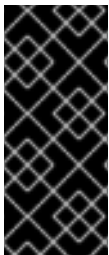
使用 `nodev` “选项可防止设备文件被客户端作为硬件设备处理”。

`resvport` 选项是一个客户端挂载选项，并且 `secure` 是对应的服务器端导出选项（请参阅上述说明）。它限制与“保留的端口”的通信。为特权用户和 `root` 用户等进程保留保留或“well known”端口。设置此选项可让客户端使用保留的源端口与服务器通信。

所有版本的 NFS 现在支持使用 Kerberos 身份验证进行挂载。启用此选项的挂载选项为：
`sec=krb5`。

NFSv4 支持使用 `krb5i` 为完整性而使用 Kerberos 挂载，为了保护隐私，使用 `krb5p` 进行挂载。使用 `sec=krb5` 挂载时会使用这些命令，但需要在 NFS 服务器上配置。如需更多信息，请参阅有关导出的 `man page`（`man 5 导出`）。

NFS `manpage`(`man 5 nfs`)有一个“SECURITY CONSIDERATIONS”部分，它解释了 NFSv4 中的安全增强功能并包含所有特定于 NFS 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。由于安全性以及某些兼容性原因，在 Kerberos 库中弃用和默认禁用 DES。如果您的环境不支持更新且更安全的算法，仅出于兼容性的原因，才使用 DES。

4.3.7.3. 留意语法错误

NFS 服务器通过查询 `/etc/exports` 文件确定要导出哪些文件系统以及要导出这些目录的主机。编辑此文件时，请不要添加无关的空格。

例如，`/etc/exports` 文件中的以下行将使用读/写权限将目录 `/tmp/nfs/` 共享给主机 `bob.example.com`：

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

另一方面，`/etc/exports` 文件中的以下行使用只读权限与主机 `bob.example.com` 共享相同的目录，并使用主机名后的单一空格字符将其共享给具有读/写权限的世界。

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

最好使用 `showmount` 命令检查配置的任何 NFS 共享，以验证正在共享的内容：

```
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` 用户更改为 `nfsnobody` 用户，这是非特权用户帐户。这会将所有根创建的文件的所有者更改为 `nfsnobody`，这会阻止使用 `setuid` 位集上传程序。

如果使用 `no_root_squash`，远程 `root` 用户可以更改共享文件系统上的任何文件，并使 Trojans 的应用保留给其他用户，让其他用户意外执行。

4.3.7.5. NFS 防火墙配置

NFSv4 是 Red Hat Enterprise Linux 7 的默认 NFS 版本，对于 TCP 仅需要打开端口 2049。如果使用 NFSv3，则需要另外四个端口，如下所述。

为 NFSv3 配置端口

用于 NFS 的端口由 `rpcbind` 服务动态分配，这可能会导致创建防火墙规则时出现问题。要简化这个过程，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** - 状态的 TCP 和 UDP 端口(`rpc.statd`)

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` 的 TCP 端口(`rpc.lockd`)
- **nlm_udpport** — UDP port `nlockmgr` (`rpc.lockd`)

指定的端口号不能供任何其他服务使用。将防火墙配置为允许指定的端口号，以及 TCP 和 UDP 端口 2049(NFS)。有关其他可定制 NFS 锁定管理器参数的说明，请参阅 `/etc/modprobe.d/lockd.conf`。

在 NFS 服务器上运行 `rpcinfo -p` 命令，以查看使用了哪些端口和 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 的环境中（包括在 Red Hat Enterprise Linux 中），可以大大简化对 Kerberos 感知的 NFS 设置。

请参阅《Red Hat Enterprise Linux 7 域身份、身份验证和策略指南》，特别是 设置 Kerberos 感知的 NFS 服务器，了解如何使用红帽身份管理时通过 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 服务器是红帽企业 Linux 7 中最稳定、最安全的服务之一。有很多选项和技术可以保护 Apache HTTP 服务器 - 太多，无法在这里深入介绍。下面的部分简要介绍了运行 Apache HTTP 服务器时的良好做法。

在将系统上运行的脚本投入生产之前，始终验证其上运行的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>
```

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

FollowSymLinks

默认情况下启用此指令，因此在创建指向 Web 服务器的文档根目录的符号链接时务必谨慎。例如，提供指向 `/` 的符号链接不是一个好主意。

索引

默认情况下启用此指令，但可能不可取。要防止访问者在服务器上浏览文件，请删除此指令。

UserDir

默认情况下禁用 **UserDir** 指令，因为它可以确认系统上存在用户帐户。要启用在服务器上浏览的用户目录，请使用以下指令：

```
UserDir enabled
UserDir disabled root
```

这些指令激活 `/root/` 之外的所有用户目录浏览用户目录。要将用户添加到禁用的帐户列表中，请在 **UserDir disabled** 行上添加一个以空格分隔的用户列表。

ServerTokens

ServerTokens 指令控制发送回客户端的服务器响应标头字段。它包括以下参数可以自定义的各种信息：

- **ServerTokens Full**（默认选项） - 提供所有可用的信息（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 Product onlyly** - 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
```

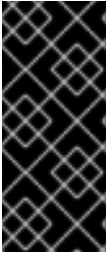
- **ServerTokens Min** 或 **ServerTokens Minimal** - 提供以下信息：

```
Apache/2.0.41
```

ServerTokens OS - 提供以下信息：

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

建议您使用 **ServerTokens Prod** 选项，以便可能的攻击者不会获得有关您的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，否则可能会让攻击者在系统上执行命令。

删除 httpd 模块

在某些情况下，删除某些 **httpd** 模块以限制 **HTTP** 服务器的功能很有用。为此，请编辑 **/etc/httpd/conf.modules.d** 目录中的配置文件。例如，要删除 **proxy** 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

要防止攻击者了解在您的服务器上运行的 **NGINX** 版本，请使用以下配置选项：

```
server_tokens    off;
```

这会产生删除版本号并只在由 NGINX 服务的所有请求中报告字符串 `nginx` 的效果：

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

包括其他与安全相关的标头

由 NGINX 提供的每个请求都可以包含额外的 HTTP 标头，以缓解某些已知的 Web 应用程序漏洞：

- **`add_header X-Frame-Options SAMEORIGIN;`** - 此选项拒绝域外的任何页面设置 NGINX 提供的任何内容，从而有效缓解了 clickjacking 攻击。
- **`add_header X-Content-Type-Options nosniff;`** - 这个选项可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";`** - 此选项允许跨站点脚本(XSS)过滤，这样可防止浏览器呈现 NGINX 响应中含有的潜在恶意内容。

禁用 Potentially Harmful HTTP 方法

如果启用，某些 HTTP 方法可能允许攻击者在专为开发人员测试 Web 应用的 Web 服务器上执行操作。例如，TRACE 方法已知允许跨站点跟踪(XST)。

您的 NGINX 服务器可以通过只将允许的方法列入白名单来禁止这些恶意 HTTP 方法以及任意方法。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

要保护由 NGINX Web 服务器服务的数据，请考虑仅通过 HTTPS 提供数据。要生成安全配置配置文件以在 NGINX 服务器中启用 SSL，请参阅 [Mozilla SSL 配置生成器](#)。生成的配置可确保禁用已知存在漏洞的协议（如 SSLv2 或 SSLv3、密码和哈希算法（如 3DES 或 MD5））。

您还可以使用 [SSL 服务器测试](#)来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)是较旧的 TCP 协议，旨在通过网络传输文件。由于与服务器的所有事务（包括用户身份验证）都未加密，因此它被视为不安全的协议，应仔细配置。

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 服务的一个独立、面向安全的实施。

以下安全准则适用于设置 vsftpd FTP 服务：

4.3.9.1. FTP Greeting Banner

在提交用户名和密码之前，所有用户都会看到一个问候横幅。默认情况下，此横幅包含了对攻击者有用的版本信息，试图识别系统中的弱点。

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>` 替换为问候语的文本。

对于 mutli-line 横幅，最好使用横幅文件。要简化多个横幅的管理，将所有横幅放在名为 `/etc/banners/` 的新目录中。本例中 FTP 连接的横幅文件为 `/etc/banners/ftp.msg`。以下是此类文件的示例：

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



注意

不需要按照 [第 4.4.1 节“使用 TCP wrapper 和 xinetd 保护服务”](#) 中指定的 220 开始文件的每一行。

要引用 `vsftpd` 的这个问候横幅文件，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 wrapper` 发送额外的横幅到传入的连接，如 [第 4.4.1.1 节“TCP 封装器和连接程序”](#) 所述。

4.3.9.2. 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。此软件包为匿名用户建立一个目录树，并将目录的权限配置为匿名用户的只读权限。

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/` 中创建仅写目录。要做到这一点，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该目录的内容：

```
~]# 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
```

允许匿名用户读取和写入目录的管理员通常发现其服务器成为被盗软件的存储库。

另外，在 `vsftpd` 下，将以下行添加到 `/etc/vsftpd/vsftpd.conf` 文件中：

```
anon_upload_enable=YES
```

4.3.9.3. 用户帐户

由于 FTP 通过不安全网络传输未加密的用户名和密码以进行身份验证，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的 FTP 访问，如 `root` 用户和具有 `sudo` 特权的用户，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`。

也可以直接在各个服务中禁用用户帐户。

要在 `vsftpd` 中禁用特定用户帐户，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 之间传递电子邮件

并通过电子邮件客户端或发送代理。尽管很多 MTA 能够加密彼此之间的通信，但大多数 MTA 都不加密，因此通过任何公共网络发送电子邮件被视为本质上不安全的通信形式。Postfix 在 Red Hat Enterprise Linux 7 中将 Sendmail 替换为默认 MTA。

建议计划实施 Postfix 服务器的任何人都解决以下问题。

4.3.10.1. 限制服务攻击(Denial of Service Attack)

由于电子邮件的性质，确定攻击者可以很轻松地使用邮件填充服务器，并导致拒绝服务。通过设置 `/etc/postfix/main.cf` 文件中的指令限制，可以限制此类攻击的有效性。您可以更改已存在的指令的值，或者您可以使用以下格式添加您需要的值：

<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** - 允许任何客户端每个时间单元对此服务进行的最大连接数（如下所述）。默认值为 0，这意味着客户端可以与 Postfix 接受的每个时间单元进行任意数量的连接。默认情况下，可信网络中的客户端不会被排除。
- **avil_rate_time_unit** - 此时间单元用于速率限值计算。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不会被排除。
- **smtpd_client_message_rate_limit** - 每个时间单元允许客户端请求的最大消息数（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。此限制可针对 `master.cf` 文件中的特定服务覆盖。默认值为 100。
- **queue_minfree** - 接收邮件所需的队列文件系统中最少可用空间量（以字节为单位）。这目前由 Postfix SMTP 服务器用于决定是否接受任何邮件。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。要指定更高的最小可用空间限制，请指定一个 `queue_minfree` 值，它至少为 `message_size_limit` 的 1.5 倍。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。如果标头大于，则会丢弃多余的。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。默认情况下，值为 10240000。

4.3.10.2. NFS 和 Postfix

切勿将邮件池目录 `/var/spool/postfix/` 放在 NFS 共享卷上。由于 NFSv2 和 NFSv3 不保持对用户和组 ID 的控制，所以两个或多个用户可以具有相同的 UID，并且接收并读取彼此的邮件。



注意

在使用 Kerberos 的 NFSv4 中，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 共享卷中。

4.3.10.3. 仅邮件用户

为了帮助防止 Postfix 服务器上利用本地用户，邮件用户最好仅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，`/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 网络监听

默认情况下，Postfix 设置为仅侦听本地环回地址。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`，以确保只出现以下 `inet_interfaces` 行：

```
inet_interfaces = localhost
```

这样可确保 Postfix 仅接受来自本地系统的邮件（如 cron 作业报告），而不接受来自网络的邮件。这是默认设置，可保护 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 实施进行 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。这部分论述了如何将 Postfix 配置为使用 Dovecot SASL 实施。

要安装 Dovecot POP/IMAP 服务器并在您的系统中提供 Dovecot SASL 实施，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 实施通信。仅当 Postfix 和 Dovecot 应用在单独的计算机上运行时，才需要采用后一种方法。本指南为 UNIX-domain 套接字方法提供了偏好，这种方法提供了更好的隐私性。

要指示 Postfix 使用 Dovecot SASL 实施，需要对这两个应用进行一些配置更改。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（默认配置文件已包含大多数相关部分，只需要取消注释行）：

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

上例假定使用 UNIX-domain sockets 在 Postfix 和 Dovecot 之间进行通信。它还假设 Postfix SMTP 服务器的默认设置，其中包括位于 `/var/spool/postfix/` 目录下的邮件队列，以及在 postfix 用户和组下运行的应用。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 通过 TCP 侦听 Postfix 身份验证请求：

```
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 服务器提供纯和登录身份验证机制：

```
auth_mechanisms = plain login
```

设置 Postfix

对于 Postfix，只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 实施用于 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，使用相对路径可确保无论 Postfix 服务器是否在 `chroot` 中运行）：

```
smtpd_sasl_path = private/auth
```

此步骤假定您要使用 **UNIX-domain sockets** 进行 Postfix 和 Dovecot 之间的通信。要将 Postfix 配置为在不同机器上查找 Dovecot，以防您使用 TCP 套接字进行通信，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 计算机的 IP 地址，并将 **12345** 替换为 Dovecot 的 `/etc/dovecot/conf.d/10-master.conf` 配置文件中指定的端口。

4.

指定 Postfix SMTP 服务器提供给客户端的 SASL 机制。请注意，可以为加密和未加密会话指定不同的机制。

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

上例指定，在未加密会话期间，不允许匿名身份验证，不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。

有关限制允许的 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有助于通过 SASL 配置 Postfix SMTP 身份验证的其他信息。

- <http://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL> - 包含如何设置 Postfix 以使用 Dovecot SASL 实现 SMTP 身份验证的信息。
- http://www.postfix.org/SASL_README.html#server_sasl - 包含如何设置 Postfix 以使用 Dovecot 或 Cyrus SASL 实现 SMTP 身份验证的信息。

4.3.11. 保护 SSH

Secure Shell (SSH) 是一个功能强大的网络协议，用于通过安全通道与其他系统通信。通过 SSH 传输是加密的，并防止拦截。有关 SSH 协议以及 Red Hat Enterprise Linux 7 中使用 SSH 服务的一般信息，请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的 [OpenSSH](#) 章节。



重要

本节旨在介绍保护 SSH 设置的最常用方法。此推荐措施列表不得被视为详尽或确定性。有关基本 SSH 概念的说明，请参阅 `sshd_config(5)`，了解可用于修改 `sshd` 守护进程行为的所有配置指令的说明。

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。如果将此方法与其他身份验证方法相结合，可以将其视为多因素身份验证。有关使用多种验证方法的详情，请查看 [第 4.3.11.2 节“多种身份验证方法”](#)。

要启用加密密钥以进行身份验证，`/etc/ssh/sshd_config` 文件中的 `PubkeyAuthentication` 配置指令需要设置为 `yes`。请注意，这是默认设置。将 `PasswordAuthentication` 指令设置为 `no`，以禁用使用密码登录的可能性。

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` 参数来修改密钥的位级。使用 2048 位密钥通常已足够。《红帽企业 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。

您应在 `~/.ssh/` 目录中看到两个密钥。如果您在运行 `ssh-keygen` 命令时接受默认值，则生成的文件分别命名为 `id_rsa` 和 `id_rsa.pub`，并且包含私钥和公钥。您应始终防止私钥被除文件的所有者以外的任何人读取。然而，公钥需要传输到您要登录的系统。您可以使用 `ssh-copy-id` 命令将密钥传输到服务器：

```
~]$ 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。执行此操作后，请确保从 `authorized_keys` 文件中删除所有未使用的密钥。

4.3.11.2. 多种身份验证方法

使用多种身份验证方法或多因素身份验证可提高防止未经授权访问的保护级别，因此在强化系统防止系统受到破坏时应考虑这样的保护级别。尝试登录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，才能被授予访问权限。

使用 `/etc/ssh/sshd_config` 文件中的 `AuthenticationMethods` 配置指令来指定要使用的身份验证方法。请注意，可以使用这个指令定义多个所需验证方法的列表。如果情况如此，用户必须至少在其中一个列表中完成每个方法。列表需要用空白空格分隔，列表中的单个验证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` 守护进程只有在试图成功登录的用户完成公钥身份验证后再通过 `gssapi-with-mic` 或键盘交互身份验证授予访问权限。请注意，需要使用 `/etc/ssh/sshd_config` 文件中的对应配置指令（如 `PubkeyAuthentication`）来显式启用每个请求的身份验证方法。有关可用身份验证方法的一般列表，请参见 `ssh(1)` 的 `AUTHENTICATION` 部分。

4.3.11.3. 保护 SSH 的其他方式

协议版本

尽管红帽企业 Linux 7 提供的 SSH 协议的实施仍支持 SSH 客户端的 SSH-1 和 SSH-2 协议版本，但应尽可能使用后者。与较旧的 SSH-1 相比，SSH-2 版本包含了许多改进，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 来最大程度利用 SSH 协议保护其使用的身份验证和通信的范围。sshd 守护进程支持的协议的版本或版本可以使用 `/etc/ssh/sshd_config` 文件中的协议配置指令指定。默认设置为 2。请注意，SSH-2 版本是唯一被 Red Hat Enterprise Linux 7 SSH 服务器支持的版本。

密钥类型

`ssh-keygen` 命令默认生成一对 SSH-2 RSA 密钥，但使用 `-t` 选项时也可指示它生成 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，从而通过模糊性提高安全性。可以使用 `/etc/ssh/sshd_config` 配置文件中的 `Port` 指令来指定端口。另请注意，必须更改默认 SELinux 策略以允许使用非默认端口。您可以通过以 root 用户身份输入以下命令来修改 `ssh_port_t` SELinux 类型：

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

在以上命令中，将 `port_number` 替换为使用 `Port` 指令指定的新端口号。

没有根登录

如果您的特定用例不需要以 root 用户身份登录，则应考虑在 `/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，管理员可以审核哪个用户在以普通用户身份登录后运行什么特权命令，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 转发的不受信任的 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`，并且 `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不要使用 X11 转发。

4.3.12. Securing PostgreSQL

PostgreSQL 是对象存储的数据库管理系统(DBMS)。在红帽企业 Linux 7 中, **postgresql-server** 软件包提供 **PostgreSQL**。如果没有安装, 以 **root** 用户身份输入以下命令安装它:

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

您必须先初始化磁盘上的数据库存储区域, 然后才能开始使用 **PostgreSQL**。这称为数据库群集。若要初始化数据库集群, 可使用随 **PostgreSQL** 安装的 **initdb** 命令。数据库集群所需的文件系统位置通过 **-D** 选项来指示。例如:

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

initdb 命令将尝试创建您指定的目录 (如果该目录尚不存在)。本例中使用名称 **/home/postgresql/db1**。**/home/postgresql/db1** 目录包含数据库中存储的所有数据和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本地用户使用其用户名访问任何数据库:

```
local all          all          trust
```

当您使用创建数据库用户而不是本地用户的分层应用时, 这可能会造成问题。如果您不想显式控制系统上的所有用户名, 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 容器内自动化部署应用，并提供将应用与其运行时依赖项打包到容器中的功能。为了提高 Docker 工作流的安全性，请按照 [Red Hat Enterprise Linux Atomic Host 7 容器安全指南中的步骤](#)。

4.3.14. 针对 DDoS Attacks 保护 memcached

Memcached 是一个开源、高性能、分布式内存对象缓存系统。虽然它本质上是通用的，但它主要用于通过降低数据库负载来提高动态 Web 应用的性能。

Memcached 是来自数据库调用、API 调用或页面渲染的结果的小块任意数据的内存中键值存储。Memcached 允许应用程序从系统部分获取内存，因为系统的资源量大于所需的部分，并且能够被应用程序少于所需区域访问。

Memcached Vulnerabilities

2018 年，通过利用暴露于公共互联网的 Memcached 服务器发现 DDoS 放大攻击的漏洞。这些攻击利用了使用 UDP 协议进行传输的 Memcached 通信。该攻击非常有效，因为扩展率很高 - 大小为几百字节请求可产生几兆字节甚至数百兆字节的响应。此问题已被记录为 [CVE-2018-1000115](#)。

在大多数情况下，memcached 服务不需要公开给公共互联网。这种暴露可能有其自身的安全问题，允许远程攻击者泄漏或修改存储在 Memcached 中的信息。

强化 memcached

要缓解安全风险，请从以下步骤中执行适用于您的配置：

-

在 LAN 中配置防火墙。如果您的 Memcached 服务器应该只能从本地网络内部访问，则不允许外部流量到 memcached 使用的端口。例如，从允许的端口列表中删除 memcached 默认使用的端口 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，请查看 [第 5.8 节“使用区管理传入的流量依赖源”](#)。

- 通过在 `/etc/sysconfig/memcached` 文件中的 `OPTIONS` 变量中添加 `-U 0 -p 11211` 值来禁用 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，请将 memcached 设置为仅侦听 `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"
```
- 如果可以更改身份验证，请启用 SASL（简单身份验证和安全层）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. 通过在 `/etc/sysconfig/memcached` 中添加 `-S` 值到 `OPTIONS` 来启用 memcached 中的 SASL 支持：


```
■
```


OPTIONS="-S"

5. *重新启动 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的 memcached 客户端配置中。*

- *使用 stunnel 加密 memcached 客户端和服务端之间的通信。由于 memcached 不支持 TLS，因此一个临时解决方案是使用一个代理，如 stunnel，它在 memcached 协议之上提供 TLS。*

您可以将 stunnel 配置为使用 PSK（共享密钥），甚至更好的使用用户证书。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，并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，或者防火墙阻止从网络访问 memcached 端口。

如需更多信息，请参阅 [第 4.8 节“使用 stunnel”](#)。

4.4. 保护网络访问

4.4.1. 使用 TCP wrapper 和 xinetd 保护服务

TCP wrapper 的能力远远不止于拒绝对服务的访问。本节介绍如何使用它们发送连接横幅、来自特定主机的攻击警告，以及增强日志记录功能。有关 TCP Wrapper 功能和控制语言的详情，请查看 `hosts_options(5)` man page。可用的标记请查看 `xinetd.conf(5)` man page，它作为您可以应用到服务的选项。

4.4.1.1. TCP 封装器和连接程序

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被滥用的好方法。您还可以控制向用户显示有关系统的信息。要为服务实施 TCP wrappers 横幅，请使用 `banner` 选项。

这个示例为 `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，但名称必须与守护进程相同。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，或者用户名和 IP 地址，以使连接更加令人生畏。

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

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

4.4.1.2. TCP wrapper 和 Attack Warning

如果检测到特定主机或网络攻击服务器，则可以使用 TCP wrapper 警告管理员使用 `generate` 指令发送该主机或网络的后续攻击。

在本例中，假设来自 `206.182.68.0/24` 网络的攻击者已被检测到试图攻击服务器。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。

要允许连接并进行记录，请将 `generate` 指令放在 `/etc/hosts.allow` 文件中。



注意

由于生成指令执行任何 `shell` 命令，因此最好创建一个特殊脚本来通知管理员，或者在特定客户端尝试连接服务器时执行命令链。

4.4.1.3. TCP wrappers 和增强的日志记录

如果某些类型的连接比其他类型的关注更大，那么可以使用 **severity** 选项提升该服务的日志级别。

在本例中，假设尝试连接到 FTP 服务器上的端口 23（Telnet 端口）都是攻击者。若要表示这一点，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
```

这将使用默认的 **authpriv** 日志记录功能，但将优先级从 **info** 的默认值提升为 **emerg**，后者将日志消息直接发送到控制台。

4.4.2. 验证正在侦听哪些端口

关闭未使用的端口非常重要，以避免可能受到的攻击。对于处于侦听状态的意外端口，您应该调查可能的入侵信号。

使用 **netstat** 进行开放端口扫描

以 **root** 用户身份输入以下命令，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 0.0.0.0:2500     0.0.0.0:*       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，也可使用 **ss** 实用程序列出处于侦听状态的打开端口。它可显示比 **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 状态显示处于 **UDP** 侦听模式的端口。

对来自外部系统的 **ss** 输出中显示的每个 IP 地址（本地主机 127.0.0.0 或 ::1 范围除外）进行扫描。使用 **-6** 选项扫描 IPv6 地址。

然后，使用通过网络连接到第一系统的另外一个远程计算机的 **nmap** 工具进行外部检查。这可用于验证 **firewalld** 中的规则。以下是确定哪个端口正在侦听 TCP 连接的示例：

```
~j# 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 (**-s S**) 不是选项时，TCP 连接扫描 (**-sT**) 是默认的 TCP 扫描类型。**O** 选项可检测主机的操作系统。

使用 netstat 和 ss 为 Open SCTP 端口扫描

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：

```
~j# 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
```

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

ss 工具也可以显示 SCTP 打开的端口：

```
~j# 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，向路由器告知数据包必须采用的路径。还有一个在路由被遍历时记录跃点的选项。“路由记录”获取的跃点列表为目的地提供源的返回路径。这允许源（发送主机）以松散或严格方式指定路由，忽略部分或所有路由器的路由表。它允许用户重定向网络流量以满足恶意用途。因此，应禁用基于源的路由。

accept_source_route 选项可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。源路由数据包的接受由 **sysctl** 设置控制。以 **root** 用户身份运行以下命令，使用 **SSR** 或 **LSR** 选项集丢弃数据包：

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

如果可能，禁用数据包转发也应与上述操作结合使用（禁用转发可能会干扰虚拟化）。以 **root** 身份发出以下列出的命令：

这些命令禁用所有接口上转发 **IPv4** 和 **IPv6** 数据包：

```
~]# /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转发所有多播数据包：

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

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

接受 **ICMP** 重定向的合法用途很少。禁用接受和发送 **ICMP** 重定向的数据包，除非特别需要。

这些命令禁止在所有接口上接受所有 **ICMP** 重定向的数据包：

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

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

这个命令禁止在所有接口上接受安全 ICMP 重定向的数据包：

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

这个命令禁止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.interface.send_redirects` 选项设置为已启用，则 ICMP 重定向的发送保持活动。确保将每个接口的 `net.ipv4.conf.interface.send_redirects` 选项设置为 0 值。要在每次添加新接口时自动禁用 ICMP 请求的发送，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 IPv4 重定向数据包的发送。有关“IPv6 节点要求 <http://www.ietf.org/rfc/rfc4294.txt> 的说明，请参阅 RFC4294”，这导致 IPv4 和 IPv6 之间的区别。

注意

要使这些设置在重新引导后保留，请修改 `/etc/sysctl.conf` 文件。例如，要禁止接受所有接口上的所有 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，并添加一行，如下所示：

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

如需更多信息，请参阅 `sysctl` man page (`sysctl(8)`)。有关基于源的路由及其变体的 Internet 选项的说明，请参阅 RFC791。

**警告**

以太网网络提供了重定向流量的其他方法，如 ARP 或 MAC 地址欺骗、未授权 DHCP 服务器和 IPv6 路由器或邻居公告。此外，单播流量偶尔会广播，从而导致信息泄漏。这些弱点只能通过网络运营商实施的具体措施来解决。基于主机的策略并不完全有效。

4.4.3.1. 反向路径转发

反向路径转发用于防止通过一个接口到达的数据包通过不同的接口离开。当传出路由和传入路由不同时，它有时被称为非对称路由。路由器通常以这种方式路由数据包，但大多数主机不需要这样做。例外是涉及通过一个链路发送流量以及通过来自其他服务提供商的另一个链接接收流量的应用。例如，将租用的行与 xDSL 或卫星链接与 3G modems 结合使用。如果您的情况适用于您，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，否则最好启用，因为它可防止用户从本地子网欺骗 IP 地址并减少 DDoS 攻击的机会。

**注意**

根据 RFC 3704(Ingress Filtering for Multihomed Network)的建议，Red Hat Enterprise Linux 7 默认使用 Strict Reverse Path 转发。

**警告**

如果启用了转发，则只有在源地址验证有其他方法（如 iptables 规则）时才应禁用 Reverse Path 转发。

rp_filter

使用 `rp_filter` 指令启用反向路径转发。`sysctl` 实用程序可用于更改正在运行的系统，并通过向 `/etc/sysctl.conf` 文件添加行来进行永久性更改。`Therp_filter` 选项用于指示内核从三种模式之一进行选择。

要进行临时全局更改，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
```

其中整数是以下之一：

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

可以使用 `net.ipv4.conf.interface.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

要使这些设置在重新引导后保留，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，并添加一行，如下所示：

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

IPv6_rpfilter

如果使用 IPv6 协议，`firewalld` 守护进程默认应用到反向路径转发。可以在 `/etc/firewalld/firewalld.conf` 文件中检查该设置。您可以通过设置 `IPv6_rpfilter` 选项来更改 `firewalld` 的行为。

如果您需要自定义配置的反向路径转发，您可以使用 `ip6tables` 命令在没有 `firewalld` 守护进程的情况下执行该配置，如下所示：

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

This 规则应该插入到 `raw/PREROUTING` 链开头附近，以便它应用到所有流量，特别是在有状态匹配规则的前面。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

要启用从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 用户身份登录，并将 `/etc/sysctl.conf` 文件中的 `net.ipv4.ip_forward = 0` 的行更改为以下内容：

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

要加载 `/etc/sysctl.conf` 文件中的更改，请输入以下命令：

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

要检查是否启用了 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它：

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

4.4.3.2. 其它资源

以下是解释更多关于反向路径转发的资源。

-

安装的文档

`/usr/share/doc/kernel-doc-版本/Documentation/networking/ip-sysctl.txt` - 此文件包含目录中可用的文件和选项的完整列表。在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

有关多主目录的 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)，使 DNS 客户端能够验证和检查来自 DNS 名称服务器的响应的完整性，以便验证其源并确认它们是否在传输中被篡改。

4.5.2. 了解 DNSSEC

为了通过互联网连接，越来越多的网站现在能够使用 HTTPS 进行安全连接。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，并会因为缺少身份验证而受到中间人攻击。换句话说，DNS 客户端无法确信，显示来自给定 DNS 名称服务器的回复是身份验证的，并且未被篡改。更重要的是，递归名称服务器无法确保从其他名称服务器获得的记录是真正的。DNS 协议没有为客户端提供一种机制，以确保其不会受到中间人攻击。引入了 DNSSEC，以解决在使用 DNS 解析域名时缺少身份验证和完整性检查的问题。它没有解决机密性问题。

发布 DNSSEC 信息涉及数字签名 DNS 资源记录以及发布公钥，从而使 DNS 解析器能够建立分层的信任链。所有 DNS 资源记录的数字签名将作为数字签名资源记录(RRSIG)生成并添加到区域中。区域的公钥添加为 DNSKEY 资源记录。为构建层次结构链，DNSKEY 的哈希发布在父区域中，作为签名(DS)资源记录委派。为了便于验证不存在，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，用于委派到子区域(NS 和粘滞记录)的记录并未签名；这些记录显示在子区域中，并在其中签名。

处理 DNSSEC 信息由配置了 root 区域公钥的解析器完成。使用此密钥，解析器可以验证 root 区域中使用的签名。例如，root 区域已为 DS 记录进行了签名：root 区域还提供 .com 名称服务器的 NS 和粘滞记录。解析器遵循这一委派，并使用这些委派的名服务器查询的 DNSKEY 记录。获取的 DNSKEY 记录的哈希应与 root 区域中的 DS 记录匹配。如果是这样，解析器将信任获得的 DNSKEY for .com。在 .com 区域中，RRSIG 记录由 .com DNSKEY 创建。此过程在 .com 内部类似地重复执行，如 redhat.com。使用此方法，验证 DNS 解析器只需要配置一个根密钥，而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 返回至应用程序。

DNSSEC 的设计方式使不支持 DNSSEC 的应用程序完全不可见。如果非 DNSSEC 应用查询 DNSSEC 解析器，它将接收答案，而无需任何这些新资源记录类型，如 RRSIG。但是，DNSSEC 解析器仍然会执行所有加密检查，如果它检测到恶意 DNS 答案，仍会向应用程序返回 SERVFAIL 错误。DNSSEC 可保护 DNS 服务器之间数据的完整性(权威和递归)，它不提供应用与解析器之间的安全性。因此，向应用程序赋予其解析器的安全传输非常重要。最简单的方法是在 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 Hots 或 VPN 依赖于“DNS：设备门户倾向于劫持”DNS，以便将用户重定向到页面，要求他们为其 Wi-Fi 服务进行身份验证（或支付费用）。用户连接到 VPN “时通常只需要使用内部”DNS 服务器，才能查找在公司网络之外不存在的资源。这需要通过软件进行额外的处理。例如，`dnssec-trigger` 可用于检测 Hotspot 是否在劫持 DNS 查询并且 `unbound` 可以充当代理服务器来处理 DNSSEC 查询。

选择 DNSSEC Capable Recursive Resolver

若要部署支持 DNSSEC 的递归解析器，可使用 BIND 或 `unbound`。默认情况下启用 DNSSEC，并使用 DNSSEC 根密钥配置。要在服务器上启用 DNSSEC，但是在移动设备（如笔记本电脑）上使用 `unbound` 是首选的，因为它允许本地用户在使用 `dnssec-trigger` 时动态重新配置 Hotspots 所需的 DNSSEC 覆盖，在使用 Libreswan 时可以动态重新配置 Hotspots 所需的 DNSSEC 覆盖。`unbound` 守护进程进一步支持部署 `etc/unbound/*.d/` 目录中列出的 DNSSEC 异常，这对于服务器和移动设备都非常有用。

4.5.3. 了解 Dnssec-trigger

在 `/etc/resolv.conf` 中安装和配置了 `unbound` 后，来自应用程序的所有 DNS 查询将由 `unbound` 来处理。`DNSSEC-trigger` 仅在触发时重新配置 `unbound` 解析器。这主要适用于连接到不同 Wi-Fi 网络的客户端机器（如笔记本电脑）。流程如下：

- 当通过 DHCP 获取新 DNS 服务器时，NetworkManager “会触发” `dnssec-trigger`。
- 然后，`DNSSEC-trigger` 对服务器执行一系列测试，并决定是否正确支持 DNSSEC。
- 如果存在，`dnssec-trigger` 会重新配置 `unbound` 以将该 DNS 服务器用作所有查询的转发器。
- 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。
- 如果它确定有无限制端口 53（UDP 和 TCP），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。
- 如果无法做到这一点，例如因为除到达网络 DNS 服务器本身之外，防火墙会阻止端口 53，它将尝试使用 DNS 端口 80，或者 TLS 封装 DNS 到端口 443。可以在 `/etc/dnssec-trigger/dnssec-trigger.conf` 中配置在端口 80 和 443 上运行 DNS 的服务器。已注释掉的示例应在默认配置文件中提供。
- 如果这些回退方法也失败，`dnssec-trigger` 将提供非安全操作（完全绕过“DNSSEC”，或者仅以缓存模式运行”，它将不尝试新的 DNS 查询，而是应答其在缓存中已有的所有内容。

Wi-Fi Hotspots 在授予访问互联网之前，将用户重定向到登录页面。在上面概述的探测序列中，如果检测到重定向，系统将提示用户询问是否需要登录才能访问 Internet。dnsssec-trigger 守护进程每十秒继续探测 DNSSEC 解析器。有关使用 dnsssec-trigger 图形工具的详情，请查看 [第 4.5.8 节“使用 Dnssec-trigger”](#)。

4.5.4. VPN 提供的域和名称服务器

某些类型的 VPN 连接可传达一个域以及用作 VPN 隧道设置一部分的名称服务器列表。在 Red Hat Enterprise Linux 中，NetworkManager 支持这个功能。这意味着 unbound、dnsssec-trigger 和 NetworkManager 的组合可以正确地支持 VPN 软件提供的域和名称服务器。当 VPN 隧道启动后，系统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域名查询。当 VPN 隧道终止时，unbound 缓存会再次刷新，以确保对域的任何查询都会返回公共 IP 地址，而不是之前获得的专用 IP 地址。请参阅 [第 4.5.11 节“为连接提供域配置 DNSSEC 验证”](#)。

4.5.5. 推荐的命名实践

红帽建议静态和临时名称与用于 DNS 中机器的完全限定域名(FQDN)匹配，如 host.example.com。

互联网编号分配公司(ICANN)有时会在公共寄存器中添加以前未注册的顶级域(如.yourcompanyany)。因此，红帽强烈建议您不要使用没有委托给您的域名，即使使用私有网络中也是如此，因为这可能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。使用未委派的域名也会增加 DNSSEC 部署和维护的难度，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，请参阅域名冲突上的 ICANN 常见问题解答。](#)

4.5.6. 了解信任 Anchors

在分层加密系统中，信任定位符是一个权威实体，假定该实体值得信赖。例如，在 X.509 架构中，根证书是一个信任定位点，从中衍生了一串信任链。必须先将信任定位点放在信任方的假定位置上，然后才能进行路径验证。

在 DNSSEC 的上下文中，信任定位符由与该名称关联的 DNS 名称和公钥（或公钥哈希）组成。它表示为基础 64 编码密钥。它类似于证书，因为它是一种交换信息（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 或 DNSKEY RR 的配置的 DNSKEY RR 哈希。验证安全感知型解析器使用此公钥或哈希作为将身份验证链构建为签名 DNS 响应的起始点。通常，验证解析器必须通过 DNS 协议之外的一些安全或可信方法获得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期望信任定位点所指向的区域。

4.5.7. 安装 DNSSEC

4.5.7.1. 安装 unbound

若要在本地使用 DNSSEC 在计算机上验证 DNS，需要安装 DNS 解析器 unbound（或绑定）。仅需要在移动设备上安装 `dnssec-trigger`。对于服务器，unbound 应该足够，但可能需要本地域的转发配置，具体取决于服务器所处的位置（LAN 或 Internet）。DNSSEC-trigger 目前将仅对全局公共 DNS 区域提供帮助。NetworkManager、dhclient 和 VPN 应用程序通常可自动收集域列表（以及名称服务器列表），但不能自动收集 `dnssec-trigger` 或 unbound。

要安装 unbound，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` 命令将报告 unbound 为 `Active: inactive (dead)`。

4.5.7.3. 启动 unbound

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

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

运行 `systemctl enable` 命令，以确保 unbound 在每次系统引导时启动：

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

unbound 守护进程允许配置本地数据或使用以下目录覆盖：

- **/etc/unbound/conf.d** 目录用于添加特定域名的配置。这用于将域名查询重定向到特定的 DNS 服务器。这通常用于仅在公司 WAN 中存在的子域。
- **/etc/unbound/keys.d** 目录用于为特定域名添加信任定位符。当仅内部名称经过 DNSSEC 签名时，这是必需的，但没有公开存在的 DS 记录来构建信任路径。另一个用例是，域的内部版本使用与公司 WAN 外公开名称不同的 DNSKEY 签名。
- **/etc/unbound/local.d** 目录用于将特定的 DNS 数据添加为本地覆盖。这可用于构建黑名单或创建手动覆盖。此数据将通过 unbound 返回给客户端，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。有关详细信息，请参阅 **unbound.conf(5)** man page。

4.5.7.4. 安装 Dnssec-trigger

dnssec-trigger 应用作为守护进程 **dnssec-triggerd** 运行。要安装 **dnssec-trigger**，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 `dnsssec-triggerd` 守护进程没有在运行，`systemctl status` 命令将报告 `dnsssec-triggerd` 为 **Active: inactive (dead)**。要为当前会话启动它，以 `root` 用户身份输入以下命令：

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

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

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

4.5.8. 使用 Dnssec-trigger

`dnsssec-trigger` 应用具有 GNOME 面板实用程序，可用于显示 DNSSEC 探测结果并按需执行 DNSSEC 探测请求。要启动实用程序，请按 **Super** 键进入“活动概览”，键入 **DNSSEC**，然后按 **Enter** 键。在屏幕底部的消息托盘中添加了重装货架定位符的图标。按屏幕右下角的循环蓝色通知图标来显示它。右键单击定位图标以显示弹出式菜单。

在正常操作中，`unbound` 在本地用作名称服务器，`resolv.conf` 则指向 `127.0.0.1`。当您在 **Hotspot Sign-On** 面板上单击“确定”时，这一点已更改。DNS 服务器从 `NetworkManager` 查询并放入 `resolv.conf`。现在您可以在 **Hotspot** 的登录页面上进行身份验证。定位图标显示一个大红色感叹号，用于提醒您 DNS 查询不安全。身份验证后，`dnsssec-trigger` 应该自动检测到这一点并切换回安全模式，尽管在某些情况下，它无法，用户必须通过选择 **Reprobe** 手动执行此操作。

DNSSEC-trigger 通常不需要任何用户交互。启动后，它会在后台工作，如果遇到问题，它会通过弹出文本框通知用户。它还会告知 `unbound` 对 `resolv.conf` 文件的更改。

4.5.9. 在 DNSSEC 中使用 dig

要查看 **DNSSEC** 是否工作，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。其他有用的工具会从 `ldns` 软件包和 `unbound` 软件包中分离。旧 DNS 实用程序 `nslookup` 和 `host` 已过时，不应使用。

要使用 `dig` 发送请求 **DNSSEC** 数据的查询，可将选项 `+dnssec` 添加到该命令中，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
HH0377I0Lsybj/uZv5hL4UwWd/lw6Gn8GPikqhztAkgMxddMQ2IARP6p
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，以及签名的最初时间和过期时间。unbound 服务器表示数据是 DNSSEC 身份验证的，方法是返回顶部的 flags: 部分中的 ad 位。

如果 DNSSEC 验证失败，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，可以通过在 dig 命令中指定 +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.
E572dLKMvYB4cgTRyAHIKKEvdOP7tockQb7hXFNZKVbfXbZJOIDREJrr
zCgAfJ2hykfY0yJHAInuQvM0s6xOnNBSvc2xLlybJdfTaN6kSR0YFdYZ
n2NpPctn2kUBn5UR1BJRin3Gqy20LZIZx2KD7cZBtieMsU/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 本身会因为错误的出现或过期时间而证明自己，但在这个示例中，www.dnssec-tools.org 的人员已有意操作了这个 RRSIG 签名，我们无法通过手动查看这个输出来检测到这一点。这个错误会显示在 `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

连接到网络时，`dnssec-trigger` 会尝试检测 Hotspot。Hotspot 通常是强制用户与网页交互的设备，然后才能使用网络资源。检测通过尝试下载包含已知内容的特定固定网页来完成。如果有 Hotspot，则收到的内容不会如预期那样。

要设置一个包含已知内容的固定网页，`dnssec-trigger` 可以用来检测 Hotspot，请按以下操作：

1. 在可通过互联网公开访问的一些计算机上设置 Web 服务器。请参阅《红帽企业 Linux 7 系统管理员指南》中的 [Web 服务器一章](#)。
2. 服务器运行之后，发布一个包含已知内容的静态页面。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 Web 服务器 `document_root/static/` 子目录中发布了 `hotspot.txt` 文件，则静态 Web 页面的地址将为 `example.com/static/hotspot.txt`。请参阅《Red Hat Enterprise Linux 7 系统管理员指南》中的 [Web 服务器章节中的 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"
```

这个命令添加一个使用 HTTP（端口 80）探测的 URL。第一部分是将要解析的 URL 和要下载的页面。命令的第二部分是下载的网页应包含的文本字符串。

有关配置选项的更多信息，请参阅 `man page dnssec-trigger.conf(8)`。

4.5.11. 为连接提供域配置 DNSSEC 验证

默认情况下，带有正确名称服务器的转发区域由任何连接提供的每个域的 `dnssec-trigger` 自动添加到 `unbound` 中，但通过 `NetworkManager` 的 Wi-Fi 连接除外。默认情况下，添加到 `unbound` 中的所有转发区域都经过 DNSSEC 验证。

可以更改验证转发区域的默认行为，以便默认情况下不对所有转发区域进行 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，按如下所示打开并编辑行：

```
validate_connection_provided_zones=no
```

不会对现有转发区进行更改，但只适用于未来的转发区。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 提供的域配置 DNSSEC 验证

可以启用为 Wi-Fi 提供的区域添加转发区。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，按如下所示打开并编辑行：

```
add_wifi_provided_zones=yes
```

不会对现有转发区进行更改，但只适用于未来的转发区。因此，如果您要为当前 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

开始添加 Wi-Fi 提供的域（如将区域转发到 `unbound`）可能会造成安全影响，例如：

1. **Wi-Fi 接入点可能会有意通过 DHCP 为您提供一个域，对其没有权威，并将您的所有 DNS 查询路由到其 DNS 服务器。**
2. **如果您已经关闭了转发区域的 DNSSEC 验证，则 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，而无需您知道它。**

4.5.12. 其它资源

以下是有关 DNSSEC 的更多解释的资源：

4.5.12.1. 安装的文档

- **DNSSEC-trigger(8)man page - 描述 `dnssec-triggerd`、`dnssec-trigger-control` 和 `dnssec-trigger-panel` 的命令选项。**
- **DNSSEC-trigger.conf(8)man page - 描述 `dnssec-triggerd` 的配置选项。**

- **unbound(8)man page** - 描述 unbound (DNS 验证解析器) 的命令选项。
- **unbound.conf(5) 手册页** - 包含如何配置 unbound 的信息。
- **resolv.conf(5) 手册页** - 包含解析器例程可读取的信息。

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/>

具有许多 DNSSEC 资源链接的网站。

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

DNSSEC 部署计划由国土安全部赞助，包含许多 DNSSEC 信息，并有一个邮件列表来讨论 DNSSEC 部署问题。

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

互联网的“Deploy 360 计划推动和协调”DNSSEC 部署，是查找全球社区和 DNSSEC 活动的好资源。

<http://www.unbound.net/>

本文档包含有关 unbound DNS 服务的一般信息。

<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 协议来配置虚拟专用网络(VPN)。Libreswan 是 Openswan 应用的延续，Openswan 文档中的许多示例可与 Libreswan 互换。NetworkManager IPsec 插件称为 NetworkManager-libreswan。GNOME Shell 用户应安装 NetworkManager-libreswan-gnome 软件包，该软件包的依赖项为 NetworkManager-libreswan。请注意，NetworkManager-libreswan-gnome 软件包只在可选频道中可用。[请参阅启用补充和可选存储库。](#)

VPN 的 IPsec 协议本身使用 Internet 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。使用级别 2 隧道协议 (L2TP)的 IPsec VPN 变体通常称为 L2TP/IPsec VPN，这需要可选通道 xl2tpd 应用程序。

Libreswan 是在红帽企业 Linux 7 中提供的开源用户空间IKE 实施。IKE 版本 1 和 2 作为用户级后台程序实施。IKE 协议本身也加密。IPsec 协议由 Linux 内核实施，Libreswan 配置内核以添加和删除 VPN 隧道配置。

TheIKE 协议使用 UDP 端口 500 和 4500。IPsec 协议由两个不同的协议组成：封装的安全负载(ESP)，其协议号为 50，经过身份验证的标头(AH)是协议号 51。不建议使用 AH 协议。建议 AH 用户迁移到使用 null 加密的ESP。

IPsec 协议具有两种不同的操作模式：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 命令配置手动密钥，但为了安全起见，强烈建议您这样做。Libreswan 使用 netlink 与 Linux 内核接口。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 均经过认证，可与联邦信息处理标准 (FIPS)出版物 140-2 一起使用。



重要

由 Libreswan 和 Linux 内核实施的 IKE/IPsec VPN 是 Red Hat Enterprise Linux 7 中唯一推荐使用的 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，以 root 用户身份输入以下命令：

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

检查是否安装了 **Libreswan** :

```
~j$ yum info libreswan
```

在新安装 **Libreswan** 后, 应在安装过程中初始化 **NSS** 数据库。在启动新数据库前, 请按如下所示删除旧数据库 :

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

然后, 要初始化一个新的 **NSS** 数据库, 以 **root** 用户身份输入以下命令 :

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

只有在 **FIPS** 模式下操作时, 才需要使用密码保护 **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 :

```
~j# 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** 用户身份运行以下命令 :

```
~j# systemctl start ipsec
```

确认守护进程现在正在运行 :

```
~j$ 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 --stopnflag (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 --checknflag (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
```

配置所有中间和基于主机的防火墙以允许 **ipsec** 服务。有关防火墙和允许特定服务通过的信息，请查看 [第 5 章 使用防火墙](#)。**libreswan** 需要防火墙以允许以下数据包：

- **UDP 端口 500 和 4500 用于 Internet 密钥交换 (IKE) 协议**
- **封装安全负载 (ESP) IPsec 数据包的协议 50**
- **用于经过身份验证的标头 (AH) IPsec 数据包 (uncommon) 的 51 协议。**

我们给出了使用 **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，以便它们能够安全地进行通信。第二个示例是将两个站点连接在一起，组成一个网络。第三个示例是支持远程用户，称为这种情况下的路战。

4.6.2. 使用 **Libreswan** 创建 VPN 配置

因为 **IKE/IPsec** 对等协议，因此 **Libreswan** 不使用术语“**source**”“和目的地或”服务器及“客户端”。相反，它使用左和“右侧术语来指代端点”（主机）。“在大多数情况下，这也允许在两种端点上使用相同的配置，尽管许多管理员选择始终将左用于本地主机”，而右侧是远程主机”。

端点验证有四种常用方法：

- **预共享密钥 (PSK) 是最简单的身份验证方法。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。主机使用彼此的公共 RSA 密钥进行手动配置。当 dozens 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通用 IPsec 网关。中央证书颁发机构(CA)用于为主机或用户签署 RSA 证书。此中央 CA 负责转发信任关系，包括单个主机或用户的吊销。
- **NULL 身份验证**用于在没有身份验证的情况下获得网络加密。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，其中客户端对服务器进行身份验证，但服务器不验证客户端。此模型与使用 TLS 的安全网站类似（也称为 https:// 网站）。

除了这些验证方法外，还可以添加额外的身份验证来防止量子计算机可能受到的攻击。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 PreShared 密钥对应的 PPKID）来使用自己的 PPK。请参阅 [第 4.6.9 节“使用对 Quantum Computers 的保护”](#)。

4.6.3. 使用 Libreswan 创建主机至主机 VPN

要将 Libreswan “配置为在两个主机（称为左和“右”）之间创建一个主机到主机的 IPsec “VPN，请在两个主机（左和“右”）上以 root 用户身份输入以下命令以创建新的原始 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，特别是具有低熵的虚拟机上。

“要查看主机公钥，使其可在配置中指定为左侧”，请使用“newhostkey 命令返回的 CKAID 在添加新 hostkey 的主机中以 root 用户身份运行以下命令：

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

leftsasigkey=0sAQPfKElpV2GdCF0Ux9Kqhcap53Kaa+uCgduoT2l3x6LkRK8N+GiVGkRH4Xg+WMrz
Rb94kDDD8m/BO/Md+A30u0NjDk724jWuUU215rnpwvbdAob8pxYc4ReSgjQ/DkqQvsemoeF4kimMU1
```

```
OBPNu7IBw4hTBFzu+iVUYMELwQSXpremLXHBNlamUbe5R1+ibgxO19l/PAbZwxyGX/ueBMBvSQ+H
0UqdGKbq7UgSEQTFa4/gqdYZDDzx55tpZk2Z3es+EWdURwJOgGiiIFuBagasHFpeu9Teb1VzRyytny
NiJCBVhWVqsB4h6eaQ9RpAMmqBdBeNHfXwb6/hg+JIKJgjidXvGtgWBYNDpG40fEFh9USaFISdiHO+
dmGyZQ74Rg9sWLtiVdIH1YEBUtQb8f8FVry9wSn6AZqPlpGgUdtkTYUCaaifsYH4hoIA0nku4Fy/Ugej8
9ZdrSN7Lt+igns4FysMmBOI9Wi9+LWnfl+dm4Nc6UNgLE8kZc+8vMJGkLi4SYjk2/MFYggGX/COxSCPE
FUZFiNK7Wda0kWea/FqE1heem7rvKAPliqMymjSmytZI9hhkCD16pCdgrO3fJXsfAUChYYSPyPQCikav
vBL/wNK9zlaOwssTaKTj4Xn90SrZaxTEjppUeQ==
```

您将需要此密钥来添加到两台主机上的配置文件，如下方所述。如果您忘记了 **CKAID**，使用以下命令获取机器上所有主机密钥的列表：

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

密钥对的机密部分存储在位于 `/etc/ipsec.d/*.db` 的“NSS 数据库中”。

为了制作此主机到主机隧道的配置文件，上面的行 `leftrsasigkey=` 和 `rightrsasigkey=` 添加到位于 `/etc/ipsec.d/` 目录中的自定义配置文件。

使用以 **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+hOafUZDICQmXFrije/oZm [...] W2n417C/4urYHQkCvulQ==
  rightid=@east.example.com
  right=192.1.2.45
  rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
  authby=rsasig
  # load and initiate automatically
  auto=start
```

公钥也可通过其 **CKAID** 而非 **RSAID** 进行配置。在这种情况下，使用“`leftckaid=`”而不是“`leftrsasigkey=`”

您可以在左侧和右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为左或“右”。如果其中一个主机是移动主机，这意味着事先不知道 IP 地址，那么在移动客户端上，

移动客户端上使用 `%defaultroute` 作为其 IP 地址。这将自动获取动态 IP 地址。在接受传入移动主机连接的静态服务器主机上，指定将 `%any` 用作其 IP 地址的移动主机。

确保从左侧主机获取了“`lefttrsasigkey`”值，并且从右侧主机获取了 `righttrsasigkey` “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，以确认隧道建立：

```
~]# 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并启动隧道：

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

4.6.3.1. 使用 Libreswan 验证主机至主机 VPN

IKE 协商在 UDP 端口 500 和 4500 上进行。IPsec 数据包显示为封装的安全支付(ESP)数据包。The ESP 协议没有端口。当 VPN 连接需要通过 NAT 路由器时，ESP 数据包在端口 4500 上封装在 UDP 数据包中。

要验证数据包是否通过 VPN 隧道发送，以以下格式以 `root` 身份发出命令：

```
~]# 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 接口是已知传输流量的界面。要使用 `tcpdump` Ctrl+C。



注意

`tcpdump` 命令稍微意外地与 IPsec 交互。它仅看到传出加密数据包，而不看到传出的纯文本数据包。它确实看到加密的传入数据包，以及加密的传入数据包。若有可能，在两台计算机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。使用虚拟隧道接口(VTI)时，物理接口上的 `tcpdump` 显示 ESP 数据包，而 VTI 接口上的 `tcpdump` 则显示明文流量。

要检查隧道是否已成功建立，并查看流量通过隧道的程度，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 VPN

为了让 **Libreswan** 创建站点到站点 IPsec VPN，将两个网络接合在一起，在两个主机之间创建一个 IPsec 隧道，这些端点配置为允许来自一个或多个子网的流量通过。因此，它们可以视为到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 IPsec VPN，请首先配置主机到主机的 IPsec VPN，如第 4.6.3 节“使用 **Libreswan** 创建主机至主机 VPN”所述，然后使用适当名称将文件复制到一个文件中，如 `/etc/ipsec.d/my_site-to-site.conf`。使用以 `root` 用户身份运行的编辑器，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+hOafUZDICQmXFrje/oZm [...] W2n417C/4urYHQkCvulQ==
    rightid=@east.example.com
    right=192.1.2.45
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    authby=rsasig
```

要启动隧道，请重启 **Libreswan**，或者以 **root** 用户身份使用以下命令手动加载并启动所有连接：

```
~]# 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 验证站点到站点的 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 验证主机至主机 VPN”所述的步骤相同。

4.6.5. 使用 Libreswan 配置 Site-to-Site Single 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（而非公共 IP 地址）相互通信。这可以通过单一隧道来完成。如果左侧主机（主机名为 **west**）具有内部 IP 地址 **192.0.1.254**，右侧主机（主机名为 **east**）具有内部 IP 地址 **192.0.2.254**，请使用单一隧道存储两个服务器上的 **/etc/ipsec.d/myvpn.conf** 文件：

```
conn mysubnet
    leftid=@west.example.com
    lefttrsasigkey=0sAQOrlo+hOafUZDICQmXFrje/oZm [...] W2n417C/4urYHQkCvulQ==
    left=192.1.2.23
    leftsourceip=192.0.1.254
    leftsubnet=192.0.1.0/24
    rightid=@east.example.com
    righttrsasigkey=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 通常部署在中心中心架构中。每个 leaf 节点都有一个 IP 范围，它是较大范围的一部分。通过 hub 相互通信。这称为子网扩展。

例 4.2. 配置简单子网扩展设置

在以下示例中，我们将头办事处配置为 10.0.0.0/8 以及两个使用小/24 子网的分支。

在总部：

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

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

在“branch1”办事处，我们使用相同的连接。此外，我们使用直通连接来排除本地 LAN 流量通过隧道发送：

```

conn branch1
  left=1.2.3.4
  leftid=@headoffice
  leftsubnet=0.0.0.0/0
  lefttrsasigkey=0sA[...]
  #
  right=10.11.12.13
  rightid=@branch2
  rightsubnet=10.0.1.0/24
  righttrsasigkey=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

路战员利用动态分配的 IP 地址（如笔记本电脑）移动客户。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，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
  leftauthserver=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

这个选项指定一个指向用于导入证书的友好名称或 **nickname** 的证书。通常，名称作为 **PKCS #12 证书捆绑包**的一部分生成，格式为 **a.p12 文件**。有关详细信息，请参见 **pkcs12(1)** 和 **pk12util(1) man page**。

在移动客户端（即路径战器设备）上，使用之前配置的稍有变化：

```
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。如果要在以后建立连接，将其替换为 **auto=add**。

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

Libreswan 提供了一种方法，用于原生分配 IP 地址和 DNS 信息，以便在使用 **XAUTH IPsec** 扩展建立连接时轮转 VPN 客户端。可以使用 **PSK** 或 **X.509** 证书部署扩展身份验证(**XAUTH**)。使用 **X.509** 部署更为安全。客户端证书可通过证书撤销列表或在线证书状态协议 (**OCSP**) 撤销。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），理论上可以这样做。

XAUTH 要求 VPN 客户端额外使用用户名和密码来标识自身。对于一次性密码(**OTP**)，如 **Google Authenticator** 或 **RSA SecureID** 令牌，一次性令牌附加到用户密码中。

XAUTH 有三个可能后端：

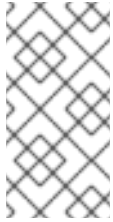
xauthby=pam

这使用 **/etc/pam.d/pluto** 中的配置来验证用户。可插拔验证模块(**PAM**)可以配置为自行使用各种后端。它可以使用系统帐户用户密码方案、**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（它不应该与 **/etc/ipsec.d/nsspassword** 文件混淆）。该文件的格式与 **Apache .htpasswd** 文件类似，并且 **Apache htpasswd** 命令可用于在该文件中创建条目。但是，在用户名和密码后，使用的 **IPsec** 连接的连接名称需要第三列，例如使用 **conn** 远程用户提供 VPN 删除用户时，密码文件条目应如下所示：

```
user1:$apr1$MlwQ3DHb$1l69LzTnZhCT2DPQmAOK.:remoteusers
```

**注意**

使用 `htpasswd` 命令时，必须在每行的 `user:password` 部分后手动添加连接名称。

xauthby=alwaysok

服务器始终预设 **XAUTH** 用户和密码组合正确。客户端仍然必须指定用户名和密码，尽管服务器忽略了这些用户名和密码。只有在用户已经由 **X.509** 证书标识时才使用，或者在测试 VPN 时不需要 **XAUTH** 后端。

带有 **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 `xauthfail` 设置为软（而不是硬）时，身份验证失败将被忽略，并且 VPN 的设置就像用户正确验证一样。可以使用自定义的 `updown` 脚本来检查环境变量 `XAUTH_FAILED`。然后，可以使用 `iptables`

DNAT “将此类用户重定向到围栏地带，他们可以联系管理员或续订付费服务订阅”。

VPN 客户端使用 `modecfgdomain` 值和 **DNS** 条目将指定域的查询重定向到这些指定的名称服务器。这允许轮转用户使用内部 **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` 仅支持最多两个名称服务器 IP 地址。要选择向 **VPN cliens** 发送横幅文本，请使用 `modecfgbanner` 选项。

如果左侧子网不是 `0.0.0.0/0`，则拆分隧道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**只会通过 **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，该密码取决于使用的后端。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` 文件中的 **PAM** 配置中指定的位置获取。

xauthby=alwaysok

未检查密码，并且始终接受该密码。使用这个选项进行测试，或者确保仅供 **xauth** 客户端兼容。

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. 使用对 Quantum Computers 的保护

将 **IKEv1** 与 **PreShared Keys** 搭配使用，可为量级攻击者提供保护。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`。要需要 **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，并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。有一个实验功能可以使用基于一次性的 **Dynamic PPK**。对于每个连接，会将一次性 pad 的新部分用作 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。如果没有再保留时间 pad 材料，连接会失败。详情请查看 [ipsec.secrets\(5\) man page](#)。



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

以下信息来源提供了有关 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](#) - 包含有关配置 ipsec 的信息。
- [ipsec.secrets\(5\) man page](#) - 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操作通过自动交换密钥创建的 Libreswan IPsec 连接。
- [ipsec_rsasigkey\(8\)man page](#) - 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。Theopenssl 命令行实用程序启用使用 shell 中的加密功能。它包括交互模式。

Theopenssl 命令行实用程序有多个 pseudo-commands，用于提供有关系统上安装的openssl 版本的命令的信息。pseudo-commands list-standard-commands、list-message-digest-commands 和 list-cipher-commands 会分别输出可在 presentopenssl 实用程序中使用的所有标准命令、消息摘要命令或密码命令的列表。

pseudo-commands list-cipher-algorithms 和 list-message-digest-algorithms 列出了所有密码和消息摘要名称。pseudo-command list-public-key-algorithms 列出了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-commandno- command-name 测试指定名称的命令名是否可用。适用于 shell 脚本。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，公钥派生自对应的私钥。因此，一旦决定算法，第一步是生成私钥。在这些示例

中，私钥被称为 **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** - 所生成的密钥中的位数。如果未指定 **1024**，则使用。
- **rsa_keygen_pubexp:value** - **RSA** 公共exponent 值。这可以是一个大的十进制值，或者一个十六进制值（以 **0x** 开头）。默认值为 **65537**。

例如，要使用 **3** 创建 **2048** 位 **RSA** 私钥，请使用以下命令：

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

要使用 **128** 位 **AES** 和密码 “**hello**” 对私钥进行加密，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，需要有一个可用的私钥。在这些示例中，私钥被称为 **privkey.pem**。如果您还没有生成私钥，请参阅 [第 4.7.1 节“创建和管理加密密钥”](#)

要让证书认证机构 (**CA**) 签署证书，必须生成证书并将其发送到 **CA** 进行签名。这称为证书签名请求。如需更多信息，请参阅 [第 4.7.2.1 节“创建证书签名请求”](#)。另一种方法是创建自签名证书。如需更多信息，请参阅 [第 4.7.2.2 节“创建自签名证书”](#)。

4.7.2.1. 创建证书签名请求

要创建证书以提交到 **CA**，请以以下格式发出命令：

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

这将创建一个名为 **cert.csr** 的 X.509 证书，以默认增强隐私的电子邮件 (PEM) 格式编码。PEM 名称来源于“Internet 电子邮件的隐私增强，如” [RFC 1424](#) 所述。要以其他 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，系统将提示您输入有关您和组织的相关信息，以便为证书创建可分辨名称(DN)。您需要以下信息：

- 您所在国家/地区的两个字母国家代码
- 您的州或省的完整名称
- 城市或富德语
- 您的机构名称
- 您机构中的单元名称
- 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 证书请求并生成工具。证书创建过程中使用的默认设置包含在 **/etc/pki/tls/openssl.cnf** 文件中。有关详细信息，请参阅 **manopenssl.cnf(5)**。

4.7.2.2. 创建自签名证书

要生成自签名证书，有效时间为 366 天，以以下格式发出命令：

```
~j$ 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。要查看用法说明，请按如下所示发出命令：

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

或者，更改到目录并发出 **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

由 CA 签名的证书被称为可信证书。因此，自签名证书是一个不受信任的证书。验证实用程序使用与 **OpenSSL** 在正常操作中相同的 **SSL** 和 **S/MIME** 函数来验证证书。如果发现错误，会报告该错误，然后尝试继续测试以报告任何其他错误。

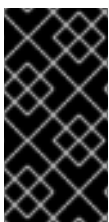
要验证 **PEM** 格式的多个单独的 **X.509** 证书，请以以下格式发出命令：

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

要验证证书链，**leaf** 证书必须在 **cert.pem** 中，您不信任的中间证书必须在不可信.pem 中直接串联。可信 **root CA** 证书必须是 `/etc/pki/tls/certs/ca-bundle.crt` 中列出的默认 **CA** 或 **cacert.pem** 文件中。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

由于此算法的强度不够，**Red Hat Enterprise Linux 7** 中禁止使用 **MD5** 哈希算法验证签名。始终使用 **SHA256** 等强算法。

4.7.4. 加密和解密文件

对于使用 OpenSSL 加密（和解密）文件，可以使用 `pkeyutil` 或 `enc-in` 命令。使用 `pkeyutil` 时，RSA 密钥用于执行加密和解密，而使用对称算法时，会使用对称算法。

使用 RSA 密钥

要加密名为 `plaintext` 的文件，请按如下所示发出命令：

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

密钥和证书的默认格式是 PEM。如果需要，使用 `-keyform DER` 选项来指定 DER 密钥格式。

要指定加密引擎，请按如下所示使用 `-engine` 选项：

```
~]$ openssl pkeyutil -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
```

要为名为 `plaintext` 的数据文件签名，请按如下所示发出命令：

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

要验证签名的数据文件并提取数据，请使用以下命令：

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

要验证签名，例如使用 DSA 密钥，请按如下方式发出命令：

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

`pkeyutil(1)` 手册页描述了公钥算法工具。

使用对称算法

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
```

若要指定算法，可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 **明文** 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

The **enc** 命令不能正确支持 **AEAD** 密码，并且 **ecb** 模式被视为不安全。要获得最佳结果，请不要使用 **cbc**、**cfb**、**sb** 或 **c tr** 之外的其他模式。

4.7.5. 生成消息摘要

dgst 命令以十六进制格式生成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。消息摘要命令采用以下格式：

```
openssl dgst algorithm -out filename -sign private-key
```

其中，算法是 **md5/md4/md2/sha1/sha/mdc2/ripemd160/dss1**。编写本文时，首选使用 **SHA1** 算法。如果您需要使用 **DSA** 签名或验证，则必须将 **dss1** 选项与包含 **-rand** 选项指定的随机数据的文件一同使用。

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
```

要以数字方式使用私钥 **privekey.pem** 以数字方式签署摘要，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. 生成密码哈希

`passwd` 命令计算密码的哈希。要在命令行中计算密码哈希，请按如下所示发出命令：

```
~]$ openssl passwd password
```

加密算法默认使用。

要从标准输入计算密码哈希，请使用基于 MD5 的 BSD 算法 1，请按以下方式发出命令：

```
~]$ openssl passwd -1 password
```

`apr1` 选项指定 BSD 算法的 Apache 变体。



注意

只有在禁用 FIPS 模式的情况下，才使用 `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哈希并使用 `salt xx`，请按如下方式发出命令：

```
~]$ openssl passwd -salt xx -in password-file
```

密码发送到标准输出，并且没有 `-out` 选项可指定输出文件。`table` 将生成密码哈希表及其相应的明文密码。

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
```

可以使用冒号指定用于查看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. 对您的系统进行基准测试

要测试给定算法的系统计算速度，请以以下格式发出命令：

```
~]$ openssl speed algorithm
```

其中，**算法**是您要使用的受支持的算法之一。要列出可用的算法，键入**openssl speed**，然后按 **Tab** 键。

4.7.9. 配置 OpenSSL

OpenSSL 具有一个配置文件 **/etc/pki/tls/openssl.cnf**，称为主配置文件，由 OpenSSL 库读取。也可以为每个应用提供单独的配置文件。配置文件包含多个部分，其名称如下：**[section_name]**。注意文件的第一部分，直至第一个 **[部分_name]**，称为默认部分。当 OpenSSL 搜索配置文件中的名称时，首先搜索指定部分。所有 OpenSSL 命令都使用 master OpenSSL 配置文件，除非命令中使用了选项来指定其他配置文件。**config(5) man page** 中详细解释了配置文件。

两个 RFC 解释证书文件的内容。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](#)
- [Internet X.509 公钥基础架构证书和证书撤销列表\(CRL\)配置文件的更新](#)

4.8. 使用 STUNNEL

stunnel 程序是客户端和服务端之间的加密打包程序。它侦听其配置文件中指定的端口，使用客户端加密连接，并将数据转发到侦听其常用端口的原始守护进程。这样，您可以保护任何本身不支持任何类型的

加密的服务，或者出于安全原因（如 SSL 版本 2 和 3）提高使用某种加密的服务的安全性，如 SSL 版本 2 和 3，受 POODLE SSL 漏洞(CVE-2014-3566)的影响。<https://access.redhat.com/solutions/1234773> 详情请查看。CUPS 是无法在其配置中禁用 SSL 的组件的示例。

4.8.1. 安装 stunnel

作为 root 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.

无论您使用何种服务，您都需要适用于 stunnel 的有效证书。如果您没有合适的证书，您可以向认证机构申请以获得证书，或者创建自签名证书。



警告

对于生产环境中运行的服务器，始终使用证书颁发机构签名的证书。自签名证书仅适用于测试用途或专用网络。

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 用户身份输入以下命令：

```
certs]# make stunnel.pem
```

回答所有问题以完成此过程。

2.

当您有证书时，请为 stunnel 创建配置文件。它是一个文本文件，其中的每一行都指定选项或服务定义的开头。您还可以在文件中保留注释和空行，以提高其可辨识度，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，但请使用 `/etc/stunnel/stunnel.conf`。以下内容将 `stunnel` 配置为 TLS 打包程序：

```
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，您可以通过将包含 `sslVersion = TLSv1` 的行替换为以下行来避免 SSL：

```
options = NO_SSLv2
options = NO_SSLv3
```

这些选项的用途如下：

- **cert** - 证书的路径
- **sslVersion** - SSL 的版本；请注意，您可以在此处使用 TLS，即使 SSL 和 TLS 是两个独立的加密协议
- **chroot** - 为提高安全性，更改了 `stunnel` 进程运行的根目录
- **setuid, setgid** - `stunnel` 进程作为运行的用户和组；`nobody` 是一个受限的系统帐户
- **pid** - `stunnel` 保存其进程 ID 的文件（相对于 `chroot`）
- **socket** - 本地和远程套接字选项；在本例中，禁用 Nagle 的算法来提高网络延迟

- **[service_name]** - 服务定义的开头；此行下方使用的选项仅适用于给定服务，而上面的选项则全局影响 **stunnel**
- **accept** - 要侦听的端口
- **connect** - 要连接的端口；这必须是您正保护的服务使用的端口
- **TIMEOUTclose** - 从客户端等待 **close_notify** 警报的秒数；0 指示 **stunnel** 完全不要等待
- **options** - OpenSSL 库选项

例 4.3. 保护 CUPS

要将 **stunnel** 配置为 **CUPS** 的 TLS 打包程序，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，并为 **setuid** 选项指定的用户授予其写入权限。要做到这一点，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.

创建配置文件和 **chroot** 目录后，当您确定可以访问指定的端口时，您已准备好使用 **stunnel**。

4.8.3. 启动、停止和重启 **stunnel**

要启动 **stunnel**，以 **root** 用户身份输入以下命令：

```
~]# stunnel /etc/stunnel/stunnel.conf
```

默认情况下，**stunnel** 使用 **/var/log/secure** 来记录其输出。

要终止 **stunnel**，请以 **root** 用户身份运行以下命令来终止进程：

```
~]# kill `cat /var/run/stunnel/stunnel.pid`
```

如果在 **stunnel** 运行时编辑配置文件，请终止 **stunnel** 并重新启动它，以使您的更改生效。

4.9. 加密

4.9.1. 使用 **LUKS** 磁盘加密

Linux 统一密钥设置磁盘格式（或 **LUKS**）允许您加密 **Linux** 计算机上的分区。当涉及到移动计算机和可移动介质时，这一点尤为重要。**LUKS** 允许多个用户密钥解密主密钥，用于批量加密分区。

LUKS 概述

LUKS 做什么

- **LUKS** 对整个块设备进行加密，因此非常适合保护移动设备的内容，如可移动存储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。这使得其在加密交换设备时很有用。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 增强了密码短语，可防止字典攻击。**
- **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：

- **对于需要很多（超过 8 个）用户拥有同一设备的不同访问密钥的情况，LUKS 并不十分适合。**
- **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。当系统处于 on 状态并且 LUKS 解密了磁盘后，该磁盘上的文件将可供通常具有访问权限的任何人使用。

4.9.1.1. Red Hat Enterprise Linux 中的 LUKS 实施

红帽企业 Linux 7 使用 LUKS 执行文件系统加密。默认情况下，在安装过程中取消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，系统将提示您输入密码短语，每次引导计算机时都会询问该密码。此密语“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，您可以选择加密哪个分区。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 - 加密的 Salt-Sector 初始化 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

这个过程将删除您要加密的分区中的所有数据。您丢失了所有信息！在开始此过程前，请确保将数据备份到外部源！

1. 通过以 **root** 用户身份在 **shell** 提示符后输入以下内容来输入运行级别 **1**：

```
telinit 1
```

2. 卸载现有的 **/home**：

```
umount /home
```

3. 如果上一步中的命令失败，使用 **fuser** 查找正在切换 **/home** 的进程并终止它们：

```
fuser -mvk /home
```

4. 验证 **/home** 不再挂载：

```
grep home /proc/mounts
```

5.

使用随机数据填充分区：

```
shred -v --iterations=1 /dev/VG00/LV_home
```

此命令会按照设备的顺序写入速度进行，可能需要一些时间才能完成。确保未加密的数据不会保留在已使用设备中，并且模糊处理包含加密数据的设备部分，而非只是随机数据，这是一个重要的步骤。

6.

初始化分区：

```
cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home
```

7.

打开新加密的设备：

```
cryptsetup luksOpen /dev/VG00/LV_home home
```

8.

确保设备存在：

```
ls -l /dev/mapper | grep home
```

9.

创建文件系统：

```
mkfs.ext3 /dev/mapper/home
```

10.

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.

在 `/etc/crypttab` 文件中添加以下内容：

```
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 中的条目使计算机在引导时询问您的 **luks** 密码短语。

17.

以 **root** 用户身份登录，再恢复您的备份。

您现在有一个加密分区，可在计算机关闭时安全地保留您的所有数据。

4.9.1.3. 添加新密码到现有设备

使用以下命令在现有设备中添加新的密码短语：

```
cryptsetup luksAddKey device
```

在提示输入任何一种现有的 **passprases** 进行身份验证后，系统将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
```

系统将提示您输入要删除的密码短语，然后提示您输入剩余的任何密码短语进行身份验证。

4.9.1.5. 在 Anaconda 中创建加密的块设备

您可以在系统安装过程中创建加密的设备。这可让您轻松使用加密分区配置系统。

要启用块设备加密，请在创建独立分区、软件 RAID 阵列或逻辑卷时选择自动分区或加密复选框时选中加密 系统复选框。完成分区后，系统将提示您输入加密密码短语。需要此密码才能访问加密的设备。如果您已有 LUKS 设备，并且在安装过程前面为它们提供了正确的密码短语，则密码输入对话框也将包含复选框。选中此复选框表示您希望将新密码添加到每个预先存在的加密块设备中的可用插槽。



注意

在自动分区屏幕上选中 **Encrypt System** 复选框，然后选择“创建自定义布局”不会导致任何块设备被自动加密。



注意

您可以使用 **Kickstart** 为每个新的加密块设备设置单独的密码短语。

4.9.1.6. 其它资源

如需关于 Red Hat Enterprise Linux 7 下 LUKS 或加密硬盘驱动器的更多信息，请访问以下链接之一：

- [LUKS 主页](#)
- [LUKS/cryptsetup 常见问题解答](#)
- [LUKS - Linux 统一密钥设置维基百科文章](#)
- [HOWTO：使用第二个硬盘和 pvmove 创建加密物理卷\(PV\)](#)

4.9.2. 创建 GPG 密钥

GPG 用于识别自身并验证您的通信，包括那些与您不知道的人员之间的通信。GPG 允许任何人读取 GPG 签名的电子邮件，以验证其真实性。换句话说，GPG 允许人们合理地确保您签名的通信确实来自您。GPG 很有用处，因为它有助于防止第三方更改代码或拦截对话并更改邮件。

4.9.2.1. 在 GNOME 中创建 GPG 密钥

要在 GNOME 中创建 GPG 密钥，请按照以下步骤执行：

1. 安装 **Seahorse** 工具，它可以简化 GPG 密钥管理：

```
~]# yum install seahorse
```
2. **Applications** → **Accessories** 菜单中选择 **Passwords** 和 **Encryption Keys**，这会启动应用 **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来描述您是谁（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。选择强的密码短语，但也易于记住。单击“确定”，该密钥已创建。



警告

如果您忘记了密码短语，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 **Key ID** 列。在大多数情况下，如果要求您输入密钥 ID，请将 **0x** 添加到密钥 ID 的前面，如 **0x6789ABCD** 中所示。您应该备份私钥，并将其保存在安全的位置。

4.9.2.2. 在 KDE 中创建 GPG 密钥.

要在 KDE 中创建 GPG 密钥，请按照以下步骤执行：

1. 通过选择“应用程序实用程序加密工具”，从主菜单中选择 KGpg 程序。如果您之前从未使用过 KGpg，则该程序将引导您完成创建您自己的 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。输入您的姓名、电子邮件地址和可选注释。您也可以为您的密钥选择到期时间，以及关键强度（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

如果您忘记了密码短语，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 Key ID 列。在大多数情况下，如果要求您输入密钥 ID，请将 0x 添加到密钥 ID 的前面，如 0x6789ABCD 中所示。您应该备份私钥，并将其保存在安全的位置。

4.9.2.3. 使用命令行创建 GPG 密钥

1. 使用以下命令：

```
~]$ 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对您的通信进行身份验证和解密。尽可能广泛分发您的公钥，特别是发送给您知道希望接收来自您的身份验证通信的人员，如邮寄列表。

2. 一系列提示会指示您完成此过程。如果需要，按 Enter 键以分配默认值。第一个提示要求您选择您首选的键类型：

```
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** 密钥不仅允许您签署通信，还可用于加密文件。

3. 选择密钥大小：

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

同样，默认值 **2048** 对于几乎所有用户来说已经足够，并且代表了非常强的安全性。

4. 选择密钥何时过期。最好选择到期日期而不是使用默认值，这是没有的。例如，如果密钥上的电子邮件地址变得无效，则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 **1y** 的值使密钥在一年内有效。（如果您改变主意，您可以在生成密钥后更改此过期日期。）

5. 在 **gpg2** 应用程序询问签名信息之前，会出现以下提示：

```
Is this correct (y/N)?
```

输入 **y** 以完成该过程。

6.

输入 **GPG 密钥** 的名称和电子邮件地址。请记住，这个过程是将您作为真实的个人进行身份验证。因此，请包含您的真实姓名。如果您选择虚假电子邮件地址，其他人很难找到您的公钥。这使得您的通信身份验证非常困难。如果您使用此 **GPG 密钥** 在邮件列表上自我引入，例如，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。（某些人将不同的密钥用于不同的目的，并通过一个注释识别每个密钥，如"Office"或"开源项目"。）

7.

在确认提示下，如果所有条目都正确，请输入字母 **O** 以继续，或使用其他选项来修复任何问题。最后，为您的 **secret** 密钥输入密语。**gpg2** 程序要求您输入两次密码短语，以确保您没有键入错误。

8.

最后，**gpg2** 生成随机数据，以使您的密钥尽可能唯一。在此步骤中移动鼠标、键入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，您的密钥就可以完成并可使用：

```
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.

密钥指纹是用于您的密钥的简写"签名"。它允许您向其他人确认自己已收到您的实际公钥，而无需任何篡改。您无需记下此指纹。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 8 位十六进制数组成，用于标识公钥。在上例中，**GPG 密钥 ID** 为 **1B2AFA1C**。在大多数情况下，如果要求您输入密钥 ID，请将 **0x** 添加到密钥 ID 的前面，如 **0x6789ABCD** 中所示。



警告

如果您忘记了密码短语，则无法使用密钥，并且任何使用该密钥加密的数据都将丢失。

4.9.2.4. 关于公钥加密

1. [Wikipedia - 公钥加密](#)
2. [HowStuffWorks - Encryption](#)

4.9.3. 将 openCryptoki 用于公共加密

OpenCryptoki 是 PKCS#11 的 Linux 实施，这是一种公共加密标准，定义了应用程序编程接口(API) 以加密设备，称为令牌。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 和 Starting Service

要在您的系统上安装基本的 openCryptoki 软件包，包括用于测试目的的软件实施，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，您可能需要安装其他软件包，以为您的特定用例提供支持。例如，要获得信任的平台模块(TPM)设备支持，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一般信息，请参阅 [Red Hat Enterprise Linux 7 系统管理员指南中的安装软件包部分](#)。

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为当前会话启动守护进程：

```
~]# systemctl start pkcsslotd
```

要确定在引导时自动启动该服务，请输入以下命令：

```
~]# systemctl enable pkcsslotd
```

如需了解有关如何使用 `systemd` 目标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为与系统及其插槽配合使用的令牌的信息。

该文件使用键值对定义各个插槽。每个插槽定义都可以包含描述、要使用的令牌库规格以及插槽制造商的 ID。（可选）可以定义插槽的硬件和固件版本。有关文件格式的描述信息，请参阅 `opencryptoki.conf(5)` 手册页，并获取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` 实用程序。此工具允许您显示和配置守护进程状态，以及列出和修改当前配置的插槽和令牌。例如，若要显示令牌相关信息，请发出下列命令（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` 系统组的一部分）：

```
~]$ pkcsconf -t
```

有关 `pkcsconf` 工具可用的参数列表，请查看 `pkcsconf(1)` 手册页。



警告

请记住，应该只有完全受信任的用户被分配在 `pkcs11` 组中，因为该组的所有成员都有权阻止 `openCryptoki` 服务的其他用户访问配置的 `PKCS#11` 令牌。此组的所有成员都可以使用 `openCryptoki` 的任何其他用户执行任意代码。

4.9.4. 使用智能卡向 OpenSSH 提供凭证

智能卡是 `USB` 记忆棒、`MicroSD` 或 `SmartCard` 形式的轻量级硬件安全模块。它提供远程管理的安全密钥存储。在红帽企业 Linux 7 中，`OpenSSH` 支持使用智能卡进行身份验证。

要将智能卡与 `OpenSSH` 一起使用，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#11` 库。`PKCS#11` 是一种公共加密标准，定义用于加密设备的应用程序编程接口(API)。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

要列出卡上的密钥，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，请将公钥传输到远程服务器。通过复制检索的字符串(key)并将其粘贴到远程 `shell`，或者将密钥存储到文件中（以下示例中的 `smartcard.pub`）并使用 `ssh-copy-id` 命令来完成此操作：

```
~]$ 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. 在智能卡中使用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，而无需暴露密钥本身。这意味着私钥不会留下卡。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`。

要在下一次连接到远程服务器时保存不必要的键入，请在 `~/.ssh/config` 文件中存储 `PKCS#11` 库的路径：

```
Host hostname
    PKCS11Provider /usr/lib64/pkcs11/opensc-pkcs11.so
```

在没有任何附加选项的情况下运行 `ssh` 命令连接：

```
[localhost ~]$ ssh hostname
Enter PIN for 'Test (UserPIN)':
[hostname ~]$
```

4.9.4.4. 使用 `ssh-agent` 自动登录 PIN Logging

设置环境变量以使用 `ssh-agent` 启动。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。使用以下命令检查您是否可以连接到身份验证代理：

```
~]$ ssh-add -l
Could not open a connection to your authentication agent.
~]$ eval `ssh-agent`
```

为了避免每次使用这个密钥连接时写入您的 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 要求个人身份验证(PIV)卡持有者明确用户操作，作为使用卡中存储的数字签名密钥的条件。OpenSC 正确强制实施此要求。

然而，对于某些应用程序，要求卡持有者为每个签名输入 PIN 是不切实际的。要缓存智能卡 PIN，请在 `pin_cache_ignore_user_consent = true` 前面删除 `#` 字符；`/etc/opensc-x86_64.conf` 中的选项。

如需更多信息，请参阅 [PIV Digital Signature Key\(NISTIR 7863\)的 Cardholder 身份验证报告](#)。

4.9.4.5. 其它资源

[Red Hat Enterprise Linux 7 中的智能卡支持](#)介绍了您的硬件或软件令牌的设置。

有关用于管理和使用智能卡和类似 PKCS#11 安全令牌的 `pkcs11-tool` 程序的更多信息，请参阅 `pkcs11-tool(1) man page`。

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，这意味着可以验证其完整性，这意味着它们可以被扩展验证模块(EVM)使用，以验证和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。

受信任的密钥需要硬件组件：受信任的平台模块 (TPM) 芯片，它用于创建和加密密钥（密封）。TPM 使用 2048 位 RSA 密钥（称为存储根密钥(SRK)）密封密钥。

此外，也可以使用一组特定的 TPM 平台配置寄存器 (PCR) 值密封可信密钥。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 密封的密钥只能被 TPM 在加密的同一系统中解密。但是，一旦加载了 aPCR-sealed 可信密钥（添加至密钥环），并且验证其关联的 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。单个密钥也可以保存为多个 blob，每个密钥都有不同的 PCR 值。

加密密钥不需要 TPM，因为它们使用 kernelAES 加密，这使其比可信密钥快。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。这个 master 密钥可以是可信密钥，也可以是用户密钥，这是它们的主要缺点 - 如果主密钥不是可信密钥，加密密钥就只像用于加密的用户密钥一样安全。

4.9.5.1. 使用密钥

在使用密钥执行任何操作前，请确保系统中载入了受信任和 加密的内核模块。在不同的 RHEL 内核构架中载入内核模块时请考虑以下几点：

- 对于具有 x86_64 架构的 RHEL 内核，将 TRUSTED_KEYS 和 ENCRYPTED_KEYS 代码构建为核心内核代码的一部分。因此，x86_64 系统用户可以使用这些密钥而无需载入 可信和 加密的密钥模块。
- 对于所有其他构架，需要先加载可信和 加密的内核模块，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 创建和密封可信密钥），它需要启用并激活。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，需要安装 `TrouSers` 应用程序（`trousers` 软件包）和 `tcscd` 守护进程（这是 `TrouSers` 套件的一部分）以与 TPM 通信。

要使用 TPM 创建可信密钥，请使用以下语法执行 `keyctl` 命令：

```
~J$ keyctl add trusted name "new keylength [options]" keyring
```

使用以上语法，可以按如下方式构建一个示例命令：

```
~J$ keyctl add trusted kmk "new 32" @u
642500861
```

上例创建一个名为 `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(@u)中。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：

```
~J$ keyctl show
Session Keyring
  -3 --alswrv 500 500 keyring:_ses
  97833714 --alswrv 500 -1 \_ keyring:_uid.1000
  642500861 --alswrv 500 500 \_ trusted: kmk
```

`print` 子命令会将加密的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：

```
~J$ keyctl pipe 642500861 > kmk.blob
```

要从用户空间 `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的密钥：

```
~]# 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
```

要在无法使用 TPM 的系统中创建加密密钥，请使用随机数字序列来生成用户密钥，然后用于密封实际加密的密钥。

```
~]# keyctl add user kmk-user "`dd if=/dev/urandom bs=1 count=32 2>/dev/null`" @u
427069434
```

然后，使用 **random-number** 用户密钥生成加密的密钥：

```
~]# keyctl add encrypted encr-key "new user:kmk-user 32" @u
1012412758
```

子命令列表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与用于加密它们的用户主密钥（随机数字密钥）一样安全。因此，主用户密钥应该尽可能安全加载，最好是在引导过程早期加载。

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密钥相关的其他信息。

安装的文档

- [keyctl\(1\)](#) - 描述 keyctl 实用程序及其子命令的使用。

在线文档

- [Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide](#) for Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南介绍了 SELinux 的基本原则，详细描述了如何通过各种服务（如 Apache HTTP 服务器）配置和使用 SELinux。
- <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”](#) 提供高级加密标准的简要描述。
- [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易损坏的安全加密密钥，需要一个随机数字源。一般来说，数字的随机性越高，获取唯一密钥的机会就越大。用于生成随机数字的熵通常从计算环境 “侦听”或使用硬件随机数字生成器获得。

`rngd` 守护进程是 `rng-tools` 软件包的一部分，能够同时使用环境嗅探和硬件随机数字生成器来提取熵。守护进程检查随机源提供的数据是否足够随机，然后将它存储在内核的随机数熵池中。它生成的随机数字通过 `/dev/random` 和 `/dev/urandom` 字符设备提供。

`/dev/random` 和 `/dev/urandom` 之间的区别在于，前者是一个块设备，这意味着当它确定熵量不足以生成正确的随机输出时，它会停止提供数字。相反，`/dev/urandom` 是一种非阻塞源，可重复利用内核的熵池，从而提供无限的伪随机数字，这是带有较少熵的保证。因此，`/dev/urandom` 不应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，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，要指定随机数输入的替代源（除 **/dev/hwrng**），请使用以下命令：

```
~]# rngd --rng-device=/dev/hwrng
```

上一命令使用 **/dev/hwrng** 启动 **rngd** 守护进程，作为从中读取随机数字的设备。同样，您可以使用 **-o**（或 **--random-device**）选项为随机数字输出（不同于默认的 **/dev/random**）选择内核设备。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，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

进入 **rngd -v** 命令后，按进程在后台继续运行。默认情况下会应用 **-b, --background** 选项（成为守护进程）。

如果没有 **TPM** 设备，您只会将 **Intel Digital Random Number Generator(DRNG)** 视为熵源。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

更多信息和软件代码示例，请参阅 [Intel Digital Random Number Generator\(DRNG\) 软件实施指南](#)。

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：

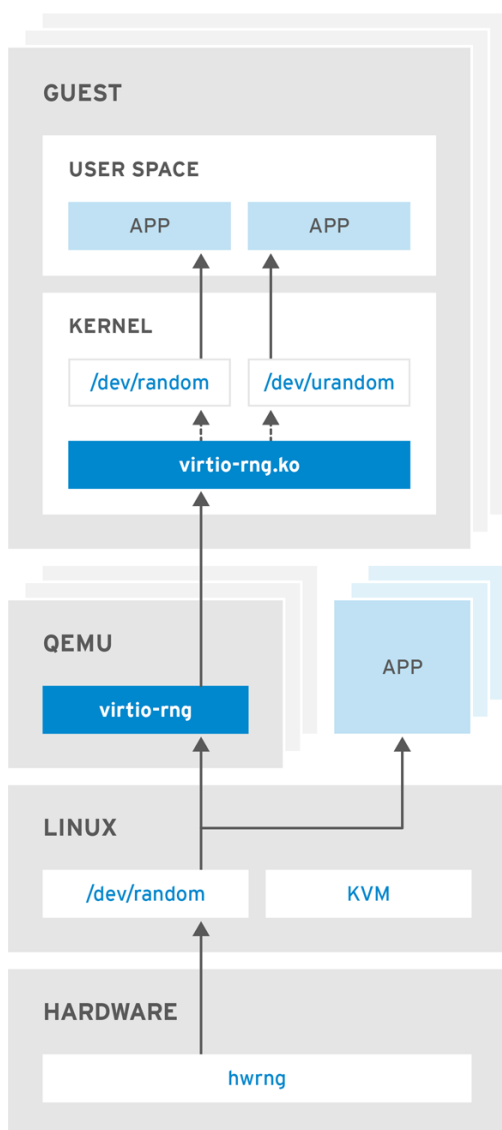
```
~]$ cat /dev/random | rngtest -c 1000
rngtest 5
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions. There is NO warranty; not even for
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

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明测试数据的随机性不足，不应依赖于。有关 **rngtest** 工具的选项列表，请查看 **rngtest(1)** 手册页。

红帽企业 Linux 7 引入了 **virtio RNG**（随机数字生成器）设备，它为 **KVM** 虚拟机提供了从主机中熵的访问权限。使用推荐的设置时，**hwrng** 馈入主机 Linux 内核的熵池中（通过 `/dev/random`），**QEMU** 将使用 `/dev/random` 作为客户机请求熵的来源。

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** 用户空间守护进程利用主机的熵。设置守护进程是每个 Red Hat Enterprise Linux 安装的手动步骤。借助红帽企业 Linux 7.1，消除了手动步骤，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，客户机内核本身会从主机获取熵。然后，客户机内核能够在应用程序请求时立即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了 **virtio-rng** 模块，在 Red Hat Enterprise Linux 安装期间提供可用的主机熵。



重要

要正确决定您应该在您的场景中使用的随机数字生成器，请参阅[了解 Red Hat Enterprise Linux 随机数字生成器文章](#)。

4.10. 使用基于策略的解密配置自动解锁加密卷

基于策略的解密(PBD)是一系列技术集，通过用户密码、受信任的平台模块(TPM)设备、与系统连接的 PKCS#11 设备（如智能卡或特殊网络服务器的帮助下）在物理和虚拟机上解锁加密的硬盘驱动器。

PBD 作为技术允许将不同的解锁方法合并到策略中，从而以不同的方式解锁同一卷。当前在红帽企业 Linux 中实施 PBD 包括 Clevis 框架和名为 pin 的插件。每个 pin 都提供单独的解锁功能。目前，唯一可用的两个插件是允许通过 TPM 解锁卷或使用网络服务器解锁卷。

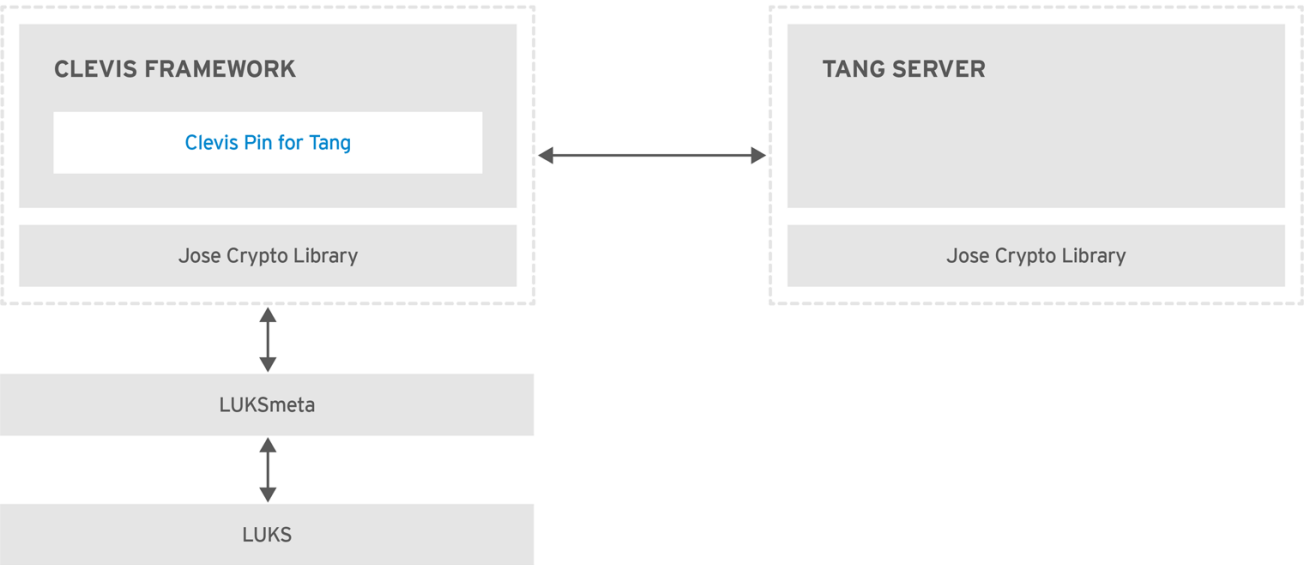
Network Bound Disc Enc Encryption(NBDE)是 PBD 技术的一个子类别，允许将加密卷绑定到特殊的网络服务器。NBDE 的当前实施包括用于 Tang 服务器的 Clevis pin 和 Tang 服务器本身。

4.10.1. Network-Bound Disk Encryption

网络绑定磁盘加密(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 加密



Tang 是一个将数据绑定到网络存在的服务器。当系统绑定到特定安全网络时，它会使包含可用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** 从未与任何客户端密钥交互，因此永远不会从客户端获得任何识别信息。

Clevis 是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** 软件包提供了该功能的客户端。

Clevis pin 是 **Clevis** 框架的一个插件。其中一个 **pins** 是实现与 **NBDE** 服务器交互的插件 - **Tang**。

Clevis 和 **Tang** 是通用客户端和服务组件，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，从而完成 **Network-Bound** 磁盘加密。

客户端和服务端组件都使用 **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** 服务器的情况下运行。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** 的调配状态存储在利用 **luksmeta** 软件包的 **LUKS** 标头中。

当客户端准备好访问其数据时，它会加载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（如 **/tmp**、**/var** 和 **/usr/local/** 目录）均包含在网络连接建立前需要启动的文件系统，被视为根卷。此外，在网络启动前，服务使用的所有挂载点（如 **/var/log/**、**var /log/audit/** 或 **/opt**）都需要在切换到 **root** 设备后尽早挂载。您还可以通过在 **/etc/fstab** 文件中没有 **_netdev** 选项来确定根卷。

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**，以 **root** 用户身份输入以下命令：

```
~]# yum install clevis
```

要解密数据，请使用 **clevis** 解密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 服务器可在 SELinux enforcing 模式下作为受限服务部署。

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，以 `root` 用户身份输入以下命令：

```
~]# yum install tang
```

2. 选择一个未设置的端口，例如 `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，某个端口一次只能由一个服务使用，因此尝试使用已占用的端口意味着 `ValueError: Port 已定义的` 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，打开了位于 `/etc/systemd/system/tangd.socket.d/` 目录中的空 `override.conf` 文件，通过添加以下行将 Tang 服务器的默认端口从 80 改为之前选择的编号：

```
[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 **systemd** 套接字激活机制，因此服务器将在第一次连接登录时立即启动。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) man page` 了解更多信息。

例 4.4. 轮转 Tang 密钥

定期轮转密钥非常重要。您轮转它们的确切间隔取决于您的应用程序、密钥大小以及机构策略。有关一些常见建议，请参阅 [Cryptary Key Length Proendation](#) 页面。

要轮转密钥，首先要在密钥数据库目录中生成新密钥，通常是 `/var/db/tang`。例如，您可以使用以下命令创建新签名和交换密钥：

```
~]# 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
```

将旧密钥重命名为具有从广告中隐藏的。请注意，以下示例中的文件名与密钥数据库目录中真实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，新客户端绑定采用新密钥，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.

客户冗余（推荐）

客户端应配置为绑定到多个 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。Clevi 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 man page：

- **Tang(8) 节高可用性**
- **Clevis(1) 节 Shamir's Secret Sharing**
- **clevis-encrypt-sss(1)**

红帽建议在高可用性部署中使用这个方法。

2.

密钥共享环

出于冗余目的，可以部署多个 Tang 实例。要设置第二个或后续的实例，请通过 SSH 安装 tang 软件包，并使用 rsync 将密钥目录复制到新主机上。请注意，红帽不推荐此方法，因为共享密钥会增加关键威胁的风险，需要额外的自动化基础架构。

4.10.4. 使用 Tang 为 NBDE 系统部署加密客户端

先决条件

- 已安装 Clevis 框架。请查看 [第 4.10.2 节“安装加密客户端 - Clevis”](#)
- Tang 服务器可用。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:

_Oslk0T-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。此密码文本是从 PLAINTEXT 输入文件中读取的。

要解密数据, 请使用 **clevis** 解密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) man page**, 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 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

在具有 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，且具有 **JSON** 配置对象的唯一参数：

```
~]$ clevis encrypt tpm2 '{}' < PLAINTEXT > JWE
```

要选择不同的层次结构、哈希和关键算法，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，只有在 **PCRs** 哈希值与密封时使用的策略匹配时，数据才能被取消密封。

例如，使用 **SHA1** 银行的索引 **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) man page**。

4.10.6. 配置手动注册卷

要自动解锁现有 **LUKS** 加密的根卷，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:
```

```
_Oslk0T-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 主密钥相同的熵创建新的密钥。
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. 启用用于 LUKS 的新密钥。

此磁盘现在可以使用您的现有密码以及 Clevis 策略解锁。如需更多信息，请参阅 `clevis-luks-bind(1)` *man page*。



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 对象是否已成功放入 LUKS 标头中，请使用 `theluksmeta 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-commandline "ip=192.0.2.10 netmask=255.255.255.0
gateway=192.0.2.1 nameserver=192.0.2.45"
```

或者，使用静态网络信息在 `/etc/dracut.conf.d/` 目录中创建 `.conf` 文件。例如：

```
~]# cat /etc/dracut.conf.d/static_ip.conf
kernel_commandline="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
```

如需更多信息，请参阅 `dracut.cmdline(7)` man page。

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报名流程。

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。密码是注册过程此步骤的临时密码。

```
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` 部分列出相关的 **Clevis** 软件包来安装相关的 **Clevis** 软件包：

```
%packages
clevis-dracut
%end
```

3.

在 `%post` 部分中调用 **clevis luks bind** 以执行绑定。之后，删除临时密码：

```
%post
clevis luks bind -f -k- -d /dev/vda2 \
tang '{"url":"http://tang.srv","thp":"_Oslk0T-E2l6qjfdDiwVmidoZjA"}' \<<< "temppass"
cryptsetup luksRemoveKey /dev/vda2 <<< "temppass"
%end
```

在上面的示例中，请注意，我们将 **Tang** 服务器上信任的指纹指定为绑定配置的一部分，启用完全非交互式的绑定。

在使用 **TPM 2.0** 策略而不是 **Tang** 服务器时，您可以使用类似的步骤。

有关 **Kickstart** 安装的详情请参考 [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化解锁

要自动解锁 **LUKS** 加密的可移动存储设备，如 **USB** 驱动器，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“配置手动注册卷”](#) 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，您可以使用类似的步骤。

4.10.9. 在引导时配置非 root 卷的自动解锁

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 解锁程序服务：

```
~]# 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“配置手动注册卷”所述。

4. 要在系统引导期间设置加密块设备，请将带有 `_netdev` 选项的对应行添加到 `/etc/crypttab` 配置文件。有关详细信息，请参见 `crypttab(5)` 手册页。

5. 将卷添加到 `/etc/fstab` 文件中的可访问文件系统列表中。在此配置文件中也使用 `_netdev` 选项。有关详细信息，请参见 `fstab(5)` 手册页。

4.10.10. 在 NBDE 网络中部署虚拟机

`clevis luks bind` 命令不会更改 LUKS 主密钥。这意味着，如果您创建 LUKS 加密镜像以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。如果您要在云中加密根卷，则需要确保为云中的每个 Red Hat Enterprise Linux 实例执行安装过程（通常使用 Kickstart）。如果没有同时共享 LUKS 主

密钥，就无法共享镜像。

如果您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”）或其他自动置备工具一起使用，以确保每个加密的虚拟机都有一个唯一的 master 密钥。

4.10.11. 使用 NBDE 为云环境构建可自动滚动的虚拟机镜像

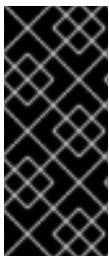
在云环境中部署可自动滚动的加密镜像会提供一套独特的挑战。与其他虚拟化环境一样，建议减少从单个镜像启动的实例数量，以避免共享 LUKS 主密钥。

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，为部署有限数量实例提供基础。要创建的实例的确切数量应当通过部署的安全策略来定义，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` 和 Kickstart 文件等系统来确保镜像构建过程中具有主密钥独有性。

云环境启用两个我们在此处考虑的 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 服务器可以在独立的基础架构上部署在云外，并在两个基础架构之间使用 VPN 链接进行部署。

在云中原生部署 Tang 有助于轻松部署。但是，由于它与其他系统的数据持久性层共享基础架构，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。访问此物理磁盘可以完全损坏密码文本数据。



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。这种云和 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。如果云基础架构面临风险，它还提供 Tang 服务器的本地控制。

4.10.12. 其它资源

[如何使用多个 LUKS 设备（Clevis+Tang 解锁）](#) 知识库文章设置网络绑定磁盘加密。

如需更多信息，请参阅以下 *man page*:

- `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(AIDE)是一个实用程序，可在系统上创建文件数据库，然后使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，以 `root` 用户身份输入以下命令：

```
~]# yum install aide
```

要生成初始数据库，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.
```

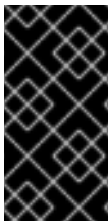
**注意**

在默认配置中，`aide --init` 命令仅检查 `/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
```

要更改 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位置，如只读介质。

**重要**

要避免 AIDE 数据库位置更改后 SELinux 拒绝，请相应地更新 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，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
...
```

至少应当将 AIDE 配置为每周运行扫描。最多应当每天运行 AIDE。例如，若要使用 `cron` 计划每天在 4:05am 执行 AIDE https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/system_administrators_guide/index#ch-Automating_System_Tasks（请参阅《系统管理员指南》中的自动系统任务章节），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

在验证了系统更改（如软件包更新或配置文件调整）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

如需有关 AIDE 的更多信息，请参阅以下文档：

- [aide\(1\) man page](#)
- [aide.conf\(5\) man page](#)
- [红帽企业 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 软件框架通过基于设备属性实施基本的白名单和黑名单功能，为防止入侵 USB 设备提供系统保护。为强制实施用户定义的策略，**USBGuard** 使用 Linux 内核 USB 设备授权功能。**USBGuard** 框架提供以下组件：

- 守护进程组件具有进程间通信(IPC)接口，以进行动态交互和策略实施。
- 与正在运行的 **USBGuard** 实例交互的命令行界面。
- 用于编写 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组件交互的 C++ API。

4.12.1. 安装 USBGuard

要安装 **usbguard** 软件包，以 **root** 用户身份输入以下命令：

```
~]# yum install usbguard
```

要创建初始规则集，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，编辑 `/etc/usbguard/rules.conf` 文件。如需更多信息，请参阅 `usbguard-rules.conf(5)` man page。另外，请参阅第 4.12.3 节“使用规则语言创建您的策略”的示例。

要启动 **USBGuard** 守护进程，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，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，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
"4PHGcaDKWtPjKDwYpIRG722cB9SIGz9l9lea93+Gt9c=" via-port "usb1" with-interface 09:00:00
...
```

```
6: block id 1b1c:1ab1 serial "000024937962" name "Voyager" hash
"CrXgiaWlf2bZAU+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 **usbguard** 命令和 **block-device** 选项：

```
~]# usbguard block-device 6
```

usbguard 使用块 并拒绝术语，其含义如下：

- **block** - 现在不和设备通信
- **拒绝** - 忽略这个设备，如 不存在

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. 创建白名单和黑名单

usbguard-daemon.conf 文件解析其命令行选项后，由 **usbguard** 守护进程加载，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。

要创建白名单或黑名单，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来加载策略规则集，再编写通过 IPC 接口接收的新规则。

IPCAccessControlFiles=<username> [<username> ...]

守护进程将接受来自这个以空格分隔的用户名列表的 IPC 连接。

IPCAllowedGroups=<groupname> [<groupname> ...]

守护进程将接受来自这个以空格分隔的组名称列表的 IPC 连接。

IPCAccessControlFiles=<path>

保存 IPC 访问控制文件的目录的路径。

ImplicitPolicyTarget=<target>

如何处理与策略中的任何规则不匹配的设备。可接受的值：**allow**、**block**、**reject**。

PresentDevicePolicy=<policy>

如何处理守护进程启动时已连接的设备：

- **allow** - 授权所有现有设备
- **block** - 取消授权所有现有设备
- **reject** - 删除所有存在的设备
- **keep** - 仅同步内部状态并保留它
- **apply-policy** - 评估每个现在设备的规则集

PresentControllerPolicy=<policy>

如何处理守护进程启动时已连接的 USB 控制器：

- **allow** - 授权所有现有设备
- **block** - 取消授权所有现有设备
- **reject** - 删除所有存在的设备
- **keep** - 仅同步内部状态并保留它
- **apply-policy** - 评估每个现在设备的规则集

例 4.5. usbguard 配置

以下配置文件会命令 **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/
```

若要指定 **IPC** 访问控制列表(**ACL**)，请使用 **usbguard add-user** 或 **usbguard remove-user** 命令。详情请查看 **usbguard(1)**。在本例中，要允许 **usbguard** 组中的用户修改 **USB** 设备的授权状态，列出 **USB** 设备，侦听异常事件，并列出 **USB** 授权策略，以 **root** 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，此接口默认仅限于 root 用户。考虑设置 `IPCAccessControlFiles` 选项（推荐）或 `IPCAAllowedUsers` 和 `IPCAAllowedGroups` 选项，以限制对 IPC 接口的访问。不要保留未配置 ACL，因为这会将 IPC 接口公开给所有本地用户，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)` man page 中的 IPC 访问控制部分。

4.12.3. 使用规则语言创建您的策略

`usbguard` 守护进程是否根据一组规则定义的策略授权 USB 设备。USB 设备插入系统时，守护进程会按顺序扫描现有规则，并且在找到匹配规则时，它会根据规则目标授权（允许）、取消授权（块）或删除（拒绝）设备。如果未找到匹配规则，则决定基于隐式默认目标。此隐式默认值将阻止设备，直到用户做出决策。

规则语言语法如下：

```
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（如目标、设备规格或设备属性）的详情，请查看 `usbguard-rules.conf(5)` man page。

例 4.6. usbguard 策略示例

允许 USB 大存储设备并阻止其他所有设备

这个策略会阻止不只是一个大容量存储设备的任何设备。USB 闪存磁盘中具有隐藏键盘接口的设备将被阻止。仅允许具有单个大容量存储接口的设备与操作系统交互。该策略由一条规则组成：

```
allow with-interface equals { 08:*:* }
```

阻止是隐式的，因为没有阻止规则。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决定为设备选择隐式目标。

允许通过特定端口连接特定的 Yubikey 设备

拒绝该端口上的其他所有内容。

```
allow 1050:0011 name "Yubico Yubikey II" serial "0001234567" via-port "1-2" hash
"044b5e168d40ee0245478416caf3d998"
reject via-port "1-2"
```

拒绝接口组合的可疑设备

USB 闪存磁盘实施键盘或网络接口非常可疑。以下规则形成一个策略，允许 USB 闪存磁盘并明确拒绝具有额外和可疑接口的设备。

```
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

黑名单错误的方法，您不应该只是将一组设备列入黑名单并允许其余设备列入黑名单。上面的策略假定阻止是隐式默认值。拒绝被视为“恶意”的一组设备是如何尽量限制此类设备暴露的好方法。

允许仅键盘 USB 设备

以下规则仅在没有允许键盘接口的 USB 设备时允许只使用键盘的 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` 命令首次生成策略后，编辑 `/etc/usbguard/rules.conf` 以自定义 USBGuard 策略规则。

```
~]$ usbguard generate-policy > rules.conf
~]$ vim rules.conf
```

要安装更新的策略并让您的更改有效，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\(1\) man page](#)
- [usbguard-rules.conf\(5\) man page](#)
- [usbguard-daemon\(8\) man page](#)
- [usbguard-daemon.conf\(5\) man page](#)
- [USBGuard 主页](#)

4.13. 强化 TLS 配置

TLS（传输层安全性）是用于保护网络通信的加密协议。在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，需要记住支持的客户端的范围越宽，进而降低由此产生的安全性。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。务必以最严格的可用配置为目标，且仅在出于兼容性原因需要时才放松。

请注意，Red Hat Enterprise Linux 7 中包含的库提供的默认设置足以满足大部分部署的需要。TLS 实施尽可能使用安全算法，而不阻止与旧客户端或服务器的连接。在具有严格安全要求的环境中应用本节中描述的强化设置，其中不支持安全算法或协议的传统客户端或服务器不会或者不允许连接。

4.13.1. 选择 Algorithms 来启用

需要选择和配置多个组件。以下每项都直接影响结果配置的稳健性（以及客户端的支持级别）或解决方案对系统的计算需求。

协议版本

TLS 的最新版本提供了最佳安全机制。除非有引人注目的原因包括支持旧版本的 TLS（甚至 SSL），否则请允许您的系统使用最新版本的 TLS 协商连接。

不允许使用 SSL 版本 2 或 3 进行协商。这两个版本都具有严重的安全漏洞。仅允许使用 TLS 1.0 或更高版本协商。应始终首选 TLS 的当前版本 1.2。



注意

请注意，目前，所有 TLS 版本的安全性取决于 TLS 扩展、特定密码（如下）的使用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)，且必须支持压缩，并且必须为针对 CBC-mode 加密器（Lucky Thirteen 攻击）的计时攻击实施缓解措施。TLS 1.0 客户端还需要实施记录分割（针对 BEAST 攻击的一个临时解决方案）。TLS 1.2 支持通过关联数据(AEAD)模式密码（如 AES-GCM、AES-CCM 或 Camellia-GCM）进行验证加密，这些密码没有已知问题。所有上述缓解方案均在 Red Hat Enterprise Linux 中包含的加密库中实施。

有关协议版本和推荐用法的快速概述，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.有无法以保证互操作性的方式缓解的已知问题，因此不默认启用缓解方案。不支持现代加密套件。

协议版本	使用建议
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 实施中包含的协议修复。不支持现代加密套件。
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，即使它们提供对 TLS 1.1 甚至 1.2 的支持。其动机在于试图实现最高级别的与可能不支持 TLS 最新版本的外部服务的互操作性。根据您的互操作性要求，启用最高可用版本的 TLS。



重要

不建议使用 SSL v3。但是，如果该事实被视为不安全且不适合一般使用，您绝对必须启用 SSL v3，请参阅第 4.8 节“使用 stunnel”以获得有关如何使用 stunnel 安全加密通信的说明，即使使用不支持加密的服务或者只能使用过时且不安全的加密模式。

密码套件

现代、更安全的密码套件应该优先于旧的不安全密码套件。总是禁用 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出口密码套件，其本意为较弱，因此容易中断。

虽然不会立即变得不安全，但提供安全性少于 128 位的密码套件在它们的短使用期中不应考虑。使用 128 位或更高安全性的算法可以预期在至少数年内不会被破坏，因此强烈建议您这样做。请注意，虽然 3DES 密码公告使用 168 位，但实际上提供了 112 位的安全性。

始终优先使用支持(perfect)转发保密 (PFS)的密码套件，这样可确保加密数据的机密性，即使在服务器密钥泄露时也是如此。此规则退出了快速 RSA 密钥交换，但允许使用 ECDHE 和 DHE。在两者中，ECDHE 是速度更快，因此是首选。

您还应优先选择 **AEAD** 密码，如 **AES-GCM**，在 **CBC-mode** 密码之前，因为它们不会受到 **padding oracle** 攻击。此外，在很多情况下，**AES-GCM** 在 **CBC** 模式中比 **AES** 快，特别是在硬件为 **AES** 加密加速器的情况下。

另请注意，在使用 **ECDSA** 证书的 **ECDHE** 密钥交换时，事务的速度甚至要快于纯 **RSA** 密钥交换。为了支持旧客户端，您可以在服务器上安装两对证书和密钥：一台带有 **ECDSA** 密钥（用于新客户端），另一个用于 **RSA** 密钥（用于旧密钥）。

公钥长度

使用 **RSA** 密钥时，始终首选使用至少由 **SHA-256** 签名的 3072 位的密钥长度，对于真正的 128 位安全性来说，这个值足够大。



警告

请记住，您的系统安全性仅与链中最弱链接一样强大。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

红帽企业 Linux 7 随附了多个功能全面的 TLS 实施。本节介绍了 **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 实施为各种密码套件提供支持，它们定义建立和使用 TLS 安全的通信时附带的所有元素。

在考虑第 4.13.1 节“选择 Algorithms 来启用”中列出的建议时，使用不同实施中包含的工具列出并指定密码套件，为您的用例提供最佳安全性。然后，生成的密码套件可用于配置各个应用程序协商和安全连接的方式。



重要

在每次更新或升级您使用的 TLS 实施或使用该实施的应用程序后，请务必检查您的设置。新版本可能会引入您不想启用的新密码套件，并且您的当前配置没有禁用。

4.13.2.1. 在 OpenSSL 中使用 Cipher Suite

OpenSSL 是一个工具包和加密库，支持 **SSL** 和 **TLS** 协议。在 **Red Hat Enterprise Linux 7** 中，`/etc/pki/tls/openssl.cnf` 中提供了一个配置文件。这个配置文件的格式在 `config(1)` 中进行描述。另请参阅 [第 4.7.9 节“配置 OpenSSL”](#)。

要获得安装 **OpenSSL** 所支持的所有密码套件的列表，请使用带有 `ciphers` 子命令的 `openssl` 命令，如下所示：

```
~]$ openssl ciphers -v 'ALL:COMPLEMENTOFALL'
```

将其他参数（在 **OpenSSL** 文档中称为密码字符串和 关键字）传递给 `ciphers` 子命令，以缩小输出范围。特殊关键字可用于仅列出满足特定条件的套件。例如，要只列出定义为属于 **HIGH** 组的套件，请使用以下命令：

```
~]$ openssl ciphers -v 'HIGH'
```

如需可用关键字和密码字符串列表，请参阅 `ciphers(1)` 手册页。

要获得满足 [第 4.13.1 节“选择 Algorithms 来启用”](#) 中概述的推荐的密码套件列表，请使用类似如下命令：

```
~]$ 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，优先选择临时销售曲线 **Diffie-Hellman** 密钥交换和 **ECDSA** 密码，省略 **RSA** 密钥交换（从而确保完美的转发保密）。

请注意，这是一种非常严格的配置，可能需要在现实场景中宽松条件，以便能够与更广泛的客户端兼容。

4.13.2.2. 在 GnuTLS 中使用 Cipher Suite

gnutls 是一个实现 **SSL** 和 **TLS** 协议及相关技术的通信库。



注意

Red Hat Enterprise Linux 7 上的 **GnuTLS** 安装提供了最佳默认配置值，可为大多数用例提供足够的安全性。除非您需要满足特殊安全要求，否则建议使用提供的默认值。

使用 **gnutls-cli** 命令和 **-l**（或 **--list**）选项列出所有支持的加密套件：

```
~J$ gnutls-cli -l
```

要缩小 **-l** 选项显示的密码套件列表，传递一个或多个参数（称为 **GnuTLS** 文档中的优先级字符串和关键字）到 **--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> 的 **GnuTLS** 文档。例如，使用以下命令获取至少 128 位安全性的密码套件列表：

```
~J$ gnutls-cli --priority SECURE128 -l
```

要获得满足第 4.13.1 节“选择 Algorithms 来启用”中概述的推荐的密码套件列表，请使用类似如下的命令：

```
~J$ 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-DSS-SHA256, SIGN-ECDSA-SHA256

以上命令将输出限制为安全性至少 128 位的密码，同时优先选择较强大的密码。它还禁止 RSA 密钥交换和 DSS 身份验证。

请注意，这是一种非常严格的配置，可能需要在现实场景中宽松条件，以便能够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。本节介绍最常用的服务器应用使用的 TLS 相关配置文件，并提供典型配置的示例。

无论您选择使用什么配置，请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。根据您的选择的 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，以 `root` 用户身份运行以下命令：

```
~]# 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 服务

器的 TLS 相关设置。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 file:///usr/share/httpd/manual/mod/mod_ssl.html `/usr/share/httpd/manual/mod_ssl.html` 中详细介绍。各种设置示例位于 [/usr/share/httpd/manual/ssl/ssl_howto.html](file:///usr/share/httpd/manual/ssl/ssl_howto.html) 中。

修改 `/etc/httpd/conf.d/ssl.conf` 配置文件中的设置时，请确定至少考虑以下三个指令：

SSLProtocol

使用此指令指定您要允许的 TLS（或 SSL）版本。

SSLCipherSuite

使用这个指令指定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，上面的配置是最少量的，可以通过遵循第 4.13.1 节“选择 Algorithms 来启用”中概述的建议来显著强化。

若要配置和使用 `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，尤其是配置文件的结构，以及可用的指令。请注意，`mod_nss` 指令具有前缀 `NSS` 而不是 `SSL`。有关 `mod_nss` 的信息，请参阅 https://git.fedorahosted.org/cgiit/mod_nss.git/plain/docs/mod_nss.html，包括不适用于 `mod_nss` 的 `mod_ssl` 配置指令列表。

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](https://www.dovecot.org/wiki/2.2.10/SSL.DovecotConfiguration.txt) 中找到该文件中提供的一些基本配置指令的说明（此帮助文件与 Dovecot 标准安装一同安装）。

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

使用此指令指定您要允许的 TLS（或 SSL）版本。

ssl_cipher_list

使用这个指令指定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，上面的配置是最少量的，可以通过遵循第 4.13.1 节“选择 Algorithms 来启用”中概述的建议来显著强化。

4.13.4. 附加信息

有关 TLS 配置和相关主题的更多信息，请参见以下列出的资源。

安装的文档

- **config(1)** - 描述 `/etc/ssl/openssl.conf` 配置文件的格式。
- **ciphers(1)** - 包括可用 OpenSSL 关键字和密码字符串列表。

- [/usr/share/httpd/manual/mod_ssl.html](#) - 包含由 mod_ssl 模块用于 Apache HTTP 服务器的 /etc/httpd/conf.d/ssl.conf 配置文件中的指令的详细描述。
- [/usr/share/httpd/manual/ssl/ssl_howto.html](#) - 包含由 Apache HTTP 服务器的 mod_ssl 模块使用的 /etc/httpd/conf.d/ssl.conf 配置文件中的真实设置的实用示例。
- [/usr/share/doc/dovecot-2.2.10/wiki/SSL.DovecotConfiguration.txt](#) - 说明 Dovecot/conf.d/10-ssl.conf 配置文件中使用的 /etc/dovecot/conf 配置文件中的的一些基本配置指令。

在线文档

- [Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide](#) for Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南介绍了 SELinux 的基本原则，详细描述了如何通过各种服务（如 Apache HTTP 服务器）配置和使用 SELinux。
- <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”](#) 此外，描述如何使用 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 共享检索系统证书定位点和黑名单信息的默认源。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 Trust Store

在 Red Hat Enterprise Linux 7 中，整合的系统范围信任存储位于 /etc/pki/ca-trust/ 和 /usr/share/pki/ca-trust-source/ 目录中。/usr/share/pki/ca-trust-source/ 中的信任设置的处理优先级低于 /etc/pki/ca-trust/ 中的设置。

证书文件根据安装到的子目录来处理：

- `/usr/share/pki/ca-trust-source/anchors/` 或 `/etc/pki/ca-trust/source/anchors/` - 用于信任定位符。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/` - 用于不可信证书。
- `/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

要在简单的 PEM 或 DER 文件格式中添加证书到系统中信任的 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，但不执行 `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以加载当前的系统证书配置。

4.14.3. 管理可信系统证书

要列出、提取、添加、删除或更改信任定位符，请使用 `trust` 命令。要查看这个命令的内置帮助信息，请在没有任何参数的情况下输入它，或使用 `--help` 指令输入它：

```
$ 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定位符和证书，请使用信任列表命令：

```
$ trust list
pkcs11:id=%d2%87%b4%e3%df%37%27%93%55%f6%56%ea%81%e5%36%cc%8c%1e%3f%bd;type=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;type=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
```

要将信任定位符存储到系统范围的信任存储中，请使用信任定位子命令并指定一个 **path.to** 证书，例如：

```
# trust anchor path.to/certificate.crt
```

要删除证书，请使用证书的路径到证书或者证书的 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 *man page*：

- [update-ca-trust\(8\)](#)
- [trust\(1\)](#)

4.15. 使用 MACSEC

介质访问控制安全（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。IPsec 在网络层（层 3）上运行，而 SSL 或 TLS 在应用层（层 7）上运行，而 MACsec 在数据链路层（层 2）中运行。将 MACsec 与其他网络层的安全协议相结合，以利用这些规则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：加密网络流量文章的不同解决方案](#)。

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` 实用程序在特殊文件或磁盘设备上设置模式，从而使检索数据变得更加困难。使用清理比在磁盘上写入随机数据要快。这个过程提供了高可用性、可靠性和数据保护。

要使用清理命令，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将被清理，并且其中的所有数据将被销毁。这是最有效的方法。

```
scrub [OPTIONS] special file
```

File

常规文件将被清理，仅文件中的数据被销毁。

```
scrub [OPTIONS] file
```

目录

使用 **-X** 选项时，将创建目录并填充文件，直到文件系统已满。然后，在文件模式中，文件将被清理为。

```
scrub -X [OPTIONS] directory
```

例 4.7. 清理 Raw 设备

要清理使用默认 **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：

```
~J$ 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：

```
~J$ head -1 file.txt
JnNpaTEveB/IYsbM9IhuJdw+0jKhwCIBUsxLXLayB8ultotUINHKKUeS/7bCRKDogE
P+yJm8VQkL
```

4.

清理文件：

```
~J$ 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：

```
~J$ cat file.txt
SCRUBBED!
```

6.

验证文件大小是否相同：

```
~J$ 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、方法和注意事项的更多信息，请参阅 **scrub(1) man page**。

第 5 章 使用防火墙

5.1. FIREWALLD入门

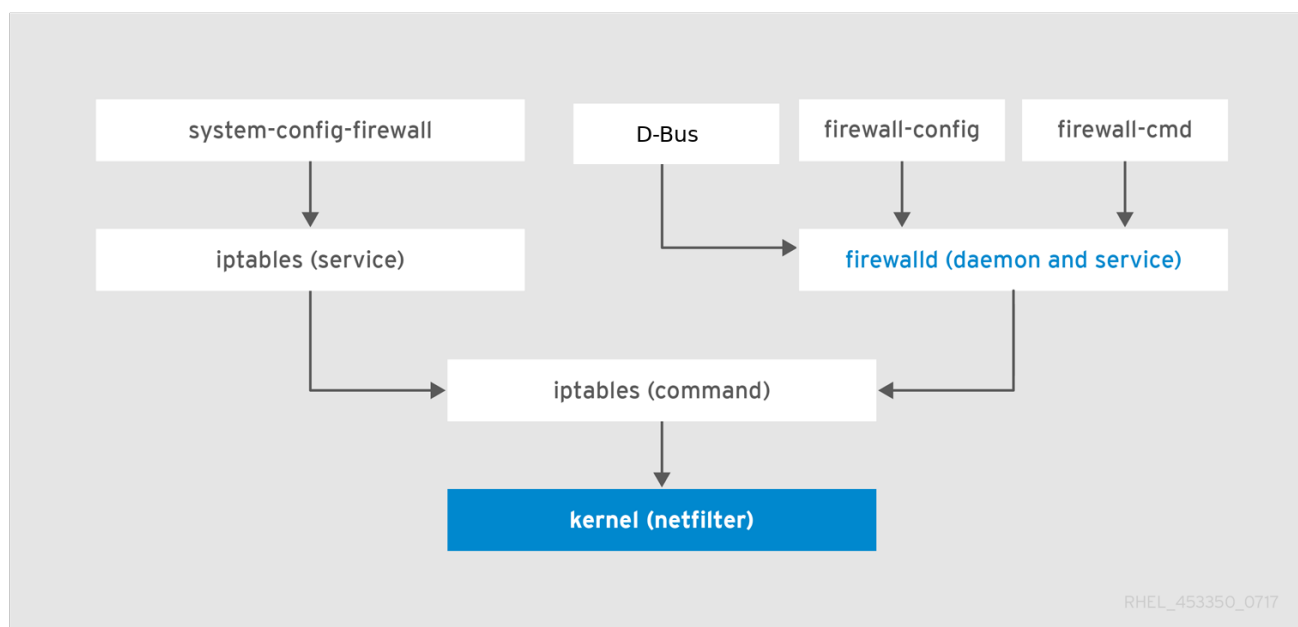
防火墙是保护机器不受来自外部的不需要的流量的一种方式。它允许用户通过定义一组防火墙规则来控制主机上的传入网络流量。这些规则用于对进入的流量进行排序，并可以阻断或允许流量。

firewalld 是一个防火墙服务守护进程，通过 **D-Bus** 接口提供动态可定制的主机防火墙。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。默认情况下，某些区（如可信区）允许所有流量。

图 5.1. 防火墙堆栈



[D]

5.1.1. Zones

firewalld 可以用来根据用户决定放置在该网络中的接口和流量级别的信任级别将网络划分为不同的区域。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 通知接口区域的 firewalld。您可以使用 firewall-config 工具或 firewall -cmd 命令行工具为接口分配区域。后两个仅编辑适当的 NetworkManager 配置文件。如果您使用 firewall-cmd 或 firewall-config 更改接口区域，则请求将转发到 NetworkManager，且不由 firewalld 处理。

预定义区域存储在 /usr/lib/firewalld/zones/ 目录中，并可立即应用于任何可用的网络接口。只有在修改后，这些文件才会复制到 /etc/firewalld/zones/ 目录中。下表描述了预定义区的默认设置：

block

对于 IPv6，任何传入的网络连接都将通过 aicmp-host-prohibited 消息来拒绝，用于 IPv4 和 icmp6-adm-prohibited 的消息。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。

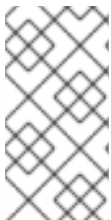
trusted

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默认区域。当接口连接添加到 **NetworkManager** 时，它们会被分配给默认区域。安装时，**firewalld** 中的默认区域设置为 **public** 区域。默认区可以被修改。



注意

已选择网络区名称进行自我解释，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。

firewalld.service(5) man page 中介绍了服务配置选项和通用文件信息。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

运行时模式中提交的任何更改都仅在 **firewalld** 运行时应用。**firewalld** 重启后，设置将恢复为其永久

值。

要使更改在重新引导后继续生效，请使用 `--permanent` 选项再次应用它们。或者，若要在 `firewalld` 运行时持久保留更改，可使用 `--runtime-to-permanent firewall-cmd` 选项。

如果在 `firewalld` 仅使用 `--permanent` 选项运行规则时设置了规则，则在重新启动 `firewalld` 之前，它们不会生效。不过，重新启动 `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。要在永久模式中修改防火墙设置，请将 `--permanent` 选项与 `firewall-cmd` 命令搭配使用。

```
~]# 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。使用此选项将排除 `--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
```

另外，在 GNOME 中，使用 Super 键并键入 **Software** 来启动 **Software Sources** 应用。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，然后单击“安装”按钮。

要运行 `firewall-config`，请使用 `firewall-config` 命令或按 Super 键进入“活动概览”，输入 **firewall**，然后按 Enter 键。

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` 安装在系统上。使用 `firewalld` CLI 界面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
```

此外，在尝试编辑设置前，了解如何设置 **firewalld** 以及哪些规则处于强制状态，这一点非常重要。要显示防火墙设置，请查看 [第 5.3.2 节“查看当前 firewalld 设置”](#)

5.3.2. 查看当前 firewalld 设置

5.3.2.1. 使用 GUI 查看允许的服务

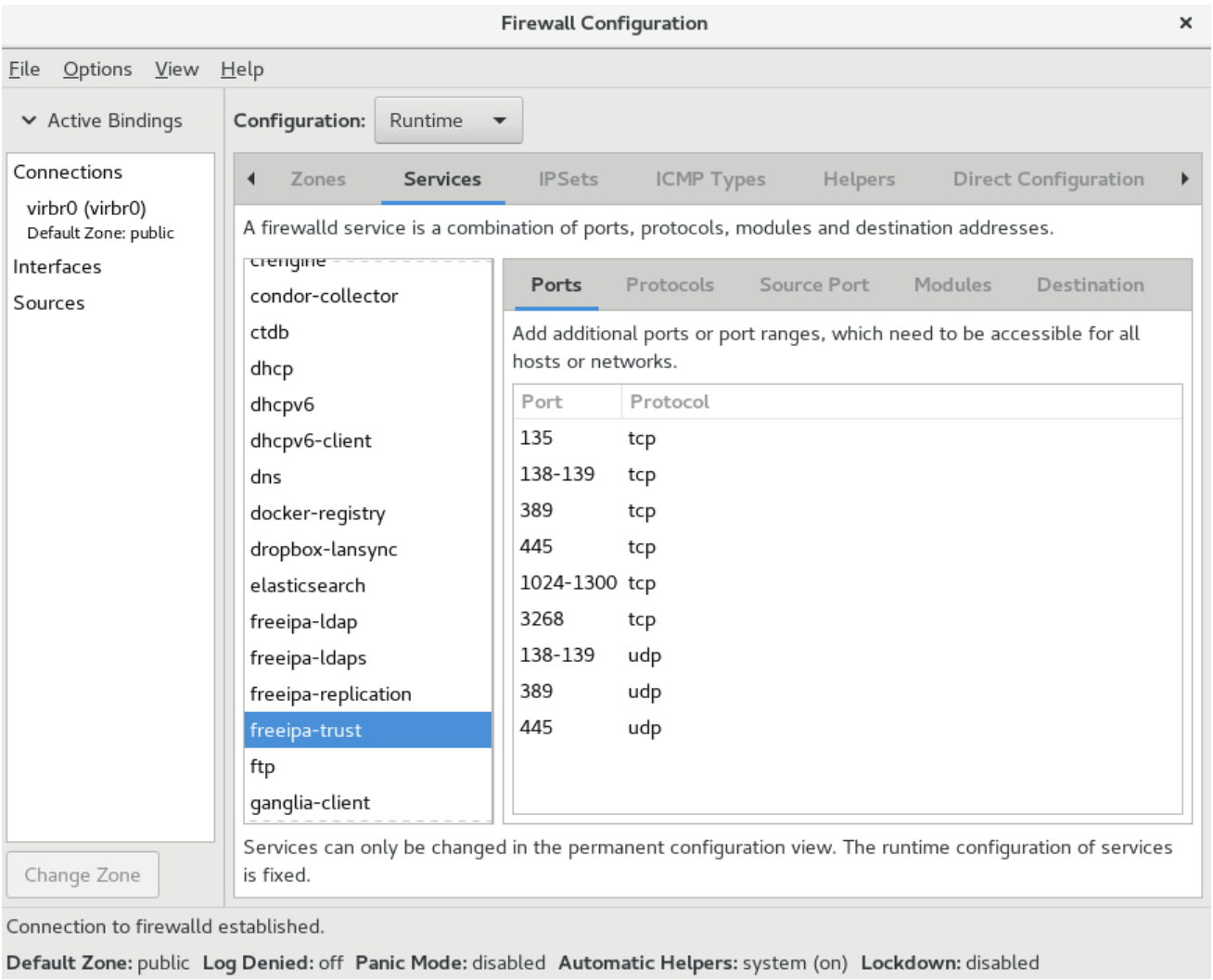
要使用图形化的 **firewall-config** 工具查看服务列表，请按 **Super** 键进入“活动概览”，键入 **firewall**，然后按 **Enter** 键。**firewall-config** 工具会出现。现在，您可以在“服务”选项卡下查看服务列表。

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
```

此时将打开防火墙配置窗口。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 中的服务选项卡



[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 使用区域来管理流量。如果 --zone 选项没有指定区域，该命令将在分配给活跃网络接口和连接的默认区域中有效。

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

要指定显示设置的区域，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，您允许 SSH 服务，`firewalld` 会打开该

服务所需的端口(22)。之后，如果您列出允许的服务，列表将显示 SSH 服务，但如果列出开放端口，则不会显示任何服务。因此，建议您使用 `--list-all` 选项来确保您收到完整信息。

5.4. 启动 FIREWALLD

要启动 `firewalld`，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，以 `root` 用户身份输入以下命令：

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

5.5. 停止 FIREWALLD

要停止 `firewalld`，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，以 `root` 用户身份输入以下命令：

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

要通过访问 `firewalld D-Bus` 接口以及其它服务需要 `firewalld` 来确保 `firewalld` 没有启动，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

可以使用图形化的 `firewall-config` 工具、`firewall-cmd` 和 `firewall-offline-cmd` 添加和删除服务。

或者，您可以编辑 `/etc/firewalld/services/` 目录中的 XML 文件。如果用户未添加或更改服务，则在 `/etc/firewalld/services/` 中找不到对应的 XML 文件。如果要添加或更改服

务，`/usr/lib/firewalld/services/` 目录中的文件可作为模板使用。

5.6.2. 使用 CLI 在紧急情况时禁用所有流量

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 使用预定义服务控制流量

控制流量的最简单方法是添加预定义服务到 **firewalld**。这会打开所有必要的端口，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** 工具并选择要配置服务的网络区。选择“服务”选项卡，然后选中您要信任的每种服务类型的复选框。清除要阻止服务的复选框。

要编辑服务，请启动 **firewall-config** 工具，然后从标记为 **Configuration** 的菜单中选择永久。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。

端口、协议和源端口 选项卡可为所选服务启用、更改和删除端口、协议和源端口。模块选项卡用于配置 Netfilter 帮助程序模块。Destination 选项卡启用限制到特定目标地址和互联网协议（IPv4 或 IPv6）的流量。



注意

在运行时模式中无法更改服务设置。

5.6.5. 添加新服务

可以使用图形化的 **firewall-config** 工具、**firewall-cmd** 和 **firewall-offline-cmd** 添加和删除服务。或者，您可以编辑 **/etc/firewalld/services/** 中的 XML 文件。如果用户未添加或更改服务，则在 **/etc/firewalld/services/** 中没有找到对应的 XML 文件。如果要添加或更改服务，则 **/usr/lib/firewalld/services/** 文件可用作模板。

要在终端中添加新服务，请使用 **firewall-cmd** 或 **firewall-offline-cmd**（如果未激活 **firewalld**）。输入以下命令来添加新和空服务：

```
~J$ 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：

```
~J$ firewall-cmd --new-service-from-file=service-name.xml
```

您可以使用 **additional --name=service-name** 选项来更改服务名称。

更改服务设置后，服务的更新副本将置于 `/etc/firewalld/services/` 中。

作为 **root** 用户，您可以输入以下命令手动复制服务：

```
~J# cp /usr/lib/firewalld/services/service-name.xml /etc/firewalld/services/service-name.xml
```

firewalld 最初从 `/usr/lib/firewalld/services` 加载文件。如果文件放置在 `/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。一旦删除了 `/etc/firewalld/services` 中的匹配文件，或者要求 **firewalld** 加载服务的默认值，则将立即使用 `/usr/lib/firewalld/services` 中的覆盖文件。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

端口是能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** 守护进程侦听端口 80。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

当不再需要打开端口时，在 **firewalld** 中关闭该端口。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

要允许通过防火墙到特定端口的流量，请启动 **firewall-config** 工具并选择要更改的网络区。选择端口选项卡，然后单击右侧的添加按钮。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 or 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** 工具并选择要更改的网络区。选择协议选项卡，然后单击右侧的添加按钮。此时会打开协议窗口。

从列表中选择协议，或者选择“其他协议”复选框并在字段中输入协议。

5.6.9. 使用 GUI 打开源端口

要允许来自特定端口的流量通过防火墙，请启动 **firewall-config** 工具并选择要更改的网络区。选择“源端口”选项卡，然后单击右侧的添加按钮。**Source Port** 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 or 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 命令显示系统上可用的所有区域，但不显示特定区域的任何详情。

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. 修改 Certain 区的 firewalld 设置

第 5.6.3 节“使用 CLI 使用预定义服务控制流量”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。

要在不同区域中工作，请使用 `--zone=zone-name` 选项。例如，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

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` 加载默认区域的设置并使它活跃。

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

遵循此过程后，该设置是永久设置，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` 选项使设置在重新启动后持久保留。如果您设置了一个新的默认区域，设置将变为永久设置。

5.7.5. 为网络连接分配默认区

当连接由 **NetworkManager** 管理时，必须了解它使用的区域。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域，请使用 **NetworkManager GUI** 或编辑 `/etc/sysconfig/network-scripts/ifcfg-connection-name` 文件并添加为这个连接分配区域的行：

```
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。新区域需要 `--permanent` 选项，否则命令不起作用。

要创建新区：

1.

创建一个新区：

```
~]# firewall-cmd --new-zone=zone-name
```

2.

检查是否在您的永久设置中添加了新的区：

```
~]# firewall-cmd --get-zones
```

3.

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

区也可以通过区域配置文件创建。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` 区域配置文件包含区域的信息。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一个服务(SSH)和一个端口范围的配置，适用于 TCP 和 UDP 协议：

```
<?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. 使用区域目标设置传入流量的默认行为

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：**default**、**ACCEPT**、**REJECT** 和 **DROP**。通过将目标设置为 **ACCEPT**，您接受除特定规则禁用的数据包外的所有传入数据包。如果将目标设置为 **REJECT** 或 **DROP**，则禁用除特定规则中允许的数据包之外的所有传入数据包。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. 使用区管理传入的流量依赖源

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 可信 区 192.168.2.15 的所有传入流量：

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排序，请使用 `--add-source-port` 选项指定一个源端口。您还可以将其与 `--add-source` 选项组合使用，将流量限制为特定的 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. 使用区和源允许服务仅用于特定域

要允许特定网络的流量在机器上使用服务，请使用区和源。以下步骤只允许来自 `192.0.2.0/24` 网络的 HTTP 流量，而任何其他流量都被阻断。



警告

配置此场景时，请使用具有默认目标的区域。使用将目标设置为 **ACCEPT** 的区域存在安全风险，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 范围添加到内部区，以通过区路由来自源的流量：

```
~]# firewall-cmd --zone=internal --add-source=192.0.2.0/24
```

3.

在内部区中添加 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.

检查内部区是否活跃，且该服务是否允许：

```
~]# 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. 配置受基于协议的区域接受的流量

您可以允许基于协议的区域接受传入的流量。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

若要接收多播流量，可将 `igmp` 值与 `--add-protocol` 选项搭配使用。

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，以便到达您系统中特定端口的任何传入流量都将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. 添加端口到重定向

在您将从一个端口的流量重定向到另一个端口或另一个地址之前，您需要了解三件事情：数据包到达哪个端口，使用什么协议，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 重定向到相同机器上的端口 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. 删除转发到相同机器上的端口 88 的 TCP 端口 80

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 伪装是一个进程，一个计算机充当网络的 IP 网关。对于伪装，网关会动态查找传出接口的 IP，并使用这个地址替换数据包中的源地址。

如果传出接口的 IP 改变，您可以使用伪装。伪装的典型用例是，路由器将私有 IP 地址（在互联网上路由）替换为路由器上传出接口的公共动态 IP 地址。

要检查 IP 伪装是否已启用（例如，对于 外部 区），以 root 用户身份输入以下命令：

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

如果已启用，命令将打印 **yes**，退出状态为 0。否则，将不会打印退出状态 1。如果省略 区域，则将使用默认区域。

要启用 IP 伪装，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，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，供各种网络设备用于发送错误消息和显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**）揭示关于您的网络的信息，并将此类信息滥用于各种类型的活动。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

位于 `/usr/lib/firewalld/icmptypes/` 目录中的独立 XML 文件中描述了 ICMP 请求。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。要查看哪个协议使用了 ICMP 请求：

```
~]# firewall-cmd --info-icmp-type=<icmp-type>
```

如果请求当前为 **blocked** 或 **no**，则 ICMP 请求的状态将显示 **yes**。查看 ICMP 请求当前是否被阻断：

```
~]# firewall-cmd --query-icmp-block=<icmp-type>
```

5.11.2. 阻塞或取消阻塞 ICMP 请求

当您的服务器阻断 ICMP 请求时，它不会提供通常会提供的信息。但这并不意味着根本不给出任何信息。客户端收到特定 ICMP 请求被阻止（拒绝）的信息。应仔细考虑阻止 ICMP 请求，因为它可能会导致通信问题，特别是 IPv6 流量。

查看 ICMP 请求当前是否被阻断：

```
~]# firewall-cmd --query-icmp-block=<icmp-type>
```

阻止 ICMP 请求：

```
~]# firewall-cmd --add-icmp-block=<icmp-type>
```

删除 ICMP 请求的块：

```
~]# firewall-cmd --remove-icmp-block=<icmp-type>
```

5.11.3. 在不提供任何信息的情况下阻止 ICMP 请求

通常，如果您阻止 **ICMP** 请求，客户端会知道您在阻止 **ICMP** 请求。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。

阻止和丢弃所有 ICMP 请求：

1.

将区的目标设置为 **DROP**：

```
~]# firewall-cmd --set-target=DROP
```

2.

使新设置具有持久性：

```
~]# firewall-cmd --runtime-to-permanent
```

现在，除您明确允许的流量外，所有流量（包括 **ICMP** 请求）将被丢弃。

阻塞和丢弃某些 ICMP 请求并允许其他请求：

1.

将区的目标设置为 **DROP**：

```
~]# firewall-cmd --set-target=DROP
```

2.

添加 **ICMP** 块 **inversion** 以阻止所有 **ICMP** 请求：

```
~]# firewall-cmd --add-icmp-block-inversion
```

3.

为您要允许的 **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
```

block inversion 反转 ICMP 请求块的设置，因此所有之前未阻断的请求都将被阻止。被拦截的人不会被拦截。这意味着，如果您需要取消阻塞请求，则必须使用 **blocking** 命令。

将其恢复到完全 **permissive** 设置：

1.

将区的目标设置为 **default** 或 **ACCEPT**:

```
~]# firewall-cmd --set-target=default
```

2.

删除 ICMP 请求添加的所有块：

```
~]# firewall-cmd --remove-icmp-block=<icmptype>
```

3.

删除 ICMP 块 **inversion**：

```
~]# 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 Filter

要启用或禁用 ICMP 过滤器，请启动 **firewall-config** 工具并选择过滤消息的网络区域。选择 **ICMP Filter** 选项卡，再选中您要过滤的每种 ICMP 消息的复选框。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。

若要启用反向 ICMP Filter，可单击右侧的 **Invert Filter** 复选框。现在仅接受标记的 ICMP 类型，所有其他均被拒绝。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** 支持的 IP 设置类型列表，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

可以在 `firewalld` 区域中使用 IP 集作为源，也可以用作富规则的来源。在 Red Hat Enterprise Linux 7 中，首选的方法是在直接规则中使用通过 `firewalld` 创建的 IP 集。

要列出永久环境中 `firewalld` 已知的 IP 集，以 `root` 用户身份运行以下命令：

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

要添加新 IP 集，以 `root` 用户身份使用永久环境运行以下命令：

```
~]# firewall-cmd --permanent --new-ipset=test --type=hash:net
success
```

上一命令为 IPv4 创建了名称 `test` 和 `hash:net` 类型的新 IP 设置。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`。使用以下命令以 `root` 用户身份列出新 IP 设置：

```
~]# firewall-cmd --permanent --get-ipsets
test
```

要获取有关 IP 集的更多信息，以 `root` 用户身份运行以下命令：

```
~]# 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。要在 `test` IP 集中添加一个条目，以 `root` 用户身份运行以下命令：

```
~]# firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
success
```

以上命令将 IP 地址 `192.168.0.1` 添加到 IP 集合中。要获取 IP 集合中当前条目列表，以 `root` 用户身份运行以下命令：


```
~]# 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。

要添加 iplist.txt 文件中的地址，以 root 用户身份运行以下命令：

```
~]# 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，以 root 用户身份运行以下命令：

```
~]# 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，以 root 用户身份运行以下命令：

```
~]# 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 区域，以丢弃来自测试 IP 集合中列出的所有条目的所有数据包，以 root 用户身份运行以下命令：

```
~]# firewall-cmd --permanent --zone=drop --add-source=ipset:test
success
```

-

源中的 **ipset**: 前缀显示 **firewalld** 表明源是 IP 集, 而不是 IP 地址或地址范围。

只有 IP 集的创建和删除仅限于永久环境, 所有其他 IP 设置选项也可以在运行时环境中使用, 而无需 **--permanent** 选项。

5.12.2. 为 IP 集合配置自定义服务

要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.

使用以 **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 IP 集:

```
~]# 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.

激活更改需要重新载入 **firewalld**:

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

这会在不丢失状态信息的情况下重新加载防火墙（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。此服务需要在 `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` 文件添加永久直接规则。

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/` 中的各种 XML 文件中。请注意，`/etc/sysconfig/iptables` 文件不存在，因为 Red Hat Enterprise Linux 中默认安装了 `firewalld`。
- 使用 `iptables` 服务时，每次更改都意味着清除所有旧规则并从 `/etc/sysconfig/iptables` 中读取所有新规则，而 `firewalld` 不会重新创建所有规则。仅应用差异。因此，`firewalld` 可以在不丢失现有连接的情况下在运行时更改设置。

两者都使用 `iptables` 工具与内核数据包过滤器进行通信。

要使用 `iptables` 和 `ip6tables` 服务而不是 `firewalld`，请以 `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，以 **root** 用户身份输入以下命令：

```
~]# systemctl start iptables
~]# systemctl start ip6tables
```

要在每次系统启动时启用服务，请输入以下命令：

```
~]# systemctl enable iptables
~]# systemctl enable ip6tables
```

ipset 实用程序用于管理 Linux 内核中的 IP 集。IP 集合是用于存储 IP 地址、端口号、IP 和 MAC 地址对或 IP 地址和端口对的框架。集合的索引方式可以非常快，即使在集合非常大时也可以对集合进行非常快速的匹配。IP 集实现了更简单、更易于管理的配置，在使用 **iptables** 时具有性能优势。**iptables** 匹配和目标，引用集合来保护内核中给定的集合。当存在指向这个集合的单个参考时，无法销毁集合。

使用 **ipset** 可让 **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
```

这个集合创建如下：

```
~]# 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，这个集合在 **iptables** 命令中被引用，如下所示：

```
~]# iptables -A INPUT -m set --set my-block-set src -j DROP
```

如果在配置时节省一次，则使用集合多次。如果集合包含许多条目，则会节省处理时间。

5.14. 使用直接接口

可以通过将 `--direct` 选项与 `firewall-cmd` 工具一起使用，在运行时添加和删除链。这里有几个示例。如需更多信息，请参阅 `firewall-cmd(1) man page`。

如果您不十分熟悉 `iptables`，因为您可能意外导致防火墙中的破坏，则使用直接接口很危险。

直接接口模式供服务或应用程序在运行时添加特定的防火墙规则。可以通过使用 `firewall-cmd --permanent --direct` 命令或修改 `/etc/firewalld/direct.xml` 添加 `--permanent` 选项，使规则永久存在。有关 `/etc/firewalld/direct.xml` 文件的详情，请查看 `man firewalld.direct (5)`。

5.14.1. 使用直接接口添加规则

要在 “`IN_public_allow`” chain 中添加规则，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`” chain 中删除规则，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，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。它不会列出通过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

“借助丰富的语言语法”，可以比直接接口方法更轻松地理解复杂的防火墙规则。此外，这些设置可以永久保留。语言使用带值的关键字和值，它是 `iptables` 规则的抽象表示。可以使用此语言配置区域；仍然支持当前的配置方法。

5.15.1. Rich Language 命令格式化

本节中的所有命令都需要以 `root` 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 `zone zone` 添加丰富的语言规则。这个选项可以多次指定。如果省略 区域，则使用默认区域。如果提供了超时，规则或规则仅在指定的时间内保持有效，之后将自动删除。时间值后可以跟 `s`（秒）、`m`（分钟）或 `h`（小时）以指定时间单位。默认值为 秒。

删除规则：

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

这将删除 `zone` 区域的语言规则。这个选项可以多次指定。如果省略 区域，则使用默认区域。

检查是否存在规则：

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

这将返回是否为 `zone` 区域添加了富语言规则规则。如果已启用，命令将打印 `yes`，退出状态为 `0`。否则，将不会打印退出状态 `1`。如果省略 区域，则使用默认区域。

有关区配置文件中使用的丰富语言表示法的详情，请参考 `firewalld.zone(5) man page`。

5.15.2. 了解 Rich 规则结构

富规则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

文件中的富规则的结构使用 **NOT** 关键字颠倒源和目标地址命令的含义，但命令行使用 **invert="true"** 选项。

规则与特定区域关联。一个区域可以有多个规则。如果某些规则交互或冲突，则将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

产品线

如果提供了规则系列（**ipv4** 或 **ipv6**），它会分别将规则限制为 **IPv4** 或 **IPv6**。如果未提供规则系列，则会为 **IPv4** 和 **IPv 6** 添加规则。如果在规则中使用源或目标地址，则需要提供规则系列。端口转发也是如此。

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** 地址或具有 **IPv4** 或 **IPv 6** 掩码的网络 **IP** 地址。对于 **IPv4**，掩码可以是网络掩码或者普通数字。对于 **IPv6**，掩码为普通数字。不支持使用主机名。通过添加 **not** 关键字，可以反转源地址命令的作用；除提供的地址匹配之外。

如果没有为该规则指定系列，则可为 **IPv4** 和 **IPv 6** 添加 **MAC** 地址以及类型为 **hash:mac** 的 **IP** 集。其他 **IP** 集需要与规则的 **family** 设置匹配。

destination

通过指定目标地址，可以将目标限制为目标地址。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，而且并非所有元素都无法使用目标地址。这取决于在服务条目中使用目标地址。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。要获取预定义的服务列表，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，并将导致错误。在内部使用目标地址的服务大部分是使用多播的服务。该命令采用以下格式：

```
service name=service_name
```

port

端口 元素可以是单个端口号或端口范围，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

协议值可以是协议 **ID** 号或协议名称。有关允许的协议条目，请查看 **/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。要获得支持的 **ICMP** 类型列表，请输入以下命令：

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

这里不允许指定操作。**ICMP-block** 在 内部使用操作 **reject**。该命令采用以下格式：

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


masquerade

在规则中打开 IP 伪装。可以提供源地址来限制伪装到此区域，但不能限制为目标地址。这里不允许指定操作。

forward-port

通过指定为 **tcp** or **udp** 的协议将数据包从本地端口转发到其他端口、到其他计算机，或者转发到另一台计算机上的其他端口。端口和 至端口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。这里不允许指定操作。**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 - 用于连接尝试来源的端口。若要匹配当前计算机上的端口，可使用 **port** 元素。**source-port** 元素可以是单个端口号或端口范围（如 5060-5062），后接协议为 **tcp** or **udp**。该命令采用以下格式：

```
source-port port=number_or_range protocol=protocol
```

日志

log

使用内核日志记录记录新连接尝试记录规则，例如在 **syslog** 中。您可以定义一个前缀文本，作为前缀添加到日志消息中。日志级别可以是 **emerg**、**alert**、**crit**、**error**、**warning**、**notice**、**info** 或 **debug** 之一。日志的使用是可选的。可以按如下所示限制日志：

```
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，这意味着每天最多有一个日志条目。

Audit

Audit 提供了一种使用发送到 **auditd** 服务的审计记录进行日志记录的替代方式。审计类型可以是 **ACCEPT**、**REJECT** 或 **DROP**，但不在命令 审核 后指定，因为审计类型将自动从规则操作中收集。

Audit 没有自己的参数，但也可选择添加限制。使用审计是可选的。

操作

accept/reject/drop/mark

操作可以是接受、拒绝、丢弃或 标记之一。规则只能包含元素或来源。如果规则包含 元素，则将使用该操作处理与该元素匹配的新连接。如果规则包含来源，则源地址中的所有内容都将通过指定的操作来处理。

```
accept | reject [type=reject type] | drop | mark set="mark[/mask]"
```

使用 **accept** 时，将授予所有新连接尝试。拒绝后，他们将会被拒绝，其来源将收到拒绝消息。**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，不会发送任何信息到源。如果标记了所有数据包，则会使用给定标记和可选掩码标记 所有数据包。

5.15.4. 使用 Rich Rule Log 命令

可以通过 **Netfilter** 日志目标以及 **audit** 目标进行日志记录。“将新链添加到所有区域，其格式为区域_log”，其中 **zone** 是区域名称。这会在 拒绝 链前进行处理，以获得正确的排序。根据规则的操作，它们将规则或部分置于单独的链中，如下所示：

```
zone_log
zone_deny
zone_allow
```

“所有日志记录规则都将放置在区域_log 链中”，首先进行解析。所有 **reject** 和 **drop** 规则都将放置在“区域_deny”链中，后者将在日志链后解析。所有 接受 规则都将放置在“区域_allow”链中，该链将在拒绝 链后解析。如果规则包含 日志 并且也 拒绝 或 允许 操作，指定这些操作的规则部分将放置在匹配链中。

5.15.4.1. 使用 Rich 规则日志命令示例 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 日志命令示例 2

通过审计为协议 **FTP** 允许新的 **IPv4** 和 **IPv6** 连接，每分钟日志 1：

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

5.15.4.3. 使用 Rich Rule 日志命令示例 3

允许从地址 192.168.0.0/24 的新 IPv4 连接用于协议 TFTP，并使用 syslog 允许每分钟记录 1:

```
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 日志命令示例 4

协议 RADIUS 的 1:2:3:4:6:: 的新 IPv6 连接将被拒绝，并以每分钟 3 的速度记录。可以接受来自其他来源的新 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 日志命令示例 5

在端口 4011 上将从 1:2:3:4:6:: 收到的 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 日志命令示例 6

将允许来自此来源的所有连接的源地址列入白名单。

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

更多示例请查看 `firewalld.richlanguage(5)` 手册页。

5.16. 配置防火墙锁定

如果本地应用或服务以 `root` 身份运行（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，以 `root` 用户身份运行以下命令：

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

如果启用锁定，该命令将输出 `yes`，退出状态为 `0`。否则，将不会打印退出状态 `1`。

要启用锁定，以 `root` 用户身份输入以下命令：

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

要禁用锁定，以 `root` 用户身份使用以下命令：

```
~]# 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。“如果白名单中的某个命令条目以星号 `*`”结尾，则以该命令开头的所有命令行都将匹配。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，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命令，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命令，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'
```

要查询命令命令是否在白名单中，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**，该命令将输出 **yes**，退出状态为 **0**。否则，将不会打印退出状态 **1**。

要列出白名单中的所有安全上下文，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**，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**，以 **root** 用户身份输入以下命令：

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

要查询上下文上下文是否在白名单中，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**，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 **IDuid**，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 **IDuid**，以 **root** 用户身份输入以下命令：

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

要查询用户 **IDuid** 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，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**，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**，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:virttd_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` 工具程序启用所有命令，对于用户 ID 为 815 的名为 `user` 的用户：

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

此示例显示用户 ID 和用户名，但只需要一个选项。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，所有实用程序都放置在 `/usr/bin/` 目录中，并且 `/bin/` 目录已符号链接到 `/usr/bin/` 目录。换句话说，尽管以 root 身份运行时的 `firewall-cmd` 的路径可能解析为 `/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 身份运行的脚本已编写为使用 `/bin/firewall-cmd` 路径，那么除了通常用于非root用户的 `/usr/bin/firewall-cmd` 路径外，还必须将命令路径列入白名单。

命令 `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 `"*"`，包括参数的绝对命令必须匹配。

5.17. 为拒绝数据包配置日志记录

在 `firewalld` 中使用 `LogDenied` 选项时，可以为被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。要更改日志记录的设置，请编辑 `/etc/firewalld/firewalld.conf` 文件或使用命令行或 GUI 配置工具。

如果启用了 `LogDenied`，则会在 `INPUT`、`FORWARD` 和 `OUTPUT` 链中的拒绝和丢弃规则前添加日志规则，并在区域中最终拒绝和丢弃规则。此设置的可能值有：`all`、`单播`、`广播`、`多播` 和 `关闭`。默认设置为 `off`。使用 `单播`、`广播` 和 `多播` 设置时，`pkttypes` 匹配用于匹配链路层数据包类型。全部情况下，都会记录所有数据包。

要使用 `firewall-cmd` 列出实际的 `LogDenied` 设置，以 `root` 用户身份运行以下命令：

```
~]# firewall-cmd --get-log-denied
off
```

要更改 `LogDenied` 设置，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`。这时会出现 `LogDenied` 窗口。从菜单中选择新的 `LogDenied` 设置，再单击 `OK`。

5.18. 其它资源

以下信息来源提供了有关 `firewalld` 的其他资源。

5.18.1. 安装的文档

- `firewalld(1)` man page - 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)` man page - 描述 `firewalld` 命令行客户端的命令选项。

- ***firewall-config(1) 手册页 - 描述 firewall-config 工具的设置。***
- ***firewall-offline-cmd(1) man page - 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 区域配置的 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) man page - 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 - 哪些区域以及如何配置它们的一般说明。***
- ***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** 工具的指定后继设备。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 IPv 6 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(**nfttrace**) 中调试和追踪以及监控追踪事件 (**nft** 工具中)
- 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** 工具取代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。

要显示规则集更改的影响，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。

使用 FIREWALLD 或 NFTABLES 时

- **firewalld**：将 **firewalld** 实用程序用于简单的防火墙用例。实用程序易于使用，并涵盖这些情况下的典型用例。
- **nftables**：使用 **nftables** 实用程序设置复杂和性能关键的防火墙，如整个网络。



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具有主要优势：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 `*.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：

- 您可以使用与 `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

您可以通过传递至 **nft** 实用程序或直接执行脚本来运行 **nftables** 脚本。

先决条件

-

本节的步骤假设您在 `/etc/nftables/example_firewall.nft` 文件中存储了 **nftables** 脚本。

过程 6.1. 使用 **nft** 实用程序运行 **nftables** 脚本

-

要运行 **nftables** 脚本，请将其传递给 **nft** 实用程序，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** 程序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** 成功执行脚本，在脚本中错误地放置规则、缺少参数或其他问题都可能导致防火墙的行为不如预期。

其它资源

•

有关设置文件所有者的详情，请查看 **chown(1) man page**。

•

有关设置文件权限的详情，请查看 **chmod(1) man page**。

•

有关使用系统引导载入 **nftables** 规则的详情，请参考 [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都视为注释。

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，请使用命名的 **set** 或 **verdict** 映射。

只有一个值的变量

以下示例定义了一个名为 **INET_DEV** 的变量，其值为 **enp1s0**：

```
define INET_DEV = enp1s0
```

您可以通过在变量名称后写入 **\$** 符号来使用脚本中的变量：

```
...
add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept
...
```

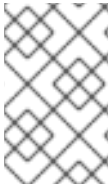
包含匿名集合的变量

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 }
```

您可以通过在变量名称后写入 `$` 符号来使用脚本中的变量：

```
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。

其它资源

- 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 `ver` 字典映射的详情请参考 [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

借助 nftables 脚本环境，管理员可以使用 `include` 语句包含 其他脚本。

如果您只指定没有绝对路径或相对路径的文件名，nftables 包括默认搜索路径中的文件，该路径设置为 Red Hat Enterprise Linux 上的 `/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. 包含目录中的所有 *.nft 文件

包含以 `*.nft` 结尾且存储在 `/etc/nftables/rulesets/` 目录中的所有文件：

```
include "/etc/nftables/rulesets/*.nft"
```

请注意，`include` 语句不匹配以点开头的文件。

其它资源

- 详情请查看 `nft(8)man page` 中的 `包含文件` 部分。

6.1.6. 系统引导时自动载入 `nftables` 规则

`nftables systemd` 服务加载 `/etc/sysconfig/nftables.conf` 文件中包含的防火墙脚本。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 `*.nft` 脚本，请取消为这些脚本的 `include` 语句添加注释。
 - 如果您从头开始编写脚本，请添加 `include` 语句以包含这些脚本。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` 服务在不重启系统的情况下载入防火墙规则：

```
# systemctl start nftables
```

3. 启用 `nftables` 服务。

```
# systemctl enable nftables
```

其它资源

- 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

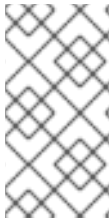
本节介绍如何显示 `nftables` 规则集以及如何管理它。

6.2.1. 显示 `nftables` 规则集

`nftables` 的规则集合包含表、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、集合和其他对象的集合的命名空间。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`**：仅与 IPv6 数据包匹配。

- **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
```

其它资源

- 有关地址系列的详情, 请查看 **nft(8)man page** 中的地址系列部分。
- 有关您可以在表中运行的其他操作的详细信息, 请参阅 **nft(8)man page** 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。

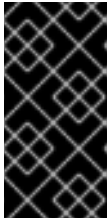
先决条件

- 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` 也解释大括号，因此使用大括号 `()` 加上大括号和其中的任何内容。

这个链过滤传入的数据包。`priority` 参数指定 `nftables` 进程使用相同 `hook` 值链的顺序。较低优先级的值优先于优先级更高的值。`policy` 参数设置此链中规则的默认操作。请注意，如果您远程登录服务器，并将默认策略设置为 `drop`，如果没有其他规则允许远程访问，您将立即断开连接。

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;
  }
}
```

其它资源

- 有关地址系列的详情，请查看 `nft(8)man page` 中的地址系列部分。
- 有关您可以在链中运行的其他操作的详细信息，请参阅 `nft(8)man page` 中的 `Chains` 部分。

6.2.4. 在 nftables 链末尾附加规则

本节介绍如何将规则附加到现有 nftables 链末尾。

先决条件

- 您要添加该规则的链已存在。

过程 6.6. 在 nftables 链末尾附加规则

1. 要添加新的规则，请使用 `nft add rule` 命令。例如，在 `example_table` 中的 `example_chain` 中添加一条规则，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。在该示例中，您可以使用 `ssh` 而不是端口号 22。请注意，服务名称会根据在 `/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
  }
}
```

其它资源

- 有关地址系列的详情，请查看 `nft(8)man page` 中的地址系列部分。
- 有关您可以在链中运行的其他操作的详细信息，请参阅 `nft(8)man page` 中的 `Rules` 部分。

6.2.5. 在 nftables 链的开头插入规则

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` 链的开头插入规则

- 要插入新规则，请使用 `nft` 插入规则命令。例如，要在 `example_table` 中将规则插入到 `example_chain`，该规则允许端口 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。在该示例中，您可以使用 `ssh` 而不是端口号 22。请注意，服务名称会根据在 `/etc/services` 文件中的条目解析为端口号。

- 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
  }
  ...
}
```

其它资源

- 有关地址系列的详情，请查看 `nft(8)man page` 中的地址系列部分。
- 有关您可以在链中运行的其他操作的详细信息，请参阅 `nft(8)man page` 中的 `Rules` 部分。

6.2.6. 在 `nftables` 链的特定位置插入规则

本节介绍如何在 `nftables` 链中现有规则前后插入规则。这样，您可以将新规则置于正确的位置。

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` 可显示句柄。您需要此信息才能在后续步骤中定位新规则。

2.

在 `example_table` 中的 `example_chain` 链中插入新规则：

•

要在处理 3 前插入允许 `port636` 上的 TCP 流量的规则，请输入：

```
# nft insert rule inet example_table example_chain position 3 tcp dport 636 accept
```

•

要在句柄 3 后添加允许端口 80 上的 TCP 流量的规则，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
  }
}
```

其它资源

- 有关地址系列的详情，请查看 `nft(8)man page` 中的地址系列部分。
- 有关您可以在链中运行的其他操作的详细信息，请参阅 `nft(8)man page` 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 类型之一来更改数据包的源 IP 地址。例如，互联网服务提供商不路由专用 IP 范围，如 `10.0.0.0/8`。如果您在网络中使用专用 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源 IP 地址映射到公共 IP 地址。

伪装和 SNAT 都非常相似。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 地址设置为指定的 IP 地址，且不会动态查找传出接口的 IP 地址。因此，SNAT 比伪装快。如果传出接口使用固定 IP 地址，请使用 SNAT。

目标 NAT (DNAT)

使用此 NAT 类型将传入的流量路由到不同的主机。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 规则以将进入的流量重定向到此服务器。

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，如果服务在其标准端口的不同端口上运行，您可以将从标准端口传入的流量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，如果接口被分配了新的 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.

在表中添加 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

即使您没有向抢占链添加规则，nftables 框架也要求此链与传入数据包回复匹配。

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.

在 **postrouting** 链中添加与 **ens3** 接口上的传出数据包匹配的规则：

```
# nft add rule nat postrouting oifname "ens3" masquerade
```

6.3.3. 使用 nftables 配置源 NAT

在路由器中，源NAT(SNAT)可让您将通过接口发送的数据包IP 改为特定的 IP 地址。

以下流程描述了如何将通过 **ens3** 接口离开路由器的数据包源 IP 替换为 **192.0.2.1**。

过程 6.10. 使用 nftables 配置源 NAT

1.

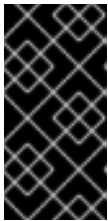
创建一个表：

```
# nft add table nat
```

2.

在表中添加 **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

即使您确实向 **postrouting** 链中添加规则，**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.

在 **postrouting** 链中添加一条规则，通过 **ens3** 将传出数据包的源 IP 替换为 **192.0.2.1**：

```
# nft add rule nat postrouting oifname "ens3" snat to 192.0.2.1
```

其它资源



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 允许您将路由器中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的端口 80 和 443 的传入流量重定向到使用 192.0.2.1 IP 地址的主机。

过程 6.11. 使用 nftables 配置目标 NAT

1.

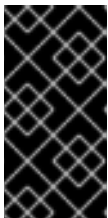
创建一个表：

```
# nft add table nat
```

2.

在表中添加 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.

在 prerouting 链中添加一条规则，将发送到端口 80 和 443 的 ens3 接口上传入的流量重定向到具有 192.0.2.1 IP 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，请添加一个伪装规则：

```
# 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
```

其它资源

•

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的端口 **22** 的传入和转发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 chain**：

```
# 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.

在 **prerouting** 链中添加一条规则，将端口 **22** 上传入的流量重定向到端口 **2222**：

■

```
# nft add rule nat prerouting tcp dport 22 redirect to 2222
```

其它资源

- 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，比如 { 22, 80, 443 }，它们直接在规则中使用。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，使用命名的集合，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，在 **example_table** 中的 **example_chain** 中添加允许传入流量到端口 22、80 和 443 的规则：

```
# 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 **mutable** 命名集。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`。
- `inet_proto`，用于包含 Internet 协议类型列表的集合，如 `tcp`。
- 包含互联网服务列表集合的 `inet_service`，如 `ssh`。
- 包含数据包标记列表的集合标记。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 地址创建了一个集合：

- 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，该规则将丢弃来自 `example_set` 中的 IPv4 地址的所有数据包。

```
# nft add rule inet example_table example_chain ip saddr @example_set drop
```

因为 `example_set` 仍然为空，因此该规则目前无效。

3. 在 `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，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)man page` 中的 **Sets** 部分。

6.5. 在 NFTABLES 命令中使用 VERDICT 映射

平均字典映射（也称为字典）使 nft 通过映射匹配到操作条件来基于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 }` 声明。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，如果要更改映射，则必须替换规则。对于动态解决方案，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.

在 `tcp_packets` 中添加计算此链中流量的规则：

```
# 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.

添加一条计算此链中流量的规则 `toudp_packets`：

```
# nft add rule inet example_table udp_packets counter
```

6.

为传入的流量创建一个链。例如，在 `example_table` 中创建一个名为 `incoming_traffic` 的链，用于过滤传入的流量：


```
# nft add chain inet example_table incoming_traffic { type filter hook input priority 0 \; }
```

7.

将带有匿名映射的规则添加到 **incoming_traffic** :

```
# 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，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** 链中的计数器同时显示收到的数据包数和字节数。

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。匿名映射的另一个优势在于，您可以更新命名映射而不替换使用它的规则。

在创建命名映射时，您必须指定元素类型：

•

ipv4_addr 用于匹配部分包含 IPv4 地址的映射，如 192.0.2.1。

- *ipv6_addr 用于匹配部分包含 IPv6 地址的映射，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。*
- *inet_proto 用于匹配部分包含 Internet 协议类型（如 tcp）的映射。*
- *匹配部分包含互联网服务名称端口号（如 ssh 或 22）的映射的 inet_service。*
- *为匹配部分包含数据包标记的映射添加标记。数据包标记可以是任意正 32 位整数值（0 到 2147483647）。*
- *映射的计数器，其匹配部分包含计数器值。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配额。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情况，同时 IP 地址和操作会动态存储在映射中。此流程还描述了如何从映射中添加和删除条目。

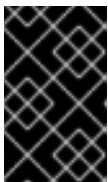
过程 6.16. 在 nftables 中使用命名映射

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，以下命令为 `example_table` 中的 `example_chain` 添加了一个规则，它把操作应用到 `example_map` 中定义的 IPv4 地址：

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

5.

在 `example_map` 中添加 IPv4 地址和对应的操作：

```
# 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。根据上述规则，防火墙接受来自 192.0.2.1 的数据包并丢弃来自 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 `ver` 字典映射的详情，请查看 `nft(8)man page` 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端口转发规则，该规则将在防火墙的端口 80 和 443 上转发传入的数据包到 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

这部分论述了如何在端口 8022 上将传入的 IPv4 数据包转发到本地系统上的端口 22 的示例。

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. 在表中添加 `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. 在 `prerouting` 链中添加一条规则，将端口 8022 上传入的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。

这个步骤描述了如何将本地端口 443 中传入的 IPv4 数据包转发到 IP 地址为 192.0.2.1 的远程系统上的相同端口号。

先决条件

- 您以 root 用户身份登录应该转发数据包。

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. 在表中添加 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. 在 prerouting chain 中添加一条规则，将端口 443 上传入的数据包重新指向 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. 为 postrouting 链添加一条规则伪装出站流量：

```
# nft add rule 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` 来限制连接数量或阻止建立给定连接量的 IP 地址，以防止它们使用过多的系统资源。

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` 中的基本 `example_chain` 存在。

过程 6.19. 使用 nftables 限制连接数量

1.

添加一条规则，该规则只允许从 IPv4 地址同时连接到 SSH 端口(22)，并从同一 IP 拒绝所有后续连接：

```
# 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` 条目显示目前与该规则匹配的地址。在本例中，`elements` 列出已连接到 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设置。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 地址系列创建过滤器表：

```
# nft add table ip filter
```

2.

在过滤器表中添加输入链：

```
# nft add chain ip filter input { type filter hook input priority 0 \; }
```

3.

在过滤器表中添加名为 **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 分钟后自动删除条目。

4.

添加一条规则，该规则会在一分钟内试图建立十个新的 TCP 连接的主机源 IP 地址添加到 **denylist** 集合中：

```
# 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，丢弃来自 **denylist** 集中 IP 地址的所有连接：

```
# nft add rule ip filter input ip saddr @denylist drop
```

6.7.3. 其它资源

- 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，并在数据包匹配时提供不同的选项。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. 在链中添加使用 **counter** 参数的新规则。以下示例添加一个带有计数器的规则，它允许端口 22 上的 TCP 流量，并计算与这个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

- 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
  }
}
```

- 通过替换规则而不是使用 **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
```

- 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，使管理员可以显示与规则匹配的数据包。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.

通过替换规则而不是使用 `meta nftrace set 1` 参数来添加追踪功能。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` 命令显示追踪。以下示例过滤命令的输出，仅显示包含 `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**

根据启用追踪的规则数量以及匹配的流量数量，`nft monitor` 命令可以显示大量输出。使用 `grep` 或其他实用程序过滤输出。

第 7 章 系统审核

Linux Audit 系统提供了一种方式来跟踪系统中的安全相关信息。根据预配置的规则，审计会生成日志条目，以记录有关系统上发生事件的尽可能多的信息。对于关键任务环境而言，此信息对于确定安全策略的违反者及其执行的操作至关重要。Audit 不会为您的系统提供额外的安全性，而是可用于发现系统上使用的安全策略违规。通过 SELinux 等其他安全措施可以进一步阻止这些冲突。

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：

- 事件的日期和时间、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相关联。
- 对 Audit 配置的所有修改，并尝试访问 Audit 日志文件。
- 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更改，如 `/etc/passwd`。
- 尝试从系统导入或导出信息。
- 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。

使用审计系统也是许多安全相关认证的一项要求。审计旨在满足或超过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。*
- *通过红帽企业 Linux 5 上的 LSPP/CAPP/RSBAC/EAL4+ 认证。*
- *红帽企业 Linux 6 上经过操作系统保护配置文件/评估保证级别 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访问、修改、执行或文件属性是否已更改。例如，这可用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` 和其他时间相关系统调用来跟踪系统时间的更改。

记录用户运行的命令

审计可以跟踪文件是否已执行，因此可以定义规则以记录特定命令的每次执行。例如，可以为 `/bin` 目录中的每个可执行文件定义规则。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转换索引节点路径的文件访问之外，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，或者在规则调用后替换了文件。这允许规则在升级程序可执行文件或甚至安装之前继续运行。

记录安全事件

pam_faillock 身份验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，并提供试图登录的用户的附加信息。

搜索事件

Audit 提供 **ausearch** 实用程序，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。

运行摘要报告

aureport 实用程序可用于生成记录的事件的日常报告等。然后，系统管理员可以分析这些报告，并进一步调查可疑活动。

监控网络访问

iptables 和 **ebtables** 实用程序可以配置为触发审计事件，使系统管理员能够监控网络访问。



注意

系统性能可能会受到影响，具体取决于审计收集的信息数量。

7.1. AUDIT 系统架构

Audit 系统由两个主要部分组成：用户空间应用程序和实用程序，以及内核端系统调用处理。内核组件从用户空间应用程序接收系统调用，并通过以下过滤器之一对其进行过滤：用户、任务、**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，这些过滤器根据 **Audit** 规则配置将其发送到 **Audit** 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 **Audit** 用户空间实用程序与 **Audit** 守护进程、内核审计组件或 **Audit** 日志文件交互：

- **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以进行进一步处理。此守护进程的目的是提供一种插件机制，让实时分析程序能够与审计事件交互。
- **auditctl** - **Audit** 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许多设置和参数。
- 剩余的 **Audit** 实用程序将 **Audit** 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** 实用程序生成所有记录事件的报告。

7.2. 安装 **AUDIT** 软件包

要使用 **Audit** 系统，必须在您的系统中安装 **audit** 软件包。在 **Red Hat Enterprise Linux 7** 中默认安装 **audit** 软件包（**audit** 和 **audit-libs**）。如果您还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 **Audit** 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。此文件由用于修改 **Audit** 守护进程行为的配置参数组成。**hash** 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 **auditd** 配置应当适合大多数环境。但是，如果您的环境必须满足严格的安全策略，建议在 **/etc/audit/auditd.conf** 文件中对 **Audit** 守护进程配置进行以下设置：

log_file

包含 **Audit** 日志文件的目录（通常为 **/var/log/audit/**）应位于单独的挂载点。这可以防止其他进程消耗此目录中的空间，并为 **Audit** 守护进程提供准确检测剩余空间。

max_log_file

指定单个审计日志文件的最大大小，必须设置该文件才能充分利用保存审计日志文件的分区上的可用空间。

max_log_file_action

决定在 **max_log_file** 中设置的限制后执行的操作，应设置为 **keep_logs**，以防止覆盖 Audit 日志文件。

space_left

指定在 **space_left_action** 参数中设置操作的磁盘上保留的可用空间量。必须设置一个数字，让管理员有足够的时间来响应和释放磁盘空间。**space_left** 值取决于审计日志文件的生成速度。

space_left_action

建议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** 参数中设置操作的绝对最小可用空间量，必须设置为保留足够空间以记录管理员执行的操作的值。

admin_space_left_action

应将 设置为 **single** 以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

指定在保存 Audit 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 Audit 无法记录事件时，这可确保系统以单用户模式关闭或运行。

disk_error_action

指定在包含 Audit 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**、**单一** 或 **停止**，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** 参数相结合，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 **audit** 值。您只能将 **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**

从 **/etc/audit/auditd.conf** 文件中重新加载 **auditd** 的配置。

rotate

轮转 **/var/log/audit/** 目录中的日志文件。

resume

在之前暂停后恢复审计事件记录，例如，当保存 Audit 日志文件的磁盘分区中没有足够的可用空间时。

condrestart 或 try-restart

只有在 **auditd** 已在运行时才重新启动。

status

显示 **auditd** 的运行状态。

7.5. 定义审计规则

Audit 系统在一组规则上运行，这些规则定义要在日志文件中捕获的内容。可以指定以下审计规则类型：

控制规则

允许修改 **Audit** 系统的行为及其部分配置。

文件系统规则

也称为文件监视，允许审核特定文件或目录的访问权限。

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** 实用程序。请注意，这些规则在重新启动后不会保留。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 Audit 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 Audit 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。此外，需要 `CAP_AUDIT_CONTROL` 来设置审计服务，并且需要 `CAP_AUDIT_WRITE` 来记录用户消息。

定义控制规则

以下是一些控制规则，允许您修改审计系统的行为：

-b

在内核中设置最大现有审计缓冲量，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
```

如果出现严重错误，以上配置会触发内核 `panic`。

-e

启用或禁用 Audit 系统或锁定其配置，例如：

```
~]# auditctl -e 2
```

以上命令将锁定 Audit 配置。

-r

设置每秒生成的消息率，例如：

```
~]# auditctl -r 0
```

以上配置不会设置所生成消息的速率限制。

-s

报告 Audit 系统的状态，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** 是被审计的文件或目录。

- 权限是记录的权限：
 - **r** - 对文件或目录的读取访问权限。
 - **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权限。
 - **a** - 更改文件或目录的属性。
- **key_name** 是一个可选字符串，可帮助您识别生成了特定日志条目的规则或一组规则。

例 7.1. 文件系统规则

要定义一条规则，记录 **/etc/passwd** 文件的所有写入访问权限和每个属性更改，请执行以下命令：

```
~]# auditctl -w /etc/passwd -p wa -k passwd_changes
```

请注意，在 **-k** 选项后面的字符串是任意的。

要定义一条规则，记录 **/etc/selinux/** 目录中所有文件的写入访问和每个属性更改，请执行以下命令：

```
~]# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

要定义一个规则来记录 **/sbin/insmod** 命令的执行（在 Linux 内核中插入模块），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：

- **操作和 过滤**指定记录特定事件的时间。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：**task**、**exit**、**user** 和 **exclude**。有关这些过滤器的更多信息，请参阅第 7.1 节“**Audit 系统架构**”的开头。
- **system_call** 指定系统调用的名称。可以在 `/usr/include/asm/unistd_64.h` 文件中找到所有系统调用的列表。可将多个系统调用分组成一个规则，各自在其自己的 **-S** 选项后指定。
- **Field=value** 指定进一步修改规则以根据指定的体系结构、组 ID、进程 ID 和其他选项匹配的额外选项。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，可帮助您识别生成了特定日志条目的规则或一组规则。

例 7.2. 系统调用规则

要定义当程序每次使用 **adjtimex** 或 **settimeofday** 系统调用时创建日志条目的规则，且系统使用 64 位构架，请执行以下命令：

```
~]# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
```

要定义规则，在每次删除文件时创建一个日志条目，或者由 ID 为 1000 或更高版本的系统用户重命名，请执行以下命令：

```
~]# 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：

- **操作和 过滤**指定记录特定事件的时间。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：**task**、**exit**、**user** 和 **exclude**。有关这些过滤器的更多信息，请参阅第 7.1 节“**Audit 系统架构**”的开头。
- **system_call** 指定系统调用的名称。可以在 `/usr/include/asm/unistd_64.h` 文件中找到所有系统调用的列表。可将多个系统调用分组成一个规则，各自在其自己的 **-S** 选项后指定。
- **path_to_executable_file** 是被审计的可执行文件的绝对路径。
- **key_name** 是一个可选字符串，可帮助您识别生成了特定日志条目的规则或一组规则。

例 7.3. 可执行文件规则

要定义记录所有 `/bin/id` 程序执行的规则，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，或者使用 **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。**hash** 符号(**#**)后面的空行和文本将被忽略。

auditctl 命令也可用于使用 **-R** 选项从指定文件读取规则，例如：

```
~]# auditctl -R /usr/share/doc/audit/rules/30-stig.rules
```

定义控制规则

文件只能包含以下控制规则来修改审计系统的行为：**-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** 语法进行定义。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
```

预配置规则文件

在 `/usr/share/doc/audit/rules/` 目录中，`audit` 软件包根据各种认证标准提供一组预配置的规则文件：

- **30-NISPOm.rules** - 审计规则配置，这些配置符合国家工业安全计划操作手册"信息系统安全"一章中指定的要求。
- **30-PCI-dss-v31.rules** - 审计规则配置，满足支付卡行业数据安全标准(PCI DSS)v3.1 的要求。
- **30-STIG.rules** - 符合安全技术实施指南(STIG)要求的审计规则配置。

要使用这些配置文件，请创建原始 `/etc/audit/audit.rules` 文件的备份，并在 `/etc/audit/audit.rules` 文件上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

Audit 规则具有一个编号方案，允许对它们进行排序。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/` 目录中的规则，并将它们编译到 `audit.rules` 文件中。这个脚本会根据自然的排序顺序按特定顺序处理以 `.rules` 结尾的所有文件。这个目录中的文件被组织到组中，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 - 主要规则**
- **40 - 可选规则**

- 50 - 服务器特定规则
- 70 - 系统本地规则
- 90 - 结束 (不可变)

规则并非是一次性全部使用。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，复制规则 `10-base-config`、`30-stig`、`31-privileged` 和 `99-finalize`。

在 `/etc/audit/rules.d/` 目录中有规则后，使用 `--load` 指令运行 `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)man page`。

7.6. 了解 AUDIT 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 `audit.log` 文件存储在同一个目录中。

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，例如使用以下命令在 `Audit` 日志文件中创建新事件：

```
~]$ cat /etc/ssh/sshd_config
```

`audit.log` 文件中的此事件如下：

```
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=` 值对组成，值对由空格或逗号分开。对上述事件的详细分析如下：

第一次记录

`type=SYSCALL`

`type` 字段包含记录的类型。在本例中，`SYSCALL` 值指定此记录是由系统调用对内核触发的。

有关所有可能类型值及其解释列表，请参阅 [Audit Record Types](#)。

`msg=audit(1364481363.243:24287):`

`msg` 字段记录：

-

时间戳和记录的唯一 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生成的，则可以共享相同的时间戳和 ID。时间戳在 1970 年 1 月 1 日使用 Unix 时间格式 - 秒，自 00:00:00 UTC 起。

● 各种特定于事件 的名称= 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。使用 **ausearch** 命令搜索 Audit 记录时, 请使用 **-i** 或 **--interpret** 选项自动将十六进制值转换为其人类可读的等效值。**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** 可以与其 **/usr/include/asm/unistd_64.h** 文件中的人类可读等效值匹配。在本例中, **2** 是 **open** 系统调用。请注意, **ausyscall** 实用程序允许您将系统调用号转换为其人类可读的等效项。使用 **ausyscall --dump** 命令显示所有系统调用的列表及其编号。详情请查看 **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, 调用没有成功。

exit=-13

exit 字段包含一个值, 指定系统调用返回的退出代码。此值因不同的系统调用而异。您可以使用以下命令将值解读为其人类可读的等效值:

```
~]# ausearch --interpret --exit -13
```

请注意, 上例假定您的审计日志包含带有退出代码 **-13** 的事件。

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** 实用程序来解释。

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 是父进程的 PPID，如 **bash**。

pid=3538

pid 字段记录了流程 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，并在每次用户的身份更改时继承，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。使用以下命令可以解读用户 ID：**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，以便在相关 **PATH** 记录中捕获相对路径。这样，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。时间戳在 1970 年 1 月 1 日使用 Unix 时间格式 - 秒，自 00:00:00 UTC 起。

cwd="/home/user_name"

cwd 字段包含系统调用所在目录的路径。

第三个记录

type=PATH

在第三条记录中，**type** 字段值为 **PATH**。Audit 事件包含作为参数传递给系统调用的每个路径的 **PATH-type** 记录。在这个审计事件中，只有一个路径(/etc/ssh/sshd_config)作为参数。

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。这个数字基于零；值 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 索引节点编号关联的文件或目录：

```
~]# 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，以数字表示法编码，如 **st_mode** 字段中 **stat** 命令返回。如需更多信息，请参阅 **stat(2) man page**。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** 文件具有读取和写入权限。

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 字段记录了每个路径记录在给定系统调用上下文中的操作意图。

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。

第四个记录

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，该事件由对内核的系统调用触发。

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 Audit 日志解析器。文本解码到触发此审计事件的命令。使用 **ausearch** 命令搜索 Audit 记录时，请使用 **-i** 或 **--interpret** 选项自动将十六进制值转换为其人类可读的等效值。**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说明的列表，请参阅 [Audit Event 字段](#)。有关所有事件类型及其说明的列表，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 **auditd** 守护进程的成功启动。**ver** 字段显示已启动的 Audit 守护进程版本。

```
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用户 **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 AUDIT 日志文件

ausearch 实用程序允许您搜索 Audit 日志文件特定事件。默认情况下，**ausearch** 搜索 **/var/log/audit/audit.log** 文件。您可以使用 **ausearch** 选项 **-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相当于在字段类型和相同字段类型的多个实例之间使用 **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` 文件以查找失败的登录尝试，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_CHAUTHOK -m
DEL_GROUP -m CHGRP_ID -m ROLE_ASSIGN -m ROLE_REMOVE -i
```

要搜索特定用户执行的所有日志操作，请使用用户的登录 ID(auid)，使用以下命令：

```
~]# ausearch -ua 1000 -i
```

要搜索直到现在为止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 实用程序允许您针对 **Audit** 日志文件中记录的事件生成摘要和列表报告。默认情况下，会查询 `/var/log/audit/` 目录中的所有 `audit.log` 文件来创建报告。您可以使用 **aureport** 选项 **-if file_name** 命令指定针对执行报告的其他文件。

例 7.8. 使用 aureport to Generate Audit Reports

要为过去三天（除当前示例日除外）内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** 查询生成报告来搜索用户 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
```

要生成查询的所有 **Audit** 文件的报告以及包括的事件的时间范围，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 **Audit** 系统的更多信息，请参见以下源：

在线源

- **RHEL Audit 系统刷新：**

- **Linux Audit [文档项目页面](#) :**

安装的文档

audit 软件包提供的文档可以在 `/usr/share/doc/audit/` 目录中找到。

手动页面

- ***audispd.conf(5)***
- ***auditd.conf(5)***
- ***ausearch-expression(5)***
- ***audit.rules(7)***
- ***audispd(8)***
- ***auditctl(8)***
- ***auditd(8)***
- ***auleast(8)***
- ***auleastlog(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性和漏洞

合规审计是一个确定给定对象是否遵循合规性策略中指定的所有规则的过程。合规策略由安全专业人员定义，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使得需要自定义策略清单。

8.1. RHEL 中的配置合规工具

红帽企业 Linux 提供了一些工具，使您可以执行完全自动化的合规性审计。这些工具基于安全内容自动化协议(SCAP)标准，专为自动定制合规性策略而设计。

- **SCAP Workbench - scap-workbench** 图形实用程序旨在在单个本地或远程系统上执行配置和漏洞扫描。您还可以使用它根据这些扫描和评估来生成安全报告。
- **OpenSCAP - OpenSCAP** 库以及附带的 `oscap` 命令行实用程序，旨在在本地系统上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 系统提供了最新的安全策略集合。该指南包括一个实际强化建议目录，在适用的情况下与政府的要求相关联。该项目填补了一般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 本身不属于 SCAP 环境。

要在多个系统上远程执行自动合规审计，您可以将 OpenSCAP 解决方案用于红帽卫星。

其它资源

- **oscap(8) - oscap** 命令行实用程序的 man page 提供了可用选项的完整列表及其用法说明。
- [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性](#) - 亲身实践实验室，利用红帽企业 Linux 中包含的工具获取自动化安全合规性的初始经验，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获取更多详细信息。

- [红帽安全演示：使用 RHEL 安全技术进行自行定义](#) - 亲身实践实验室，了解如何在 RHEL 系统的所有级别使用您在红帽企业 Linux 中可用的关键安全技术（包括 OpenSCAP）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信息。
- `scap-workbench(8)` - SCAP Workbench 应用的手册页提供有关应用的基本信息和 SCAP 内容潜在来源的链接。
- `scap-security-guide(8)` - `scap-security-guide` 项目的 man page 提供了有关各种可用的 SCAP 安全配置集的更多文档。它还包含使用 OpenSCAP 实用程序提供的基准测试的示例。
- [《管理红帽卫星指南》](#) 中的安全合规性管理提供了有关将 OpenSCAP 与红帽卫星搭配使用的更多详细信息。

8.2. 漏洞扫描

8.2.1. 红帽安全公告 OVAL Feed

红帽企业 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并不强制实施。这使得组织能够构建一次安全策略（SCAP 内容），无论他们采用的是多少家安全供应商。

开放式漏洞评估语言(OVAL)是 SCAP 的基本和最旧组件。与其他工具和自定义脚本不同，OVAL 以声明性方式描述资源的必需状态。OVAL 代码绝不直接执行，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 也基于 XML。SCAP 标准定义多种文档格式。它们各自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。它在红帽客户门户上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、版本及其他因素之间存在差异，红帽产品安全严重性等级评级无法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 定义可以单独提供完整的软件包，并在红帽客户门户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顾问(RHSA)。由于 RHSA 可以包含多个漏洞的修复，每个漏洞都通过其通用漏洞和风险(CVE)名称单独列出，并在我们的公共错误数据库中有一个链接。

RHSA OVAL 定义旨在检查系统中安装的 RPM 软件包是否存在安全漏洞的版本。可以扩展这些定义以包括进一步检查，例如，查找软件包是否在易受攻击的配置中使用。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是否有安全漏洞，请参阅第 8.9 节“针对漏洞扫描容器和容器镜像”。

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“针对漏洞扫描容器和容器镜像”](#)

8.2.2. 扫描系统是否有漏洞

Theoscapy 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。此实用程序充当 **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. 扫描系统是否有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 检查漏洞。因此，结果 **"True"** 意味着系统存在安全漏洞，而 **"False"** 则表示扫描没有发现漏洞。在 **HTML** 报告中，这通过结果行的颜色进一步区分。

其它资源

- **oscap(8)** 的手册页面。
- [红帽 OVAL 定义列表](#)。

8.2.3. 扫描远程系统是否有漏洞

您还可以通过 SSH 协议使用 the `oscap-ssh` 工具通过 OpenSCAP 扫描程序检查远程系统是否有漏洞。

先决条件

- **openscap-scanner** 软件包安装在远程系统中。
- **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. 使用 **machine1** 主机名扫描远程系统，**SSH** 在端口 22 上运行，**joesec** 用户名查找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)man** 页面。
- [红帽 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扫描来符合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器，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性。

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，因为它符合受影响组件的红帽最佳实践。

SCAP 安全指南软件包提供符合 SCAP 1.2 和 SCAP 1.3 标准的内容。openscap 扫描程序实用程序与 SCAP 安全指南软件包中提供的 SCAP 1.2 和 SCAP 1.3 内容兼容。

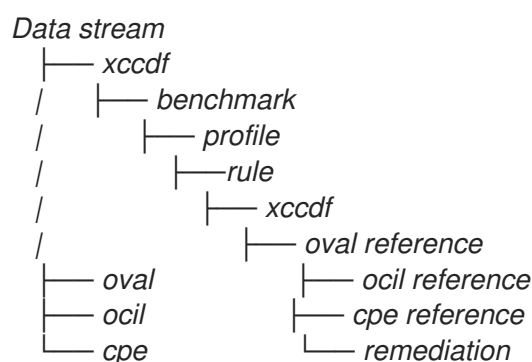


重要

执行配置合规扫描不能保证系统合规。

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集。数据流是包含定义、基准、配置集和个别规则的文件。每条规则都规定合规的适用性和要求。RHEL 7 提供多个配置集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



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（尾部）配置集来自定义某些规则，例如密码长度。有关定制配置集的更多信息，请参阅第 8.7.2 节“使用 SCAP Workbench 自定义安全配置集”



注意

要扫描容器或容器镜像的配置合规性，请参阅 [第 8.9 节“针对漏洞扫描容器和容器镜像”](#)

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集，每个规则可能会生成特定的结果。这是可能的结果列表，并简要解释了它们的含义。

表 8.1. OpenSCAP 扫描的可能结果

结果	解释
PASS	扫描没有发现与该规则有任何冲突。
失败	扫描发现与此规则冲突。
未检查	OpenSCAP 不对此规则执行自动评估。检查您的系统是否手动符合此规则。
不适用	此规则不适用于当前配置。
未选择	这个规则不是配置集的一部分。OpenSCAP 不评估此规则，也不会 在结果中显示这些规则。
错误	扫描会出现错误。如需其他信息，您可以使用 -verbose DEVEL 选项输入 theoscap-scanner 命令。 考虑打开错误报告。
Unknown	扫描遇到了意外情况。如需其他信息，您可以使用 -verbose DEVEL 选项输入 theoscap-scanner 命令。 考虑打开错误报告。

8.3.3. 查看配置合规性配置集

在决定使用配置集进行扫描或修复前，您可以使用 **theoscap info** 子命令列出配置文件并检查其详细描述。

先决条件

- 已安装 **openscap-scanner** 和 **scap-security-guide** 软件包。

流程

1.

使用 **SCAP** 安全指南项目提供的配置合规配置集列出所有可用的文件：

```
~]$ 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 **theoscap 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 文件的名称中通过 **-ds** 字符串来指示。在 **Profiles** 部分，您可以找到可用配置集及其 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集，并显示所选配置集的附加详情。为此，可使用 **oscap info** 及 **--profile** 选项，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 是：**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 Web 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《红帽企业 Linux 7 安全配置指南》的右上角选择所需配置集，您可以看到相关命令中已包含的 ID，供后续评估使用。

其它资源

•

scap-security-guide(8)man page 还包含配置集列表。

8.3.4. 使用特定基行评估配置合规性

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集 ID。要查找 ID，请参阅第 8.3.3 节“查看配置合规性配置集”。

流程

1. 评估系统与所选配置集的合规性，并将扫描结果保存到 **report.html** HTML 文件中，例如：

```
~]$ sudo oscap xccdf eval --report report.html --profile osp
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

2. 可选：扫描带有 **machine1** 主机名的远程系统，在端口 22 中运行 SSH，为漏洞扫描 **joesec** 用户名，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 osp
/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。
- 红帽企业 Linux 7 安全配置指南与 **scap-security-guide-doc** 软件包一起安装。

8.4. 使用特定基线将系统修复为强制

使用此流程修复 RHEL 7 系统，使其与特定基准一致。这个示例为常规目的操作系统(OSPP)使用保护配置集。



警告

如果不小心使用，则启用 **Remediate** 选项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方法来恢复安全补救所做的更改。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. 评估系统与 **OSPP** 配置集的合规性，并将扫描结果保存在 **theospp_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)man page**

8.5. 使用 SSG ANSIBLE PLAYBOOK，使用特定基线将系统修复为强制

通过" SCAP 安全指南 "项目中的 Ansible playbook 文件, 使用此流程通过特定基准修复您的系统。这个示例为常规目的操作系统(OSPP)使用保护配置集。



警告

如果不小心使用, 则启用 **Remediate** 选项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方法来恢复安全补救所做的更改。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. 评估系统与 **OSPP** 配置集的合规性, 并将扫描结果保存在 **theospp_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)man page**
- **[Ansible 文档](#)**

8.6. 创建修复 ANSIBLE PLAYBOOK 以使用特定基本行的系统

使用此流程创建仅包含将系统与特定基准保持一致所需的修复的 **Ansible playbook**。这个示例为常规目的操作系统(OSPP)使用保护配置集。通过这个过程，您可以创建一个不满足已满足要求的较小的 **playbook**。按照以下步骤，您不会以任何方式修改您的系统，您只需为后续应用程序准备文件。

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 osppe --results osppe-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 Ansible playbook：**

```
~]# oscap xccdf generate fix --fix-type ansible --profile osppe --output osppe-remediations.yml
osppe-results.xml
```

3. **Theosppe-remediations.yml 文件包含步骤 1 中执行扫描期间失败的规则的 Ansible 修复。查看此生成的文件后，您可以使用 `ansible-playbook osppe-remediations.yml` 命令应用该文件。**

验证

1. **在您选择的文本编辑器中，查看 the osppe-remediations.yml 文件包含在第 1 步执行的扫描中失败的规则。**

其它资源

- **scap-security-guide(8)和 oscap(8)man page**
- [Ansible 文档](#)

8.7. 使用 SCAP WORKBENCH 使用自定义配置集扫描系统

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** 命令行实用程序相比，**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 扫描和修复系统

要根据所选安全策略评估您的系统，请使用以下步骤。

先决条件

- **scap-workbench** 软件包安装在您的系统中。

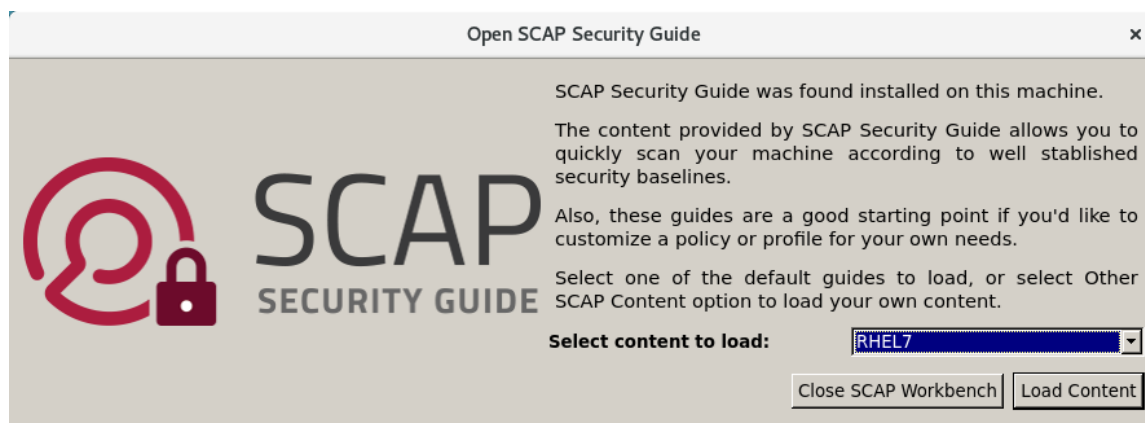
流程

1. 要从 **GNOME Classic** 桌面环境运行 **SCAP Workbench**，请按 **Super** 键进入 "活动概览"，键入 **scap-workbench**，然后按 **Enter** 键。或者，使用：

```
~]$ scap-workbench &
```


2. 使用以下任一选项选择安全策略：

- 在起始窗口中加载内容按钮
- 从 **SCAP 安全指南** 打开内容
- 在 **File** 菜单 中打开 **Other Content**，搜索相应的 **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** 会尝试根据策略应用的安全规则更改系统配置。这个过程尝试修复系统扫描过程中失败的相关检查。

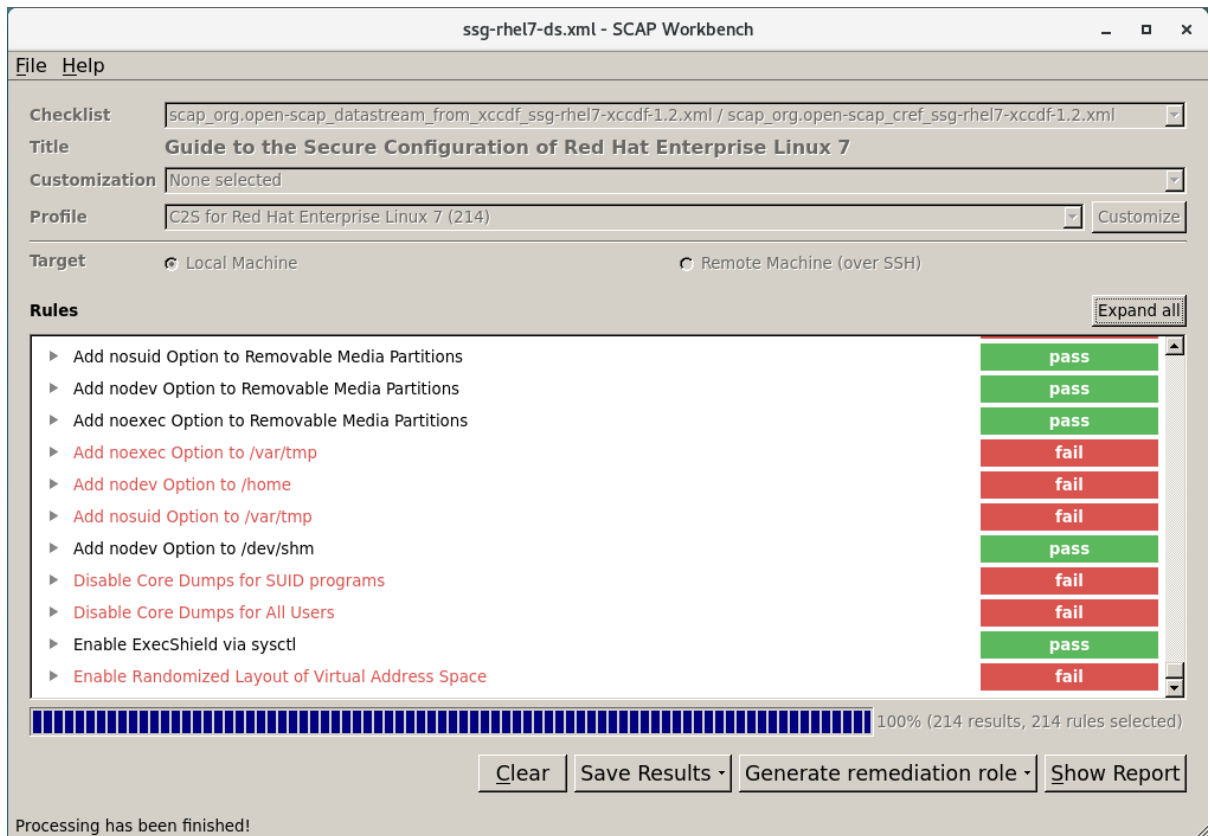


警告

如果不小心使用，则启用 **Remediate** 选项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方法来恢复安全补救所做的更改。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.

单击扫描按钮，使用所选配置集扫描您的系统。



5.

要以 XCCDF、ARF 或 HTML 文件的形式保存扫描结果，请点击 **Save Results** combo 框。选择 **HTML Report** 选项，以人类可读格式生成扫描报告。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 自定义安全配置集

您可以通过更改特定规则中的参数（如最小密码长度）、删除您涵盖的规则不同方式，并选择附加规则来自定义安全配置集，以实施内部策略。您无法通过自定义配置集来定义新规则。

以下步骤演示了如何使用 SCAP Workbench 自定义（定制）配置集。您还可以保存用于 **oscap** 命令行实用程序的定制配置集。

流程

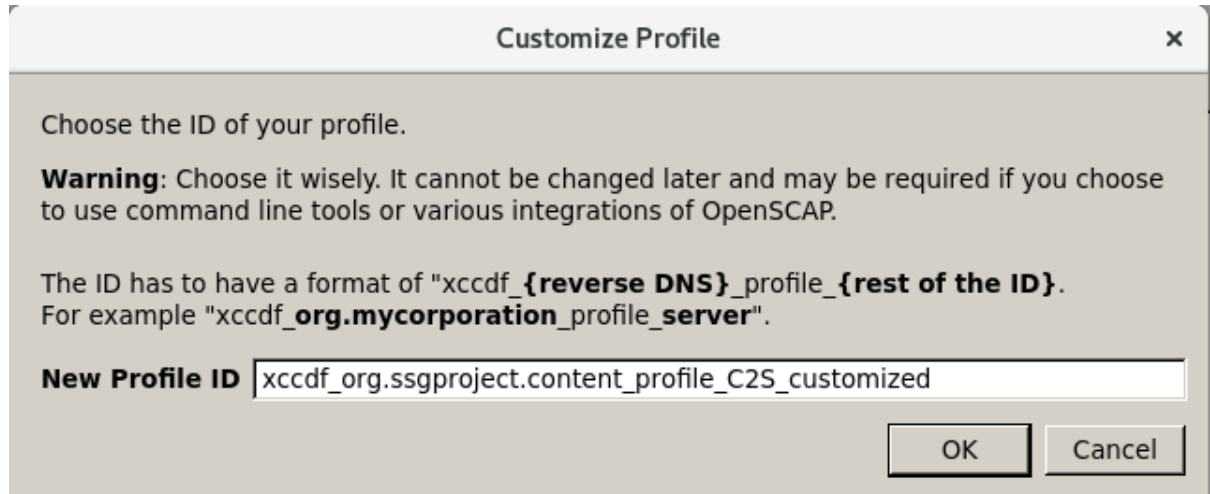
1.

Run SCAP Workbench，然后使用 SCAP 安全指南中的 **Open content** 或 **File** 菜单中的 **Open Content** 来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集，请单击 **Customize** 按钮。

这会打开新的 **Customization** 窗口，允许您修改当前选定的 **XCCDF** 配置集，而不更改原始 **XCCDF** 文件。选择新的配置文件 ID。

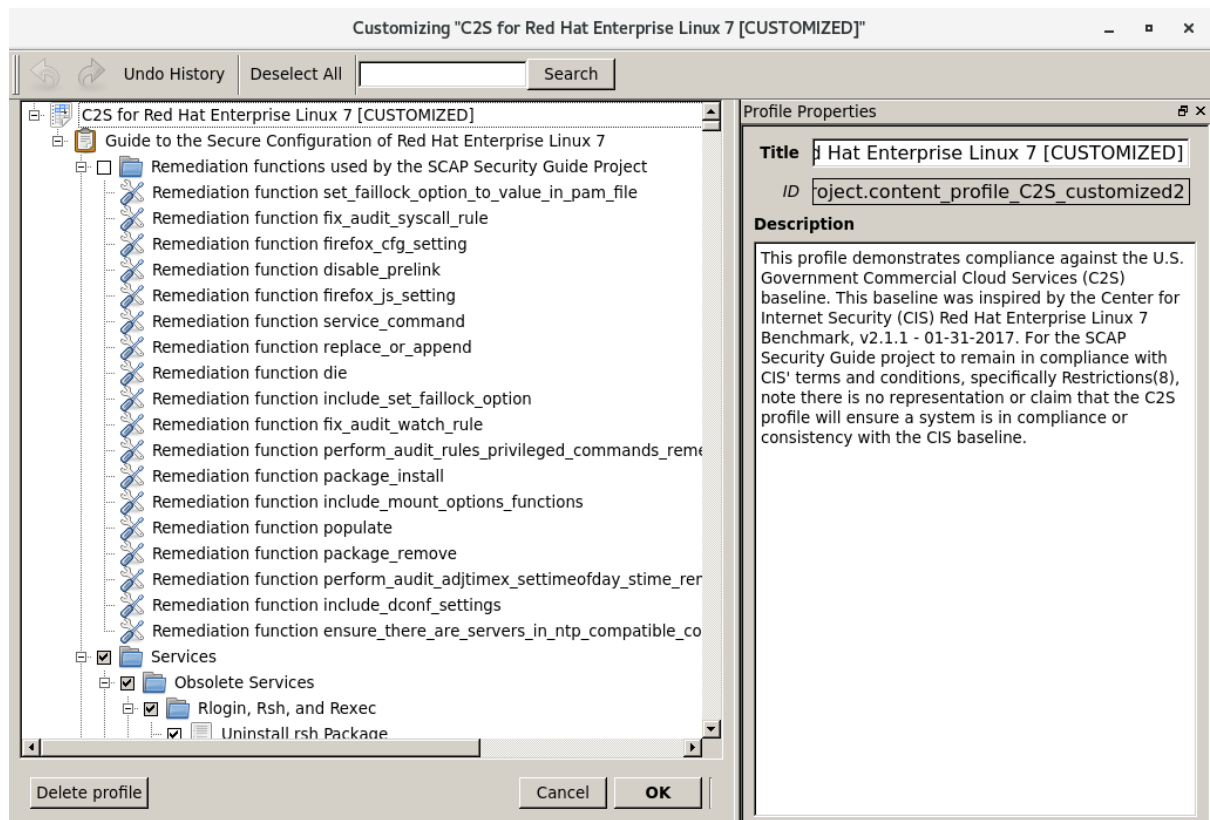


3.

使用树结构以及规则组织到逻辑组或搜索字段查找要修改的规则。

4.

使用树结构中的复选框包含或排除规则，或者在适用情况下修改规则中的值。



5.

单击 **OK** 按钮以确认更改。

6.

要永久存储您的更改，请使用以下选项之一：

- 使用 **File** 菜单中的 **Save Customization Only** 单独保存自定义文件。
- 使用 **File** 菜单中的 **Save All**，一次性保存所有安全内容。

如果您选择了 **Into a directory** 选项，**SCAP Workbench** 将 **XCCDF** 或数据流文件和自定义文件保存到指定的位置。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。这可用于将安全内容分发到无法远程扫描的系统，以及提供内容以便进一步处理。



注意

因为 **SCAP Workbench** 不支持对定制配置集的基于结果的补救，所以请使用带有 **oscap** 命令行实用程序导出的补救。

8.7.3. 相关信息

- [scap-workbench\(8\)man page](#)
- [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，使用安全配置集部署与安全配置集兼容的系统

您可在安装过程后立即使用 **OpenSCAP** 套件部署符合安全配置集的 **RHEL** 系统，如 **OSPP** 或 **PCI-DSS**。使用此部署方法时，您可以使用修复脚本（例如密码强度和分区的规则）应用之后无法应用的特定规则。

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** 附加组件不支持只使用文本的安装。
- 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

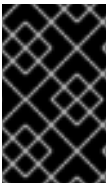
1. 在您选择的编辑器中打开 `/usr/share/scap-security-guide/kickstart/ssg-rhel7-ospp-ks.cfg` Kickstart 文件。
2. 更新分区方案以符合您的配置要求。对于 **OSPP** 合规性, 必须保留 `/boot`、`/home`、`/var`、`/var/log`、`/var/tmp` 和 `/var/log/audit` 的独立分区, 但您可以更改这些分区的大小。

**警告**

因为 **OSCAP Anaconda** 附加组件 不支持只使用文本安装，所以请不要在 **Kickstart** 文件中使用 **text** 选项。如需更多信息，请参阅 [RHBZ#1674001](#)。

3.

按照使用 **Kickstart** 执行自动安装中所述启动 **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. 针对漏洞扫描容器和容器镜像

使用以下步骤查找容器或容器镜像中的安全漏洞。

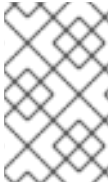
您可以使用 **aoscap-docker** 命令行实用程序或 **atomic** 扫描命令行实用程序来查找容器或容器镜像中的安全漏洞。

With **oscap-docker**，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 扫描功能扫描系统上的容器镜像和容器。您可以扫描已知的 CVE 漏洞和配置合规性。另外，您还可以将容器镜像修复为指定的策略。

8.9.1. 使用oscap-docker扫描容器镜像和容器的漏洞

您可以使用 **theoscap-docker** 实用程序扫描容器和容器镜像。



注意

Theoscap-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。

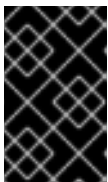
流程

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中的漏洞，并将结果保存到 **vulnerability.html** 文件中：

```
~]# oscap-docker image-cve 096cae65a207 --report vulnerability.html
```



重要

若要扫描容器，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 `the oscap-docker(8)` 和 `oscap(8)man page`。

8.9.2. 使用原子扫描扫描容器镜像和容器以了解漏洞

通过 `atomic` 扫描实用程序 <https://www.redhat.com/security/data/oval/v2/RHEL7/>，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。

使用案例

- 若要扫描所有容器镜像，可使用 `--images` 指令。
- 若要扫描所有容器，可使用 `--containers` 指令。
- 若要扫描这两种类型，可使用 `--all` 指令。
- 若要列出所有可用的命令行选项，请使用 `atomic scan --help` 命令。

`atomic` 扫描命令的默认扫描类型是 CVE 扫描。使用它检查红帽发布的 CVE OVAL 定义中定义的已知安全漏洞目标。

先决条件

- 您已使用 `atomic install rhel7/openscap` 命令从红帽容器目录(RHCC) 下载并安装了 OpenSCAP 容器镜像。

流程

1. 验证您有最新的 OpenSCAP 容器镜像，以确保定义是最新的：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap | grep version
```



重要

红帽每周提供容器镜像更新。始终使用最新的 **OpenSCAP** 容器镜像来确保由 **CVE** 扫描类型使用的 **OVAL** 定义是最新的。

2.

使用几个已知的安全漏洞扫描 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/oscaped:/etc/oscaped:ro registry.access.redhat.com/rhel7/openscap oscaped-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

- 红帽企业 **Linux** 原子主机的产品文档包含 **atomic** 命令用法和容器的详细描述。
- 红帽客户门户为 **Atomic** 命令行界面(CLI)提供了指南。

8.10. 使用特定基础镜像评估容器或容器镜像的配置合规性

按照以下步骤，利用特定的安全基准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. 评估容器镜像与 **OSPP** 配置集的合规性，并在 **report.html** HTML 文件中保存扫描结果。

```
~]$ sudo oscap-docker 096cae65a207 xccdf eval --report report.html --profile osp
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

如果要评估使用 **PCI-DSS** 基准的配置合规性，Replace **096cae65a207** 带有容器镜像 ID 和 **aosp** 值 **withpci-dss**。

验证

1. 在您选择的浏览器中检查结果，例如：

```
~]$ firefox report.html &
```



注意

标记为不可应用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 **the oscap-docker(8)** 和 **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。

8.11. 使用原子扫描扫描并修复容器镜像和容器的配置合规性

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

使用这种扫描类型，使用捆绑在 OpenSCAP 容器镜像中的 SCAP 安全指南(SSG)提供的 SCAP 内容评估基于红帽企业 Linux 的容器镜像和容器。这将启用对 SCAP 安全指南提供的任何配置集的扫描。



注意

有关使用 **atomic** 命令和容器的详细说明，请参阅红帽企业 Linux Atomic 主机的产品文档。红帽客户门户还提供了 **Atomic** 命令行界面(CLI)的指南。

先决条件

- 您已使用 **atomic install rhel7/openscap** 命令从红帽容器目录(RHCC) 下载并安装了 OpenSCAP 容器镜像。

流程

1. 列出 OpenSCAP 镜像提供的用于 **configuration_compliance** 扫描的 SCAP 内容：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap
```

使用国防部安全技术实施指南(DISA STIG)策略验证最新红帽企业 Linux 7 容器镜像的合规性，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
```

上一命令的输出包含与扫描相关的文件的信息：

```
.....
```

```
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

原子扫描生成一个子目录，其中包含 `/var/lib/atomic/openscap/` 目录中的扫描结果和报告。每次扫描时都会生成带有结果的 `arf.xml` 文件，以确保配置合规性。要生成人类可读的 HTML 报告文件，请将 `report` 子选项添加到 `--scanner_args` 选项。

2.

可选：要生成可由 *DISA STIG Viewer* 读取的 *XCCDF* 结果，将 `stig-viewer` 子选项添加到 `-scanner_args` 选项。结果放置在 `stig.xml` 中。

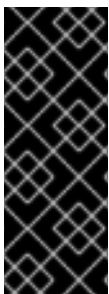


注意

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中的第一个 *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，以检查其是否符合 *DISA STIG* 策略。根据扫描结果，将生成一个包含失败扫描结果 `bash` 补救的修复脚本。然后，修复脚本应用于原始容器镜像 - 这称为补救。补救会生成带有更改配置的容器镜像，该配置作为原始容器镜像顶部的新层添加。



重要

请注意，原始容器镜像保持不变，并且在其上仅创建一个新层。补救过程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 - 在先前情况下，*DISA STIG* 策略。这也意味着红帽不再签名修复的容器镜像，这是预期的，因为它与原始容器镜像不同，因为它包含修复的层。

先决条件

-

您已使用 `atomic install rhel7/openscap` 命令从红帽容器目录(RHCC) 下载并安装了 *OpenSCAP* 容器镜像。

流程

1.

列出 OpenSCAP 镜像提供的用于 `configuration_compliance` 扫描的 SCAP 内容：

```
~]# atomic help registry.access.redhat.com/rhel7/openscap
```

2.

若要将容器镜像修复到指定的策略，可在扫描配置合规性时将 `--remediate` 选项添加到 `atomic scan` 命令。以下命令从 Red Hat Enterprise Linux 7 容器镜像构建符合 DISA STIG 策略的新修复容器镜像：

```
~]# 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
db7a70a0414e589d7c8c162712b329d4fc670fa47ddde721250fb9fcdbe9cc2.
```

Files associated with this scan are in `/var/lib/atomic/openscap/2017-11-06-13-01-42-785000`.

3.

可选：`atom scan` 命令的输出报告一个修复的镜像 ID。为了更容易记住镜像，使用一些名称标记它，例如：

```
~]# docker tag 9bbc7083760e rhel7_disa_stig
```

8.12. RHEL 7 中支持的 SCAP 安全指南配置集

仅使用 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定期更新为新功能。SCAP 内容会改变来反映这些更新，但并不总是向后兼容。

在下表中，您可以找到每个 RHEL 次要版本中提供的配置集，以及配置集与其匹配的策略版本。

表 8.2. RHEL 7.9 中支持的 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
CIS Red Hat Enterprise Linux 7 基准 2 - Server	xccdf_org.ssgproject.content_profile_cis	RHEL 7.9.9 和 less:2.2.0 RHEL 7.9.10 及更高版本 : 3.1.1
CIS Red Hat Enterprise Linux 7 基准 1 - Server	xccdf_org.ssgproject.content_profile_cis_server_l1	RHEL 7.9.10 及更高版本 : 3.1.1
CIS Red Hat Enterprise Linux 7 基准 1 - Workstation	xccdf_org.ssgproject.content_profile_cis_workstation_l1	RHEL 7.9.10 及更高版本 : 3.1.1
CIS Red Hat Enterprise Linux 7 基准 2 - Workstation	xccdf_org.ssgproject.content_profile_cis_workstation_l2	RHEL 7.9.10 及更高版本 : 3.1.1
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 和 lower:draft RHEL 7.9.5 或更高版本 : 1.2
法国信息系统安全局(ANSSI)BP-028 高级别	xccdf_org.ssgproject.content_profile_anssi_nt28_high	RHEL 7.9.6 和 less:draft RHEL 7.9.7 及更高版本 : 1.2
法国信息系统安全局(ANSSI)BP-028 Intermediary Level	xccdf_org.ssgproject.content_profile_anssi_nt28_intermediary	RHEL 7.9.4 及下面 : 草案 RHEL 7.9.5 或更高版本 : 1.2
法国信息系统安全局(ANSSI)BP-028 最低级别	xccdf_org.ssgproject.content_profile_anssi_nt28_minimal	RHEL 7.9.4 和 lower:draft RHEL 7.9.5 或更高版本 : 1.2
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis	5.4
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_cui	r1
澳大利亚网络安全中心(ACSC)Essential Eight	xccdf_org.ssgproject.content_profile_e8	未版本化
健康保障便携性和责任法案(HIPAA)	xccdf_org.ssgproject.content_profile_hipaa	未版本化
NIST 国家检查清单程序安全指南	xccdf_org.ssgproject.content_profile_ncp	未版本化

配置集名称	配置集 ID	策略版本
OSPP - 常规目的操作系统 v4.2.1 保护配置集	xccdf_org.ssgproject.content_profile_ospp	4.2.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss_centric	3.2.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.2.1
[DRAFT] 红帽企业 Linux 虚拟化主机(RHELH)的 DISA STIG	xccdf_org.ssgproject.content_profile_rhelh-stig	草案
VPP - 虚拟化保护配置文件与.1.0 用于红帽企业 Linux 管理程序 (RHELH)	xccdf_org.ssgproject.content_profile_rhelh-vpp	1.0
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
Red Hat Enterprise Linux 7 标准系统安全配置集	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 and 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 或更高版本 : V3R6
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 或更高版本 : V3R6

表 8.3. RHEL 7.8 中支持的 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
DRAFT - ANSSI DAT-NT28 (增强)	xccdf_org.ssgproject.content_profile_anssi_nt28_enhanced	草案
DRAFT - ANSSI DAT-NT28 (高)	xccdf_org.ssgproject.content_profile_anssi_nt28_high	草案

配置集名称	配置集 ID	策略版本
DRAFT - ANSSI DAT-NT28 (中间)	xccdf_org.ssgproject.content_profile_anssi_nt28_intermediary	草案
DRAFT - ANSSI DAT-NT28 (最小)	xccdf_org.ssgproject.content_profile_anssi_nt28_minimal	草案
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis	5.4
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_cui	r1
澳大利亚网络安全中心 (ACSC) Essential Eight	xccdf_org.ssgproject.content_profile_e8	未版本化
健康保障便携性和责任法案 (HIPAA)	xccdf_org.ssgproject.content_profile_hipaa	未版本化
NIST 国家检查清单程序安全指南	xccdf_org.ssgproject.content_profile_ncp	未版本化
OSPP - 常规目的操作系统 v4.2.1 保护配置集	xccdf_org.ssgproject.content_profile_ospp	4.2.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss_centric	3.2.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.2.1
[DRAFT] 红帽企业 Linux 虚拟化主机(RHELH)的 DISA STIG	xccdf_org.ssgproject.content_profile_rhelh-stig	草案
VPP - 虚拟化保护配置文件与1.0 用于红帽企业 Linux 管理程序 (RHELH)	xccdf_org.ssgproject.content_profile_rhelh-vpp	1.0
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
Red Hat Enterprise Linux 7 标准系统安全配置集	xccdf_org.ssgproject.content_profile_standard	未版本化

配置集名称	配置集 ID	策略版本
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

配置集名称	配置集 ID	策略版本
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis	5.4
健康保障便携性和责任法案 (HIPAA)	xccdf_org.ssgproject.content_profile_hipaa	未版本化
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_nist-800-171-cui	r1
OSPP - 常规目的操作系统保护配置集 v.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
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss-centric	3.2.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3.2.1 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.2.1
VPP - 虚拟化保护配置文件与.1.0 用于红帽企业 Linux 管理程序 (RHELH)	xccdf_org.ssgproject.content_profile_rhelh-vpp	1.0
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
Red Hat Enterprise Linux 7 标准系统安全配置集	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 安全指南配置集

配置集名称	配置集 ID	策略版本
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis	5.4
健康保障便携性和责任法案 (HIPAA)	xccdf_org.ssgproject.content_profile_hipaa	未版本化
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_nist-800-171-cui	r1
OSPP - 常规目的操作系统保护配置集 v.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
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss-centric	3.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.1
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
Red Hat Enterprise Linux 7 标准系统安全配置集	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 安全指南配置集

配置集名称	配置集 ID	策略版本
用于 Red Hat Enterprise Linux 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis-rhel7-server	5.4
General-Purpose Systems 的通用配置集	xccdf_org.ssgproject.content_profile_common	未版本化

配置集名称	配置集 ID	策略版本
标准 Docker 主机安全配置集	xccdf_org.ssgproject.content_profile_docker-host	未版本化
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_nist-800-171-cui	r1
美国政府配置基线(USGCB / STIG)- DRAFT	xccdf_org.ssgproject.content_profile_ospp-rhel7	3.9
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss-centric	3.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.1
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
标准系统安全配置集	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
Red Hat Virtualization Hypervisor 的 STIG	xccdf_org.ssgproject.content_profile_stig-rhevh-upstream	1.4

表 8.7. RHEL 7.4 中支持的 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis-rhel7-server	5.4
General-Purpose Systems 的通用配置集	xccdf_org.ssgproject.content_profile_common	未版本化
标准 Docker 主机安全配置集	xccdf_org.ssgproject.content_profile_docker-host	未版本化
非联邦信息系统和组织中未分类的信息(NIST 800-171)	xccdf_org.ssgproject.content_profile_nist-800-171-cui	r1

配置集名称	配置集 ID	策略版本
美国政府配置基线(USGCB / STIG)- DRAFT	xccdf_org.ssgproject.content_profile_ospp-rhel7	3.9
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss-centric	3.1
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.1
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化
标准系统安全配置集	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
Red Hat Virtualization Hypervisor 的 STIG	xccdf_org.ssgproject.content_profile_stig-rhev-upstream	

表 8.8. RHEL 7.3 中支持的 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
Red Hat Enterprise Linux 7 的 C2S	xccdf_org.ssgproject.content_profile_C2S	未版本化
隐私信息服务(CJIS)安全策略	xccdf_org.ssgproject.content_profile_cjis-rhel7-server	5.4
General-Purpose Systems 的通用配置集	xccdf_org.ssgproject.content_profile_common	未版本化
Red Hat Enterprise Linux 7 的 CNSSI 1253 低/Low/Low Control Baseline	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
Red Hat Enterprise Linux 7 的 PCI-DSS v3 Control Baseline	xccdf_org.ssgproject.content_profile_pci-dss	3.1
认证云供应商的红帽企业配置文件 (RH CCP)	xccdf_org.ssgproject.content_profile_rht-ccp	未版本化

配置集名称	配置集 ID	策略版本
标准系统安全配置集	<code>xccdf_org.ssgproject.content_profile_standard</code>	未版本化
用于运行 GUI 的红帽企业 Linux 7 服务器的 STIG	<code>xccdf_org.ssgproject.content_profile_stig-rhel7-server-gui-upstream</code>	1.4
Red Hat Enterprise Linux 7 服务器的 STIG	<code>xccdf_org.ssgproject.content_profile_stig-rhel7-server-upstream</code>	1.4
Red Hat Enterprise Linux 7 Workstation 的 STIG	<code>xccdf_org.ssgproject.content_profile_stig-rhel7-workstation-upstream</code>	1.4

表 8.9. RHEL 7.2 中支持的 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
General-Purpose Systems 的通用配置集	<code>xccdf_org.ssgproject.content_profile_common</code>	未版本化
Red Hat Enterprise Linux 7 的文档 PCI-DSS v3 Control Baseline	<code>xccdf_org.ssgproject.content_profile_pci-dss</code>	草案
认证云供应商的红帽企业配置文件 (RH CCP)	<code>xccdf_org.ssgproject.content_profile_rht-ccp</code>	未版本化
标准系统安全配置集	<code>xccdf_org.ssgproject.content_profile_standard</code>	未版本化
Red Hat Enterprise Linux 7 服务器的预发行版本 Draft STIG	<code>xccdf_org.ssgproject.content_profile_stig-rhel7-server-upstream</code>	草案

表 8.10. RHEL 7.1 支持 SCAP 安全指南配置集

配置集名称	配置集 ID	策略版本
认证云供应商的红帽企业配置文件 (RH CCP)	<code>xccdf_org.ssgproject.content_profile_rht-ccp</code>	未版本化

其它资源



有关 RHEL 8 中的配置集的详情，请查看 [RHEL 8 支持的 SCAP 安全指南配置集](#)

8.13. 相关信息

- [支持版本的 SCAP 安全指南](#) - 文章列出了不同版本的 RHEL 中受支持的 SCAP 安全指南版本。
- [OpenSCAP 项目页面](#) - OpenSCAP 项目的主页提供有关 `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 项目的主页提供有关 `scap-workbench` 应用的详细信息。
- [SCAP 安全指南\(SSG\)项目页面](#) - 为红帽企业 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性](#) - 亲身实践实验室，利用红帽企业 Linux 中包含的工具获取自动化安全合规性的初始经验，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信息。
- [红帽安全演示：使用 RHEL 安全技术进行自行定义](#) - 亲身实践实验室，了解如何在 RHEL 系统的所有级别使用您在红帽企业 Linux 中可用的关键安全技术（包括 OpenSCAP）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信息。
- [美国国家标准与技术研究院\(NIST\)SCAP 页面](#) - 此页面包含大量 SCAP 相关材料，包括 SCAP 出版物、规范和 SCAP 验证计划。
- [国家漏洞数据库\(NVD\)](#) - 此页面是 SCAP 内容和其他基于 SCAP 标准漏洞管理数据的最大存储库。
- [红帽 OVAL 内容存储库](#) - 此存储库含有红帽企业 Linux 系统漏洞的 OVAL 定义。这是推荐的漏洞内容来源。
- [MITRE CVE](#) - 这个数据库包含 MITRE 公司提供的公开安全漏洞。对于 RHEL，建议您使用红帽提供的 OVAL CVE 内容。
- [MITRE OVAL](#) - 此页面代表 MITRE 公司提供的 OVAL 相关项目。除了与 OVAL 相关的信息

外，这些页面包含最新版本的 OVAL 语言，以及包含数千个 OVAL 定义的 OVAL 内容存储库。请注意，要扫描 RHEL，建议使用红帽提供的 OVAL CVE 内容。

- [红帽卫星文档](#) - 这组指南除了其他主题外，介绍了如何使用 OpenSCAP 在多个系统上维护系统安全性。

第 9 章 联邦标准和强制

为了保持安全水平，您的企业可以努力遵守联邦和行业安全规范、标准和法规。本章介绍了其中一些标准和法规。

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 是一种计算机安全标准，由美国政府和行业工作组验证加密模块的质量。请参阅[NIST Computer 安全资源中心上的官方 FIPS 出版物](#)。

FIPS 140-2 标准确保加密工具正确实施了它们的算法。有关这些级别以及 FIPS 标准的其他规格的详情，请参阅<http://dx.doi.org/10.6028/NIST.FIPS.140-2> 完整的 FIPS 140-2 标准。

要了解合规性要求，请参阅 [Red Hat Government Standards](#) 页。

9.1.1. 启用 FIPS 模式

要使红帽企业 Linux 符合联邦信息处理标准(FIPS)出版物 140-2，您需要进行多项更改，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。

系统安装过程中

要强化 FIPS 140-2 合规性，请在系统安装期间将 `fips=1` 内核选项添加到内核命令行中。使用这个选项时，所有密钥的生成都使用 FIPS 批准的算法和持续监控测试完成。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，确保系统在安装过程中有大量熵。推荐的击键次数为 256 及更多。小于 256 个击键操作可能会生成一个非唯一密钥。

系统安装后

要在安装后将系统的内核空间 and 用户空间切换到 FIPS 模式，请按照以下步骤执行：

1.

安装 `dracut-fips` 软件包：

```
~]# yum install dracut-fips
```

对于支持 **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
```

要启用模块内完整性验证并在内核引导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.

修改启动加载器配置。

要引导进入 **FIPS** 模式，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** 或 **/boot/EFI/** 分区位于单独的分区中，请在内核命令行中添加 **boot= <partition>**（其中 **<partition>** 代表 **/boot**）参数。

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
```

要确保 **boot=** 配置选项在引导之间更改设备命名时也能正常工作，请运行以下命令来识别分区的通用唯一标识符(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 **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**, 请按如下所示重建 **grub.cfg** 文件:

- 在基于 **BIOS** 的机器中以 **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 Systems 架构)**

将 **fips=1** 和 **boot=<partition** 添加到 **/etc/zipl.conf** 中的内核命令行, 并通过输入以下内容应用更改:

```
~]# zipl
```

4.

确保已禁用预链接。

若要正确运行模块内完整性验证，必须禁用库和二进制文件的预先链接。Prelinking 由 prelink 软件包执行，该软件包默认不安装。除非安装了 prelink，否则不需要这个步骤。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上的现有预链接，请使用 prelink -u -a 命令。

5.

重启您的系统。

在容器中启用 FIPS 模式

如果主机也以 FIPS140-2 模式设定，且满足以下要求之一，则容器可切换到 FIPS140-2 模式：

- **dracut-fips** 软件包安装在容器中。
- **/etc/system-fips** 文件从主机上挂载到容器上。

9.2. 国家工业安全计划操作手册(NISPOM)

NISPOM（也称为国防部 5220.22-M）作为国家工业安全计划(NISP)的一部分，为所有政府承包商就分类信息制定一系列程序和要求。当前 NISPOM 日期为 2006 年 2 月 28 日，整合了 2013 年 3 月 28 日的显著更改。NISPOM 文档可以从以下 URL 下载：

9.3. 支付卡行业数据安全标准(PCI DSS)

从 <https://www.pcisecuritystandards.org/about/index.shtml>：PCI 安全标准委员会是 2006 年启动的开放全球论坛，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)是计算机软件和硬件标准化安全安装和维护的方法。

有关 **STIG** 的更多信息，请参阅以下 **URL**：

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)是美国采用的加密标准。政府该标准包含三个块密码：AES-128、AES-192 和 AES-256，最初发布为 Rijndael 的更大集合。每个 AES 密码都具有 128 位的块大小，密钥大小分别为 128、192 和 256 位。AES 密码经过了广泛的分析，如今已在全球范围内使用，如同其前身的数据加密标准(DES)一样。[3]

A.1.1.1. AES 历史记录

美国国家标准与技术研究院(NIST)宣布 AES。2001 年 11 月 26 日，经过 5 年标准化流程后，FIPS PUB 197(FIPS 197)。在选择 Rijndael 前，会呈现和评估 15 个竞争设计。它作为一个标准，于 2002 年 5 月 26 日生效。它在许多不同的加密软件包中可用。AES 是 NSA 批准的首个公开访问和开放密码，用于顶级机密信息（请参阅 AES 上的维基百科文章中的 Security 部分）。[4]

Rijndael 密码由两个 Belgian 加密师（Joan Daemen 和文莱 Rijmen）开发，由他们提交到 AES 选择流程。Rijndael 是两个发明人的 portmanteau。[5]

A.1.2. 数据加密标准 - DES

数据加密标准(DES)是一种块加密（一种共享密码加密），国家标准局于 1976 年被美国官方联邦信息处理标准(FIPS)选择，随后在 1976 年广泛应用。它基于使用 56 位密钥的对称密钥算法。该算法最初针对分类设计元素、相对较短的关键长度以及国家安全局(NSA)后门问题。因此，DES 深受学术影响力，促使现代化理解块密码和他们的加密分析。[6]

A.1.2.1. DES History

对于许多应用程序，DES 现在被视为不安全。这主要是因为 56 位密钥大小太小；在 1999 年 1 月，分布式.net 以及电子前线基金会合作，在 22 小时和 15 分钟内公开终止 DES 密钥。有一些分析结果也说明了密码的理论缺点，尽管它们无法在实践中挂载。尽管存在理论上的攻击，但算法以 Triple DES 的形式具有实际的安全性。近年来，该密码已被高级加密标准(AES)取代。[7]

某些文档中将 DES 作为标准进行区分，而 DES 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，许多加密算法和加密系统使用，它们的区别在于使用非对称密钥算法而不是使用对称密钥算法。使用公钥-私钥加密技术，许多之前未知的消息保护通信或身份验证方法变得可行。它们不需要一个或多个 **secret** 密钥的安全初始交换，在使用对称密钥算法时需要。它还可用于创建数字签名。[9]

公钥加密是全球范围内的基本和广泛使用的技术，也是作为传输层安全(TLS)(SSL)、PGP 和 GPG 等互联网标准的基础方法。[10]

公钥加密中使用的区分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消息的密钥不同。每个用户都有一对加密密钥 - 公钥和一个私钥。私钥是保密的，而公钥也可能广泛分发。消息使用收件人的公钥加密，并且只能使用对应的私钥解密。在数学上，密钥是相关的，但私钥不能完全（例如，实际或投射实践）派生自公钥。正是此类算法的发现改变了从 1970 年代中期开始的加密技术实践。[11]

与之相反，**Symmetric-key** 算法（已经使用了几十年）的对称密钥算法已使用几十年，使用由发件人和接收方共享的单一 **secret** 密钥（必须保持私密性，因此考虑到常用术语的不确定性），用于加密和解密。要使用对称加密方案，发送者和接收方必须提前安全地共享密钥。[12]

由于对称密钥算法的计算密集型几乎始终降低，因此通常使用密钥交换算法并使用该密钥和对称密钥算法传输数据。PGP 和 SSL/TLS 方案系列能够执行此操作，因此称为混合加密系统。[13]

A.2.1. Diffie-Hellman

Diffie-Hellman 密钥交换(D-H)是一种加密协议，允许没有预先了解对方的双方通过不安全的通信渠道共同建立共享密钥。然后，此密钥可用于使用对称密钥密码加密随后的通信。[14]

A.2.1.1. Diffie-Hellman History

这个方案最初于 1976 年由 Whitfield Diffie 和 Martin Hellman 发布，但后来发现，它几年前已在英国的英国情报机构 Malcolm J. Williamson 中单独发明。2002 年，Hellman 建议将算法称为 **Diffie-Hellman-Merkle** 密钥交换，以认可 Ralph Merkle 对公钥加密技术(Hellman, 2002)的贡献。[15]

尽管 **Diffie-Hellman** 密钥协议本身是一个匿名（未经身份验证的）密钥声明协议，但它为各种经过身份验证的协议提供了基础，并用于在传输层安全性的临时模式中提供完美的转发保密（称为 **EDH** 或 **DHE**，具体取决于密码套件）。[16]

美国. 专利 4,200,770, 现在过期, 描述算法和信用额 Hellman、Diffie 和 Merkle 作为发明者。[17]

A.2.2. RSA

在加密学中, RSA (代表 Rivest、Shamir 和 Adleman, 首先公开描述) 是公钥加密的算法。众所周知, 这是适用于签名和加密的首个算法, 也是公钥加密的首批重大进步。RSA 在电子商业协议中广泛使用, 并相信会给予足够长的密钥并使用最新的实施。

A.2.3. DSA

DSA (数字签名算法) 是美国联邦政府数字签名标准, 是数字签名的美国政府标准。DSA 仅用于签名, 不是加密算法。[18]

A.2.4. SSL/TLS

传输层安全(TLS)及其前身的安全套接字层(SSL)是加密协议, 为互联网等网络通信提供安全性。TLS 和 SSL 通过传输层端到端加密网络连接的分段。

这些协议的多个版本广泛应用于 Web 浏览、电子邮件、互联网传真、即时消息和语音 overIP(VoIP)等应用程序中。[19]

A.2.5. cramer-Shoup Cryptosystem

Cramer-Shoup 系统是一种非对称密钥加密算法, 经证实是第一个高效方案, 可防止使用标准加密假设的自适应所选密码攻击。其安全性基于决策 Diffie-Hellman 假设的计算实用性 (普遍假定, 但没有证明)。Ronald Cramer 和 Victor Shoup 在 1998 年由 Ronald Cramer 和 Victor Shoup 开发, 是 ElGamal 加密系统的扩展。相比 ElGamal, Cramer-Shoup 增加了额外的元素来确保非可测量性, 即使有资源攻击者也是如此。这种不可变性通过使用冲突补救哈希函数和其他计算来实现, 从而导致密码短语比 ElGamal 大的两倍。[20]

A.2.6. Elgamal Encryption

在加密中, ElGamal 加密系统是一种用于公钥加密的非对称密钥加密算法, 基于 Diffie-Hellman 密钥协议。1985 年被 Taher ElGamal 描述。Elgamal 加密用于免费 GNU Privacy Guard 软件、最新版本 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 加密" 维基百科. 24 February 2010 http://en.wikipedia.org/wiki/Cramer-Shoup_cryptosystem
- [21] 维基百科. 24 February 2010 http://en.wikipedia.org/wiki/ElGamal_encryption

附录 B. 修订历史记录

修订 1-43 带有更新 Compliance 和漏洞扫描一章的异步发行版本。	Fri Feb 7 2020	Jan Fiala
修订 1-42 7.7 GA 发行的版本。	Fri Aug 9 2019	Mirek Jahoda
修订 1-41 7.6 GA 发行的版本。	Sat Oct 20 2018	Mirek Jahoda
修订 1-32 7.5 GA 发行的版本。	Wed Apr 4 2018	Mirek Jahoda
修订 1-30 7.4 GA 发行的版本。	Thu Jul 27 2017	Mirek Jahoda
修订 1-24 带有 misc. 更新的同步版本，特别是 firewalld 部分。	Mon Feb 6 2017	Mirek Jahoda
修订 1-23 7.3 GA 发布版本。	Tue Nov 1 2016	Mirek Jahoda
修订 1-19 添加了智能卡部分。	Mon Jul 18 2016	Mirek Jahoda
修订 1-18 添加了 OpenSCAP-daemon 和 Atomic Scan 部分。	Mon Jun 27 2016	Mirek Jahoda
修订 1-17 带有 misc 更新的异步版本。	Fri Jun 3 2016	Mirek Jahoda
修订 1-16 7.2 后 GA 修复程序。	Tue Jan 5 2016	Robert Krátký
修订 1-15 7.2 GA 版本。	Tue Nov 10 2015	Robert Krátký
修订 1-14.18 带有 misc 更新的异步版本。	Mon Nov 09 2015	Robert Krátký
修订 1-14.17 7.1 GA 版本。	Wed Feb 18 2015	Robert Krátký
修订 1-14.15 更新 以在红帽客户门户上排序顺序。	Fri Dec 06 2014	Robert Krátký
修订 1-14.13 反映 POODLE vuln 的更新。	Thu Nov 27 2014	Robert Krátký
修订 1-14.12 7.0 GA 版本。	Tue Jun 03 2014	Tomáš Čapek

