Red Hat Enterprise Linux 9

9.3 发行注记

Red Hat Enterprise Linux 9.3 发行注记

Last Updated: 2024-06-27

Red Hat Enterprise Linux 9 9.3 发行注记

Red Hat Enterprise Linux 9.3 发行注记

法律通告

摘要

本发行注记提供了在 Red Hat Enterprise Linux 9.3 中实现的改进和附加组件的高级信息，并在本版本中记录了已知的问题，以及重要的 bug 修复、技术预览、已弃用的功能和其他详情。 有关安装 Red Hat Enterprise Linux 的详情，请参考 Installation。

目录

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 JIRA 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 点顶部导航栏中的 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 概述

1.1. RHEL 9.3 的主要变化

安装程序和镜像创建
镜像构建器的主要亮点：

- 对 AWS EC2 AMD 或 Intel 64 位架构 AMI 镜像的增强，以支持 UEFI 引导，以及支持旧的 BIOS 引导。

如需更多信息，请参阅 新功能 - 安装程序和镜像创建。

1.1.1. 引导加载程序

带有 BLS 的 grub2-mkconfig 的新默认行为

有了此版本，**grub2-mkconfig** 命令默认不再使用 **GRUB_CMDLINE_LINUX** 覆盖引导装载程序规格(BLS)片断中的内核命令行。引导装载程序菜单中的每个内核都从其 BLS 代码段获取其内核命令行。这个新的默认行为是由 **GRUB_ENABLE_BLSCFG=true** 选项造成的。

详情请查看 引导装载程序中的新功能。

RHEL for Edge
RHEL for Edge 的主要亮点：

- 添加了对以下镜像类型的支持：
  - **minimal-raw**
  - **edge-vsphere**
  - **edge-ami**
- 提供了新的 FIDO Device Onboarding Servers 容器镜像
  - rhel9/fdo-manufacturing-server
  - rhel9/fdo-owner-onboarding-server
  - rhel9/fdo-rendezvous-server
  - rhel9/fdo-serviceinfo-api-server

如需更多信息，请参阅 新功能 - RHEL for Edge 。

安全性
与安全相关的主要亮点：

- **Keylime** 被 rebase 到版本 7.3.0。
- **keylime RHEL 系统角色** 可用。使用此角色，您可以更轻松地配置 Keylime 验证器 和 Keylime 注册器。
- 出于加密目的，**OpenSSH** 从不太安全的 SHA-1 消息摘要被进一步迁移，但在其他场景中应用了更安全的 SHA-2。

- **pcsc-lite-ccid** USB 芯片/智能卡接口设备(CCID)和集成电路卡设备(ICCD)驱动程序已 rebase 到版本 1.5.2。
- RHEL 9.3 引入了进一步改进，以支持用于所有 TLS 1.2 连接的 FIPS-140-3 标准所需的 **Extended Master Secret (EMS)** 扩展(RFC 7627)。
- **SEtools**，用于 SELinux 策略分析的图形工具、命令行工具和库的集合被 rebase 到版本 4.4.3。
- **OpenSCAP** 已 rebase 到版本 1.3.8。
- **SCAP 安全指南** 已 rebase 到版本 0.1.69，最值得注意的是：
  - ANSSI 配置文件已更新至版本 2.0。
  - 为 RHEL 9 添加了三个新的 SCAP 配置文件，与 CCN-STIC-610A22 指南保持一致。

如需更多信息，请参阅 新功能 - 安全性。

动态编程语言、网页和数据库服务器

以下应用程序流的后续版本现在可用：

- **Redis 7**
- **Node.js 20**

另外，**Apache HTTP 服务器** 已更新至版本 2.4.57。

如需更多信息，请参阅 新功能 - 动态编程语言、Web 和数据库服务器。

编译器和开发工具

更新了系统工具链

以下系统工具链组件已在 RHEL 9.3 中进行了更新：

- **GCC 11.4.1**

更新了性能工具和调试器

以下性能工具和调试器已在 RHEL 9.3 中进行了更新：

- **Valgrind 3.21**
- **SystemTap 4.9**
- **elfutils 0.189**

更新了性能监控工具

以下性能监控工具已在 RHEL 9.3 中进行了更新：

- **PCP 6.0.5**
- **Grafana 9.2.10**

更新了编译器工具集

以下编译器工具集已在 RHEL 9.3 中进行了更新：

- **GCC Toolset 13**（新的）
- **LLVM Toolset 16.0.6**

- **Rust Toolset 1.71.1**
- **Go Toolset 1.20.10**

有关详细更改，请参阅 新功能 - 编译器和开发工具。

RHEL 9 中的 Java 实现
RHEL 9 AppStream 软件仓库包括：

- **java-21-openjdk** 软件包，其提供了 OpenJDK 21 Java 运行时环境和 OpenJDK 21 Java 软件开发套件。OpenJDK 21.0.1 安全发行版本也可用于安装。建议您安装 OpenJDK 21.0.1 更新以获取最新的安全修复。
- **java-17-openjdk** 软件包，提供 OpenJDK 17 Java 运行时环境和 OpenJDK 17 Java 软件开发组件。
- **java-11-openjdk** 软件包，提供 OpenJDK 11 Java 运行时环境和 OpenJDK 11 Java 软件开发组件。
- **java-1.8.0-openjdk** 软件包，提供 OpenJDK 8 Java 运行时环境和开源 JDK 8 Java 软件开发组件。

OpenJDK 软件包的红帽构建的在其可移植 Linux 版本和 RHEL 9.3 及更新版本之间共享一组二进制文件。有了这个更新，在 RHEL 上从源 RPM 重建 OpenJDK 软件包的过程有一个变化。有关新重建过程的更多信息，请参阅 README.md 文件，该文件包含在 Red Hat build of OpenJDK 的 SRPM 软件包中，并由 **/usr/share/doc** 树下的 **javaawa-openjdk-headless** 软件包安装。

如需更多信息，请参阅 OpenJDK 文档。

1.2. 原位升级

从 RHEL 8 原位升级到 RHEL 9
目前支持的原位升级路径包括：

- 在以下构架上，从 RHEL 8.6 升级到 RHEL 9.0、从 RHEL 8.8 升级到 RHEL 9.2，以及从 RHEL 8.9 升级到 RHEL 9.3：
  - 64 位 Intel
  - 64 位 AMD
  - 64-bit ARM
  - IBM POWER 9(little endian)
  - IBM Z 架构，不包括 z13
- 在带有 SAP HANA 的系统上，从 RHEL 8.6 升级到 RHEL 9.0 和从 RHEL 8.8 升级到 RHEL 9.2

如需更多信息，请参阅支持的 Red Hat Enterprise Linux 原位升级路径 。

有关执行原位升级的步骤，请参阅从 RHEL 8 升级到 RHEL 9 。

如果您要升级到带有 SAP HANA 的 RHEL 9.2，请确保系统在升级前已进行了 SAP 认证。有关在具有 SAP 环境的系统上执行原位升级的说明，请参阅 如何将 SAP 环境从 RHEL 8 原位升级到 RHEL 9 。

主要改进包括：

- 在使用 **ftype=0** 格式的 XFS 文件系统的系统上，对磁盘空间的要求已显著降低。
- 出于升级目的，在升级过程中创建的磁盘镜像现在具有动态大小。不再需要 **LEAPP_OVL_SIZE** 环境变量。
- 修复了现有磁盘分区上计算所需可用空间的问题。现在，在系统需要重启前可以正确地检测到缺少的可用磁盘空间，报告可以正确地显示文件系统没有足够的可用空间来继续升级 RPM 事务。
- 现在，第三方驱动程序可以在原位升级过程中使用自定义 leapp 执行器进行管理。
- 预升级概述和升级报告现在可以在终端中输出。
- 现在支持在 Red Hat OpenStack Platform 中升级 RHEL Real Time 和 RHEL Real Time for Network Functions Virtualization (NFV)。

从 RHEL 7 原位升级到 RHEL 9

无法执行从 RHEL 7 直接升级到 RHEL 9 的原位升级。但是，您可以执行从 RHEL 7 原位升级到 RHEL 8，然后再执行到 RHEL 9 的第二个原位升级。如需更多信息，请参阅从 RHEL 7 升级到 RHEL 8 。

1.3. 红帽客户门户网站 LABS

红帽客户门户网站 Labs 是客户门户网站的一个部分中的一组工具，地址为 https://access.redhat.com/labs/。红帽客户门户网站 Labs 中的应用程序可帮助您提高性能、快速解决问题、发现安全问题以及快速部署和配置复杂应用程序。一些最常用的应用程序有：

- Registration Assistant
- Kickstart Generator
- Red Hat Product Certificates
- Red Hat CVE Checker
- Kernel Oops Analyzer
- Red Hat Code Browser
- VNC 配置器
- Red Hat OpenShift Container Platform Update Graph
- Red Hat Satellite Upgrade Helper
- JVM Options Configuration Tool
- 负载均衡配置工具
- Red Hat OpenShift Data Foundation Supportability and Interoperability Checker
- Ansible Automation Platform Upgrade Assistant
- 每个池计算器的 Ceph 放置组（PG）

1.4. 其他资源

与其他版本系统相比， Red Hat Enterprise Linux 9 的**能力和限制**可在知识库文章Red Hat Enterprise Linux 技术能力和限制中获得。

有关 Red Hat Enterprise Linux **生命周期** 的详情请查看 Red Hat Enterprise Linux 生命周期文档 。

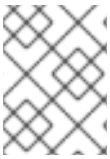
软件包清单文档为 RHEL 9 提供 **软件包列表**，包括许可证和应用程序兼容性等级。

Red Hat Enterprise Linux 9: Application Compatibility Guide 文档中的**解释应用程序兼容性等级**。

RHEL 8 和 RHEL 9 的主要区别 （包括删除的功能）包括在使用 RHEL 9 时的注意事项 。

有关如何执行**从 RHEL 8 到 RHEL 9 的原位升级**的说明，请参考从 RHEL 8 升级到 RHEL 9 的文档。

Red Hat Insights服务可让您主动发现、检查并解决已知的技术问题，所有 RHEL 订阅都可以使用它。有关如何安装 Red Hat Insights 客户端并将您的系统注册到该服务的说明，请查看 Red Hat Insights 入门 页面。



注意

公共发行注记包括访问原始跟踪票据的链接，但私有发行注记无法查看，因此不包括链接。[1]

[1] 公共发行注记包括访问原始跟踪票据的链接，但私有发行注记无法查看，因此不包括链接。

第 2 章 构架

Red Hat Enterprise Linux 9.3 与内核版本 5.14.0-362.8.1 一起分发，它对以下构架提供最低要求的版本支持（在括号中所述）：

- AMD 和 Intel 64 位体系架构 (x86-64-v2)
- 64 位 ARM 架构(ARMv8.0-A)
- IBM Power Systems, Little Endian(POWER9)
- 64 位 IBM Z (z14)

请确定为每个构架购买正确的订阅。如需更多信息,请参阅 Red Hat Enterprise Linux 入门 - 附加构架 。

第 3 章 RHEL 9 发布的内容

3.1. 安装

Red Hat Enterprise Linux 9 使用 ISO 镜像安装。AMD64、Intel 64 位、64 位 ARM、IBM Power Systems 和 IBM Z 架构有两种类型的 ISO 镜像：

- 安装 ISO：包含 BaseOS 和 AppStream 软件仓库的完整安装镜像，并允许您在没有附加软件仓库的情况下完成安装。在产品下载页面中，**安装 ISO** 被称为 **Binary DVD**。

注意

安装 ISO 镜像的大小为几个 GB，因此可能不适用于光盘介质格式。当使用安装 ISO 镜像时，建议使用 USB 盘或 USB 硬盘驱动器创建可引导安装介质。您还可以使用 Image Builder 工具创建自定义的 RHEL 镜像。有关镜像构建器的更多信息，请参阅*编写自定义的 RHEL 系统镜像*文档。

- 引导 ISO：用来引导到安装程序的最小引导 ISO 镜像。这个选项需要访问 BaseOS 和 AppStream 软件仓库来安装软件包。软件仓库是安装 ISO 镜像的一部分。您还可以在安装过程中注册红帽 CDN 或 Satellite，以使用来自红帽 CDN 或 Satellite 的最新 BaseOS 和 AppStream 内容。

有关下载 ISO 镜像、创建安装介质以及完成 RHEL 安装的说明，请参阅 执行标准的 RHEL 9 安装 文档。有关自动 Kickstart 安装和其他高级主题，请参阅 执行高级 RHEL 9 安装 文档。

3.2. 软件仓库

Red Hat Enterprise Linux 9 由两个主要软件仓库发布：

- BaseOS
- AppStream

两个软件仓库都需要一个基本的 RHEL 安装，所有 RHEL 订阅都包括它们。

BaseOS 存储库中的内容旨在提供底层操作系统功能的核心集合，其为所有安装提供基础。这部分内容采用 RPM 格式，它的支持条款与之前的 RHEL 版本相似。如需更多信息，请参阅覆盖范围详情文档。

AppStream 仓库的内容包括额外的用户空间应用程序、运行时语言和数据库来支持各种工作负载和使用案例。

另外，所有 RHEL 订阅都可以使用 CodeReady Linux Builder 软件仓库。它为开发人员提供了额外的软件包。不支持包括在 CodeReady Linux Builder 存储库中的软件包。

有关 RHEL 9 软件仓库及其提供的软件包的更多信息，请参阅软件包清单。

3.3. 应用程序流

用户空间组件的多个版本会以 Application Streams（应用程序流）的形式提供，其更新频率会比核心操作系统软件包的更新频率更快。这为自定义 RHEL 提供了更大的灵活性，而不影响平台或特定部署的基本稳定性。

应用程序流以 RPM 格式提供，可以是一个模块（RPM 格式的一个扩展），软件集合（Software Collections,），或 Flatpaks。

每个 Application Stream 组件都有其特定的生命周期，可能和 RHEL 9 的生命周期相同或更短。有关 RHEL 生命周期信息，请查看 Red Hat Enterprise Linux 生命周期 。

RHEL 9 改进了应用程序流的使用体验，它提供了初始的应用程序流版本，可以使用传统的 **dnf install** 命令作为 RPM 软件包进行安装。

注意

某些 RPM 格式的初始应用程序流的生命周期比 Red Hat Enterprise Linux 9 要短。

一些额外的 Application Stream 版本将作为模块发布，并在以后的 RHEL 9 次要发行本中带有较短的生命周期。模块是代表逻辑单元的软件包集合： 应用程序、语言堆栈、数据库或一组工具。这些软件包被一同构建、测试并发布。

始终决定要安装哪个版本的应用程序流，并确保首先查看 Red Hat Enterprise Linux 应用程序流生命周期。

需要快速更新的内容（例如备用编译器和容器工具）会在滚动流中提供，且不会并行提供替代版本。滚动流可以打包为 RPM 或模块。

有关 RHEL 9 中可用的 Application Streams 及其应用程序兼容性级别的详情，请查看软件包清单。Red Hat Enterprise Linux 9: Application Compatibility Guide 文档中的解释应用程序兼容性等级。

3.4. 使用 YUM/DNF 的软件包管理

在 Red Hat Enterprise Linux 9 中，使用 **DNF** 确保软件安装 。红帽继续支持使用 **yum** 术语，以便与以前的 RHEL 主版本保持一致。如果您键入 **dnf** 而不是 **yum**，则命令按预期运行，因为它们都是兼容性的别名。

虽然 RHEL 8 和 RHEL 9 基于 **DNF**，但它们与 RHEL 7 中使用的 **YUM** 兼容。

如需更多信息，请参阅使用 DNF 工具管理软件。

第 4 章 新功能

这部分论述了 Red Hat Enterprise Linux 9.3 中引入的新功能和主要改进。

4.1. 安装程序和镜像创建

支持 AWS EC2 镜像的传统的和 UEFI 引导

在以前的版本中，RHEL 镜像构建器创建 EC2 AMD 或 Intel 64 位架构 AMI 镜像，但只支持旧的引导类型。因此，无法利用需要 UEFI 引导的某些 AWS 功能，如安全引导。此增强扩展了 AWS EC2 AMD 或 Intel 64 位架构 AMI 镜像，以支持 UEFI 引导，以及旧的 BIOS 引导。因此，现在可以利用需要带有 UEFI 引导的镜像的 AWS 功能。

Jira:RHELDOCS-16339[1]

新的引导选项 inst.wait_for_disks= 为加载 Kickstart 文件或内核驱动程序添加了等待时间

有时，在引导过程中可能需要过几秒钟才能从带有 **OEMDRV** 标签的设备加载 Kickstart 文件或内核驱动程序。要调整等待时间，您现在可以使用新的引导选项 **inst.wait_for_disks=**。使用这个选项，您可以指定安装前要等待多少秒。默认时间被设置为 **5** 秒，但您可以使用 **0** 秒来最小化延迟。有关这个选项的更多信息，请参阅 存储引导选项。

Bugzilla:2171811

在使用 GUI 和 TUI 在 ARM 上安装 RHEL 时能够选择所需的内核

在以前的版本中，您只能使用 Kickstart 方法在带有 kernel-64k 页大小的 ARM 上安装 RHEL。有了此更新，您现在可以使用 GUI 或 TUI 在 ARM 上安装 RHEL，并选择所需的内核版本。选择所需内核的选项在 Kernel Options 下的 Software Selection 屏幕上提供。

Bugzilla:2164819[1]

支持 VMware VSphere (OVA)

此更新添加了对使用 RHEL 镜像构建器构建 VMware VSphere OVA 文件的支持。Open Virtual Appliance(OVA)文件是 VMware VSphere 虚拟化应用程序使用的虚拟设备。OVA 文件包含用于描述虚拟机的文件，如 OVF 描述符文件、一个或多个虚拟机磁盘镜像文件(VMDK)、可选的清单(MF)和证书文件。通过使用 VMware VSphere (.ova)，您可以使用 vSphere GUI 客户端更轻松地将镜像部署到 VMware vSphere。在引导镜像前，您可以进一步自定义生成的虚拟机。

Jira:RHELDOCS-16877[1]

新的 network Kickstart 选项来控制 DNS 处理

现在，您可以使用带有以下新选项的 **network** Kickstart 命令控制 DNS 处理。通过 **--device** 选项使用这些新选项。

- **--ipv4-dns-search** 和 **--ipv6-dns-search** 选项允许您手动设置 DNS 搜索域。这些选项镜像其 NetworkManager 属性，例如：

```
network --device ens3 --ipv4-dns-search domain1.example.com,domain2.example.com
```

- **--ipv4-ignore-auto-dns** 和 **--ipv6-ignore-auto-dns** 选项允许您忽略 DHCP 中的 DNS 设置。它们不需要任何参数。

Bugzilla:2065754[1]

最小 RHEL 安装现在只安装 s390utils-core 软件包

在 RHEL 8.4 及之后的版本中，**s390utils-base** 软件包被分成 **s390utils-core** 软件包，以及一个辅助 **s390utils-base** 软件包。因此，将 RHEL 安装设置为 **minimal-environment** 只安装必要的 **s390utils-core** 软件包，而不是辅助 **s390utils-base** 软件包。如果要在最小 RHEL 安装中使用 **s390utils-base** 软件包，您必须在完成 RHEL 安装后手动安装软件包，或使用 Kickstart 文件显式安装 **s390utils-base**。

Bugzilla:1932480[1]

4.2. 安全性

Keylime rebase 到版本 7.3.0

Keylime 软件包已更新至上游版本 7.3.0。这个版本提供各种改进和 bug 修复。最值得注意的是，allow 和 exclude 列表被合并到 Keylime 运行时策略中。您可以使用 **convert_runtime_policy.py** 脚本合并这两个列表。

另外，更新修复了两个中等影响评级的漏洞： CVE-2023-38200 和 CVE-2023-38201。

Jira:RHEL-476[1]

Keylime 的端口在 SELinux 策略中有严格的规则

Keylime 使用的端口现在在 Keylime SELinux 策略中被标记为 **keylime_port_t**。策略现在允许具有此标签的端口的 TCP 连接。这是因为之前的 Keylime SELinux 策略允许连接到所有未定义端口，而且 Keylime 使用的大多数端口也位于 undefined 组中。因此，这个更新会增加 Keylime SELinux 策略的粒度，端口安全性可以更严格，更有针对性。

Jira:RHEL-595[1]

审计现在支持 FANOTIFY 记录字段

这个 **audit** 软件包的更新引进了对 **FANOTIFY** 审计记录字段的支持。审计子系统现在在 **AUDIT_FANOTIFY** 记录中记录其他信息，特别是：

- 指定 **FANOTIFY** 事件的类型的 **fan_type**
- 指定其他上下文信息的 **fan_info**
- 指示事件中涉及的主题和对象的信任级别的 **sub_trust** 和 **obj_trust**

因此，您可以更好地理解为什么在某些情况下审计系统会拒绝访问。这可帮助您为 **fapolicyd** 框架等工具编写策略。

Jira:RHELPLAN-161087[1]

fapolicyd 现在为故障排除提供规则号

有了这个增强，新的内核和审计组件允许 **fapolicyd** 服务发送导致拒绝 **fanotify** API 的规则号。因此，您可以更精确地对与 **fapolicyd** 相关的问题进行故障排除。

Jira:RHEL-624

crypto-policies 现在在 FIPS 模式下为 TLS 1.2 连接提供 NO-ENFORCE-EMS 子策略

系统范围的加密策略现在包含 **NO-ENFORCE-EMS** 子策略。应用新子策略后，对于在 FIPS 模式下协商的所有 TLS 1.2 连接，系统不再需要 Extended Master Secret (EMS)扩展(RFC 7627)。这可允许系统连接

到不支持 EMS 或 TLS 1.3 的旧系统。请注意，这违反了 FIPS-140-3 标准的要求。您可以通过输入 **update-crypto-policies --set FIPS:NO-ENFORCE-EMS** 来应用子策略。

Bugzilla:2216257[1]

在 FIPS 模式下，GnuTLS 需要带有 TLS 1.2 的 EMS

为了遵守 FIPS-140-3 标准，对于在 FIPS 模式下协商的所有 TLS 1.2 连接，GnuTLS 服务器和客户端需要 Extended Master Secret (EMS)扩展(RFC 7627)。如果您的场景需要保持与不支持 EMS 且无法使用 TLS 1.3 的旧服务器和客户端的兼容性，您可以应用 **NO-ENFORCE-EMS** 系统范围的加密策略：

```
# update-crypto-policies --set FIPS:NO-ENFORCE-EMS
```

警告

如果您允许没有 EMS 的 TLS 1.2 连接，则您的系统将不再满足 FIPS-140-3 要求。

Bugzilla:2157953

NSS 现在在 FIPS 模式下强制实施 EMS

网络安全服务(NSS)库现在包含 **TLS-REQUIRE-EMS** 策略，来要求所有 TLS 1.2 连接的 Extended Master Secret (EMS)扩展(RFC 7627)，如 FIPS 140-3 标准强制的那样。当系统范围的加密策略被设置为 **FIPS** 时，NSS 使用新策略。

如果您的场景需要与不支持 EMS 或 TLS 1.3 的旧系统进行交互，您可以应用 **NO-ENFORCE-EMS** 系统范围的加密策略。此更改违反了 FIPS-140-3 要求。

Bugzilla:2157950

OpenSSL 现在支持在 FIPS 模式下禁用 EMS

现在，您可以通过编辑 **/etc/pki/tls/fips_local.cnf** 文件，在 FIPS 模式下将 OpenSSL 加密库配置为允许没有 Extended aster Secret (EMS)扩展(RFC 7627)的 TLS 1.2 连接。在您选择的文本编辑器中，在配置文件中添加以下部分：

```
[fips_sect]
tls1-prf-ems-check = 0
activate = 1
```

然后，在 **/etc/pki/tls/openssl.cnf** 文件中找到 SSL 配置部分。默认 SSL 配置部分是 **crypto_policy**。在 SSL 配置部分的末尾，添加以下行：

```
Options=RHNoEnforceEMSinFIPS
```

以前的配置更改允许 FIPS 模式的系统连接到不支持 EMS 或 TLS 1.3 的旧系统。

警告

您可以通过输入 **update-crypto-policies --set FIPS:NO-ENFORCE-EMS** 命令来停止在 FIPS 模式下为 TLS 1.2 强制使用 EMS。在这两种情况下，此类配置更改违反了 FIPS-140-3 标准的要求。

Bugzilla:2216256[1]

OpenSSH 进一步强制实施 SHA-2

出于加密目的，作为从不太安全的 SHA-1 消息摘要中进一步迁移努力的一部分，OpenSSH 中进行了以下更改：

- 对 **sshd** 启动添加了一个检查，检查是否在系统上配置了使用 SHA-1。如果不可用，OpenSSH 不会尝试对操作使用 SHA-1。这可消除在存在 DSS 密钥时加载它们，并在**rsa-sha2** 组合可用时强制发布这些组合。
- 在 SSH 私钥转换中，OpenSSH 明确使用 SHA-2 测试 RSA 密钥。
- 当 SHA-1 签名在服务器端不可用时，**sshd** 使用 SHA-2 来确认主机密钥证明。这可能与 RHEL 8 及更早版本上的客户端不兼容。
- 当客户端上 SHA-1 算法不可用时，OpenSSH 使用 SHA-2。
- 在客户端上，当 SHA-1 在密钥证明请求中使用或未指定哈希算法（假设默认）时，OpenSSH 允许来自服务器的基于 SHA2 的密钥证明。这与 RSA 证书已存在的异常一致，并允许在支持时使用现代算法进行连接。

Bugzilla:2070163

OpenSSL 现在包含针对类似 Bleichenbacher 的攻击的保护

这个 OpenSSL TLS 工具包发行版本引入了类似对 RSA PKCS #1 v1.5 解密过程的 Bleichenbacher 攻击的保护。如果 RSA 解密在 PKCS #1 v1.5 解密过程中检测到一个错误，则它现在返回一个随机生成的确定性消息，而不是一个错误。这个变化提供了对漏洞的通用保护，如 CVE-2020-25659 和 CVE-2020-25657。

您可以通过对 RSA 解密上下文调用 **EVP_PKEY_CTX_ctrl_str （ctx, "rsa_pkcs1_implicit_rejection"."0"）** 函数来禁用这个保护，但这会使您的系统更易受攻击。

Bugzilla:2153471

OpenSSL 现在支持通过 Groups 选项配置 Brainpool 曲线

这个 OpenSSL TLS 工具包的更新引进了对 Elliptic Curve Cryptography (ECC)中 Brainpool 曲线的支持。另外，您可以通过 **Groups** 配置选项使用系统范围的加密策略控制曲线。

OpenSSL ECC 中启用了以下 Brainpool 曲线：

- **brainpoolP256r1**
- **brainpoolP256t1**

- **brainpoolP320r1**
- **brainpoolP320t1**
- **brainpoolP384r1**
- **brainpoolP384t1**
- **brainpoolP512r1**
- **brainpoolP512t1**

Bugzilla:2188180

crypto-policies 现在支持 OpenSSL ECC Brainpool 曲线

有了此系统范围加密策略的更新，您现在可以使用 **group** 选项控制 OpenSSL 中的以下 Brainpool Elliptic Curve Cryptography (ECC)曲线：

- **BRAINPOOL-P256R1**
- **BRAINPOOL-P384R1**
- **BRAINPOOL-P512R1**.

例如，您可以通过创建一个包含以下行的子策略，来在 OpenSSL 中启用所有支持的 Brainpool elliptic 曲线：

```
group = BRAINPOOL-*+
```

Bugzilla:2193324[1]

crypto-policies 现在默认使用与 OpenSSL 相同的组顺序

在这个发行版本中，系统范围的加密策略(**crypto-policies**)在 OpenSSL **Groups** 配置选项中控制组顺序。要在 OpenSSL 中保留性能，**crypto-policies** 使用与 OpenSSL 内置首选项的顺序匹配的默认组顺序。因此，支持 **crypto-policies** 控制组顺序的 RHEL 加密后端，如 GnuTLS，现在使用与 OpenSSL 相同的顺序。

Jira:RHEL-591[1]

crypto-policies permitted_enctypes 不再在 FIPS 模式下破坏复制

在此更新前，运行在 RHEL 8 上的 IdM 服务器发送一个 AES-256-HMAC-SHA-1- 加密服务票据，该票据表示在 FIPS 模式下运行 RHEL 9 的一个 IdM 副本。因此，默认的 **permitted_enctypes krb5** 配置破坏了 RHEL 8 IdM 服务器和 FIPS 下 RHEL 9 IdM 副本之间的复制。

这个系统范围的加密策略的更新会重新排序 **permitted_enctypes krb5** 配置选项值，以允许默认优先处理更具互操作性的加密类型。因此，**permitted_enctypes** 配置不再破坏 RHEL 8 IdM 服务器和 FIPS 模式下 RHEL 9 IdM 副本之间的复制。

注意

如果使用 Kerberos，请验证 **/etc/crypto-policies/back-ends/krb5.config** 文件中 **permitted_enctypes** 的值的顺序。如果您的场景需要不同的顺序，请应用一个自定义加密子策略。

Bugzilla:2225222

pcsc-lite-ccid rebase 到 1.5.2

pcsc-lite-ccid 软件包已更新至版本 1.5.2。此版本提供各种程序错误修复和增强，最重要的是：

- 对新读卡器的支持
- 为 Alcor Micro AU9560 进行了修复

Bugzilla:2209457

OpenSC rebase 到 0.23

opensc 软件包已更新至版本 0.23。此版本提供各种程序错误修复和增强，最重要的是：

- 添加了对使用对称密钥的加密和解密的支持
- 添加了对长度超过 512 字节的签名数据的支持
- 默认禁用了旧的卡驱动程序支持
- 删除了对旧驱动程序 MioCOS 和 JCOP 的支持

Jira:RHEL-280[1]

setools rebase 到 4.4.3

setools 软件包已更新至版本 4.4.3。此版本提供各种程序错误修复和增强，最重要的是：

- 修复了使用 Cython 3.0.0 的编译
- 改进了手册页
- 删除了 **sediff**、**sesearch** 和 **apol** 中未使用的选项
- 在 **seinfoflow** 命令中添加了 **-r** 选项，以便获得对源类型的流分析
- 没有权限的规则会作为无效策略自动被拒绝

Bugzilla:2231801, Bugzilla:2184140

SELinux 策略中限制的其他服务

此更新将额外的规则添加到限制以下 **systemd** 服务的 SELinux 策略中：

- **qat**
- **systemd-pstore**
- **boothd**
- **fdo-manufacturing-server**
- **fdo-rendezvous-server**
- **fdo-client-linuxapp**
- **fdo-owner-onboarding-server**

因此，这些服务不会再使用 **unconfined_service_t** SELinux 标签运行，并可在 SELinux enforcing 模式下成功运行。

Bugzilla:2080443[1], Bugzilla:2026795,Bugzilla:2181565,Bugzilla:2128833

OpenSCAP rebase 到 1.3.8

OpenSCAP 软件包已更新到上游版本 1.3.8。此版本提供各种程序错误修复和增强，最重要的是：

- 修复了 **systemd** 探测，使其不忽略某些 **systemd** 单元
- 向 **shadow** OVAL 探测添加了离线功能
- 向 **sysctl** OVAL 探测添加了离线功能
- 向网络文件系统列表中添加了**uristorfs**
- 为 **autotailor** 工具生成的定制文件创建了一个临时解决方案

Bugzilla:2217442

SCAP 安全指南 rebase 到版本 0.1.69

SCAP 安全指南(SSG)软件包已 rebase 到上游版本 0.1.69。这个版本提供各种改进和 bug 修复。最值得注意的是，它引入了与西班牙国家加密中心于 2022 年 10 月发布的 CCN-STIC-610A22 指南一致的 RHEL 9 的三个新 SCAP 配置文件：

配置文件名称	配置文件 ID	策略版本
CCN Red Hat Enterprise Linux 9 - 高级	**xccdf_org.ssgproject.content_profile_ccn_advanced**	2022-10
CCN Red Hat Enterprise Linux 9 - 基本	**xccdf_org.ssgproject.content_profile_ccn_basic**	2022-10
CCN Red Hat Enterprise Linux 9 - 中级	**xccdf_org.ssgproject.content_profile_ccn_intermediate**	2022-10

Bugzilla:2221697

ANSSI-BP-028 安全配置文件已更新至版本 2.0

SCAP 安全指南中的以下法国信息系统安全局(ANSSI) BP-028 已更新，以与版本 2.0 保持一致：

- ANSSI-BP-028 最低级别
- ANSSI-BP-028 中间级别
- ANSSI-BP-028 增强级别
- ANSSI-BP-028 高级别

Bugzilla:2155790

python3-greenlet-devel 现在在 CRB 中 提供

python3-greenlet-devel 软件包现在在 CodeReady Linux Builder (CRB)存储库中提供，您必须明确启用该存储库。如需更多信息，请参阅 如何启用和使用 CodeReady Linux Builder 中的内容 知识库文章。请注意，不支持 CRB 存储库中包含的软件包。

Bugzilla:2149497

SSG 规则，用来检查 pam_wheel.so 模块使用的组是否已简化

CIS Benchmark 需要限制 **su** 命令，而使用 **sudo** 命令。SCAP 安全指南(SSG)通过 **pam_wheel.so** 模块来满足这个要求，该模块将 **su** 命令限制到特定的组。此更新改进了检查这个组是否存在且没有成员的规则。因此，规则效率更高，并简化了评估报告的解释。

Jira:RHEL-1905

4.3. RHEL FOR EDGE

提供了新的 FIDO Device Onboarding Servers 容器镜像

以下用于加入 IoT 和边缘计算设备的 FIDO Device Onboarding Servers 容器镜像在 红帽容器目录 中提供：

- rhel9/fdo-manufacturing-server 容器镜像
- rhel9/fdo-owner-onboarding-server 容器镜像
- rhel9/fdo-rendezvous-server 容器镜像
- rhel9/fdo-serviceinfo-api-server 容器镜像

Jira:RHELPLAN-163133[1]

minimal-raw 镜像类型现在支持 64 位 ARM 架构

有了这个增强，您可以创建一个支持 64 位 ARM 架构以及 AMD 和 Intel 64 位构架的 **minimal-raw** 镜像类型。**minimal-raw** 镜像是预先打包的、可引导的、最小 RPM 镜像，以 **xz** 格式压缩。要引导镜像，您必须解压缩它，并将其复制到任何可引导设备上，如 SD 卡。要解压缩镜像，请运行以下命令：

```
$ xz -d <_uuid-minimal-raw.img_.xz>
```

Jira:RHELPLAN-163665[1]

现在支持 Commit ID 作为 composer-cli CLI 的 --parent 参数的值

现在，您可以将镜像 Commit ID 用作 **composer-cli** 命令行 **--parent** 参数的值。要获取镜像 Commit ID，请下载并提取 RHEL for Edge Commit 镜像。您可以在提取的 **.tar** 文件中找到 **ref** 名称和提交 ID。

Jira:RHELDOCS-16386[1]

对构建 RHEL for Edge .ami 镜像的支持

有了此增强，您可以使用内部 RHEL 镜像构建器为 RHEL for Edge 构建 **.ami** 镜像。在初始引导过程中，您可以使用 Ignition 自定义蓝图，来将凭证注入到镜像中。您可以将 **.ami** 镜像上传到 AWS，并在 AWS 中引导 EC2 实例。

Jira:RHELDOCS-16708[1]

支持为 RHEL for Edge 构建 .vmdk 镜像

有了此增强，您可以使用内部 RHEL 镜像构建器为 RHEL for Edge 构建 **.vmdk** 镜像。您可以使用 Ignition 自定义蓝图，以便在初始引导过程中将凭证注入到镜像中。您可以在 vSphere 上载入镜像，并在虚拟机 vSphere 中引导镜像。该镜像与 ESXi 7.0 U2、ESXi 8.0 及之后的版本兼容。虚拟机与版本 19 和 20 兼容。

Jira:RHELDOCS-16709[1]

现在，您可以在不设置密码的情况下以初始用户身份登录到 Edge 系统

在以前的版本中，以 FDO 载入过程中创建的初始用户身份登录无法正常工作，因为系统要求的密码没有使用 **useradd** 命令设置。有了这个增强，密码现在设置为可选，即使您之前没有使用 **useradd** 命令设置密码，您也可以登录。请注意，您可以在不输入密码的情况下使用 SSH 密钥登录，如果失败，会提示您输入密码。

Jira:RHELDOCS-17101[1]

4.4. 软件管理

升级后自动重启新的 DNF Automatic reboot 选项

有了这个增强，您可以使用 DNF Automatic **reboot** 选项将您的系统设置为自动重启，以便在升级后应用更改。

reboot 选项支持以下设置：

- **never** 不会重启系统。这是当前的行为。
- **when-changed** 会在任何升级后触发重启。
- **when-needed** 只有在需要重启以应用更改时才触发重启，例如当 systemd 或内核被升级时。

您可以使用 **reboot_command** 选项自定义用于重启的命令。默认重启命令为 **shutdown -r**。

Bugzilla:2124793

新的 --poweroff 选项允许您在安装更新后关闭系统

有了此增强，新的 **--poweroff** 选项已添加到 **dnf system-upgrade** 插件的 **reboot** 命令中。您可以在安装更新而不是重启后使用这个选项来关闭系统。

Bugzilla:2157844

新的 dnf leaves 和 show-leaves 插件现在可用于 DNF API

有了此增强，提供了以下新的 DNF 插件，其列出了系统上安装的不是作为其他安装的软件包的依赖项所需的软件包：

- **dnf leave** 列出了所有软件包。
- **show-leaves** 列出了新安装的软件包和事务后作为其他已安装软件包的依赖项不需要的软件包。

Bugzilla:2134638

4.5. SHELL 和命令行工具

NetBackup 服务现在为备份恢复启用

使用 NetBackup (NBU)备份方法时，ReaR 现在在救援镜像中包含 NetBackup 服务版本 10.1.1 的单元文件，并在救援系统引导时启动它们。因此，您可以在恢复过程中使用 NBU 备份方法恢复系统备份，并成功完成恢复。

Bugzilla:2188593

opencryptoki rebase 到 3.21.0

opencryptoki 软件包已 rebase 到版本 3.21.0，它提供很多改进和 bug 修复。最值得注意的是，**opencryptoki** 现在支持以下功能：

- 并发硬件安全模块(HSM)主密钥更改
- **protected-key** 选项，来将所选密钥转换为受保护的密钥
- 其他密钥类型，如 DH、DSA 和通用 secret 密钥类型
- EP11 主机库版本 4
- AES-XTS 密钥类型
- 特定于 IBM 的 Kyber 密钥类型和机制
- 其他特定于 IBM 的 Dilithium 密钥第 2 轮和第 3 轮变体

另外，**pkcsslotd** 插槽管理器不再以 root 用户身份运行， **opencryptoki** 提供了进一步强化。有了此更新，您也可以使用以下一组新命令：

p11sak set-key-attr
: 要修改密钥

p11sak copy-key
: 要复制密钥

p11sak import-key
: 要导入密钥

p11sak export-key
: 要导出密钥

Bugzilla:2160061[1]

更新的 systemd-udevd 将一致的网络设备名称分配给 InfiniBand 接口

RHEL 9 中引入的 **systemd** 软件包的新版本包含更新的 **systemd-udevd** 设备管理器。设备管理器将 InfiniBand 接口的默认名称更改为 **systemd-udevd** 选择的一致性名称。

您可以根据 使用 systemd 链接文件重命名 IPoIB 设备 流程来为命名 InfiniBand 接口定义自定义命名规则。

有关命名方案的详情，请查看 **systemd.net-naming-scheme (7)** 手册页。

Bugzilla:2136937

4.6. 基础架构服务

Postfix 现在支持 SRV 查找

有了这个增强，您现在可以使用 Postfix DNS 服务记录解析(SRV)来自动配置邮件客户端并平衡服务器的负载。另外，您可以通过在 Postfix 配置中使用以下与 SRV 相关的选项来防止由临时 DNS 问题或错误配置的 SRV 记录导致的邮件发送中断：

use_srv_lookup

您可以使用 DNS SRV 记录为指定的服务启用发现。

allow_srv_lookup_fallback

您可以使用级联方法查找服务。

ignore_srv_lookup_error

即使 SRV 记录不可用或遇到错误，您也可以确保服务发现仍然可以正常工作。

Bugzilla:2134789

通用 LF-to-CRLF 驱动程序在 cups-filters 中可用

有了这个增强，您现在可以使用通用 LF-to-CRLF 驱动程序，它为接受带有 CR+LF 字符的文件的打印机将 LF 字符转换为 CR+LF 字符。回车(CR)和换行(LF)是标记行末尾的控制字符。因此，使用这个驱动程序，您可以将来自应用程序的 LF 字符终止的文件发送到只接受 CR+LF 字符的打印机。通用 LF-to-CRLF 驱动程序是 RHEL 7 中 **text-only** 驱动程序的重命名版本。新名称反映了其实际功能。

Bugzilla:2229784

4.7. 网络

ARM 上的 RHEL 现在在 RHEL 9.3 中完全支持 wifi 适配器

有了这个增强，您可以启用对 **arm64** 平台的多个卡的 wifi 适配器的访问。

有关配置 wifi 连接的详情，请参阅 管理 wifi 连接。

Bugzilla:2208365[1]

NetworkManager 现在支持 resolv.conf 中的 no-aaaa 选项

NetworkManager 现在支持在 resolv.conf 文件中添加 **no-aaaa** DNS 选项。通过在 DNS 选项设置中使用 **no-aaaa** 值，您可以禁用 IPv6 DNS 解析。

Bugzilla:2176137

nmstate 现在支持混合静态 DNS 搜索和动态 DNS 名称服务器

nmstate 框架现在支持静态域名系统(DNS)搜索域和动态 DNS 名称服务器，这些服务器是 **nmstate** 从动态主机配置协议(DHCP)或 **autoconf** 机制获得的。在以前的版本中，静态 DNS 搜索域无法与动态 DNS 名称服务器共存，因为动态配置被 **nmstate** 丢弃了。这通常导致网络设置和管理中不必要的复杂性和限制。此功能增强旨在在管理 DNS 配置方面带来更多的灵活性。因此，**nmstate** 会尝试查找网络接口，以按照以下顺序存储 DNS 配置：

1. 首选接口，当前包含 DNS 配置，仍对 DNS 有效
2. 自动接口
3. 启用了 IP 的接口

请注意，这个增强不会删除从 DHCP 中学到的 DNS 名称服务器。

以下是应用此功能的 YAML 文件的一个示例：

```
---
dns-resolver:
  config:
    search:
      - example.com
      - example.org
interfaces:
  - name: eth1
    type: ethernet
    state: up
    ipv4:
      enabled: true
      dhcp: true
    ipv6:
      enabled: true
      dhcp: true
      autoconf: true
```

Bugzilla:2179916

nmstate 现在支持 bridge.vlan-default-pvid NetworkManager 配置选项

有了此更新，您可以使用 **nmstate** 框架来配置 **bridge.vlan-default-pvid** NetworkManager 配置选项。通过使用这个选项，您可以在使用 Linux 网桥 VLAN 过滤时，为支持 VLAN 的桥接接口上的未标记流量设置默认端口 VLAN 标识符(PVID)。为此，请使用以下 YAML 配置：

```
interfaces:
  - name: linux-br0
    type: linux-bridge
    state: up
    bridge:
      options:
        vlan-default-pvid: 5
      port:
        - name: eth1
          stp-hairpin-mode: false
          stp-path-cost: 100
          stp-priority: 32
          vlan:
            mode: access
            tag: 100
```

请注意，**bridge.vlan-default-pvid** 的默认值为 1。当设置为 0 启用了 VLAN 过滤时，未标记的流量将被丢弃。

Bugzilla:2180795

NetworkManager 服务在 dbus 服务重启后立即重启

在以前的版本中，由于某些原因重启 **dbus** 后，**NetworkManager** 会停止。这个行为不是最佳的，导致了连接丢失。因此，这个增强更新了 **NetworkManager**，使其更加强大，并使其在 **dbus** 重启时自动重启。

Bugzilla:2161915

nm-cloud-setup 工具现在支持 IMDSv2 配置

用户可以使用 **nm-cloud-setup** 工具配置带有实例元数据服务版本 2 (IMDSv2)的 AWS Red Hat Enterprise Linux EC2 实例。为了遵守改进的安全性，其限制对 EC2 元数据和新功能的未经授权的访问，需要 AWS 和红帽服务间的集成来提供高级功能。此功能增强使 **nm-cloud-setup** 工具能够获取并保存 IMDSv2 令牌，验证 EC2 环境，并使用安全 IMDSv2 令牌检索有关可用接口和 IP 配置的信息。

Bugzilla:2151986

当使用已弃用的 ifcfg 格式时，NetworkManager 会发出通知

ifcfg 格式的连接配置文件已在 RHEL 9 中被弃用（请参阅 **ifcfg** 格式的 NetworkManager 连接配置文件已弃用）。有了此更新，NetworkManager 会通知用户有关此格式的弃用：

- 如果 NetworkManager 在 **/etc/sysconfig/network-scripts/** 目录中处理**ifcfg** 格式的连接配置文件，它会在**systemd** 日志中记录以下警告：

  > Warning: the ifcfg-rh plugin is deprecated, please migrate connections to the keyfile format using "nmcli connection migrate"

- 如果您试图修改一个 **ifcfg** 格式不支持的属性，**nmcli** 工具会报告以下错误：

  > Error: Failed to modify connection '<name>': failed to update connection: The ifcfg-rh plugin doesn't support setting '<property>'. If you are modifying an existing connection profile saved in ifcfg-rh format, please migrate the connection to keyfile using 'nmcli connection migrate <connection_uuid>' or via the Update2() D-Bus API and try again.

由于这些增强，如果用户仍然使用或修改已弃用的 **ifcfg** 格式的连接配置文件，NetworkManager 现在会通知用户。

有关将配置文件从 **ifcfg** 迁移到 keyfile 格式的详情，请参考 将 NetworkManager 配置文件从 ifcfg 迁移到 keyfile 格式。

Bugzilla:2190375

NetworkManager 现在在绑定配置中支持 lacp_active 选项

通过使用 **NetworkManager**，绑定配置中的 **lacp_active** 选项对链路聚合控制协议数据单元(LACPDU)帧提供精细控制。**lacp_active** 选项还调整了 LACPDU 帧的行为，并在绑定设置中控制这些帧的定期传输。要自定义网络配置，您可以通过将 **lacp_active** 设置为 **ON** 或 **OFF** 来启用或禁用 LACPDU 帧的定期传输。

Bugzilla:2069001

NetworkManager 现在支持为绑定接口配置 ns_ip6_target 选项

此增强通过在 **NetworkManager** 中为绑定接口的 **ns_i6_target** 选项的配置指定最多 16 个 IPv6 地址作为监控对等点，来允许设置 **arp_interval** 选项。在以前的版本中，无法在 **NetworkManager** 中指定 IPv6 监控对等点。有了此更新，您可以使用 **nmcli** 工具在 **bond.options** 参数中配置 **ns_ip6_target** 选项。**NetworkManager** 通过启用最多 16 个 IPv6 地址的规范来将此设置应用到绑定接口。此增强同样适用于 IPv4 和 IPv6 设置。

Bugzilla:2069004

NetworkManager 现在支持同一网络接口上的静态和 DHCP IP 配置

通过使用 **nmstate** 工具，您现在可以在 DHCP 或启用了 Ad-Hoc Network Autoconfiguration (autoconf) 的接口上分配一个带有 **dhcp: true** 或 **autoconf: true** 值的静态 IP 地址。

有了此增强，**nmstate** 支持 IP 地址的两个属性：

- **valid_lft** 表示有效的生命周期（以秒为单位）
- **preferred_lft** 表示首选的生命周期（以秒为单位）

两个参数的默认值是 **forever** 表示静态。

有了上述属性，**nmstate** 可以忽略基于 DHCP/autoconf 的 IP 地址，以避免在应用查询的状态后将动态 IP 地址转换为静态 IP。如果您的场景需要禁用带有动态 IP 地址的 DHCP/autoconf 设置，则 **nmstate** 会将这些动态 IP 转换为静态 IP 地址。

Bugzilla:2177733

nmstate 支持 MAC 地址可识别的网络接口

nmstate 工具支持直接到具有 MAC 地址而不是接口名称的网络接口的网络配置。

此增强引入了基本接口的两个属性：

- **identifier**：标识网络上的 **name** 或 **mac-address**。默认值为 **name**。
- **profile-name**：字符串

当 **identifier** 变量设置为 **mac-address** 值时，**nmstate** 使用 **interface.mac-address** 而不是**interface.name**来为特定的网络状态选择网络接口。当存储网络配置时，如果没有分配 **interface.profile-name** 变量，则 **nmstate** 优先选择 **interface.profile-name** 而不是 **interface.name**。如果您检查当前的网络状态，如果 interface.profile-name 等于 **interface.name**，则 **interface.profile-name** 会保持隐藏状态。

Bugzilla:2183214

NetworkManager 支持定义在多少次 ARP 检查失败后，绑定驱动程序将端口标记为 down

此增强将**arp_missed_max** 选项添加到 NetworkManager 中的绑定连接配置文件中。如果您使用地址解析协议(ARP)监控器来检查绑定的端口是否已启动，您可以设置 **arp_missed_max** 来定义多少次检查失败后，绑定驱动程序将端口标记为 down。

Bugzilla:2148684

NetworkManager 支持指定与链接相关的属性

此增强在 NetworkManager 连接配置文件中添加了以下网络链接属性：

- **link.tx-queue-length** - 传输(TX)队列长度的大小，以数据包数为单位。
- **link.gro-max-size** - 设备接受的通用接收卸载(GRO)数据包的最大大小，以字节为单位。
- **link.gso-max-segments** - 设备接受的通用段卸载(GSO)数据包的最大段数。
- **link.gso-max-size** - GSO 数据包的最大大小，以字节为单位。

在以前的版本中，您只能使用 **ip** 命令，或使用 NetworkManager 分配程序脚本中的此类命令来配置这些内核设置。有了这个增强，您可以直接在连接配置文件中配置这些设置。

请注意，NetworkManager 仅在连接配置文件中支持**keyfile** 格式的这些属性，而不是已弃用的 **ifcfg** 格式。

Bugzilla:2158328

nmstate API 支持 dhcp-send-hostname 和 dhcp-custom-hostname DHCP 选项

有了此增强，**nmstate** 工具支持在连接文件中以下两个 DHCP 选项的配置：

- **dhcp-send-hostname**:**true** 或 **false** 值。如果 DHCP 请求需要主机名或完全限定域名(FQDN)选项，则会从该选项设置主机名。默认值是 **true**。
- **dhcp-custom-hostname**: <string>。使用这个选项在 DHCP 请求中配置主机名或 FQDN 选项，值类型是字符串。

  对于 DHCPv4 网络协议

- 如果主机名是 FQDN，请参阅 RFC 4702 中的 **完全限定域名(FQDN)**，选项(81)。
- 如果主机名不是 FQDN，请参阅 RFC 2132 中的 **主机名**, 选项(12)。

  对于 DHCPv6 网络协议

  支持自定义字符串，空域名，覆盖 DHCP 请求的主机名。请参阅 RFC 4704 中的 **完全限定域名(FQDN)**, 选项(29)。

Bugzilla:2187622

NetworkManager rebase 到版本 1.44.0

NetworkManager 软件包已升级到上游版本 1.44.0，与之前的版本相比，它提供了一些改进和 bug 修复：

- 与链接相关的属性已添加到 NetworkManager 中。
- **arp_missed_max**、**lacp_active** 和 **ns_ip6_target** 属性已添加到绑定连接配置文件中。
- 现在，您可以在 **ipv6.dhcp-pd-hint** 连接属性中设置 DHCPv6 前缀委托提示。
- 在 **/etc/NetworkManager/NetworkManager.conf** 文件的 **[keyfile]** 部分中启用新的 **rename** 参数会导致 NetworkManager 在 **/etc/NetworkManager/system-connections/** 中重命名连接配置文件（如果更改了配置文件名称(**connection.id**)）。如果外部应用程序或脚本依赖于文件名，请不要启用此参数。
- 当您设置包含非公共顶级域(TLD)的主机名时，NetworkManager 现在使用此 TLD 作为 DNS 搜索域，而不是完整主机名。
- NetworkManager 现在从 **/etc/NetworkManager/NetworkManager.conf** 文件中的 **[global-dns]** 部分中应用 DNS 选项。
- 为了避免与其他依赖服务的竞争条件，NetworkManager 现在仅在填充 D-Bus 树后获取 D-Bus 名称。请注意，这可能会在 NetworkManager 启动时添加一个延迟。
- NetworkManager 现在向 **Update2（）** D-Bus 调用添加了一个 **version-id** 参数，以防止并发配置文件修改。
- NetworkManager 不再使用临时 IPv6 地址从 DNS 解析系统主机名。
- 为了防止多连接配置文件时的意外行为，NetworkManager 现在跟踪每个设备和连接剩余的自动连接重试次数，而不是每个连接。
- NetworkManager 使用内核的 **netlink** 接口而不是 **sysfs** 文件系统来设置 VLAN 过滤选项。

- **nm-cloud-setup** 工具现在在 Amazon EC2 上支持实例元数据服务版本 2 (IDMSv2)。
- 用户现在可以在 **nmtui** 应用程序中启用和禁用 wifi 和 Wireless Wide Area Networks (WWAN)。
- bond、bridge 和 team 连接现在使用 **/etc/NetworkManager/NetworkManager.conf** 文件的 **[main]** 部分中的 **ignore-carrier=no** 设置。

Bugzilla:2180966

SCTP rebase 到 RHEL 9 的内核网络树的最新版本

Stream Control Transport Protocol (SCTP)网络子系统中的显著变化包括：

- 虚拟路由和转发(VRF)支持复杂网络环境中的段和隔离 SCTP 流量。
- 新的流调度程序(**fair capacity**，**weighted fair queueing**)以确保网络中有效的和相等的资源分配。

Bugzilla:2189292

MPTCP rebase 到 RHEL 9 的内核网络树的最新版本

多路径 TCP (MPTCP)协议扩展中的显著变化包括：

- 支持 TCP fastopen (TFO)扩展，包括客户端支持。此功能为您的网络提供延迟、效率和性能改进。
- 支持多个混合 IPv4/IPv6 子流，以便在同时使用两个 IP 版本的网络中具有更大的灵活性和适应性。

Bugzilla:2193330[1]

xdp-tools 软件包 rebase 到版本 1.4.0

xdp-tools 软件包已升级到 1.4.0 版本，其提供多个 bug 修复和增强。主要变更包括：

- **xdp-bench** 工具获得对多缓冲 eXpress Data Path (XDP)的支持，以及对内核中 **xdp_load_bytes （）** 帮助程序进行基准测试的支持。此功能允许使用大型最大传输单元(MTU)进行网络基准测试。
- 改进了 **xdp-tools** 的命令行工具的锁，以便在工具未完全退出时防止过时的锁。
- **libxdp** 库包含一个新的 **xsk_umem__create_with_fd （）** API，该 API 接受一个已打开的 **AF_XDP** 套接字的额外文件描述符。当进程没有 **CAP_NET_RAW** 特权时，您可以使用此函数替换常规 **xsk_umem__create（）** 函数。

Bugzilla:2218500

iproute rebase 到版本 6.2.0

iproute 软件包已升级到上游版本 6.2.0，与之前的版本相比，它提供了一些改进和 bug 修复。最显著的更改有：

- 新的 **ip stats** 命令管理和显示接口统计信息。默认情况下，**ip stats show** 命令显示所有网络设备（包括网桥和绑定）的统计信息。您可以使用 **dev** 和 **group** 选项过滤输出。详情请查看 **ip-stats (8)** 手册页。
- **ss** 工具现在提供 **-T** (**--threads**)选项来显示线程信息，这扩展了 **-p** (**--processes**)选项。详情请查看 **ss (8)** 手册页。

- 您可以使用新的 **bridge fdb flush** 命令删除与提供的选项匹配的特定转发数据库(fdb)条目。详情请查看 **bridge (8)** 手册页。

Jira:RHEL-428[1]

内核支持以特定顺序激活绑定端口

有了这个增强，如果您在 **active-backup** 中配置了绑定、**balance-tlb** 或 **balance-alb** 模式，则内核的 **netlink** 接口支持在每个端口上设置优先级。优先级值使用 32 位整数，较高的值表示较高的优先级。现在，您可以按特定顺序激活绑定端口。

要使用这个功能，您可以在创建或修改 NetworkManager 端口连接配置文件时设置 **bond-port.prio** 属性来配置优先级。

Bugzilla:2092194[1]

firewalld 现在避免不必要的防火墙规则刷新

随着 RHBA-2023:7748 的发布，建议升级 **firewalld** 服务，如果满足以下条件，则不会从 **iptables** 配置中删除所有现有的规则：

- **firewalld** 使用 **nftables** 后端。
- 没有使用 **--direct** 选项创建的防火墙规则。

这个变化旨在减少不必要的操作（防火墙规则刷新），并改进了与其他软件的集成。

Jira:RHEL-14694[1]

为 VLAN 接口引进新的 nmstate 属性

使用此 **nmstate** 框架的更新，引进了以下 VLAN 属性：

- **registration-protocol**: VLAN 注册协议。有效值为 **gvrp** (GARP VLAN Registration Protocol), **mvrp** (Multiple VLAN Registration Protocol)和 **none**。
- **reorder-headers** ：重新排序输出数据包标头。有效值为 **true** 和 **false**。
- **loose-binding** ：放松接口到其主设备的操作状态的绑定。有效值为 **true** 和 **false**。

您的 YAML 配置文件类似以下示例：

```
---
interfaces:
  - name: eth1.101
    type: vlan
    state: up
    vlan:
      base-iface: eth1
      id: 101
      registration-protocol: mvrp
      loose-binding: true
      reorder-headers: true
```

Jira:RHEL-19142[1]

4.8. 内核

RHEL 9.3 中的内核版本

Red Hat Enterprise Linux 9.3 与内核版本 5.14.0-362.8.1 一起分发。

Bugzilla:2232554

添加了对 NVIDIA Grace CPU 的支持

Red Hat Enterprise Linux 9.3 添加了对 NVIDIA Grace ARM 64 位 CPU 的支持。

Jira:RHELDOCS-17055[1]

RHEL 内核现在支持 AutoIBRS

Automatic Indirect Branch Restricted Speculation (AutoIBRS)是由 AMD EPYC 9004 Genoa 处理器系列以及更新的 CPU 版本提供的一个功能。AutoIBRS 是 Spectre v2 CPU 漏洞的默认缓解方案，它提升了性能，并改进了可扩展性。

Bugzilla:1898184[1]

perf rebase 到版本 6.2

perf 性能分析工具已 rebase 到版本 6.2。除了大量次要 bug 修复和更新外， **perf list** 命令现在显示包含人类可读名称和描述的 Performance Monitor Unit (PMU)事件。另外，这个更新添加了对以下处理器的支持：

- Intel 第 13 代核心处理器(Intel Raptor Lake-S)
- Intel 第 14 代处理器(Intel Meteor Lake)
- Intel 第 5 代 Xeon 服务器处理器(Intel Emerald Rapids)

Bugzilla:2177180[1]

Intel® QAT 内核驱动程序 rebase 到上游版本 6.2

Intel® Quick Assist Technology (QAT)已 rebase 到上游版本 6.2。Intel® QAT 包括针对对称和非对称加密、压缩性能和其他 CPU 密集型任务优化的加速器。

rebase 包括很多 bug 修复和增强。最显著的增强是提供了对用于 QAT GEN4 的以下硬件加速器设备的支持：

- Intel Quick Assist Technology 401xx 设备
- Intel Quick Assist Technology 402xx 设备

Bugzilla:2144528[1]

vTPM 功能可用于 Linux 容器

此增强为 Linux 容器和其它虚拟环境引入了虚拟受信任的平台模块(**vTPM**)。**vTPM** 是 TPM 的一个虚拟化版本，提供一个用于确保运行环境安全的专用的 TPM 实例。使用 **vTPM** 代理驱动程序，程序与模拟 TPM 交互的方式与它们与物理 TPM 交互的方式相同。

因此，每个虚拟机现在可以有一个被隔离和加密的专用 **vTPM** 实例。

Bugzilla:2210263[1]

crash rebase 到版本 8.0.3

crash 是一个交互式工具，用于在内核崩溃时分析正在运行的系统和 **kdump** 创建的内核转储文件。**crash** 工具已更新至 8.0.3 版本，其中包括很多 bug 修复和增强。最显著的改进是增加了 IPv6 支持。

对于支持 IPv6 的网络接口，**crash** 使用 **net** 或 **net -s** 命令打印 IPv6 地址。

- **net** 命令显示网络设备、名称和 IP 地址的列表。
- **net -s** 命令显示以下信息：
  - 开放网络套接字和 sock 地址
  - 系列以及套接字类型和 sock 地址
  - **INET** 和 **INET6** 系列的源和目标地址和端口

Bugzilla:2170283

支持 LVM 精简置备存储卷作为 vmcore 转储目标

kdump 机制现在支持精简配置逻辑卷作为 **vmcore** 目标。要配置 LVM 精简配置，请完成以下步骤：

1. 创建一个 LVM 卷组。

```
vgcreate vg00 /dev/sdb
```

2. 创建一个 10 MB 的 LVM 精简池。

```
lvcreate -L 10M -T vg00/thinpool
```

3. 创建一个具有 300 MB 文件系统空间的 LVM 精简卷。

```
lvcreate -V 300M -T vg00/thinpool -n thinvol
mkfs.ext4 /dev/vg00/thinvol
```

4. 配置 LVM 精简池阈值，以自动扩展空间。

```
cat /etc/lvm/lvm.conf
activation {
 thin_pool_autoextend_threshold = 70
 thin_pool_autoextend_percent = 20
 monitoring = 1
}
```

5. 为第一个内核启用 LVM 精简池监控服务。

```
systemctl enable lvm2-monitor.service
systemctl start lvm2-monitor.service
```

6. 在 **kdump.conf** 文件中附加以下行，来将 LVM 精简卷设置为 **kdump** 目标。

```
ext4 /dev/vg00/thinvol
path /
```

7. 启动 **kdump** 服务。

```
kdumpctl restart
```

8. 通过触发内核 panic 验证配置，并检查 **vmcore** 是否已保存到 **/dev/vg00/thinvol** 。

因此，使用此增强，**kdump** 机制扩展了能力，来在精简置备的存储卷上保存 **vmcore** 转储文件。

Bugzilla:2083475

makedumpfile rebase 到上游版本 1.7.3

通过压缩页或排除不需要的内存页来使崩溃转储文件变小的**makedumpfile** 工具，已从上游版本 1.7.3 rebase 到上游版本 1.7.3。rebase 包括很多 bug 修复和增强。

最显著的变化是在 AMD 和 Intel 64 位构架上为独立转储(**sadump**)机制添加了 5 级分页模式。5 级分页模式扩展了处理器的线性地址宽度，以允许应用程序访问更大的内存。5 级分页将虚拟地址的大小从 48 位扩展到 57 位，并将物理地址从 46 位扩展到 52 位。

Bugzilla:2173815

Red Hat Enterprise Linux 支持 ARM 的 SystemReady ES 和 IR 层

Red Hat Enterprise Linux 现在支持 ARM 的 SystemReady ES 和 IR，而之前只支持 SR 层。在 RHEL 9.3 中，启用了 NVIDIA Orin、NXP i.MX 8M 和 NXP i.MX 8M Mini 模块已启用，并且是 RHEL 硬件认证的候选模块。硬件合作伙伴可以通过在红帽硬件认证之旅注册来 提交认证。客户可以使用目录中列出的受支持硬件来提高生产体验。

Bugzilla:2195986[1]

ARM 上的 RHEL 现在支持蓝牙

有了这个增强，您可以在命令行界面上使用 **bluetoothctl** 工具来配置蓝牙设备。

Bugzilla:2187856[1]

ARM 上的 RHEL 现在在 RHEL 9.3 中完全支持附加了 USB 的相机

此增强为 AMD 和 Intel 64 位构架平台上的 RHEL 启用了 **CONFIG_MEDIA_SUPPORT** 内核配置。现在，您可以在 AMD 和 Intel 64 位构架系统上使用 USB 相机。

Bugzilla:2192722[1]

bpf rebase 到版本 6.3

Berkeley Packet Filter (BPF)工具已 rebase 到 Linux 内核版本 6.3。主要变化和增强包括：

- BPF trampoline 现在在 64 位 IBM Z 构架上可用。
- 新的映射类型 - **BPF_MAP_TYPE_USER_RINGBUF** - 以及相关的帮助程序已为通过特定于 BPF 的环缓冲的用户空间和内核之间的通信进行了定义。
- BPF 现在提供新的复杂的数据结构：链接的列表和 **rbtree**。

- 跟踪程序的 BPF trampoline 现在支持 **struct** 参数。
- BPF 现在提供了一种导出 NIC 支持的 XDP 功能的方法。
- 硬件元数据现在通过使用具有对 RX 哈希和时间戳初始支持的元数据 BPF 内核函数(**kfuncs**)暴露给 XDP 程序。
- BPF 现在提供了一个帮助程序，其在 BPF 程序的新 **conntrack** 模块条目中设置源和目标 NAT 地址和端口。
- BPF 现在可以对 netfilter 数据包过滤框架的 **nf_conn:mark** 连接标记进行直接写。

Bugzilla:2178930[1]

4.9. 引导加载程序

带有 BLS 的 grub2-mkconfig 的新默认行为

在 Boot Loader Specification (BLS)框架中，GRUB 在引导时从 BLS 片断动态生成引导菜单，且不会在 **grub.cfg** 文件中预定义。

在以前的版本中，**grub2-mkconfig** 命令生成一个新的 **grub.cfg** 文件，并总是使用**/etc/default/grub** 文件中发现的 **GRUB_CMDLINE_LINUX** 变量的值覆盖所有 BLS 代码片段中的命令行参数。

有了此版本，**grub2-mkconfig** 命令不再默认使用 **GRUB_CMDLINE_LINUX** 覆盖 BLS 片段中的内核命令行。引导装载程序菜单中的每个内核都从其 BLS 代码段获取其内核命令行。这个新的默认行为是由 **GRUB_ENABLE_BLSCFG=true** 选项造成的。

要重新生成 **grub.cfg**，以便内核忽略 BLS 片断，并从 **GRUB_CMDLINE_LINUX** 中获取命令行，请设置 **GRUB_ENABLE_BLSCFG=false** 选项。

要根据 **GRUB_CMDLINE_LINUX** 更新 BLS 片断中的内核命令行，请添加 **--update-bls-cmdline** 选项：

```
# grub2-mkconfig -o /path/to/grub.cfg --update-bls-cmdline
```

另请注意，您可以使用 **grubby** 为各个内核更改 BLS 段：

```
# grubby --update-kernel /path/to/kernel --args "new args"
```

Jira:RHELDOCS-16752[1]

4.10. 文件系统和存储

NFS 服务器现在为 nfsd 实现了礼貌服务器代码

这个更新在 RHEL 内核 NFS 服务器中为 **nfsd** 引入了礼貌服务器代码的实现。借助这一新功能，NFS 服务器避免为与服务器长时间失去联系的客户端撤销租期，只要客户端在失去联系时不存在访问冲突。

Bugzilla:2180124

DAX 挂载选项和 reflink 现在兼容

有了此更新，重新链接的文件通常与 DAX 模式兼容。文件系统 DAX 挂载选项 **-o dax=always** 与启用了重新链接的文件系统兼容。已重新链接的文件可以使用 inode 标记设置为 DAX 模式。详情请查看 **xfs (5)** 手册页。

Bugzilla:2192730[1]

RPCSEC GSS Kerberos V5 的新加密类型

RPCSEC GSS Kerberos V5 机制现在支持 RFC 6803 (Kerberos 5 的 Camellia 加密)和 RFC 8009 （Kerberos 5 的带有 HMAC-SHA2 的 AES 加密）中定义的加密类型。

添加了以下加密类型：

- **camellia128-cts-cmac**
- **camellia256-cts-cmac**
- **aes128-cts-hmac-sha256-128**
- **aes256-cts-hmac-sha384-192**

这允许 NFS 客户端和服务器在协商 GSS 上下文时使用更强大的加密类型。

Bugzilla:2178741

fuse3 现在允许在不触发 umount的情况下使目录条目无效

有了此更新，在 **fuse3** 软件包中添加了一个新的机制，它允许使目录条目无效，而无需自动触发条目上存在的任何挂载的 **umount**。

Bugzilla:2188182

Stratis 存储管理器现在可用

Stratis 是一个本地存储管理器。它在存储池的上面为用户提供额外的功能：

- 管理快照和精简配置
- 根据需要自动增大文件系统大小
- 维护文件系统
- 池级加密
- TMP2 和 NBDE 支持

要管理 Stratis 存储，使用 **stratis** 工具来与 **stratisd** 后台服务进行通信。

如需更多信息，请参阅 Stratis 文档： 设置 Stratis 文件系统。

Bugzilla:2041558

对 GFS2 文件系统配置和操作的改进

已对 GFS2 文件系统实现了以下更新：

- **mkfs.gfs2** 命令现在支持新的 **-U** 选项，该选项可为您创建的文件系统指定文件系统 UUID。如果省略这个选项，会随机生成文件系统的 UUID。
- **gfs2_jadd** 命令创建日志的速度比之前的版本快得多。
- GFS2 手册页已改进。

Bugzilla:2170017

dmpd rebase 到版本 1.0.2

dmpd 软件包已升级至版本 1.0.2 。主要变更包括：

- 使用 Rust 语言为内存安全重写了工具，并使用多个线程来提高性能。
- 改进了 **thin_check** 和 **cache_check** 工具，以节省 LVM 池激活以及系统启动的时间。与之前的版本相比，这些工具所需的执行时间已提高了十多倍。
- 更新 **thin_dump** 和 **thin_restore** 工具，以避免丢失快照的元数据 **btrees** 的共享。现在，恢复的元数据不需要更多空间。
- 添加新的 **thin_metadata_pack** 和 **thin_metadata_unpack** 工具，来压缩精简元数据，通常压缩到其大小的十分之一。这比通用的压缩器要好。使用这个工具，可以更容易地传递损坏的元数据以进行检查。

Bugzilla:2175198

为 SCSI 设备添加了新的每设备计数器

现在，为 SCSI 更新中的 I/O 超时添加了一个新的每设备计数器 **iotmo_cnt**。除了 I/O 请求的 **iorequest_cnt** 计数外，还可以看到 **iodone_cnt** I/O 完成和 **ioerr_cnt** I/O 错误，请求超时的数量。例如：

```
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iorequest_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iodone_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iotmo_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/ioerr_cnt
```

Bugzilla:2171093[1]

mpathcleanup 清除 device-mapper-multipath 中的多路径设备

mpathcleanup 工具在基于 SCSI 的多路径设备上可以正常工作，并删除了多路径设备以及 SCSI 路径设备。有些用户需要定期删除多路径设备及其路径设备。在以前的版本中，没有删除多路径设备的工具，以及此操作所需的用户定义的脚本。

使用这个新工具，用户现在可以轻松地删除多路径设备及其底层存储，且不需要为此操作创建任何脚本。

Jira:RHEL-782[1]

nvme-cli rebase 到版本 2.4

nvme-cli 软件包已升级到 2.4 版本，其提供多个 bug 修复和增强。主要变更包括：

- 支持通过 TCP 的 TLS。
- 修复了 **systemd** 自动连接服务的不正确的排序，以使用 **/etc/fstab** 文件挂载文件系统。
- 修复了 **u32** 值的打印。
- 正确验证存储标签大小。
- 支持用于光纤控制器的 **nvme effects-log** 命令。

Bugzilla:2159929[1]

4.11. 高可用性和集群

支持对缺少物理卷的 LVM 卷组的故障切换

LVM-activate 资源代理现在支持两个新选项，它们允许在卷组缺少物理卷时进行卷组故障切换：

- **majoritypvs** 选项允许在卷组缺失物理卷时，更改卷组上的系统 ID，只要大多数物理卷存在。
- **degraded_activation** 选项允许在 leg 缺失时激活卷组中的 RAID 逻辑卷，只要 RAID 有足够的设备来在逻辑卷中提供所有数据。

Bugzilla:2174911[1]

IPaddr2 和 IPsrcaddr 集群资源代理现在支持基于策略的路由

IPaddr2 和 **IPsrcaddr** 集群资源代理现在支持基于策略的路由，这可让您配置复杂的路由场景。基于策略的路由要求您配置资源代理的 **table** 参数。

Bugzilla:2142518

Filesystem 资源代理现在支持 EFS 文件系统类型

ocf:heartbeat:Filesystem 集群资源代理现在支持 Amazon Elastic File System (EFS)。现在，您可以在配置 **Filesystem** 资源时指定 **fstype=efs**。

Bugzilla:2142002

在指定克隆 meta 属性时，新的 pcs 解析需要 meta 关键字

为确保 **pcs** 命令格式的一致性，配置克隆 meta 属性，而无需指定 **meta** 关键字的 **pcs resource clone**、**pcs resource promotable** 和 **pcs resource create** 命令现在已弃用。

在以前的版本中，**meta** 关键字在**pcs resource clone** 和 **pcs resource promotable** 命令中被忽略。但是，在 **pcs resource create** 命令中，当 meta 关键字跟在 **clone** 关键字后面时，在 **meta** 关键字后指定的 meta 属性被分配给资源而不是克隆。使用此更新的解析算法，在 meta 关键字后，当 meta关键字跟在 **clone** 关键字后面时，在 **meta** 关键字后指定的 meta 属性被分配给克隆。要保持与依赖旧格式的现有脚本的兼容性，您必须指定 **--future** 命令选项，以在使用 **pcs resource create** 命令创建克隆资源时启用这个新参数处理。

以下命令现在使用 meta 属性 **mv=v1** 创建资源，使用 meta 属性 **mv=v2** 创建克隆：

pcs resource create dummy1 ocf:pacemaker:Dummy meta m1=v1 clone meta m2=v2 --future

Bugzilla:2168155

显示重新创建配置的资源约束的 pcs 命令

现在，您可以使用带有新的 **--output-format=cmd** 选项的 **pcs constraint** 命令显示用于在不同系统上重新创建配置的资源约束的 **pcs constraint** 命令。与之前的版本一样，默认的输出格式是纯文本，您可以使用 **--output-format=text** 选项指定。纯文本格式已稍微更改，以使其与其它 **pcs** 命令的输出格式保持一致。

Bugzilla:2163953

将 Pacemaker 软件包 rebase 到版本：2.1.6

Pacemaker 软件包已升级到上游版本 2.1.6，与之前的版本相比，它提供了几个改进和 bug 修复。

添加了以下功能：

- 在以前的版本中，当 Pacemaker 远程连接丢失时，Pacemaker 总是清除其临时节点属性。如果连接快速恢复，且此时远程守护进程没有重启，则不需要此项。Pacemaker 远程节点现在在简短、可恢复的连接中断后保留临时节点属性。
- **alert_snmp.sh.sample** 警报代理是 Pacemaker 提供的示例警报代理，现在支持 SNMPv3 协议和 SNMPv2。有了此更新，您可以复制 **alert_snmp.sh.sample** 代理，而无需修改，以使用带有 Pacemaker 警报的 SNMPv3。
- Pacemaker 警报和警报接收者现在支持 **enabled** meta 选项。将警报的此选项设置为 **false** 可禁用警报。将警报的此选项设置为 **true**，将特定接收者的此选项设置为 **false** 会禁用该接收者的警报。此选项的默认值为 **true**。出于任何原因，如计划维护，您可以使用此选项来临时禁用警报。

以下 bug 已修复：

- Pacemaker Designated Controller 选举不再最终确定，直到所有待处理的操作都完成且没有丢失任何操作。
- **fence_scsi** 代理现在可以在 **devices** 属性未设置时自动检测共享的 **lvmlockd** 设备。
- 资源粘性现在可以与主机代管分数进行适当比较。
- **crm_resource** 命令现在允许只使用一个活跃副本来清理或移动捆绑包。
- 在以前的版本中，可升级的克隆实例按数字顺序分配，第一个是提升的实例。因此，如果需要启动提升的克隆实例，在某些情况下，未提升的实例会意外重启，因为实例号变了。有了此修复，当为节点分配实例号时，角色会被考虑，因此不会发生不必要的重启。

Bugzilla:2189301

对 pcs property 命令的改进

pcs property 命令现在支持以下改进：

- **pcs property config --output-format=** 选项
  - 指定 **--output-format=cmd** 来显示从当前集群属性配置创建的 **pcs property set** 命令。您可以使用这个命令在不同的系统上重新创建配置的集群属性。
  - 指定 **--output-format=json** 以 JSON 格式显示配置的集群属性。
  - 指定 **output-format=text** 以纯文本格式显示配置的集群属性，这是此选项的默认值。
- **pcs property defaults** 命令，它替换了弃用的 **pcs property --defaults** 选项
- **pcs property describe** 命令，它描述了集群属性的含义

Bugzilla:2163914

4.12. 动态编程语言、网页和数据库服务器

Python 中的一个控制电子邮件地址解析的新环境变量

为缓解 CVE-2023-27043，一个向后兼容的更改，以确保在 Python 3 中引入了更严格的电子邮件地址的解析。

RHSA-2024:2024 中的更新引入了一个新的 **PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING** 环境变量。当您将此变量设置为 **true** 时，以前的、不太严格的解析行为是整个系统的默认设置：

```
export PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING=true
```

但是，对受影响函数的单个调用可能仍然启用更严格的行为。

您可以通过使用以下内容创建 **/etc/python/email.cfg** 配置文件来取得相同的结果：

```
[email_addr_parsing]
PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING = true
```

如需更多信息，请参阅知识库文章 缓解 Python 中引入更严格的电子邮件地址解析的 CVE-2023-27043。

Jira:RHELDOCS-17369[1]

完全支持新的 nodejs:20 模块流

以前作为技术预览提供的新模块流 **nodejs:20** 被 RHEA-2023:7252 公告的发布完全支持。**nodejs:20** 模块流现在提供 **Node.js 20.9**，它是一个长期支持(LTS)版本。

从 RHEL 9.1 开始，与 **Node.js 18** 相比，RHEL 9.3 中包含的 **Node.js 20** 提供了新功能、bug 修复、安全修复和性能改进。

主要变更包括：

- **V8** JavaScript 引擎已升级至版本 11.3。
- **npm** 软件包管理器已升级至版本 9.8.0。
- **Node.js** 引入了一个新的实验性权限模型。
- **Node.js** 引入了一个新的实验性单可执行文件应用程序(SEA)功能。
- **Node.js** 提供了对实验性 ECMAScript 模块(ESM)加载程序的改进。
- 在 **Node.js 18** 中作为实验性 **node:test** 模块引进的原生测试运行程序现在被视为是稳定的。
- **Node.js** 提供了各种性能改进。

要安装 **nodejs:20** 模块流，请使用：

```
# dnf module install nodejs:20
```

如果要从 **nodejs:18** 流升级，请参阅 切换到更新的流。

有关 **nodejs** 应用程序流支持长度的详情，请查看 Red Hat Enterprise Linux 应用程序流生命周期 。

Bugzilla:2186717

Python tarfile 提取函数的新的 filter参数

要缓解 CVE-2007-4559，Python 向 **tarfile** 提取函数中添加了一个 **filter** 参数。参数允许关闭 **tar** 功能，以提高安全性（包括阻止 CVE-2007-4559 目录遍历攻击）。如果没有指定过滤器，则默认在 RHEL 中使用 **'data'** 过滤器，该过滤器是最安全但最受限的。另外，当应用程序会受到影响时，Python 会发出一

个警告。

如需更多信息，包括隐藏警告的说明，请参阅知识库文章 Python tarfile 库中目录遍历攻击的缓解(CVE-2007-4559) 。

Jira:RHELDOCS-16405[1]

HTTP::Tiny Perl 模块现在默认验证 TLS 证书

HTTP::Tiny Perl 模块中 **verify_SSL** 选项的默认值已从 **0** 改为 **1**，以在使用 HTTPS 时验证 TLS 证书。这个更改修复了用于 **HTTP::Tiny** 的 CVE-2023-31486 和用于 CPAN Perl 模块的 CVE-2023-31484 。

为了支持 TLS 验证，这个更新在 **perl-HTTP-Tiny** 软件包中添加了以下依赖项：

- **perl-IO-Socket-SSL**
- **perl-Mozilla-CA**
- **perl-Net-SSLeay**

Bugzilla:2228412[1]

httpd rebase 到版本 2.4.57

Apache HTTP 服务器已更新至版本 2.4.57，与 2.4.53 版本相比，它提供了自 RHEL 9.1 以来的 bug 修复、功能增强和安全修复。

主要改进包括：

- **httpd** 提供的 **rotatelogs** 工具引入了一个新的 **-T** 选项，来截断除所有轮转日志文件，除了初始日志文件。
- **mod_ldap** 模块的 **LDAPConnectionPoolTTL** 指令现在接受负值，以便能够重新使用任何时候的连接。在以前的版本中，负值作为错误处理。
- **mod_proxy_hcheck** 模块的 worker 现在可以根据 worker 超时设置正确超时。
- **mod_proxy_hcheck** 模块的 **hcmethod** 参数现在为 HTTP/1.1 请求提供新的 **GET11**、**HEAD11** 和 **OPTIONS11** 方法。

Bugzilla:2184403

httpd中新 mod_authnz_fcgi 模块

Apache HTTP 服务器现在包含 **mod_authnz_fcgi** 模块，它使 FastCGI 授权应用程序可以验证用户并授权对资源的访问。

mod_authnz_fcgi 模块不会默认加载。要载入此模块，请取消 **/etc/httpd/conf.modules.d/00-optional.conf** 文件中以下行的注释：

```
LoadModule authnz_fcgi_module modules/mod_authnz_fcgi.so
```

Bugzilla:2173295[1]

nginx:1.22中新的 ssl_pass_phrase_dialog 指令

有了对 **nginx:1.22** 模块流的此更新，您可以使用新的 **ssl_pass_phrase_dialog** 指令配置一个外部程序，对于每个加密的私钥，该程序在 **nginx** 启动时被调用。

要使用新指令，请在 **/etc/nginx/nginx.conf** 文件中添加以下行之一：

- 要为每个加密的私钥文件调用外部程序，请输入：

  ```
  ssl_pass_phrase_dialog exec:<path_to_program>;
  ```

  nginx 使用以下两个参数调用该程序：

  - 服务器名称在 **server_name** 设置中指定。
  - 以下一种算法之一：**RSA**、**DSA**、**EC**、**DH** 或 **UNK**（如果无法识别加密算法）。

- 如果要为每个加密的私钥文件手动输入密码短语，请输入：

  ```
  ssl_pass_phrase_dialog builtin;
  ```

  如果没有配置 **ssl_pass_phrase_dialog**，这是默认行为。

  请注意，如果您使用这个方法，则 **nginx** 服务无法启动，但至少有一个受密码短语保护的私钥。在这种情况下，请使用其它方法。

- 如果您希望 **systemd** 在使用 **systemctl** 工具启动 **nginx** 服务时对每个加密的私钥提示输入密码短语，请输入：

  ```
  ssl_pass_phrase_dialog exec:/usr/libexec/nginx-ssl-pass-dialog;
  ```

请注意，**nginx** 中的 **ssl_pass_phrase_dialog** 指令与 Apache HTTP 服务器中的 **SSLPassPhraseDialog** 指令类似。

Bugzilla:2170808

一个新的 rhel9/squid 容器镜像

rhel9/squid 容器镜像现在在 Red Hat Container Registry 中提供。 **Squid** 是 Web 客户端的高性能代理缓存服务器，支持 FTP、gopher 和 HTTP 数据对象。与传统的缓存软件不同，**Squid** 在单一的、非阻塞的、I/O 驱动的进程中处理所有请求。**Squid** 在 RAM 中保留缓存的元数据，特别是热对象，缓存 DNS 查找，支持非阻塞 DNS 查找，并实现失败请求的负缓存。

要拉取新容器镜像，请运行：

```
# podman pull registry.redhat.io/rhel9/squid
```

Bugzilla:2178953

新模块流：redis:7

Redis 7 一个高级的键-值存储，现在作为新的模块流 **redis:7** 提供。

相对于 **Redis 6** 的主要变化包括：

- Redis Functions API 中的服务器端脚本
- 支持细粒度访问控制列表(ACL)

- 集群的共享发布/订阅(pub/sub)支持
- 各种新命令和命令参数

Redis 7 引入了几个向后不兼容更改，例如：

- **Redis 7** 现在将仅附加文件(AOF)作为文件夹中的多个文件存储
- **Redis 7** 为与早期版本不兼容的 Redis 数据库(RDB)文件使用一个新的版本格式

有关功能和不兼容更改的完整列表，请参阅 上游发行注记。

要安装 **redis:7** 模块流，请使用：

```
# dnf module install redis:7
```

有关 **redis** 应用程序流支持长度的详情，请查看 Red Hat Enterprise Linux 应用程序流生命周期 。

Bugzilla:2129826

4.13. 编译器和开发工具

新的 glibc 选项，以影响 IBM Z 上优化的日常使用情况

在 IBM Z 构架上，**glibc** 库根据硬件功能（如 **hwcaps** 和 **stfle** 位）选择功能实现。有了此更新，您可以通过设置 **glibc.cpu.hwcaps** 可调项来指导库所做的选择。

Bugzilla:2169978[1]

改进了 glibc 中基于 Intel® Xeon® v5 硬件的字符串和内存例程性能

在以前的版本中，**glibc** 用于字符串和内存例程的默认缓存量导致在基于 Intel® Xeon® v5 的系统上性能低于预期。有了此更新，要使用的缓存量已被调整，以提高性能。

Bugzilla:2213907

系统 GCC 编译器更新至版本 11.4.1

GNU Compiler Collection (GCC) 提供用于使用 C、C++ 和 Fortran 编程语言开发应用程序的工具。

系统 GCC 编译器已更新至版本 11.4.1，其中包括上游 GCC 中提供的大量 bug 修复和增强。

有关使用信息，请参阅 RHEL 9 中开发 C 和 C++ 应用程序 。

Bugzilla:2193180

GCC 现在支持保留寄存器参数

有了此更新，您可以将参数寄存器内容存储到堆栈，并产生合适的 Call Frame Information (CFI)，以允许 unwinder 找到它，而不会对性能产生负面影响。

Bugzilla:2168204[1]

64 位 Intel 构架上 GCC 中的一个新的 -mdaz-ftz 选项

64 位 Intel 架构上的 GNU Compiler Collection (GCC)的系统版本现在支持 **-mdaz-ftz** 选项，来在 MXCSR 控制和状态寄存器中启用 flush-to-zero (FTZ)和 denormals-are-zero (DAZ)标志。

Bugzilla:2208908

新的 GCC Toolset 13

GCC Toolset 13 是一个编译器工具集，其提供开发工具的最新版本。它以 AppStream 存储库中的 Software Collection 的形式作为 Application Stream 提供。

GCC 编译器已更新至版本 13.1.1，它提供上游 GCC 中提供的很多 bug 修复和增强。

GCC Toolset 13 提供了以下工具和版本：

工具	版本
GCC	13.1.1
GDB	12.1
binutils	2.40
dwz	0.14
annobin	12.20

要安装 GCC Toolset 13，以 root 用户身份运行以下命令：

```
# dnf install gcc-toolset-13
```

要运行 GCC Toolset 13 中的工具：

```
$ scl enable gcc-toolset-13 tool
```

要运行一个 shell 会话，其中 GCC Toolset 13 中的工具版本会覆盖这些工具的系统版本：

```
$ scl enable gcc-toolset-13 bash
```

如需更多信息，请参阅 GCC Toolset 13 和 使用 GCC Toolset。

Bugzilla:2171919[1], Bugzilla:2171930

GCC Toolset 13：GCC rebase 到版本 13.1.1

在 GCC Toolset 13 中，GNU Compiler Collection (GCC)已更新至版本 13.1.1。主要变更包括：

常规改进

- OpenMP:
  - OpenMP 5.0： Fortran 现在支持一些非矩形循环嵌套。GCC 11 中添加了对 C/C++ 的此类支持。
  - 添加了许多 OpenMP 5.1 功能。
  - 添加了对 OpenMP 5.2 功能的初始支持。

- 现在提供了一个新的调试信息压缩选项值 **-gz=zstd**。
- **-Ofast**、**-ffast-math** 和 **-funsafe-math-optimizations** 选项不再添加启动代码，以在使用 **-shared** 选项生成一个共享对象时更改浮点环境。
- GCC 现在可以使用 Static Analysis Results Interchange Format (SARIF)发出其诊断，此格式是一种适合捕获静态分析工具（如 GCC 的 **-fanalyzer**）结果的基于 JSON 的格式。您还可以使用 SARIF 来捕获机器可读格式的其他 GCC 警告和错误。
- 实现了链接时优化改进。

新语言和特定语言的改进

C 系列：

- 新的 **-Wxor-used-as-pow** 选项会警告用户使用专用或(**^**)运算符时可能指的是求幂。
- 为记录是文件描述符的 **int** 参数添加了三个新的函数属性：
  - ***attribute*((fd_arg(N)))**
  - ***attribute*((fd_arg_read(N)))**
  - ***attribute*((fd_arg_write(N)))**

  -fanalyzer 也使用这些属性来检测文件描述符的滥用。
- 已为 C++23 可移植假设添加了一个新的语句属性***attribute*((assume(EXPR)));**。属性在 C 或更早的 C++ 中也被支持。
- GCC 现在可以控制何时将一个结构的尾部数组视为一个灵活的数组成员，以便访问此类数组的元素。默认情况下，聚合中的所有尾部数组都被视为灵活的数组成员。使用新的命令行选项 **-fstrict-flex-arrays** 控制哪些数组成员被视为灵活的数组。

C:

- 实现了几个 C23 功能：
  - 引入了 **nullptr** 常量。
  - 增强了枚举来指定底层类型。
  - 对可变参数列表的要求已经放宽。
  - 引入了 **auto** 功能，以启用对象定义的类型推断。
  - 为对象定义引入了 **constexpr** 指定符。
  - 为复合字面引入了存储类指定符。
  - 引入了**typeof** 对象（以前作为扩展支持）和 **typeof_unqual** 对象。
  - 添加了新的关键字：**alignas**，**alignof** ,**bool**,**false**,**static_assert**,**thread_local** 和 **true**。
  - 添加了 **[[noreturn]]** 属性，以指定函数不向其调用者返回执行。
  - 添加了对空初始化器大括号的支持。

  - 添加了对 ***STDC_VERSION keyring_H*** 标头版本宏的支持。
  - 删除了 **ATOMIC_VAR_INIT** 宏。
  - 为 **<stddef.h>** 标头添加了 **unreachable** 宏。
  - 删除了三角图。
  - 删除了非原型函数。
  - 通过 **%wN**的**-Wformat** 选项和 **%wfN** 格式长度修饰符添加了 **printf** 和 **scanf** 格式检查。
  - 添加了对 Unicode Standard Annex (UAX) 31 的标识符语法的支持。
  - C23 中采用的现有功能已被调整以遵循 C23 要求，且没有使用 **-std=c2x -Wpedantic** 选项进行诊断。

- 新的 **-Wenum-int-mismatch** 选项会警告枚举类型和整数类型之间的不匹配。

C++:

- 通过 **-fexcess-precision** 选项实现了超精度支持。它默认在严格的标准模式下启用，如 **-std=c++17**，其中其默认为 **-fexcess-precision=standard**。在 GNU 标准模式中，如 **-std=gnu++20**，它默认为 **-fexcess-precision=fast**，其恢复之前的行为。
  -fexcess-precision 选项会影响以下构架：
  - 在某些情况下，在使用 x87 数学的 Intel 32 和 64 位的 Motorola 68000 上，其中 **float** 和 **double** 表达式以 **long double** 精度评估。
  - 64 位 IBM Z 系统，其中**float**表达式以 **double** 精度评估。
  - 支持 **std::float16_t** 或 **std::bfloat16_t** 类型的几个架构，其中这些类型以**float**精度评估。
- 改进了对 C++23 的实验性支持，包括：
  - 添加了对复合语句末尾的标签的支持。
  - 添加了一个类型 trait 来检测绑定到临时对象的引用。
  - 重新引入对易失性复合操作的支持。
  - 添加了对 **#warning** 指令的支持。
  - 添加了对分隔的转义序列的支持。
  - 添加了对命名的通用字符转义的支持。
  - 为 **char8_t** 类型添加了兼容性和可移植性修复。
  - 添加了静态 **operator（）** 函数对象。
  - 简化的隐式移动。
  - 在表达式中重写等式现在不再是一个突破性的变化。
  - 删除了不可编码的宽字符量和宽多字符量。
  - 放宽了一些 **constexpr** 函数限制。

- 扩展的浮点类型和标准名称。
- 实现了可移植假设。
- 添加了对 UTF-8 作为可移植源文件编码标准的支持。
- 添加了对静态 **operator[]** 子脚本的支持。

- 新警告：
  - 当值通过 **std::move** 移动到其自身时，**-Wself-move** 会发出警告。
  - 当参考绑定到其生命周期已结束的临时对象时，**-Wdangling-reference** 会发出警告。
  - **-Wpessimizing-move** 和 **-Wredundant-move** 警告已扩展到在更多上下文中进行警告。

- 新的 **-nostdlib++** 选项启用了与 **g++** 的链接，而没有 C++ 标准库中的隐式链接。

libstdc++ 运行时库的更改

- 改进了对 C++20 的实验性支持，包括：
  - 添加了 **<format>** 标头和 **std::format** 函数。
  - 在**<chrono>**表头中添加了对 **std::chrono::utc_clock** 时钟、其他时钟、时区和 **std::format** 函数的支持。

- 改进了对 C++23 的实验性支持，包括：
  - 添加到 **<ranges>** 标头：**view::zip**,**views::zip_transform**,**views::adjacent**,**views::adjacent_transform**,**views::pairwise**,**views::slide**,**views::chunk**,**views::chunk_by**,**views::repeat**,**views::chunk_by**,**views::cartesian_product**,**views::as_rvalue**,**views::enumerate**,**views::as_const**.
  - 添加到 **<algorithm>** 标头：**ranges::contains**,**ranges::contains_subrange**,**ranges::iota**,**ranges::find_last**,**ranges::find_last_if**,**ranges::find_last_if_not**,**ranges::fold_left**,**ranges::fold_left_first**,**ranges::fold_right**,**ranges::fold_right_last**,**ranges::fold_left_with_iter**,**ranges::fold_left_first_with_iter**.
  - 对 **std::expected** 类模板的一元操作的支持。
  - 向 **std::bitset**、**std::to_chars** 和 **std::from_chars** 函数中添加了 **constexpr** 修饰符。
  - 添加了对扩展的浮点类型的库支持。

- 添加了对 Library Fundamentals Technical Specification (TS)版本 3 中**<experimental/scope>**标头的支持。
- 添加了对 Concurrency TS 版本 2 中**<experimental/synchronized_value>**标头的支持 。
- 添加了对 freestanding 模式下许多以前不可用功能的支持。例如：
  - **std::tuple** 类模板现在可用于独立编译。
  - **libstdc++** 库向独立子集中添加了组件，如 **std::array** 和 **std::string_view**。

- **libstdc++** 库现在遵循 **-ffreestanding** 编译器选项，因此它不再需要构建一个 **libstdc++** 库的独立安装。使用 **-ffreestanding** 编译会将可用的功能限制到独立的子集，即使 **libstdc++** 库是作为一个完整的托管实现构建。

新目标和特定于目标的改进

64 位 ARM 架构：

- 添加了对 **-march=** 选项的 **armv9.1-a**、**armv9.2-a** 和 **armv9.3-a** 参数的支持。

32 位和 64 位 AMD 以及 Intel 架构：

- 对于 C 和 C++，在启用了流 SIMD 扩展 2 及更高版本的系统上支持 **__bf16** 类型。
- 真正的 **__bf16** 类型现在用于 **AVX512BF16** 指令的内在函数。在以前的版本中，使用 **__bfloat16**，一种 short 的 typedef。在将 GCC 12 升级到 GCC 13 时，调整您的与 **AVX512BF16** 相关的源代码。
- 添加了新的指令集架构(ISA)扩展，以支持以下 Intel 指令：
  - **AVX-IFMA**，其指令内在函数通过 **-mavxifma** 编译器开关提供。
  - **AVX-VNNI-INT8**，其指令内在函数通过 **-mavxvnniint8** 编译器开关提供。
  - **AVX-NE-CONVERT**，其指令内在函数通过 **-mavxneconvert** 编译器开关提供。
  - **CMPccXADD**，其指令内在函数通过 **-mcmpccxadd** 编译器开关提供。
  - **AMX-FP16**，其指令内在函数通过 **-mamx-fp16** 编译器开关提供。
  - **PREFETCHI**，其指令内在函数通过 **-mprefetchi** 编译器开关提供。
  - **RAO-INT**，其指令内在函数通过 **-mraoint** 编译器开关提供。
  - **AMX-COMPLEX**，其指令内在函数通过 **-mamx-complex** 编译器开关提供。
- GCC 现在通过 **-march=znver4** 编译器开关支持基于 **znver4** 核的 AMD CPU。开关使 GCC 在自动向量化时考虑使用 512 位向量。

对静态分析器的改进

- 静态分析器已得到 20 个新警告：
  - **-Wanalyzer-allocation-size**
  - **-Wanalyzer-deref-before-check**
  - **-Wanalyzer-exposure-through-uninit-copy**
  - **-Wanalyzer-imprecise-fp-arithmetic**
  - **-Wanalyzer-infinite-recursion**
  - **-Wanalyzer-jump-through-null**
  - **-Wanalyzer-out-of-bounds**
  - **-Wanalyzer-putenv-of-auto-var**

    - **-Wanalyzer-tainted-assertion**
    - 与文件描述符滥用相关的 7 个新警告：
        - **-Wanalyzer-fd-access-mode-mismatch**
        - **-Wanalyzer-fd-double-close**
        - **-Wanalyzer-fd-leak**
        - **-Wanalyzer-fd-phase-mismatch** （例如，在对套接字调用 **listen** 前调用 **accept**）
        - **-Wanalyzer-fd-type-mismatch** （例如，对数据报套接字使用流套接字操作）
        - **-Wanalyzer-fd-use-after-close**
        - **-Wanalyzer-fd-use-without-check**
            - 另外，还实现了 **open**、**close**、**creat**、**dup** 、**dup2**、**dup3**、**pipe** 、**pipe2**、**read**、和 **write** 函数的行为的特殊处理。
    - 滥用**<stdarg.h>** 标头的 4 个新警告：
        - **-Wanalyzer-va-list-leak** 警告在 **va_start** 或 **va_copy** 宏后缺少 **va_end** 宏。
        - **-Wanalyzer-va-list-use-after-va-end** 警告对已对其调用了 **va_end** 宏的 **va_list** 对象类型使用了 **va_arg** 或 **va_copy** 宏。
        - **-Wanalyzer-va-arg-type-mismatch** 类型检查**va_arg**宏在针对实际传递给 variadic 调用的参数类型的过程间执行路径中的使用情况。
        - **-Wanalyzer-va-list-exhausted** 警告是否 **va_arg** 宏在进程间的执行路径中对**va_list** 对象类型使用了太多次。
- 很多其他改进。

后向不兼容的更改

对于 C++，全局 iostream 对象的结构，如 **std::cout**，**std::cin**，现在是在标准库内完成的，而不是在包含 **<iostream>** 标头的每个源文件中完成的。这个更改提高了 C++ 程序的启动性能，但这意味着如果在运行时没有使用正确的 **libstdc++.so** 版本，则使用 GCC 13.1 编译的代码将崩溃。请参阅有关在运行时使用正确的 **libstdc++.so**的 文档。将来的 GCC 版本会缓解这个问题，以便程序根本不能使用早期不兼容的 **libstdc++.so** 运行。

Bugzilla:2172093[1]

GCC Toolset 13: annobin rebase 到版本 12.20

GCC Toolset 13 提供了 **annobin** 软件包版本 12.20。主要改进包括：

- 添加了对将 **annobin** 备注移到单独的调试信息文件中的支持。这导致减小了可执行二进制文件的大小。
- 添加了对新的较小的备注格式的支持，减少了单独 debuginfo 文件的大小以及创建这些文件所需的时间。

Bugzilla:2171923[1]

GCC Toolset 13：GDB rebase 到版本 12.1

GCC Toolset 13 提供 GDB 版本 12.1。

重要的程序错误修复和增强包括：

- GDB 现在默认设置源代码和反汇编器的样式。如果样式干扰了 GDB 的自动化或脚本，您可以使用 **maint set gnu-source-highlight enabled off** 和 **maint set style disassembler enabled off** 命令禁用它。
- GDB 现在每当遇到内部错误时都会显示回溯追踪。如果这会影响脚本或自动化，您可以使用 **maint set backtrace-on-fatal-signal off** 命令禁用此功能。

C/C++ 改进：

- GDB 现在像对待功能过载一样对待涉及 C++ 模板的功能或类型。您可以省略参数列表，以在模板功能系列上设置断点，包括类型或由多个模板类型组成的功能。**Tab** 补全已得到类似的改进。

终端用户界面(TUI)：

- **tui layout**
  tui focus

  tui refresh

  tui window height
  是旧 **layout**， **focus**,**refresh** 和 **winheight** TUI 命令的新名称。旧名称仍作为这些新命令的别名存在。
- **tui window width**
  winwidth

  使用新的 **tui window width** 命令或 **winwidth** 别名，来在窗口在水平模式中布局时调整 TUI 窗口的宽度。
- **info win**
  这个命令现在在其输出中包含有关 TUI 窗口宽度的信息。

机器接口(MI)更改：

- MI 解释器的默认版本现在是 4 (-i=mi4)。
- 没有标志的 **-add-inferior** 命令现在继承当前下级的连接。这会恢复 GDB 版本 10 之前的行为。
- **-add-inferior** 命令现在接受一个 **--no-connection** 标志，这导致新的下级在没有连接的情况下启动。
- 断点输出中的 **script** 字段（其在 MI 3 及更早版本中的语法不正确）已在 MI 4 中成为一个列表。这会影响以下命令和事件：
  - **-break-insert**
  - **-break-info**
  - **=breakpoint-created**
  - **=breakpoint-modified**
    使用 **-fix-breakpoint-script-output** 命令启用使用早期 MI 版本的新行为。

新命令：

- **maint set internal-error backtrace [on|off]**
  maint show internal-error backtrace

  maint set internal-warning backtrace [on|off]

  maint show internal-warning backtrace

  GDB 现在在遇到内部错误或内部警告时打印其自身的回溯追踪。对于内部错误，这默认是启用的，对于内部警告，默认禁用。

- **exit**
  除了现有的 **quit** 命令外，您还可以使用新的 **exit** 命令退出 GDB。

- **maint set gnu-source-highlight enabled [on|off]**
  maint show gnu-source-highlight enabled
  启用或禁用 GNU Source Highlight 库来向源代码添加样式。禁用时，即使库可用，也不会使用它。当 GNU Source Highlight 库没有使用时，使用 Python Pygments 库。

- **set suppress-cli-notifications [on|off]**
  show suppress-cli-notifications

  控制是否禁止为 CLI 打印通知。当您更改所选上下文（如当前的下级、线程或帧）或正在调试的程序停止时（例如：由于遇到断点、完成源步进或中断）时，会发生 CLI 通知。

- **set style disassembler enabled [on|off]**
  show style disassembler enabled

  启用后，如果 GDB 是使用支持的 Python 编译的，且有 Python Pygments 软件包可用，则命令会向反编译器输出应用样式。

更改的命令：

- **set logging [on|off]**
  弃用，并被 **set logging enabled [on|off]** 命令替换。

- **print**
  使用基数修改格式（如 **/x** ）打印浮点值，已改为以所需基数显示底层值的字节。

- **clone-inferior**
  clone-inferior 命令现在确保 **TTY**、**CMD** 和 **ARGs** 设置从原始下级复制到新的下级。所有使用**set environment**或**unset environment** 命令对环境变量的修改也会被复制到新的下级中。

Python API：

- 新的 **gdb.add_history()** 函数接受一个 **gdb.Value** 对象，并将其代表的值添加到 GDB 的历史记录列表中。函数返回一个整数，这是历史记录列表中新项目的索引。

- 新的 **gdb.history_count（）** 函数返回 GDB 值历史记录中的值数。

- 新的 **gdb.events.gdb_exiting** 事件通过 **gdb.GdbExitingEvent** 对象调用，该对象具有包含 GDB 退出码值的只读属性 **exit_code**。这个事件会在 GDB 突出前触发，然后 GDB 开始清理其内部状态。

- 新的 **gdb.architecture_names（）** 函数返回一个包含所有可能的 **Architecture.name（）** 值的列表。每个条目都是一个字符串。

- 新的 **gdb.Architecture.integer_type（）** 函数返回一个给定大小和符号的整数类型。
- 新的 **gdb.TargetConnection** 对象类型表示一个连接（如 **info connections** 命令显示的那样）。子类 **gdb.RemoteTargetConnection** 表示 **remote** 和 **extended-remote** 连接。
- **gdb.Inferior** 类型现在有一个 **connection** 属性，它是 **gdb.TargetConnection** 对象的一个实例，连接被这个下级使用。如果下级没有连接，则这可以是 **None**。
- 当从 GDB 中删除连接时，新的 **gdb.events.connection_removed** 事件注册中心会发出一个 **gdb.ConnectionEvent** 事件。此事件有一个 **connection** 属性，一个用于要删除的连接的 **gdb.TargetConnection** 对象。
- 新的 **gdb.connections（）** 函数返回所有当前活跃连接的列表。
- 新的 **gdb.RemoteTargetConnection.send_packet (PACKET)** 方法等同于现有的 **maint packet** CLI 命令。您可以使用它来向远程目标发送指定的数据包。
- 新的 **gdb.host_charset（）** 函数返回作为字符串的当前主机字符集的名称。
- 新的 **gdb.set_parameter (*NAME*,*VALUE*)** 函数将 GDB 参数 ***NAME*** 设置为 ***VALUE***。
- 新的 **gdb.with_parameter (*NAME*,*VALUE*)** 函数返回一个上下文管理器，该管理器临时将 GDB 参数 ***NAME*** 设置为 ***VALUE***，然后在上下文退出时重置它。
- **gdb.Value.format_string** 方法现在接受一个 **styling** 参数，该参数是一个布尔值。当为 **true** 时，返回的字符串可以包含转义序列以应用样式。只有在 GDB 上打开样式时，才会显示样式（请参阅 **帮助设置样式**）。当未指定 **styling** 参数时，其为默认值**false**，则不会将样式应用到返回的字符串。
- 新的只读属性 **gdb.InferiorThread.details** 是一个包含额外的特定于目标的线程信息的字符串，或者如果没有这样的附加信息，则为 **None**。
- 新的只读属性 **gdb.Type.is_scalar** 对于 scalar 类型为 **True**，对所有其他类型为 **False**。
- 新的只读属性 **gdb.Type.is_signed** 仅在 **Type.is_scalar** 为 **True** 时可读，对于有符号类型将为**True**，对于所有其他类型将为**False**。尝试读取 non-scalar 类型的此属性将引发一个 **ValueError**。
- 现在，您可以添加在 Python 中添加 GDB 和实施的 MI 命令。

如需更多信息，请参阅上游发行注记：

GDB 中有哪些变化？

Bugzilla:2172096[1]

GCC Toolset 13：binutils rebase 到版本 2.40

GCC Toolset 13 提供 **binutils** 软件包版本 2.40。主要改进包括：

链接器：

- 链接器的新的 **-w** (**--no-warnings**)命令行选项压制任何警告或错误消息 产生。如果您需要创建一个已知无法正常工作的二进制文件，则这非常有用。
- 现在，ELF 链接器会产生一条警告信息，如果：
  - 堆栈为可执行的

    - 它创建一个内存驻留段，设置了所有三个 **Read**、**Write** 和 **eXecute** 权限集
    - 它创建一个具有 **eXecute** 权限集的线程本地数据段。
      您可以使用 **--no-warn-exec-stack** 或 **--no-warn-rwx-segments** 选项禁用这些警告。
- 链接器现在可以将任意 JSON 格式的元数据插入到它创建的二进制文件中。

其他工具：

- 新的 **objdump** 工具的 **--private** 选项，用于显示文件标头中的字段，以及 Portable Executable (PE)格式的部分标头。
- **objcopy** 和 **strip** 工具的新的 **--strip-section-headers** 命令行选项，来从 ELF 文件中删除 ELF 部分标头。
- **objdump** 工具的新的 **--show-all-symbols** 命令行选项，以显示反汇编时与给定地址匹配的所有符号，而不是仅显示与地址匹配的第一个符号的默认功能。
- **nm** 工具的新的 **-W** (**--weak**)选项，以使其忽略弱符号。
- **objdump** 工具现在支持对某些架构的反汇编输出的语法高亮显示。使用 **--disassembler-color=*MODE*** 命令行选项，*MODE* 是以下之一：
    - **off**
    - **color** - 所有终端模拟器都支持这个选项。
    - **extended-color** - 这个选项使用所有终端模拟器都不支持的 8 位颜色。

Bugzilla:2171926[1]

libabigail rebase 到版本 2.3

libabigail 软件包已更新至版本 2.3。主要改进包括：

- 现在支持 BTF debuginfo 格式。
- 改进了对 Ada 范围类型的支持。
- 现在支持压制规范中新的 **[allow_type]** 指令。
- 为 **[supress_type]** 压制规范添加了各种新属性。
- ABIXML 文件格式已更新至版本 2.2。
- 库的 SONAME 已更改，以反映自己的 ABI 变化。

libabigail 软件包在 CodeReady Linux Builder (CRB)存储库中提供。请注意，不支持 CodeReady Linux Builder 存储库中包含的软件包。

Bugzilla:2186931

debugedit 中的 find-debuginfo 脚本现在支持 -q (--quiet)标记

有了此更新，您可以使用 **debugedit** 工具中的 **find-debuginfo** 脚本的 **-q** (**--quiet**)标记来屏蔽脚本中的非错误输出。

Bugzilla:2177302

Valgrind rebase 到版本 3.21.0

Valgrind 已更新至版本 3.21.0。主要改进包括：

- **--vgdb-stop-at=*event1*,*event2*,...** 选项的新 **abexit** 值在程序异常退出时（如具有非零退出码）通知 **gdbserver** 工具。
- 新的 **--enable-debuginfod=[yes|no]** 选项指示 Valgrind 使用 **DEBUGINFOD_URLS** 环境变量中列出的 **debuginfod** 服务器来获取在 Valgrind 下运行的程序缺少的 DWARF debuginfo 信息。此选项的默认值为 **yes**。

注意

默认不设置 **DEBUGINFOD_URLS** 环境变量。

- Valgrind 现在提供 GDB Python 命令。这些 GDB 前端命令在 GDB 命令行界面中提供更好的集成。这样做的好处是，例如 GDB 自动完成功能，以及特定于命令的帮助，搜索与正则表达式匹配的命令或命令帮助。对于相关的监控命令，GDB 会评估参数以简化监控命令的使用。
- 使用 **--multi** 选项调用时，**vgdb** 工具现在支持扩展的远程协议。此模式下支持 GDB **run** 命令，因此您可以从单个终端运行 GDB 和 Valgrind。
- 对于截获 **malloc（）** 调用的工具，您可以使用 **--realloc-zero-bytes-frees=[yes|no]** 选项更改 **realloc（）** 函数的行为。
- **memcheck** 工具现在对大小为零的 **realloc （）** 函数的使用情况进行检查。使用新的 **--show-realloc-size-zero=[yes|no]** 开关来禁用此功能。
- 您可以对 **helgrind** 工具使用新的 **--history-backtrace-size=*value*** 选项，来配置要在早期访问的堆栈跟踪中记录的条目的数量。
- **--cache-sim=[yes|no] cachegrind** 选项现在默认为 **no**，因此默认只收集指令缓存读事件。
- **cg_annotate**、**cg_diff** 和 **cg_merge cachegrind** 工具的源代码已被重写，因此工具具有更灵活的命令行选项处理。例如，它们现在支持 **--show-percs** 和 **--no-show-percs** 选项以及现有的 **--show-percs=yes** 和 **--show-percs=no** 选项。
- **cg_annotate cachegrind** 工具现在支持区分（使用 **--diff**、**--mod-filename** 和 **--mod-funcname** 选项）和合并（通过传递多个数据文件）。另外，**cg_annotate** 现在在文件和函数级别提供更多信息。
- **DHAT** 工具的新的用户请求允许您覆盖对内存块的访问数直方图的 1024 字节限制。

现在支持以下特定于架构的指令集：

- 64 位 ARM:
  - v8.2 scalar 和 vector Floating-point Absolute Difference (FABD), Floating-point Absolute Compare Greater than or Equal (FACGE), Floating-point Absolute Compare Greater Than (FACGT)和 Floating-point Add (FADD)指令。
  - v8.2 浮点(FP)比较和条件比较指令。
  - v8.2 浮点(FP)的零变体比较指令。
- 64-位 IBM Z:

    - 支持各种 **miscellaneous-instruction-extensions facility 3** 和 **vector-enhancements facility 2**。这使用 **-march=arch13** 或 **-march=z15** 选项编译的程序能够在 Valgrind 下执行。

- IBM Power：
    - ISA 3.1 支持现已完成。
    - ISA 3.0 现在支持交付一个随机数字(darn)指令。
    - ISA 3.0 现在支持 System Call Vectored(scv)指令。
    - ISA 3.0 现在支持复制、粘贴和 cpabort 指令。

Bugzilla:2124346

SystemTap rebase 到版本 4.9

systemtap 软件包已升级到版本 4.9。主要变更包括：

- 新的 Language-Server-Protocol (LSP)后端，用于在支持 LSP 的编辑器上更轻松地交互式起草 **systemtap** 脚本。
- 访问 Python/Jupyter 交互笔记本前端。
- 改进了 DWARF 5 位字段的处理。

Bugzilla:2186934

elfutils rebase 到版本 0.189

elfutils 软件包已更新至版本 0.189。主要改进和 bug 修复包括：

libelf

elf_compress 工具现在支持 **ELFCOMPRESS_ZSTD** ELF 压缩类型。

libdwfl

dwfl_module_return_value_location 函数现在为指向 **DW_TAG_unspecified_type** 类型标签的 DWARF Information Entries (DIEs) 返回 0（无返回类型）。

eu-elfcompress

-t 和 **--type=** 选项现在通过**zstd**参数支持 Zstandard (**zstd**)压缩格式。

Bugzilla:2182061

libpfm rebase 到版本 4.13

libpfm 软件包已更新至版本 4.13。有了此更新， **libpfm** 可以访问以下处理器微架构的性能监控硬件原生事件：

- AMD Zen 2
- AMD Zen 3
- AMD Zen 4
- ARM Neoverse N1
- ARM Neoverse N2

- ARM Neoverse V1
- ARM Neoverse V2
- IBM z16
- 第四代 Intel® Xeon® 可扩展处理器

Bugzilla:2185652, Bugzilla:2047720, Bugzilla:2111940, Bugzilla:2111924, Bugzilla:2111930, Bugzilla:2111933, Bugzilla:2111957, Bugzilla:2111946

papi 支持新的处理器微架构

有了此增强，您可以使用以下处理器微架构上存在的 **papi** 事件访问性能监控硬件：

- AMD Zen 2
- AMD Zen 3
- ARM Neoverse N1
- ARM Neoverse N2
- ARM Neoverse V1
- ARM Neoverse V2

Bugzilla:2111923[1], Bugzilla:2111947, Bugzilla:2111942

papi 现在支持 64 位 ARM 处理器的快速性能事件数读取操作

在以前的版本中，在 64 位 ARM 处理器上，所有性能事件计数器读取操作都需要使用资源密集型系统调用。已为 64 位 ARM 更新了 **papi**，以便让使用性能计数器监控其自身的进程使用更快的性能事件计数器的用户空间读取。将 **/proc/sys/kernel/perf_user_access** 参数设置为 1，来将 **papi** 读取 2个计数器的平均时钟周期数从 724 个周期减少到 29 个周期。

Bugzilla:2186927[1]

LLVM Toolset rebase 到版本 16.0.6

LLVM Toolset 已更新至版本 16.0.6。

主要改进包括：

- 对优化的改进
- 对新的 CPU 扩展的支持
- 改进了对新 C++ 版本的支持。

主要的向后不兼容的更改包括：

- clang 的默认 C++ 标准现在是 **gnu++17** 而不是 **gnu++14**。
- **-Wimplicit-function-declaration**,**-Wimplicit-int** 和 **-Wincompatible-function-pointer-types** 选项现在默认为 C 代码的错误。这可能会影响配置脚本的行为。

默认情况下，Clang 16 使用 GCC Toolset 13 提供的 **libstdc++** 库版本 13 和 **binutils 2.40**。

如需更多信息，请参阅 LLVM 发行注记 和 Clang 发行注记。

Bugzilla:2178796

Rust Toolset rebase 到版本 1.71.1

Rust Toolset 已更新至版本 1.71.1。主要变更包括：

- 多个生成者(mpsc)，一个消费者(mpsc)渠道的新实现，以提高性能
- 新的 Cargo **sparse** 索引协议，以更有效地使用 **crates.io** 注册中心
- 用于一次性值初始化的新的 **OnceCell** 和 **OnceLock** 类型
- 新的 **C-unwind** ABI 字符串，以能够在跨 Foreign Function Interface (FFI)边界使用强制展开

如需了解更多详细信息，请参阅上游发布公告系列：

- 宣布 Rust 1.67.0
- 宣布 Rust 1.68.0
- 宣布 Rust 1.69.0
- 宣布 Rust 1.70.0
- 宣布 Rust 1.71.0

Bugzilla:2191743

Rust profiler_builtins 运行时组件现在可用

有了此更新，Rust **profile_builtins** 运行时组件现在可用。此运行时组件启用以下编译器选项：

-C instrument-coverage
启用覆盖率分析

-C profile-generate
启用配置文件引导的优化

Bugzilla:2227082[1]

Go Toolset rebase 到版本 1.20.10

Go Toolset 已更新到版本 1.20.10。

主要改进包括：

- 在 **unsafe** 软件包中添加了新功能，以处理片段和字符串，而不依赖于内部表示。
- 可比较类型现在可以满足可比较约束。
- 新的 **crypto/ecdh** 软件包。
- **go build** 和 **go test** 命令不再接受 **-i** 标志。
- **go generate** 和 **go test** 命令现在接受 **-skip *pattern*** 选项。
- **go build**、**go install** 和其它与构建相关的命令现在支持 **-pgo** 和 **-cover** 标志。

- **go** 命令现在在没有 C 工具链的系统上默认禁用 **cgo**。
- **go version -m** 命令现在支持读取更多的 Go 二进制文件类型。
- **go** 命令现在在没有 C 工具链的系统上默认禁用 **cgo**。
- 添加了对从应用程序和集成测试收集代码覆盖配置文件的支持，而不是仅从单元测试收集它们。

Bugzilla:2185259[1]

pcp rebase 到版本 6.0.5

pcp 软件包已更新至版本 6.0.5。主要变更包括：

收集器工具功能

- **pmdaproc** ：
  - 添加了对最近内核中每个 cgroup IRQ PSI 指标的支持
  - 添加了一个新的 **proc.smaps.pss_dirty** 指标
- **pmdasmart** ：添加了 NVME 磁盘信息和电源状态指标
- **pmdalinux**:
  - 添加了对最近内核中系统范围 IRQ PSI 指标的支持
  - 添加了 NUMA 外部内存碎片指标
  - 添加了新的网络(TCP、ICMP)指标
- **pmdaoverhead** ：一个测量进程组开销的新的 PMDA
- **pmdahacluster**: 已更新，来处理 Pacemaker 2.1.5 **crm_mon** 输出更改

监控工具功能

- **pmieconf**:
  - 添加了对 webhook 操作(事件驱动的 Ansible)的支持
  - 添加了一个新的检查文件描述符限制的 **pmie** 规则
- **pcp2json**: 带有一个选项的扩展的 **pcp2json**，以发送 HTTP POSTs
- **pcp-atop**: 添加了 **cgroup**、NUMA 内存和 NUMA CPU 支持
- **pcp-htop**: 添加了对新打开的文件描述符 Meter 的支持
- **pcp-ps**: 添加了显示多个归档示例的能力

Bugzilla:2175602

PCP 的 pmie 工具现在支持生成 webhook 事件

Performance Co-Pilot (PCP)中的性能指标推理引擎(**pmie**)工具现在支持生成 webhook 事件。有了此更新，配置的 **pmie** 规则生成可由 Event-Driven Ansible (EDA)消费的格式的事件。因此，EDA 可以响应 PCP 规则。

要启用此功能，请配置所有本地 **pmie** 规则，以便在给定的端点(URL)发送到 webhook：

```
# pmieconf modify global webhook_endpoint https://localhost:443/<endpoint>
# pmieconf modify global webhook_action yes
```

Bugzilla:2185803

grafana rebase 到版本 9.2.10

grafana 软件包已更新至版本 9.2.10。主要变更包括：

- 现在在整个 Grafana 中使用 heatmap 面板。
- Geomaps 现在可测量距离和区域。
- Alertmanager 现在基于 **Prometheus Alertmanager** 版本 0.24。
- Grafana 警告规则现在在执行错误或超时时默认返回一个 **Error** 状态。
- 表达式现在可以用于公共仪表盘。
- 加入转换现在支持内联。
- 公共仪表盘现在允许共享 Grafana 仪表盘。
- 新的 Prometheus streaming 解析器现在作为一个可选功能提供。

如需更多信息，请参阅上游发行注记：

- Grafana v9.1 中的新功能
- Grafana v9.2 中的新功能

Bugzilla:2193018

Grafana 不再启用弱加密密码

有了此更新，对于加密安全通信，Grafana 不再启用被视为弱的密码。受影响的密码有：

- **AES128-GCM-SHA256**
- **AES128-SHA**
- **AECDHE-RSA-AES128-SHA**
- **AES256-GCM-SHA384**
- **AES256-SHA**
- **ECDHE-RSA-AES256-SHA**

Bugzilla:2190025[1]

.NET 8.0 可用

Red Hat Enterprise Linux 9.3 与 .NET 版本 8.0 一起分发。主要改进包括：

- 添加了对 C#12 和 F#8 语言版本的支持。

- 添加了对直接使用 .NET 软件开发套件构建容器镜像的支持。
- 许多对垃圾收集器(GC)、Just-In-Time (JIT)编译器和基本库的性能改进。

Jira:RHELPLAN-164399[1]

4.14. 身份管理

samba rebase 到版本 4.18.6

samba 软件包已升级到上游版本 4.18.6，与之前的版本相比，它提供了 bug 修复和增强。最显著的更改：

- 之前版本中的安全性改进影响了高元数据工作负载的服务器消息块(SMB)服务器的性能。此更新在这种情况下提高了性能。
- 新的 **wbinfo --change-secret-at=<domain_controller>** 命令强制执行指定域控制器上信任帐户密码的更改。
- 默认情况下，Samba 将访问控制列表(ACL)存储在文件的 **security.NTACL** 扩展属性中。现在，您可以使用 **/etc/samba/smb.conf** 文件中的 **acl_xattr:<security_acl_name>** 设置自定义属性名称。请注意，自定义扩展属性名称不是一个作为 **security.NTACL** 的受保护的位置。因此，具有本地访问服务器权限的用户可以修改自定义属性的内容并破坏 ACL。

请注意，从 Samba 4.11 开始，服务器消息块块版本 1 (SMB1)协议已被弃用，并将在以后的发行版本中删除。

在启动 Samba 前备份数据库文件。当 **smbd**、**nmbd** 或 **winbind** 服务启动时，Samba 会自动更新其 **tdb** 数据库文件。红帽不支持降级 **tdb** 数据库文件。

更新 Samba 后，使用 **testparm** 工具来验证 **/etc/samba/smb.conf** 文件。

Bugzilla:2190415

ipaclient 角色现在允许在 IdM 级别上配置用户 subID 范围

有了此更新，**ipaclient ansible-freeipa** 角色提供 **ipaclient_subid** 选项，您可以使用此选项在身份管理(IdM)级别上配置 subID 范围。没有明确设置为 **true** 的新选项，**ipaclient** 角色会保持默认行为，并在没有为 IdM 用户配置 subID 范围的情况下安装客户端。

在以前的版本中，角色配置了 **sssd authselect** 配置文件，它会自定义 **/etc/nsswitch.conf** 文件。subID 数据库没有使用 IdM，只依赖于 **/etc/subuid** 和 **/etc/subgid** 的本地文件。

Bugzilla:2175767

现在，可以在单个 Ansible 任务中管理多个 IdM 组和服务

通过 **ansible-freeipa** 中的这一增强，您可以使用单个 Ansible 任务添加、修改和删除多个身份管理(IdM)用户组和服务。为此，请使用 **ipagroup** 和 **ipaservice** 模块的 **groups** 和 **services** 选项。

使用 **ipagroup** 中的 **groups** 选项，您可以指定仅应用到特定组的多个组变量。这个组由 **name** 变量定义，这是 **groups** 选项的唯一强制变量。

同样，使用 **ipaservice** 中的 **services** 选项，您可以指定仅适用于特定服务的多个服务变量。此服务由 **name** 变量定义，这是 **services** 选项的唯一强制变量。

Jira:RHELDOCS-16474[1]

ansible-freeipa ipaserver 角色现在支持随机序列号

有了此更新，您可以将 **ipaserver_random_serial_numbers=true** 选项与 **ansible-freeipa ipaserver** 角色一起使用。这样，当使用 Ansible 安装身份管理(IdM)服务器时，您可以为 PKI 中的证书和请求生成完全随机的序列号。使用 RSNv3，您可以避免大型 IdM 安装中的范围管理，并防止重新安装 IdM 时常见的冲突。

重要

RSNv3 仅支持新的 IdM 安装。如果启用，则需要在所有 PKI 服务上使用 RSNv3。

Jira:RHELDOCS-16462[1]

ipa rebase 到版本 4.10.2

ipa 软件包已升级到版本 4.10.2 。主要变更包括：

- 在 IdM CLI 和 Web UI 中搜索和列出证书现在提供更好的性能。

如需更多信息，请参阅 上游 FreeIPA 发行注记。

Bugzilla:2196426

ipaserver_remove_on_server 和 ipaserver_ignore_topology_disconnect 选项现在在 ipaserver 角色中提供

如果使用 **ipaserver ansible-freeipa** 角色的 **remove_server_from_domain** 选项从身份管理(IdM)拓扑中删除一个副本导致一个断开的拓扑，则必须指定您要保留域的哪一部分。具体来说，您必须执行以下操作：

- 指定 **ipaserver_remove_on_server** 值，以识别您要保留拓扑的哪一部分。
- 将 **ipaserver_ignore_topology_disconnect** 设置为 True。

请注意，如果使用 **remove_server_from_domain** 选项从 IdM 中删除一个副本会保留一个连接的拓扑，则不需要这些选项的任何一个。

Bugzilla:2127903

IdM 现在支持 min_lifetime 参数

有了此增强，**min_lifetime** 参数已添加到 **/etc/gssproxy114.conf** 文件中。如果剩余生命周期低于这个值，则 **min_lifetime** 参数会触发续订服务票据。

默认情况下，其值为 15 秒。对于 NFS 等网络卷客户端，若要降低 KDC 暂时不可用时丢失访问权限的风险，请将此值设为 60 秒。

Bugzilla:2181465

您现在可以使用 ipacert Ansible 模块管理 IdM 证书

您现在可以使用 **ansible-freeipa ipacert** 模块为身份管理(IdM)用户、主机和服务请求或检索 SSL 证书。然后，用户、主机和服务可以使用这些证书向 IdM 进行身份验证。您还可以撤销证书，并恢复已搁置的证书。

Bugzilla:2127907

optional_pac_tkt_chksum 选项帮助保留不同 krb5 版本间的互操作性

现在，您可以使用 **optional_pac_tkt_chksum** 选项来保留运行 **krb5** 软件包不同版本的 RHEL Kerberos 分发中心(KDC)服务器之间的互操作性。特别是，您可以更改其有关 Privilege Attribute Certificate(PAC) 票据名验证的行为。如果将期望签名票据的 Kerberos 主体的 **optional_pac_tkt_chksum** 字符串属性设为 **true**，则 KDC 不会拒绝包含缺少 PAC 票据签名的票据的用户(S4U)请求的服务。签名票据的主体是票据授予服务(TGS)主体或跨领域 TGS 主体，具体取决于票据目标服务的领域。

由于 **krb5-1.20** 发行版，MIT Kerberos KDC 已要求在基于 Kerberos 票据的加密部分的 PACs 中存在票据签名，以便可以成功处理 S4U 请求。在以前的版本中，这在逐步升级场景中是一个问题，其中某些 KDC 使用 **krb5-1.19** 或更早版本，而其他 KDC 使用 **krb5-1.20** 或更新版本。对 S4U 请求使用较新版本 **krb5** 的 KDC 拒绝使用旧版本 **krb5** 的 KDC 提供的服务票据，如果服务为 S4U 请求使用了它们。

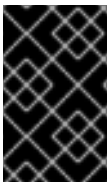
有关此功能如何在身份管理(IdM)中使用的更多信息，请参阅 此拉取请求。

Bugzilla:2178298

IdM 现在支持基于资源的受限委托

有了此更新，IdM 支持基于资源的受限委托(RBCD)。RBCD 允许资源级别上委派的精细控制，并且可由凭据委派给的服务的所有者来设置访问权限。

例如在 IdM 和活动目录(AD)之间的集成中 RBCD 可能很有用，因为在目标和代理服务都属于不同的林时 AD 强制使用 RBCD。



重要

目前，只有 IdM 域中的服务才能使用 RBCD 规则进行配置。如果目标服务是 AD 域的一部分，则只能在 AD 端授予权限。因为 AD 域控制器无法解析创建规则的 IdM 服务信息，因此目前还不支持此功能。

有关委派场景的更多信息，请参阅 FreeIPA 设计页。

Bugzilla:2165880

RHEL 9.3 提供 389-ds-base 2.3.4

RHEL 9.3 与 **389-ds-base** 软件包版本 2.3.4 一起分发。与 2.3.4 版本相比，重要的 bug 修复和增强包括：

- https://www.port389.org/docs/389ds/releases/release-2-2-8.html
- https://www.port389.org/docs/389ds/releases/release-2-2-9.html
- https://www.port389.org/docs/389ds/releases/release-2-3-0.html
- https://www.port389.org/docs/389ds/releases/release-2-3-1.html
- https://www.port389.org/docs/389ds/releases/release-2-3-2.html
- https://www.port389.org/docs/389ds/releases/release-2-3-3.html
- https://www.port389.org/docs/389ds/releases/release-2-3-4.html

Bugzilla:2188627

现在，如果 bind 操作失败，目录服务器现在可以关闭客户端连接

在以前的版本中，当 **bind** 操作失败时，一些忽略 **bind** 返回码的应用程序可能会通过进一步请求加载目录服务器。

使用 **cn=config** 条目下的新的 **nsslapd-close-on-failed-bind** 配置属性，服务器可在 **bind** 操作失败时关闭客户端连接。因此，可以减少服务器负载。

Bugzilla:1987471

Automembership 插件改进。默认情况下，它不再清理组

在以前的版本中，automember 重建任务会遍历所有 automember 规则，并删除所有成员资格，然后任务从头开始重新构建成员资格。因此，重建任务的成本很高，特别是在启用了其他 **be_txn** 插件时。

有了此更新，Automembership 插件有以下改进：

- 一次只允许一个重建任务。
- Automembership 插件现在默认不会清理以前的成员。使用新的 CLI 选项 **--cleanup**，来在从头开始重新构建前有意清理成员资格：

```
# dsconf slapd-instance_name plugins automember fixup -f objectclass=posixaccount -s sub
--cleanup "ou=people,dc=example,dc=com"
```

- 改进了日志记录以显示修复进度。

Bugzilla:2149025

新的 passwordAdminSkipInfoUpdate: on/off 配置选项现在可用

您可以在 **cn=config** 条目下添加一个新的 **passwordAdminSkipInfoUpdate: on/off** 设置，以对密码管理员执行的密码更新提供精细控制。当您启用此设置时，密码更新不会更新某些属性，例如 **passwordHistory**,**passwordExpirationTime**,**passwordRetryCount**,**pwdReset** 和 **passwordExpWarned**。

Bugzilla:2166332

新的 slapi_memberof（） 插件函数现在可用于目录服务器插件和客户端应用程序

新的 **slapi_memberof（）** 函数检索给定条目直接或间接所属组的可分辨名称(DN)。在以前的版本中，MemberOf、referential Integrity 和 ACL 插件实现了自己的机制来检索此类组。有了此更新，您可以使用引入了一个统一机制的 **slapi_memberof（）** 函数来返回组 DN。

Bugzilla:2189946

目录服务器现在使用受管的和过滤的角色的索引属性来替换虚拟属性 nsRole

在以前的版本中，LDAP 搜索过滤器中包含的虚拟属性 **nsRole** 非常耗时，因为该属性无法被索引。有了此更新，当您在过滤器中使用虚拟属性 **nsRole** 执行 **ldapsearch** 时，目录服务器使用以下方法替换 **nsRole** 属性：

- 对于受管角色，**nsRole** 属性被 **nsRoleDN** 属性替代。
- 对于过滤的角色，**nsRole** 属性被 **nsRoleFilter** 属性替代。

因此，使用 **nsRole** 属性搜索的响应时间会改进，因为搜索变为索引的。

请注意，此次更新不适用于嵌套角色。

Bugzilla:2189954

新的 nsslapd-numlisteners 配置选项现在可用

nsslapd-numlisteners 属性指定目录服务器用来监控已建立连接的监听程序线程的数量。您可以通过增加属性值来提高服务器遇到大量客户端连接时的响应时间。

Bugzilla:1975930

IdM 支持选项来控制用于为 PAC 签名的加密类型

默认情况下，Kerberos 密钥分发中心(KDC)为 Privilege Attribute 证书(PAC)生成 AES HMAC-SHA2 签名。但是，活动目录(AD)不支持此加密类型。因此，AD 跨领域约束的委托请求无法被正确处理。

有了这个增强，您现在可以通过在 TGS 主体，**krbtgt/[realm]@[realm]** 上设置 **pac_privsvr_entype** 属性，来将用于为 PAC 签名的加密类型控制为目标域所需的加密类型。在 IdM 中，当 AD 信任存在时，会自动配置此字符串属性。

> WARNING: This update is about standalone MIT realms. Do not change the Kerberos Distribution Center (KDC) configuration in RHEL Identity Management.

例如，对于 **MIT** 领域和 **AD** 领域，以确保跨域票据授予票据(TGT)使用 AD 兼容的加密类型，管理员必须配置跨领域 TGS 主体，如在 MIT 端如下所示。这会使使用 AES 256 HMAC-SHA1 加密类型和受限委托请求的跨领域 TGT 被正确处理。

```
kadmin.local <<EOF
setstr krbtgt/AD@IPA pac_privsvr_enctype aes256-cts-hmac-sha1-96
setstr krbtgt/IPA@AD pac_privsvr_enctype aes256-cts-hmac-sha1-96
EOF
```

Bugzilla:2060421

现在完全支持身份管理 API

在 RHEL 9.2 中，身份管理(IdM) API 作为技术预览提供，它从 RHEL 9.3 开始被完全支持。

用户可以使用现有工具和脚本来启用多个 API 命令版本，即使 IdM API 已被改进。这些改进不会以不兼容的方式更改命令的行为。这有以下优点：

- 管理员可以在服务器上而不是在管理客户端上使用之前或更高版本的 IdM。
- 开发人员可以使用 IdM 调用的特定版本，即使 IdM 版本在服务器上发生了变化。

无论使用哪一端，都可以与服务器进行通信，例如，未来引入新选项的较新版本。

注意

虽然 IdM API 提供了 JSON-RPC 接口，但不支持这种类型的访问。红帽建议使用 Python 访问 API。使用 Python 自动化重要的部分，如从服务器检索的元数据，其允许列出所有可用的命令。

Bugzilla:1513934

4.15. 图形基础结构

现在完全支持 Intel Arc A 系列图形

之前作为技术预览提供的 Intel Arc A 系列图形(Alchemist 或 DG2)功能现在完全支持。Intel Arc A 系列图形是一个启用了硬件加速的 GPU，主要用于 PC 游戏。

Bugzilla:2101598[1]

4.16. WEB 控制台

Podman 健康检查操作现在可用

您可以在创建新容器时选择以下 Podman 健康检查操作之一：

- No action（默认）：不执行任何操作。
- Restart : 重启容器。
- Stop ：停止容器。
- Force stop ：强制停止容器，它不等待容器退出。

Jira:RHELDOCS-16247[1]

Stratis 现在在 RHEL web 控制台中可用

有了此更新，Red Hat Enterprise Linux web 控制台提供了管理 Stratis 存储的功能。

要了解更多有关 Stratis 的信息，请参阅 使用 Web 控制台设置 Stratis 文件系统。

Jira:RHELPLAN-122345[1]

4.17. RED HAT ENTERPRISE LINUX 系统角色

用于管理 systemd 单元的新的 RHEL 系统角色

rhel-system-role 软件包现在包含 **systemd** RHEL 系统角色。您可以使用此角色在多个系统上部署单元文件并管理 **systemd** 单元。您可以通过提供 **systemd** 单元文件和模板，并通过指定这些单元的状态，如 started、stoped、masked 等，来自动执行 **systemd** 功能。

Bugzilla:2224384

ssh 角色中的新选项，以禁用配置备份

现在，您可以通过将新的 **ssh_backup** 选项设置为 **false** 来防止在覆盖旧配置文件前备份它们。在以前的版本中，备份配置文件会自动创建，这可能是必需的。**ssh_backup** 选项的默认值为 **true**，它会保留原始行为。

Bugzilla:2216753

keylime_server RHEL 系统角色

有了新的 **keylime_server** RHEL 系统角色，您可以使用 Ansible Playbook 在 RHEL 9 系统上配置验证器和注册器 Keylime 组件。Keylime 是一个使用可信平台模块(TPM)技术的远程机器认证工具。

Bugzilla:2224385

支持新的 ha_cluster 系统角色功能

ha_cluster 系统角色现在支持以下功能：

- 资源配置和资源操作的默认值，包括带有规则的多组默认值。
- SBD watchdog 内核模块的加载和阻止。这使得安装的硬件 watchdog 可供集群使用。
- 为集群主机和仲裁设备分配不同的密码。这可让您配置一个部署，其中同一仲裁主机加入到多个独立的集群，这些集群上 **hacluster** 用户的密码不同。

有关您配置实现这些功能的参数的详情，请参考 使用 ha_cluster RHEL 系统角色配置高可用性集群。

Bugzilla:2185065,Bugzilla:2185067,Bugzilla:2216481

storage 系统角色支持为 RAID LVM 卷配置条带大小

有了此更新，您现在可以在创建 RAID LVM 设备时指定自定义条带大小。为提高性能，请为 SAP HANA 使用自定义条带大小。建议的 RAID LVM 卷的条带大小为 64 KB。

Bugzilla:2181656

network RHEL 系统角色支持 auto-dns 选项，以控制自动 DNS 记录更新

此功能增强位定义的名称服务器和搜索域提供支持。现在，您只能使用 **dns** 和 **dns_search** 属性中指定的名称服务器和搜索域，同时禁用自动配置的名称服务器和搜索域，如 DHCP 的 **dns record**。有了这个增强，您可以通过更改 **auto-dns** 设置来禁用自动 dns 记录。

Bugzilla:2211194

network RHEL 系统角色支持 no-aaaa DNS 选项

现在，您可以使用 **no-aaaa** 选项在受管节点上配置 DNS 设置。在以前的版本中，没有选项来压制 stub 解析器产生的 AAAA 查询，包括由基于 NSS 的接口触发的 AAAA 查找（如 **getaddrinfo** ）；只有 DNS 查找会受到影响。有了这个增强，您可以压制由 stub 解析器生成的 AAAA 查询。

Bugzilla:2218592

ad_integration RHEL 系统角色现在可以重新加入 AD 域

有了此更新，您可以使用 **ad_integration** RHEL 系统角色重新加入活动目录(AD)域。为此，请将 **ad_integration_force_rejoin** 变量设置为 **true**。如果 **realm_list** 输出显示该主机已在一个 AD 域中，则它将在重新加入前离开现有域。

Bugzilla:2211723

certificate RHEL 系统角色现在在使用 certmonger 时允许更改证书文件模式

在以前的版本中，由具有 **certmonger** 提供者的 **certificate** RHEL 系统角色创建的证书使用默认的文件模式。但是，在某些用例中，您可能需要更严格的模式。有了此更新，您现在可以使用 **mode** 参数设置一个不同的证书和密钥文件模式。

Bugzilla:2218204

postgresql RHEL 系统角色现在可用

新的 **postgresql** RHEL 系统角色安装、配置、管理和启动 **PostgreSQL** 服务器。该角色还优化了数据库服务器设置，以提高性能。

该角色在 RHEL 8 和 RHEL 9 受管节点上支持当前发布和支持的 **PostgreSQL** 版本。

如需更多信息，请参阅 使用 postgresql RHEL 系统角色安装和配置 PostgreSQL。

Bugzilla:2151373

Podman RHEL 系统角色现在支持 Quadlets、健康检查和 secret

从 Podman 4.6 开始，您可以在 **podman** RHEL 系统角色中使用 **podman_quadlet_specs** 变量。您可以通过指定一个单元文件来定义 Quadlet，或通过名称、单元类型和规格在清单中定义 Quadlet。一个单元的类型可以是以下： **container**、**kube**、**network** 和 **volume**。请注意，Qadlets 仅适用于 RHEL 8 上的 root 容器。Quadlets 适用于 RHEL 9 上的无根容器。

健康检查只支持 Quadlet 容器类型。在 **[Container]** 部分中，指定 **HealthCmd** 字段来定义健康检查命令，指定 **HealthOnFailure** 字段来在容器不健康时定义操作。可能的选项为 **none**,**kill**,**restart**, and **stop**。

您可以使用 **podman_secrets** 变量来管理 secret。详情请查看 上游文档。

Jira:RHELPLAN-154441[1]

使用 restorecon -T 0 提高了 selinux 系统角色的性能

在所有适用的情况下，**selinux** 系统角色现在将 **-T 0** 选项和 **restorecon** 命令一起使用。这提高了在文件上恢复默认的 SELinux 安全上下文的任务的性能。

Bugzilla:2179460

rhc 系统角色现在支持设置代理服务器类型

在 **rhc_proxy** 参数下新引入的属性 **scheme**，使您可以使用 **rhc** 系统角色配置代理服务器类型。您可以设置两个值：**http**，默认值和 **https**。

Bugzilla:2211748

firewall RHEL 系统角色支持与 ipset 相关的变量

有了 **firewall** RHEL 系统角色的这个更新，您可以定义、修改和删除 **ipset**。您还可以从防火墙区中添加和删除 **ipset**。或者，您可以在定义防火墙富规则时使用这些 **ipset**。

您可以使用以下变量，使用 **firewall** RHEL 系统角色管理 **ipset** ：

- **ipset**
- **ipset_type**
- **ipset_entries**
- **short**
- **description**
- **state: present** 或 **state: absent**
- **permanent: true**

以下是此改进的一些显著优点：

- 您可以降低为许多 IP 地址定义规则的富规则的复杂度。
- 您可以根据需要从集合中添加或删除 IP 地址，而无需修改多个规则。

如需了解更多详细信息，请参阅 **/usr/share/doc/rhel-system-roles/firewall/** 目录中的资源。

Bugzilla:2229802

RHEL 系统角色现在对挂载点自定义有新的卷选项

有了此更新，您现在可以为挂载目录指定 **mount_user**、**mount_group** 和 **mount_permissions** 参数。

Bugzilla:2181657

firewall RHEL 系统角色有一个禁用冲突服务的选项，如果 firewalld 被屏蔽，它不再失败

在以前的版本中，当角色运行时 **firewalld** 被屏蔽或存在冲突服务时，**firewall** 系统角色失败。这个更新引入了两个显著改进：

linux-system-roles.firewall 角色总是尝试在角色运行时安装、取消屏蔽及启用 **firewalld** 服务。您现在可以在 playbook 中添加一个新的变量 **firewall_disable_conflicting_services**，来禁用已知的冲突服务，如 **iptables.service**、**nftables.service** 和 **ufw.service**。**firewall_disable_conflicting_services** 变量默认被设置为 **false**。要禁用冲突服务，请将变量设置为 **true**。

Bugzilla:2222761

重置 firewall RHEL 系统角色配置现在需要较少的停机时间

在以前的版本中，当使用 **previous: replaced** 变量重置 **firewall** 角色配置时，**firewalld** 服务会重启。重启会增加停机时间并延长打开连接的时间，其中**firewalld** 不会阻止来自活跃连接的流量。有了这个增强，**firewalld** 服务通过重新加载而不是重启来完成配置重置。重新加载会最大限度地减少停机时间，并减少绕过防火墙规则的机会。因此，使用 **previous: replaced** 变量重置 **firewall** 角色配置现在需要最少的停机时间。

Bugzilla:2223764

4.18. 虚拟化

sevctl 现在与 AMD EPYC Rome 和 Milan 完全兼容

有了此更新，**sevctl** 工具可以正确地识别最新的 AMD EPYC 核，包括 AMD EPYC Rome 和 AMD EPYC Milan 系列。因此，您可以使用 **sevctl** 配置在这些 CPU 上可用的 AMD Secure Encrypted Virtualization (SEV)的功能。

但请注意，高级 SEV 功能（如 SEV-ES 和 SEV-SNP ）仅在 RHEL 9 中作为技术预览提供，因此不被支持。

Bugzilla:2104857[1]

virtio-vga 和 virtio-gpu 设备现在支持 blob 资源

现在，**virtio-vga** 和 **virtio-gpu** 设备可以使用 **blob** 内存资源，这在某些情况下提高了其性能。要将 **blob** 资源附加到 **virtio** 图形设备，请在虚拟机的 XML 配置中相应的 **<video>** 部分中添加一个 **blob="on"** 选项。例如：

```
<video>
  <model type="virtio" heads="1" primary="yes" blob="on"/>
  <address type="pci" domain="0x0000" bus="0x00" slot="0x01" function="0x0"/>
</video>
```

但请注意，这个功能目前无法在 IBM Z 主机上工作。

Bugzilla:2032406

对第 4 代 Intel Xeon 可扩展处理器的虚拟化支持

有了这个更新，RHEL 9 上的虚拟化增加了对第 4 代 Intel Xeon 可扩展处理器的支持，其以前被称为 Sapphire Rapids。因此，在 RHEL 9 中托管的虚拟机现在可以使用 **SapphireRapids** CPU 型号，并使用处理器提供的新功能。

Bugzilla:1880531[1]

改进了 IBM Z 上安全执行的内存回收

当在 IBM Z 上使用带有 IBM Secure Execution 的虚拟机(VM)时，您现在可以为虚拟机设置增强的内存回收。如果虚拟机使用 32 GiB 或更多 RAM，此设置可以提高重启或停止虚拟机的性能。

要在虚拟机中设置增强的内存回收，请将 **<async-teardown enabled='yes'/>** 行添加到 XML 配置中的 **<features></features>** 部分。

Bugzilla:2168499[1]

RHEL web 控制台中的新虚拟化功能

在这个版本中，RHEL web 控制台在 Virtual Machines 页面中包含新功能。您现在可以：

- 根据云镜像对虚拟机选择 **Create and edit** 按钮，这允许您在安装虚拟机前编辑所有 VM 属性。
- 在虚拟机创建过程中创建一个 **raw** 存储卷。
- 设置一个虚拟套接字(vsock)，来启用通过套接字的主机和虚拟机之间的通信。
  请注意，虚拟套接字需要 vsock 感知软件，如 **socat**，已启用通信。

Jira:RHELDOCS-16487[1]

4.19. 云环境中的 RHEL

cloud-init 支持 NetworkManager keyfiles

有了此更新，**cloud-init** 工具可以使用 NetworkManager (NM) keyfile 来配置创建的云实例的网络。

请注意，默认情况下，**cloud-init** 仍然使用 **sysconfig** 方法进行网络设置。要将 **cloud-init** 配置为使用 NM keyfile，请编辑 **/etc/cloud/cloud.cfg**，并将 **network-manager** 设置为主网络呈现器：

```
# cat /etc/cloud/cloud.cfg

  network:
    renderers: ['network-manager', 'eni', 'netplan', 'sysconfig', 'networkd']
```

Bugzilla:2118235[1]

cloud-init 现在默认使用 ESXi 上的 VMware 数据源

当在使用 VMware ESXi hypervisor（如 VMware vSphere 云平台）的主机上创建 RHEL 虚拟机(VM)时。这提高了使用 **cloud-init** 创建 RHEL 的 ESXi 实例的性能和稳定性。但请注意，ESXi 仍与 Open Virtualization Format (OVF)数据源兼容，如果 VMware 不可用，您可以使用 OVF 数据源。

Bugzilla:2172341[1]

4.20. 支持性

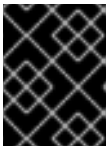
sos rebase 到版本 4.6

用于收集配置、诊断和故障排除数据的 **sos** 工具已 rebase 到版本 4.6。此更新提供了以下改进：

- **sos** 报告现在包含 **/boot/grub2/custom.cfg** 和 **/boot/grub2/user.cfg** 文件的内容，它们可能包含故障排除引导问题的重要信息。(BZ#2213951)
- OVN-Kubernetes 的 **sos** 插件为互连环境收集额外的日志。有了此更新，当 **ovnkube-node** 和 **ovnkube-controller** 容器合并成一个时，**sos** 也从 **ovnkube-controller** 容器收集日志。

另外，重要的 bug 修复包括：

- **sos** 现在可以正确地收集 OpenShift Container Platform 4 环境中的 **cgroup** 数据 (BZ#2186361)。
- 在收集启用了 **sudo** 插件的 **sos** 报告时，**sos** 现在正确地删除了 **bindpw** 选项。(BZ#2143272)
- **subscription_manager** 插件不再从 **/var/lib/rhsm/** 路径收集代理用户名和密码。(BZ#2177282)
- **virsh** 插件不再收集 virt-manager 日志中的 SPICE 远程-显示密码，这阻止 **sos** 在其报告中披露密码。(BZ#2184062)
- **sos** 现在屏蔽了之前在 **/var/lib/iscsi/nodes/<IQN>/<PortalIP>/default** 文件中显示的用户名和密码。

  重要

  生成的存档可能包含被视为敏感的数据。因此，在将其传递给任何第三方之前，您应该始终检查内容。

  (BZ#2187859)
- 即使超过了日志文件的大小，且插件超时，**sos** 也会完成尾部日志收集。(BZ#2203141)
- 当在 Pacemaker 集群节点上输入 **sos collect** 命令时，**sos** 会从同一集群节点收集 sos 报告。(BZ#2186460)
- 当从 OpenShift Container Platform 4 环境中的主机收集数据时，**sos** 现在使用 **sysroot** 路径，这保证只组装正确的数据。(BZ#2075720)
- **sos report --clean** 命令按照预期模糊处理所有 MAC 地址。(BZ#2207562)
- 禁用 **hpssm** 插件不再导致异常。(BZ#2216608)
- **sos clean** 命令遵循已清理文件的权限。(BZ#2218279)

有关 **sos** 的每个发行版本的详情，请参阅 上游发行注记。

Jira:RHELPLAN-156196[1]

4.21. 容器

Podman 支持拉取和推送使用 zstd 压缩的镜像

您可以拉取和推送使用 **zstd** 格式压缩的镜像。zstd 压缩效率比 gzip 更高，更快。它可以减少拉取和推送镜像所需的网络流量和存储量。

Jira:RHELPLAN-154314[1]

Podman 中的 Quadlet 现在可用

从 Podman v4.6 开始，您可以使用 Quadlet 从容器描述中自动生成 **systemd** 服务文件。Quadlets 比 **podman generate systemd** 命令更容易使用，因为描述侧重于相关的容器详情，且没有在 **systemd** 下运行容器的技术复杂性。

如需了解更多详细信息，请参阅 Quadlet 上游文档和使 systemd 更好地用于带有 Quadlet 的 Podman 文章。

Jira:RHELPLAN-154432[1]

Container Tools 软件包已更新

更新的 Container Tools RPM 元软件包现已正式发布，其包括 Podman、Buildah、Skopeo、crun 和 runc 工具，现在可用。与之前的版本相比，这个版本应用了一系列 bug 修复和增强。

Podman v4.6 中的显著变化包括：

- **podman kube play** 命令现在支持 **--configmap=*<path>*** 选项，为 Kubernetes YAML 文件提供 pod 容器中使用的环境变量。
- **podman kube play** 命令现在支持 **--configmap** 选项的多个 Kubernetes YAML 文件。
- **podman kube play** 命令现在支持存活度探测中的 containerPort 名称和端口号。
- **podman kube play** 命令现在将 ctrName 作为别名添加到 pod 网络。
- **podman kube play** 和 **podman kube generate** 命令现在支持 SELinux filetype 标签和 ulimit 注解。
- 添加了一个新的命令 **podman secret**，它验证具有指定名称的 secret 是否存在。
- **podman create**,**podman run**,**podman pod create** 和 **podman pod clone** 命令现在支持一个新选项 **--shm-size-systemd**，其允许为特定于 systemd 的挂载限制 tmpfs 大小。
- **podman create** 和 podman run 命令现在支持一个新选项 **--security-opt label=nested**，它允许 SELinux 在受限容器中进行标记。
- Podman 现在支持为 pod 内运行的容器进行自动更新。
- Podman 现在可以使用 SQLite 数据库作为后端来提高稳定性。默认保留 BoltDB 数据库。您可以通过在 **containers.conf** 文件中设置 **database_backend** 字段来选择数据库。
- Podman 现在支持 Quadlets 从容器描述中自动生成一个 **systemd** 服务文件。该描述侧重于相关的容器详情，隐藏了在 **systemd** 下运行容器的技术复杂性。

有关显著变化的详情，请查看 上游发行注记。

Jira:RHELPLAN-154438[1]

Podman 现在支持 Podmansh 登录 shell

从 Podman v4.6 开始，您可以使用 **Podmansh** 登录 shell 管理用户访问权限和控制。将您的设置配置为使用 **/usr/bin/podmansh** 命令作为登录 shell，而不是标准 shell 命令，例如 **/usr/bin/bash**。当用户登录到系统设置时，**podmansh** 命令将用户的会话运行到名为 **podmansh** 的 Podman 容器中。用户登录的容

器是使用 Quadlet 文件定义的，这些文件创建在 **/etc/containers/systemd/users/** 目录中。在这些文件中，将 **[Container]** 部分中的 **ContainerName** 字段设置为 **podmansh**。当用户会话启动时，systemd 会自动启动 **podmansh**，并继续运行直到所有用户会话退出为止。

如需更多信息，请参阅 Podman v4.6.0 引入了 Podmansh:一个革命性的 Login Shell 。

Jira:RHELPLAN-163003[1]

现在，可提供带有 Fulcio 和 Rekor 的 sigstore 签名的客户端

有了 Fulcio 和 Rekor 服务器，您现在可以根据 OpenID Connect (OIDC)服务器身份验证使用短期证书来创建签名，而不是手动管理私钥。以前作为技术预览提供的带有 Fulcio 和 Rekor 的 sigstore 签名的客户端现在完全支持。这个添加的功能只是客户端侧支持，不包括 Fulcio 或 Rekor 服务器。

在 **policy.json** 文件中添加 **fulcio** 部分。要签署容器镜像，请使用 **podman push --sign-by-sigstore=file.yml** 或 **skopeo copy --sign-by-sigstore=*file.yml*** 命令，其中 ***file.yml*** 是 sigstore 签名参数文件。

要验证签名，请在 **policy.json** 文件中添加 **fulcio** 部分和 **rekorPublicKeyPath** 或 **rekorPublicKeyData** 字段。如需更多信息，请参阅 **containers-policy.json** 手册页。

Jira:RHELPLAN-160660[1]

pasta 网络模式现在可用

从 Podman v4.4.1 开始，您可以使用 **pasta** 网络模式。它是默认网络模式 **slirp4netns** 的高性能替换，支持 IPv6 转发。要选择 **pasta** 网络模式，请安装 **passt** 软件包，以使用带有**--network=pasta** 选项的 **podman run** 命令。使用 Podman v4.6，您可以使用 **[network]** 部分下的 **default_rootless_network_cmd** 字段在 **/etc/containers/containers.conf** 配置文件中设置默认的无根网络模式。

Jira:RHELDOCS-16240[1]

UBI 9 微容器镜像不再包含 tzdata安装的 zoneinfo

有了此更新，**tzdata** 软件包提供的时区信息不再包含在 UBI 9 Micro 容器镜像中，从而减少了镜像大小。UBI 9 Minimal 和 UBI 9 微容器是仅 UTC 的，用户应重新安装 **tzdata** 软件包，以获得完整的 **zoneinfo**（如果需要的话）。

Bugzilla:2223028

第 5 章 对外部内核参数的重要更改

本章为系统管理员提供了与 Red Hat Enterprise Linux 9.3 一起分发的内核中显著变化的总结。这些更改可能包括，例如，添加或更新的 **proc** 条目、**sysctl** 和 **sysfs** 默认值、引导参数、内核配置选项或任何明显的行为更改。

新内核参数

amd_pstate=[X86]

使用这个内核参数，您可以扩展 AMD CPU 的性能。可用值包括：

- **disable** - 不启用 **amd_pstate** 作为支持的处理器的默认扩展驱动程序。
- **passive** - 使用带有被动模式的 **amd_pstate** 作为扩展驱动程序。在这个模式中，禁用了自主选择。驱动程序请求一个所需的性能级别，如果保证的性能级别满足它，则平台将尝试匹配同样的性能水平。
- **active** - 使用 **amd_pstate_epp** 驱动程序实例作为扩展驱动程序，如果软件希望向 CPPC 固件的性能(0x0)或能源效率(0xff)倾斜，则驱动程序向硬件提供一个提示。然后，CPPC 电源算法将计算运行时工作负载，并调整实时核频率。
- **guided** - 激活指导的自主模式.驱动程序请求最小和最大的性能级别，平台可自主选择此范围内的性能级别，并适合当前的工作负载。

arm64.nosve=[ARM64]

使用这个内核参数，您可以无条件地禁用 Scalable Vector Extension 支持。

arm64.nosme=[ARM64]

使用这个内核参数，您可以无条件地禁用 Scalable Matrix Extension 支持。

gather_data_sampling=[X86,INTEL]

使用这个内核参数，您可以控制 Gather Data Sampling (GDS)缓解。
GDS 是一个硬件漏洞，它允许对之前存储在向量寄存器中的数据进行非特权推测访问。

默认情况下，此问题在更新的微码中被缓解。缓解可能会有性能影响，但可以被禁用。在没有微码缓解的系统上，禁用 AVX 服务器作为一种缓解。可用值包括：

- **force** - 禁用 AVX 以缓解没有微码缓解的系统。如果存在微码缓解，则无效。已知在带有 bug 的 AVX 枚举用户空间中会导致崩溃。
- **off** - 禁用 GDS 缓解。

nospectre_bhb=[ARM64]

使用这个内核参数，您可以禁用对 Spectre-BHB (分支历史记录注入)漏洞的所有缓解。系统可能允许使用此选项的数据泄漏。

trace_clock=[FTRACE]

使用这个内核参数，您可以设置在引导时用于追踪事件的时钟。可用值包括：

- **local** - 使用每个 CPU 时间戳计数器。
- **global** - 事件时间戳在 CPU 之间被同步。可能比本地时钟慢，但在某些罕见情况下会好些。
- **counter** - 简单的事件(1、2、..)计数，注意，由于基础设施在每个事件中多次抓取时钟，一些计数可能会被跳过。

- **uptime** - 使用 jiffies 作为时间戳。
- **perf** - 使用与 perf 使用的一样的时钟。
- **mono** - 对时间戳使用 **ktime_get_mono_fast_ns（）** 函数。
- **mono_raw** - 对时间戳使用 **ktime_get_raw_fast_ns（）** 函数。
- **boot** - 对时间戳使用 **ktime_get_boot_fast_ns（）** 函数。
  架构可能会添加更多时钟，详情请参阅 **Documentation/trace/ftrace.rst**。

更新的内核参数

cgroup.memory=[KNL]

使用这个内核参数，您可以将选项传给 **cgroup** 内存控制器。

- 这个参数采用的格式：**<string>**
  可用值包括：
- **nosocket** - 禁用套接字内存记帐。
- **nokmem** - 禁用内核内存记帐。
- **[NEW] nobpf** - 禁用 BPF 内存记帐。

hugetlb_free_vmemmap=[KNL]

此内核参数启用在引导时释放与每个 hugetlb 页面关联的未使用 **vmemmap** 页面的功能。要使此参数正常工作，必须启用 **CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP** 配置选项。
这个参数采用的的格式是：**{ on | off (default)}**

可用值包括：

- **on** - 启用此功能
- **off** - 禁用此功能

注意

当启用了 **memory_hotplug.memmap_on_memory** 模块参数时，可以从添加的内存块本身分配 **vmemmap** 页。即使启用了此功能，也无法优化 **vmemmap** 页。其他没有从添加的内存块本身分配的 **vmemmap** 页不会受到影响。

intel_pstate=[X86]

您可以为 CPU 性能扩展使用此内核参数。可用值包括：

- **disable** - 不启用 **intel_pstate** 作为支持的处理器的默认扩展驱动程序。
- **[NEW] active** - 使用 **intel_pstate** 驱动程序绕过 **cpufreq** 的扩展调控层，并为 p-state 选择提供自己的算法。在 active 模式下，**intel_pstate** 提供两种 P-state 选择算法：powersave 和 performance。它们运行的方式取决于硬件管理的 P-states (HWP)功能是否已在处理器中启用了，并可能在处理器型号上也启用了。
- **passive** - 使用 **intel_pstate** 作为扩展驱动程序，但将其配置为与通用 **cpufreq** 调控一起工作（而不是启用其内部调控）。这个模式不能与硬件管理的 P-states (HWP)功能一同使用。

- **force** - 在默认禁止它，而使用**acpi-cpufreq**的系统上启用 **intel_pstate**。强制 **intel_pstate** 驱动程序而不是 **acpi-cpufreq** 可能会禁用平台功能，如热控制和功率封顶，这依赖指示给 OSPM 的 ACPI P-States 信息，因此应谨慎使用。这个选项不能与 **intel_pstate** 驱动程序不支持的处理器一起工作，或不能在使用 **pcc-cpufreq** 而不是 **acpi-cpufreq** 的平台上工作。

- **no_hwp** - 不启用硬件 P 状态控制(HWP)（如果可用）。

- **hwp_only** - 仅在支持硬件 P 状态控制(HWP)的系统上加载 **intel_pstate** （如果可用）。

- **support_acpi_ppc** - 强制 **ACPI _PPC** 性能限制。如果 Fixed ACPI Description Table 将首选的电源管理配置文件指定为 "Enterprise Server" 或 " Performance Server"，则默认打开此功能。

- **per_cpu_perf_limits** - 使用 **cpufreq sysfs** 接口允许每个逻辑 CPU P-State 性能控制限制。

kvm-arm.mode=[KVM,ARM]

使用这个内核参数，您可以选择 KVM/arm64 的操作模式之一。可用值包括：

- **none** - 强制禁用 KVM。

- **nvhe** - 标准的基于 nVHE 的模式，不支持受保护的客户机。

- **protected** - 基于 nVHE 的模式，支持其状态对主机保持私有的客户机。将模式设置为 **protected** 禁用主机的 **kexec** 和休眠。

- **[NEW] nested** - 基于 VHE 的模式，支持嵌套虚拟化。需要至少 ARMv8.3 硬件。 **nested** 选项是实验性的，应该谨慎使用。
  根据硬件支持，默认为 VHE/nVHE。

libata.force=[LIBATA]

使用这个内核参数，您可以强制配置。
格式是以逗号分隔的 "[ID:]VAL" 列表，其中 ID 为 PORT[.DEVICE]。PORT 和 DEVICE 是与端口、链接或设备匹配的十进制数。基本上，它匹配 **libata** 在控制台上打印的 ATA ID 字符串。

- 如果省略整个 ID 部分，则使用最后的 **PORT** 和 **DEVICE** 值。

- 如果尚未指定 ID，则配置应用到所有端口、链接和设备。

- 如果只省略 **DEVICE** 值，则参数应用到端口以及它后面的所有链接和设备。DEVICE 0 要么选择第一个设备，要么选择 PMP 设备后的第一个 fan-out 链接。它不选择主机链接。DEVICE 15，选择主机链接和附加到它的设备。

- VAL 指定要强制的配置。只要没有歧义，则允许使用快捷表示法。例如，1.5 和 1.5G 都适用于 1.5Gbps。
  使用 **libata.force=** 参数，您可以强制以下配置：

- 电缆类型： 40c、80c、short40c、unk、ign 或 sata。使用任何匹配 PORT 的任何 ID。

- SATA 链接速度限制 1.5Gbps 或 3.0Gbps.

- 传输模式：pio[0-7], mwdma[0-4] 和 udma[0-7]。也允许 udma[/][16,25,33,44,66,100,133] 表示法。

- **nohrst**,**nosrst**,**norst**: 抑制硬、软重置和两者都重置。

- **rstonce** ：在热拔链接恢复过程中只尝试一次重置。

- **[NEW] [no]dbdelay**: 在解除一个链接 PHY 和设备存在检测前启用或禁用额外的 200ms 延迟。
- **[no]ncq**: 打开或关闭 NCQ。
- **[no]ncqtrim**: 启用或禁用排队的 DSM TRIM。
- **[NEW] [no]ncqati**: 在 ATI 芯片组上启用或禁用 NCQ trim。
- **[NEW] [no]trim**: 启用或禁用(未排队的) TRIM。
- **[NEW] trim_zero**: 表示 TRIM 命令将数据清零。
- **[NEW] max_trim_128m**: 设置 128M 最大 trim 大小限制。
- **[NEW] [no]dma** ：打开或关闭 DMA 传输。
- **atapi_dmadir**: 启用 ATAPI DMADIR 网桥支持。
- **atapi_mod16_dma** ：为不是 16 字节倍数的命令启用 ATAPI DMA。
- **[no]dmalog** ：启用或禁用 READ LOG DMA EXT 命令来访问日志。
- **[no]iddevlog**: 启用或禁用对识别设备数据日志的访问。
- **[no]logdir**: 启用或禁用对通用目的日志目录的访问。
- **[NEW] max_sec_128**: 将传输大小限制设置为 128 个扇区。
- **[NEW] max_sec_1024**: 将传输大小限制设置为 1024 个扇区或清除传输大小限制。
- **[NEW] max_sec_lba48**: 将传输大小限制设置为 65535 个扇区或清除传输大小限制。
- **[NEW] [no]lpm**: 启用或禁用链路电源管理。
- **[NEW] [no]setxfer**: 指示是否应跳过传输速度模式设置。
- **[NEW] [no]fua**: 禁用或启用 FUA (Force unit Access)对支持此功能的设备的支持。
- **dump_id** ：转储 IDENTIFY 数据。
- **disable** : 禁用这个设备。

注意

如果有更改同一属性的多个匹配配置，则使用最后一个。

mitigations=[X86,PPC,S390,ARM64]

使用这个内核参数，您可以控制对 CPU 漏洞的可选缓解。这是一组策展的、架构独立的选项，每个选项都是现有的特定架构选项的聚合。可用值包括：

- **off** - 禁用所有可选 CPU 缓解。这提高了系统性能，但也将用户暴露给多个 CPU 漏洞。 **off** 值等同于：
  - if nokaslr then kpti=0 [ARM64]
  - gather_data_sampling=off [X86]

- kvm.nx_huge_pages=off [X86]
- l1tf=off [X86]
- mds=off [X86]
- mmio_stale_data=off [X86]
- no_entry_flush [PPC]
- no_uaccess_flush [PPC]
- nobp=0 [S390]
- nopti [X86,PPC]
- nospectre_bhb [ARM64]
- nospectre_v1 [X86,PPC]
- nospectre_v2 [X86,PPC,S390,ARM64]
- retbleed=off [X86]
- spec_store_bypass_disable=off [X86,PPC]
- spectre_v2_user=off [X86]
- srbds=off [X86,INTEL]
- ssbd=force-off [ARM64]
- tsx_async_abort=off [X86]
  例外： 当 **kvm.nx_huge_pages** =force 时，这对 **kvm.nx_huge_pages=force** 没有任何影响。

- **auto**（默认）- 缓解所有 CPU 漏洞，但启用 SMT，即使它存在安全漏洞。这适用于不希望在内核升级过程中禁用 SMT，或者有其他方法避免 SMT 攻击的用户。
- **auto,nosmt** - 缓解所有 CPU 漏洞，如果需要，禁用 SMT。这适用于需要实施所有缓解方案的用户，即使这意味着会丢失 SMT 的功能。**auto,nosmt** 选项等同于：
  - l1tf=flush,nosmt [X86]
  - mds=full,nosmt [X86]
  - tsx_async_abort=full,nosmt [X86]
  - mmio_stale_data=full,nosmt [X86]
  - retbleed=auto,nosmt [X86]

nomodeset

使用这个内核参数，您可以禁用内核模式设置。大多数系统的固件会设置显示模式，并为输出提供帧缓冲内存。使用 **nomodeset** 时，如果 DRM 和 **fbdev** 驱动程序可能替换预初始化的输出，则不会加载它们。只有系统帧缓冲才能使用。驱动程序不会执行显示模式更改或加速渲染。
此参数在错误回退或测试和调试时特别有用。

rdt=[HW,X86,RDT]

使用这个内核参数，您可以打开或关闭单个 RDT 功能。列表中包括：**cmt**,**mbmtotal**,**mbmlocal**,**l3cat**,**l3cdp**,**l2cat**,**l2cdp**,**mba**,**smba**,**bmec**.
例如，要打开 **cmt** 并关闭 **mba**，请使用：

```
rdt=cmt,!mba
```

rodata=[KNL]

使用这个内核参数，您可以禁用只读内核映射。可用选项包括：

- **on** - 将只读内核内存标记为只读（默认）。
- **off** - 使只读内核内存对调试可写。
- **[NEW] full** - 将只读内核内存和别名标记为只读 [arm64]。

删除的内核参数

nobats=[PPC]

使用这个内核参数，您可以禁止在 "Classic" PPC 核上对映射内核 lowmem 使用 BATs。

noltlbs=[PPC]

使用这个内核参数，您可以禁止对 PPC40x 和 PPC8xx 上的内核 lowmem 映射使用巨页和 tlb 条目。

swapaccount=[0|1]=[KNL]

使用这个内核参数，您可以在内存资源控制器中启用或禁用交换记账。如需更多信息，请参阅 **Documentation/admin-guide/cgroup-v1/memory.rst**。

第 6 章 设备驱动程序

6.1. 新驱动程序

网络驱动程序

- MediaTek MT7601U (USB)支持(**mt7601u**)，增加了对基于 MT7601U 的无线 USB 加密狗的支持（仅在 64 位 ARM 架构中）
- MediaTek MT76x0E (PCIe)支持(**mt76x0e**)，增加了对基于 MT7610/MT7630 的无线 PCIe 设备的支持（仅适在 64 位 ARM 架构中）
- MediaTek MT76x0U (USB)支持(**mt76x0u**)，增加了对基于 MT7610U 的无线 USB 2.0 加密狗的支持（仅在 64 位 ARM 架构中）
- MediaTek MT76x2E (PCIe)支持(**mt76x2e**)，增加了对基于 MT7612/MT7602/MT7602/MT7662 的无线 PCIe 设备的支持（仅在 64 位 ARM 架构中）
- MediaTek MT76x2U (USB)支持(**mt76x2u**)，增加了对基于 MT7612U 的无线 USB 3.0 加密狗的支持（仅在 64 位 ARM 架构中）
- MediaTek MT7921E (PCIe)支持(**mt7921e**)，增加了对 MT7921E 802.11ax 2x2:2SS 无线设备的支持（仅在 64 位 ARM 架构中）
- 基于 Atheros 驱动程序 802.11n HTC 的无线设备(**ath9k_htc**)（仅在 64 位 ARM 架构中）
- Broadcom 802.11n 无线 LAN 驱动程序(**brcmsmac**)（仅在 64 位 ARM 架构中）
- Broadcom 802.11n 无线 LAN 驱动程序工具(**brcmutil**)（仅在 64 位 ARM 架构中）
- Broadcom 802.11 无线 LAN fullmac 驱动程序(**brcmfmac**)（仅在 64 位 ARM 架构中）
- Qualcomm Atheros 802.11ac 无线 LAN 卡的核心模块(**th10k_core**)（仅在 64 位 ARM 架构中）
- Qualcomm Atheros 802.11ax 无线 LAN 卡的核心模块 (**ath11k**)（仅在 64 位 ARM 架构中）
- WWAN 框架的设备模拟器(**wwan_hwsim**)
- 对 Qualcomm Atheros 802.11ac WLAN PCIe/AHB 设备的驱动程序支持 (**ath10k_pci**)（仅在 64 位 ARM 架构中）
- 对 Qualcomm Technologies 802.11ax WLAN PCIe 设备的驱动程序支持(**ath11k_pci**)（仅在 64 位 ARM 架构中）
- 用于 Linux 的 Intel® Wireless Wi-Fi 驱动程序(**iwlwifi**)（仅适用于 64 位 ARM 架构）
- 用于 Linux 的 Intel® Wireless Wi-Fi Link AGN 驱动程序(**iwldvm**)-（仅适用于 64 位 ARM 架构）
- IOSM 驱动程序(**iosm**)
- Marvell WiFi-Ex Driver 版本 1.0 (**mwifiex**)（仅在 64 位 ARM 架构中）
- Marvell WiFi-Express 驱动程序版本 1.0 (**mwifiex_pcie**)（仅在 64 位 ARM 架构中）
- Marvell WiFi-Ex SDIO 驱动程序 1.0 (**mwifiex_sdio**)（仅在 64 位 ARM 架构中）
- Marvell WiFi-Ex USB 驱动程序版本 1.0 (**mwifiex_usb**)（仅在 64 位 ARM 架构中）

- MediaTek PCIe 5G WWAN 模式m T7xx 驱动程序(**mtk_t7xx**)
- MHI 上的网络/MBIM (**mhi_wan_mbim**)（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架中）
- rtlwifi的 PCI 基本驱动程序 (**rtl_pci**)（仅在 64 位 ARM 架构中）
- Ralink RT2800 库版本 2.3.0 (**rt2800lib**)（仅在 64 位 ARM 架构中）
- Ralink RT2800 PCI 和 PCMCIA Wireless LAN 驱动程序版本 2.3.0 (**rt2800pci**)（仅在 64 位 ARM 架构中）
- Ralink RT2800 USB Wireless LAN 驱动程序版本 2.3.0 (**rt2800usb**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8821c 驱动程序(**rtw88_8821c**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8821ce 驱动程序(**rtw88_8821ce**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8822b 驱动程序(**rtw88_8822b**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8822be 驱动程序(**rtw88_8822be**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8822c 驱动程序(**rtw88_8822c**)-（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless 8822ce 驱动程序(**rtw88_8822ce**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless core module (**rtw88_core**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ac wireless PCI 驱动程序(**rtw88_pci**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ax wireless 8852A 驱动程序(**rtw89_8852a**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ax wireless 8852AE 驱动程序(**rtw89_8852ae**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ax wireless 8852B 驱动程序(**rtw89_8852b**)（仅在 64 位 ARM 架构和 AMD 和 Intel 64 位构架中）
- Realtek 802.11ax wireless 8852BE 驱动程序(**rtw89_8852be**)（仅在 64 位 ARM 架构和 AMD 和 Intel 64 位构架中）
- Realtek 802.11ax wireless core module (**rtw89_core**)（仅在 64 位 ARM 架构中）
- Realtek 802.11ax wireless PCI 驱动程序(**rtw89_pci**)（仅在 64 位 ARM 架构中）
- Realtek 802.11n PCI 无线内核(**btcoexist**)（仅在 64 位 ARM 架构中）
- Realtek 802.11n PCI 无线内核(**rtlwifi**)（仅在 64 位 ARM 架构中）
- Realtek 802.11n wireless 8723d 驱动程序(**rtw88_8723d**)（仅在 64 位 ARM 架构中）
- Realtek 802.11n wireless 8723de 驱动程序(**rtw88_8723de**)（仅在 64 位 ARM 架构中）
- Realtek 8188E 802.11n PCI 无线(**rtl8188ee**) 在 64 位 ARM 架构中）
- Realtek 8192C/8188C 802.11n PCI 无线(**rtl8192c-common**)（仅在 64 位 ARM 架构中）
- Realtek 8192C/8188C 802.11n PCI 无线(**rtl8192ce**)（仅在 64 位 ARM 架构中）

- Realtek 8192C/8188C 802.11n USB 无线(**rtl8192cu**)（仅在 64 位 ARM 架构中）
- Realtek 8192DE 802.11n Dual Mac PCI 无线(**rtl8192de**)（仅在 64 位 ARM 架构中）
- Realtek 8192EE 802.11n PCI 无线(**rtl8192ee**)（仅在 64 位 ARM 架构中）
- Realtek 8192S/8191S 802.11n PCI 无线(**rtl8192se**)（仅在 64 位 ARM 架构中）
- Realtek 8723BE 802.11n PCI 无线(**rtl8723be**)（仅在 64 位 ARM 架构中）
- Realtek 8723E 802.11n PCI 无线(**rtl8723ae**)（仅在 64 位 ARM 架构中）
- Realtek 8821ae 802.11ac PCI 无线(**rtl8821ae**)（仅在 64 位 ARM 架构中）
- Realtek RTL8723AE/RTL8723BE 802.11n PCI 无线常规例程(**rtl8723-common**)（仅在 64 位 ARM 架构中）
- rt2800 MMIO 库版本 2.3.0 (**rt2800mmio**)（仅在 64 位 ARM 架构中）
- rt2x00 库版本 2.3.0 (**rt2x00lib**)（仅在 64 位 ARM 架构中）
- rt2x00 mmio 库版本 2.3.0 (**rt2x00mmio**)（仅在 64 位 ARM 架构中）
- rt2x00 pci 库版本 2.3.0 (**rt2x00pci**)（仅在 64 位 ARM 架构中）
- rt2x00 usb library 版本 2.3.0 (**rt2x00usb**)（仅在 64 位 ARM 架构中）
- RTL8XXXu USB mac80211 Wireless LAN 驱动程序 (**rtl8xxxu**)（仅在 64 位 ARM 架构中）
- Atheros 无线 802.11n LAN 卡的共享库(**th9k_common**)（仅在 64 位 ARM 架构中）
- Aros 无线 LAN 卡的共享库(**ath**)（仅在 64 位 ARM 架构中）
- 支持 Atheros 802.11n 无线 LAN 卡(**th9k_hw**)（仅在 64 位 ARM 架构中）
- 支持 Atheros 802.11n 无线 LAN 卡(**th9k**)（仅在 64 位 ARM 架构中）
- Linux 的新 Intel® 无线 AGN 驱动程序(**iwlmvm**)（仅在 64 位 ARM 架构中）
- Thunderbolt/USB4 网络驱动程序(**thunderbolt_net**)
- rtlwifi 的 USB 基本驱动程序(**rtl_usb**)（仅在 64 位 ARM 架构中）

图形驱动程序和各种驱动程序

- Atheros AR30xx 固件驱动程序 1.0 (**th3k**)（仅在 64 位 ARM 架构中）
- BlueFRITZ!USB 驱动程序版本 1.2 (**bfusb**)（仅在 64 位 ARM 架构中）
- 蓝牙 HCI UART 驱动程序版本 2.3 (**hci_uart**)（仅在 64 位 ARM 架构中）
- Broadcom 设备版本 0.1 (**btbcm**)的蓝牙支持（只在 64 位 ARM 架构中）
- Intel 设备版本 0.1 的蓝牙支持(**btintel**)（仅在 64 位 ARM 架构中）
- MediaTek 设备的蓝牙支持 0.1 (**btmtk**)（仅在 64 位 ARM 架构中）
- 对 Realtek 设备版本 0.1 的蓝牙支持(**btrtl**)（仅在 64 位 ARM 架构中）

- 蓝牙虚拟 HCI 驱动程序版本 1.5 (**hci_vhci**)（仅在 64 位 ARM 架构中）
- Broadcom Blutonium 固件驱动程序版本 1.2 (**bcm203x**)（仅在 64 位 ARM 架构中）
- Digianswer Bluetooth USB 驱动程序版本 0.11 (**bpa10x**)（仅在 64 位 ARM 架构中）
- 通用蓝牙 SDIO 驱动程序 0.1 (**btsdio**)（仅在 64 位 ARM 架构中）
- 通用蓝牙 USB 驱动程序版本 0.8 (**btusb**)（仅在 64 位 ARM 架构中）
- Marvell Bluetooth 驱动程序版本 1.0 (**btmrvl**)（仅在 64 位 ARM 架构中）
- Marvell BT-over-SDIO 驱动程序 1.0 (**btmrvl_sdio**)（仅在 64 位 ARM 架构中）
- BMC IPMI SSIF 接口的 Linux 设备驱动程序(**ssif_bmc**)（仅在 64 位 ARM 架构中）
- vTPM Driver 版本 0.1 (**tpm_vtpm_proxy**)
- AMD P-state 驱动程序 Test 模块(**amd-pstate-ut**)（仅在 AMD 和 Intel 64 位构架中）
- Compute Express Link (CXL) ACPI 驱动程序(**cxl_acpi**)（仅在 64 位 ARM 架构和 AMD 和 Intel 64 位构架中）
- Compute Express Link (CXL) core 驱动程序 (**cxl_core**)
- Compute Express Link (CXL)端口驱动程序(**cxl_port**)
- NVIDIA Tegra GPC DMA Controller 驱动程序(**tegra186-gpc-dma**)（仅在 64 位 ARM 架构中）
- DRM Buddy Allocator (**drm_buddy**)（仅在 64 位 IBM Z 架构中）
- DRM 显示适配器帮助程序(**drm_display_helper**)（仅在 64 位 IBM Z 构架中）
- EVision 设备的 HID 驱动程序 (**hid-evision**)（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）
- Texas Instruments INA3221 HWMon 驱动程序(**ina3221**)（仅在 64 位 ARM 架构中）
- I3C 内核(**i3c**)（仅在 64 位 ARM 架构中）
- Silvaco dual-role I3C master 驱动程序(**svc-i3c-master**)（仅在 64 位 ARM 架构中）
- Microsoft Azure Network Adapter IB 驱动程序(**mana_ib**)（仅在 AMD 和 Intel 64 位构架中）
- 软 RDMA 传输(**rdma_rxe**)
- I.MX8MP 互连驱动程序 - 用于 i.MX SOC 的通用互连驱动程序(**imx8mp-interconnect**)（仅在 64 位 ARM 架构中）
- Linux USB 视频类(**uvc**)（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）
- videobuf2 的常见内存处理例程(**videobuf2-memops**)（仅在 64 位 ARM 架构中）
- 用于**cec**驱动程序的设备节点注册（仅在 64 位 IBM Z 构架中）
- 设备节点注册介质驱动程序(**mc**)（仅在 64 位 ARM 架构中）

- Linux 2 视频的驱动程序帮助程序框架(**videobuf2-v4l2**)（仅在 64 位 ARM 架构中）
- 媒体缓冲区内核框架(**videobuf2-common**)（仅在 64 位 ARM 架构中）
- USB 视频类驱动程序版本 1.1.1 (**uvcvideo**)（仅在 64 位 ARM 架构中）
- V4L2 DV Timings Helper Functions (**v4l2-dv-timings**)（仅在 64 位 ARM 架构中）
- Video4Linux2 核心驱动程序(**videodev**)（仅在 64 位 ARM 架构中）
- videobuf2 的 vmalloc 内存处理例程(**videobuf2-vmalloc**)（仅在 64 位 ARM 架构中）
- SPI NOR 的框架(**spi-nor**)（仅在 64 位 ARM 架构中）
- Marvell CN10K DRAM subsystem (DSS) PMU (**marvell_cn10k_ddr_pmu**)（仅在 64 位 ARM 架构中）
- Marvell CN10K LLC-TAD Perf 驱动程序(**marvell_cn10k_tad_pmu**)（仅在 64 位 ARM 架构中）
- Intel Meteor Lake PCH pinctrl/GPIO 驱动程序(**pinctrl-meteorlake**)（仅在 AMD 和 Intel 64 位构架中）
- Intel In Field Scan (IFS)设备(**intel_ifs**)（仅在 AMD 和 Intel 64 位构架中）
- NVIDIA WMI EC Backlight 驱动程序(**nvidia-wmi-ec-backlight**)（仅在 AMD 和 Intel 64 位构架中）
- QMI encoder/decoder 帮助程序(**qmi_helpers**)（仅在 64 位 ARM 架构中）
- AMD SoundWire 驱动程序(**soundwire-amd**)（仅在 AMD 和 Intel 64 位构架中）
- NVIDIA Tegra114 SPI Controller 驱动程序(**spi-tegra114**)（仅在 64 位 ARM 架构中）
- STMicroelectronics STUSB160x Type-C 控制器驱动程序(**stusb160x**)（仅在 64 位 ARM 架构中）
- MLX5 VFIO PCI - MLX5 设备系列的用户级元驱动程序(**mlx5-vfio-pci**)

6.2. 更新的驱动程序

网络驱动程序更新

- Realtek RTL8152/RTL8153 Based USB Ethernet Adapters (**r8152**)已更新至 v1.12.13 版本（只在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）。

存储驱动程序更新

- Broadcom MegaRAID SAS 驱动程序(**megaraid_sas**)已更新至版本 07.725.01.00-rc1（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）。
- Microchip Smart Family Controller 的驱动程序(**smartpqi**)已更新至版本 2.1.22-040（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）。
- Emulex LightPulse Fibre Channel SCSI 驱动程序(**lpfc**)已更新至版本 0:14.2.0.12（仅在 64 位 ARM 架构、IBM Power Systems、Little Endian 、AMD 和 Intel 64 位构架中）。
- MPI3 Storage Controller Device Driver (**mpi3mr**)已更新至版本 8.4.1.0.0。

第 7 章 可用的 BPF 功能

本章提供了这个 Red Hat Enterprise Linux 9 次版本的 kernel 中 **Berkeley Packet Filter** (**BPF**)功能的完整列表。表包括：

- 系统配置和其他选项
- 可用的程序类型和支持的帮助程序
- 可用的映射类型

本章包含 **bpftool feature** 命令自动生成的输出。

表 7.1. 系统配置和其他选项

选项	值
unprivileged_bpf_disabled	2 (bpf() 系统调用限制为特权用户，管理员可以改变。)
JIT 编译器	1（启用）
JIT 编译器强化	1（为非特权用户启用）
JIT 编译器 kallsyms 导出	1（为 root 用户启用）
非特权用户的 JIT 的内存限制	528482304
CONFIG_BPF	y
CONFIG_BPF_SYSCALL	y
CONFIG_HAVE_EBPF_JIT	y
CONFIG_BPF_JIT	y
CONFIG_BPF_JIT_ALWAYS_ON	y
CONFIG_DEBUG_INFO_BTF	y
CONFIG_DEBUG_INFO_BTF_MODULES	y
CONFIG_CGROUPS	y
CONFIG_CGROUP_BPF	y
CONFIG_CGROUP_NET_CLASSID	y
CONFIG_SOCK_CGROUP_DATA	y

选项	值
CONFIG_BPF_EVENTS	y
CONFIG_KPROBE_EVENTS	y
CONFIG_UPROBE_EVENTS	y
CONFIG_TRACING	y
CONFIG_FTRACE_SYSCALLS	y
CONFIG_FUNCTION_ERROR_INJECTION	y
CONFIG_BPF_KPROBE_OVERRIDE	n
CONFIG_NET	y
CONFIG_XDP_SOCKETS	y
CONFIG_LWTUNNEL_BPF	y
CONFIG_NET_ACT_BPF	m
CONFIG_NET_CLS_BPF	m
CONFIG_NET_CLS_ACT	y
CONFIG_NET_SCH_INGRESS	m
CONFIG_XFRM	y
CONFIG_IP_ROUTE_CLASSID	y
CONFIG_IPV6_SEG6_BPF	y
CONFIG_BPF_LIRC_MODE2	n
CONFIG_BPF_STREAM_PARSER	y
CONFIG_NETFILTER_XT_MATCH_BPF	m
CONFIG_BPFILTER	n
CONFIG_BPFILTER_UMH	n

选项	值
CONFIG_TEST_BPF	m
CONFIG_HZ	1000
bpf() syscall	可用
大型程序大小限制	可用
绑定的循环支持	可用
ISA 扩展 v2	可用
ISA 扩展 v3	可用

表 7.2. 可用的程序类型和支持的帮助程序

程序类型	可用的帮助程序
socket_filter	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
kprobe	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
sched_cls	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_skb_load_bytes, bpf_skb_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_PROFILE_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_latencies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_xxxx_output, bpf_zFCP_reserve, bpf_numpy_reserve, bpf_wagon_submit, bpf_ iwl_discard, bpf_ iwl_query, bpf_csum_level, bpf_categories_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_wagon_to_tcp_timewait_sock, bpf_PROFILE_to_tcp_request_sock, bpf_categories_to_udp6_sock, bpf_categories_to_udp6_sock, bpf_wagon_wagon, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_ iwl, bpf_ktime_get_MAPPING_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_for_each_map_elem, bpf_ProductShortName, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_PROFILEmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_PROFILE_to_mptcp_sock, bpf_PROFILE_from_mem, bpf_PROFILE_reserve_PROFILE, bpf_PROFILE_submit_PROFILE, bpf_PROFILE_discard_criu, bpf_PROFILE_read, bpf_PROFILE_write, bpf_PROFILE_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
sched_act	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_skb_load_bytes, bpf_skb_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_PROFILE_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_latencies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_xxxx_output, bpf_zFCP_reserve, bpf_numpy_reserve, bpf_wagon_submit, bpf_ iwl_discard, bpf_ iwl_query, bpf_csum_level, bpf_categories_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_wagon_to_tcp_timewait_sock, bpf_PROFILE_to_tcp_request_sock, bpf_categories_to_udp6_sock, bpf_categories_to_udp6_sock, bpf_wagon_wagon, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_ iwl, bpf_ktime_get_MAPPING_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_for_each_map_elem, bpf_ProductShortName, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_PROFILEmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_PROFILE_to_mptcp_sock, bpf_PROFILE_from_mem, bpf_PROFILE_reserve_PROFILE, bpf_PROFILE_submit_PROFILE, bpf_PROFILE_discard_criu, bpf_PROFILE_read, bpf_PROFILE_write, bpf_PROFILE_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
xdp	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_redirect, bpf_perf_event_output, bpf_csum_diff, bpf_get_current_task, bpf_get_numa_node_id, bpf_xdp_adjust_head, bpf_redirect_map, bpf_xdp_adjust_meta, bpf_xdp_adjust_tail, bpf_fib_lookup, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_xdp_get_buff_len, bpf_xdp_load_bytes, bpf_xdp_store_bytes, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
perf_event	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_skb_cgroup_id, bpf_get_local_storage, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_sk_cgroup_id, bpf_sk_ancestor_cgroup_id, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
cgroup_sock	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_in	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_push_encap, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
lwt_out	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_perf_event_output, bpf_skb_load_bytes, bpf_skb_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_ 0.11.0-64, bpf_ktime_boot_ns, bpf_probe_read_output, bpf_probe_read_output, bpf_categories_reserve, bpf_wagon_submit, bpf_ iwl_discard, bpf_categories_query, bpf_categories_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_wagon_to_tcp_timewait_sock, bpf_PROFILE_to_tcp_sock, bpf_categories_to_udp6_sock, bpf_categories_to_udp6_sock, bpf_unmarshal_wagon, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_PROFILE, bpf_ktime_get_PROFILE_ns, bpf_for_each_map_elem, bpf_PROFILE, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_ iwl_to_unix_sock, bpf_loop, bpf_loopmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_criu_to_mptcp_sock, bpf_PROFILE_from_mem, bpf_PROFILE_reserve_numpy, bpf_zFCP_submit_numpy, bpf_criu_submit_numpy, bpf_wagon_discard_criu, bpf_ProductShortName_read, bpf_ iwl_write, bpf_unmarshal_data, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_xmit	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_lwt_push_encap, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
sock_ops	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_sock_map_update, bpf_getsockopt, bpf_sock_ops_cb_flags_set, bpf_sock_hash_update, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_adjust_room, bpf_sk_redirect_map, bpf_sk_redirect_hash, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
cgroup_device	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_csum64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_categories_output, bpf_categories_reserve, bpf_numpy_submit, bpf_PROFILE_discard, bpf_PROFILE_query, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_frequency, bpf_for_each_map_elem, bpf_for_each_map_elem, bpf_ bpf_PROFILE, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_loopmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_csum_from_mem, bpf_criu_reserve_PROFILE, bpf_unmarshal_submit_criu, bpf_numpy_discard_criu, bpf_categories_read, bpf_unmarshal_write, bpf_unmarshal_data, bpf_ktime_get_tai_ns, bpf_user_criu_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_msg	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_msg_redirect_hash, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_criu64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ktime_output, bpf_numpy_reserve, bpf_wagon_submit, bpf_ iwl_discard, bpf_ iwl_query, bpf_categories_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_PROFILE_to_tcp_timewait_sock, bpf_ iwl_to_tcp_request_sock, bpf_categories_to_udp6_sock, bpf_PROFILE_to_tcp_sock, bpf_criu_to_tcp_sock, bpf_ iwl_to_tcp_sock bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_PROFILE, bpf_ktime_get_PROFILE_ns, bpf_for_each_map_elem, bpf_numpy, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_ iwlmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_PROFILE_to_mptcp_sock, bpf_criu_from_mem, bpf_PROFILE_reserve_zFCP, bpf_PROFILE_submit_numpy, bpf_PROFILE_discard_numpy, bpf_criu_discard_numpy, bpf_wagon_read, bpf_ProductShortName_write, bpf_unmarshal_data, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
raw_tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sock_addr	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_getsockopt, bpf_bind, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_ CephFS_lookup_tcp, bpf_strtol, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_ PlacementRule64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_categories_output, bpf_categories_reserve, bpf_numpy_submit, bpf_PROFILE_discard, bpf_PROFILE_to_tcp6_sock, bpf_ iwl_to_tcp_sock, bpf_PROFILE_to_tcp_timewait_sock, bpf_PROFILE_to_tcp_sock, bpf_PROFILE_to_tcp_sock, bpf_ iwl_to_tcp_sock, bpf_ , bpf_PROFILE_to_udp6_sock, bpf_PROFILE_setuptools, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_PROFILE, bpf_ktime_get_MAPPING_ns, bpf_for_each_map_elem, bpf_zFCP, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_loopmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_2023_to_mptcp_sock, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem bpf_wagon_from_mem, bpf_numpy_reserve_criu, bpf_setuptools_submit_criu, bpf_categories_discard_criu, bpf_PROFILE_read, bpf_criu_write, bpf_wagon_data, bpf_ktime_get_tai_ns, bpf_user_mvapich_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
lwt_seg6local	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_perf_event_output, bpf_skb_load_bytes, bpf_skb_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_seg6_store_bytes, bpf_lwt_seg6_adjust_srh, bpf_lwt_seg6_action, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_wagon64, bpf_ktime_get_boot_ns, bpf_categories_output, bpf_PROFILE_reserve, bpf_PROFILE_discard, bpf_categories_query, bpf_categories_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_2023_to_tcp_sock, bpf_criu_to_tcp_timewait_sock, bpf_ProductShortName_to_tcp_request_sock, bpf_PROFILE_to_udp6_sock, bpf_PROFILE_PROFILE, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_xxxx, bpf_ktime_get_PROFILE_ns, bpf_for_each_map_elem, bpf_per_cpu_ptr, bpf_get_current_task_xxxx, bpf_ktime_get_ptr, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_netnsmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_2023_to_mptcp_sock, bpf_wagon_from_mem, bpf_numpy_reserve_criu, bpf_setuptools_submit_criu, bpf_categories_discard_criu, bpf_PROFILE_read, bpf_criu_write, bpf_wagon_data, bpf_ktime_get_tai_ns, bpf_user_mvapich_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lirc_mode2	不支持
sk_reuseport	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_skb_load_bytes_relative, bpf_sk_select_reuseport, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
flow_dissector	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtoul, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_categories64, bpf_ktime_get_boot_ns, bpf_categories_output, bpf_categories_reserve, bpf_categories_submit, bpf_PROFILE_discard, bpf_PROFILE_to_tcp6_sock, bpf_PROFILE_to_tcp_sock, bpf_PROFILE_to_tcp_sock, bpf_PROFILE_to_tcp_timewait_sock, bpf_ProductShortName_to_tcp_request_sock, bpf_PROFILE_to_udp6_sock, bpf_PROFILE_PROFILE, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_xxxx, bpf_ktime_get_PROFILE_ns, bpf_for_each_map_elem, bpf_per_cpu_ptr, bpf_get_current_task_xxxx, bpf_ktime_get_ptr, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_PROFILE_to_unix_sock, bpf_loop, bpf_netnsmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_2023_to_mptcp_sock, bpf_wagon_from_mem, bpf_numpy_reserve_criu, bpf_setuptools_submit_criu, bpf_categories_discard_criu, bpf_PROFILE_read, bpf_criu_write, bpf_wagon_data, bpf_ktime_get_tai_ns, bpf_user_mvapich_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sysctl	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_MAPPING64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_PROFILE_output, bpf_numpy_reserve, bpf_PROFILE_submit, bpf_PROFILE_discard, bpf_ktime_discard, bpf_unmarshal_query, bpf_ iwl_numpy, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_PROFILE, bpf_ktime_get_categories_ns, bpf_for_each_map_elem, bpf_PROFILE, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_loopmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_wagon_from_mem, bpf_PROFILE_reserve_PROFILE, bpf_ 0.11.0-_submit_PROFILE, bpf_criu_discard_PROFILE, bpf_criu_read, bpf_unmarshal_write, bpf_unmarshal_data, bpf_ktime_get_tai_ns, bpf_user_categories_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

程序类型	可用的帮助程序
raw_tracepoint_writable	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sockopt	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
tracing	不支持
struct_ops	不支持
ext	不支持
lsm	不支持

程序类型	可用的帮助程序
sk_lookup	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtoul, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_categories64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_PROFILE_output, bpf_criu_reserve, bpf_categories_submit, bpf_categories_discard, bpf_PROFILE_query, bpf_numpy_to_tcp6_sock, bpf_categories_to_tcp_sock, bpf_PROFILE_to_tcp_timewait_sock, bpf_ iwl_to_tcp_request_sock, bpf_categories_to_udp6_sock, bpf_PROFILE_criu, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_get_current_task_PROFILE, bpf_ktime_get_categories_ns, bpf_for_each_map_elem, bpf_ProductShortName, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_ iwl_to_unix_sock, bpf_loop, bpf_framemp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_categories_to_mptcp_sock, bpf_PROFILE_from_mem, bpf_PROFILE_reserve_PROFILE, bpf_criu_submit_PROFILE, bpf_PROFILE_discard_criu, bpf_PROFILE_read, bpf_PROFILE_write, bpf_PROFILE_data, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
syscall	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_get_socket_cookie, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_sk_storage_delete, bpf_send_signal, bpf_skb_output, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_ netobserv64, bpf_get_ns_current_pid_tgid, bpf_xdp_output, bpf_xdp_output, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_unmarshal_output, bpf_PROFILE_reserve, bpf_PROFILE_discard, bpf_ netobserv_query, bpf_PROFILE_to_tcp6_sock, bpf_zFCP_to_tcp_sock, bpf_PROFILE_to_tcp_sock, bpf_PROFILE_to_tcp_timewait_sock, bpf_PROFILE_to_tcp_request_sock, bpf_PROFILE_to_udp6_sock, bpf_get_task_stack, bpf_d_path, bpf_copy_from_user, bpf_PROFILE_numpy, bpf_per_cpu_ptr, bpf_ this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_criu, bpf_sock_from_file, bpf_for_each_map_elem, bpf_categories, bpf_sys_bpf, bpf_criu_find_by_name_kind, bpf_sys_close, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_categories_to_unix_sock, bpf_kallsyms_lookup_name, bpf_find_vma, bpf_loop, bpf_mvapichmp, bpf_xdp_get_buff_len, bpf_copy_from_user_task, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_PROFILE_to_mptcp_sock, bpf_PROFILE_from_mem, bpf_criu_reserve_categories, bpf_numpy_submit_criu, bpf_PROFILE_discard_PROFILE, bpf_zFCP_read, bpf_csum_write, bpf_criu_data, bpf_criu_submit_data, bpf_ktime_get_tai_ns, bpf_user_PROFILE_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

表 7.3. 可用的映射类型

映射类型	可用
hash	是
数组	是
prog_array	是
perf_event_array	是
percpu_hash	是
percpu_array	是
stack_trace	是
cgroup_array	是
lru_hash	是
lru_percpu_hash	是
lpm_trie	是
array_of_maps	是
hash_of_maps	是
devmap	是
sockmap	是
cpumap	是
xskmap	是
sockhash	是
cgroup_storage	是
reuseport_sockarray	是
percpu_cgroup_storage	是
queue	是
queue	是

映射类型	可用
sk_storage	是
devmap_hash	是
struct_ops	是
ringbuf	是
inode_storage	是
task_storage	是
bloom_filter	是
user_ringbuf	是
cgrp_storage	是

第 8 章 程序错误修复

这部分论述了 Red Hat Enterprise Linux 9.3 中修复的、对用户有严重影响的 bug。

8.1. 安装程序和镜像创建

安装程序现在可以正确地处理 url Kickstart 命令的 --proxy 选项

在以前的版本中，安装程序无法正确处理 **url** Kickstart 命令的 **--proxy** 选项。因此，您无法使用指定的代理来获取安装镜像。有了此更新，这个问题已被解决，**--proxy** 选项现在可以按预期正常工作。

Bugzilla:2177219

liveimg 的 --noverifyssl 选项不再为使用 HTTPS 下载的镜像检查服务器的证书

在以前的版本中，安装程序会忽略 **liveimg** Kickstart 命令中的 **--noverifyssl** 选项。因此，如果无法为使用 HTTPS 协议下载的镜像验证服务器证书，安装过程会失败。有了这个更新，这个问题已解决，**liveimg** Kickstart 命令的 **--noverifyssl** 选项工作正常。

Bugzilla:2157921

Anaconda 现在验证 FIPS 要求的 LUKS 密码短语

在以前的版本中，Anaconda 不检查 LUKS 密码短语的长度是否满足 FIPS 要求，即使底层工具执行了这个检查。因此，使用小于 8 个字符的密码短语在 FIPS 模式下安装会导致安装程序过早停止。

有了此更新，安装程序已得到了改进，可以验证并强制执行密码短语的最小长度。因此，如果在 FIPS 模式下使用的 LUKS 密码短语太短，安装程序会通知，并防止意外停止。

Bugzilla:2163497

xfsprogs 的新版本不再缩小 /boot的大小

在以前的版本中，RHEL 9.3 中 5.19 版本的 **xfsprogs** 软件包会导致 **/boot** 的大小缩小。因此，与 RHEL 9.2 版本相比，它会导致 **/boot** 分区上可用空间的不同。此修复为所有镜像将 **/boot** 分区增加到 600 MiB，而不是 500 MiB，**/boot** 分区不再受到空间问题的影响。

Jira:RHEL-7999

8.2. 安全性

OpenSSL 命令 cms 和 smime 可以在 FIPS 模式下加密文件

在以前的版本中，**cms** 和 **smime** OpenSSL 命令使用旧的加密算法，如 3DES 或 PKCS #1 v1.5。这些算法在 FIPS 模式下被禁用。因此，使用**smime** 命令用默认设置加密文件无法在 FIPS 模式下的系统上工作。此更新引进了以下更改：

- 在 FIPS 模式下，OpenSSL API 默认使用带有 RSA 密钥的 OAEP 创建 CMS 数据。
- 在 FIPS 模式下，**cms** OpenSSL 命令在提供 RSA 密钥时，创建使用 **aes-128-cbc** 和 OAEP 加密的 CMS 文件。

使用 ECDSA 密钥不受影响。在非 FIPS 模式下，OpenSSL API 和 **cms** 命令默认继续使用 PKCS#1 v1.5 填充和 3DES 加密。

因此，您可以在 FIPS 模式下使用 **cms** 和 **smime** OpenSSL 命令来加密文件。

Bugzilla:2160797

SELinux 允许在 Dovecot 中进行邮件复制

您可以为具有双向复制集的高可用性配置 Dovecot 高性能邮件发送代理，但之前的 SELinux 策略不包含规则，以使 **dovecot-deliver** 工具可在运行时文件系统中通过管道进行通信。因此，Dovecot 中的邮件复制无法正常工作。有了此更新，权限已添加到 SELinux 策略中，因此 Dovecot 中的邮件复制可以正常工作。

Bugzilla:2170495[1]

从 NFS 文件系统引导现在可以在 SELinux 设置为 enforcing 模式的情况下工作

在以前的版本中，当使用 NFS 作为 root 文件系统时，SELinux 标签不会从服务器转发，从而导致 SELinux 设置为 enforcing 模式时引导失败。

有了此更新，SELinux 已被修复，来在初始 SELinux 策略加载前正确地将创建的 NFS 挂载标记为支持安全标签。因此，NFS 挂载现在在服务器和客户端之间转发 SELinux 标签，引导可以在 SELinux 设置为 enforcing 模式的情况下成功。

Bugzilla:2218207[1]

RabbitMQ 使用 IPv6 不再失败

在以前的版本中，当您部署启用了 IPv6 的 **rabbitmq** 服务器时，**inet_gethost** 命令会尝试访问 **/proc/sys/net/ipv6/conf/all/disable_ipv6** 文件。因此，系统拒绝访问 **/proc/sys/net/ipv6/conf/all/disable_ipv6**。有了这个更新，系统现在可以读取 **/proc/sys/net/ipv6/conf/all/disable_ipv6**，**rabbitmq** 现在可以使用 IPv6。

Bugzilla:2184999

SELinux 不再阻止通过 cloud-init 注册到 Insights

在以前的版本中，SELinux 策略不包含允许 **cloud-init** 脚本运行 **insights-client** 服务的规则。因此，尝试通过 **cloud-init** 脚本运行 **insights-client --register** 命令会失败。有了此更新，缺少的规则已被添加到策略中，您可以使用 enforcing 模式的 SELinux，通过 **cloud-init** 注册到 Insights。

Bugzilla:2162663

staff_r SELinux 角色中的用户现在可以运行 scap_workbench 探测

在以前的版本中，**selinux-policy** 软件包不包含用于运行 **scap-workbench** 工具所需的 **staff_r** SELinux 角色中的用户的规则。因此，当用户在 **staff_r** SELinux 角色下运行时，**scap-workbench** 探测会失败。有了此更新，缺少的规则已被添加到 **selinux-policy** 中，SELinux 用户现在可以运行 **scap_workbench** 探测。

Bugzilla:2112729

在 SELinux 策略中添加 insights-client 的权限

insights-client 服务需要不在 **selinux-policy** 之前版本中的权限。因此，**insights-client** 的一些组件无法正常工作，并报告了访问向缓存(AVC)错误消息。在这个版本中，SELinux 策略添加了新权限。因此，**insights-client** 在不报告 AVC 错误的情况下正确运行。

JIRA:RHELPLAN-163014[1], Bugzilla:2190178,Bugzilla:2224737, Bugzilla:2207894, Bugzilla:2214581

Keylime allowlist 生成脚本已更新

Keylime 脚本 **create_allowlist.sh** 为 Keylime 策略生成一个 allowlist。在 RHEL 9.3 中，它被 **create_runtime_policy.sh** 脚本替代，该脚本在尝试将允许列表转换为 JSON 运行时策略时失败。

有了此更新，脚本被恢复为 **create_allowlist.sh**。现在，您可以使用 **keylime_create_policy** 脚本将 allowlist 和 excludelist 合并到 JSON 运行时策略中。

Jira:RHEL-11866[1]

Keylime 不再需要 tls_dir = default 的特定文件

之前，当在 Keylime verifier 或 registrar 配置中将 **tls_dir** 变量设为 **default** 时，Keylime 会拒绝不同于 **cacert.crt** 文件名的自定义证书颁发机构(CA)证书。有了此更新，这个问题不再发生，您甚至可以使用具有 **tls_dir = default** 设置的自定义 CA 证书文件。

Jira:RHELPLAN-157337[1]

环境变量可以覆盖带有下划线的 Keylime 代理选项

在以前的版本中，当 Keylime 代理配置选项名称包含下划线(_)时，通过环境变量覆盖这个选项无法正常工作。有了此更新，即使选项名称包含下划线，通过环境变量的覆盖也可以正常工作。

Jira:RHEL-395[1]

Keylime 注册中心可以正确地识别 IPv6 地址

在以前的版本中，Keylime 注册中心无法正确识别 IPv6 地址，因此无法绑定其监听端口。有了此更新，注册中心可以正确地识别 IPv6 地址，因此可以正确地绑定到其端口。

Jira:RHEL-392[1]

Keylime 代理可以正确地处理 IPv6 地址

在以前的版本中，当使用未包含在括号 [ ] 中的 IPv6 地址注册 Keylime 代理时，**keylime_tenant** 工具失败，并显示错误。有了此更新，**keylime_tenant** 可以正确处理 IPv6 地址，即使它们没有包括在括号中。

Jira:RHEL-393[1]

Keylime 不再会由于 QEMU 虚拟机中的新事件而导致测量引导证明失败

edk2-ovmf 软件包的更新在 QEMU 操作的虚拟系统的测量引导日志中引入了一个新的事件类型。这些事件会在 Keylime 测量的引导证明中导致失败。有了此更新，Keylime 可以正确地处理这些事件。

Jira:RHEL-947[1]

Keylime Webhook 通知程序可以正确地关闭 TLS 会话

在以前的版本中，keylime Webhook 通知程序无法正确关闭 TLS 会话。这会导致在监听器端报告警告。此更新解决了这个问题，webhook 通知程序现在可以正确地关闭 TLS 会话。

Jira:RHEL-1252[1]

gpg-agent 现在在 FIPS 模式下作为 SSH 代理工作

在以前的版本中，当将密钥添加到 **ssh-agent** 程序中时，**gpg-agent** 工具会创建 MD5 指纹，即使 FIPS 模式禁用了 MD5 摘要。因此，**ssh-add** 工具无法将密钥添加到身份验证代理中。

有了这个版本，**gpg-agent** 不再使用 MD5 校验和。因此， **gpg-agent** 现在作为 SSH 身份验证代理可在 FIPS 模式下运行的系统上工作。

Bugzilla:2073567

tangd-keygen 现在可以正确地处理非默认 umask

在以前的版本中，**tangd-keygen** 脚本不会更改生成的密钥文件的文件权限。因此，在具有阻止向其他用户读取密钥的默认用户文件创建模式掩码(**umask**)的系统上，**tang-show-keys** 命令会返回错误消息 **Internal Error 500** 而不是显示密钥。有了这个更新，**tangd-keygen** 为生成的密钥文件设置文件权限，因此脚本现在可以在具有非默认 **umask** 的系统上正常工作。

Bugzilla:2188743

fapolicyd 服务不再运行从可信数据库中删除的程序

在以前的版本中，**fapolicyd** 服务错误地将程序作为可信程序处理，即使它已从可信数据库中删除了。因此，输入 **fapolicyd-cli --update** 命令没有效果，即使被删除后程序也可以执行。有了此更新，**fapolicyd-cli --update** 命令可以正确地更新可信程序数据库，删除的程序无法再执行。

Jira:RHEL-622

fapolicyd 不再导致系统在 mount 和 umount后挂起

在以前的版本中，当 **mount** 或 **umount** 操作运行两次，然后运行 **fapolicyd-cli --update** 命令，**fapolicyd** 服务可能会进入无限循环。因此，系统会停止响应。有了此更新，服务可以正确地运行 **fapolicyd-cli --update** 命令，服务可以处理任意数量的 **mount** 或 **umount** 操作。

Jira:RHEL-817

Keylime 现在接受串联的 PEM 证书

在以前的版本中，当 Keylime 收到作为单个文件中多个 PEM 格式的证书的证书链时，**keylime-agent-rust** Keylime 组件会在生成 TLS 握手时失败。因此，客户端组件(**keylime_verifier** 和 **keylime_tenant**)无法连接到 Keylime 代理。有了此更新，**keylime-agent-rust** 可以正确处理多个包括中间 CA 证书的证书。因此，您现在可以使用与 Keylime 串联的 PEM 证书。

Jira:RHEL-396[1]

即使没有能力，Rsyslog 也可以启动

当 Rsyslog 以普通用户或在容器化环境中执行时，**rsyslog** 进程没有能力。因此，在这种情况下 Rsyslog 无法丢弃能力，并在启动时退出。有了此更新，如果没有能力，进程不再尝试丢弃能力。因此，Rsyslog 可以启动，即使它没有能力。

Jira:RHELPLAN-160541[1]

io_uring 现在可以在没有 SELinux 拒绝的情况下正常工作

在以前的版本中，**io_uring** 内核接口在 SELinux 策略中缺少 **map** 权限。因此，**mmap** 系统调用失败，**io_uring** 接口无法正常工作。有了此更新，SELinux 策略中允许 **map** 权限，接口现在可以正常工作，而没有 SELinux 拒绝。

Bugzilla:2187745

oscap-anaconda-addon 现在为 CIS 强化网络服务器

在以前的版本中，安装带有 CIS 安全配置文件(**cis**、**cis_server_l1**、 **cis_workstation_l1** 或 **cis_workstation_l2**) 的 RHEL 网络服务器不可能带所选的 Network Servers 软件包组。这个问题已通过在 RHEL 9.3 提供的 **oscap-anaconda-addon-2.0.0-17.el9** 中排除 **tftp** 软件包得到了解决。因此，您可以安装带有 Network Servers 软件包组的 CIS 强化的 RHEL Network 服务器。

Bugzilla:2172264

检查主目录的规则只适用于本地用户

scap-security-guide 软件包提供的多个合规性配置文件包含以下检查用户主目录的正确配置的规则：

- **accounts_umask_interactive_users**
- **accounts_user_dot_group_ownership**
- **accounts_user_dot_user_ownership**
- **accounts_user_interactive_home_directory_exists**
- **accounts_users_home_files_groupownership**
- **accounts_users_home_files_ownership**
- **accounts_users_home_files_permissions**
- **file_groupownership_home_directories**
- **file_ownership_home_directories**
- **file_permissions_home_directories**

这些规则正确地检查本地用户的配置。在以前的版本中，扫描程序还会错误地检查网络源（如 NSS）提供的远程用户的配置，即使补救脚本无法更改远程用户的配置。这是因为 OpenSCAP 扫描程序之前使用 **getpwent（）** 系统调用。此更新更改了这些规则的内部实现，使其只依赖于 **/etc/passwd** 文件中的数据。因此，规则现在只适用于本地用户的配置。

Bugzilla:2203791

密码过期规则只适用于本地用户

某些合规性配置文件（如 CIS 和 DISA STIG）包含以下检查用户帐户密码的期限和密码过期的规则：

- **accounts_password_set_max_life_existing**
- **accounts_password_set_min_life_existing**
- **accounts_password_set_warn_age_existing**
- **accounts_set_post_pw_existing**

这些规则正确地检查本地用户的配置。在以前的版本中，扫描程序还会错误地检查网络源（如 NSS）提供的远程用户的配置，即使补救脚本无法更改远程用户的配置。这是因为 OpenSCAP 扫描程序之前使用 **getpwent（）** 系统调用。

此更新更改了这些规则的内部实现，使其只依赖于 **/etc/shadow** 文件中的数据。因此，规则现在只适用于本地用户的配置。

Bugzilla:2213958

Red Hat CVE 源已更新

位于 https://access.redhat.com/security/data/oval/ 的红帽常见漏洞和暴露(CVE)版本 1 已停用，并被 https://access.redhat.com/security/data/oval/v2/ 提供的 CVE 版本 2 替代。

因此，**scap-security-guide** 软件包提供的 SCAP 源数据流中的链接已更新为链接到 Red Hat CVE 源的新版本。

Bugzilla:2223178

与 journald 配置相关的规则不再添加额外的引号

在以前的版本中，SCAP 安全指南规则 **journald_compress**、**journald_forward_to_syslog** 和 **journald_storage** 之前在补救脚本中包含了一个 bug，这导致在 **/etc/systemd/journald.conf** 配置文件中的配置选项中添加了额外的引号。因此，**journald** 系统服务无法解析配置选项，并忽略它们。因此，配置选项无效。这会在 OpenSCAP 扫描中导致错误的**pass** 结果。有了此更新，规则和补救脚本不再添加额外的引号。因此，这些规则现在会为 **journald** 生成一个有效的配置。

Bugzilla:2193169

/var/lib/fdo 下的文件现在得到正确的 SElinux 标签

在以前的版本中，有一个允许 FDO 进程访问整个主机的安全问题。有了这个更新，通过使用带有 SElinux 的 **service-info-api** 服务器，您可以添加任何文件，来发给 **/var/lib/fdo** 目录下的设备，因此 **/var/lib/fdo** 下的文件现在可以得到正确的 SElinux 标签。

Bugzilla:2229722

8.3. 订阅管理

subscription-manager 不再在终端中保留非必要的文本

从 RHEL 9.1 开始，**subscription-manager** 在处理任何操作时会显示进度信息。在以前的版本中，对于某些语言（通常为非拉丁语言），在操作完成后不会清除进度消息。有了这个更新，在操作完成后，所有信息都会被正确清除。

如果您之前禁用了进度信息，您可以输入以下命令重新启用它们：

```
# subscription-manager config --rhsm.progress_messages=1
```

Bugzilla:2136694[1]

8.4. 软件管理

dnf needs-restarting -s 命令现在可以正确地显示 systemd 服务列表

在以前的版本中，当您使用带有 **-s** 或 **--services** 选项的 **needs-restarting** 命令时，当检测到非 systemd 或故障进程时会出现一个错误。有了此更新，**dnf needs-restarting -s** 命令会忽略这样的进程，并显示带有受影响的 systemd 服务列表的警告。

Bugzilla:2203100

dnf-automatic 命令现在正确地报告事务的退出状态

在以前的版本中，**dnf-automatic** 命令返回事务的一个成功的退出代码，即使此事务过程中一些操作没有成功完成。这可能会对使用 **dnf-automatic** 进行自动部署勘误的机器造成安全风险。有了此更新，这个问题已被解决，**dnf-automatic** 现在会在事务中报告软件包的每个问题。

Bugzilla:2212262

在没有扩展文件属性的文件系统上安装带有 IMA 签名的软件包不再失败

在以前的版本中，RPM 尝试将 IMA 签名应用到文件，即使它们不支持这些签名。因此，软件包安装失败。有了此更新，RPM 跳过应用 IMA 签名。因此，软件包安装不再失败。

Bugzilla:2157836

8.5. SHELL 和命令行工具

rsyslog 日志记录服务现在在启动救援系统时启动

在以前的版本中，消息日志记录的 **rsyslog** 服务不会自动在救援系统中启动。**/dev/log** 套接字在恢复过程中保持接收消息，而没有服务侦听此套接字。因此，**/dev/log** 套接字填充了信息，并导致恢复过程卡住。例如，用于重新生成 GRUB 配置的 **grub2-mkconfig** 命令会根据挂载的文件系统数量来生成大量日志消息。如果您使用 ReaR 恢复具有许多挂载的文件系统的系统，则大量日志消息会填充 **/dev/log** 套接字，恢复过程会被冻结。

有了此修复，救援系统中的 **systemd** 单元现在在引导过程中包含套接字目标，以在引导时启动日志记录套接字。因此，如果需要，**rsyslog** 服务会在救援环境中启动，需要在恢复过程中记录日志消息的进程不再被卡住。恢复过程成功完成，您可以在救援 RAM 磁盘中的 **/var/log/messages** 文件中找到日志消息。

Bugzilla:2172912

对于长路径，which 命令不再失败

在以前的版本中，当您在路径超过 256 个字符的目录中执行 **which** 命令时，命令会失败，并显示 **Can't get current working directory** 错误信息。有了此修复，**which** 命令现在对路径长度限制使用 **PATH_MAX** 值。因此，命令不再失败。

Bugzilla:2181974

ReaR 现在支持带有 OUTPUT=USB的 UEFI 安全引导

在以前的版本中，**OUTPUT=USB** ReaR 输出方法（其在可引导磁盘驱动器上存储救援镜像）不遵循 **SECURE_BOOT_BOOTLOADER** 设置。因此，在启用了 UEFI 安全引导的系统上，带有救援镜像的磁盘不会引导，因为引导装载程序没有签名。

有了这个修复，**OUTPUT=USB** ReaR 输出方法现在创建救援磁盘时使用在 **SECURE_BOOT_BOOTLOADER** 设置中指定的引导装载程序。要使用签名的 UEFI shim 引导装载程序，请在 **/etc/rear/local.conf** 文件中更改以下设置：

```
SECURE_BOOT_BOOTLOADER=/boot/efi/EFI/redhat/shimx64.efi
```

因此，当启用 UEFI 安全引导时，救援磁盘是可以启动的。在具有 UEFI 的所有系统上，将变量设置为此值是安全的，即使安全引导未启用。它甚至被推荐用于保持一致性。有关 UEFI 引导流程和 shim 引导装载程序的详情，请参考 UEFI：引导系统时会发生什么。

Bugzilla:2196445

ReaR 恢复的系统不再无法挂载所有 VG 逻辑卷

/etc/lvm/devices/system.devices 文件代表逻辑卷管理器(LVM)系统设备，并控制设备对 LVM 的可见性和可用性。默认情况下，**system.devices** 功能在 RHEL 9 中被启用，当活跃时，它替换 LVM 设备过滤器。

在以前的版本中，当使用 ReaR 将系统恢复到与原始系统使用的不同的硬件 ID 的磁盘时，恢复的系统没有找到所有 LVM 卷，无法引导。有了此更新，如果 ReaR 找到 **system.devices** 文件，ReaR 会在恢复的末尾将此文件移到 **/etc/lvm/devices/system.devices.rearbak** 中。因此，恢复的系统不使用 LVM 设备文件来限制设备的可见性，系统会在引导时找到恢复的卷。

可选：如果要恢复默认行为并重新生成 LVM 设备文件，请在引导恢复的系统并连接正常操作所需的所有磁盘设备后使用 **vgimportdevices -a** 命令，以防在恢复过程之前断开任何磁盘。

Bugzilla:2145014

8.6. 网络

Intel Corporation I350 Gigabit Fiber Network Connection 现在在内核更新后提供一个链接

在以前的版本中，带有 Small Formfactor Pluggable (SFP) transceiver 模块，而没有 External Thermal Sensor (ETS)的硬件配置导致 **igb** 驱动程序错误地初始化 Inter-Integrated Circuit (I2C)，以读取 ETS。因此，连接无法获取链接。有了此 bug 修复，**igb** 驱动程序仅在带有 ETS 的 SFP 可用时初始化 I2C。因此，连接获取了链接。

Bugzilla:2173594[1]

nm-cloud-setup 服务不再从接口中删除手动配置的辅助 IP 地址

根据从云环境收到的信息，**nm-cloud-setup** 服务配置了网络接口。虽然您有为手动接口配置禁用 **nm-cloud-setup** 的选项，但在某些情况下会导致冲突。在某些情况下，主机上的其他服务将独立配置接口，包括添加辅助 IP 地址。当被 **systemd** 计时器单元再次触发时，**nm-cloud-setup** 会错误地删除这些辅助 IP 地址。这个 **NetworkManager** 软件包的更新解决了这个问题。您只需要等待 **systemd** 定时器单元触发 **nm-cloud-setup**。如果您不想等待计时器，您可以使用以下命令手动启用 **nm-cloud-setup** ：

```
# systemctl enable nm-cloud-setup.service
```

因此，**nm-cloud-setup** 不再从接口中删除手动配置的辅助 IP 地址。

Bugzilla:2151040

8.7. 内核

在启用了 VMD 时，RHEL 之前无法识别 NVMe 磁盘

当您重置或重新附加驱动程序时，卷管理设备(VMD)域之前没有软重置。因此，硬件无法正确检测并枚举其设备。有了此更新，启用了 VMD 的操作系统可以正确地识别 NVMe 磁盘，特别是在重置服务器或使用虚拟机时。

Bugzilla:2128610[1]

8.8. 引导加载程序

GRUB 现在可以正确地处理非调试内核变体

在以前的版本中，在安装了多个内核 RPM 的系统中，输入 **dnf install kernel-$VERSION** 或 **dnf update** 命令会将最后一个安装的内核设置为默认内核。例如，这会在 AMD 和 Intel 64 位构架上，或者在 64 位 ARM 架构上的内核(4k)和 **kernel-64k** 上带有标准内核和实时内核的系统上发生。因此，系统可能会在以后的重启时引导到不需要的内核。有了此更新，GRUB 使用 **/etc/sysconfig/kernel** 配置文件中的 **DEFAULTKERNEL** 变量，默认内核保留正确的变体和最新的版本。

如需更多信息，请参阅 更改 Red Hat Enterprise Linux 8 和 9 中默认的内核 解决方案。

Bugzilla:2184069[1]

8.9. 文件系统和存储

lpfc 驱动程序在 D_ID 端口交换过程中处于有效状态

在以前的版本中，在发出 NetApp giveback 操作后 SAN Boot 主机会导致 LVM 挂起任务警告和停滞的 I/O。由于光纤通道 **D_ID** 端口交换，也会在 DM-Multipath 环境中提供备用路径时发生此问题。由于竞争条件，**D_ID** 端口交换会导致 **lpfc** 驱动程序中的状态不一致，这会阻止 I/O 被发出。

有了这个修复，**lpfc** 驱动程序在 **D_ID** 端口交换发生时确保有效状态。因此，光纤频道 **D_ID** 端口交换不会挂起 I/O。

Bugzilla:2173947[1]

multipathd 向所有路径中添加了持久性保留注册密钥

在以前的版本中，当 **multipathd** 守护进程启动时，它识别现有多路径设备的一个路径上持久性保留的注册密钥，并不是该设备的所有路径都有注册密钥。因此，如果在 **multipathd** 停止时，如果有带有持久性保留的多路径设备的新路径，则不会在这些路径上设置持久性保留。这允许路径上的 IO 处理，即使它们应该被保留密钥所禁止。

有了此修复，如果 **multipathd** 在任何设备路径上找到了持久保留注册密钥，它会将密钥添加到所有活动路径中。因此，多路径设备现在都在所有路径上正确设置了持久性保留，即使 **multipathd** 未运行时首先出现了路径设备。

Bugzilla:2164869

LUN 现在在操作系统安装过程中可见

在以前的版本中，系统没有使用固件源的身份验证信息，特别是在涉及带有存储在 iSCSI iBFT (Boot Firmware Table)中的 CHAP (Challenge-Handshake Authentication Protocol)的 iSCSI 硬件卸载的情况。因此，在安装过程中 iSCSI 登录会失败。

有了**udisks2-2.9.4-9.el9** 固件身份验证中的修复，这个问题已被解决，LUN 在安装和初始引导时可见。

Bugzilla:2213769[1]

当在 /etc/fstab中将 NVMe-FC 设备添加为挂载点时，系统可以正确启动

在以前的版本中，由于 **nvme-cli nvmf-autoconnect systemd** 服务中的一个已知问题，在将光纤通道上的 Non-volatile Memory Express (NVMe-FC)设备添加为 **/etc/fstab** 文件中的挂载点时，系统无法引导。因此，系统进入紧急模式。有了此更新，当挂载 NVMe-FC 设备时，系统可以引导，而没有任何问题。

Jira:RHEL-8171[1]

8.10. 高可用性和集群

pcs config checkpoint diff 命令现在可以对所有配置部分正常工作

从 RHEL 9.0 发行版本开始，**pcs config checkpoint diff** 命令已停止显示以下配置部分的不同：隔离级别、排序约束、托管约束、票据约束、资源默认值和操作默认值。从 RHEL 9.1 发行版本开始，**pcs config checkpoint diff** 命令已停止显示资源和 Stonith 设备配置部分的不同。这是因为，作为负责显示每个不同配置部分的代码切换到了加载 CIB 文件的新机制，加载的内容被缓存。用于差异比较第二个文件没有加载，而是使用了第一个文件的缓存内容。因此，**diff** 命令没有产成任何输出。有了此更新，CIB 文件内容不再被缓存，**pcs config checkpoint diff** 命令显示所有配置部分的不同。

Bugzilla:2175881

现在，当配置了隔离级别时，pcsd Web UI 会显示集群状态

在以前的版本中，当配置了隔离级别时，**pcsd** Web UI 不显示集群状态。有了此更新，您可以在配置了隔离级别时查看集群状态，并使用 Web UI 更改集群设置。

Bugzilla:2182810

现在，配置为第二个隔离设备的隔离 watchdog 现在在第一个设备超时时隔离节点

在以前的版本中，当将 watchdog 隔离设备在隔离拓扑中配置为第二个设备时，在计算隔离操作时不会考虑 watchdog 超时。因此，如果第一个设备超时了，隔离操作也会超时，即使 watchdog 会隔离节点。有了此修复，watchdog 超时包含在隔离操作超时中，如果第一个设备超时，则隔离操作成功。

Bugzilla:2182482

当列表按节点分组时，带有规则的位置约束不再显示

不能给带有规则的位置约束分配节点。在以前的版本中，当您按节点分组列表时，带有规则的位置约束会在空节点下显示。有了此修复，带有规则的位置约束不再显示，会给出一个警告信息，表示带有规则的约束没有显示。

Bugzilla:1423473

更新多路径 SCSI 设备的pcs 命令现在可以正常工作

由于 Pacemaker CIB 文件中的变化，按照设计，**pcs stonith update-scsi-devices** 命令会停止工作，从而导致一些集群资源不必要的重启。有了此修复，这个命令可以正常工作，并更新 SCSI 设备，而无需运行在同一节点上的其他集群资源的重启。

Bugzilla:2177996

当 pscd Web UI 打开时，pcsd-ruby 守护进程的内存占用量现在减少了

在以前的版本中，当 **pcsd** Web UI 打开时，**pcsd-ruby** 守护进程的内存占用量在过去几小时稳步增加。有了此修复，在 **pcsd-ruby** 守护进程中运行的 web 服务器现在定期执行安全重启。这会释放分配的内存，并减少内存占用。

Bugzilla:1860626[1]

azure-events-az 资源代理不再产生与 Pacemaker 2.1 及之后版本有关的错误

azure-events-az 资源代理执行 **crm_simulate -Ls** 命令并解析输出。使用 Pacemaker 2.1 及更高版本时，**crm_simulate** 命令的输出不再包含文本 **Transition Summary:**，这会导致错误。有了此修复，当此文本缺失时，代理不再产生错误。

Bugzilla:2182415

mysql 资源代理现在可以与可升级的克隆资源一起工作

在以前的版本中，由于提升分数在提升的和非提升的值之间变化，**mysql** 资源代理会移动在节点间的 Promoted 角色中操作的克隆资源。有了此修复，Promoted 角色中的节点保留在 Promoted 角色中。

Bugzilla:2179003[1]

fence_scsi 代理现在可以自动检测共享的 lvmlockd 设备

在以前的版本中，**fence_scsi** 代理不会自动检测共享的 **lvmlockd** 设备。有了此更新，当 **devices** 属性没有设置时，**fence_scsi** 能够自动检测 **lvmlockd** 设备。

Bugzilla:2187327

8.11. 编译器和开发工具

glibc system（） 函数现在无条件地恢复以前的信号掩码

在以前的版本中，如果 **glibc system（）** 函数从多个线程并行调用，**SIGCHLD** 信号的信号掩码可能无法被正确恢复。因此，在某些线程的 **glibc system（）** 函数返回后，**SIGCHLD** 信号保持阻止状态。

有了这个更新，**glibc system（）** 函数现在无条件恢复以前的信号掩码，即使并行 **system（）** 函数调用正在运行。因此，如果从多个线程同时调用 **glibc system（）** 函数，则 **SIGCHLD** 信号不再被错误地阻止。

Bugzilla:2177235

eu-addr2line -C 现在可以正确地识别其他参数

在以前的版本中，当您使用来自 **elfutils** 的 **eu-addr2line** 命令中的 **-C** 参数时，以下单个字符参数会消失。因此，**eu-addr2line -Ci** 命令的行为与 **eu-addr2line -C** 相同，而 **eu-addr2line -iC** 可以按预期正常工作。这个 bug 已被修复，**eu-addr2line -Ci** 现在识别这两个参数。

Bugzilla:2182059

eu-addr2line -i 现在可以正确地处理 GCC 链接时间优化编译的代码

在以前的版本中，包含在**elfutils** 中的 **libdw** 库的 **dwarf_getscopes**函数无法找到 GCC link-time 优化编译的函数的抽象原始定义。因此，当您在 **eu-addr2line** 命令中使用 **-i** 参数时，**eu-addr2line** 无法显示 **gcc -flto** 编译的代码的内联函数。有了此更新，**libdw dwarf_getscopes** 函数可以在内联范围的正确编译单元中看到，**eu-addr2line -i** 可以按预期正常工作。

Bugzilla:2236182

使用 papi 的程序在关闭时不再停止

在以前的版本中，在 **papi** 初始化一些组件前，**papi** 初始化了线程。因此，描述数组中元素数量的某些组件的条目没有被设置正确的值，并尝试了零大小的内存分配。因此，后续访问和这些零大小内存分配的释放导致程序停止。

这个 bug 已解决，使用 **papi** 的程序在关闭时不再停止。

Bugzilla:2215582

OpenJDK XML 签名提供者现在可以在 FIPS 模式下正常工作

在以前的版本中，OpenJDK XML 签名提供者无法在 FIPS 模式下操作。由于对 FIPS 模式支持的改进，OpenJDK XML 签名提供者现在在 FIPS 模式下启用。

Bugzilla:2186647

8.12. 身份管理

现在，常规用户的分页搜索不会影响性能

在以前的版本中，当目录服务器位于搜索负载下时，常规用户的分页搜索可能会影响服务器性能，因为锁与轮询网络事件的线程发生冲突。另外，如果在发送页搜索时发生网络问题，则整个服务器都没有响应，直到 **nsslapd-iotimeout** 参数过期为止。有了此更新，锁被分成几个部分，以避免与网络事件争用。因此，在常规用户的分页搜索过程中不会影响性能。

Bugzilla:1974242

模式复制现在可以在目录服务器中正常工作

在以前的版本中，当目录服务器将模式复制到新服务器时，它会将所有模式添加到远程副本上的 **99user.ldif** 文件中。这似乎都是自定义模式，因为对所有定义，**X-ORIGIN** 关键字都设为了 **user defined**。因此，可能会导致 web 控制台出现问题，也可能导致对监控架构并期望 **X-ORIGIN** 关键字有特定值的客户出现问题。有了此更新，模式复制可以按预期工作。

Bugzilla:1759941

Referra 模式现在可以在目录服务器中正常工作

在以前的版本中，CLI 将 **nsslapd-referral** 配置属性设置为后端，而不是映射树。因此，referral 模式无法正常工作。有了此更新，**nsslapd-referral** 属性会被正确设置，referral 模式可以按预期工作。

Bugzilla:2053204

LMDB 导入现在可以更快地工作

在以前的版本中，要构建 **entryrdn** 索引，LMDB 导入 worker 线程会等待其他 worker 线程，以确保父条目被处理。这会产生锁竞争，从而大大减慢导入速度。有了此更新，通过 LMDB 数据库的 LDIF 导入已被重新设计，提供者线程会在 worker 线程用来构建 **entryrdn** 索引的临时数据库中存储有关条目 RDN 及其父级的数据。因此，不再需要 worker 线程同步，平均导入率更佳。

请注意，LMDB 导入速率仍然比 BDB 导入慢三倍，因为 LMDB 不支持并发写事务。

Bugzilla:2116948

dirsrv 服务现在在重启后正确启动

在以前的版本中，**dirsrv** 服务在重启后无法启动，因为 **dirsrv** 服务没有明确等待 **systemd-tmpfiles-setup.service** 完成。这会导致竞争条件。有了此更新，**dirsrv** 服务会等待 **systemd-tmpfiles-setup.service** 完成，重启后不再无法启动。

Bugzilla:2179278

更改安全参数现在可以正常工作

在以前的版本中，当使用 **dsconf *instance_name* security set** 命令更改安全参数时，操作会失败并显示错误：

```
Name 'log' is not defined
```

有了此更新，security 参数更改可以正常工作。

Bugzilla:2189717

SSSD 现在在评估基于 GPO 的访问控制时使用 sAMAccountName

在以前的版本中，如果 **ldap_user_name** 在 AD 客户端上被设置为 **sAMAccountName** 以外的值，则基于 GPO 的访问控制失败。有了此更新，在评估基于 GPO 的访问控制时，SSSD 总是使用 **sAMAccountName**。即使 **ldap_user_name** 在 AD 客户端上被设置为与 **sAMAccountName** 不同的值，基于 GPO 的访问控制现在也可以正常工作。

Jira:SSSD-6107

在检索用户时，SSSD 现在可以处理 user_attributes 选项中的重复属性

在以前的版本中，如果 **sssd.conf** 在 **user_attributes** 选项中包含重复属性，则 SSSD 无法正确处理这些重复。因此，具有这些属性的用户无法被检索。有了此更新，SSSD 可以正确地处理重复。因此，具有重复属性的用户现在可以被检索。

Jira:SSSD-6177

现在，动态 Kerberos PAC 票据签名强制机制修复了 IdM 中的跨版本不兼容

在以前的版本中，如果您的身份管理(IdM)部署的服务器同时在 RHEL 9 和 RHEL 8 上运行，则由 Privilege Attribute 证书(PAC)票据签名支持的上游实现导致的不兼容性会导致某些操作失败。有了此更新，RHEL 9 中的动态票据签名强制机制功能的实现修复了这个跨版本不兼容。要使这个功能实际生效，您必须：

1. 更新 域中的所有服务器。
2. 重启所有 IdM Kerberos 分发中心(KDC)服务。

这两个操作的顺序非常重要。启动时，KDC 会查询域中所有其他服务器的元数据，以检查它们是否都支持 PAC 票据签名。否则，签名不会被强制执行。

有关动态 Kerberos PAC 票据签名强制机制的更多信息，包括受限委托请求的示例，请参阅此 知识库文章。

JIRA:RHELDOCS-17011[1], Bugzilla:2182683,Bugzilla:2178298

现在，在 FIPS 模式下允许 SHA-1 签名验证

在以前的版本中，当身份管理在(IdM) FIPS 模式下时，不允许使用 SHA-1 签名验证。这是因为 IdM 使用 FIPS-140-3 标准，这不允许 SHA-1 签名。这种情况导致了与活动目录(AD)的互操作性问题，因为 AD 仅符合旧的 FIPS-140-2 标准，因此需要 SHA-1 签名。

此更新为 PKINIT 签名验证引入了一个 FIPS 异常。当在 IdM 中启用了 FIPS 模式时，其限制被忽略。仅应用默认模式限制，即使在 FIPS 模式下，也允许使用 **SHA1** 加密模块。因此，在 FIPS 模式下， AD 互操作性按预期正常工作。

在 IdM/AD 信任，或使用 RHEL 9.2 或更高版本的主机作为 AD 客户端的情况下，您需要将加密策略设置为 FIPS:AD-SUPPORT:SHA1，来在 FIPS 模式下支持 PKINIT。

Bugzilla:2155607

现在不再允许删除 IdM admin 用户

在以前的版本中，如果您是 **admins** 组的成员，则无法阻止您删除身份管理(IdM) **admin** 用户。缺少 **admin** 用户会导致 IdM 和活动目录(AD)之间的信任停止正常工作。有了此更新，您可以不再删除 **admin** 用户。因此，IdM-AD 信任可以正常工作。

Bugzilla:2229712

ipa-kdb 不再导致 krb5kdc 失败

在以前的版本中，**ipa-kdb** 驱动程序不会区分没有服务器主机对象和连接失败。因此，**krb5kdc** 服务器有时会意外停止，由于与 LDAP 服务器的连接问题产生的 **NULL** LDAP 上下文。

有了此更新，**ipa-kdb** 驱动程序可以正确地识别连接失败，并将它们与没有服务器主机对象进行区分。因此，**krb5kdc** 服务器不会再失败。

Bugzilla:2227831

IdM 客户端安装程序不再在 ldap.conf 文件中指定 TLS CA 配置

在以前的版本中，IdM 客户端安装程序在 **ldap.conf** 文件中指定 TLS CA 配置。有了此更新，OpenSSH 使用默认的信任存储，IdM 客户端安装程序不会在 **ldap.conf** 文件中设置 TLS CA 配置。

Bugzilla:2094673

当可信 AD 用户的名称包含混合问题单字符时，IdM 客户端可以正确地检索它们的信息

在以前的版本中，如果您尝试用户查找或用户的身份验证，并且可信活动目录(AD)用户在其名称中包含混合大小写字符，且在 IdM 中使用覆盖进行了配置，则会返回一个错误，阻止用户访问 IdM 资源。

随着 RHBA-2023:4359 的发布，区分大小写的比较将被忽略字符大小写的不区分大小写的比较替代。因此，IdM 客户端现在可以查找 AD 可信域的用户，即使其用户名包含混合大小写字符，且它们在 IdM 中使用覆盖进行了配置。

JIRA:SSSD-6096

8.13. WEB 控制台

Web 控制台 NBDE 绑定步骤现在也适用于带有根文件系统的卷组

在 RHEL 9.2 中，由于确定用户是否向 root 文件系统添加了 Tang 密钥的代码中的一个 bug，当 LUKS 容器上根本没有文件系统时，web 控制台中的绑定进程会崩溃。因为在点击了 **Verify key** 对话框中的 **Trust key** 按钮后， web 控制台会显示出错信息 **TypeError: Qe (...) is undefined**，所以您必须在上述场景中的命令行界面中执行所有必要的步骤。

有了此更新，web 控制台可以正确地处理在根文件系统中添加 Tang 密钥。因此，web 控制台完成了在不同场景中使用 Network-Bound Disk Encryption (NBDE)自动解锁 LUKS 加密卷所需的所有绑定步骤。

Bugzilla:2203361

VNC 控制台现在可以在大多数分辨率下正常工作

在以前的版本中，当在某些显示分辨率下使用虚拟网络计算(VNC)控制台时，会出现鼠标偏移问题，或者只有接口的一部分可见。因此，无法使用 VNC 控制台。

有了这个更新，这个问题已被解决，VNC 控制台可以在大多数分辨率下正常工作，但非常高的分辨率除外，如 3840x2160。

请注意，记录的和显示的光标位置之间的小偏移可能仍然存在。但是，这不会影响 VNC 控制台的可用性。

Bugzilla:2030836

8.14. RED HAT ENTERPRISE LINUX 系统角色

storage 角色现在可以在不卸载的情况下调整挂载的文件系统大小

在以前的版本中，**storage** 角色无法调整挂载的设备的大小，即使文件系统支持在线调整大小。因此，**storage** 角色会在调整大小前卸载所有文件系统，对于正在使用的文件系统会失败，例如，在调整正在运行的系统的 / 目录的大小时。

有了此更新，**storage** 角色支持调整支持在线调整大小（如 XFS 和 Ext4）的挂载的文件系统的大小。因此，现在可以调整挂载的文件系统的大小，而无需卸载它们。

Bugzilla:2168692

podman_registries_conf 变量现在可以正确地配置 unqualified-search-registries 字段

在以前的版本中，配置了 **podman_registries_conf** 变量后，**podman** RHEL 系统角色失败。因此，在 **/etc/containers/registries.conf.d/50-systemroles.conf** 文件中不会生成 **unqualified-search-registries = ["registry.access.redhat.com"]** 设置。有了这个更新，此问题已被解决。

Bugzilla:2211984

kdump 角色以幂等方式添加 authorized_keys

在以前的版本中，添加 **authorized_key** 的任务每次都添加一个额外的换行符。因此，该角色不是幂等的。有了此修复，添加新的 **authorized_key** 可以正常工作，并智能以幂等方式添加单个密钥值。

Bugzilla:2232241

如果缺少 kdump_authorized_keys，kdump 系统角色不会失败

在以前的版本中，如果 **kdump_ssh_user** 变量中定义的用户无法访问 **主目录中的 .ssh 目录或空的.ssh/authorized_keys** 文件，**kdump** 系统角色将无法添加 **SSH** 授权密钥。有了此修复，**kdump** 系统角色可以正确地将授权密钥添加到 **SSH** 配置中。因此，基于密钥的身份验证可以在上述场景中可靠地工作。

Bugzilla:2232231

在创建前无法从成员磁盘中删除数据的问题不再存在

在以前的版本中，当创建 RAID 卷时，在组成 RAID 卷前，系统不能有效地从成员磁盘中删除现有的数据。有了此更新，RAID 卷可以根据需要从成员磁盘中删除任何已存在的数据。

Bugzilla:2224090

在有不存在服务的检查模式下运行 firewall RHEL 系统角色不再失败

在以前的版本中，在检查模式下使用不存在的服务运行 **firewall** 角色会失败。此修复更好地实现了遵守检查模式的 Ansible 最佳实践。因此，启用或禁用不存在的服务不再在检查模式中使角色失败。相反，会有一个警告提示您确认服务是否已在前面的 playbook 中定义。

Bugzilla:2222428

RHEL 7 上的 firewall RHEL 系统角色不再尝试安装不存在的 Python 软件包

在以前的版本中，当 RHEL 7 上的 **firewall** 角色从另一个角色调用，且该角色在使用 **python3** 时，**firewall** 角色会尝试安装那个 Python 版本的 **python3-firewall** 库。但是，在 RHEL 7 上不提供该库。因此，**python3-firewall** 库没有找到，您会收到以下出错信息：

```
No package matching 'python3-firewall' found available, installed or updated
```

有了此更新，**firewall** 角色不会尝试安装 **python-firewall** 或 **python3-firewall** 库。因此，当**python3** 已在受管节点上安装了，**firewall** 角色不会在 RHEL 7 上失败。

Bugzilla:2216520

kdump RHEL 系统角色更新

kdump RHEL 系统角色已更新至更新的版本，其带来以下值得注意的改进：

- 安装 **kexec-tools** 后，工具套件不再产生 **/etc/sysconfig/kdump** 文件，因为您不再需要管理此文件。

- 角色支持 **auto_reset_crashkernel** 和 **dracut_args** 变量。

如需了解更多详细信息，请参阅 **/usr/share/doc/rhel-system-roles/kdump/** 目录中的资源。

Bugzilla:2211187

使用 rhc 角色创建的 Insights 标签现在可以正确应用

在以前的版本中，当使用 **rhc** 角色创建 Insights 标签时，标签没有存储在正确的文件中。因此，标签没有发送给 Insights，因此它们没有应用到 Insights 清单中的系统。

有了此修复，标签会被正确存储，并应用到 Insights 清单中存在的系统。

Bugzilla:2209200

raid_chunk_size 参数不再返回一个错误信息

在以前的版本中，RAID 池和卷不允许用于 **raid_chunk_size** 属性。有了此更新，您现在可以为 RAID 池和卷配置 **raid_chunk_size** 属性，而不会遇到任何限制。

Bugzilla:2193058

certificate RHEL 系统角色现在在确定是否执行新证书请求时检查证书密钥大小

在以前的版本中，在评估是否请求新证书时，**certificate** RHEL 系统角色不会检查证书的密钥大小。因此，角色有时在应该发布新证书请求时没有发布新证书请求。有了此更新，**certificate** 现在检查 **key_size** 参数，以确定是否应执行一个新证书请求。

Bugzilla:2186057

kdump 角色以等幂方式向 authorized_keys 添加多个密钥

在以前的版本中，同时向 **authorized_keys** 文件中添加多个 SSH 密钥会将一个主机的密钥值替换为另一个主机的密钥值。此更新通过使用 **lineinfile** 模块来管理 **authorized_keys** 文件解决了这个问题。**lineinfile** 按顺序迭代任务，检查现有的密钥，并一次在一台主机上的一个原子操作中写入新密钥。因此，在多个主机上添加 SSH 密钥可以正常工作，不会替换另一个主机的密钥值。

注意：在 play 级别使用 **serial: 1** play serial 关键字来控制一次执行的主机的数量。

Jira:RHEL-1499[1]

kdump 角色成功为 kdump_ssh_server 身份验证更新了 .ssh/authorized_keys

在以前的版本中，**kdump** 角色无法访问 **.ssh** 目录，来安全地验证用户，以登录到 **kdump_ssh_server**。因此，**kdump** 角色没有更新 **.ssh/authorized_keys** 文件和 SSH 机制，来验证**kdump_ssh_server**是否失败。在这个版本中解决了这个问题。因此，**kdump_ssh_server** 上的 **kdump_ssh_user** 身份验证可以可靠地工作。

Jira:RHEL-1397[1]

为系统角色启用 kdump 需要在 RHEL 9 及更新的版本上使用 failure_action 配置参数

在以前的版本中，在 **kdump** 配置过程中使用**default**选项不成功，并在日志中打印以下警告：

```
kdump: warning: option 'default' was renamed 'failure_action' and will be removed in the future.
please update /etc/kdump.conf to use option 'failure_action' instead.
```

因此，如果使用了 **default** 选项，角色无法成功启用 **kdump**。此更新解决了这个问题，您可以使用 **failure_action** 参数在多个系统上配置内核转储参数。因此，在上述场景中启用 **kdump** 可以正常工作。

Jira:RHEL-906[1]

firewall 系统角色的 previous: replaced 参数现在覆盖以前的配置，而不删除它

在以前的版本中，如果您将 **previous: replaced** 参数添加到变量列表中，**firewall** 系统角色会删除所有现有用户定义的设置，并将 **firewalld** 重置为默认设置。此修复使用 **firewalld** 中的回退配置（其在 EL7 版本中引入）来保留前面的配置。因此，当您在变量列表中使用 **previous: replaced** 参数时，在重置时不会删除 **firewall.conf** 配置文件，但文件中的文件和注释会保留。

Jira:RHEL-1495[1]

在检查模式下使用 previous: replaced 时，firewall RHEL 系统角色可以正确地报告变化

在以前的版本中，当在检查模式下使用 **previous: replaced** 参数时，**firewall** 角色不会检查是否有文件更改了。因此，角色会给出一个关于未定义变量的错误。此修复向检查模式添加了新的检查变量，以评估是否有任何文件将被 **previous: replaced** 参数更改。检查 **firewalld.conf** 文件会评估 **rpm** 数据库，以确定文件是否已被软件包中提供的版本更改了。因此，在使用 **previous: replaced** 参数时，**firewall** 角色现在可以正确地报告更改。

Jira:RHEL-898[1]

在将区域分配给网络管理器接口时，firewall RHEL 系统角色可以正确地报告更改

在以前的版本中，当不存在更改时，网络管理器接口分配会报告更改。有了此修复，文件 **library/firewall_lib.py** 中的 **try_set_zone_of_interface** 模块会返回第二个值，它表示接口的区域是否被更改了。因此，当向 Network Manager 处理的接口分配区域时，模块现在可以正确地报告更改。

Jira:RHEL-885[1]

当 rhc_auth 包含激活码时，rhc 系统角色不会在注册的系统上失败

在以前的版本中，当使用 **rhc_auth** 参数中指定的激活码在注册的系统上执行 playbook 文件时，会出现失败。这个问题已解决。现在，可以在已经注册的系统上执行 playbook 文件，即使在 **rhc_auth** 参数中提供了激活码。

Bugzilla:2186218

8.15. 虚拟化

虚拟机关闭后，NVIDIA 图形设备继续工作

在以前的版本中，在 RHEL 内核中，设备电源转换延迟与 PCIe 规格要求的延迟更加一致。因此，在附加的虚拟机关闭后，一些 NVIDIA GPU 可能在用于设备分配时变得没有响应。这个更新扩展了 NVIDIA 音频设备功能的设备电源转换延迟。因此，在此场景下，Nvidia GPU 可以继续正常工作。

Bugzilla:2178956[1]

现在，故障转移 virtio NIC 在 Windows 虚拟机上被正确分配了 IP 地址

在以前的版本中，当启动带有故障转移 virtio NIC Windows 虚拟机(VM)时，虚拟机无法为 NIC 分配 IP 地址。因此，NIC 无法建立网络连接。这个问题已被解决，VM NIC 现在可以在上述场景中按预期建立网络连接。

Bugzilla:1969724

安装程序显示要在虚拟机上安装 RHEL 的预期的系统磁盘

在以前的版本中，当使用 **virtio-scsi** 设备在虚拟机上安装 RHEL 时，这些设备可能会因为 **device-mapper-multipath** bug 而不在安装程序中出现。因此，在安装过程中，如果某些设备设置了串口，而有些设备没有，则 **multipath** 命令会声明所有具有串口的设备。因此，安装程序无法在虚拟机中找到要安装 RHEL 的预期的系统磁盘。

有了这个更新，**multipath** 可以正确地将没有串口的设备设置为没有全局识别符(WWID)，并忽略它们。在安装时，**multipath** 只声明 **multipathd** 用来绑定多路径设备的设备，安装程序会在虚拟机中显示要安装 RHEL 的预期的系统磁盘。

Bugzilla:1926147[1]

实时迁移后，Broadcom 网络适配器现在可以在 Windows 虚拟机上正常工作

在以前的版本中，Broadcom 系列设备的网络适配器（如 Broadcom、Qlogic 或 Marvell）无法在 Windows 虚拟机实时迁移过程中进行热拔。因此，迁移完成后，适配器不能正常工作。此问题只会影响使用单根 I/O 虚拟化(SR-IOV)附加到 Windows 虚拟机的适配器。有了此更新，底层代码已被修复，这个问题不再发生。

Jira:RHEL-910, Bugzilla:2091528, Bugzilla:2111319

nodedev-dumpxml 可以正确列出某些介质设备的属性

在此更新前，**nodedev-dumpxml** 工具无法正确列出使用 **nodedev-create** 命令创建的介质设备的属性。这个问题已被解决，**nodedev-dumpxml** 现在可以正确地显示受影响的介质设备的属性。

Bugzilla:2143158

重启 virtqemud 或 libvirtd后，无法附加 virtiofs 设备

在以前的版本中，重启 **virtqemud** 或 **libvirtd** 服务会阻止 **virtiofs** 存储设备附加到主机上的虚拟机(VM)。这个 bug 已被解决，您现在可以在上述场景中附加 **virtiofs** 设备。

Bugzilla:2078693

向虚拟机热插一块 Watchdog 卡不再失败

在以前的版本中，如果没有 PCI 插槽可用，向正在运行的虚拟机(VM)添加一块 Watchdog 卡会失败，并显示以下错误：

```
Failed to configure watchdog
ERROR Error attempting device hotplug: internal error: No more available PCI slots
```

有了此更新，这个问题已被解决，向正在运行的虚拟机中添加一块 Watchdog 卡现在可以按预期正常工作。

Bugzilla:2173584

对于 IBM Z 上的 virtio-gpu，blob 资源无法正常工作

virtio-gpu 设备目前与 IBM Z 系统上的 **blob** 内存资源不兼容。因此，如果您在 IBM Z 主机上配置带有 **virtio-gpu** 的虚拟机(VM)，以使用 **blob** 资源，则虚拟机没有任何图形输出。

Jira:RHEL-7135

第 9 章 技术预览

这部分列出了 Red Hat Enterprise Linux 9 中的所有技术预览。

如需有关红帽对技术预览功能支持范围的信息，请参阅 技术预览功能支持范围。

9.1. 安装程序和镜像创建

光纤通道设备上的 NVMe 现在在 RHEL 安装程序中作为一个技术预览提供

现在，您可以将光纤通道设备上的 NVMe 作为技术预览添加到 RHEL 安装中。在 RHEL 安装程序中，您可以在 Installation Destination 屏幕中添加磁盘时，在 NVMe Fabrics Devices 部分中选择这些设备。

Bugzilla:2107346

9.2. 安全性

gnutls 现在使用 kTLS 作为技术预览

更新的 **gnutls** 软件包可以将内核 TLS (kTLS)作为技术预览，来在加密通道上加速数据传输。要启用 kTLS，请使用 **modprobe** 命令添加 **tls.ko** 内核模块，并使用以下内容为系统范围的加密策略创建一个新的配置文件 **/etc/crypto-policies/local.d/gnutls-ktls.txt** ：

```
[global]
ktls = true
```

请注意，当前版本不支持通过 TLS **KeyUpdate** 消息更新流量密钥，这会影响 AES-GCM passwordsuites 的安全性。如需更多信息，请参阅 RFC 7841 - TLS 1.3 文档。

Bugzilla:2108532[1]

9.3. SHELL 和命令行工具

GIMP 在 RHEL 9 中作为技术预览提供

GNU Image Manipulation Program(GIMP)2.99.8 现在作为技术预览在 RHEL 9 中提供。**gimp** 软件包版本 2.99.8 是一个预发行版本，它有一组改进，但只能保证稳定性。发布官方 GIMP 3 后，将作为此预发布版本的更新，在 RHEL 9 中引入。

在 RHEL 9 中，您可以作为 RPM 软件包轻松安装 **gimp**。

Bugzilla:2047161[1]

9.4. 基础架构服务

TuneD 的套接字 API 作为技术预览提供

通过 UNIX 域套接字控制 TuneD 的套接字 API 现在作为技术预览提供。套接字 API 将一对一与 D-Bus API 映射，并为 D-Bus 不可用的情况提供替代通信方法。通过使用套接字 API，您可以控制 TuneD 守护进程来优化性能，并更改各种调优参数的值。套接字 API 默认被禁用，您可以在 **tuned-main.conf** 文件中启用它。

Bugzilla:2113900

9.5. 网络

WireGuard VPN 作为技术预览提供

WireGuard（红帽作为技术预览提供）是一个在 Linux 内核中运行的高性能 VPN 解决方案。它使用现代加密，比其他 VPN 解决方案更容易配置。此外，因为 WireGuard 较小的代码基础，减少了受攻击的风险，因此提高了安全性。

详情请查看设置 WireGuard VPN。

Bugzilla:1613522[1]

KTLS 作为技术预览提供

RHEL 将内核传输层安全(KTLS)作为技术预览提供。kTLS 使用内核中的对称加密或解密算法为 AES-GCM 密码处理 TLS 记录。kTLS 还包括用来将 TLS 记录加密卸载到提供此功能的网络接口控制器(NIC)的接口。

Bugzilla:1570255[1]

systemd-resolved 服务作为技术预览提供

systemd-resolved 为本地应用程序提供名字解析。该服务实现了缓存和验证 DNS stub 解析器、链接本地多播名称解析(LLMNR)和多播 DNS 解析器和响应程序。

请注意, **systemd-resolved** 是一个不受支持的技术预览。

Bugzilla:2020529

PRP 和 HSR 协议现在作为技术预览提供

这个更新添加了提供以下协议的 **hsr** 内核模块：

- Parallel Redundancy Protocol (PRP)
- High-availability Seamless Redundancy(HSR)

IEC 62439-3 标准定义了这些协议，您可以使用此功能在以太网网络中配置零损失冗余。

Bugzilla:2177256[1]

将 IPsec 封装卸载到 NIC 现在作为技术预览提供

此更新向内核添加了 IPsec 数据包卸载功能。在以前的版本中，只能将加密卸载到网络接口控制器(NIC)。有了此增强，内核现在可将整个 IPsec 封装过程卸载到 NIC，以减少工作负载。

请注意，将 IPsec 封装过程卸载到 NIC 也会减少内核监控和过滤此类数据包的能力。

Bugzilla:2178699[1]

RHEL 中 modems 的网络驱动程序作为技术预览提供

设备制造商支持将联邦通信委托(FCC)锁定作为默认设置。FCC 提供了一个锁，来将 WWAN 驱动程序绑定到特定的系统，其中 WWAN 驱动程序提供了一个与调制解调器进行通信的通道。根据调制解调器 PCI ID，制造商在 Red Hat Enterprise Linux 上为 ModemManager 集成了解锁工具。但是，如果之前未解锁，调制解调器仍不可用，即使 WWAN 驱动程序兼容并可以正常工作。Red Hat Enterprise Linux 为以下带有有限功能的调制解调器提供了驱动程序，来作为技术预览：

- Qualcomm MHI WWAM MBIM - Telit FN990Axx
- Intel IPC over Shared Memory (IOSM)- Intel XMM 7360 LTE Advanced
- MediaTek t7xx (WWAN)- Fibocom FM350GL
- Intel IPC over Shared Memory (IOSM)- Fibocom L860GL modem

JIRA:RHELDOCS-16760[1], Bugzilla:2123542, JIRA:RHEL-6564, Bugzilla:2110561, Bugzilla:2222914

IPv6 上的段路由(SRv6)作为技术预览提供

RHEL 内核将 IPv6 (SRv6)上的段路由作为技术预览提供。您可以使用此功能来优化边缘计算中的流量流，或提高数据中心中的网络可编程性。但是，最重要的用例是在 5G 部署场景中的端到端(E2E)网络分片。在这个区域中，SRv6 协议为您提供可编程自定义网络分片和资源保留，以解决特定应用程序或服务的网络要求。同时，解决方案可以部署到单一用途设备上，其满足较小计算占用的需求。

Bugzilla:2186375[1]

KTLS rebase 到版本 6.3

内核传输层安全(KTLS)功能是一个技术预览。有了此 RHEL 版本，kTLS 已 rebase 到 6.3 上游版本，重要的更改包括：

- 添加了对带有 TX 设备卸载的 256 位密钥的支持
- 提供各种 bug 修复

Bugzilla:2183538[1]

9.6. 内核

带有统一内核镜像的 kdump 机制作为技术预览提供

带有包含统一内核镜像(UKI)中内核镜像的 **kdump** 机制作为技术预览提供。UKI 是一个可执行文件，将 **initramfs**、**vmlinuz** 和内核命令行合并到一个文件中。UKI 密钥的好处是一次将 SecureBoot 的加密签名扩展到所有组件。

要使功能正常工作，使用 UKI 中包含的内核命令行，使用合适的值设置 **crashkernel=** 参数。这会为 **kdump** 保留所需的内存。

注：目前，Linux 内核的 **kexec_file_load** 系统调用无法加载 UKI。因此，当使用 **kexec_file_load** 系统调用加载崩溃内核时，只使用 UKI 中包含的内核镜像。

Bugzilla:2169720[1]

SGX 作为技术预览

软件扩展（SGX）是一个 Intel® 技术，用于保护软件代码和数据不受公开和修改的影响。RHEL 内核部分提供 SGX v1 和 v1.5 功能。版本 1 使用 **Flexible Launch Control** 机制启用平台，以使用 SGX 技术。版本 2 添加了 **Enclave Dynamic Memory Management**(EDMM)。主要特性包括：

- 修改属于初始化 enclave 的常规 enclave 页的 EPCM 权限。
- 动态将常规 enclave 页添加到初始化的 enclave。
- 扩展初始化的 enclave ，以容纳更多线程。

- 从初始化的 enclave 中删除常规的 enclave 页和 TCS 页。

Bugzilla:1874182[1]

用于内核的 Intel 数据流加速器驱动程序作为技术预览提供

内核的 Intel 数据流加速器驱动程序(IDXD)目前作为技术预览提供。它是一个 Intel CPU 集成的加速器，包括共享工作队列 ID(pasid)提交和共享虚拟内存(SVM)。

Bugzilla:2030412

Soft-iWARP 驱动程序作为技术预览提供

软硬件硬件(siw)是一种软件，互联网是 RDMA 协议(iWARP)，适用于 Linux 的内核驱动程序。soft-iWARP 通过 TCP/IP 网络堆栈实施 iWARP 协议套件。这个协议套件在软件中完全实现，不需要特定的远程直接内存访问(RDMA)硬件。Soft-iWARP 使具有标准以太网适配器的系统连接到 iWARP 适配器或安装了 Soft-iWARP 的其他系统。

Bugzilla:2023416[1]

SGX 作为技术预览

软件扩展（SGX）是一个 Intel® 技术，用于保护软件代码和数据不受公开和修改的影响。RHEL 内核部分提供 SGX v1 和 v1.5 功能。版本 1 使用 **Flexible Launch Control** 机制启用平台，以使用 SGX 技术。版本 2 添加了 **Enclave Dynamic Memory Management**(EDMM)。主要特性包括：

- 修改属于初始化 enclave 的常规 enclave 页的 EPCM 权限。
- 动态将常规 enclave 页添加到初始化的 enclave。
- 扩展初始化的 enclave，以容纳更多线程。
- 从初始化的 enclave 中删除常规的 enclave 页和 TCS 页。

Bugzilla:1660337[1]

rvu_af,rvu_nicpf 和 rvu_nicvf 作为技术预览提供

对于 Marvell OCTEON TX2 Infrastructure Processor 系列，以下内核模块作为技术预览提供：

- **rvu_nicpf** - Marvell OcteonTX2 NIC 物理功能驱动程序
- **rvu_nicvf** - Marvell OcteonTX2 NIC 虚拟功能驱动程序
- **rvu_nicvf** - Marvell OcteonTX2 RVU Admin 功能驱动程序

Bugzilla:2040643[1]

9.7. 文件系统和存储

DAX 现在作为技术预览供 ext4 和 XFS 使用

在 RHEL 9 中，DAX 文件系统作为技术预览提供。DAX 提供了将持久内存直接映射到其地址空间的方法。要使用 DAX，系统必须有某种可用的持久性内存，通常使用一个或多个非线性内存模块(NVDIMM)，必须在 NVDIMM 上创建 DAX 兼容文件系统。另外，该文件系统必须使用 **dax** 挂载选项挂载。然后，在 dax 挂载的文件系统中的一个文件 **mmap** 会导致存储直接映射到应用程序的地址空间中。

Bugzilla:1995338[1]

NVMe-oF Discovery Service 功能作为技术预览

NVMe-oF Discovery Service 功能（在 NVMexpress.org 技术 Proposals(TP)8013 和 8014 中）作为技术预览提供。要预览这些功能，请使用 **nvme-cli 2.0** 软件包，并将主机附加到实现 TP-8013 或 TP-8014 的 NVMe-oF 目标设备。有关 TP-8013 和 TP-8014 的更多信息，请参阅 https://nvmexpress.org/specifications/ 网站中的 NVM Express 2.0 Ratified TPs。

Bugzilla:2021672[1]

NVMe-stas 软件包作为技术预览

nvme-stas 软件包，它是 Linux 的中央 Discovery Controller (CDC) 客户端，现在作为技术预览提供。它处理异步事件通知 (AEN)、自动化的 NVMe 子系统连接控制、错误处理和报告以及自动 (**zeroconf**) 和手动配置。

这个软件包由两个守护进程组成，分别是 Storage Appliance Finder (**stafd**) 和存储设备连接器 (**stacd**)。

Bugzilla:1893841[1]

NVMe TP 8006 in-band 身份验证作为技术预览提供

实现 Non-Volatile Memory Express (NVMe) TP 8006，它是一种针对 NVMe over Fabrics (NVMe-oF) 的带内验证，现在作为不支持的技术预览提供。NVMe Technical Proposal 8006 为 NVMe-oF 定义了 **DH-HMAC-CHAP** 带内验证协议，该协议由这个增强提供。

如需更多信息，请参阅 **nvme-connect (1)** 手册页中的 **dhchap-secret** 和 **dhchap-ctrl-secret** 选项描述。

Bugzilla:2027304[1]

io_uring 接口作为技术预览提供

io_uring 是一个新的有效的异步 I/O 接口，现在作为技术预览提供。默认情况下禁用此功能。您可以通过将 **kernel.io_uring_disabled** sysctl 变量设置为以下值之一来启用这个接口：

0

所有进程都可以正常创建 **io_uring** 实例。

1

对非特权进程，**io_uring** 创建被禁用。**io_uring_setup** 失败并显示 **-EPERM** 错误，除非调用过程具有 **CAP_SYS_ADMIN** 功能的特权。仍可使用现有的 **io_uring** 实例。

2

对所有进程，**io_uring** 创建被禁用。**io_uring_setup** 使用 **-EPERM** 总是失败。仍可使用现有的 **io_uring** 实例。这是默认设置。

使用此功能也需要 SELinux 策略的更新版本，来在匿名内节点上启用 **mmap** 系统调用。

通过使用 **io_uring** 命令直通，应用程序可以直接向底层硬件发出命令，如 **nvme**。使用 **io_uring** 命令直通目前需要自定义 SELinux 策略模块。创建一个自定义 SELinux 策略模块：

1. 将以下行保存为 **io_uring_cmd_passthrough.cil** 文件：

```
---cut here---
( allow unconfined_domain_type device_node ( io_uring ( cmd )))
( allow unconfined_domain_type file_type ( io_uring ( cmd )))
```

```
---cut here---
```

2. 加载策略模块：

```
# semodule -i io_uring_cmd_passthrough.cil
```

Bugzilla:2068237[1]

9.8. 编译器和开发工具

jmc-core 和 owasp-java-encoder 作为技术预览

RHEL 9 与 **jmc-core** 和 **owasp-java-encoder** 软件包一起分发，作为 AMD 和 Intel 64 位架构的技术预览功能提供。

jmc-core 是一个为 Java Development Kit (JDK) Mission Control 提供核心 API 的库，包括用于解析和编写 JDK Flight Recording 文件的库，以及用于通过 Java 发现协议(JDP)的 Java 虚拟机(JVM)发现的库。

owasp-java-encoder 软件包提供了 Java 的高性能低后台上下文组。

请注意，自 RHEL 9.2 开始，**jmc-core** 和 **owasp-java-encoder** 在 CodeReady Linux Builder (CRB)存储库中提供，您必须明确启用。如需更多信息，请参阅 如何在 CodeReady Linux Builder 中启用和使用内容。

Bugzilla:1980981

9.9. 身份管理

DNSSEC 在 IdM 中作为技术预览提供

带有集成 DNS 的身份管理(IdM)服务器现在实现了 DNS 安全扩展(DNSSEC)，这是一组增强 DNS 协议安全的 DNS 扩展。托管在 IdM 服务器上的 DNS 区可以使用 DNSSEC 自动签名。加密密钥是自动生成和轮转的。

建议那些决定使用 DNSSEC 保护 DNS 区的用户读取并遵循这些文档：

- DNSSEC 操作实践, 版本 2
- 安全域名系统(DNS)部署指南
- DNSSEC 键翻滚时间注意事项

请注意，集成了 DNSSEC 的 IdM 服务器验证从其他 DNS 服务器获取的 DNS 答案。这可能会影响未按照推荐的命名方法配置的 DNS 区域可用性。

Bugzilla:2084180

身份管理 JSON-RPC API 作为技术预览

一个 API 可用于 Identity Management(IdM)。要查看 API，IdM 还提供了一个 API 浏览器作为技术预览。

在以前的版本中，IdM API 被改进来启用多个 API 命令版本。这些增强可能会以不兼容的方式改变命令的行为。用户现在可以继续使用已有的工具和脚本，即使 IdM API 发生了变化。这可启用：

- 管理员要在服务器中使用之前或更高版本的 IdM，而不是在管理客户端中使用。

- 开发人员可以使用 IdM 调用的特定版本，即使 IdM 版本在服务器上发生了变化。

在所有情况下，与服务器进行通信是可能的，无论是否一方使用，例如，一个新的版本会为这个功能引进新的选项。

有关使用 API 的详细信息，请参阅使用身份管理 API 与 IdM 服务器通信(TECHNOLOGY PREVIEW)。

Bugzilla:2084166

sssd-idp 子软件包作为技术预览提供

SSSD 的 **sssd-idp** 子软件包包含 **oidc_child** 和 krb5 **idp** 插件，它们是对身份管理(IdM)服务器执行 OAuth2 身份验证的客户端组件。此功能仅适用于 RHEL 9.1 及更高版本上的 IdM 服务器。

Bugzilla:2065693

SSSD 内部 krb5 idp 插件作为技术预览提供

SSSD krb5 **idp** 插件允许您使用 OAuth2 协议对外部身份提供者(IdP)进行身份验证。此功能仅适用于 RHEL 9.1 及更高版本上的 IdM 服务器。

Bugzilla:2056482

RHEL IdM 允许将用户身份验证委派给外部身份提供程序作为技术预览

在 RHEL IdM 中，您可以把用户与支持 OAuth 2 设备授权流的外部身份提供程序(IdP)关联。当这些用户使用 RHEL 9.1 或更高版本中的 SSSD 版本进行身份验证时，它们会在执行身份验证和在外部 IdP 授权后接收到带有 Kerberos 票据的 RHEL IdM 单点登录功能。

主要特性包括：

- 使用 **ipa idp-*** 命令为外部 IdP 添加、修改和删除引用
- 使用 **ipa user-mod --user-auth-type=idp** 命令为用户启用 IdP 验证

如需更多信息，请参阅使用外部身份提供程序向 IdM 进行身份验证。

Bugzilla:2069202

ACME 作为技术预览提供

自动证书管理环境(ACME)服务现在作为技术预览在 Identity Management(IdM)中提供。ACME 是一个用于自动标识符验证和证书颁发的协议。它的目标是通过缩短证书生命周期并避免证书生命周期管理中的手动过程来提高安全性。

在 RHEL 中，ACME 服务使用红帽认证系统(RHCS)PKI ACME 响应程序。RHCS ACME 子系统自动部署到 IdM 部署中的每个证书颁发机构(CA)服务器上，但只有管理员启用它之后，它才会为请求提供服务。RHCS 在发布 ACME 证书时使用 **acmeIPAServerCert** 配置文件。签发的证书的有效期为 90 天。启用或禁用 ACME 服务会影响整个 IdM 部署。

重要

建议仅在所有服务器都运行 RHEL 8.4 或以上版本的 IdM 部署中启用 ACME。早期的 RHEL 版本不包括 ACME 服务，这可能会在混合版本部署中引起问题。例如，没有 ACME 的 CA 服务器可能会导致客户端连接失败，因为它使用不同的 DNS Subject Alternative Name(SAN)。

警告

目前，RHCS 不会删除过期的证书。由于 ACME 证书在 90 天后过期，因此过期的证书可能会累积，这会影响性能。

- 要在整个 IdM 部署中启用 ACME，请使用 **ipa-acme-manage enable** 命令：

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

- 要在整个 IdM 部署中禁用 ACME，请使用 **ipa-acme-manage disable** 命令：

```
# ipa-acme-manage disable
The ipa-acme-manage command was successful
```

- 要检查是否安装了 ACME 服务，以及它是否启用或禁用了，请使用 **ipa-acme-manage status** 命令：

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

Bugzilla:2084181[1]

9.10. 桌面

GNOME 用于 64 位 ARM 架构，作为一个技术预览

GNOME 桌面环境可用于 64 位 ARM 架构，作为技术预览。

现在，您可以使用 VNC 连接到 64 位 ARM 服务器上的桌面会话。因此，您可以使用图形应用程序管理服务器。

64 位 ARM 提供了有限的图形应用程序集合。例如：

- Firefox Web 浏览器
- Red Hat 订阅管理器 (**subscription-manager-cockpit**)
- 防火墙配置(**firewall-config**)
- 磁盘用量分析器(**baobab**)

使用 Firefox，您可以连接到服务器上的 Cockpit 服务。

某些应用程序，如 LibreOffice，只提供命令行界面，其图形界面被禁用。

Jira:RHELPLAN-27394[1]

用于 IBM Z 架构的 GNOME 作为技术预览提供

对于 IBM Z 架构，GNOME 桌面环境作为技术预览。

现在，您可以使用 VNC 连接到 IBM Z 服务器上的桌面会话。因此，您可以使用图形应用程序管理服务器。

IBM Z 上提供了一组有限的图形应用程序。例如：

- Firefox Web 浏览器
- Red Hat 订阅管理器 (**subscription-manager-cockpit**)
- 防火墙配置(**firewall-config**)
- 磁盘用量分析器(**baobab**)

使用 Firefox，您可以连接到服务器上的 Cockpit 服务。

某些应用程序，如 LibreOffice ，只提供命令行界面，其图形界面被禁用。

Jira:RHELPLAN-27737[1]

9.11. 虚拟化

创建嵌套虚拟机

对于运行在 Intel、AMD64 和 IBM Z 主机上的 RHEL 9 KVM 虚拟机，嵌套的 KVM 虚拟化作为技术预览提供。有了这个功能，运行在物理 RHEL 9 主机上的 RHEL 7、RHEL 8 或 RHEL 9 虚拟机可以充当 hypervisor，并托管自己的虚拟机。

Jira:RHELDOCS-17040[1]

用于 KVM 虚拟机的 AMD SEV 和 SEV-ES

作为技术预览，RHEL 9 为使用 KVM 管理程序的 AMD EPYC 主机提供安全加密虚拟化(SEV)功能。如果在虚拟机(VM)上启用，SEV 会加密虚拟机的内存来保护虚拟机被主机访问。这提高了虚拟机的安全性。

另外，增强的 Encrypted State 版本 SEV-ES）也作为技术预览提供。SEV-ES 在虚拟机停止运行时加密所有 CPU 注册内容。这可防止主机修改虚拟机的 CPU 注册或读取它们中的任何信息。

请注意，SEV 和 SEV-ES 仅适用于第 2 代 AMD EPYC CPU（代号 Rome）或更新版本。另请注意，RHEL 9 包括 SEV 和 SEV-ES 加密，但不包括 SEV 和 SEV-ES 安全测试。

Jira:RHELPLAN-65217[1]

虚拟化现在在 ARM 64 上可用

作为技术预览，现在可以使用 ARM 64 CPU 在系统中创建 KVM 虚拟机。

Jira:RHELPLAN-103993[1]

virtio-mem 现在包括在 AMD64、Intel 64 和 ARM 64 中

作为技术预览，RHEL 9 在 AMD64、Intel 64 和 ARM 64 系统中引入了 **virtio-mem** 功能。使用 **virtio-mem** 可让虚拟机(VM)动态添加或删除主机内存。

要使用 **virtio-mem**，请在虚拟机 XML 配置中定义 **virtio-mem** 内存设备，并使用 **virsh update-memory-device** 命令请求 VM 运行期间内存设备大小更改。要查看此类内存设备向正在运行的虚拟机公开的当前内存大小，请查看虚拟机的 XML 配置。

但请注意，**virtio-mem** 目前无法在使用 Windows 操作系统的虚拟机上工作。

Bugzilla:2014487, Bugzilla:2044162, Bugzilla:2044172

RHEL 客户机中的 Intel TDX

作为技术预览，Intel Trust Domain Extension (TDX)功能现在可以在 RHEL 9.2 及之后的版本中使用。如果主机系统支持 TDX，您可以部署硬件隔离的 RHEL 9 虚拟机(VM)，称为信任域(TD)。但请注意，TDX 目前无法与 **kdump** 一起工作，启用 TDX 会导致 **kdump** 在虚拟机上失败。

Bugzilla:1955275[1]

RHEL 的统一内核镜像现在作为技术预览提供

作为技术预览，您现在可以获取 RHEL 内核作为虚拟机(VM）的统一内核镜像(UKI)。统一内核镜像将 kernel、initramfs 和内核命令行合并成一个签名的二进制文件。

UKI 可用于虚拟和云环境中，特别是在需要强大的 SecureBoot 功能的机密虚拟机中。UKI 作为 RHEL 9 存储库中的 **kernel-uki-virt** 软件包提供。

目前，RHEL UKI 只能在 UEFI 引导配置中使用。

Bugzilla:2142102[1]

Intel vGPU 作为技术预览提供

作为技术预览，可以将物理 Intel GPU 设备划分为多个虚拟设备，称为 **介质设备**。然后可将这些介质设备分配给多个虚拟机(VM)作为虚拟 GPU。因此,这些虚拟机共享单个物理 Intel GPU 的性能。

请注意，这个功能已弃用，并完全在 RHEL 9.3 发行版本中删除。

Jira:RHELDOCS-17050[1]

9.12. 云环境中的 RHEL

RHEL 现在在 Azure 机密虚拟机上作为技术预览提供

使用更新的 RHEL 内核，您现在可以在 Microsoft Azure 上作为技术预览创建并运行 RHEL 机密虚拟机(VM)。新添加的统一内核镜像(UKI)现在在 Azure 上可以引导加密的机密虚拟机镜像。UKI 作为 RHEL 9 存储库中的 **kernel-uki-virt** 软件包提供。

目前，RHEL UKI 只能在 UEFI 引导配置中使用。

Jira:RHELPLAN-139800[1]

9.13. 容器

Podman 的 SQLite 数据库后端作为技术预览提供

从 Podman v4.6 开始，Podman 的 SQLite 数据库后端作为技术预览提供。要将数据库后端设置为 SQLite，请在 **/etc/containers/containers.conf** 配置文件中添加 **database_backend = "sqlite"** 选项。在切换到 SQLite 数据库后端前，请运行 **podman system reset** 命令，来将存储后端重置为初始状态。

请注意，您必须重新创建所有容器和 pod。SQLite 数据库保证好的稳定性和一致性。容器堆栈中的其他数据库也将被移到 SQLite。BoltDB 保留默认的数据库后端。

Jira:RHELPLAN-154429[1]

podman-machine 命令不被支持

用于管理虚拟机的 **podman-machine** 命令仅作为技术预览提供。相反，直接从命令行运行 Podman。

Jira:RHELDOCS-16861[1]

第 10 章 过时的功能

这部分提供在 Red Hat Enterprise Linux 9 中*弃用*的功能概述。

弃用的功能可能在以后的主要发行本中不被支持，因此不建议在新的部署中使用。有关特定主要发行本中已弃用功能的最新列表，请参考最新版本的发行文档。

弃用的设备被完全支持，这意味着它们经过测试和维护，并且它们的支持状态在 Red Hat Enterprise Linux 9 中保持不变。有关支持长度的详情，请查看 Red Hat Enterprise Linux 生命周期 和 Red Hat Enterprise Linux 应用程序流生命周期。

对于当前或将来的主发行版本中的新部署，我们不推荐使用已弃用的硬件组件。红帽建议尽快替换这个硬件。

一个软件包可以被弃用，我们不推荐在以后使用。在某些情况下，软件包可从产品中删除。然后，产品文档可识别提供类似、完全相同或者更高级功能的最新软件包，并提供进一步建议。

有关 RHEL 8 中存在但已在 RHEL 9 中 *删除* 的功能的详情，请参阅 采用 RHEL 9 时的注意事项。

10.1. 安装程序和镜像创建

弃用的 Kickstart 命令

以下 Kickstart 命令已弃用：

- **timezone --ntpservers**
- **timezone --nontp**
- **logging --level**
- **%packages --excludeWeakdeps**
- **%packages --instLangs**
- **%anaconda**
- **pwpolicy**

请注意，当只列出具体选项时，基础命令及其其它选项仍可用且未被弃用。在 Kickstart 文件中使用已弃用的命令会在日志中显示警告信息。您可以使用 **inst.ksstrict** 引导选项将已弃用的命令警告转换为错误。

Bugzilla:1899167[1]

edge-commit 和 edge-container 蓝图中的用户和组自定义已弃用

在蓝图中指定用户或组自定义对于 **edge-commit** 和 **edge-container** 镜像类型已弃用，因为在升级镜像时用户自定义会消失，且不能在蓝图中再次指定用户。

请注意，在用于部署现有的 OSTree 提交的蓝图中指定用户或组自定义，如 **edge-raw-image**、**edge-installer** 和 **edge-simplified-installer** 镜像类型继续被支持。

Bugzilla:2173928

initial-setup 软件包现已弃用

initial-setup 软件包已在 Red Hat Enterprise Linux 9.3 中被弃用，并将在下一个主 RHEL 发行版本中删除。作为替换，对图形用户界面，使用 **gnome-initial-setup**。

Jira:RHELDOCS-16393[1]

inst.geoloc 引导选项的 provider_hostip 和 provider_fedora_geoip 值已弃用

为 **inst.geoloc=** 引导选项指定 GeoIP API 的 **provider_hostip** 和 **provider_fedora_geoip** 值已弃用。作为替换，您可以使用 **geolocation_provider=URL** 选项在安装程序配置文件中设置所需的地理位置。您仍然可以使用 **inst.geoloc=0** 选项禁用地理位置。

Bugzilla:2127473

10.2. 安全性

对于加密目的，SHA-1 已被弃用

使用 SHA-1 消息摘要用于加密目的在 RHEL 9 中已被弃用。SHA-1 生成的摘要不被视为是安全的，因为已发现多个基于哈希进行的安全攻击。RHEL 核心加密组件不再默认使用 SHA-1 创建签名。RHEL 9 中的应用程序已更新，以避免在与安全相关的用例中使用 SHA-1。

其中一个例外是，仍然可以使用 SHA-1 创建 HMAC-SHA1 消息验证代码和 Universal Unique Identifier(UUID)值，因为这些用例目前不存在安全风险。另外，为了保持一些重要的互操作性和兼容性，SHA-1 还会在一些有限的情况下使用，例如 Kerberos 和 WPA-2。详情请查看 RHEL 9 安全强化文档中的使用与 FIPS 140-3 不兼容的加密系统的 RHEL 应用程序列表。

如果您需要使用 SHA-1 来验证现有或第三方加密签名，您可以输入以下命令启用它：

```
# update-crypto-policies --set DEFAULT:SHA1
```

或者，您可以将系统范围的加密策略切换到 **LEGACY** 策略。请注意，**LEGACY** 也启用了其他一些不安全的算法。

Jira:RHELPLAN-110763[1]

fapolicyd.rules 已被弃用

包含允许和拒绝执行规则的文件的 **/etc/fapolicyd/rules.d/** 目录替代了 **/etc/fapolicyd/fapolicyd.rules** 文件。**fagenrules** 脚本现在将此目录中的所有组件规则文件合并到 **/etc/fapolicyd/compiled.rules** 文件。**/etc/fapolicyd/fapolicyd** 中的规则仍由 **fapolicyd** 框架处理，但只是为了保证向后兼容。

Bugzilla:2054740

在 RHEL 9 中弃用 SCP

安全复制协议(SCP)已弃用，因为它有已知的安全漏洞。SCP API 仍可用于 RHEL 9 生命周期，但使用它可以降低系统安全性。

- 在 **scp** 实用程序中，默认情况下，SCP 被 SSH 文件传输协议(SFTP)替代。
- OpenSSH 套件在 RHEL 9 中不使用 SCP。
- SCP 在 **libssh** 库中已弃用。

Jira:RHELPLAN-99136[1]

OpenSSL 需要在 FIPS 模式下对 RSA 加密进行填充

OpenSSL 在 FIPS 模式下不再支持没有填充的 RSA 加密。没有填充的 RSA 加密不常见，很少使用。请注意，带有 RSA (RSASVE)的密钥封装不使用填充，但仍支持。

Bugzilla:2168665

NTLM 和 Krb4 在 Cyrus SASL 中已弃用

NTLM 和 Kerberos 4 验证协议已弃用，并可能在以后的 RHEL 主版本中删除。这些协议不再被视为安全的，已从上游实现中删除。

Jira:RHELDOCS-17380[1]

SASL 中的 digest-MD5 已被弃用

Simple Authentication Security Layer(SASL)框架中的 Digest-MD5 身份验证机制已弃用，并可能在以后的主发行版本中从 **cyrus-sasl** 软件包中删除。

Bugzilla:1995600[1]

OpenSSL 弃用了 MD2, MD4, MDC2, Whirlpool, Blowfish, CAST, DES, IDEA, RC2, RC4, RC5, SEED, 和 PBKDF1

OpenSSL 项目已弃用了一组加密算法，因为它们不安全，不常用，或两者都不安全。红帽还不建议使用这些算法，RHEL 9 则为其提供迁移加密数据以使用新的算法。对于系统的安全性，用户不得依赖于这些算法。

以下算法的实现已移到 OpenSSL 中的旧提供者：MD2、MD4、MD4、MDC2、Mlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED 和 PBKDF1。

有关如何载入旧供应商的说明，请参阅 **/etc/pki/tls/openssl.cnf** 配置文件，并启用对已弃用算法的支持。

Bugzilla:1975836

/etc/system-fips 现已弃用

支持通过 **/etc/system-fips** 文件指定 FIPS 模式，该文件将不会包含在将来的 RHEL 版本中。要在 FIPS 模式中安装 RHEL，请在系统安装过程中将 **fips=1** 参数添加到内核命令行。您可以使用 **fips-mode-setup --check** 命令检查 RHEL 是否以 FIPS 模式运行。

Jira:RHELPLAN-103232[1]

libcrypt.so.1 现已弃用

libcrypt.so.1 库现已弃用，它可能会在以后的 RHEL 版本中删除。

Bugzilla:2034569

10.3. 订阅管理

subscription-manager 命令的 --token 选项已弃用

subscription-manager register 命令的 **--token=<TOKEN>** 选项是一种身份验证方法，可帮助将您的系统注册到红帽。这个选项取决于授权服务器提供的功能。默认授权服务器 **subscription.rhsm.redhat.com** 计划关闭此功能。因此，尝试使用 **subscription-manager register --token=<TOKEN>** 可能会失败，并显示以下错误消息：

```
Token authentication not supported by the entitlement server
```

您可以使用其他授权方法继续注册您的系统，例如包括 **subscription-manager register** 命令的成对的选项 **--username** / **--password** 和 **--org** / **--activationkey**。

Bugzilla:2163716

10.4. SHELL 和命令行工具

在 ReaR 配置文件中设置 TMPDIR 变量已弃用

使用诸如 **export TMPDIR=...** 的声明，在 **/etc/rear/local.conf** 或 **/etc/rear/site.conf** ReaR 配置文件中设置 **TMPDIR** 环境变量无法工作，且已弃用。

要为 ReaR 临时文件指定一个自定义目录，请在执行 ReaR 前在 shell 环境中导出变量。例如，执行 **export TMPDIR=...** 语句，然后在同一 shell 会话或脚本中执行 **rear** 命令。

Jira:RHELDOCS-18049

dump 软件包中的 dump 工具已弃用

用于文件系统备份的 **dump** 工具已弃用，在 RHEL 9 中将不再提供。

在 RHEL 9 中，红帽建议根据使用情况使用 **tar**、**dd** 或 **bacula** 、备份工具，对 ext2、ext3 和 ext4 文件系统提供了完整和安全的备份。

请注意，**dump** 软件包中的 **restore** 工具仍可用，在 RHEL 9 中也被支持，并作为 **restore** 软件包提供。

Bugzilla:1997366[1]

Bacula 中的 SQLite 数据库后端已被弃用

Bacula 备份系统支持多个数据库后端：PostgreSQL、MySQL 和 SQLite。SQLite 后端已被弃用，并将在以后的 RHEL 版本中不被支持。作为一种替代，迁移到其他一种后端(PostgreSQL 或 MySQL)，且在新部署中不使用 SQLite 后端。

Jira:RHEL-6856

10.5. 网络

RHEL 9 中已弃用网络团队（Network teams）

teamd 服务和 **libteam** 库在 Red Hat Enterprise Linux 9 中已弃用，并将在下一个主发行版本中删除。作为替换，配置绑定而不是网络组。

红帽注重于基于内核的绑定操作，以避免维护具有类似功能的两个功能：绑定和团队（team）。绑定代码具有较高的客户采用率，非常可靠，具有活跃的社区开发。因此，绑定代码会收到功能增强和更新。

有关如何将团队迁移到绑定的详情，请参阅将网络组配置迁移到网络绑定。

Bugzilla:1935544[1]

ifcfg 格式的 NetworkManager 连接配置文件已弃用

在 RHEL 9.0 及更高版本中，**ifcfg** 格式的连接配置文件已弃用。下一个主要 RHEL 发行版本将删除对这个格式的支持。但是，在 RHEL 9 中，如果修改了配置文件，NetworkManager 仍然会使用这个格式处理和更新现有的配置文件。

默认情况下，NetworkManager 现在在 **/etc/NetworkManager/system-connections/** 目录中以 keyfile 格式存储连接配置文件。与 **ifcfg** 格式不同，keyfile 格式支持 NetworkManager 提供的所有连接设置。有关 keyfile 格式以及如何迁移配置集的详情，请参考 keyfile 格式的 NetworkManager 连接配置文件。

Bugzilla:1894877[1]

firewalld 中的 iptables 后端已弃用

在 RHEL 9 中，**iptables** 框架已弃用。因此，**firewalld** 中的 **iptables** 后端和 **direct interface** 也被弃用。您可以使用 **firewalld** 中的原生功能，而不是 **direct interface** 来配置所需的规则。

Bugzilla:2089200

PF_KEYv2 内核 API 已被弃用

应用程序可以使用 **PV_KEYv2** 和较新的 **netlink** API 配置内核的 IPsec 实现。**PV_KEYv2** 没有在上游进行主动维护，并且缺少重要的安全功能，如现代密码、卸载和扩展的序列号支持。因此，从 RHEL 9.3 开始，**PV_KEYv2** API 已被弃用，并将在下一个主 RHEL 发行版本中删除。如果您在应用程序中使用此内核 API，请对其进行迁移，以使用现代 **netlink** API 作为替代。

Jira:RHEL-1015[1]

10.6. 内核

在 RHEL 9 中弃用 ATM 封装

异步传输模式(ATM)封装为 ATM Adaptation Layer 5(AAL-5)提供第 2 层（Point-to-Point 协议、以太网）或第 3 层（IP）连接。从 RHEL 7 开始，红帽尚未为 ATM NIC 驱动程序提供支持。RHEL 9 中丢弃对 ATM 实施的支持。这些协议目前仅在芯片组中使用，该协议支持 ADSL 技术，并由制造商逐步淘汰。因此，Red Hat Enterprise Linux 9 中已弃用 ATM 封装。

如需更多信息，请参阅 PPP Over AAL5, Multiprotocol Encapsulation over ATM Adaptation Layer 5 , 和 Classical IP and ARP over ATM 。

Bugzilla:2058153

kexec-tools 的 kexec_load 系统调用已弃用

在以后的 RHEL 版本中将不支持 **kexec_load** 系统调用（其载入第二个内核）。**kexec_file_load** 系统调用替换了 **kexec_load**，它现在是所有架构上的默认系统调用。

如需更多信息，请参阅 RHEL9 中是否支持 kexec_load？

Bugzilla:2113873[1]

RHEL 9 中已弃用网络团队（Network teams）

teamd 服务和 **libteam** 库在 Red Hat Enterprise Linux 9 中已弃用，并将在下一个主发行版本中删除。作为替换，配置绑定而不是网络组。

红帽注重于基于内核的绑定操作，以避免维护具有类似功能的两个功能：绑定和团队（team）。绑定代码具有较高的客户采用率，非常可靠，具有活跃的社区开发。因此，绑定代码会收到功能增强和更新。

有关如何将团队迁移到绑定的详情，请参阅将网络组配置迁移到网络绑定。

Bugzilla:2013884[1]

10.7. 文件系统和存储

lvm2-activation-generator 及其生成的服务在 RHEL 9.0 中删除

lvm2-activation-generator 程序及其生成的服务 **lvm2-activation**、**lvm2-activation-early**、**lvm2-activation-net** 已在 RHEL 9.0 中删除。 **lvm.conf event_activation** 设置用于激活服务将不再起作用。自动激活卷组的唯一方法是基于事件激活。

Bugzilla:2038183

在 RHEL 9 中已弃用了持久性内存开发套件(pmdk)和支持库

pmdk 是用于系统管理员和应用程序开发者的库和工具集合，以简化管理和访问持久内存设备。RHEL 9 中已弃用了 **pmdk** 和支持库。这还包括 **-debuginfo** 软件包。

由 **pmdk** 产生的以下二进制软件包列表，包括 **nvml** 源软件包，已被弃用：

- **libpmem**
- **libpmem-devel**
- **libpmem-debug**
- **libpmem2**
- **libpmem2-devel**
- **libpmem2-debug**
- **libpmemblk**
- **libpmemblk-devel**
- **libpmemblk-debug**
- **libpmemlog**
- **libpmemlog-devel**
- **libpmemlog-debug**
- **libpmemobj**
- **libpmemobj-devel**
- **libpmemobj-debug**
- **libpmempool**
- **libpmempool-devel**
- **libpmempool-debug**
- **pmempool**
- **daxio**
- **pmreorder**

- **pmdk-convert**
- **libpmemobj++**
- **libpmemobj++-devel**
- **libpmemobj++-doc**

Jira:RHELDOCS-16432[1]

10.8. 动态编程语言、网页和数据库服务器

libdb 已被弃用

RHEL 8 和 RHEL 9 目前提供 Berkeley DB(**libdb**)版本 5.3.28，该版本根据 LGPLv2 许可证发布。上游 Berkeley DB 版本 6 在 AGPLv3 许可证下提供，该许可证更严格。

从 RHEL 9 开始，**libdb** 软件包已弃用，可能不会在以后的 RHEL 版本中可用。

另外，在 RHEL 9 中，加密算法已从 **libdb** 中删除，从 RHEL 9 中删除了多个 **libdb** 依赖项。

建议 **libdb** 用户迁移到其他键值数据库。如需更多信息，请参阅 RHEL 中已弃用的 Berkeley DB(libdb) 的知识库文章。

Bugzilla:1927780[1], JIRA:RHELPLAN-80695, Bugzilla:1974657

10.9. 编译器和开发工具

Go 的 FIPS 模式下，比 2048 小的密钥已被 openssl 3.0 弃用

openssl 3.0 弃用了小于 2048 位的密钥，在 Go 的 FIPS 模式中无法正常工作。

Bugzilla:2111072

有些 PKCS1 v1.5 模式现在在 Go 的 FIPS 模式下被弃用

一些 **PKCS1** v1.5 模式在 **FIPS-140-3** 中未被批准用于加密，并被禁用。它们将不再在 Go 的 FIPS 模式下工作。

Bugzilla:2092016[1]

10.10. 身份管理

OpenDNSSec 中的 SHA-1 现已弃用

OpenDNSSEC 支持使用 **SHA-1** 算法导出数字签名和身份验证记录。不再支持使用 **SHA-1** 算法。在 RHEL 9 发行版本中，OpenDNSSec 中的 **SHA-1** 已被弃用，并可能在以后的次版本中删除。另外，OpenDNSSec 支持仅限于与红帽身份管理的集成。OpenDNSSEC 不支持独立。

Bugzilla:1979521

SSSD 隐式文件供应商域默认禁用

SSSD 隐式 **文件** 供应商域，从 **/etc/shadow** 和 **/etc/** groups 等本地文件检索用户信息，现已默认禁用。

使用 SSSD 从本地文件检索用户和组信息：

1. 配置 SSSD.选择以下选项之一：

   a. 使用 **sssd.conf** 配置文件中的 **id_provider=files** 选项明确配置本地域。

   ```
   [domain/local]
   id_provider=files
   ...
   ```

   b. 通过在 **sssd.conf** 配置文件中设置 **enable_files_domain=true** 来启用 **文件** 供应商。

   ```
   [sssd]
   enable_files_domain = true
   ```

2. 配置名称服务切换。

   ```
   # authselect enable-feature with-files-provider
   ```

Jira:RHELPLAN-100639[1]

SSSD 文件 提供者已弃用

SSSD **文件** 提供者已在 Red Hat Enterprise Linux (RHEL) 9 中弃用。 **文件** 提供者可能会从以后的版本中删除。

Jira:RHELPLAN-139805[1]

nsslapd-ldapimaprootdn 参数已弃用

在目录服务器中，**nsslapd-ldapimaprootdn** 配置参数用于将系统根条目映射到根 DN 条目。通常，**nsslapd-ldapimaprootdn** 参数具有与 **nsslapd-rootdn** 参数相同的值。另外，更改一个属性但不更改其它属性会导致一个无法正常工作的自动绑定配置，其会破坏 **dsconf** 工具及对 Web 控制台的访问。

有了此更新，目录服务器只使用 **nsslapd-rootdn** 参数将系统根条目映射到根 DN 条目。因此， **nsslapd-ldapimaprootdn** 参数被弃用，根 DN 更改不会破坏 **dsconf** 工具以及对 web 控制台的访问。

Bugzilla:2170494

nsslapd-conntablesize 配置参数已从 389-ds-base中删除

nsslapd-conntablesize 配置参数已从 RHEL 9.3 中的 **389-ds-base** 软件包中删除。在以前的版本中，**nsslapd-conntablesize** 配置属性指定管理建立的连接的连接表的大小。随着多监听器功能的引入，它改进了已建立连接的管理，目录服务器现在可以动态计算连接表的大小。这也解决了这类问题，当连接表大小被设置得太小时，它会影响服务器能够支持的连接的数量。从 RHEL 9.3 开始，只使用 **nsslapd-maxdescriptors** 和 **nsslapd-reservedescriptors** 属性来管理目录服务器可以支持的 TCP/IP 连接的数量。

Bugzilla:2098236

SMB1 协议在 Samba 中已弃用

从 Samba 4.11 开始，不安全的服务器消息块版本 1 (SMB1)协议已弃用，并将在以后的发行版本中删除。

为提高安全性，在 Samba 服务器和客户端工具中默认禁用 SMB1。

Jira:RHELDOCS-16612[1]

10.11. 桌面

GTK 2 现已弃用

旧的 GTK 2 工具包及以下相关软件包已弃用：

- **adwaita-gtk2-theme**
- **gnome-common**
- **gtk2**
- **gtk2-immodules**
- **hexchat**

其它几个软件包目前依赖于 GTK 2。这些已被修改，以便它们不再依赖于未来的主 RHEL 发行版本中已弃用的软件包。

如果您维护使用 GTK 2 的应用程序，红帽建议您将应用移植到 GTK 4。

Jira:RHELPLAN-131882[1]

libreoffice 已被弃用

LibreOffice RPM 软件包现已弃用，并将在以后的主 RHEL 发行版本中删除。LibreOffice 在 RHEL 7、8 和 9 的整个生命周期中仍然被完全支持。

作为 RPM 软件包的替代，红帽建议您从 Document Foundation 提供的以下源中安装 LibreOffice：

- Flathub 存储库中的官方 Flatpak 软件包：https://flathub.org/apps/org.libreoffice.LibreOffice。
- 官方 RPM 软件包：：https://www.libreoffice.org/download/download-libreoffice/

Jira:RHELDOCS-16300[1]

10.12. 图形基础结构

Motif 已被弃用

Motif 小部件工具包已在 RHEL 中被弃用，因为上游 Motif 社区的开发不活跃。

以下 Motif 软件包已被弃用，包括其开发和调试变体：

- **motif**
- **openmotif**
- **openmotif21**
- **openmotif22**

另外，**motif-static** 软件包已删除。

红帽建议使用 GTK 工具包作为替代品。与 Motif 相比，GTK更易于维护，并提供了新功能。

JIRA:RHELPLAN-98983[1]

10.13. RED HAT ENTERPRISE LINUX 系统角色

在 RHEL 9 节点上配置 team 时， network 系统角色显示一条弃用警告

网络 team 功能在RHEL 9 中已被弃用。因此，在 RHEL 8 控制节点上使用 **network** RHEL 系统角色在 RHEL 9 节点上配置网络 team ，显示一条有关弃用的警告。

Bugzilla:1999770

10.14. 虚拟化

使用基于 SHA1 的签名进行 SecureBoot 镜像验证已弃用

在 UEFI（PE/COFF）可执行文件中使用基于 SHA1 的签名执行 SecureBoot 镜像验证已过时。反之，红帽建议使用基于 SHA2 算法或更新版本的签名。

Bugzilla:1935497[1]

对虚拟机快照的支持有限

目前只对使用 UEFI 固件的虚拟机支持创建虚拟机(VM)的快照。另外，在快照操作过程中，QEMU 监控可能会被阻断，这会对某些工作负载的 hypervisor 性能造成负面影响。

另请注意，创建虚拟机快照的当前机制已被弃用，红帽不推荐在生产环境中使用虚拟机快照。但是，一个新的虚拟机快照机制正在开发中，计划在以后的 RHEL 9 次要发行本中完全实施。

JIRA:RHELDOCS-16948[1], Bugzilla:1621944

虚拟软盘驱动程序已弃用

用于控制虚拟软盘设备的 **isa-fdc** 驱动程序现已弃用，并将在以后的 RHEL 发行版本中不被支持。因此，为了确保与迁移的虚拟机(VM)兼容，红帽不建议在 RHEL 9 上托管的虚拟机中使用软盘磁盘设备。

Bugzilla:1965079

qcow2-v2 镜像格式已弃用

在 RHEL 9 中，虚拟磁盘镜像的 qcow2-v2 格式已弃用，并将在以后的 RHEL 主发行版本中不被支持。另外，RHEL 9 Image Builder 无法以 qcow2-v2 格式创建磁盘镜像。

红帽强烈建议您使用 qcow2-v3，而不是 qcow2-v2。要将 qcow2-v2 镜像转换为更新的格式版本，请使用 **qemu-img amend** 命令。

Bugzilla:1951814

virt-manager 已被弃用

虚拟机管理器（也称 **virt-manager**） 已弃用。RHEL web 控制台（也称为 **Cockpit** ）旨在在以后的版本中成为其替代品。因此，建议您使用 web 控制台使用 GUI 管理虚拟化。但请注意，**virt-manager** 中的一些可用功能可能在 RHEL web 控制台中不可用。

Jira:RHELPLAN-10304[1]

libvirtd 已被弃用

单体 **libvirt** 守护进程 **libvirtd** 已在 RHEL 9 中弃用，并将在以后的 RHEL 主发行版本中删除。请注意，您仍然可以使用 **libvirtd** 在虚拟机监控程序上管理虚拟化，但红帽建议您切换到新引入的模块化 **libvirt** 守护进程。具体说明和详情，请参阅 RHEL 9 配置和管理虚拟化 文档。

Jira:RHELPLAN-113995[1]

旧的 CPU 型号现已弃用

大量 CPU 模型已被弃用，并将在以后的 RHEL 主发行版本中的虚拟机 (VM) 不被支持。弃用的模型如下：

- 对于 Intel：Intel Xeon 55xx 和 75xx Processor 系列前的型号（也称为 Nehalem）
- 对于 AMD：AMD Opteron G4 之前的型号
- 对于 IBM Z：IBM z14 之前的型号

要检查您的虚拟机是否使用已弃用的 CPU 模型，请使用 **virsh dominfo** 工具，并在 **Messages** 部分查找类似如下的行：

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'i486'
```

Bugzilla:2060839

基于 RDMA 的实时迁移已弃用

有了这个更新，使用 Remote Direct Memory Access (RDMA)迁移正在运行的虚拟机已被弃用。因此，仍可以使用 **rdma://** 迁移 URI 来通过 RDMA 请求迁移，但这个功能将在以后的 RHEL 主发行版本中不被支持。

Jira:RHELPLAN-153267[1]

Intel vGPU 功能已被删除

在以前的版本中，作为技术预览，可以将物理 Intel GPU 设备划分为多个虚拟设备，称为 **介质设备**。然后，这些介质设备可以分配给多个虚拟机(VM)作为虚拟 GPU。因此，这些虚拟机共享单个物理 Intel GPU 的性能，但只有所选的 Intel GPU 与此功能兼容。

从 RHEL 9.3 开始，Intel vGPU 功能已完全删除。

Bugzilla:2206599[1]

10.15. 容器

不支持在 RHEL 7 主机上运行 RHEL 9 容器

不支持在 RHEL 7 主机上运行 RHEL 9 容器。它可能可以正常工作，但却没有保证。

如需更多信息，请参阅 Red Hat Enterprise Linux Container Compatibility Matrix 。

Jira:RHELPLAN-100087[1]

Podman 中的 SHA1 哈希算法已弃用

Podman 不再支持用来生成无根网络命名空间的文件名的 SHA1 算法。因此，如果在使用 Podman 4.1.1 或更高版本之前启动无根容器，则必须重启它们（而不只是使用 **slirp4netns**），以确保它们可以在升级后启动容器。

Bugzilla:2069279[1]

rhel9/pause 已被弃用

rhel9/pause 容器镜像已被弃用。

Bugzilla:2106816

CNI 网络堆栈已弃用

Container Network Interface (CNI)网络堆栈已弃用，并将从以后 RHEL 次要发行本中的 Podman 中删除。在以前的版本中，容器只能通过 DNS 连接到单个 Container Network Interface (CNI)插件。podman v.4.0 引入了一个新的 Netavark 网络堆栈。您可以将 Netavark 网络堆栈与 Podman 和其他Open Container Initiative(OCI)容器管理应用程序一起使用。Podman 的 Netavark 网络堆栈也与高级 Docker 功能兼容。多个网络中的容器可以访问任何这些网络上的容器。

如需更多信息，请参阅 将网络堆栈从 CNI 切换到 Netavark。

Jira:RHELDOCS-16756[1]

Inkscape 和 LibreOffice Flatpak 镜像已弃用

作为技术预览提供的 **rhel9/inkscape-flatpak** 和 **rhel9/libreoffice-flatpak** Flatpak 镜像已被弃用。

红帽建议对这些镜像使用以下替代方案：

- 要替换 **rhel9/inkscape-flatpak**，请使用 **inkscape** RPM 软件包。
- 要替换 **rhel9/libreoffice-flatpak**，请参阅 LibreOffice 弃用发行注记。

Jira:RHELDOCS-17102[1]

10.16. 已弃用的软件包

本节列出了已弃用的软件包，可能不会包括在 Red Hat Enterprise Linux 未来的主发行版本中。

有关 RHEL 8 和 RHEL 9 之间软件包的更改，请参阅*使用 RHEL 9 文档*中的软件包的更改。

重要

在 RHEL 9 中，已弃用软件包的支持状态保持不变。有关支持长度的更多信息，请参阅 Red Hat Enterprise Linux 生命周期 和 Red Hat Enterprise Linux 应用程序流生命周期 。

以下软件包已在 RHEL 9 中弃用：

- adwaita-gtk2-theme
- autocorr-af
- autocorr-bg
- autocorr-ca
- autocorr-cs
- autocorr-da
- autocorr-de

- autocorr-dsb
- autocorr-el
- autocorr-en
- autocorr-es
- autocorr-fa
- autocorr-fi
- autocorr-fr
- autocorr-ga
- autocorr-hr
- autocorr-hsb
- autocorr-hu
- autocorr-is
- autocorr-it
- autocorr-ja
- autocorr-ko
- autocorr-lb
- autocorr-lt
- autocorr-mn
- autocorr-nl
- autocorr-pl
- autocorr-pt
- autocorr-ro
- autocorr-ru
- autocorr-sk
- autocorr-sl
- autocorr-sr
- autocorr-sv
- autocorr-tr
- autocorr-vi

- autocorr-vro
- autocorr-zh
- cheese
- cheese-libs
- clutter
- clutter-gst3
- clutter-gtk
- cogl
- daxio
- dbus-glib
- dbus-glib-devel
- enchant
- enchant-devel
- eog
- evolution
- evolution-bogofilter
- evolution-devel
- evolution-help
- evolution-langpacks
- evolution-mapi
- evolution-mapi-langpacks
- evolution-pst
- evolution-spamassassin
- festival
- festival-data
- festvox-slt-arctic-hts
- flite
- flite-devel
- gedit

- gedit-plugin-bookmarks
- gedit-plugin-bracketcompletion
- gedit-plugin-codecomment
- gedit-plugin-colorpicker
- gedit-plugin-colorschemer
- gedit-plugin-commander
- gedit-plugin-drawspaces
- gedit-plugin-findinfiles
- gedit-plugin-joinlines
- gedit-plugin-multiedit
- gedit-plugin-sessionsaver
- gedit-plugin-smartspaces
- gedit-plugin-synctex
- gedit-plugin-terminal
- gedit-plugin-textsize
- gedit-plugin-translate
- gedit-plugin-wordcompletion
- gedit-plugins
- gedit-plugins-data
- gnome-common
- gnome-photos
- gnome-photos-tests
- gnome-screenshot
- gnome-themes-extra
- gtk2
- gtk2-devel
- gtk2-devel-docs
- gtk2-immodule-xim
- gtk2-immodules

- highcontrast-icon-theme
- Inkscape
- inkscape-docs
- inkscape-view
- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- libgdata
- libgdata-devel
- libpmem
- libpmem-debug
- libpmem-devel
- libpmem2
- libpmem2-debug
- libpmem2-devel
- libpmemblk
- libpmemblk-debug
- libpmemblk-devel
- libpmemlog
- libpmemlog-debug
- libpmemlog-devel
- libpmemobj
- libpmemobj-debug
- libpmemobj-devel
- libpmempool
- libpmempool-debug

- libpmempool-devel
- libreoffice
- libreoffice-base
- libreoffice-calc
- libreoffice-core
- libreoffice-data
- libreoffice-draw
- libreoffice-emailmerge
- libreoffice-filters
- libreoffice-gdb-debug-support
- libreoffice-graphicfilter
- libreoffice-gtk3
- libreoffice-help-ar
- libreoffice-help-bg
- libreoffice-help-bn
- libreoffice-help-ca
- libreoffice-help-cs
- libreoffice-help-da
- libreoffice-help-de
- libreoffice-help-dz
- libreoffice-help-el
- libreoffice-help-en
- libreoffice-help-eo
- libreoffice-help-es
- libreoffice-help-et
- libreoffice-help-eu
- libreoffice-help-fi
- libreoffice-help-fr
- libreoffice-help-gl

- libreoffice-help-gu
- libreoffice-help-he
- libreoffice-help-hi
- libreoffice-help-hr
- libreoffice-help-hu
- libreoffice-help-id
- libreoffice-help-it
- libreoffice-help-ja
- libreoffice-help-ko
- libreoffice-help-lt
- libreoffice-help-lv
- libreoffice-help-nb
- libreoffice-help-nl
- libreoffice-help-nn
- libreoffice-help-pl
- libreoffice-help-pt-BR
- libreoffice-help-pt-PT
- libreoffice-help-ro
- libreoffice-help-ru
- libreoffice-help-si
- libreoffice-help-sk
- libreoffice-help-sl
- libreoffice-help-sv
- libreoffice-help-ta
- libreoffice-help-tr
- libreoffice-help-uk
- libreoffice-help-zh-Hans
- libreoffice-help-zh-Hant
- libreoffice-impress

- libreoffice-langpack-af
- libreoffice-langpack-ar
- libreoffice-langpack-as
- libreoffice-langpack-bg
- libreoffice-langpack-bn
- libreoffice-langpack-br
- libreoffice-langpack-ca
- libreoffice-langpack-cs
- libreoffice-langpack-cy
- libreoffice-langpack-da
- libreoffice-langpack-de
- libreoffice-langpack-dz
- libreoffice-langpack-el
- libreoffice-langpack-en
- libreoffice-langpack-eo
- libreoffice-langpack-es
- libreoffice-langpack-et
- libreoffice-langpack-eu
- libreoffice-langpack-fa
- libreoffice-langpack-fi
- libreoffice-langpack-fr
- libreoffice-langpack-fy
- libreoffice-langpack-ga
- libreoffice-langpack-gl
- libreoffice-langpack-gu
- libreoffice-langpack-he
- libreoffice-langpack-hi
- libreoffice-langpack-hr
- libreoffice-langpack-hu

- libreoffice-langpack-id
- libreoffice-langpack-it
- libreoffice-langpack-ja
- libreoffice-langpack-kk
- libreoffice-langpack-kn
- libreoffice-langpack-ko
- libreoffice-langpack-lt
- libreoffice-langpack-lv
- libreoffice-langpack-mai
- libreoffice-langpack-ml
- libreoffice-langpack-mr
- libreoffice-langpack-nb
- libreoffice-langpack-nl
- libreoffice-langpack-nn
- libreoffice-langpack-nr
- libreoffice-langpack-nso
- libreoffice-langpack-or
- libreoffice-langpack-pa
- libreoffice-langpack-pl
- libreoffice-langpack-pt-BR
- libreoffice-langpack-pt-PT
- libreoffice-langpack-ro
- libreoffice-langpack-ru
- libreoffice-langpack-si
- libreoffice-langpack-sk
- libreoffice-langpack-sl
- libreoffice-langpack-sr
- libreoffice-langpack-ss
- libreoffice-langpack-st

- libreoffice-langpack-sv
- libreoffice-langpack-ta
- libreoffice-langpack-te
- libreoffice-langpack-th
- libreoffice-langpack-tn
- libreoffice-langpack-tr
- libreoffice-langpack-ts
- libreoffice-langpack-uk
- libreoffice-langpack-ve
- libreoffice-langpack-xh
- libreoffice-langpack-zh-Hans
- libreoffice-langpack-zh-Hant
- libreoffice-langpack-zu
- libreoffice-math
- libreoffice-ogltrans
- libreoffice-opensymbol-fonts
- libreoffice-pdfimport
- libreoffice-pyuno
- libreoffice-sdk
- libreoffice-sdk-doc
- libreoffice-ure
- libreoffice-ure-common
- libreoffice-wiki-publisher
- libreoffice-writer
- libreoffice-x11
- libreoffice-xsltfilter
- libreofficekit
- libsoup
- libsoup-devel

- libuser
- libuser-devel
- libwpe
- libwpe-devel
- mcpp
- mod_auth_mellon
- motif
- motif-devel
- pmdk-convert
- pmempool
- python3-pytz
- qt5
- qt5-assistant
- qt5-designer
- qt5-devel
- qt5-doctools
- qt5-linguist
- qt5-qdbusviewer
- qt5-qt3d
- qt5-qt3d-devel
- qt5-qt3d-doc
- qt5-qt3d-examples
- qt5-qtbase
- qt5-qtbase-common
- qt5-qtbase-devel
- qt5-qtbase-doc
- qt5-qtbase-examples
- qt5-qtbase-gui
- qt5-qtbase-mysql

- qt5-qtbase-odbc
- qt5-qtbase-postgresql
- qt5-qtbase-private-devel
- qt5-qtbase-static
- qt5-qtconnectivity
- qt5-qtconnectivity-devel
- qt5-qtconnectivity-doc
- qt5-qtconnectivity-examples
- qt5-qtdeclarative
- qt5-qtdeclarative-devel
- qt5-qtdeclarative-doc
- qt5-qtdeclarative-examples
- qt5-qtdeclarative-static
- qt5-qtdoc
- qt5-qtgraphicaleffects
- qt5-qtgraphicaleffects-doc
- qt5-qtimageformats
- qt5-qtimageformats-doc
- qt5-qtlocation
- qt5-qtlocation-devel
- qt5-qtlocation-doc
- qt5-qtlocation-examples
- qt5-qtmultimedia
- qt5-qtmultimedia-devel
- qt5-qtmultimedia-doc
- qt5-qtmultimedia-examples
- qt5-qtquickcontrols
- qt5-qtquickcontrols-doc
- qt5-qtquickcontrols-examples

- qt5-qtquickcontrols2
- qt5-qtquickcontrols2-devel
- qt5-qtquickcontrols2-doc
- qt5-qtquickcontrols2-examples
- qt5-qtscript
- qt5-qtscript-devel
- qt5-qtscript-doc
- qt5-qtscript-examples
- qt5-qtsensors
- qt5-qtsensors-devel
- qt5-qtsensors-doc
- qt5-qtsensors-examples
- qt5-qtserialbus
- qt5-qtserialbus-devel
- qt5-qtserialbus-doc
- qt5-qtserialbus-examples
- qt5-qtserialport
- qt5-qtserialport-devel
- qt5-qtserialport-doc
- qt5-qtserialport-examples
- qt5-qtsvg
- qt5-qtsvg-devel
- qt5-qtsvg-doc
- qt5-qtsvg-examples
- qt5-qttools
- qt5-qttools-common
- qt5-qttools-devel
- qt5-qttools-doc
- qt5-qttools-examples

- qt5-qttools-libs-designer
- qt5-qttools-libs-designercomponents
- qt5-qttools-libs-help
- qt5-qttools-static
- qt5-qttranslations
- qt5-qtwayland
- qt5-qtwayland-devel
- qt5-qtwayland-doc
- qt5-qtwayland-examples
- qt5-qtwebchannel
- qt5-qtwebchannel-devel
- qt5-qtwebchannel-doc
- qt5-qtwebchannel-examples
- qt5-qtwebsockets
- qt5-qtwebsockets-devel
- qt5-qtwebsockets-doc
- qt5-qtwebsockets-examples
- qt5-qtx11extras
- qt5-qtx11extras-devel
- qt5-qtx11extras-doc
- qt5-qtxmlpatterns
- qt5-qtxmlpatterns-devel
- qt5-qtxmlpatterns-doc
- qt5-qtxmlpatterns-examples
- qt5-rpm-macros
- qt5-srpm-macros
- webkit2gtk3
- webkit2gtk3-devel
- webkit2gtk3-jsc

- webkit2gtk3-jsc-devel
- wpebackend-fdo
- wpebackend-fdo-devel
- xorg-x11-server-Xorg

第 11 章 已知问题

这部分论述了 Red Hat Enterprise Linux 9.3 中的已知问题。

11.1. 安装程序和镜像创建

auth 和 authconfig Kickstart 命令需要 AppStream 软件仓库

auth 和 **authconfig** Kickstart 命令在安装过程中需要 **authselect-compat** 软件包。如果没有这个软件包，如果使用了 **auth** 或 **authconfig**，则安装会失败。但根据设计，**authselect-compat** 软件包只包括在 AppStream 仓库中。

要临时解决这个问题，请验证 BaseOS 和 AppStream 存储库是否对安装程序可用，或者在安装过程中使用 **authselect** Kickstart 命令。

Bugzilla:1640697[1]

reboot --kexec 和 inst.kexec 命令不提供可预测的系统状态

使用 **reboot --kexec Kickstart** 命令或 **inst.kexec** 内核引导参数执行 RHEL 安装不会提供与完全重启相同的可预期系统状态。因此，在不重启的情况下切换安装的系统可能会导致无法预计的结果。

请注意，**kexec** 功能已弃用，并将在以后的 Red Hat Enterprise Linux 版本中删除。

Bugzilla:1697896[1]

在 Anaconda 作为应用程序运行的系统中意外 SELinux 策略

当 Anaconda 作为应用程序运行在已安装的系统上（例如，使用 **-image** anaconda 选项对镜像文件执行另一次安装）时，不禁止系统在安装过程中修改 SELinux 类型和属性。因此，某些 SELinux 策略的元素可能会在运行Anaconda 的系统上发生更改。

要临时解决这个问题，请不要在生产系统上运行 Anaconda。相反，在临时虚拟机中运行 Anaconda，以使 SELinux 策略在生产系统上保持不变。作为系统安装过程的一部分运行 anaconda，如从 **boot.iso** 或 **dvd.iso** 安装不会受此问题的影响。

Bugzilla:2050140

当使用使用第三方工具创建的 USB 引导安装时，不会检测 本地介质 安装源

当从使用第三方工具创建的 USB 引导 RHEL 安装时，安装程序无法检测 **Local Media** 安装源（只检测到 *Red Hat CDN* ）。

出现这个问题的原因是，默认的引导选项 **int.stage2=** 会尝试搜索 **iso9660** 镜像格式。但是，第三方工具可能会创建具有不同格式的 ISO 镜像。

作为临时解决方案，请使用以下解决方案之一：

- 当引导安装时，点击 **Tab** 键来编辑内核命令行，并将引导选项 **inst.stage2=** 改为 **inst.repo=**。
- 要在 Windows 中创建可引导 USB 设备，使用 Fedora Media Writer。
- 当使用 Rufus 等第三方工具创建可引导的 USB 设备时，首先在 Linux 系统上重新生成 RHEL ISO 镜像，然后使用第三方工具创建可引导的 USB 设备。

有关执行任何指定的临时解决方案的步骤的更多信息，请参阅 安装介质在 RHEL 8.3 的安装过程中没有被自动探测到。

Bugzilla:1877697[1]

USB CD-ROM 驱动器作为 Anaconda 中的安装源不可用

当源为 USB CD-ROM 驱动器，并且指定了 Kickstart **ignoredisk --only-use=** 命令时，安装会失败。在这种情况下，Anaconda 无法找到并使用这个源磁盘。

要临时解决这个问题，请使用 **harddrive --partition=sdX --dir=/** 命令从 USB CD-ROM 驱动器安装。因此，安装不会失败。

Jira:RHEL-4707

带有 iso9660 文件系统的硬盘分区安装失败

您不能在使用 **iso9660** 文件系统进行分区的系统中安装 RHEL。这是因为将设置为忽略包含 **iso9660** 文件系统分区的硬盘的更新安装代码。即使在没有使用 DVD 的情况下安装 RHEL，也会发生这种情况。

要临时解决这个问题，请在 Kickstart 文件中添加以下脚本，以便在安装开始前格式化磁盘。

注：在执行临时解决方案前，请备份磁盘上的数据。**erafs** 命令对磁盘中的所有现有数据进行格式化。

```
%pre
wipefs -a /dev/sda
%end
```

因此，安装可以正常工作，且没有任何错误。

Jira:RHEL-4711

Anaconda 无法验证管理员用户帐户是否存在

在使用图形用户界面安装 RHEL 时，Anaconda 无法验证管理员帐户是否已创建。因此，用户可以在没有管理员用户帐户的情况下安装系统。

要临时解决这个问题，请确保配置管理员用户帐户或 root 密码已设置，且 root 帐户被解锁。因此，用户可以在安装的系统中执行管理任务。

Bugzilla:2047713

新的 XFS 功能可防止使用比版本 5.10 更早的固件引导 PowerNV IBM POWER 系统

PowerNV IBM POWER 系统使用 Linux 内核进行固件，并使用 Petitboot 作为 GRUB 的替代。这会导致固件内核挂载 **/boot**，Petitboot 读取 GRUB 配置和引导 RHEL。

RHEL 9 内核为 XFS 文件系统引入了 **bigtime=1** 和 **inobtcount=1** 功能，而使用比版本 5.10 旧固件的内核不理解。

要临时解决这个问题，您可以为 **/boot** 使用另一个文件系统，例如 ext4。

Bugzilla:1997832[1]

RHEL for Edge 安装程序镜像在安装 rpm-ostree 有效负载时无法创建挂载点

当部署 **rpm-ostree** 有效负载时，例如在 RHEL for Edge 安装程序镜像中，安装程序不会为自定义分区正确创建一些挂载点。因此，安装会中止，并报以下错误：

```
The command 'mount --bind /mnt/sysimage/data /mnt/sysroot/data' exited with the code 32.
```

要临时解决这个问题：

- 使用自动分区方案，且不手动添加任何挂载点。
- 只在 **/var** 目录中手动分配挂载点。例如： **/var/*my-mount-point*** 和以下标准目录：/、**/boot**、**/var**。

因此，安装过程成功完成。

Jira:RHEL-4741

当连接到网络但没有配置 DHCP 或静态 IP 地址时，NetworkManager 无法在安装后启动

从 RHEL 9.0 开始，当没有设置特定的 **ip=** 或 Kickstart 网络配置时，Anaconda 会自动激活网络设备。Anaconda 为每个以太网设备创建默认的持久配置文件。连接配置文件的 **ONBOOT** 和 **autoconnect** 值设为 **true**。因此，在启动安装的系统的过程中，RHEL 会激活网络设备，**networkManager-wait-online** 服务会失败。

作为临时解决方案，请执行以下操作之一：

- 使用 **nmcli** 工具删除所有连接，但您要使用的一个连接除外。例如：
  a. 列出所有连接配置文件：

  ```
  # nmcli connection show
  ```

  b. 删除您不需要的连接配置文件：

  ```
  # nmcli connection delete <connection_name>
  ```

  将 <connection_name> 替换为您要删除的连接的名称。
- 如果没有设置特定的 **ip=** 或 Kickstart 网络配置，请在 Anaconda 中禁用自动连接网络功能。
  a. 在 Anaconda GUI 中，导航到 **Network & Host Name**。
  b. 选择要禁用的网络设备。
  c. 单击 **Configure**。
  d. 在 **General** 选项卡中，取消 **Connect automatically with priority** 复选框。
  e. 点击 **Save**。

Bugzilla:2115783[1]

无法从安装环境中的驱动程序更新磁盘载入更新的驱动程序

如果已经加载了安装初始 RM 磁盘中同样的驱动程序，则可能不会加载驱动程序更新磁盘中的驱动程序的新版本。因此，驱动程序的更新版本无法应用到安装环境。

作为临时解决方案，请将 **modprobe.blacklist=** 内核命令行选项与 **inst.dd** 选项一起使用。例如，要确保是否加载了驱动程序更新磁盘中的 **virtio_blk** 驱动程序的更新版本，请使用 **modprobe.blacklist=virtio_blk**，然后继续通常的流程来应用驱动程序更新磁盘中的驱动程序。因此，系统会加载驱动程序的更新版本，并在安装环境中使用它。

Jira:RHEL-4762

Kickstart 安装无法配置网络连接

Anaconda 只能通过 NetworkManager API 执行 Kickstart 网络配置。Anaconda 在 **%pre** Kickstart 部分之后处理网络配置。因此，Kickstart **%pre** 部分中的一些任务被阻止。例如，因为网络配置不可用，从 **%pre** 部分中下载软件包会失败。

要临时解决这个问题：

- 配置网络，例如使用 **nmcli** 工具作为 **%pre** 脚本的一部分。
- 使用安装程序引导选项为 **%pre** 脚本配置网络。

因此，可以对 **%pre** 部分中的任务使用网络，Kickstart 安装过程完成。

Bugzilla:2173992

使用 RHEL 镜像构建器构建 rpm-ostree 镜像时，不支持启用 FIPS 模式

目前，在使用 RHEL 镜像构建器构建 **rpm-ostree** 镜像时，不支持启用 FIPS 模式。

Jira:RHEL-4655

使用 stig 配置文件补救的镜像构建无法引导，并显示 FIPS 错误

RHEL 镜像构建器不支持 FIPS 模式。当使用通过 **xccdf_org.ssgproject.content_profile_stig** 配置文件补救自定义的 RHEL 镜像构建器时，系统无法引导，并显示以下错误：

```
Warning: /boot//.vmlinuz-<kernel version>.x86_64.hmac does not exist
FATAL: FIPS integrity test failed
Refusing to continue
```

使用 **fips-mode-setup --enable** 命令进行系统镜像安装后手动启用 FIPS 策略无法正常工作，因为 **/boot** 目录在不同的分区上。如果禁用了 FIPS，则系统引导成功。目前，还没有可用的临时解决方案。

注意

您可以在使用 **fips-mode-setup --enable** 命令安装镜像后手动启用 FIPS。

Jira:RHEL-4649

驱动程序磁盘菜单无法在控制台上显示用户输入

当您在带有驱动程序磁盘的内核命令行上使用 **inst.dd** 选项启动 RHEL 安装时，控制台将无法显示用户输入。因此，应用程序似乎没有响应用户输入，并停止响应，但会显示使用户混淆的输出。但是，此行为不会影响功能，用户输入会在按 **Enter** 后被注册。

作为临时解决方案，要查看预期结果，请忽略控制台中缺少用户输入，并在完成添加输入后按 **Enter** 键。

Jira:RHEL-4737

11.2. 安全性

OpenSSL 不会检测 PKCS #11 令牌是否支持原始 RSA 或 RSA-PSS 签名的创建

TLS 1.3 协议需要支持 RSA-PSS 签名。如果 PKCS #11 令牌不支持原始 RSA 或 RSA-PSS 签名，如果密钥由 PKCS#11 令牌保存，则使用 OpenSSL 库的服务器应用程序将无法使用 RSA 密钥。因此，在上述场景中 TLS 通信会失败。

要临时解决这个问题，请配置服务器和客户端以使用 TLS 版本 1.2 作为可用最高 TLS 协议版本。

Bugzilla:1681178[1]

OpenSSL 错误处理 PKCS #11 tokens 不支持原始 RSA 或 RSA-PSS 签名

OpenSSL 库不会检测到 PKCS #11 令牌的与键相关的功能。因此，当使用不支持原始 RSA 或 RSA-PSS 签名的令牌创建签名时，建立 TLS 连接会失败。

要临时解决这个问题，请在 **/etc/pki/tls/openssl.cnf** 文件的 **crypto_policy** 部分的 **.include** 行后面添加以下行：

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

因此，可以在描述的场景中建立 TLS 连接。

Bugzilla:1685470[1]

使用特定语法，scp 会清空复制到其自身的文件

scp 实用程序从安全复制协议 (SCP) 改为更安全的 SSH 文件传输协议 (SFTP)。因此，将文件从位置复制到同一位置，从而擦除文件内容。此问题会产生以下语法：

scp localhost:/myfile localhost:/myfile

要临时解决这个问题，请不要使用这个语法将文件复制到与源位置相同的目标。

这个问题已针对以下语法解决：

- **scp /myfile localhost:/myfile**
- **scp localhost:~/myfile ~/myfile**

Bugzilla:2056884

OSCAP Anaconda 附加组件不会在图形安装中获取定制的配置文件

OSCAP Anaconda 附加组件不提供一个选项，来在 RHEL 图形安装中选择或取消选择安全配置文件的定制。从 RHEL 8.8 开始，当从存档或 RPM 软件包安装时，附加组件不会考虑定制。因此，安装会显示以下出错信息，而不是获取 OSCAP 定制的配置文件：

```
There was an unexpected problem with the supplied content.
```

要临时解决这个问题，您必须在 Kickstart 文件的 **%addon org_fedora_oscap** 部分中指定路径，例如：

```
xccdf-path = /usr/share/xml/scap/sc_tailoring/ds-combined.xml
tailoring-path = /usr/share/xml/scap/sc_tailoring/tailoring-xccdf.xml
```

因此，您只能将用于 SCAP 定制的配置文件的图形安装与相应的 Kickstart 规格一起使用。

Jira:RHEL-1824

Ansible 补救需要额外的集合

用 **ansible-core** 软件包替换 Ansible Engine 时，RHEL 订阅提供的 Ansible 模块的列表会减少。因此，运行使用包含在 **scap-security-guide** 软件包中的 Ansible 内容的补救需要来自 **rhc-worker-playbook** 软件包的集合。

对于 Ansible 补救，请执行以下步骤：

1. 安装所需的软件包：

```
# dnf install -y ansible-core scap-security-guide rhc-worker-playbook
```

2. 进到 **/usr/share/scap-security-guide/ansible** 目录：

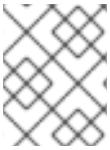
```
# cd /usr/share/scap-security-guide/ansible
```

3. 运行使用环境变量的相关 Ansible playbook，这些变量定义了到额外 Ansible 集合的路径：

```
# ANSIBLE_COLLECTIONS_PATH=/usr/share/rhc-worker-
playbook/ansible/collections/ansible_collections/ ansible-playbook -c local -i localhost, rhel9-
playbook-cis_server_l1.yml
```

将 ***cis_server_l1*** 替换为您要修复系统的配置文件的 ID。

因此，Ansible 内容会被正确处理。



注意

对 **rhc-worker-playbook** 中提供的集合的支持仅限于启用 **scap-security-guide** 中提供的 Ansible 内容。

Jira:RHEL-1800

Keylime 不接受串联的 PEM 证书

当 Keylime 将证书链作为 PEM 格式的、串联在一个文件中的多个证书接收时，**keylime-agent-rust** Keylime 组件在签名验证过程中不能正确地使用所有提供的证书，导致 TLS 握手失败。因此，客户端组件(**keylime_verifier** 和 **keylime_tenant**)无法连接到 Keylime 代理。要临时解决这个问题，请只使用一个证书而不是多个证书。

Jira:RHELPLAN-157225[1]

Keylime 拒绝其摘要以反斜杠开头的运行时策略

生成运行时策略的当前脚本 **create_runtime_policy.sh**，使用 SHA 校验函数，如 **sha256sum** ，来计算文件摘要。但是，当输入的文件名包含反斜杠或 **\n** 时，校验和函数会在其输出中的摘要前添加一个反斜杠。在这种情况下，生成的策略文件的格式不正确。提供错误格式的策略文件时，Keylime 租户会产生以下或类似错误消息：**me.tenant - ERROR - Response code 400: Runtime policy is malformat**。要临时解决这个问题，请通过输入以下命令从错误格式的策略文件中手动删除反斜杠：**sed -i 's/^\\//g' <malformed_file_name>**。

Jira:RHEL-11867[1]

更新后，Keylime 代理拒绝来自验证器的请求

当 Keylime 代理的 API 版本号(**keylime-agent-rust**)已更新时，代理会拒绝使用不同版本的请求。因此，如果 Keylime 代理被添加到验证器中，然后被更新，则验证器会尝试使用旧的 API 版本联系代理。代理拒绝此请求并使认证失败。要临时解决这个问题，请在更新代理(**keylime-agent-rust**)前更新验证器

(**keylime-verifier**) 。因此，当代理被更新时，verifier 会检测 API 更改，并相应地更新其存储的数据。

Jira:RHEL-1518[1]

fapolicyd 工具错误地允许执行更改的文件

在对文件进行任何更改后，文件的 IMA 哈希应该可以正确地更新，**fapolicyd** 应该阻止更改的文件的执行。但是，由于 IMA 策略设置和 **evctml** 进行哈希的文件中的区别，这不会发生。因此，IMA 哈希在更改的文件的扩展属性中不会被更新。因此，**fapolicyd** 错误地允许更改的文件的执行。

Jira:RHEL-520[1]

默认 SELinux 策略允许无限制的可执行文件使其堆栈可执行

SELinux 策略中的 **selinuxuser_execstack** 布尔值的默认状态是 on，这意味着无限制的可执行文件可以使其堆栈为可执行。可执行文件不应该使用这个选项，这通常代表开发的可执行代码的质量较差，或可能存在安全攻击的风险。但是，由于需要与其他工具、软件包和第三方产品保持兼容，红帽无法更改默认策略中的这个布尔值。如果您的环境没有此类兼容性问题，请使用 **setsebool -P selinuxuser_execstack off** 命令在您的本地策略中将这个布尔值设置为 off。

Bugzilla:2064274

STIG 配置文件中的 SSH 超时规则配置了不正确的选项

对 OpenSSH 的更新会影响以下 Defense Information Systems Agency Security Technical Implementation Guide (DISA STIG) 配置集中的规则：

- DISA STIG for RHEL 9 (**xccdf_org.ssgproject.content_profile_stig**)
- DISA STIG with GUI for RHEL 9 (**xccdf_org.ssgproject.content_profile_stig_gui**)

在每个配置集中，以下两条规则会受到影响：

```
Title: Set SSH Client Alive Count Max to zero
CCE Identifier: CCE-90271-8
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_keepalive_0

Title: Set SSH Idle Timeout Interval
CCE Identifier: CCE-90811-1
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_idle_timeout
```

当应用到 SSH 服务器时，每个规则都会配置一个选项（**ClientAliveCountMax** 和 **ClientAliveInterval**），其行为不再像之前一样。因此，当 OpenSSH 达到这些规则配置的超时时，OpenSSH 不再断开空闲的 SSH 用户。作为临时解决方案，这些规则已从 DISA STIG for RHEL 9 和 DISA STIG with GUI for RHEL 9 配置集中临时删除，直到开发出解决方案为止。

Bugzilla:2038978

GnuPG 错误地允许使用 SHA-1 签名，即使通过 crypto-policies 禁止使用 SHA-1 签名

无论系统范围的加密策略中定义的设置如何，GNU Privacy Guard(GnuPG)加密软件可以创建和验证使用 SHA-1 算法的签名。因此，您可以在 **DEFAULT** 加密策略中将 SHA-1 用于加密目的，这与这个不安全算法的系统范围弃用没有一致的。

要临时解决这个问题，请不要使用涉及 SHA-1 的 GnuPG 选项。因此，您将使用不安全的 SHA-1 签名来防止 GnuPG 降低默认的系统安全性。

Bugzilla:2070722

OpenSCAP 内存消耗问题

在内存有限的系统上，OpenSCAP 扫描程序可能过早停止，或者可能没有生成结果文件。要临时解决这个问题，您可以自定义扫描配置文件，以取消选择涉及递归整个 / 文件系统的规则：

- **rpm_verify_hashes**
- **rpm_verify_permissions**
- **rpm_verify_ownership**
- **file_permissions_unauthorized_world_writable**
- **no_files_unowned_by_user**
- **dir_perms_world_writable_system_owned**
- **file_permissions_unauthorized_suid**
- **file_permissions_unauthorized_sgid**
- **file_permissions_ungroupowned**
- **dir_perms_world_writable_sticky_bits**

如需了解更多详细信息和临时解决方案，请参阅相关的 知识库文章。

Bugzilla:2161499

在 kickstart 安装过程中修复与服务相关的规则可能会失败

在 kickstart 安装过程中，OpenSCAP 工具有时会错误地显示服务的 **enable** 或**disable** 状态补救不需要。因此，OpenSCAP 可能会将安装的系统上的服务设置为不合规的状态。作为临时解决方案，您可以在 kickstart 安装后扫描并修复该系统。这可以解决与服务相关的问题。

BZ#1834716

11.3. RHEL FOR EDGE

edge-vsphere 镜像中没有 open-vm-tools 软件包

目前，在 **edge-vsphere** 镜像中不会默认安装 **open-vm-tools** 软件包。要临时解决这个问题，请在蓝图自定义中包含软件包。使用 **edge-vsphere** 镜像类型时，请在 RHEL for Edge 容器镜像或 RHEL for Edge 提交镜像的蓝图中添加 **open-vm-tools**。

Jira:RHELDOCS-16574[1]

11.4. 软件管理

安装过程有时将变为无响应

安装 RHEL 时，安装过程有时会变得无响应。**/tmp/packaging.log** 文件在末尾显示以下消息：

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

要解决这个问题，重启安装过程。

Bugzilla:2073510

在本地仓库上运行 createrepo_c 会产生重复的 repodata 文件

当您在本地存储库上运行 **createrepo_c** 命令时，它会产生 **repodata** 文件的重复副本，其中一个副本是压缩的，另一个副本不是。但是，没有可用的临时解决方案，您可以安全地忽略重复的文件。**createrepo_c** 命令会产生重复的副本，因为其他工具中的要求和差异依赖于使用 **createrepo_c** 创建的存储库。

Bugzilla:2056318

11.5. SHELL 和命令行工具

如果在配置文件中设置了 TMPDIR 变量，则 ReaR 在恢复过程中失败

在 **/etc/rear/local.conf** 或 **/etc/rear/site.conf** ReaR 配置文件中设置和导出 **TMPDIR** 无法工作，且已弃用。

ReaR 默认配置文件 **/usr/share/rear/conf/default.conf** 包含以下说明：

```
# To have a specific working area directory prefix for Relax-and-Recover
# specify in /etc/rear/local.conf something like
#
# export TMPDIR="/prefix/for/rear/working/directory"
#
# where /prefix/for/rear/working/directory must already exist.
# This is useful for example when there is not sufficient free space
# in /tmp or $TMPDIR for the ISO image or even the backup archive.
```

上述说明无法正常工作，因为 **TMPDIR** 变量在救援环境中有相同的值，如果 **TMPDIR** 变量中指定的目录在救援镜像中不存在，则这是不正确的。

因此，在 **/etc/rear/local.conf** 文件中设置和导出 **TMPDIR** 在救援镜像引导时导致以下错误：

```
mktemp: failed to create file via template '/prefix/for/rear/working/directory/tmp.XXXXXXXXXX': No
such file or directory
cp: missing destination file operand after '/etc/rear/mappings/mac'
Try 'cp --help' for more information.
No network interface mapping is specified in /etc/rear/mappings/mac
```

或者，在运行 **rear recover** 时导致以下错误，并稍后中止：

```
ERROR: Could not create build area
```

要临时解决这个问题，如果您想有一个自定义临时目录，请在执行 ReaR 环境前在 shell 环境中导出变量，来为 ReaR 临时文件指定一个自定义目录。例如，执行 **export TMPDIR=...** 语句，然后在同一 shell 会话或脚本中执行 **rear** 命令。因此，在上述配置中，恢复成功。

Jira:RHEL-24847

使用 ifcfg 文件重命名网络接口失败

在 RHEL 9 中，默认情况下不会安装 **initscripts** 软件包。因此，使用 **ifcfg** 文件重命名网络接口会失败。要解决这个问题，红帽建议您使用 **udev** 规则或链接文件来重命名接口。详情请查看 Consistent 网络接口设备命名 和 **systemd.link(5)** man page。

如果您无法使用推荐的解决方案之一，请安装 **initscripts** 软件包。

Bugzilla:2018112[1]

RHEL 9 中不默认安装 chkconfig 软件包

RHEL 9 中不默认安装 **chkconfig** 软件包（更新和查询系统服务运行级别信息）。

要管理服务，请使用 **systemctl** 命令或手动安装 **chkconfig** 软件包。

有关 **systemd** 的更多信息，请参阅 systemd 简介。有关如何使用 **systemctl** 实用程序的步骤，请参阅使用 systemctl 管理系统服务。

Bugzilla:2053598[1]

设置控制台 keymap 在最小安装上需要 libxkbcommon 库

在 RHEL 9 中，某些 **systemd** 库依赖项已从动态链接转换为动态加载，以便您的系统在运行时打开并使用库（当它们可用时）。有了这个更改，除非您安装必要的库，否则无法使用依赖于此类库的功能。这也会影响在最小安装的系统上设置键盘布局。因此，**localectl --no-convert set-x11-keymap gb** 命令会失败。

要临时解决这个问题，请安装 **libxkbcommon** 库：

```
# dnf install libxkbcommon
```

Jira:RHEL-6105

sysstat 软件包中的 %vmeff 指标显示不正确的值

sysstat 软件包提供 **%vmeff** 指标来测量页面回收效率。**sar -B** 命令返回的 **%vmeff** 列的值不正确，因为 **sysstat** 不会解析后续内核版本提供的所有相关的 **/proc/vmstat** 值。要临时解决这个问题，您可以从 **/proc/vmstat** 文件中手动计算 **%vmeff** 值。详情请查看 为什么在 RHEL 8 和 RHEL 9 中 **sar (1)** 工具报告 **%vmeff** 值超过 100 % ？

Jira:RHEL-12009

服务位置协议(SLP)易受到通过 UDP 的攻击

OpenSLP 为本地区域网络中的应用程序提供动态配置机制，如打印机和文件服务器。但是，SLP 会受到通过连接到互联网的系统上的 UDP 的反射性拒绝服务放大攻击。SLP 允许未经身份验证的攻击者注册新服务，而不受由 SLP 实现设置的限制。通过使用 UDP 和欺骗源地址，攻击者可以请求服务列表，在欺骗地址上创建拒绝服务。

要防止外部攻击者访问 SLP 服务，请在不受信任的网络上运行的所有系统上禁用 SLP，比如那些直接连接到互联网的系统。另外，要解决这个问题，请配置防火墙以阻止或过滤 UDP 和 TCP 端口 427 上的流量。

Jira:RHEL-6995[1]

11.6. 基础架构服务

bind 和 unbound 都禁用基于 SHA-1- 的签名验证

bind 和 **unbound** 组件禁用所有 RSA/SHA1（算法 5）和 RSASHA1-NSEC3-SHA1（算法号 7）签名，且签名的 SHA-1 用法在 DEFAULT 系统范围的加密策略中受到限制。

因此，某些 DNSSEC 记录使用 SHA-1、RSA/SHA1 和 RSASHA1-NSEC3-SHA1 摘要算法无法验证在 Red Hat Enterprise Linux 9 中，受影响的域名会存在安全漏洞。

要临时解决这个问题，升级到不同的签名算法，如 RSA/SHA-256 或 elliptic curve 键。

有关受影响和存在安全漏洞的顶级域的信息和列表，请参阅使用 RSASHA1 签名的 DNSSEC 记录失败来验证 解决方案。

Bugzilla:2070495

如果在多个区域中使用相同的可写区域文件，named 无法启动

BIND 不允许在多个区域中具有相同的可写区域文件。因此，如果配置包含多个区域，它们共享到可由 **named** 服务修改的文件的路径，则 **named** 无法启动。要临时解决这个问题，请使用 **in-view** 子句在多个视图间共享一个区域，并确保为不同的区使用不同的路径。例如，在路径中包含视图名称。

请注意，可写的区域文件通常在带有允许的动态更新的区域、DNSSEC 维护的次要区域或区域中使用。

Bugzilla:1984982

libotr 与 FIPS 不兼容

libotr 库和非记录(OTR)消息的工具包为即时消息会话提供了端到端加密。但是，由于其使用了 **gcry_pk_sign()** 和 **gcry_pk_verify()** 函数，**libotr** 库不符合联邦信息处理标准(FIPS)。因此，您无法在 FIPS 模式下使用 **libotr** 库。

Bugzilla:2086562

11.7. 网络

使用带有 mlx5 驱动程序的 XDP 多缓冲模式和大于 3498 字节的 MTU 需要禁用 RX Striding RQ

在与以下所有条件匹配的主机上运行带有多缓冲区模式的 eXpress Data Path (XDP)脚本失败：

- 主机使用 **mlx5** 驱动程序。
- 最大传输单元(MTU)值大于 3498 字节。
- 接收跨步接收队列(RX Striding RQ)功能已在 Mellanox 接口上启用。

如果所有条件都适用，脚本会失败，并显示 **link set xdp fd failed** 错误。要在具有较高 MTU 的主机上运行 XDP 脚本，请在 Mellanox 接口上禁用 RX Striding RQ：

```
# ethtool --set-priv-flags <interface_name> rx_striding_rq off
```

因此，您可以在使用 **mlx5** 驱动程序和有大于 3498 字节的 MTU 值的接口上使用 XDP 多缓冲区模式。

Jira:RHEL-6496[1]

KTLS 不支持将 TLS 1.3 卸载到 NIC

内核传输层安全(kTLS)不支持将 TLS 1.3 卸载到 NIC。因此，即使 NIC 支持 TLS 卸载，软件加密也会与 TLS 1.3 一起使用。要临时解决这个问题，如果需要卸载，禁用 TLS 1.3。因此，您只能卸载 TLS 1.2。当使用 TLS 1.3 时，性能较低，因为无法卸载 TLS 1.3。

Bugzilla:2000616[1]

更新会话密钥失败会导致连接中断

内核传输层安全(kTLS)协议不支持更新会话密钥，这些密钥由对称密码使用。因此，用户无法更新密钥，从而导致连接中断。要临时解决这个问题，请禁用 kTLS。因此，解决这一问题，可以成功更新会话密钥。

Bugzilla:2013650[1]

默认情况下不安装 initscripts 软件包

默认情况下，不会安装 **initscripts** 软件包。因此，**ifup** 和 **ifdown** 工具不可用。一个替代的方法是，可以使用 **nmcli connection up** 和 **nmcli connection down** 命令来启用和禁用连接。如果这个替代方法无法正常工作，请报告这个问题并安装 **NetworkManager-initscripts-updown** 软件包，该软件包为 **ifup** 和 **ifdown** 工具提供了一个 NetworkManager 解决方案。

Bugzilla:2082303

使用 Mellanox ConnectX-5 适配器时， mlx5 驱动程序会失败

在以太网交换机设备驱动程序型号(**switchdev**)模式下，当使用设备管理的流控制(DMFS)参数和 **ConnectX-5** 适配器支持的硬件配置时，**mlx5** 驱动程序会失败。因此，您可以看到以下错误信息：

```
BUG: Bad page cache in process umount pfn:142b4b
```

要临时解决这个问题，请使用软件管理的流控制 (SMFS)参数，而不是 DMFS。

Jira:RHEL-9897[1]

Intel® i40e 适配器在 IBM Power10 上永久失败

当 **i40e** 适配器在 IBM Power10 系统上遇到 I/O 错误时，增强的 I/O 错误处理(EEH)内核服务会触发网络驱动程序的重置和恢复。 但是，EEH 重复报告 I/O 错误，直到 **i40e** 驱动程序达到预定义的最大 EEH 冻结为止。因此，EEH 会导致设备永久失败。

Jira:RHEL-15404[1]

xdp-loader features 命令失败

xdp-loader 工具是针对之前版本的 **libbpf** 编译的。因此，**xdp-loader features** 命令会失败，并显示错误：

```
Cannot display features, because xdp-loader was compiled against an old version of libbpf without support for querying features.
```

没有可用的临时解决方案。因此，您无法使用 **xdp-loader features** 命令来显示接口功能。

Jira:RHEL-3382[1]

11.8. 内核

内核中的 kdump 机制会引起 64K 内核的 OOM 错误

64 位 ARM 架构上的 64K 内核页大小使用的内存比 4KB 内核使用的多。因此，**kdump** 会导致内核 panic，内存分配失败，并报内存不足(OOM)错误。作为临时解决方案，手动将 **crashkernel** 的值配为 640 MB。例如，将 **crashkernel=** 参数设为 **crashkernel=2G- :640M**。

因此，**kdump** 机制对上述场景的 64K 内核不会失败。

Bugzilla:2160676[1]

当从 4k 迁移到 64k 页大小内核时，依赖内核页大小的客户应用程序可能需要更新

RHEL 与 4k 和 64k 页大小内核都兼容。当从 4k 迁移到 64k 页大小内核时，依赖 4k 内核页大小的客户应用程序可能需要更新。已知的实例包括 **jemalloc** 和依赖的应用程序。

jemalloc 内存分配器库对系统运行时环境中使用的页大小敏感。库可以构建成与 4k 和 64k 页大小内核兼容，例如，当使用 **--with-lg-page=16** 或 **env JEMALLOC_SYS_WITH_LG_PAGE=16** 配置时（用于 **jemallocator** Rust crate）。因此，运行时环境的页大小与编译依赖于 **jemalloc** 的二进制文件时出现的页大小之间可能会出现不匹配。因此，使用基于 **jemalloc** 的应用程序会触发以下错误：

```
<jemalloc>: Unsupported system page size
```

要避免这个问题，请使用以下方法之一：

- 使用合适的构建配置或环境选项来创建 4k 和 64k 页大小兼容二进制文件。
- 在引导到最后的 64k 内核和运行时环境后，构建任何使用 **jemalloc** 的用户空间软件包。

例如，您可以构建 **fd-find** 工具，该工具也通过 **cargo** Rust 软件包管理器使用 **jemalloc**。在最后的 64k 环境中，输入 **cargo** 命令触发所有依赖项的新构建，以解决页大小中的不匹配：

```
# cargo install fd-find --force
```

Bugzilla:2167783[1]

使用 dnf 升级到最新的实时内核不会并行安装多个内核版本

使用 **dnf** 软件包管理器安装最新的实时内核需要解决软件包依赖，来同时保留新的和当前的内核版本。默认情况下，**dnf** 在升级过程中删除旧的 **kernel-rt** 软件包。

作为临时解决方案，将当前的 **kernel-rt** 软件包添加到 **/etc/yum.conf** 配置文件中的 **installonlypkgs** 选项中，例如 **installonlypkgs=kernel-rt**。

installonlypkgs 选项将 **kernel-rt** 附加到 **dnf** 使用的默认列表中。**installonlypkgs** 指令中列出的软件包不会被自动删除，因此支持多个内核版本来同时安装。

请注意，安装了多个内核是一种在使用新内核版本时具有回退选项的方法。

Bugzilla:2181571[1]

默认情况下，Delay Accounting 功能不会显示 SWAPIN 和 IO% 统计列

Delayed Accounting 功能与早期版本不同，它们会被默认禁用。因此，**iotop** 应用程序不显示 **SWAPIN** 和 **IO%** 统计列，并显示以下警告：

```
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO%
```

Delay Account 功能使用 **taskstats** 接口，为属于线程组的所有任务或线程提供延迟统计。当任务等待 kernel 资源可用时，会延迟执行，例如：等待空闲 CPU 运行的任务。统计有助于设置任务的 CPU 优先级、I/O 优先级和 **rss** 限制值。

作为临时解决方案，您可以在运行时或引导时启用 **delayacct** 引导选项。

- 要在运行时启用 **delayacct**，请输入：

  ```
  echo 1 > /proc/sys/kernel/task_delayacct
  ```

  请注意，这个命令可启用系统范围功能，但只适用于您在运行此命令后启动的任务。

- 要在引导时永久启用 **delayacct**，请使用以下步骤之一：
  - 编辑 **/etc/sysctl.conf** 文件以覆盖默认参数：
    a. 在 **/etc/sysctl.conf** 文件中添加以下条目：

    ```
    kernel.task_delayacct = 1
    ```

    如需更多信息，请参阅 如何在 Red Hat Enterprise Linux 上设置 sysctl 变量 。

    b. 重启系统以使更改生效。
  - 在内核命令行中添加 **delayacct** 选项。
    如需更多信息，请参阅 配置内核命令行参数。

因此，**iotop** 应用程序会显示 **SWAPIN** 和 **IO%** 统计列。

Bugzilla:2132480[1]

在具有大型核数的系统上实时内核的硬件认证可能需要传递 skew-tick=1 引导参数

具有大量插槽和大核数的大型或中型系统可能会因为对 **xtime_lock**（其在计时系统中使用）的锁争用而遇到延迟峰值。因此，硬件认证中的延迟峰值和延迟可能会在多处理系统上发生。作为临时解决方案，您可以通过添加 **skew_tick=1** 引导参数，偏移每个 CPU 的计时器刻度，来在不同的时间启动。

要避免锁冲突，请启用 **skew_tick=1** ：

1. 使用 **grubby** 启用 **skew_tick=1** 参数。

   ```
   # grubby --update-kernel=ALL --args="skew_tick=1"
   ```

2. 重启以使更改生效。
3. 通过显示您在启动过程中传递的内核参数来验证新设置。

   ```
   cat /proc/cmdline
   ```

请注意，启用 **skew_tick=1** 会导致功耗的大量增加，因此只有在运行延迟敏感实时工作负载时才必须启用它。

Jira:RHEL-9318[1]

kdump 机制无法捕获 LUKS 加密目标上的 vmcore 文件

当在使用 Linux Unified Key Setup(LUKS)加密分区的系统中运行 **kdump** 时，系统需要特定的可用内存。当可用内存小于所需内存量时，**systemd-cryptsetup** 服务将无法挂载分区。因此，第二个内核无法捕获 LUKS 加密目标上的崩溃转储文件。

作为临时解决方案，查询 **推荐的 crashkernel 值** ，并逐渐将内存大小增加到合适的值。**推荐的 crashkernel 值** 可作为设置所需内存大小的参考。

1. 打印估计的崩溃内核值。

```
# kdumpctl estimate
```

2. 通过增加 **crashkernel** 值来配置所需的内存量。

```
# grubby --args=crashkernel=652M --update-kernel=ALL
```

3. 重启系统以使更改生效。

```
# reboot
```

因此，**kdump** 在带有 LUKS 加密分区的系统上可以正常工作。

Jira:RHEL-11196[1]

kdump 服务无法在 IBM Z 系统中构建 initrd 文件

在 64 位 IBM Z 系统中，当 **znet** 相关配置信息（如 **s390-subchannels** ）位于不活跃 **NetworkManager** 连接配置集时，**kdump** 服务无法加载初始 RAM 磁盘 (**initrd**)。因此，**kdump** 机制会失败并显示以下错误：

```
dracut: Failed to set up znet
kdump: mkdumprd: failed to make kdump initrd
```

作为临时解决方案，请使用以下解决方案之一：

- 通过重新使用具有 **znet** 配置信息的连接配置集来配置网络绑定或桥接：

```
$ nmcli connection modify enc600 master bond0 slave-type bond
```

- 将 **znet** 配置信息从不活跃连接配置集复制到活跃连接配置集中：

  a. 运行 **nmcli** 命令查询 **NetworkManager** 连接配置集：

```
# nmcli connection show

NAME                     UUID              TYPE    Device

bridge-br0           ed391a43-bdea-4170-b8a2 bridge    br0
bridge-slave-enc600  caf7f770-1e55-4126-a2f4 ethernet enc600
enc600               bc293b8d-ef1e-45f6-bad1 ethernet --
```

  b. 使用不活跃连接中的配置信息更新活跃的配置集：

```
#!/bin/bash
 inactive_connection=enc600
```

```
active_connection=bridge-slave-enc600
for name in nettype subchannels options; do
field=802-3-ethernet.s390-$name
val=$(nmcli --get-values "$field"connection show "$inactive_connection")
nmcli connection modify "$active_connection" "$field" $val"
done
```

c. 重启 **kdump** 服务以使更改生效：

```
# kdumpctl restart
```

Bugzilla:2064708

iwl7260-firmware 破坏了 Intel Wi-Fi 6 AX200、AX210 和 Lenovo ThinkPad P1 Gen 4 上的 Wi-Fi

在将 **iwl7260-firmware** 或 **iwl7260-wifi** 驱动程序更新到 RHEL 9.1 及之后的版本提供的版本后，硬件会进入不正确的内部状态。错误地报告其状态。因此，Intel Wifi 6 卡可能无法正常工作，并显示错误信息：

```
kernel: iwlwifi 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlwifi 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlwifi 0000:09:00.0: Failed to run INIT ucode: -110
```

未确认的临时解决方法是关闭系统并再次打开。不要重启。

Bugzilla:2129288[1]

kmod 中的 weak-modules 不能与模块间依赖一起工作

kmod 软件包提供的 **weak-modules** 脚本决定了哪些模块与安装的内核 kABI 兼容。但是，在检查模块的内核兼容性时，**weak-modules** 按照构建它们的内核的从高到低版本来处理模块符号依赖项。因此，针对不同内核版本构建的具有相互依赖关系的模块可能会被解释为不兼容，因此 **weak-modules** 脚本不能在此场景下工作。

要临时解决这个问题，请在安装新内核前针对最新的库存内核进行构建或放置额外的模块。

Bugzilla:2103605[1]

dkms 对 64 位 ARM CPU 上正确编译的驱动程序提供了一条有关程序失败的不正确的警告

动态内核模块支持(**dkms**)工具无法识别用于 4 KB 内核和 64 KB 页大小的内核 64 位 ARM CPU 的内核标头。因此，当执行内核更新且 **kernel-64k-devel** 软件包未安装时，**dkms** 会提供一条有关为什么程序在正确编译的驱动程序上失败的错误警告。要临时解决这个问题，请安装 **kernel-headers** 软件包，其包含两种类型的 ARM CPU 架构的头文件，且不特定于 **dkms** 及其要求。

JIRA:RHEL-25967[1]

11.9. 文件系统和存储

在出现不成功的 CHAP 验证尝试后，Anaconda 无法使用 no authentication 方法登录 iSCSI 服务器

当您使用 CHAP 身份验证添加 iSCSI 磁盘时，如果因为凭证不正确而导致登录失败，使用 **no authentication** 方法尝试重新登录也将失败。要解决这个问题，请先关闭当前会话，再使用 **no authentication** 方法登录。

Bugzilla:1983602[1]

NVMe/TCP 不支持设备映射器多路径

使用带有 **nvme-tcp** 驱动程序的设备映射器多路径可能会导致 Call Trace 警告和系统不稳定。要临时解决这个问题，NVMe/TCP 用户必须启用原生 NVMe 多路径，且不能在 NVMe 中使用 **device-mapper-multipath** 工具。

默认情况下，RHEL 9 中启用了原生 NVMe 多路径。如需更多信息，请参阅在 NVMe 设备 上启用多路径。

Bugzilla:2033080[1]

blk-availability systemd 服务停用了复杂的设备堆栈

在 **systemd** 中，默认的块停用代码并不总是正确处理虚拟块设备的复杂堆栈。在一些配置中，虚拟设备在关闭过程中可能无法被删除，这会导致记录错误信息。要临时解决这个问题，请执行以下命令来停用复杂块设备堆栈：

```
# systemctl enable --now blk-availability.service
```

因此，复杂虚拟设备堆栈会在关闭过程中被正确停用，且不会生成错误消息。

Bugzilla:2011699[1]

对于启用了配额的情况下挂载的 XFS 文件系统，不再可能禁用配额记帐

从 RHEL 9.2 开始，无法在已挂载的启用了配额的 XFS 文件系统上禁用配额记帐。

要临时解决这个问题，请去掉配额选项，重新挂载文件系统来禁用配额记账。

Bugzilla:2160619[1]

对 NVMe 设备的 udev 规则更改

对 NVMe 设备有一个 udev 规则更改，即添加了 **OPTIONS="string_escape=replace"** 参数。如果您设备的序列号前面有空格，则这会导致某些厂商的对按 id 命名的磁盘进行更改。

Bugzilla:2185048

不能在 Kickstart 文件中可靠地使用 NVMe/FC 设备

在解析或执行 Kickstart 文件的预脚本时，NVMe/FC 设备可能不使用，这会导致 Kickstart 安装失败。要临时解决这个问题，将引导参数更新为 **inst.wait_for_disks=30**。这个选项会导致 30 秒的延迟，应为 NVMe/FC 设备提供充足的时间进行连接。使用这个临时解决方案以及及时连接的 NVMe/FC 设备，Kickstart 安装可以正常进行。

Jira:RHEL-8164[1]

在使用 qedi 驱动程序时内核 panic

在使用 **qedi** iSCSI 驱动程序时，操作系统引导后内核 panics。要临时解决这个问题，请向内核引导命令行中添加 **kfence.sample_interval=0** 来禁用 **kfence** 运行时内存错误检测器功能。

Jira:RHEL-8466[1]

无法引导具有内核 64k 页大小的基于 ARM 的系统

安装 **vdo** 软件包时，4k 页大小的内核被作为依赖项安装。因此，即使在 **Software Selection** 屏幕上选择了 64k 页大小，系统也会使用 4k 页大小引导。要临时解决这个问题，在 **Base Environment** 下选择 **Minimal Install**，并在 **Kernel options** 下选择 64k 作为页大小。当系统首次启动时，使用 DNF 软件包管理器安装其他软件。

Jira:RHEL-8354

11.10. 动态编程语言、网页和数据库服务器

python3.11-lxml 不提供 lxml.isoschematron 子模块

python3.11-lxml 软件包不与 **lxml.isoschematron** 子模块一起分发，因为它不是开源许可证。子模块实现 ISO 架构支持。作为替代方案，**lxml.etree.Schematron** 类种提供了 pre-ISO-Schematron 验证。**python3.11-lxml** 软件包的其余内容不受影响。

Bugzilla:2157708

MySQL 和 MariaDB 中的 --ssl-fips-mode 选项不会改变 FIPS 模式

RHEL 中 **MySQL** 和 **MariaDB** 中的 **--ssl-fips-mode** 选项与上游中的工作方式不同。

在 RHEL 9 中，如果您使用 **--ssl-fips-mode** 作为 **mysqld** 或 **mariadbd** 守护进程的参数，或者在 **MySQL** 或 **MariaDB** 服务器配置文件中使用 **ssl-fips-mode**，则 **--ssl-fips-mode** 不会更改这些数据库服务器的 FIPS 模式。

相反：

- 如果将 **--ssl-fips-mode** 设为 **ON**，则 **mysqld** 或 **mariadbd** 服务器守护进程不会启动。
- 如果您在启用了 FIPS 的系统上将 **--ssl-fips-mode** 设为 **OFF**，则 **mysqld** 或 **mariadbd** 服务器守护进程仍然在 FIPS 模式下运行。

这是预期的，因为应该为整个 RHEL 系统启用或禁用 FIPS 模式，而不是为特定组件。

因此，不要在 RHEL 中的 **MySQL** 或 **MariaDB** 中使用 **--ssl-fips-mode** 选项。相反，请确保在整个 RHEL 系统上启用 FIPS 模式：

- 最好安装启用了 FIPS 模式的 RHEL。在安装过程中启用 FIPS 模式可确保系统使用 FIPS 批准的算法生成所有的密钥，并持续监控测试。有关在 FIPS 模式下安装 RHEL 的详情，请参考 在 FIPS 模式下安装系统。
- 或者，您可以按照 将系统切换成 FIPS 模式 中的流程，为整个 RHEL 系统切换 FIPS 模式。

Bugzilla:1991500

11.11. 身份管理

MIT Kerberos 不支持 PKINIT 的 ECC 证书

MIT Kerberos 不对评论文档实施 RFC5349 请求，它描述了公钥 Cryptography 中的 elliptic-curve 加密 (ECC) 支持。因此，RHEL 使用的 MIT **krb5-pkinit** 软件包不支持 ECC 证书。如需更多信息，请参阅 Kerberos (PKINIT)对公共密钥加密加密支持(ECC) 支持。

Jira:RHEL-4902

必须在 RHEL 9 客户端上设置 DEFAULT:SHA1 子策略，以使 PKINIT 能够针对 AD KDC 工作

现在，RHEL 9 中弃用了 SHA-1 摘要算法，对公共密钥 Cryptography for Public Key Cryptography 的 CMS 消息使用更强大的 SHA-256 算法签名。

但是，Active Directory (AD) Kerberos Distribution Center (KDC) 仍然使用 SHA-1 摘要算法为 CMS 信息签名。因此，RHEL 9 Kerberos 客户端无法通过对 AD KDC 使用 PKINIT 来验证用户。

要临时解决这个问题，使用以下命令在 RHEL 9 系统上启用对 SHA-1 算法的支持：

```
# update-crypto-policies --set DEFAULT:SHA1
```

Bugzilla:2060798

如果 RHEL 9 Kerberos 代理与非 RHEL-9 和非AD Kerberos 代理通信，则用户的 PKINIT 身份验证会失败

如果 RHEL 9 Kerberos 代理（客户端或 Kerberos 分发中心(KDC) 与不是 Active Directory (AD) 代理的非 RHEL-9 Kerberos 代理交互，则用户的 PKINIT 身份验证会失败。要临时解决这个问题，请执行以下操作之一：

- 将 RHEL 9 代理的 crypto-policy 设置为 **DEFAULT:SHA1** 以允许验证 SHA-1 签名：

  ```
  # update-crypto-policies --set DEFAULT:SHA1
  ```

- 更新非 RHEL-9 和非 AD 代理，以确保它不使用 SHA-1 算法为 CMS 数据签名。因此，将您的 Kerberos 客户端或 KDC 软件包更新至使用 SHA-256 而不是 SHA-1 的版本：
  - CentOS 9 Stream: krb5-1.19.1-15
  - RHEL 8.7: krb5-1.18.2-17
  - RHEL 7.9: krb5-1.15.1-53
  - Fedora Rawhide/36: krb5-1.19.2-7
  - Fedora 35/34：krb5-1.19.2-3

因此，用户的 PKINIT 身份验证可以正常工作。

请注意，对于其他操作系统，这是 krb5-1.20 版本，可确保代理使用 SHA-256 而不是 SHA-1 为 CMS 数据进行签名。

另请参阅 必须在 RHEL 9 客户端上设置 DEFAULT:SHA1 子策略，以使 PKINIT 能够针对 AD KDC 工作 。

Jira:RHEL-4875

AD 信任的 FIPS 支持需要 AD-SUPPORT 加密子策略

Active Directory(AD)使用 AES SHA-1 HMAC 加密类型，默认情况下在 RHEL 9 上不允许 FIPS 模式。如果要使用带有 AD 信任的 RHEL 9 IdM 主机，请在安装 IdM 软件前支持 AES SHA-1 HMAC 加密类型。

由于 FIPS 合规性是一个涉及技术和机构协议的过程，因此，请在启用 **AD-SUPPORT** 子策略前咨询 FIPS 审核员，以允许采取技术措施支持 AES SHA-1 HMAC 加密类型，然后安装 RHEL IdM：

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

Bugzilla:2057471

Heimdal 客户端无法针对 RHEL 9 KDC 使用 PKINIT 来验证用户

默认情况下，Heimdal Kerberos 客户端通过使用 Modular Exponential (MODP) Diffie-Hellman Group 2 用于互联网密钥交换 (IKE) 启动 IdM 用户的 PKINIT 身份验证。但是，RHEL 9 上的 MIT Kerberos 分配中心 (KDC) 仅支持 MODP 组 14 和 16。

因此，pre-autentication 请求会失败并显示 **krb5_get_init_creds: PREAUTH_FAILED** 错误，在 RHEL MIT KDC 中 **不接受 Key 参数**。

要临时解决这个问题，请确保 Heimdal 客户端使用 MODP Group 14。将客户端配置文件的 **libdefaults** 部分中的 **pkinit_dh_min_bits** 参数设置为 1759：

```
[libdefaults]
pkinit_dh_min_bits = 1759
```

因此，Heimdal 客户端可以针对 RHEL MIT KDC 完成 PKINIT 预验证。

Jira:RHEL-4889

FIPS 模式下的 IdM 不支持使用 NTLMSSP 协议来建立双向跨林信任

在活动目录(AD)和启用了 FIPS 模式的身份管理(IdM)之间建立双向跨林信任会失败，因为新技术局域网管理器安全支持提供程序 (NTLMSSP)身份验证不符合 FIPS。FIPS 模式下的 IdM 不接受在尝试验证时 AD 域控制器使用的 RC4 NTLM 哈希。

Jira:RHEL-12154[1]

IdM Vault 加密和解密在 FIPS 模式下失败

如果启用了 FIPS 模式，则 OpenSSL RSA-PKCS1v15 填充加密会被阻止。因此，身份管理(IdM) Vault 无法正常工作，因为 IdM 目前正在使用 PKCS1v15 填充来用传输证书包装会话密钥。

Jira:RHEL-12143[1]

升级后，没有 SID 的用户无法登录到 IdM

将 IdM 副本升级到 RHEL 9.2 后，IdM Kerberos 分发中心(KDC)可能无法向没有分配给其帐户安全标识符(SID)的用户发出票据授予票(TGT)。因此，用户无法登录到其帐户。

要临时解决这个问题，请以 IdM 管理员身份在拓扑中的另一个 IdM 副本上运行以下命令来生成 SID：

```
# ipa config-mod --enable-sid --add-sids
```

之后，如果用户仍然无法登录，请检查目录服务器错误日志。您可能需要调整 ID 范围使其包含用户 POSIX 身份。

如需更多信息，请参阅 升级到 RHEL9 时，IDM 用户无法再登录 知识库解决方案。

Jira:RHELPLAN-157939[1]

迁移的 IdM 用户可能会因为不匹配的域 SID 而无法登录

如果您使用 **ipa migrate-ds** 脚本将用户从一个 IdM 部署迁移到另一个，则这些用户可能会在使用 IdM 服务时有问题，因为它们之前存在的安全标识符(SID)没有当前 IdM 环境的域 SID。例如，这些用户可以使用 **kinit** 工具检索 Kerberos 票据，但不能登录。要临时解决这个问题，请参阅以下知识库文章：Migrated IdM 用户因为不匹配的域 SID 而无法登录 。

Jira:RHELPLAN-109613[1]

由于生成用户 PAC 的加密类型不兼容，MIT krb5 用户无法获取 AD TGT

在 MIT **krb5 1.20** 以及后续的软件包中，默认在所有 Kerberos 票据中都包括特权属性证书(PAC)。MIT Kerberos 分发中心(KDC)选择可用的最强的加密类型，来在 PAC 中生成 KDC 校验和，这目前是 RFC8009 中定义的 **AES HMAC-SHA2** 加密类型。但是，活动目录(AD)不支持这个 RFC。因此，在 AD-MIT 跨领域设置中，MIT **krb5** 用户无法获取 AD 票据授予票据(TGT)，因为 MIT KDC 生成的跨领域 TGT 在 PAC 中包含不兼容的 KDC 校验和类型。

要临时解决这个问题，对于 **/var/kerberos/krb5kdc/kdc.conf** 配置文件的 **[realms]** 部分中的 MIT 领域，请将 **disable_pac** 参数设为 **true**。因此，MIT KDC 会生成没有 PAC 的票据，这意味着 AD 会跳过失败的校验和验证，MIT **krb5** 用户可以获取 AD TGT。

Bugzilla:2016312

对 ldap_id_use_start_tls 选项使用默认值时的潜在风险

当使用没有 TLS 的 **ldap://** 进行身份查找时，可能会对攻击向量构成风险。特别是中间人(MITM)攻击，例如，攻击者可以通过更改 LDAP 搜索中返回的对象的 UID 或 GID 来冒充用户。

目前，强制 TLS 的 SSSD 配置选项 **ldap_id_use_start_tls** 默认为 **false**。确保您的设置在可信环境中操作，并决定对 **id_provider = ldap** 使用未加密的通信是否是安全的。注意 **id_provider = ad** 和 **id_provider = ipa** 不受影响，因为它们使用 SASL 和 GSSAPI 保护的加密连接。

如果使用未加密的通信不安全，请在 **/etc/sssd/sssd.conf** 文件中将 **ldap_id_use_start_tls** 选项设置为 **true** 来强制使用 TLS。计划在以后的 RHEL 版本中更改默认行为。

Jira:RHELPLAN-155168[1]

将 FIPS 模式下的 RHEL 9 副本添加到用 RHEL 8.6 或更早版本初始化的 FIPS 模式下的 IdM 部署会失败

略旨在遵守 FIPS 140-3 的默认 RHEL 9 FIPS 加密策不允许使用 AES HMAC-SHA1 加密类型的密钥派生功能，如 5.1 章节 RFC3961 所定义的。

当在 FIPS 模式下将 RHEL 9 身份管理(IdM)副本添加到 FIPS 模式下的 RHEL 8 IdM 环境（其中，第一个服务器安装在 RHEL 8.6 系统或更早的版本上）中时，这个约束是一个阻止因素。这是因为在 RHEL 9 和之前的 RHEL 版本之间没有通用的加密类型，它们通常使用 AES HMAC-SHA1 加密类型，但不使用 AES HMAC-SHA2 加密类型。

您可以通过在服务器上输入以下命令来查看 IdM 主密钥的加密类型：

```
# kadmin.local getprinc K/M | grep -E '^Key:'
```

要临时解决这个问题，在 RHEL 9 副本上启用 AES HMAC-SHA1 ：

```
update-crypto-policies --set FIPS:AD-SUPPORT
```

WARNING

这个临时解决方案可能会违反 FIPS 合规性。

因此，向 IdM 部署添加 RHEL 9 副本可以正确进行。

请注意，目前有一个正在进行的工作来提供一个在 RHEL 7 和 RHEL 8 服务器上生成缺少的 AES HMAC-SHA2 加密的 Kerberos 密钥的流程。这将在 RHEL 9 副本上取得 FIPS 140-3 合规性。但是，这个过程将

无法完全自动化，因为 Kerberos 密钥加密的设计无法将现有的密钥转换为不同的加密类型。唯一的方法是要求用户更新其密码。

Jira:RHEL-4888

SSSD 可正确注册 DNS 名称

在以前的版本中，如果 DNS 被错误建立，第一次尝试注册 DNS 名称时，SSSD 总是失败。要临时解决这个问题，这个更新提供了一个新的参数 **dns_resolver_use_search_list**。设置 **dns_resolver_use_search_list = false**，以避免使用 DNS 搜索列表。

Bugzilla:1608496[1]

因为 EMS 强制，使用 RHEL 9.2+ IdM 服务器在 FIPS 模式下安装 RHEL 7 IdM 客户端会失败

对于启用了 FIPS 的 RHEL 9.2 及更新系统上的 TLS 1.2 连接，TLS **Extended Master Secret** (EMS)扩展 (RFC 7627)现在是强制的。这符合 FIPS-140-3 要求。但是，RHEL 7.9 及较低版本中提供的 **openssl** 版本不支持 EMS。因此，使用在 RHEL 9.2 及更新版本上运行的启用了 FIPS 的 IdM 服务器安装 RHEL 7 身份管理(IdM)客户端会失败。

如果在安装 IdM 客户端前将主机升级到 RHEL 8 不是一个选项，请通过在 FIPS 加密策略之上应用 NO-ENFORCE-EMS 子策略，删除 RHEL 9 服务器上 EMS 使用的要求来临时解决此问题：

```
# update-crypto-policies --set FIPS:NO-ENFORCE-EMS
```

请注意，这个删除不符合 FIPS 140-3 要求。因此，您可以建立并接受不使用 EMS 的 TLS 1.2 连接，RHEL 7 IdM 客户端的安装可以成功。

Jira:RHEL-4955

当 nsslapd-numlisteners 属性值超过 2 时，目录服务器会失败

如果 **nsslapd-numlisteners** 属性值大于 **2**，则目录服务器可能会关闭侦听文件描述符，而不是接受的文件描述符。因此，在一段时间后，Directory 服务器会停止侦听某些端口并失败。

要临时解决这个问题，请将 **nsslapd-numlisteners** 属性值设置为 **1**。

Jira:RHEL-17178[1]

11.12. 桌面

升级到 RHEL 9 后，VNC 没有运行

从 RHEL 8 升级到 RHEL 9 后，VNC 服务器无法启动，即使之前启用它。

要临时解决这个问题，在系统升级后手动启用 **vncserver** 服务：

```
# systemctl enable --now vncserver@:port-number
```

现在，每个系统引导后都会启用 VNC 并按预期启动。

Bugzilla:2060308

用户创建屏幕没有响应

当使用图形用户界面安装 RHEL 时，用户创建屏幕没有响应。因此，在安装过程中创建用户更为困难。

要临时解决这个问题，请使用以下解决方案之一创建用户：

- 在 VNC 模式下运行安装并重新定义 VNC 窗口的大小。
- 完成安装过程后创建用户。

Jira:RHEL-11924[1]

WebKitGTK 无法在 IBM Z 上显示网页

当尝试在 IBM Z 架构上显示网页时，WebKitGTK 网页浏览器引擎会失败。网页保持空白，WebKitGTK 进程意外终止。

因此，您无法使用使用 WebKitGTK 的应用程序的某些功能来显示网页，如下所示：

- Evolution 邮件客户端
- GNOME 在线帐户设置
- GNOME 帮助应用程序

Jira:RHEL-4157

11.13. 图形基础结构

NVIDIA 驱动程序可能会恢复到 X.org

在某些情况下，专有 NVIDIA 驱动程序会禁用 Wayland 显示协议并恢复到 X.org 显示服务器：

- 如果 NVIDIA 驱动程序的版本低于 470。
- 如果系统是使用混合图形的笔记本电脑。
- 如果您还没有启用所需的 NVIDIA 驱动程序选项。

另外，启用 Wayland，但如果 NVIDIA 驱动程序的版本低于 510，则桌面会话默认使用 X.org。

Jira:RHELPLAN-119001[1]

使用 NVIDIA 在 Wayland 上无法使用 night Light

当您的系统上启用了专有 NVIDIA 驱动程序时，Wayland 会话将无法使用 GNOME 的 **Night Light** 功能。NVIDIA 驱动程序目前不支持 **Night Light**。

Jira:RHELPLAN-119852[1]

x.org 配置工具无法在 Wayland 下工作

用于操作屏幕的 x.org 实用程序无法在 Wayland 会话中工作。值得注意的是，**xrandr** 实用程序无法在 Wayland 下工作，因为其处理、解析、轮转和布局的不同方法。

Jira:RHELPLAN-121049[1]

11.14. RED HAT ENTERPRISE LINUX 系统角色

如果 firewalld.service 被屏蔽了，使用 firewall RHEL 系统角色失败

如果在 RHEL 系统上 **firewalld.service** 被屏蔽了，则 **firewall** RHEL 系统角色失败。要临时解决这个问题，请对 **firewalld.service** 取消屏蔽：

```
systemctl unmask firewalld.service
```

Bugzilla:2123859

无法使用环境名称注册系统

当在 **rhc_environment** 中指定环境名称时，**rhc** 系统角色注册系统失败。作为临时解决方案，请在注册时使用环境 ID 而不是环境名称。

Jira:RHEL-1172

11.15. 虚拟化

在某些情况下，通过 https 或 ssh 安装虚拟机会失败

目前，当尝试通过 https 或 ssh 连接从 ISO 源安装客户机操作系统时，**virt-install** 工具会失败 - 例如使用 **virt-install --cdrom https://example/path/to/image.iso**。上述操作意外中止，并显示 **internal error: process exited while connecting to monitor** 消息，而不是创建虚拟机(VM)。

同样，使用 RHEL 9 web 控制台安装客户机操作系统失败，如果使用了 https 或 ssh URL，或 **Download OS** 功能，则会显示 **Unknown driver 'https'** 错误。

要临时解决这个问题，请在主机上安装 **qemu-kvm-block-curl** 和 **qemu-kvm-block-ssh**，以启用 https 和 ssh 协议支持。或者，使用不同的连接协议或不同的安装源。

Bugzilla:2014229

在虚拟机中使用 NVIDIA 驱动程序会禁用 Wayland

目前，NVIDIA 驱动程序与 Wayland 图形会话不兼容。因此，使用 NVIDIA 驱动程序的 RHEL 客户机操作系统会自动禁用 Wayland 并加载 Xorg 会话。这主要在以下情况下发生：

- 当您通过 NVIDIA GPU 设备传递给 RHEL 虚拟机(VM)
- 当您为 RHEL 虚拟机分配 NVIDIA vGPU mediated 设备

Jira:RHELPLAN-117234[1]

在 AMD Milan 系统上有时无法提供 Milan VM CPU 类型

在某些 AMD Milan 系统上，默认在 BIOS 中禁用了增强 REP MOVSB(**erms**)和 Fast Short REP MOVSB(**fsrm**)功能标记。因此，在这些系统上可能无法使用 **Milan** CPU 类型。另外，在具有不同功能标志设置的 Milan 主机之间的虚拟机实时迁移可能会失败。要临时解决这个问题，在主机的 BIOS 中手动打开 **erms** 和 **fsrm**。

Bugzilla:2077767[1]

带有故障切换设置的 hostdev 接口在热拔后无法进行热插

从正在运行的虚拟机(VM)中删除带有故障切换配置的 **hostdev** 网络接口后，该接口目前无法重新连接到同一正在运行的虚拟机。

Jira:RHEL-7337

带有故障切换 VF 的虚拟机实时复制迁移失败

目前，如果虚拟机使用启用了虚拟功能(VF)故障转移功能的设备，则试图对一个正在运行的虚拟机(VM)进行 post-copy 迁移会失败。要临时解决这个问题，请使用标准迁移类型，而不要使用 post-copy 迁移方式。

Jira:RHEL-7335

主机网络无法在实时迁移过程中 ping 使用 VF 的虚拟机

当使用配置的虚拟功能 (VF) 实时迁移虚拟机时，如使用虚拟 SR-IOV 软件的虚拟机，虚拟机的网络不对其它设备看到，如 **ping** 之类的命令无法访问虚拟机。完成迁移后，问题将不再发生。

Jira:RHEL-7336

禁用 AVX 会导致虚拟机无法引导

在使用具有高级向量扩展(AVX)支持的 CPU 的主机上，尝试引导明确禁用 AVX 的虚拟机当前会失败，并触发虚拟机中的内核 panic。

Bugzilla:2005173[1]

在网络接口重置后，Windows VM 无法获取 IP 地址

有时，Windows 虚拟机在自动网络接口重置后无法获取 IP 地址。因此，虚拟机无法连接到网络。要临时解决这个问题，在 Windows 设备管理器中禁用并重新启用网络适配器驱动程序。

Jira:RHEL-11366

Windows Server 2016 虚拟机有时会在热插拔 vCPU 后停止工作

目前，将 vCPU 分配给运行 Windows Server 2016 客户机操作系统的虚拟机(VM)可能会导致各种问题，如虚拟机意外终止、变得没有响应或重启。

Bugzilla:1915715

使用大量队列可能会导致虚拟机失败

当启用了虚拟可信平台模块(vTPM)设备并且 *multi-queue virtio-net* 功能被配置为使用超过 250 个队列时，虚拟机(VM)可能会失败。

这个问题是由 vTPM 设备的限制造成的。vTPM 设备对打开的文件描述符的最大数有一个硬编码的限制。因为会为每个新队列打开多个文件描述符，因此可能会超过内部 vTPM 的限值，从而导致虚拟机失败。

要临时解决这个问题，请选择以下两个选项之一：

- 保持 vTPM 设备启用，但使用少于 250 个队列。
- 禁用 vTPM 设备以使用超过 250 个队列。

Jira:RHEL-13335[1]

在具有 NVIDIA passthrough 设备的虚拟机上的冗余错误消息

使用带有 RHEL 9.2 及更新版本的操作系统的 Intel 主机机器时，带有直通 NVDIA GPU 设备的虚拟机(VM)会频繁地记录以下错误信息：

```
Spurious APIC interrupt (vector 0xFF) on CPU#2, should never happen.
```

但是，这个错误消息不会影响虚拟机的功能，可以忽略。详情请查看 红帽知识库。

Bugzilla:2149989[1]

在带有 AMD EPYC CPU 的主机上进行 v2v 转换后，一些 Windows 客户机不能启动

在使用 **virt-v2v** 工具将使用 Windows 11 或 Windows Server 2022 的虚拟机(VM)转换为客户机操作系统后，虚拟机当前不能启动。这会在使用 AMD EPYC 系列 CPU 的主机上发生。

Bugzilla:2168082[1]

在主机上重启 OVS 服务可能会阻止在其上运行的虚拟机的网络连接

当 Open vSwitch (OVS)服务在主机上重启时或崩溃时，在此主机上运行的虚拟机(VM)无法恢复网络设备的状态。因此，虚拟机可能无法完全接收数据包。

此问题只会影响在 **virtio** 网络堆栈中使用压缩 virtqueue 格式的系统。

要临时解决这个问题，使用 **virtio** 网络设备定义中的 **packed=off** 参数来禁用压缩的 virtqueue。当禁用压缩的 virtqueue 时，网络设备的状态在某些情况下可以从 RAM 中恢复。

Jira:RHEL-333

恢复中断的复制后虚拟机迁移可能会失败

如果虚拟机(VM)的复制后迁移中断，然后在同一传入端口上立即恢复，则迁移可能会失败，并显示以下错误 **Address already in use**

要临时解决这个问题，请在恢复后复制迁移或切换到迁移恢复的另一个端口前至少等待 10 秒。

Jira:RHEL-7096

NUMA 节点映射在 AMD EPYC CPU 上无法正常工作

QEMU 无法正确处理 AMD EPYC CPU 上的 NUMA 节点映射。因此，如果使用 NUMA 节点配置，具有这些 CPU 的虚拟机(VM)的性能可能会受到负面影响。另外，虚拟机在启动过程中会显示类似如下的警告。

```
sched: CPU #4's llc-sibling CPU #3 is not on the same node! [node: 1 != 0]. Ignoring dependency.
WARNING: CPU: 4 PID: 0 at arch/x86/kernel/smpboot.c:415 topology_sane.isra.0+0x6b/0x80
```

要临时解决这个问题，请不要将 AMD EPYC CPU 用于 NUMA 节点配置。

Bugzilla:2176010

虚拟机迁移过程中的 NFS 故障会导致迁移失败和源虚拟机 coredump

目前，如果 NFS 服务或服务器在虚拟机(VM)迁移过程中关闭，则源虚拟机的 QEMU 在重新开始运行时无法重新连接到 NFS 服务器。因此，迁移会失败，并在源虚拟机上发起 coredump。目前，还没有可用的临时解决方案。

Bugzilla:2058982

PCIe ATS 设备无法在 Windows 虚拟机上工作

当您在带有 Windows 客户机操作系统的虚拟机的 XML 配置中配置 PCIe 地址转换服务(ATS)设备时，在引导虚拟机后，客户机不会启用 ATS 设备。这是因为 Windows 目前不支持 **virtio** 设备上的 ATS。

如需更多信息，请参阅 红帽知识库。

Bugzilla:2073872

virsh blkiotune --weight 命令无法设置正确的 cgroup I/O 控制器值

目前，使用 **virsh blkiotune --weight** 命令设置 VM 权重无法按预期工作。该命令无法在 cgroup I/O 控制器接口文件中设置正确的 **io.bfq.weight** 值。目前还没有临时解决方案。

Bugzilla:1970830

启动带有 NVIDIA A16 GPU 的虚拟机有时会导致主机 GPU 停止工作

目前，如果您启动一个使用 NVIDIA A16 GPU 直通设备的虚拟机，在某些情况下，主机系统上的 NVIDIA A16 GPU 物理设备会停止工作。

要临时解决这个问题，请重启 hypervisor，并将 GPU 设备的 **reset_method** 设置为 **bus** ：

```
# echo bus > /sys/bus/pci/devices/<DEVICE-PCI-ADDRESS>/reset_method
# cat /sys/bus/pci/devices/<DEVICE-PCI-ADDRESS>/reset_method
bus
```

详情请查看 红帽知识库。

Jira:RHEL-7212[1]

带有 FIFO 调度程序的 RT 虚拟机无法引导

目前，在将实时(RT)虚拟机(VM)设置为对 vCPU 调度程序使用 **fifo** 设置后，当您尝试引导它时，虚拟机变得无响应。相反，虚拟机显示 **Guest has not initialized the display (yet)** 错误。

Jira:RHEL-2815[1]

Windows 虚拟机可能会因为存储错误而变得无响应

在使用 Windows 客户机操作系统的虚拟机上，在高 I/O 负载下，系统在某些情况下会变得无响应。当发生这种情况时，系统会记录一个 **viostor Reset to device, \Device\RaidPort3, was issued** 错误。

Jira:RHEL-1609[1]

在引导时，带有某些 PCI 设备的 Windows 10 虚拟机可能会变得无响应

目前，如果将具有本地磁盘后端的 **virtio-win-scsi** PCI 设备被附加到虚拟机，则使用 Windows 10 客户机操作系统的虚拟机(VM)会在启动过程中变得无响应。要临时解决这个问题，请引导启用了 **multi_queue** 选项的虚拟机。

Jira:RHEL-1084[1]

virtio-win 驱动程序的 virtio-win-guest-tool 的修复功能无法工作

目前，当对 **virtio-win** 驱动程序使用 **virtio-win-guest-tool** 的 **Repair** 按钮时，如 Virtio Balloon Driver，则按钮无效。因此，在从客户机上删除后，驱动程序无法重新安装。

Jira:RHEL-1517[1]

具有内存气球设备集的 Windows 11 虚拟机在重启过程中可能会意外关闭

目前，重新引导使用 Windows 11 客户机操作系统和内存 balloon 设备的虚拟机(VM)在某些情况下会失败，并显示 **DRIVER POWER STAT FAILURE** 蓝屏错误。

Jira:RHEL-935[1]

在高网络负载情况下迁移 Windows 11 或 Windows Server 2022 虚拟机有时会失败

当实时迁移使用 Windows Server 2022 或 Windows 11 作为客户机操作系统的虚拟机(VM)时，如果网络受到高打包损失的影响，迁移可能会变得无响应或意外终止。

Jira:RHEL-2316[1]

在某些情况下恢复复制后虚拟机迁移失败

目前，当执行虚拟机(VM)的复制后迁移时，如果在迁移的恢复阶段发生代理网络故障，则虚拟机会变得无响应，且迁移无法恢复。相反，恢复命令显示以下错误：

```
error: Requested operation is not valid: QEMU reports migration is still running
```

Jira:RHEL-7115

virtio balloon 驱动程序有时在 Windows 10 虚拟机上无法工作

在某些情况下，virtio-balloon 驱动程序无法在使用 Windows 10 客户机操作系统的虚拟机(VM)上正常工作。因此，此类虚拟机可能无法有效地使用其分配的内存。

Jira:RHEL-12118

virtio 文件系统在 Windows 虚拟机中性能不佳

目前，当在使用 Windows 客户机操作系统的虚拟机(VM)上配置了 virtio 文件系统(virtiofs)时，虚拟机中的 virtiofs 性能比使用 Linux 客户机的虚拟机中的性能要差的多。

Jira:RHEL-1212[1]

在 Windows 虚拟机上热拔存储设备可能会失败

在使用 Windows 客户机操作系统的虚拟机(VM)上，当虚拟机运行时删除存储设备（也称为设备热拔）在某些情况下会失败。因此，存储设备一直附加在虚拟机上，磁盘管理器服务可能会变得无响应。

Jira:RHEL-869

将 CPU 热插到 Windows 虚拟机可能会导致系统失败

当将最大数量的 CPU 热插到启用了巨页的 Windows 虚拟机(VM)时，客户机操作系统可能会崩溃，并显示以下 *停止错误* ：

```
PROCESSOR_START_TIMEOUT
```

Jira:RHEL-1220

在 Windows 虚拟机上更新 virtio 驱动程序可能会失败

当在 Windows 虚拟机(VM)上更新 KVM 半虚拟化(**virtio**)驱动程序时，更新可能会导致鼠标停止工作，可能无法对新安装的驱动程序签名。当通过从 **virtio-win-guest-tools** 软件包（其是 **virtio-win.iso** 文件的一部分）安装来更新 **virtio** 驱动程序时，此问题会发生。

要临时解决这个问题，请使用 Windows 设备管理器更新 **virtio** 驱动程序。

Jira:RHEL-574[1]

kdump 在带有 AMD SEV-SNP 的虚拟机上失败

目前，kdump 在使用带有 Secure Nested Paging (SNP)功能的 AMD Secure Encrypted Virtualization (SEV)的 RHEL 9 虚拟机(VM)上失败。

Jira:RHEL-10019[1]

11.16. 云环境中的 RHEL

在 Nutanix AHV 中使用 LVM 克隆或恢复 RHEL 9 虚拟机会导致非 root 分区消失

当在 Nutanix AHV 虚拟机监控程序上托管的虚拟机中运行 RHEL 9 客户机操作系统时，从快照中恢复虚拟机或克隆虚拟机目前会导致虚拟机中的非 root 分区在虚拟机中使用逻辑卷管理(LVM)时消失。因此，会出现以下问题：

- 从快照恢复虚拟机后，虚拟机无法引导，而是进入紧急模式。
- 通过克隆创建的虚拟机无法引导，而是进入紧急模式。

要临时解决这个问题，在虚拟机的紧急模式下执行以下操作：

1. 删除 LVM 系统设备文件：**rm /etc/lvm/devices/system.devices**
2. 重新创建 LVM 设备设置：**vgimportdevices -a**
3. 重启虚拟机

这样，克隆或恢复的虚拟机可以正确引导。

另外，为了避免这个问题发生，请在克隆虚拟机或创建虚拟机快照前进行以下操作：

1. 在 **/etc/lvm/lvm.conf** 文件中取消注释 **use_devicesfile = 0** 行
2. 重启虚拟机

Bugzilla:2059545[1]

在 ESXi 上自定义 RHEL 9 客户机有时会导致网络问题

目前，在 VMware ESXi hypervisor 中自定义 RHEL 9 客户机操作系统无法正常工作。因此，如果客户机使用这样的密钥文件，它有不正确的网络设置，如 IP 地址或网关。

有关详情和临时解决方案说明，请参阅 VMware 知识库。

Bugzilla:2037657[1]

如果 RHEL 实例是由 cloud-init 提供的，且使用 NFSv3 挂载条目配置的，则其在 Azure 上无法引导

目前，如果 VM 是由 **cloud-init** 工具提供的，且虚拟机的客户机操作系统在 **/etc/fstab** 文件中有 NFSv3 挂载条目，则在 Microsoft Azure 云平台上引导 RHEL 虚拟机(VM)会失败。

Bugzilla:2081114[1]

在 VMware 主机上的 RHEL 虚拟机中设置静态 IP 无法正常工作

目前，当使用 RHEL 作为 VMware 主机上虚拟机(VM)的客户机操作系统时，DatasourceOVF 功能无法正常工作。因此，如果您使用 **cloud-init** 实用程序将虚拟机的网络设置为静态 IP，然后重启虚拟机，则虚拟机的网络将更改为 DHCP。

要临时解决这个问题，请参阅 VMware 知识库。

Jira:RHEL-12122

当启用了 kmemleak 选项时，大型虚拟机可能无法引导到 debug 内核

当试图将 RHEL 9 虚拟机(VM)引导到 debug 内核时，如果机器内核使用 **kmemleak=on** 参数，则引导可能会失败，并显示以下错误。

```
Cannot open access to console, the root account is locked.
See sulogin(8) man page for more details.

Press Enter to continue.
```

这个问题主要影响大型虚拟机，因为它们在引导序列中花费了大量时间。

要临时解决这个问题，请编辑机器上的 **/etc/fstab** 文件，并向 **/boot** 和 **/boot/efi** 挂载点添加额外的超时选项。例如：

```
UUID=e43ead51-b364-419e-92fc-b1f363f19e49 /boot xfs defaults,x-systemd.device-timeout=600,x-systemd.mount-timeout=600 0 0

UUID=7B77-95E7 /boot/efi vfat defaults,uid=0,gid=0,umask=077,shortname=winnt,x-systemd.device-timeout=600,x-systemd.mount-timeout=600 0 2
```

Jira:RHELDOCS-16979[1]

11.17. 支持性

在 IBM Power Systems Little Endian 上运行 sos report 时超时

当在具有带有数百或数千个 CPU 的 IBM Power Systems，Little Endian 上运行 **sos report** 命令时，处理器插件会在收集 **/sys/devices/system/cpu** 目录的大量内容时达到默认的 300 秒超时时间。作为临时解决方案，请相应地增加插件的超时时间：

- 对于一次性设置，请运行：

```
# sos report -k processor.timeout=1800
```

- 对于永久性更改，请编辑 **/etc/sos/sos.conf** 文件的 **[plugin_options]** 部分：

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

示例值设为 1800。特定的超时值高度依赖于特定的系统。要适当地设置插件超时，您可以首先通过运行以下命令来估算收集一个没有超时的插件所需的时间：

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

Bugzilla:1869561[1]

11.18. 容器

在较旧的容器镜像中运行 systemd 无法正常工作

在较旧的容器镜像（如 **centos:7**）中运行 systemd 将无法正常工作：

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
 Storing signatures
 Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
 [!!!!!!] Failed to mount API filesystems, freezing.
```

要临时解决这个问题，请使用以下命令：

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

Jira:RHELPLAN-96940[1]

附录 A. 按组件划分的问题单列表

在本文档中列出了 Bugzilla 和 JIRA 问题单以供参考。这些链接会指向本文档中描述问题单的发行注记。

组件	票证
389-ds-base	Bugzilla:2188627,Bugzilla:1987471,Bugzilla:2149025,Bugzilla:2166332,Bugzilla:2189946,Bugzilla:2189954,Bugzilla:1975930,Bugzilla:1974242,Bugzilla:1759941,Bugzilla:2053204, Bugzilla:2116948,Bugzilla:2179278,Bugzilla:2189717,Bugzilla:2170494,Bugzilla:2098236,JIRA:RHEL-17178
NetworkManager	Bugzilla:2176137,Bugzilla:2161915,Bugzilla:2151986,Bugzilla:2190375,Bugzilla:2069001,Bugzilla:2069004,Bugzilla:2148684,Bugzilla:2158328,Bugzilla:2180966,Bugzilla:2151040,Bugzilla:1894877
发行注记	Jira:RHELDOCS-16861, Jira:RHELDOCS-16760, Jira:RHELDOCS-16756, Jira:RHELDOCS-16612, Jira:RHELDOCS-17102, Jira:RHELDOCS-16979
anaconda	Bugzilla:2171811,Bugzilla:2164819,Bugzilla:2177219,Bugzilla:2157921,Bugzilla:2065754,Bugzilla:2107346,Bugzilla:2127473,Bugzilla:2050140,Bugzilla:1877697, JIRA:RHEL-4707 ,JIRA:RHEL-4707, JIRA:RHEL-4711,Bugzilla:1997832,JIRA:RHEL-4741,Bugzilla:2115783,JIRA:RHEL-4762,Bugzilla:2163497,JIRA:RHEL-4737
ansible-freeipa	Bugzilla:2175767,Bugzilla:2127903,Bugzilla:2127907
audit	Jira:RHELPLAN-161087
bacula	Jira:RHEL-6856
bind	Bugzilla:1984982
cloud-init	Bugzilla:2118235, Bugzilla:2172341, Jira:RHEL-12122
cockpit	Bugzilla:2203361
cockpit-appstream	Bugzilla:2030836
cockpit-machines	Bugzilla:2173584
crash	Bugzilla:2170283
createrepo_c	Bugzilla:2056318
crypto-policies	Bugzilla:2216257,Bugzilla:2193324,JIRA:RHEL-591,Bugzilla:2225222
cups-filters	Bugzilla:2229784

组件	票证
cyrus-sasl	Bugzilla:1995600
debugedit	Bugzilla:2177302
device-mapper-multipath	JIRA:RHEL-782,Bugzilla:2164869,Bugzilla:2033080,Bugzilla:2011699,Bugzilla:1926147
device-mapper-persistent-data	Bugzilla:2175198
dnf	Bugzilla:2124793,Bugzilla:2212262,Bugzilla:2073510
dnf-plugins-core	Bugzilla:2157844, Bugzilla:2134638, Bugzilla:2203100
edk2	Bugzilla:1935497
elfutils	Bugzilla:2182061,Bugzilla:2182059
fapolicyd	JIRA:RHEL-624, JIRA:RHEL-622,JIRA:RHEL-817,Bugzilla:2054740,JIRA:RHEL-520
fence-agents	Bugzilla:2187327
fuse3	Bugzilla:2188182
gcc	Bugzilla:2193180,Bugzilla:2168204,Bugzilla:2208908
gcc-toolset-13	Bugzilla:2171919
gcc-toolset-13-annobin	Bugzilla:2171923
gcc-toolset-13-binutils	Bugzilla:2171926
gcc-toolset-13-gcc	Bugzilla:2172093
gcc-toolset-13-gdb	Bugzilla:2172096
gfs2-utils	Bugzilla:2170017
gimp	Bugzilla:2047161
glibc	Bugzilla:2169978, Bugzilla:2213907, Bugzilla:2177235
gnupg2	Bugzilla:2073567, Bugzilla:2070722

组件	票证
gnutls	Bugzilla:2157953, Bugzilla:2108532
golang	Bugzilla:2185259,Bugzilla:2111072,Bugzilla:2092016
grafana	Bugzilla:2193018,Bugzilla:2190025
grub2	Bugzilla:2184069
gssproxy	Bugzilla:2181465
gtk3	Jira:RHEL-11924
httpd	Bugzilla:2184403, Bugzilla:2173295
ipa	Bugzilla:2196426, Bugzilla:2165880, Bugzilla:2229712, Bugzilla:2227831, Bugzilla:2084180, Bugzilla:2084166, Bugzilla:2069202, Bugzilla:2094673, Bugzilla:2057471, Jira:RHEL-12154, Jira:RHEL-12143, Jira:RHEL-4955
iproute	Jira:RHEL-428
java-17-openjdk	Bugzilla:2186647
jmc-core	Bugzilla:1980981
kdump-anaconda-addon	Jira:RHEL-11196
kernel	Bugzilla:1898184,Bugzilla:2177180,Bugzilla:2144528,Bugzilla:2210263,Bugzilla:2180124,Bugzilla:2192730,Bugzilla:2178741,Bugzilla:2195986,Bugzilla:2208365,Bugzilla:2187856, Bugzilla:2192722,Bugzilla:2171093,Bugzilla:2189292,Bugzilla:2193330,Bugzilla:2178930,Bugzilla:2092194,Bugzilla:2101598,Bugzilla:2218207,Bugzilla:2173947, Bugzilla:2178956,Bugzilla:2173594,Bugzilla:1613522,Bugzilla:1874182,Bugzilla:1995338,Bugzilla:1570255,Bugzilla:2177256,Bugzilla:2178699, Bugzilla:2023416 ,Bugzilla:2023416, Bugzilla:2021672,Bugzilla:2027304,Bugzilla:1660337,Bugzilla:1955275,Bugzilla:2142102,Bugzilla:2068237,Bugzilla:2040643,Bugzilla:2186375,Bugzilla:2183538, Bugzilla:2206599,Bugzilla:2167783,Bugzilla:2000616,Bugzilla:2013650,Bugzilla:2132480,Bugzilla:2059545,Bugzilla:2005173,Bugzilla:2128610,Bugzilla:2129288, Bugzilla:2013884,Bugzilla:2149989
内核/网络/IPSec	Jira:RHEL-1015
内核/网络/NIC 驱动程序	Jira:RHEL-6496, Jira:RHEL-9897, Jira:RHEL-15404

组件	票证
内核/平台启用/NVMe	Jira:RHEL-8171, Jira:RHEL-8164
内核/存储/存储驱动程序	Jira:RHEL-8466
内核/虚拟化/KVM	Jira:RHEL-7212, Jira:RHEL-2815
kernel-rt	Bugzilla:2181571
kernel-rt/其它	Jira:RHEL-9318
kexec-tools	Bugzilla:2083475,Bugzilla:2173815,Bugzilla:2169720, Bugzilla:2160676,Bugzilla:2113873,Bugzilla:2064708
keylime	JIRA:RHEL-595,JIRA:RHEL-11866,JIRA:RHEL-392,JIRA:RHEL-393, JIRA:RHEL-947,JIRA:RHEL-1252,JIRA:RHEL-11867,JIRA:RHEL-1518
keylime-agent-rust	Jira:RHEL-476, Jira:RHEL-395, Jira:RHEL-396
kmod	Bugzilla:2103605
kmod-kvdo	Jira:RHEL-8354
krb5	Bugzilla:2178298, Bugzilla:2155607, Jira:RHEL-4902, Bugzilla:2060798, Jira:RHEL-4875, Jira:RHEL-4889, Bugzilla:2060421, Bugzilla:2016312, Jira:RHEL-4888
libabigail	Bugzilla:2186931
libotr	Bugzilla:2086562
libpfm	Bugzilla:2185652
libvirt	Bugzilla:2032406,Bugzilla:2168499,Bugzilla:2014487,Bugzilla:2143158,Bugzilla:2078693
libxcrypt	Bugzilla:2034569
llvm-toolset	Bugzilla:2178796
lvm2	Bugzilla:2038183
mysql	Bugzilla:1991500
nfs-utils	Bugzilla:2081114

组件	票证
nginx-1.22-module	Bugzilla:2170808
nmstate	Bugzilla:2179916,Bugzilla:2180795,Bugzilla:2177733,Bugzilla:2183214,Bugzilla:2187622
nodejs	Bugzilla:2186717
nss	Bugzilla:2157950
nvme-cli	Bugzilla:2159929
nvme-stas	Bugzilla:1893841
open-vm-tools	Bugzilla:2037657
opencryptoki	Bugzilla:2160061
opensc	Jira:RHEL-280
openscap	Bugzilla:2217442,Bugzilla:2161499
openslp	Jira:RHEL-6995
openssh	Bugzilla:2070163, Bugzilla:2056884
openssl	Bugzilla:2216256,Bugzilla:2153471,Bugzilla:2188180,Bugzilla:2160797,Bugzilla:2168665,Bugzilla:1975836,Bugzilla:1681178,Bugzilla:1685470
osbuild	Jira:RHEL-4655
osbuild-composer	Bugzilla:2173928, Jira:RHEL-7999, Jira:RHEL-4649
oscap-anaconda-addon	Bugzilla:2172264, Jira:RHEL-1824
pacemaker	Bugzilla:2189301,Bugzilla:2182482
papi	Bugzilla:2111923,Bugzilla:2186927,Bugzilla:2215582
pause-container	Bugzilla:2106816
pcp	Bugzilla:2175602, Bugzilla:2185803
pcs	Bugzilla:2168155,Bugzilla:2163953,Bugzilla:2175881,Bugzilla:2182810,Bugzilla:1423473,Bugzilla:2177996,Bugzilla:1860626,Bugzilla:2163914

组件	票证
pcsc-lite-ccid	Bugzilla:2209457
perl-HTTP-Tiny	Bugzilla:2228412
pki-core	Bugzilla:2084181
podman	JIRA:RHELPLAN-154314,JIRA:RHELPLAN-154432,JIRA:RHELPLAN-154441,JIRA:RHELPLAN-154438,JIRA:RHELPLAN-163003,JIRA:RHELPLAN-160660,JIRA:RHELPLAN-154429,Bugzilla:2069279
postfix	Bugzilla:2134789
python-greenlet	Bugzilla:2149497
python3.11-lxml	Bugzilla:2157708
qemu-kvm	Bugzilla:1880531,Bugzilla:1965079,Bugzilla:1951814,Bugzilla:2060839,Bugzilla:2014229,JIRA:RHEL-7335,JIRA:RHEL-7336,Bugzilla:1915715,JIRA:RHEL-13335,JIRA:RHEL-333,Bugzilla:2176010,Bugzilla:2058982,Bugzilla:2073872
qemu-kvm / 设备	Jira:RHEL-1220
qemu-kvm / 图形	Jira:RHEL-7135
qemu-kvm/Live Migration	Jira:RHEL-7096, Jira:RHEL-2316, Jira:RHEL-7115
qemu-kvm/网络	Jira:RHEL-7337
rear	Bugzilla:2188593,Bugzilla:2172912,Bugzilla:2196445,Bugzilla:2145014
redis	Bugzilla:2129826
resource-agents	Bugzilla:2174911,Bugzilla:2142518,Bugzilla:2142002,Bugzilla:2182415,Bugzilla:2179003
restore	Bugzilla:1997366

组件	票证
rhel-system-roles	Bugzilla:2224384,Bugzilla:2216753,Bugzilla:2224385,Bugzilla:2185065,Bugzilla:2181656,Bugzilla:2211194,Bugzilla:2218592,Bugzilla:2211723,Bugzilla:2218204,Bugzilla:215137 3, Bugzilla:2179460,Bugzilla:2211748,Bugzilla:2229802,Bugzilla:2181657,Bugzilla:2168692,Bugzilla:2211984,Bugzilla:2232241,Bugzilla:2232231,Bugzilla:2224090, Bugzilla:2222761,Bugzilla:2223764,Bugzilla:2222428,Bugzilla:2216520,Bugzilla:2211187,Bugzilla:2209200,Bugzilla:2193058,Bugzilla:2186057,Jira:RHEL-1499, JIRA:RHEL-1397,JIRA:RHEL-906,JIRA:RHEL-1495,JIRA:RHEL-898,JIRA:RHEL-885,Bugzilla:1999770,Bugzilla:2123859,JIRA:RHEL-1172,Bugzilla:2186218
rpm	Bugzilla:2157836
rsyslog	Jira:RHELPLAN-160541
rust	Bugzilla:2191743,Bugzilla:2227082
s390utils	Bugzilla:1932480
samba	Bugzilla:2190415
scap-security-guide	Bugzilla:2221697,Bugzilla:2155790,JIRA:RHEL-1905,Bugzilla:2203791, Bugzilla:2213958,Bugzilla:2223178,Bugzilla:2193169,JIRA:RHEL-1800,Bugzilla:2038978
selinux-policy	Bugzilla:2080443,Bugzilla:2170495,Bugzilla:2184999,Bugzilla:2162663,Bugzilla:2112729,JIRA:RHELPLAN-163014,Bugzilla:2187745,Bugzilla:2229722,Bugzilla:2064274
setools	Bugzilla:2231801
sevctl	Bugzilla:2104857
scs	Bugzilla:1869561
squid-container	Bugzilla:2178953
sssd	Bugzilla:2065693, Bugzilla:2056482, Bugzilla:1608496
stratisd	Bugzilla:2041558
subscription-manager	Bugzilla:2163716,Bugzilla:2136694
sysstat	Jira:RHEL-12009
systemd	Bugzilla:2018112, Jira:RHEL-6105

组件	票证
systemtap	Bugzilla:2186934
tang	Bugzilla:2188743
tigervnc	Bugzilla:2060308
tuned	Bugzilla:2113900
ubi9-micro-container	Bugzilla:2223028
udisks2	Bugzilla:1983602,Bugzilla:2213769
unbound	Bugzilla:2070495
valgrind	Bugzilla:2124346
virt-v2v	Bugzilla:2168082
virtio-win	Bugzilla:1969724,JIRA:RHEL-11366,JIRA:RHEL-910,JIRA:RHEL-1609,JIRA:RHEL-869
virtio-win / distribution	Jira:RHEL-1517, Jira:RHEL-574
virtio-win / virtio-win-prewhql	Jira:RHEL-1084, Jira:RHEL-935, Jira:RHEL-12118, Jira:RHEL-1212
webkit2gtk3	Jira:RHEL-4157
which	Bugzilla:2181974
xdp-tools	Bugzilla:2218500, Jira:RHEL-3382

组件	票证
其他	Bugzilla:2232554, Jira:RHELDOCS-17055, Jira:RHELPLAN-163133, Jira:RHELPLAN-163665, Jira:RHELDOCS-16405, Jira:RHELDOCS-16247, Bugzilla:2136937, Jira:RHELDOCS-16474, Jira:RHELDOCS-16462, Jira:RHELDOCS-16386, Jira:RHELPLAN-156196, Jira:RHELDOCS-16708, Jira:RHELDOCS-16709, Jira:RHELDOCS-16339, Jira:RHELDOCS-16877, Jira:RHELPLAN-122345, Jira:RHELDOCS-16487, Jira:RHELDOCS-16752, Jira:RHELDOCS-17101, Bugzilla:2236182, Jira:RHELDOCS-17040, Bugzilla:2020529, Bugzilla:2030412, Jira:RHELPLAN-103993, Jira:RHELPLAN-27394, Jira:RHELPLAN-27737, Jira:RHELDOCS-16861, Jira:RHELDOCS-17050, Bugzilla:1927780, Jira:RHELPLAN-110763, Bugzilla:1935544, Bugzilla:2089200, Jira:RHELDOCS-16948, Jira:RHELPLAN-99136, Jira:RHELDOCS-17380, Jira:RHELPLAN-103232, Bugzilla:1899167, Bugzilla:1979521, Jira:RHELPLAN-100087, Jira:RHELPLAN-100639, Bugzilla:2058153, Jira:RHELPLAN-113995, Jira:RHELPLAN-98983, Jira:RHELPLAN-131882, Jira:RHELPLAN-139805, Jira:RHELDOCS-16756, Jira:RHELPLAN-153267, Jira:RHELDOCS-16300, Jira:RHELDOCS-16432, Jira:RHELDOCS-16393, Jira:RHELDOCS-16612, Jira:RHELDOCS-17102, Jira:RHELPLAN-157225, Jira:RHELPLAN-157337, Bugzilla:1640697, Bugzilla:1697896, Bugzilla:2047713, Jira:RHELPLAN-96940, Jira:RHELPLAN-117234, Jira:RHELPLAN-119001, Jira:RHELPLAN-119852, Bugzilla:2077767, Bugzilla:2053598, Bugzilla:2082303, Jira:RHELPLAN-121049, Jira:RHELPLAN-157939, Jira:RHELPLAN-109613, Bugzilla:2160619, Bugzilla:2173992, Bugzilla:2185048, Bugzilla:1970830, Jira:RHELDOCS-16574

附录 B. 修订历史

0.2-2

2024 年 6 月 11 日星期二, Brian Angelica (bangelic@redhat.com)

- 添加已弃用的功能 RHELDOCS-18049 (Shells 和命令行工具)。

0.2-1

2024 年 6 月 11 日星期二, Brian Angelica (bangelic@redhat.com)

- 添加了一个已知问题 RHEL-24847 (Shells 和命令行工具)。

0.2-0

2024 年 5 月 16 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 RHEL-10019（虚拟化）。

0.1-9

2024 年 4 月 18 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个改进 RHEL-19142 （网络）。

0.1-8

2024 年 4 月 11 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个改进 BZ#1513934 (IdM)

0.1-7

2024 年 3 月 14 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 JIRA:RHEL-25967 （内核）

0.1-6

2024 3 月 4 日星期一, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 bug 修复 JIRA:SSSD-6096（身份管理）。

0.1-5

2024 年 2 月 28 日星期三， Gabriela Fialová(gfialova@redhat.com)

- 更新了一个 Bug 修复的已知问题 RHEL-8171 （存储）。

0.1-4

2024 年 2 月 7 日星期三, Lucie Vařáková(lvarakova@redhat.com)

- 添加了一个新功能 RHEL-14694（网络）。

0.1-3

2024 年 2 月 1 日星期四， Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 KI BZ#1834716（安全）

- 更新了一个已弃用的功能 RHELDOCS-16756（容器工具）

0.1-2

2024 年 1 月 29 日星期一，Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 bug 修复 RHELPLAN-157337（安全）

0.1-1

2024 年 1 月 4 日星期四，Lenka Špačková (lspackova@redhat.com)

- 添加了一个与 Python 有关的改进 RHELDOCS-17369 （动态编程语言、Web 和数据库服务器）。

0.1-0

2024 年 1 月 10 日星期三，Gabriela Fialová(gfialova@redhat.com)

- 添加已弃用的功能 RHELDOCS-17380（安全）

0.0-9

2024 年 1 月 2 日星期二, Gabriela Fialová(gfialova@redhat.com)

- 更新了增强 BZ#2184403 中的描述

0.0-8

2023 年 11 月 23 日星期四，Gabriela Fialová(gfialova@redhat.com)

- 添加了 KI RHEL-8354 (安装程序)

0.0-7

2023 年 11 月 22 日星期三, Gabriela Fialová(gfialova@redhat.com)

- Add IdM KI RHEL-17178

0.0-6

2023 年 11 月 21 日星期二，David Vozenilek (dvozenil@redhat.com)

- 添加了系统角色 RNs BZ#2211723,BZ#2218204,BZ#2186057

0.0-5

2023 年 11 月 20 日星期一，Jana Heves (jsvarova@redhat.com)

- 添加了 KI RHEL-15404 sst_kernel_generalists

0.0-4

2023 年 11 月 19 日星期日, Filip Hanzelka (fhanzelk@redhat.com)

- 在 IdM 中添加了 BF RHELDOCS-17011

0.0-3

2023 年 11 月 16 日星期四, Marek Suchánek (msuchane@redhat.com)

- 弃用了 Inkscape 和 LibreOffice Flatpak RHELDOCS-17102

0.0-2

2023 年 11 月 16 日星期四，Lenka Špačková (lspackova@redhat.com)

- **Node.js 20** 现在被完全支持(BZ#2186717)。

0.0-1

2023 年 11 月 8 日星期三, Gabriela Fialová(gfialova@redhat.com)

- 发布 Red Hat Enterprise Linux 9.3 发行注记。

0.0-0

2023 年 9 月 27 日星期三, Gabriela Fialová(gfialova@redhat.com)

- 发布 Red Hat Enterprise Linux 9.3 Beta 发行注记。