Red Hat Enterprise Linux 9

9.1 发行注记

Release Notes for Red Hat Enterprise Linux 9.1

Last Updated: 2024-06-27

Red Hat Enterprise Linux 9 9.1 发行注记

Release Notes for Red Hat Enterprise Linux 9.1

法律通告

摘要

本发行注记提供了在 Red Hat Enterprise Linux 9.1 中已实现的改进和附加组件的高级信息，并记录了本版本中已知的问题，以及重要的 bug 修复、技术预览、已弃用的功能和其他详情。 有关如何安装 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. 单击顶部导航栏中的 **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 9.1 的主要变化

安装程序和镜像创建

以下是 RHEL 9.1 GA 中的镜像构建程序关键亮点：

- 镜像构建程序现在支持：
  - 将镜像上传到 GCP
  - 自定义 **/boot** 分区
  - 将容器镜像直接推送到 registry
  - 现在，用户可以在镜像创建过程中自定义其蓝图。

更多信息请参阅 第 4.1 节 “安装程序和镜像创建”。

RHEL for Edge

以下是 RHEL 9.1-GA 中的 RHEL for Edge 的关键亮点：

- RHEL for Edge 现在支持使用默认配置安装服务，并使用 **fdo-admin** CLI 实用程序运行它们

更多信息请参阅 第 4.2 节 “RHEL for Edge”。

安全

RHEL 9.1 引进了 **Keylime**，它是使用可信平台模块 (TPM) 技术测试的远程机器。借助 Keylime，您可以验证并持续监控远程机器的完整性。

SELinux 用户空间软件包已升级至 3.4 版本。最显著的变化包括：

- 通过并行重新标记改进了重新标记的性能
- 在 **semodule** 工具中支持 SHA-256
- **libsepol-utils** 软件包中的新策略工具

在系统配置和 **clevis-luks-systemd** 子软件包中的更改使 Clevis 加密客户端也能够解锁在引导过程后期挂载的 LUKS 加密的卷，而无需在部署过程中使用 **systemctl enable clevis-luks-askpass.path** 命令。

如需更多信息，请参阅 新特性 - 安全。

Shell 和命令行工具

RHEL 9.1 引入了一个新的软件包 **xmlstarlet**。有了 **XMLStarlet**，您可以解析、转换、查询、验证和编辑 XML 文件。

RHEL 9.1 中更新了以下命令行工具：

- **opencryptoki** 更新到版本 3.18.0
- **powerpc-utils** 更新到版本 1.3.10
- **libvpd** 更新到 2.2.9 版本
- **lsvpd** 更新到 1.7.14 版本

- **ppc64-diag** 更新到版本 2.7.8

如需更多信息，请参阅 新功能 - Shell 和命令行工具

基础架构服务

RHEL 9.1 中更新了以下基础架构服务工具：

- **chrony** 更新到版本 4.2
- **unbound** 更新到版本 1.16.2
- **frr** 更新到版本 8.2.2

如需更多信息，请参阅 新功能 - 基础设施服务。

网络

NetworkManager 支持将连接配置集从已弃用的 **ifcfg** 格式迁移到 keyfile 格式。

NetworkManager 现在明确表示 RHEL 9 不提供 WEP 支持。

内核中的多路径 TCP (MPTCP)代码已从上游 Linux 5.19 更新。

详情请查看 新功能 - Networking。

动态编程语言、网页和数据库服务器

以下组件的更新版本现在作为新的模块流提供：

- PHP 8.1
- Ruby 3.1
- Node.js 18

另外，**Apache HTTP 服务器** 已更新至版本 2.4.53。

如需更多信息，请参阅 新特性 - 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

RHEL 9.1 中更新了以下系统工具链组件：

- GCC 11.2.1
- glibc 2.34
- binutils 2.35.2

更新了性能工具和调试器

RHEL 9.1 中更新了以下性能工具和调试器：

- GDB 10.2
- Valgrind 3.19
- SystemTap 4.7
- Dyninst 12.1.0

- elfutils 0.187

更新了性能监控工具
RHEL 9.1 中更新了以下性能监控工具：

- PCP 5.3.7
- Grafana 7.5.13

更新了编译器工具集
RHEL 9.1 中更新了以下编译器工具集：

- GCC Toolset 12
- LLVM Toolset 14.0.6
- Rust Toolset 1.62
- Go Toolset 1.18

具体更改请查看 第 4.14 节 “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-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。

如需更多信息，请参阅 OpenJDK 文档。

Java 工具
RHEL 9.1 引进了 **Maven 3.8** 作为新模块流。

如需更多信息，请参阅 第 4.14 节 “编译器和开发工具”。

身份管理

RHEL 9.1 中的身份管理 (IdM) 引入了一个技术预览，您可以将用户身份验证委派给支持 OAuth 2 设备授权流的外部身份供应商 (IdP)。当这些用户使用 SSSD 进行身份验证后，并在外部 IdP 完成验证和授权后，它们会收到使用 Kerberos ticket 的 RHEL IdM 单点登录功能。

如需更多信息，请参阅 技术预览 - 身份管理

Red Hat Enterprise Linux 系统角色

9.1 RHEL 系统角色中值得注意的新功能：

- RHEL 系统角色现在在禁用了事实收集的 playbook 中提供。
- **ha_cluster** 角色现在支持 SBD 隔离、Corosync 设置和配置捆绑包资源。
- **network** 角色现在为路由规则配置网络设置，支持使用 **nmstate API** 的网络配置，用户可以使用 IPoIB 能力创建连接。

- **microsoft.sql.server** 角色具有新的变量，如用于控制高可用性集群的变量、自动管理防火墙端口的变量，或者用于搜索 **mssql_tls_cert** 和 **mssql_tls_private_key** 值的变量。
- **logging** 角色支持各种新选项，如文件输入的 **startmsg.regex** 和 **endmsg.regex**，或 **template**, **severity** 和 **facility** 选项。
- **storage** 角色现在支持精简配置的卷，角色现在默认也有较少的详细程度。
- **sshd** 角色验证 drop-in 目录的 include 指令，角色现在可以通过 /etc/ssh/sshd_config 进行管理。
- **metrics** 角色现在可以导出 postfix 性能数据。
- **postfix** 角色现在有一个新选项来覆盖以前的配置。
- 在配置 masquerade 或 icmp_block_inversion 时，**firewall** 角色不需要 state 参数。在 **firewall** 角色中，您现在可以使用 absent 和 present 状态添加、更新或删除服务。该角色也可以提供 Ansible 事实，并使用 PCI 设备 ID 向区域添加或删除接口。**firewall** 角色具有用于覆盖之前配置的新选项。
- **selinux** 角色现在包含 **seuser** 和 **selevel** 参数的设置。

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：

- 在以下构架中，从 RHEL 8.6 升级到 RHEL 9.0：
  - 64 位 Intel
  - 64 位 AMD
  - 64-bit ARM
  - IBM POWER 9(little endian)
  - IBM Z 架构，不包括 z13
- 在使用 SAP HANA 的系统上，从 RHEL 8.6 升级到 RHEL 9.0

要确保您的系统在升级到 RHEL 9.0 后仍然被支持，您可以升级到最新的 RHEL 9.1 版本，或启用 RHEL 9.0 延长更新支持(EUS)存储库。

有关执行原位升级的步骤，请参阅从 RHEL 8 升级到 RHEL 9 。

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：

- 现在，可以在 Microsoft Azure 和带有 Red Hat Update Infrastructure (RHUI)上的 Microsoft Azure 和 Google Cloud Platform 上进行原位升级。
- OpenSSH 和 OpenSSL 配置现在在原位升级过程中被迁移。

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 Configurator
- 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
- Ceph 每个池 PG 数量计算器

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。

第 2 章 构架

Red Hat Enterprise Linux 9.1 带有内核版本 5.14.0-162，它支持以下构架（最低版本）：

- 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、创建安装介质和完成 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，请参阅执行高级的 RHEL 9 安装 文档 。

有关在基础 RHEL 安装中由 RPM 创建的用户和组列表，以及获取此列表的步骤，请查看 基本 RHEL 安装中的所有用户和组是什么？知识库文章。

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 仓库的内容旨在提供底层操作系统功能的核心组件，为所有安装提供基础操作系统的基础。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。

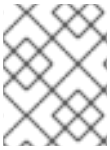
3.3. 应用程序流

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

某些初始 Application Streams (RPM 格式的初始 Application Streams) 的生命周期比 Red Hat Enterprise Linux 9 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。

决定您要安装的应用程序流的版本，并确保首先检查 Red Hat Enterprise Linux Application Stream Lifecycle。

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 9.1 中引进的新功能及主要改进。

4.1. 安装程序和镜像创建

安装程序中支持自动 FCP SCSI LUN 扫描

当在 IBM Z 系统上附加 FCP SCSI LUN 时，安装程序现在可以使用自动 LUN 扫描。如果没有通过 **zfcp.allow_lun_scan** 内核模块参数禁用，自动 LUN 扫描可用于在 NPIV 模式下对 FCP 设备的操作。它会被默认启用。它提供了对附加到具有指定的设备总线 ID 的 FCP 设备的存储区域网络中发现的所有 SCSI 设备的访问。不需要指定 WWPN 和 FCP LUN，只提供 FCP 设备总线 ID 就足够了。

(BZ#1937031)

内部镜像构建器现在支持 /boot 分区自定义

内部镜像构建器版本现在支持使用自定义的 **/boot** 挂载点分区大小构建镜像。您可以在蓝图自定义中指定 **/boot** 挂载点分区的大小，以便在默认引导分区太小时增加 **/boot** 分区的大小。例如：

```
[[customizations.filesystem]]
mountpoint = "/boot"
size = "20 GiB"
```

(JIRA:RHELPLAN-130379)

添加了 --allow-ssh kickstart 选项以启用基于密码的 SSH 根登录

在图形安装过程中，您可以选择启用基于密码的身份验证的 SSH root 登录。Kickstart 安装中没有此功能。在这个版本中，在 **rootpw** kickstart 命令中添加了一个 **--allow-ssh** 选项。这个选项可让 root 用户使用 SSH 和密码登录系统。

(BZ#2083269)

默认隐藏引导装载程序菜单

GRUB 引导装载程序现在被配置为默认隐藏引导菜单。这会带来更顺畅的引导体验。以下所有情况下都会隐藏引导菜单：

- 从桌面环境或登录屏幕重启系统时。
- 在安装过程后的第一次系统引导过程中。
- 当安装并启用 **greenboot** 软件包。

如果上一个系统引导失败，则 GRUB 总是在下次引导时显示引导菜单。

要手动访问引导菜单，请使用以下选项之一：

- 在启动过程中重复按 **Esc**。
- 在启动过程中重复按 **F8**。
- 在启动过程中按住 **Shift**.

要禁用这个功能并配置引导装载程序菜单默认显示，请使用以下命令：

```
# grub2-editenv - unset menu_auto_hide
```

(BZ#2059414)

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，则必须在完成 RHEL 安装或使用 kickstart 文件显式安装 **s390utils-base** 后手动安装软件包。

(BZ#1932480)

内部镜像构建器现在支持将镜像上传至 GCP

有了这个增强，您可以使用镜像构建器 CLI 来构建 **gce** 镜像，为您要上传镜像的用户或服务帐户提供凭证。因此，镜像构建器会创建镜像，然后将 **gce** 镜像直接上传到您指定的 GCP 环境中。

(BZ#2049492)

内部镜像构建器 CLI 支持将容器镜像直接推送到注册中心

有了这个增强，您可以使用镜像构建器 CLI 将 Edge 的 RHEL 容器镜像直接推送到容器注册中心。要构建容器镜像：

1. 设置上传供应商，并可以选择添加凭证。
2. 构建容器镜像，将容器注册中心和存储库作为参数传给 **composer-cli**。
   镜像就绪后，它在您设置的容器注册中心中提供。

(JIRA:RHELPLAN-130376)

镜像构建器内部用户现在在镜像创建过程中自定义其蓝图

有了这个更新，**Edit Blueprint** 页面已被删除，来统一镜像构建器服务中和 **cockpit-composer** 中镜像构建器应用程序的用户体验。现在，用户可以在镜像创建过程中创建他们的蓝图并添加他们的自定义，如添加软件包和创建用户。蓝图版本也已被删除，这样蓝图只有一个版本：当前版本。用户能够通过其已创建的镜像访问旧的蓝图版本。

(JIRA:RHELPLAN-122735)

4.2. RHEL FOR EDGE

RHEL for Edge 现在支持 fdo-admin cli 工具

在这个版本中，您可以使用 CLI 直接在所有部署场景中配置 FDO 服务。

运行以下命令为服务生成证书和密钥：

注意

此示例考虑您已安装了 **fdo-admin-cli** RPM 软件包。如果您使用了源代码并编译了它，则正确的路径为 **./target/debug/fdo-admin-tool** 或 **./target/debug/fdo-admin-tool**，具体取决于您的构建选项。

```
$ mkdir keys
```

```
$ for i in "diun" "manufacturer" "device_ca" "owner"; do fdo-admin-tool generate-key-and-cert $i; done
$ ls keys
device_ca_cert.pem  device_ca_key.der  diun_cert.pem  diun_key.der  manufacturer_cert.pem
manufacturer_key.der  owner_cert.pem  owner_key.der
```

因此，在安装并启动该服务后，它会使用默认设置运行。

(JIRA:RHELPLAN-122776)

4.3. 订阅管理

subscription-manager 工具显示操作的当前状态

subscription-manager 程序现在显示在处理当前操作期间的进度信息。当 **subscription-manager** 完成与服务器通信相关的操作（例如注册）的时间比预期的要长时，这非常有用。

要恢复到之前的行为，请输入：

```
# subscription-manager config --rhsm.progress_messages=0
```

(BZ#2092014)

4.4. 软件管理

modulesync 命令现在可以替换 RHEL 9 中的某些工作流

在 RHEL 9 中，在没有模块化元数据的情况下无法安装模块化软件包。在以前的版本中，您可以使用 **dnf** 命令下载软件包，然后使用 **createrepo_c** 命令重新分发这些软件包。

此增强引入了 **modulesync** 命令，以确保模块元数据的存在，从而确保软件包的可安装性。此命令从模块下载 RPM 软件包，并在工作目录中创建带有模块元数据的存储库。

(BZ#2066646)

4.5. SHELL 和命令行工具

Cronie 添加了对所选范围内随机时间的支持

Cronie 工具现在支持 cronjob 执行的 ~（范围内随机）运算符。因此，您可以在所选范围内的随机时间启动 cronjob。

(BZ#2090691)

ReaR 添加了在恢复前后执行命令的新变量

有了这个增强，ReaR 引进了两个新变量，用于在恢复前后轻松地自动化要执行的命令：

- **PRE_RECOVERY_COMMANDS** 接受一个命令数组。将在恢复开始前执行这些命令。
- **POST_RECOVERY_COMMANDS** 接受一个命令数组。将在恢复完成后执行这些命令。

这些变量是 **PRE_RECOVERY_SCRIPT** 和 **POST_RECOVERY_SCRIPT** 的替代方案，其有以下不同：

- 早期的 **PRE_RECOVERY_SCRIPT** 和 **POST_RECOVERY_SCRIPT** 变量接受单个 shell 命令。要将多个命令传递给这些变量，您必须使用分号将命令分开。

- 新的 **PRE_RECOVERY_COMMANDS** 和 **POST_RECOVERY_COMMANDS** 变量接受命令数组，数组的每个元素都会作为单独的命令执行。

因此，在恢复前后，在救援系统中提供要执行的多个命令现在更为简单且更不容易出错。

如需更多信息，请参阅 **default.conf** 文件。

(BZ#2111059)

新软件包：xmlstarlet

XMLStarlet 是一组用于解析、转换、查询、验证和编辑 XML 文件的命令行工具。新的 **xmlstarlet** 软件包提供了一组简单的 shell 命令，您可以像使用 UNIX 命令处理纯文本文件（如 **grep**、**sed**、**awk**、**diff**、**patch**、**join** 等）那样使用它们。

(BZ#2069689)

opencryptoki rebase 到版本 3.18.0

opencryptoki 软件包，它是公匙加密标准(PKCS)#11 的实现，已更新到版本 3.18.0。主要改进包括：

- 默认为符合联邦信息处理标准（FIPS）的令牌数据格式（tokversion=3.12）。
- 添加了对带有全局策略的限制机制和密钥的使用的支持。
- 添加了对机制使用统计计数的支持。
- **ICA/EP11** 令牌现在支持 **libica** 库版本 4。
- **p11sak** 工具可以为公钥和私钥设置不同的属性。
- **C_GetMechanismList** 不会在 EP11 令牌中返回 **CKR_BUFFER_TOO_SMALL**。

opencryptoki 支持两个不同的令牌数据格式：

- 较早的数据格式，它使用非FIPS 批准的算法（如 DES 和 SHA1）
- 新数据格式，只使用 FIPS 批准的算法。

旧数据格式不再可以正常工作，因为 FIPS 供应商只允许使用 FIPS 批准的算法。

重要

为了在 RHEL 9 上使用 openCryptoki，请在为系统启用 FIPS 模式前将令牌迁移为使用新数据格式。这是必要的，因为旧数据格式仍然是 **openCryptoki 3.17** 中的默认设置。当系统改为启用 FIPS 时，使用旧令牌数据格式的现有 **openCryptoki** 安装将无法正常工作。

您可以使用 **pkcstok_migrate** 程序（由 **openCryptoki** 提供），将令牌迁移为使用新的数据格式。请注意，**pkcstok_migrate** 在迁移过程中使用非FIPS 批准的算法。因此，在系统中启用 FIPS 模式前使用这个工具。如需更多信息，请参阅迁移到 FIPS 合规性 - pkcstok_migrate 工具程序 。

(BZ#2044179)

PowerPC-utils rebase 到版本 1.3.10

为 PowerPC 平台提供了各种工具的 **powerpc-utils** 软件包已更新至版本 1.3.10。主要改进包括：

- 添加了为 **ppc64_cpu** 工具中的能源和频率解析 Power 架构平台引用(PAPR)信息的能力。

- 改进了 **lparstat** 工具，来在 max 配置系统上 **lparstat -E** 命令失败时显示增强的错误消息。**lparstat** 命令报告逻辑分区有关的信息。
- 修复了 **lparstat** 命令中以传统格式报告的在线内存。
- 添加了对 **acc** 命令的支持，来动态更改 NX GZIP 加速器的服务信誉质量(QoS)。
- 添加了在 **printf()** 和 **sprintf()** 调用中格式化指定符的改进。
- **hcnmgr** 工具为混合虚拟网络提供了 HMC 工具，包括以下改进：
  - 在混合网络虚拟化 **HNV FEATURE** 列表中添加了 **wicked** 功能。**hcnmgr** 工具支持 wicked 混合网络虚拟化(HNV)使用 **wicked** 功能进行绑定。
  - **hcnmgr** 维护一个 **hcnid** 状态，以便稍后进行清理。
  - **hcnmgr** 排除了 NetworkManager (NM) **nmcli** 代码。
  - NM HNV **主从** 设置已被修复。
  - **hcnmgr** 支持将虚拟网络接口控制器(vNIC)用作备份设备。
- 修复了 **bootlist** 中无效的十六进制编号系统消息。
- **kpartx** 工具中包含的 **-l** 标志作为 **bootlist** 命令中的**-p** 分隔符值。
- 在 **sslot** 工具中添加了修复以防止在列出 IO 插槽时的内存泄漏。
- 在 **lsslot** 工具中为最新的外围设备组件互连快速(PCIe)插槽类型添加了 DRC 类型描述字符串。
- 修复了 **errinjct** 工具中 RTAS 的无效配置地址。
- 添加了对 **ofpathname** 工具中结构设备上非易失性内存(NVMf)的支持。此工具提供了一种将逻辑设备名称转换为开放固件设备路径的机制，反之亦然。
- 添加了对 **ofpathname** 工具中非对称名字空间访问（ANA）模式下非易失性内存(NVMe)支持的修复。
- 将 **smt.state** 文件作为配置文件安装。

(BZ#1920964)

Redfish 模块现在是 redhat.rhel_mgmt Ansible 集合的一部分

redhat.rhel_mgmt Ansible 集合现在包含以下模块：

- **redfish_info**
- **redfish_command**
- **redfish_config**

有了它，用户可以从管理自动化中受益，使用 Redfish 模块检索服务器健康状态，获取有关硬件和固件清单的信息、执行电源管理、更改 BIOS 设置、配置 Out-Of-Band (OOB)控制器、配置硬件 RAID 和执行固件更新。

(BZ#2112434)

libvpd 更新到版本 2.2.9

libvpd 软件包包含访问 Vital 产品数据 (VPD) 的类，它已更新至 2.2.9 版本。主要改进包括：

- 修复了数据库锁定
- 更新了 **libtool** 工具版本信息

(BZ#2051288)

lsvpd 更新到版本 1.7.14

提供构成硬件清单系统命令的**lsvpd** 软件包已更新至版本 1.7.14。在这个版本中，**lsvpd** 工具可防止运行 **vpdupdate** 命令时数据库文件崩溃。

(BZ#2051289)

ppc64-diag 更新到版本 2.7.8

平台诊断的 **ppc64-diag** 软件包已更新至 2.7.8 版本。主要改进包括：

- 更新了构建依赖项，以使用 **libvpd** 工具版本 2.2.9 或更高版本
- 修复了在不支持平台上的 **extract_opal_dump** 错误消息
- 修复了 **GCC-8.5** 和 **GCC-11** 编译器的构建警告

(BZ#2051286)

sysctl 引入了参数识别语法，作为 systemd-sysctl

procps-ng 软件包中的 **sysctl** 工具（您可以用来在运行时修改内核参数）现在使用与 **systemd-sysctl** 程序相同的参数。在这个版本中，**sysctl** 会解析在配置行中包含连字符 (**-**) 或 globs (*) 的配置文件。有关 **systemd-sysctl** 语法的详情请参考 **sysctl.d(5)** man page。

(BZ#2052536)

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。

您可以按照 重命名 IPoIB 设备 流程为命名 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。

(BZ#2136937)

4.6. 基础架构服务

Chrony 现在使用 DHCPv6 NTP 服务器

chrony 的 NetworkManager 分配程序脚本会更新从 Dynamic Host Configuration Protocol (DHCP) 选项传递的网络时间协议 (NTP) 源。从 RHEL 9.1 开始，除了 DHCPv4 外，该脚本使用 DHCPv6 提供的 NTP 服务器。DHCP 选项 56 指定 DHCPv6 的使用，DHCP 选项 42 是特定于 DHCPv4 的使用。

(BZ#2047415)

chrony rebase 到版本 4.2

chrony 套件已更新至版本 4.2。与版本 4.1 相比，主要的改进包括：

- 服务器交错模式已得到改进，更加可靠，并支持单一地址转换器（网络地址转换 - NAT）后的多个客户端。
- 添加了对网络时间协议版本 4 (NTPv4)扩展字段的实验性支持，以提高时间同步的稳定性和估计错误的精度。您可以使用 **extfield F323** 选项启用此字段，这可以扩展协议 NTPv4 的功能。
- 添加了精度时间协议 (PTP)上 NTP 转发的实验性支持，以便在对 PTP 数据包有时间戳限制的网络接口卡(NIC)上启用完整的硬件时间戳。您可以使用 **ptpport 319** 指令来启用 PTP 上的 NTP 。

(BZ#2051441)

unbound rebase 到版本 1.16.2

unbound 组件已更新至版本 1.16.2。**unbound** 是一种验证、递归和缓存 DNS 解析器。主要改进包括：

- 有了带有 **RFC 8976** 的 ZONEMD 区域验的支持，接收者现在可以验证数据完整性和来源真实性的区域内容。
- 有了 **unbound**，您现在可以配置持久性 TCP 连接。
- 根据 DNS **draft-ietf-dnsop-svcb-https** 文档中的服务绑定和参数规范添加了 SVCB 和 HTTPS 类型及处理。
- **unbound** 从加密策略获取默认的 TLS 密码。
- 您可以根据 **RFC8375** 使用特殊用途域 **home.arpa**。这个域被指定为在住宅家庭网络中非唯一使用。
- **unbound** 现在支持为 stub 或转发区域启用 **tcp-upstream** 查询选择。
- 现在，**aggressive-nsec** 选项默认为 **yes**。
- **ratelimit** 逻辑已更新。
- 当查询被 Unbound 响应策略区域(RPZ) nxdomain 回复阻止时，您可以使用新的 **rpz-signal-nxdomain-ra** 选项来取消 **RA** 标志设置。
- 根据 **RFC8914**，有了对扩展 DNS 错误(EDE)的基本支持，您可以从其他错误信息中受益。

(BZ#2087120)

现在，在 whois中提供了密码加密功能

whois 软件包现在提供 **/usr/bin/mkpasswd** 二进制文件，您可以使用该二进制文件通过 **crypt** C 库接口对密码进行加密。

(BZ#2054043)

frr 更新到版本 8.2.2

管理动态路由堆栈的 **frr** 软件包已更新至 8.2.2 版本。8.0 的主要变化和增强包括：

- 添加了以太网 VPN (EVPN) 路由类型-5 网关 IP Overlay Index。

- 在 Open-shortest-path-first (OSPFv3) 协议中添加了 Autonomous system border router (ASBR) summarization。
- 改进了 OSPFv3 中的 stub 和 not-so-stubby-areas (NSSA) 的使用。
- 添加了 OSPFv2 和 OSPFv3 中安全重启功能。
- 现在，边框网关协议 (BGP) 中的链路带宽根据 IEEE 754 标准进行编码。要使用上述编码方法，请在现有配置中运行 **邻居 PEER disable-link-bw-encoding-ieee** 命令。
- 在 BGP 中添加长期安全重启功能。
- 实施扩展管理关闭通信 **rfc9003**，并在 BGP 中扩展可选参数长度 **rfc9072**。

(BZ#2069563)

tuned 实时配置集现在会自动决定初始 CPU 隔离设置

tuned 是监控系统并优化性能配置集的服务。您还可以使用 **tuned-profiles-realtime** 软件包来隔离中央处理单元 (CPU)，为应用程序提供最执行时间的线程。

在以前的版本中，如果您没有指定在 **isolated_cores** 参数中隔离的 CPU 列表，则运行实时内核的实时配置集不会加载。

在这个版本中，TuneD 引入了 **calc_isolated_cores** 内置函数，用于自动计算日常和隔离内核列表，并将计算应用到 **isolated_cores** 参数。使用自动预设置时，每个插槽中的一个内核被保留用于内务处理，您可以在没有任何额外步骤的情况下开始使用实时配置集。如果要更改预先设置，请通过指定要隔离的 CPU 列表来自定义 **isolated_cores** 参数。

(BZ#2093847)

4.7. 安全

新软件包： keylime

RHEL 9.1 引进了 Keylime，这是一个用于测试远程系统的工具，它使用可信平台模块 (TPM) 技术。借助 Keylime，您可以验证并持续监控远程系统的完整性。您还可以指定对被监控的机器提供的加密有效负载，并定义在系统无法进行完整性测试时触发的自动化操作。

如需更多信息，请参阅 RHEL 9 安全强化文档中的浏览系统完整性。

(JIRA:RHELPLAN-92522)

OpenSSH 中的新选项支持设置最小 RSA 密钥长度

意外使用简短的 RSA 密钥可以使系统受到安全攻击的影响。在这个版本中，您可以为 OpenSSH 服务器和客户端设置最小 RSA 密钥长度。要定义最小 RSA 密钥长度，请在 OpenSSH 服务器的 **/etc/ssh/sshd_config** 文件中使用新的 **RequiredRSASize** 选项，并在 OpenSSH 客户端的 **/etc/ssh/ssh_config** 文件中使用新的 RequiredRSASize 选项。

(BZ#2066882)

crypto-policies 默认为 OpenSSH 强制 2048 位 RSA 密钥长度

使用短 RSA 密钥会使系统受到安全攻击的影响。因为 OpenSSH 现在支持最小 RSA 密钥长度，所以系统范围的加密策略会默认强制实施 RSA 的 2048 位最小密钥长度。

如果您遇到 OpenSSH 失败并显示 **Invalid key length** 错误消息，请开始使用较长的 RSA 密钥。

另外，您可以在安全性能方面使用自定义子策略来放宽限制。例如，如果 **update-crypto-policies --show** 命令报告当前策略为 **DEFAULT** ：

1. 通过将 **min_rsa_size@openssh = 1024** 参数插入到 **/etc/crypto-policies/policies/modules/RSA-OPENSSH-1024.pmod** 文件来定义自定义子策略。
2. 使用 **update-crypto-policies --set DEFAULT:RSA-OPENSSH-1024** 命令应用自定义子策略。

(BZ#2102774)

OpenSSL 中的新选项支持 SHA-1 进行签名

RHEL 9 中的 openssl 3.0.0 不支持 SHA-1 进行签名创建和验证(SHA-1 密钥分离功能(KDF)和基于哈希的消息验证代码(HMAC))。但是，为了与仍使用 SHA-1 进行签名的 RHEL 8 系统向后兼容，RHEL 9 增加了一个新的配置选项 **rh-allow-sha1-signatures**。如果在 **openssl.cnf** 的 **alg_section** 中启用此选项，允许创建和验证 SHA-1 签名。

如果设置了 LEGACY 系统范围的加密策略（而非传统供应商），则会自动启用这个选项。

请注意，这也会影响安装带有 SHA-1 签名的 RPM 软件包，这可能需要切换到 LEGACY 系统范围的加密策略。

(BZ#2060510, BZ#2055796)

crypto-policies 现在支持 sntrup761x25519-sha512@openssh.com

这个系统范围的加密策略的更新增加了对 **sntrup761x25519-sha512@openssh.com** 密钥交换 (KEX) 方法的支持。OpenSSH 套件中已经提供了 post-quantum **sntrup761** 算法，这种方法可以为来自上级计算机的攻击提供更好的安全性。要启用 **sntrup761x25519-sha512@openssh.com**，请创建并应用子策略，例如：

```
# echo 'key_exchange = +SNTRUP' > /etc/crypto-policies/policies/modules/SNTRUP.pmod
# update-crypto-policies --set DEFAULT:SNTRUP
```

如需更多信息，请参阅 RHEL 9 安全强化文档中的使用 subpolicies 自定义系统范围的加密策略。

(BZ#2070604)

NSS 不再支持少于 1023 位的 RSA 密钥

网络安全服务(NSS)库的更新将所有 RSA 操作的最小密钥大小从 128 改为 1023 位。这意味着 NSS 不再执行以下功能：

- 生成小于 1023 位的 RSA 密钥。
- 使用小于 1023 位的 RSA 密钥进行签名或验证 RSA 签名。
- 使用小于 1023 位的 RSA 密钥的加密或解密值。

(BZ#2091905)

SELinux 策略限制其他服务

selinux-policy 软件包已更新，因此以下服务现在被 SELinux 限制：

- **ksm**
- **nm-priv-helper**

- **rhcd**
- **stalld**
- **systemd-network-generator**
- **targetclid**
- **wg-quick**

(BZ#1965013, BZ#1964862, BZ#2020169, BZ#2021131, BZ#2042614, BZ#2053639, BZ#2111069)

SELinux 在类型转换中支持 self 关键字

SELinux 工具现在支持在策略源中使用 **self** 关键字的类型转换规则。支持使用 **self** 关键字进行类型转换，准备 SELinux 策略以标记匿名内节点。

(BZ#2069718)

SELinux 用户空间软件包已更新

SELinux 用户空间软件包 **libsepol**,**libselinux**,**libsemanage**,**policycoreutils**,**checkpolicy**, 和 **mcstrans** 更新为最新的上游版本 3.4。最显著的更改有：

- 添加了对通过 **setfiles**、**restorecon** 和 **fixfiles** 工具中的 **-T** 选项并行重新标记的支持。
  - 您可以在这个选项中指定进程线程数量，或使用 **-T 0** 来使用最大可用处理器内核。这可显著减少重新标记所需的时间。
- 添加了新的 **--checksum** 选项，该选项会输出模块的 SHA-256 哈希。
- 在 **libsepol-utils** 软件包中添加了新的策略实用程序。

(BZ#2079276)

SELinux 自动重新标记现在默认并行

因为新引入的并行重新标记选项可显著减少在多核系统上 SELinux 重新标记进程所需的时间，所以自动重新标记脚本现在包含 **fixfiles** 命令行中的 **-T 0** 选项。**t 0** 选项确保 **setfiles** 程序默认使用最大可用处理器内核重新标记。

如象以前的 RHEL 一样对重新标记只使用一个处理线程，输入 **fixfiles -T 1 onboot** 命令而不是 **fixfiles onboot**，或使用 **echo "-T 1" > /.autorelabel** 命令而不是 **touch /.autorelabel**。

(BZ#2115242)

SCAP 安全指南 rebase 到 0.1.63

SCAP 安全指南(SSG)软件包已更新到上游版本 0.1.63。此版本提供各种改进和程序错误修复，最重要的是：

- 添加了 **sysctl**、**grub2**、**pam_pwquality** 的新合规性规则以及构建时间内核配置。
- 强化 PAM 堆栈的规则现在使用 **authselect** 作为配置工具。注意：如果以其他方式编辑 PAM 堆栈，则通过此修改，将不会应用强化 PAM 堆栈的规则。

(BZ#2070563)

为 Rsyslog 错误文件添加了一个最大大小选项

使用新的 **action.errorfile.maxsize** 选项，您可以为 Rsyslog 日志处理系统指定错误文件的最大字节数。当错误文件达到指定大小时，Rsyslog 无法在其中写入额外的错误或其他数据。这可防止错误文件填满文件系统，并使主机不可用。

(BZ#2064318)

clevis-luks-askpass 现在被默认启用

/lib/systemd/system-preset/90-default.preset 文件现在包含 **enable clevis-luks-askpass.path** 配置选项，并且 **clevis-systemd** 子软件包的安装确保 **clevis-luks-askpass.path** 单元文件被启用。这使 Clevis 加密客户端也可解锁在引导过程后期挂载的 LUKS 加密的卷。在此次更新之前，管理员必须使用 **systemctl enable clevis-luks-askpass.path** 命令来使 Clevis 解锁此类卷。

(BZ#2107078)

fapolicyd rebase 到 1.1.3

fapolicyd 软件包已升级到版本 1.1.3。主要改进和 bug 修复包括：

- 规则现在可以包含新的主题 PPID 属性，该属性与主题的父 PID （进程 ID）匹配。
- OpenSSL 库代替了 Libgcrypt 库，来作为哈希计算的加密引擎。
- **fagenrules --load** 命令现在可以正常工作。

(BZ#2100041)

4.8. 网络

添加了 act_ctinfo 内核模块

在这个版本中，在 RHEL 中添加 **act_ctinfo** 内核模块。使用 **tc** 工具程序的 **ctinfo** 操作，管理员可以将 **conntrack** 标记或将网络数据包区分服务代码点 (DSCP) 的值复制到套接字缓冲区的 **mark** metadata 字段。因此，您可以根据 **conntrack** 标记或 DSCP 值过滤流量来使用条件。详情请查看 **tc-ctinfo (8)** man page。

(BZ#2027894)

cloud-init 每次在 Microsoft Azure 上引导时都更新网络配置

当管理员在虚拟机离线期间更新网络接口配置时，Microsoft Azure 不会更改实例 ID。有了这个增强，**cloud-init** 服务总会在虚拟机引导时更新网络配置，以确保 Microsoft Azure 上的 RHEL 使用最新的网络设置。

因此，如果您对接口手动配置设置，如额外的搜索域，**cloud-init** 可能会在重启虚拟机时覆盖它们。有关详情和临时解决方法，请参阅 在每次引导时 cloud-init-22.1-5 都会更新网络配置 解决方案。

(BZ#2144898)

PTP 驱动程序现在支持虚拟时钟和时间戳

在这个版本中，PTP 驱动程序可在空闲的 PHCs 之上创建虚拟 PTP 硬件 Clocks (PHCs)，方法是写入 **/sys/class/ptp/ptp/ptp*/n_vclocks**。因此，用户可以在一个接口上运行多个域同步与硬件时间戳。

(BZ#2066451)

firewalld 被更新到版本 1.1.1

firewalld 软件包已升级到 1.1.1 版本。与之前的版本相比，这个版本提供多个程序错误修复和增强：

新特性：

- 富规则支持用户空间日志记录的 NetFilter-log (NFLOG)目标。请注意，RHEL 中没有 NFLOG 能够记录守护进程。但是，您可以使用 **tcpdump -i nflog** 命令来收集您需要的日志。
- 支持使用 **ingress-zones=HOST** 和 **egress-zones={ANY, *source based zone*}** 中的端口转发。

其他显著变化包括：

- 支持 **afp**、**http3**、 **jellyfin**、**netbios-ns**、**ws-discovery**、**ws-discovery-client** 服务
- Z Shell 中的 tab-completion 和 sub-options 用于 **policy** 选项

(BZ#2040689)

NetworkManager 现在支持 advmss、rto_min 和 quickack 路由属性

在这个版本中，管理员可以使用以下属性配置 **ipv4.routes** 设置：

- **rto_min** (TIME) - 在与路由目的地通信时以毫秒为单位配置最小 TCP 重新传输超时。
- **quickack** (BOOL)- 一个每个路由设置来启用或禁用 TCP 快速 ACK
- **advms** (NUMBER)- 在建立 TCP 连接时，将最大片段大小 (MSS) 公告至路由目的地。如果未指定，Linux 将使用从第一个跃点设备的最大传输单元 (MTU) 计算的默认值

使用上述属性实施 **ipv4.routes** 的新功能是不需要运行 **dispatcher** 脚本。

请注意，当您激活了上述路由属性的连接后，会在内核中设置此类更改。

(BZ#2068525)

支持 nmstate中的 802.ad vlan-protocol 选项

nmstate API 现在支持使用 802.ad **vlan-protocol** 选项创建 **linux-bridge** 接口。此功能支持配置 Service-Tag VLAN。以下示例演示了在 **yaml** 配置文件中使用此功能。

```
---
interfaces:
  - name: br0
    type: linux-bridge
    state: up
    bridge:
      options:
        vlan-protocol: 802.1ad
      port:
        - name: eth1
          vlan:
            mode: trunk
            trunk-tags:
            - id: 500
```

(BZ#2084474)

firewalld 服务可以将源自本地主机的 NAT 数据包转发到不同主机和端口

您可以将运行 **firewalld** 服务的 localhost 发送的数据包转发到不同的目标端口和 IP 地址。该功能很有用，例如要将 **loopback** 设备上的端口转发到容器或虚拟机。在更改之前，**firewalld** 只能在收到源自于其他主机的数据包转发端口。如需了解更多详情以及说明性配置，请参阅 使用 DNAT 将 HTTPS 流量转发到其他主机。

(BZ#2039542)

NetworkManager 现在支持从 ifcfg-rh 迁移到密钥文件

用户可以将现有连接配置集文件从 **ifcfg-rh** 格式迁移到密钥文件格式。这样，所有连接配置集都将是一个位置，采用首选格式。密钥文件格式有以下优点：

- 这与 NetworkManager 如何表达网络配置的方式类似
- 保证与将来的 RHEL 版本兼容
- 更易于阅读
- 支持所有连接配置集

要迁移连接，请运行：

```
# nmcli connection migrate
```

请注意，在 RHEL 9 生命周期中，**ifcfg-rh** 文件将可以正常工作。但是，将配置迁移到关键文件格式可保证 RHEL 9 之外的兼容性。

详情请查看 **nmcli (1)**、nm-settings-keyfile (5) 和 nm-settings-ifcfg-rh (5)手册页。

(BZ#2059608)

在 nmstate API 中添加更多 DHCP 和 IPv6 自动配置属性

在这个版本中，在 nmstate API 中添加了对以下属性的支持：

- **DHCP-client-id** 用于 DHCPv4 连接，如 RFC 2132 和 4361 所述。
- **DHCP-duid** 用于 DHCPv6 连接，如 RFC 8415 中所述。
- 用于 IPv6 自动配置的 **addr-gen-mode**。您可以将此属性设置为：
  - **eui64**，如 RFC 4862 所述
  - **stable-privacy**，如 RFC 7217 所述

(BZ#2082043)

NetworkManager 现在明确表示 RHEL 9 不提供 WEP 支持

RHEL 9.0 中的 **wpa_supplicant** 软件包不再包含已弃用的和不安全的 Wiquivalent Privacy (WEP) 安全算法。此增强更新了 NetworkManager 以反应这些更改。例如，**nmcli device wifi list** 命令现在以灰色在列表的末尾返回 WEP 访问点，连接到 WEP-protected 网络会返回有意义的错误消息。

对于安全加密，只使用带有 Wi-Fi Protected Access 2 (WPA2) 和 WPA3 身份验证的 wifi 网络。

(BZ#2030997)

MPTCP 代码已更新

内核中的 MultiPath TCP (MPTCP) 代码已更新，上游 Linux 5.19。与之前的版本相比，这个更新提供了很多程序错误修复和增强：

- 添加了 **FASTCLOSE** 选项以关闭 MPTCP 连接，而无需完整的三向握手。
- 现在，添加了 **MP_FAIL** 选项，以便在初始握手后启用回退到 TCP。
- 通过添加额外的管理信息基础 (MIB) 计数器，改进了监控功能。
- 添加了对 MPTCP 侦听器套接字的监控支持。使用 **ss** 实用程序监控套接字。

(BZ#2079368)

4.9. 内核

RHEL 9.1 中的内核版本

Red Hat Enterprise Linux 9.1 与内核版本 5.14.0-162 一起发布。

(BZ#2125549)

list_lru 的内存消耗已被优化

内部内核数据结构 **list_lru** 跟踪内核索引节点和文件目录条目的"最早使用"状态。在以前的版本中，**list_lru** 分配结构的数量与挂载点的数量和存在内存 **cgroup** 的数量直接成比例。两个数字都随着正在运行的容器增加，导致 **O (n^2)** 的内存消耗量为 **n**，其中 n 是正在运行的容器数。此更新可优化系统中 **list_lru** 到 **O(n)** 的内存消耗。因此，现在有足够的内存供用户应用程序使用，特别是在有大量运行容器的系统中。

(BZ#2013413)

BPF 更新到 Linux 内核版本 5.16

Berkeley Packet Filter (BPF) 工具已更新至 Linux 内核版本 5.16，具有多个程序错误修复和增强。最显著的变化包括：

- 简化了对内部 BPF 程序项的处理，以及 **libbpf** 用户空间库中的 **bpf_program__set_attach_target()** API。**bpf_program__set_attach_target()** API 为基于 BPF 的程序设置基于 BTF 的附加目标。
- 添加了对 **BTF_KIND_TAG** kind 的支持，允许您标记声明。
- 添加了对 **bpf_get_branch_snapshot()** 帮助程序的支持，它允许追踪程序从硬件捕获最后一个分支记录(LBR)。
- 在 **libbpf** 用户空间库中添加了旧的 **kprobe** 事件，它允许通过旧接口创建 **kprobe** tracepoint 事件。
- 添加了通过 BPF 特定的结构使用 **__sk_buff** helper 功能访问硬件时间戳的功能。
- 添加了对 **AF_XDP** 缓冲池中 RX 缓冲区分配的批处理接口的支持，支持 **i40e** 和 **ice**。
- 添加了 **libbpf** 用户空间库中的传统 **uprobe** 支持，以补充最近合并的传统 **kprobe**。
- 将 **bpf_trace_vprintk()** 添加为 variadic **printk** helper。
- 添加了 **libbpf** opt-in for stricter BPF 程序部分处理，作为 **libbpf** 1.0 工作的一部分。

- 添加了 **libbpf** 支持来查找专用映射，如 **perf RB** 和内部删除 BTF 类型标识符。
- 添加了 **bloomfilter** BPF map 类型来测试集合中是否存在某一元素。
- 添加了对 BPF 中的内核模块功能调用的支持。
- 添加了在 light skeleton 中的无类型和弱 **ksym** 的支持。
- 添加了对 **BTF_KIND_DECL_TAG** 类型的支持。

有关运行中内核中可用的 BPF 功能的完整列表，请使用 **bpftool feature** 命令。

(BZ#2069045)

BTF 数据现在位于内核模块中

BPF 类型格式 (BTF) 是元数据格式，对与 BPF 程序和映射相关的调试信息进行编码。在以前的版本中，内核模块的 BTF 数据保存在 **kernel-debuginfo** 软件包中。因此，需要安装对应的 **kernel-debuginfo** 软件包才能将 BTF 用于内核模块。在这个版本中，BTF 数据现在直接位于内核模块中。因此，您不需要安装任何软件包就可以使 BTF 正常工作。

(BZ#2097188)

kernel-rt 源树已更新至 RHEL 9.1 树

kernel-rt 源已更新为使用最新的 Red Hat Enterprise Linux 内核源树。实时补丁集也更新至最新的上游版本 **v5.15-rt**。这些更新提供了很多程序错误修正和增强。

(BZ#2061574)

ARM 和 AMD 和 Intel 64 位构架启用动态抢占调度

RHEL 9 在 ARM 和 AMD 和 Intel 64 位构架中提供动态调度功能。此功能增强支持在引导时或运行时更改内核的抢占模式，而不是编译时间。**/sys/kernel/debug/sched/preempt** 文件包含当前的设置，并可以在运行时修改。

使用 **DYNAMIC_PREEMPT** 选项时，您可以在引导时将 **preempt=** 变量设置为 **none**、**voluntary** 或 **full**，**voluntary** 抢占为默认值。使用动态抢占处理，您可以覆盖默认的抢占模型，以改进调度延迟。

(BZ#2065226)

stalld 更新到版本 1.17

stalld 程序（提供 **stall** 守护进程）是防止 Linux 系统中操作系统线程的不足状态的机制。此版本监控星级状态的线程。当线程位于 CPU 运行队列中长于 starvation 的阈值，就会发生 starvation。

与之前的版本相比，这个 **stalld** 版本包括了很多改进和程序错误修复。值得注意的更改包括检测可运行的补救任务的能力。

当 **stalld** 检测到星号线程时，程序会将线程的调度类改为 **SCHED_DEADLINE** 策略，它为线程提供了指定 CPU 运行线程的小片段。当使用 **timeslice** 时，线程会返回其原始调度策略，而 **stalld** 会继续监控线程状态。

(BZ#2107275)

tpm2-tools 软件包已更新至 tpm2-tools-5.2-1 版本

tpm2-tools 软件包已更新到版本 **tpm2-tools-5.2-1**。此升级提供了很多重要的功能增强和程序错误修复。最显著的变化包括：

- 在使用 **tpm2_createprimary** 和 **tpm2_create** 工具创建的主对象时添加对公钥输出的支持。
- 添加对 **tpm2_print** 工具的支持，以打印公钥输出格式。**tpm2_print** 解码一个受信任的平台模块(TPM)数据结构，并打印括起的元素。
- 添加了对 **tpm2_eventlog** 工具的支持，以读取超过 64 KB 的日志。
- 添加 **tpm2_sessionconfig** 工具以支持显示和配置会话属性。

有关显著变化的更多信息，请参阅 **/usr/share/doc/tpm2-tools/Changelog.md** 文件。

(BZ#2090748)

Intel E800 设备现在支持 iWARP 和 RoCE 协议

在这个版本中，您可以使用 **enable_iwarp** 和 **enable_roce** devlink 参数打开和关闭 iWARP 或 RoCE 协议支持。使用这个强制功能，您可以使用其中一个协议配置该设备。Intel E800 设备不支持同一端口上这两个协议。

要为特定 E800 设备启用或禁用 iWARP 协议，首先获取卡的 PCI 位置：

```
$ lspci | awk '/E810/ {print $1}'
44:00.0
44:00.1
$
```

然后，启用或禁用协议。您可以将 **pci/0000:44:00.0** 用作第一个端口，并将 **pci/0000:44:00.1** 用于卡的第二个端口，作为 devlink 命令的参数

```
$ devlink dev param set pci/0000:44:00.0 name enable_iwarp value true cmode runtime
$ devlink dev param set pci/0000:44:00.0 name enable_iwarp value false cmode runtime
```

要为特定 E800 设备启用或禁用 RoCE 协议，获取卡的 PCI 位置，如上所示。然后使用以下命令之一：

```
$ devlink dev param set pci/0000:44:00.0 name enable_roce value true cmode runtime
$ devlink dev param set pci/0000:44:00.0 name enable_roce value false cmode runtime
```

(BZ#2096127)

4.10. 引导加载程序

GRUB 由新密钥签名

由于安全原因，GRUB 现在由新密钥签名。因此，您需要将 RHEL 固件更新至 FW1010.30 版本（或更新版本）或 FW1020，以便在启用了安全引导机制的情况下引导 IBM Power Systems 的 little-endian 变体。

(BZ#2074761)

在 IBM POWER 上启动虚拟机时可配置的磁盘访问重试

现在，您可以在逻辑分区(**lpar**)虚拟机(VM)在 IBM POWER 构架上引导时配置 GRUB 引导装载程序重试访问远程磁盘的次数。降低重试次数可防止某些情况下引导较慢。

在以前的版本中，当磁盘访问失败时，GRUB 会重试访问磁盘 20 次。如果您在连接到慢的存储区域网络(SAN)磁盘的 **lpar** 系统上执行实时分区移动(LPM)迁移，这会导致问题。因此，启动可能在系统上花费很长时间，直到 20 次重试完成为止。

有了这个更新，您现在可以使用 **ofdisk_retries** GRUB 选项配置和减少磁盘访问重试的次数。详情请参阅在 IBM POWER 上引导虚拟机时配置磁盘访问重试。

因此，在 POWER 上的 LPM 后，**lpar** 引导不再慢，而 **lpar** 系统可在没有失败的磁盘的情况下引导。

(BZ#2070725)

4.11. 文件系统和存储

Stratis 现在启用在创建时设置文件系统大小

现在，您可以在创建文件系统时设置所需的大小。在以前的版本中，自动默认大小为 1 TiB。在这个版本中，用户可以设置任意文件系统大小。较低限制不得低于 512 MiB。

(BZ#1990905)

改进了 Stratis 池的置备管理

借助对精简配置的管理改进，您现在可以改进警告，为池元数据精确分配空间，提高了可预测性、总体安全性和可靠性。新的不同模式会禁用过度置备。在这个版本中，用户可以禁用过度置备，以确保池包含足够的空间来支持其所有文件系统，即使这些完全已满。

(BZ#2040352)

Stratis 现在提供改进的独立池管理

现在，您可以停止并启动停止的独立 Stratis 池。在以前的版本中，**stratisd** 会试图为它检测到的所有设备启动所有可用的池。此功能增强提供了在 Stratis 中对独立池进行更灵活的管理，更好地调试和恢复功能。系统不再需要重启来为单个池执行恢复和维护操作。

(BZ#2039960)

启用了多路径设备路径的具体配置

在以前的版本中，因为不同协议的不同最佳配置，在没有为每个独立协议设置选项的情况下无法正确设置配置。在这个版本中，用户可以根据其路径传输协议配置多路径设备路径。使用 **/etc/multipath.conf** 文件中的 **overrides** 部分中的 **protocol** 子部分根据其协议正确配置多路径设备路径。

(BZ#2084365)

新的 libnvme 功能库

在以前的版本中，NVMe 存储命令行界面实用程序 (**nvme-cli**) 包含所有帮助程序函数和定义。此功能增强为 RHEL 9.1 增加了一个新的 **libnvme** 库。这个程序库包括：

- NVMe 规格结构的类型定义
- Enumerations 和 bit 字段
- 用于构建、分配和解码命令和有效负载的帮助功能
- 连接、扫描和管理 NVMe 设备的工具

在这个版本中，用户不需要复制代码和多个项目和软件包，如 **nvme-stas**，并可以依赖这个通用库。

(BZ#2099619)

新的库 libnvme 现在可用

有了这个更新，nvme-cli 被分为两个不同的项目：* **nvme-cli** 现在只包含特定于 **nvme** 工具的代码 * **libnvme** 库现在包含 NVMe 规格结构的所有类型定义、枚举、位字段、助手函数来构造、分配、解码命令和有效负载，以及用于连接、扫描和管理 NVMe 设备的工具。

(BZ#2090121)

4.12. 高可用性和集群

支持 Red Hat OpenStack 平台上的高可用性

现在，您可以在 Red Hat OpenStack 平台上配置高可用性集群。为了支持这个功能，红帽提供了以下新的集群代理：

- **fence_openstack**: OpenStack 上 HA 集群的隔离代理
- **openstack-info**: 配置 **openstack-info** 克隆资源的资源代理，这是 OpenStack 上 HA 集群所需的
- **openstack-virtual-ip**: 配置虚拟 IP 地址资源的资源代理
- **openstack-floating-ip**: 配置浮动 IP 地址资源的资源代理
- **openstack-cinder-volume**: 配置块存储资源的资源代理

(BZ#2121838)

pcs 支持在不需要系统重启的情况下更新多路径 SCSI 设备

现在，您可以使用 **pcs stonith update-scsi-devices** 命令更新多路径 SCSI 设备。这个命令更新 SCSI 设备，而不会导致运行在同一节点上其他集群资源的重启。

(BZ#2024522)

支持集群 UUID

在集群设置过程中，**pcs** 命令现在会为每个集群生成一个 UUID。因为集群名称不是一个唯一的集群标识符，因此您可以在管理多个集群时使用集群 UUID 来识别具有相同名称的集群。

您可以使用 **pcs cluster config [show]** 命令来显示当前集群的 UUID。您可以使用 **pcs cluster config uuid generate** 命令来向现有集群添加一个 UUID 到或重新生成一个 UUID（如果其 UUID 已存在）。

(BZ#2054671)

新的 pcs resource config 命令选项来显示重新创建配置的资源的 pcs 命令

pcs resource config 命令现在接受 **--output-format=cmd** 选项。指定这个选项会显示用来在不同系统上重新创建配置的资源的 **pcs** 命令。

(BZ#2058251)

新的 pcs stonith config 命令选项来显示重新创建配置的隔离设备的 pcs 命令

pcs stonith config 命令现在接受 **--output-format=cmd** 选项。指定这个选项会显示您用来在不同系统上重新创建配置的隔离设备的 **pcs** 命令。

(BZ#2058252)

pacemaker 更新到版本 2.1.4

Pacemaker 软件包已升级到 Pacemaker 2.1.4 的上游版本。主要变更包括：

- 现在，**multiple-active** 资源参数接受 **stop_unexpected** 的值，**multiple-active** 资源参数决定了当资源在不应该激活时在多个节点上活跃的恢复行为。默认情况下，这种情况需要资源的全面重启，即使资源在其应该运行的地方在成功运行。此参数的 **stop_unexpected** 值指定，只有多活跃资源意外实例才会停止。用户负责验证服务及其资源代理是否可以与额外的活跃实例一起正常工作，而无需全面重启。

- pacemaker 现在支持 **allow-unhealthy-node** 资源 meta-attribute。当此 meta-attribute 设为 **true** 时，由于降级节点健康状况，资源不会强制关闭节点。当健康资源设置了此属性时，集群可以自动检测节点的健康状态恢复，并将资源移回节点。

- 用户现在可以使用 **pcs acl group** 命令为系统组群指定访问控制列表 (ACLS)。Pacemaker 之前允许为单个用户指定 ACL，但有时更简单，更符合本地策略，来为系统组指定 ACL，并将其应用到该组中的所有用户。这个命令存在于早期版本中，但没有影响。

(BZ#2072108)

Samba 不再自动安装集群软件包

在本发行版本中，为 RHEL High Availability Add-On 安装软件包不再会自动安装 Samba 软件包。这也允许您删除 Samba 软件包，而无需自动删除 HA 软件包。如果您的集群使用 Samba 资源，则必须手动安装它们。

(BZ#1826455)

4.13. 动态编程语言、网页和数据库服务器

nodejs:18 模块流现已全面支持

以前作为技术预览提供的 **nodejs:18** 模块流，随着 RHSA-2022:8832 公告的发布而完全支持。**nodejs:18** 模块流现在提供 **Node.js 18.12**，它是一个长期支持(LTS)版本。

Node.js 18 包括在 RHEL 9.1 中，与 **Node.js 16** 相比，提供了许多新功能，以及程序错误和安全修复。

主要变更包括：

- **V8** 引擎已升级至版本 10.2。

- **npm** 软件包管理器已升级至 8.19.2 版本。

- **Node.js** 现在提供了一个新的实验性 **fetch** API。

- **Node.js** 现在提供了一个新的实验性 **node:test** 模块，它便于创建以 test Anything Protocol (TAP)格式报告结果的测试。

- **Node.js** 现在更喜欢使用 IPv6 地址，而不是 IPv4。

要安装 **nodejs:18** 模块流，请使用：

```
# dnf module install nodejs:18
```

(BZ#2083072)

新模块流： php:8.1

RHEL 9.1 添加 **PHP 8.1** 作为新的 **php:8.1** 模块流。

使用 **PHP 8.1**，您可以：

- 使用枚举 (Enums) 功能，定义仅限于离散值数的自定义类型
- 使用 **readonly** 修饰符声明属性，以防止在初始化后修改属性
- 使用光纤、全堆栈、中断功能

要安装 **php:8.1** 模块流，请使用：

```
# dnf module install php:8.1
```

有关 RHEL 9 上 PHP 用法的详情，请参阅 使用 PHP 脚本语言。

(BZ#2070040)

新模块流：ruby:3.1

RHEL 9.1 在新的 **ruby:3.1** 模块流中引入了 **Ruby 3.1.2**。与 RHEL 9.0 提供的 **Ruby 3.0** 相比，这个版本提供了很多性能改进、程序错误和安全修复以及新功能。

主要改进包括：

- **Interactive Ruby** (IRB)工具现在提供自动完成功能以及文档对话框
- 新的 **debug** gem，它替换了 **lib/debug.rb**，提供了改进的性能，并支持远程调试和多进程/多线程调试
- **error_highlight** gem 现在在 backtrace 中提供精细的错误位置
- 现在可以省略哈希文本数据类型和关键字参数中的值
- pin 运算符(**^**)现在接受模式匹配中的表达式
- 现在，可以在单行模式匹配省略括号
- YJIT 一种新的实验性进程内实时(JIT)编译器，现在在 AMD 和 Intel 64 位构架上提供
- 已引进了 **TypeProf For IDE** 工具，这是 IDE 中 **Ruby** 代码的实验性静态类型分析工具。

以下性能改进已在基于方法的实时编译器(MJIT)中实现了：

- 对于像 **Rails** 这样的工作负载，默认的最大 JIT 缓存值从 100 增加到 10000
- 当启用了 class 事件的 **TracePoint** 时，使用 JIT 编译的代码将不再被取消

其他显著变化包括：

- **tracer.rb** 文件已被删除
- 自版本 4.0 起，**Psych** YAML 解析器默认使用 **secure_load** 方法

要安装 **ruby:3.1** 模块流，请使用：

```
# dnf module install ruby:3.1
```

(BZ#2063773)

httpd 更新到版本 2.4.53

Apache HTTP 服务器已更新至 2.4.53 版本，它比 RHEL 9.0 发布的版本 2.4.51 提供了程序错误修正、功能增强和安全修复。

mod_proxy 和 **mod_proxy_connect** 模块中的显著变化包括：

- **mod_proxy** ：控制器名称的长度限制已增加
- **mod_proxy** ：您现在可以选择性地为后端和前端配置超时
- **mod_proxy** ：现在您可以通过设置 **SetEnv proxy-nohalfclose** 参数来禁用 TCP 连接重定向
- **mod_proxy** 和 **mod_proxy_connect** ：禁止在将状态代码发送到客户端后更改状态代码

另外，在表达式 API 中添加了新的 **ldap** 功能，这有助于防止 LDAP 注入漏洞。

(BZ#2079939)

httpd 配置中 LimitRequestBody 指令的新默认值

要修复 CVE-2022-29404，Apache HTTP 服务器中的 **LimitRequestBody** 指令的默认值已从 **0** (无限)变为 1 GiB。

在 **httpd** 配置文件中没有明确指定 **LimitRequestBody** 的值系统上，更新 **httpd** 软件包会将 **LimitRequestBody** 设为默认值 1 GiB。因此，如果 HTTP 请求正文的总大小超过这个 1 GiB 默认限制，则 **httpd** 会返回 **413 Request Entity Too Large** 错误码。

如果 HTTP 请求消息正文的新默认允许的大小不满足您的用例，请在相应的上下文中（服务器、每目录、每文件或每位置）更新您的 **httpd** 配置文件，并以字节为单位设置您的首选限制。例如，要设置一个新的 2 GiB 限制，请使用：

```
LimitRequestBody 2147483648
```

已被配置为使用 **LimitRequestBody** 指令的任何显式值的系统不受此更改的影响。

(BZ#2128016)

新软件包：httpd-core

从 RHEL 9.1 开始，所有基本文件的 **httpd** 二进制文件已移至新的 **httpd-core** 软件包，以便在需要基本 **httpd** 功能的情况下限制 Apache HTTP 服务器的依赖项，如容器中。

httpd 软件包现在提供 **systemd**- 相关文件，包括 **mod_systemd**、**mod_brotli** 以及文档。

在这个版本中，**httpd** 软件包不再提供 **httpd** 模块魔法号 (MMN) 值。相反， **httpd-core** 软件包现在提供 **httpd-mmn** 值。因此，无法从 **httpd** 软件包获取 **httpd-mmn**。

要获得安装的 **httpd** 二进制文件的 **httpd-mmn** 值，您可以使用 **apxs** 二进制文件，这是 **httpd-devel** 软件包的一部分。要获取 **httpd-mmn** 值，请使用以下命令：

```
# apxs -q HTTPD_MMN
20120211
```

(BZ#2065677)

pcre2 更新到版本 10.40

提供 Perl 兼容正则表达式库 v2 的 **pcre2** 软件包已更新至版本 10.40。

在这个版本中，与 **Perl 5.32** 中的相应变化一致，在 lookaround 断言时使用 **\K** 转义序列被禁止。如果依赖以前的行为，您可以使用 **PCRE2_EXTRA_ALLOW_LOOKAROUND_BSK** 选项。请注意，当设定这个选项时，**\K** 只接受正的断言，但会在负断言中忽略。

(BZ#2086494)

4.14. 编译器和开发工具

RHEL 9.1 提供了更新的 GCC 编译器。

系统 GCC 编译器版本 11.2.1 已更新，在上游 GCC 中包括大量程序错误修复和增强。

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。

有关使用信息，请参阅 RHEL 9 中开发 C 和 C++ 应用程序 。

(BZ#2063255)

新的 GCC 工具集 12

GCC Toolset 12 是一个编译器工具集，提供了开发工具的最新版本。它在 **AppStream** 存储库中以软件集合的形式作为应用程序流提供。

GCC 编译器已更新至版本 12.1.1，它提供了上游 GCC 中提供的很多程序错误修复和增强。

GCC Toolset 12 提供了以下工具和版本：

工具	版本
GCC	12.1.1
GDB	11.2
binutils	2.35
dwz	0.14
annobin	10.76

要安装 GCC Toolset 12，以 root 用户身份运行以下命令：

```
# dnf install gcc-toolset-12
```

要从 GCC Toolset 12 运行工具：

```
$ scl enable gcc-toolset-12 tool
```

要运行一个 shell 会话，其中 GCC Toolset 12 中的工具版本会覆盖这些工具的系统版本：

```
$ scl enable gcc-toolset-12 bash
```

如需更多信息，请参阅 GCC Toolset 12。

(BZ#2077465)

GCC Toolset 12：Anobin 更新到版本 10.76

在 GCC Toolset 12 中，Anobin 软件包已更新至版本 10.76。

重要的程序错误修复和增强包括：

- anocheck 的新命令行选项告诉它避免使用 **debuginfod** 服务（如果它无法用其他方式查找调试信息）。使用 **debuginfod** 为 anocheck 提供了更多信息，但如果 **debuginfod** 服务器不可用，它也会导致 annocheck 的性能下降。
- Annobin 源现在可以使用 **meson** 和 **ninja** 来构建，而不是根据需要进行配置和制作。
- Annocheck 现在支持 Rust 1.18 编译器构建的二进制文件。

另外，已在 Annobin 的 GCC 工具集 12 版本中报告了以下已知问题：

在某些情况下，编译可能会失败，并显示类似以下内容的错误消息：

```
cc1: fatal error: inaccessible plugin file
opt/rh/gcc-toolset-12/root/usr/lib/gcc/architecture-linux-gnu/12/plugin/gcc-annobin.so
expanded from short plugin name gcc-annobin: No such file or directory
```

要临时解决这个问题，请在 plugins 目录中创建一个从 **annobin.so** 到 **gcc-annobin.so** 的符号链接：

```
# cd /opt/rh/gcc-toolset-12/root/usr/lib/gcc/architecture-linux-gnu/12/plugin
# ln -s annobin.so gcc-annobin.so
```

其中 *architecture* 被正在使用的构架替换：

- **aarch64**
- **i686**
- **ppc64le**
- **s390x**
- **x86_64**

(BZ#2077438)

GCC 工具集 12：binutils rebase 到版本 2.38

在 GCC 工具集 12 中，**binutils** 软件包已更新至版本 2.38。

重要的程序错误修复和增强包括：

- **binutils** 软件包中的所有工具现在支持显示或警告存在多字节字符的选项。
- **readelf** 和 **objdump** 工具现在默认遵循指向各个 **debuginfo** 文件的任何链接。可以使用 **readelf** 的 **--debug-dump=no-follow-links** 选项或 **objdump** 的 **--dwarf=no-follow-links** 选项来禁用此行为。

(BZ#2077445)

GCC 12 和更高版本支持 _FORTIFY_SOURCE 级别 3

有了此增强，用户在使用 GCC 版本 12 或更高版本构建时，可以在编译器命令行中使用 **-D_FORTIFY_SOURCE=3** 来构建应用程序。**_FORTIFY_SOURCE** 级别 3 提高了源代码强化的覆盖率，从而提高了在编译器命令行中使用 **-D_FORTIFY_SOURCE=3** 构建的应用的安全性。这在 GCC 版本 12 及更高版本中支持，RHEL 9 的所有 Clang 都带有 **__builtin_dynamic_object_size** 内置。

(BZ#2033683)

DNS stub 解析器选项现在支持 no-aaaa 选项

有了这个增强，**glibc** 现在识别 **/etc/resolv.conf** 和 **RES_OPTIONS** 环境变量中的 **no-aaaa** stub 解析器选项。当此选项处于活跃状态时，不会通过网络发送 AAAA 查询。系统管理员可以出于诊断目的禁用 AAAA DNS 查询，例如，例如排除仅在 IPv4网络上的多余查询不会导致 DNS 问题。

(BZ#2096191)

添加了对 IBM Z 系列 z16 的支持

现在，对使用 **IBM z16** 平台设置的 **s390** 指令提供支持。**IBM z16** 在 **glibc** 中提供两个额外的硬件功能，即 **HWCAP_S390_VXRS_PDE2** 和 **HWCAP_S390_NNPA**。现在，应用程序可以使用这些功能提供优化的库和功能。

(BZ#2077838)

应用程序可以通过新的 glibc 接口使用可重启序列功能

要加快 **sched_getcpu** 功能（特别是在 aarch64 中），需要在 **glibc** 中使用可重启的序列 (rseq) 内核功能。为了允许应用程序持续使用共享的 rseq 区域，**glibc** 现在提供 **__rseq_offset**、**__rseq_size** 和 **__rseq_flags** 符号，这些符号在 **glibc** 2.35 上游版本中被第一次添加。在这个版本中，**sched_getcpu** 功能的性能会增加，应用程序现在可以通过新的 **glibc** 接口使用可重启的序列功能。

(BZ#2085529)

GCC 工具集 12：GDB 已 rebase 到版本 11.2

在 GCC 工具集 12 中，GDB 软件包已更新至版本 11.2。

重要的程序错误修复和增强包括：

- 对 64 位 ARM 架构内存标记 (MTE) 的新支持。请参阅带有 **memory-tag** 前缀的新命令。
- **-break-insert** 和 **-dprintf-insert** 的 **--qualified** 选项。这个选项查找用户事件位置的确切匹配，而不是在所有范围内搜索。
  例如，**break --qualified *foo*** 将在全局范围内查找名为 *foo* 的符号。没有 **--qualified**，GDB 将在所有范围内搜索具有该名称的符号。
- **--force-condition**: 任何提供的条件都会被定义，即使当前无效也是如此。
- **-break-condition --force**: 与 MI 命令类似。
- **-file-list-exec-source-files** 接受可选 **REGEXP** 来限制输出。
- **.gdbinit** 搜索路径包含配置目录。顺序是：
  a. **$XDG_CONFIG_HOME/gdb/gdbinit**
  b. **$HOME/.config/gdb/gdbinit**

c. **$HOME/.gdbinit**

- 支持 **~/.config/gdb/gdbearlyinit** 或 **~/.gdbearlyinit**。
- **-eix** 和 **-eiex** 早期初始化文件选项.

终端用户界面(TUI)：

- 支持终端用户界面(TUI)窗口中的鼠标操作。
- 不在聚焦窗口上操作的组合键现在传给 GDB。

新命令：

- **show print memory-tag-violations**
- **set print memory-tag-violations**
- **memory-tag show-logical-tag**
- **memory-tag with-logical-tag**
- **memory-tag show-allocation-tag**
- **memory-tag check**
- **show startup-quietly** 和 **set startup-quietly** ：一种在 GDB 脚本中指定 **-q** 或 **-quiet** 的方法。仅在早期初始化文件中有效。
- **show print type hex** 和 **set print type hex** ：告诉 GDB 以十六进制形式而不是十进制形式打印结构成员的大小或偏移量。
- **show python ignore-environment** 和 **set python ignore-environment** ：如果启用了，GDB 的 Python 解释器回忽略 Python 环境变量，就像将 **-E** 传给 Python 可执行文件。仅在早期初始化文件中有效。
- **show python dont-write-bytecode** 和 **set python dont-write-bytecode**: 如果为 **off**，则这些命令会阻止 GDB 的 Python 解释器编写导入模块的字节代码编译的对象，就像将 **-B** 传给 Python 可执行文件一样。仅在早期初始化文件中有效。

更改的命令：

- **break *LOCATION* if *CONDITION***：如果 *CONDITION* 无效，则 GDB 会拒绝设置断点。**-force-condition** 选项会覆盖它。
- ***CONDITION* -force N COND** ：与前面的命令相同。
- **inferior [ID]**: 当 ID 被省略时，这个命令会打印有关当前 inferior 的信息。否则，没有变化。
- **ptype[*/FLAGS*] TYPE | EXPRESSION**: 在打印 struct 成员的大小和偏移时使用 **/x** 标记来使用十六进制表示法。使用 **/d** 标志来做同样的事情，但使用十进制。
- **info sources** ：输出已重构。

Python API：

- inferior 对象包含一个只读 **connection_num** 属性。

- 新的 **gdb.Frame.level()** 方法。
- 新的 **gdb.PendingFrame.level()** 方法。
- 忽略了 **gdb.BreakpoiontEvent** 而不是 **gdb.Stop**。

(BZ#2077494)

GDB 支持 Power 10 PLT 指令

GDB 现在支持 Power 10 PLT 指令。在这个版本中，用户可以步骤到共享库功能，并使用 GDB 版本 10.2-10 及之后的版本检查堆栈后端。

(BZ#1870017)

dyninst 更新到版本 12.1

dyninst 软件包已更新至 12.1 版本。重要的程序错误修复和增强包括：

- **glibc-2.35** 多个命名空间的初始支持
- DWARF 并行解析的并发修复
- 更好地支持 **CUDA** 和 **CDNA2** GPU 二进制文件
- 更好地支持 IBM POWER 系统 (little endian) 注册访问
- 更好地支持 PIE 二进制文件
- 更正了捕获块的解析
- 更正了对 64 位 Arm (**arch64**) 浮动点注册的访问

(BZ#2057675)

新文件集 /etc/profile.d/debuginfod.*

添加了用于激活机构调试信息服务的新文件集。要获得系统范围的 **debuginfod** 客户端激活，您必须将 URL 添加到 **/etc/debuginfod/FOO.urls** 文件中。

(BZ#2088774)

Rust Toolset rebase 到版本 1.62.1

Rust Toolset 已更新到版本 1.62.1。主要变更包括：

- 解构赋值允许模式在赋值的左侧赋值现有的变量。例如，元组赋值可以交换变量：**(a、b) =(b、a)**
- 现在，使用 **core::arch::asm!** 宏在 64 位 x86 和 64 位 ARM 上支持内联 assembly。请参阅参考文档中的 "Inline assembly" 章节，**/usr/share/doc/rust/html/reference/inline-assembly.html** (https://doc.rust-lang.org/reference/inline-assembly.html)
- Enums 现在可以使用显式注解 **#[default]** 变体生成 **Default** trait。
- **Mutex** 、**CondVar** 和 **RwLock** 现在使用基于自定义的 **futex** 实现，而非 pthreads，Rust 语言保证提供了新的优化。
- Rust 现在支持 **main** 中的自定义退出代码，包括实现新稳定的 **Termination** 特征的用户定义的类型。

- cargo 支持更多对依赖项功能的控制。**dep:** 前缀可以在不公开为功能的情况下指向可选的依赖项，如果依赖项在其它地方启用了（如 **package-name?/feature-name**），**?** 才启用依赖项功能。
- cargo 有一个新的 **cargo add** 子命令，用于向 **Cargo.toml** 添加依赖项。
- 详情请查看上游发布公告系列：
  - 宣布 Rust 1.59.0
  - 宣布 Rust 1.60.0
  - 宣布 Rust 1.61.0
  - 宣布 Rust 1.62.0
  - 宣布 Rust 1.62.1

(BZ#2075337)

LLVM Toolset 更新到版本 14.0.6

LLVM Toolset 更新到版本 14.0.6。主要变更包括：

- 在 64 位 x86 上，添加了对 **AVX512-FP16** 指令的支持。
- 添加了对 Armv9-A、Armv9.1-A 和 Armv9.2-A 架构的支持。
- 在 PowerPC 上，添加了 **__ibm128** 类型来代表 IBM double-double 格式，也称为 **__attribute__(mode (IF))**。

clang 更改了：

- 现在为 **C++2b** 实现了 **if consteval**。
- 在 64 位 x86 上，添加了对 **AVX512-FP16** 指令的支持。
- 对处于实验状态的 OpenCL 2021 的 OpenCL C 3.0 和 **C++** 的支持。
- 现在 **-E -P** 预处理器输出始终省略空白行，匹配 GCC 行为。以前，输出中可能会出现高达 8 个空行。
- 不仅仅支持 C89，还支持 **C99** 和更高标准的 **-Wdeclaration-after-statement**，匹配 GCC 的行为。值得注意的用例是支持禁止混合声明和代码的风格指南，但希望迁移到新的 C 标准。

如需更多信息，请参阅 LLVM 工具集 和 Clang 上游发行注记。

(BZ#2061041)

Go Toolset 更新到版本 1.18.2

Go Toolset 已更新到版本 1.18.2。

主要变更包括：

- 在保持与之前版本的 Go 的向后兼容性的同时引入一般性。
- 新的 fuzzing 库。

- 新的 **debug/buildinfo** 和 **net/netip** 软件包。
- **go get** 工具不再构建或安装软件包。现在，它只会处理 **go.mod** 中的依赖项。
- 如果主模块的 **go.mod** 文件指定了 **go 1.17** 或更高版本，则在没有指定任何参数的情况下运行 **go mod download** 命令只会下载主模块的 **go.mod** 文件中明确需要的模块的源代码。要下载用于传输依赖项的源代码，请使用 **go mod download all** 命令。
- **go mod vendor** 子命令现在支持 **-o** 选项来设置输出目录。
- **go mod tidy** 命令现在为需要其源代码的模块保留额外校验和的 **go.sum** 文件中，以验证构建列表中只有一个模块提供每个导入的软件包。这个更改不适用于主模块的 **go.mod** 文件中的 Go 版本。

(BZ#2075169)

新模块流：maven:3.8

RHEL 9.1 引进了 **Maven 3.8** 作为新模块流。

要安装 **maven:3.8** 模块流，请使用：

```
# dnf module install maven:3.8
```

(BZ#2083112)

.NET 版本 7.0 可用

Red Hat Enterprise Linux 9.1 带有 **.NET** 版本 7.0。主要改进包括：

- 支持 IBM Power (**ppc64le**)

如需更多信息，请参阅 .NET 7.0 RPM 软件包 和 .NET 7.0 容器发行注记。

(BZ#2112027)

4.15. 身份管理

SSSD 现在支持 SID 请求的内存缓存

有了这个增强，SSSD 现在支持 SID 请求的内存缓存，它们是按 SID 查询的 GID 和 UID，反之亦然。内存缓存提高了性能，例如，当向或从 Samba 服务器拷贝大量文件时。

(JIRA:RHELPLAN-123369)

ipaservicedelegationtarget 和 ipaservicedelegationrule Ansible 模块现在可用

现在，您可以使用 **ipaservicedelegationtarget** 和 **ipaservicedelegationrule ansible-freeipa** 模块，将 web 控制台客户端配置为允许使用智能卡验证的 Identity Management (IdM)用户来执行以下操作：

- 在运行 web 控制台服务的 RHEL 主机上使用 **sudo**，而无需再次进行身份验证。
- 使用 **SSH** 访问远程主机并访问主机上的服务，而无需再次进行身份验证。

ipaservicedelegationtarget 和 **ipaservicedelegationrule** 模块使用 Kerberos **S4U2proxy** 功能（也称为受限委托）。IdM 通常使用此功能来允许 Web 服务器框架为用户获取 LDAP 服务票据。IdM-AD 信任系统使用该功能获取 cifs 主体。

(JIRA:RHELPLAN-117109)

SSSD 支持 FAST 的匿名 PKINIT

在这个版本中，SSSD 支持通过 Secure Tunneling (FAST) （在 Active Directory 中称为 Kerberos armoring）实现灵活的身份验证。到目前为止，要使用 FAST，需要一个 Kerberos keytab 来请求所需的凭证。现在，您可以使用匿名 PKINIT 创建此凭据缓存来建立 FAST 会话。

要启用匿名 PKINIT，请执行以下步骤：

1. 在 **sssd.conf** 文件的 **[domain]** 部分中，将 **krb5_fast_use_anonymous_pkinit** 设置为 **true**。
2. 重启 SSSD。
3. 在 IdM 环境中，您可以以 IdM 用户身份登录，验证使用匿名 PKINIT 来构建 FAST 会话。一个带有 FAST ticket 的缓存文件会被创建，**Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS** 表示使用了匿名 PKINIT：

```
klist /var/lib/sss/db/fast_ccache_IPA.VM
Ticket cache: FILE:/var/lib/sss/db/fast_ccache_IPA.VM
Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS
Valid starting Expires Service principal
03/10/2022 10:33:45 03/10/2022 10:43:45 krbtgt/IPA.VM@IPA.VM
```

(JIRA:RHELPLAN-123368)

IdM 现在支持随机序列号

在这个版本中，Identity Management (IdM) 包含 **dogtagpki 11.2.0**，它允许您使用 Random Serial Numbers 版本 3 (RSNv3)。在运行 **ipa-server-install** 或 **ipa-ca-install** 时，您可以使用 **--random-serial-numbers** 选项启用 RSNv3。启用 RSNv3 后，IdM 为 PKI 中的证书和请求生成完全随机的序列号，而不管理范围。使用 RSNv3，您可以避免在大型 IdM 安装中进行范围管理，并防止重新安装 IdM 时出现常见的冲突。

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。

(BZ#747959)

IdM 现在支持在用户密码过期后对允许的 LDAP 绑定数量的限制

有了这个增强，当身份管理(IdM)用户的密码已过期时，您可以设置允许 LDAP 绑定的数量：

-1
: 在用户必须重置密码之前，IdM 授予用户无限的 LDAP 绑定。这是默认值，与之前的行为匹配。

0
: 一旦密码过期，这个值会禁用所有 LDAP 绑定。生效时，用户必须立即重置其密码。

1-MAXINT
: 输入的值可以完全允许许多绑定后过期。

该值可以在全局密码策略和组策略中设置。

请注意，计数为每个服务器存储。

要让用户重置自己的密码，他们需要与其当前到期的密码绑定。如果用户已耗尽所有部署后绑定，则必须管理重置密码。

(BZ#2091988)

新的 ipasmartcard_server 和 ipasmartcard_client 角色

有了这个更新，**ansible-freeipa** 软件包提供了 Ansible 角色来配置身份管理(IdM)服务器和客户端，以进行智能卡验证。**ipasmartcard_server** 和 **ipasmartcard_client** 角色替代了 **ipa-advise** 脚本来自动化和简化集成。与其它 **ansible-freeipa** 角色使用相同的清单和命名方案。

(BZ#2076567)

IdM 现在支持使用 Windows Server 2022 配置 AD Trust

有了这个增强，您可以在身份管理(IdM)域和使用运行 Windows Server 2022 的域控制器的活动目录林之间建立跨林信任。

(BZ#2122716)

ipa-dnskeysyncd 和 ipa-ods-exporter debug 消息默认不再记录到 /var/log/messages

在以前的版本中，**ipa-dnskeysyncd**（负责 LDAP-to-OpenDNSSEC 同步）和 **ipa-ods-exporter**（Identity Management (IdM) OpenDNSSEC exporter 服务）将所有 debug 信息默认记录到 **/var/log/messages**。因此，日志文件的显著增加。在这个版本中，您可以通过在 **/etc/ipa/dns.conf** 文件中设置 **debug=True** 来配置日志级别。如需更多信息，请参阅 IdM 配置文件的 **default.conf (5)** 手册页。

(BZ#2083218)

samba 更新到版本 4.16.1

samba 软件包升级至上游版本 4.16.1，它提供程序错误修复和增强：

- 默认情况下，**smbd** 进程会根据需要自动启动新的 **samba-dcerpcd** 进程，来为分布式计算环境/远程过程调用(DCERPC)提供服务。请注意，Samba 4.16 及更高版本始终需要 **samba-dcerpcd** 来使用 DCERPC。如果您在 **/etc/samba/smb.conf** 文件中的 **[global]** 部分中禁用了 **rpc start on require helpers** 设置，则您必须创建一个 **systemd** 服务单元来在独立模式下运行 **samba-dcerpcd**。
- Cluster Trivial Database (CTDB) **recovery master** 角色已被重命名为 **leader**。因此，以下 **ctdb** 子命令被重命名为：
  - **recmaster** 变为 **leader**
  - **setrecmasterrole** 变为 **setleaderrole**
- CTDB **recovery lock** 配置已被重命名为 **cluster lock**。
- CTDB 现在使用领导广播和关联的超时来确定是否需要选举。

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。请注意，红帽不支持降级 **tdb** 数据库文件。

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

有关显著变化的更多信息，请在更新前阅读 上游发行注记。

(BZ#2077487)

SSSD 现在支持直接与 Windows Server 2022 集成

有了这个增强，您可以使用 SSSD 将 RHEL 系统直接与使用运行 Windows Server 2022 的域控制器的活动目录林集成。

(BZ#2070793)

改进了 SSSD 多线程性能

在以前的版本中，SSSD 从多线程应用程序（如 Red Hat Directory Server 和 Identity Management）序列化并行请求。在这个版本中修复了所有 SSSD 客户端库，如 **nss** 和 **pam**，因此它们不会序列化请求，因此允许多个线程的请求并行执行以提高性能。要启用以前的序列化行为，请将环境变量 **SSS_LOCKFREE** 设置为 **NO**。

(BZ#1978119)

目录服务器现在支持取消 Auto Membership 插件任务。

在以前的版本中，如果目录服务器有复杂的配置（大组、复杂规则以及与其他插件的交互），则 Auto Membership 插件任务可以在服务器上产生高 CPU 使用率。有了这个增强，您可以取消 Auto Membership 插件任务。因此，性能问题不会再发生。

(BZ#2052527)

目录服务器现在在使用 ldapdelete 时支持递归删除操作

在这个版本中，Directory 服务器支持 **Tree Delete Control [1.2.840.113556.1.4.805]** OpenLDAP 控制。因此，您可以使用 **ldapdelete** 程序以递归方式删除父条目的子条目。

(BZ#2057063)

现在，您可以在目录服务器安装过程中设置基本复制选项

有了这个增强，您可以使用 **.inf** 文件，在实例安装过程中配置基本复制选项，如身份验证凭证和更改日志修剪。

(BZ#2057066)

目录服务器现在支持非 root 用户创建的实例

在以前的版本中，非 root 用户不能创建目录服务器实例。有了这个增强，非 root 用户可以使用 **dscreate ds-root** 子命令配置一个环境，在此环境中，**dscreate**，**dsctl**，**dsconf** 命令通常被用来创建和管理目录服务器实例。

(BZ#1872451)

pki 软件包重命名为 idm-pki

以下 **pki** 软件包现在被重命名为 **idm-pki**，以便更好地区分 IDM 软件包和 Red Hat 证书系统：

- **idm-pki-tools**
- **idm-pki-acme**
- **idm-pki-base**
- **idm-pki-java**

- **idm-pki-ca**
- **idm-pki-kra**
- **idm-pki-server**
- **python3-idm-pki**

(BZ#2139877)

4.16. 图形基础结构

Wayland 现在通过 Matrox GPU 启用

桌面会话现在启用使用 Matrox GPU 的 Wayland 后端。

在以前的版本中，由于性能和其他限制，V Wayland 已被 Matrox GPU 禁用。这个问题现已解决。

您仍然可以将桌面会话从 Wayland 切回到 Xorg。如需更多信息，请参阅 GNOME 环境概述。

(BZ#2097308)

现在支持 12 代 Intel Core GPU

此发行版本添加了对 12th Gen Intel Core CPU 的多个集成 GPU 的支持。这包括在以下 CPU 型号中的 Intel UHD Graphics 和 Intel Xe 集成的 GPU：

- Intel Core i3 12100T 到 Intel Core i9 12900KS
- Intel Pentium Gold G7400 和 G7400T
- Intel Celeron G6900 和 G6900T
- Intel Core i5-12450HX 到 Intel Core i9-12950HX
- Intel Core i3-1220P 到 Intel Core i7-1280P

(JIRA:RHELPLAN-135601)

支持新的 AMD GPU

此发行版本添加了对几个 AMD Radeon RX 6000 系列 GPU 和 AMD Ryzen 6000 系列 CPU 的集成图形的支持。

现在支持以下 AMD Radeon RX 6000 系列 GPU 模型：

- AMD Radeon RX 6400
- AMD Radeon RX 6500 XT
- AMD Radeon RX 6300M
- AMD Radeon RX 6500M

AMD Ryzen 6000 系列包括使用以下 CPU 模型找到的集成的 GPU：

- AMD Ryzen 5 6600U

- AMD Ryzen 5 6600H
- AMD Ryzen 5 6600HS
- AMD Ryzen 7 6800U
- AMD Ryzen 7 6800H
- AMD Ryzen 7 6800HS
- AMD Ryzen 9 6900HS
- AMD Ryzen 9 6900HX
- AMD Ryzen 9 6980HS
- AMD Ryzen 9 6980HX

(JIRA:RHELPLAN-135602)

4.17. WEB 控制台

Web 控制台中的更新进度页面现在支持自动重启选项

更新进度页面现在有一个 **Reboot after completion** 开关。这会在安装更新后自动重启系统。

(BZ#2056786)

4.18. RED HAT ENTERPRISE LINUX 系统角色

network RHEL 系统角色支持使用 nmstate API 的网络配置

有了此更新，**network** RHEL 系统角色支持通过 **nmstate** API 进行网络配置。用户现在可以将所需网络状态的配置直接应用到网络接口，而不是创建连接配置集。该功能还允许部分配置网络。因此，存在以下优点：

- 网络配置复杂性降低
- 应用网络状态更改的可靠方法
- 不需要跟踪整个网络配置

(BZ#2072385)

用户可以使用 network RHEL 系统角色，创建具有 IPoIB 功能的连接

network RHEL 系统角色的 **infiniband** 连接类型现在支持 Internet Protocol over Infiniband (IPoIB) 功能的互联网协议。要启用此功能，请为 **infiniband** 的 **p_key** 选项定义一个值。请注意，如果指定了 **p_key**，则 **network_connections** 变量的 **interface_name** 选项必须保留为未设置。以前的 **network** RHEL 系统角色的实现没有正确地验证 **p_key** 值和用于 **infiniband** 连接类型的 **interface_name** 选项。因此，IPoIB 功能在以前不能工作。如需更多信息，请参阅 **/usr/share/doc/rhel-system-roles/network/** 目录中的 README 文件。

(BZ#2086965)

HA 集群 RHEL 系统角色现在支持 SBD 隔离和 Corosync 设置的配置

HA Cluster 系统角色现在支持以下功能：

SBD 隔离

隔离是 HA 集群配置的重要组成部分。SBD 为节点提供了一种在需要隔离时提供可靠的自终止的方法。在无法实现传统隔离机制的环境中，SBD 隔离特别有用。现在，可以使用 HA 集群系统角色配置 SBD 隔离。

corosync 设置

HA Cluster 系统角色现在支持 Corosync 设置的配置，如传输、压缩、加密、链接、图腾和仲裁。当默认设置不合适时，需要使用这些设置来将集群的配置与客户的需要和环境匹配。

(BZ#2065337, BZ#2070452, BZ#2079626, BZ#2098212, BZ#2120709, BZ#2120712)

network RHEL 角色现在为路由规则配置网络设置

在以前的版本中，您可以根据数据包中的目标地址字段路由数据包，但您无法定义源路由和其他策略路由规则。在这个版本中，**network** RHEL 角色支持路由规则，以便用户可以控制数据包传输或路由选择。

(BZ#2079622)

previous:replaced 配置可让 firewall 系统角色将防火墙设置重置为默认值

管理不同机器集合（其中每台机器都有不同的预先存在的防火墙设置）的系统管理员，现在可以使用 **firewall** 角色中的 **previous: replaced** 配置，来确保所有机器都有相同的防火墙配置设置。**previous: replaced** 的配置可能会清除所有现有的防火墙设置，并使用一致的设置替换它们。

(BZ#2043010)

postfix RHEL 系统角色中用于覆盖以前配置的新选项

如果您管理一组具有 **postfix** 配置不一致的系统，则可能要在所有这些配置上保持一致。在这个版本中，您可以指定 **postfix_conf** 字典中的 **previous: replaced** 选项，以移除任何现有配置并在清 **postfix** 安装之上应用所需的配置。因此，您可以清除任何现有的 **postfix** 配置并确保所有被管理的系统上的一致性。

(BZ#2065383)

增强的 microsoft.sql.server RHEL 系统角色

以下新变量现在对 **microsoft.sql.server** RHEL 系统角色可用：

- 带有 **mssql_ha_** 前缀的变量用来控制配置高可用性集群。
- **mssql_tls_remote_src** 变量，用来在受管节点上搜索 **mssql_tls_cert** 和 **mssql_tls_private_key** 值。如果保持默认的 **false** 设置，则该角色会在控制节点上搜索这些文件。
- **mssql_manage_firewall** 变量用来自动管理防火墙端口。如果此变量被设为 **false**，则您必须手动启用防火墙端口。
- **mssql_pre_input_sql_file** 和 **mssql_post_input_sql_file** 变量来控制您要在角色执行前或之后运行的 SQL 脚本。这些新变量替换以前的 **mssql_input_sql_file** 变量，这不会影响 SQL 脚本执行的时间。

(BZ#2066337)

logging RHEL 系统角色支持文件输入中的 startmsg.regex 和 endmsg.regex

有了这个更新，您现在可以使用正则表达式过滤来自文件的日志消息。**startmsg_regex** 和 **endmsg_regex** 选项现在包含在文件的输入中。**startmsg_regex** 代表与消息的开始部分匹配的正则表达

式，而 **endmsg_regex** 代表与消息的最后部分匹配的正则表达式。现在，您可以根据日期、优先级和严重性等属性过滤消息。

(BZ#2112145)

sshd RHEL 系统角色验证用于置入目录的 include 指令

RHEL 9 上的 **sshd** RHEL 系统角色只管理置入目录中的文件，但之前不验证目录是否包含在主 **sshd_config** 文件中。有了这个更新，该角色会验证 **sshd_config** 是否包含置入目录的 include 指令。因此，该角色可以更可靠地应用提供的配置。

(BZ#2052081)

sshd RHEL 系统角色可以通过 /etc/ssh/sshd_config 进行管理

应用到 RHEL 9 受管节点的 **sshd** RHEL 系统角色将 SSHD 配置放在置入目录（默认为**/etc/ssh/sshd_config.d/00-ansible_system_role.conf** ）中。在以前的版本中，对 **/etc/ssh/sshd_config** 文件的任何更改都会覆盖 **00-ansible_system_role.conf** 中的默认值。在这个版本中，您可以使用 **/etc/ssh/sshd_config** 而不是 **00-ansible_system_role.conf** 管理 SSHD，同时保留 **00-ansible_system_role.conf** 中的系统默认值。

(BZ#2052086)

metrics 角色在其管理的配置文件中使用 "Ansible_managed" 注释

在这个版本中，**metrics** 角色使用 Ansible 标准 **ansible_managed** 变量将"Ansible managed"注释插入到配置文件。注释指示不应直接编辑配置文件，因为 **metrics** 角色将覆盖该文件。因此，配置文件包含一个声明，表示配置文件由 Ansible 管理。

(BZ#2065392)

storage RHEL 系统角色现在支持管理池成员

storage RHEL 系统角色现从现有 LVM 池中添加或删除磁盘，而无需先删除池。要增大池容量， **storage** RHEL 系统角色可以在池中添加新磁盘，并目前在池中为另一个用途分配的磁盘。

(BZ#2072742)

storage RHEL 系统角色现在支持精简置备的卷

storage RHEL 系统角色现在可以创建和管理精简置备的 LVM 逻辑卷(LV)。精简配置的 LV 会在编写时分配，在为精简置备的 LV 提供的物理存储创建卷时具有更大的灵活性。LVM 精简配置还允许创建效率更高的快照，因为对于精简 LV 的通用数据块及其任何快照都是共享的。

(BZ#2072745)

storage RHEL 系统角色对缓存的卷提供更好的支持

storage RHEL 系统角色现在可以将缓存添加到现有 LVM 逻辑卷上。LVM 缓存可用于通过在较小的、更快速的设备上（如 SSD）临时存储 LV 数据的子集来提高较慢的逻辑卷的性能。这提高了创建缓存卷的支持，方法是允许将缓存（附加）缓存添加到现有的未缓存卷中。

(BZ#2072746)

logging RHEL 系统角色现在支持 template,severity and facility 选项

logging RHEL 系统角色现在对文件输入提供新的有用的 **severity** 和 **facility** 选项，以及对文件和转发输出提供新的 **template** 选项。使用 **template** 选项使用 **traditional** 参数指定传统时间格式，使用参数 **syslog** 指定 syslog 协议 23 格式，使用参数 **modern** 指定现代风格格式。现在，您可以使用 **logging** 角

色根据严重性和工具过滤，并从模板创建输出格式。

(BZ#2075119)

RHEL 系统角色现在也在禁用了事实收集的 playbook 中提供

由于性能或其他原因，可能会在您的环境中禁用 Ansible 事实收集。在以前的版本中，无法在这样的配置中使用 RHEL 系统角色。有了这个更新，系统会在您的配置中检测 **ANSIBLE_GATHERING=explicit** 参数，在 playbook 中检测 **gather_facts: false** 参数，并使用 **setup:** 模块来仅收集给定角色所需要的事实（如果在事实缓存中没有）。

注意

如果您因为性能而禁用了 Ansible 事实收集，您可以启用 Ansible 事实缓存，这会导致从来源检索它们的性能。

(BZ#2078989)

存储角色现在默认具有较少的详细程度

存储角色输出现在默认为详细。在这个版本中，用户可以提高存储角色输出的详细程度，仅在使用 Ansible 详细程度 1 或更高版本时生成调试输出。

(BZ#2079627)

在配置 masquerade 或 icmp_block_inversion时，firewall RHEL 系统角色不需要 state 参数

在配置自定义防火墙区时，变量 **masquerade** 和 **icmp_block_inversion** 是布尔值设置。**true** 值表示 **state: present** 和 **false** 值表示 **state: absent**。因此，配置 **masquerade** 或 **icmp_block_inversion** 时不需要 **state** 参数。

(BZ#2093423)

现在，您可以在 firewall RHEL 系统角色中使用 absent 和 present 状态添加、更新或删除服务

在这个版本中，您可以使用 **present** 状态来添加端口、模块、协议、服务和目标地址，或使用 **absent** 状态来删除它们。请注意，要在 **firewall** RHEL 系统角色中使用 **absent** 和 **present** 状态，请将 **permanent** 选项设置为 **true**。将 **permanent** 选项设置为 **true** 时，状态设置将应用至更改后，并且保持不受角色重新加载的影响。

(BZ#2100292)

firewall 系统角色可以使用 PCI 设备 ID 向区添加或删除接口

使用 PCI 设备 ID，**firewall** 系统角色现在可以为区分配或删除网络接口。在以前的版本中，如果只有 PCI 设备 ID 已知而不是接口名称，用户必须首先识别对应的接口名称才能使用 **firewall** 系统角色。在这个版本中，**firewall** 系统角色可以使用 PCI 设备 ID 来管理区中的网络接口。

(BZ#2100942)

firewall RHEL 系统角色可以提供 Ansible 事实

有了此增强，您现在可以通过在 playbook 中包含不带参数的 **firewall:** 变量，来从所有系统中收集 **firewall** RHEL 系统角色的 Ansible 事实。要收集更多 Ansible 事实的详细版本，请使用 **detailed: true** 参数，例如：

```
vars:
  firewall:
    detailed: true
```

(BZ#2115154)

将 seuser 和 selevel 的设置添加到 selinux RHEL 系统角色中

有时，在设置 SELinux 上下文文件系统映射时，需要设置 **seuser** 和 **selevel** 参数。在这个版本中，您可以使用 **selinux_fcontext** 中的 **seuser** 和 **selevel** 可选参数来指定 SELinux 上下文文件系统映射中的 SELinux 用户和级别。

(BZ#2115157)

用于设置自定义监听端口的新的 cockpit 系统角色变量

cockpit 系统角色引入了 **cockpit_port** 变量，它允许您设置默认的 9090 端口以外的自定义监听端口。请注意，如果您决定设置自定义监听端口，您也需要调整 SELinux 策略以允许 Web 控制台侦听该端口。

(BZ#2115152)

metrics 角色可以导出 postfix 性能数据

现在，您可以使用 **metrics** 角色中的新 **metrics_from_postfix** 布尔值变量记录和详细性能分析。在这个版本中，设置变量会在系统中启用 **pmdapostfix** 指标代理，生成关于 **postfix** 可用的统计信息。

(BZ#2051737)

postfix 角色在其管理的配置文件中使用 "Ansible_managed" 注释

postfix 角色生成 **/etc/postfix/main.cf** 配置文件。在这个版本中，**postfix** 角色使用 Ansible 标准 **ansible_managed** 变量将"Ansible managed"注释插入到配置文件。注释表示不应直接编辑配置文件，因为 **postfix** 角色可以覆盖该文件。因此，配置文件包含一个声明，表示配置文件由 Ansible 管理。

(BZ#2065393)

nbde-client RHEL 系统角色支持静态 IP 地址

在之前的 RHEL 版本中，重启具有静态 IP 地址并使用 **nbde_client** RHEL 系统角色配置的系统更改了系统的 IP 地址。在这个版本中，**nbde_client** 角色支持使用静态 IP 地址的系统，重启后它们的 IP 地址不会改变。

请注意，默认情况下，**nbde_client** 角色在引导时使用 DHCP，并在系统引导后切换到配置的静态 IP。

(BZ#2070462)

4.19. 虚拟化

RHEL web 控制台现在将 RHEL 作为 Download a OS VM 工作流的选项

有了这个增强，RHEL web 控制台支持使用默认的 **Download an OS** 工作流安装 RHEL 虚拟机。因此，您可以在 web 控制台中直接下载并安装 RHEL OS 作为虚拟机。

(JIRA:RHELPLAN-121982)

改进了 KVM 架构合规性

在这个版本中，KVM 管理程序的架构合规性已被增强并更严格。现在，管理程序已准备好解决未来对基于 Linux 和其他操作系统的更改。

(JIRA:RHELPLAN-117713)

ap-check 现在包括在 RHEL 9 中

mdevctl 工具现在提供了一个新的 **ap-check** 支持实用程序。您可以使用 **mdevctl** 永久配置加密适配器和域，允许将它们直通到虚拟机以及 **矩阵** 和 **vfio-ap** 设备。使用 **mdevctl** ，您无需在每次 IPL 后重新配置这些适配器、域和设备。此外，**mdevctl** 可以防止服务器发明其他方法来重新配置它们。

当对 **vfio-ap** 设备调用 **mdevctl** 命令时，会将新的 **ap-check** 支持工具作为 **mdevctl** 命令的一部分来调用，来对 **vfio-ap** 设备配置执行额外的有效性检查。

另外，**chzdev** 工具现在提供了管理系统范围 Adjunct Processor (AP)掩码设置的功能，它决定了 **vfio-ap** 设备可以使用哪些 AP 资源。使用时，**chzdev** 通过生成关联的 **udev** 规则来持久保留这些设置。使用 **lszdev**，您现在可以查询系统范围的 AP 掩码设置。

(BZ#1870699)

open-vm-tools 更新到 12.0.5

open-vm-tools 软件包升级至版本 12.0.5，它引入了大量的 bug 修复和新功能。最值得注意的是，增加了对通过客户机操作系统变量管理的 Salt Minion 工具的支持。

(BZ#2061193)

IBM Z 上的所选虚拟机现在可以使用大于 896 字节的内核命令行引导

在以前的版本中，如果虚拟机的内核命令行超过 896 字节，在 RHEL 9 IBM Z 主机上引导虚拟机 (VM) 始终会失败。在这个版本中，QEMU 模拟器可以处理大于 896 字节的内核命令行。现在，如果 VM 内核支持，您可以为具有非常长的内核命令行的虚拟机使用 QEMU 直接内核引导。具体来说，要使用命令行大于 896 字节，虚拟机必须使用 Linux 内核版本 5.16-rc1 或更高版本。

(BZ#2044218)

IBM Z 上的安全执行功能现在支持远程测试

IBM Z 架构上的安全执行功能现在支持远程测试。**pvattest** 实用程序可以创建远程的 attestation 请求，以验证启用了安全执行的虚拟机的完整性。

另外，现在可以使用 GISA 将中断注入带有安全执行（安全执行）的客户机。

(BZ#2001936, BZ#2044300)

使用多个线程的虚拟机内存预分配

现在，您可以在域 XML 配置中为虚拟机 (VM) 内存分配定义多个 CPU 线程，如下所示：

```
<memoryBacking>
  <allocation threads='8'/>
</memoryBacking>
```

这样可确保多个线程用于在启动虚拟机时分配内存页面。因此，配置了多个分配线程的虚拟机会显著提高，特别是当虚拟机分配了大量 RAM 且由巨页支持时。

(BZ#2064194)

RHEL 9 客户机现在支持 SEV-SNP

在使用 RHEL 9 作为客户机操作系统的虚拟机(VM)上，您现在可以使用具有安全嵌套分页(SNP)功能的 AMD 安全加密虚拟化(SEV)功能。除了其他好处外，SNP 通过改进内存完整性保护来增强 SEV，这有助于防止基于 hypervisor 的攻击，如数据重放或内存重新映射。请注意，要使 SEV-SNP 在 RHEL 9 虚拟机上工作，运行虚拟机的主机也必须支持 SEV-SNP。

(BZ#2169738)

4.20. 云环境中的 RHEL

cloud-init 的新 SSH 模块

有了这个更新，SSH 模块已被添加到 **cloud-init** 工具中，它会在实例创建过程中自动生成主机密钥。

请注意，有了这个变化，默认的 **cloud-init** 配置已被更新。因此，如果您有一个本地修改，请确保 /etc/cloud/cloud.cfg 包含 "ssh_genkeytypes: ['rsa', 'ecdsa', 'ed25519']" 行。

否则，**cloud-init** 会创建一个不能启动 **sshd** 服务的镜像。如果出现这种情况，请执行以下操作来临时解决这个问题：

1. 确保 **/etc/cloud/cloud.cfg** 文件包含以下行：

```
ssh_genkeytypes:  ['rsa', 'ecdsa', 'ed25519']
```

2. 检查实例中是否存在 **/etc/ssh/ssh_host_*** 文件。
3. 如果 **/etc/ssh/ssh_host_*** 文件不存在，请使用以下命令生成主机密钥：

```
cloud-init single --name cc_ssh
```

4. 重启 sshd 服务：

```
systemctl restart sshd
```

(BZ#2115791)

4.21. 容器

Container Tools 软件包已更新

包含 Podman、Buildah、Skopeo、crun 和 runc 工具的容器工具软件包现在可用。与之前的版本相比，这个更新提供了 bug 修复和增强的列表。

主要变更包括：

- **podman pod create** 命令现在支持设置 CPU 和内存限值。您可以为 pod 中的所有容器设置限制，pod 中的单个容器则可能有自己的限制。
- **podman pod clone** 命令创建现有 pod 的副本。
- **podman play kube** 命令现在支持使用 **BlockDevice** 和 **CharDevice** 卷的安全上下文设置。
- 由 **podman play kube** 创建的 Pod 现在可使用 **podman-kube@<service>.service** 管理（例如 **systemctl --user start podman-play-kube@$(systemd-escape my.yaml**）。
- **podman push** 和 **podman push manifest** 命令现在支持 sigstore 签名。

- Podman 网络现在可以使用 **podman network --opt isolate** 命令来隔离。

Podman 已升级至版本 4.2，要了解有关显著变化的更多信息，请参阅 上游发行注记。

(JIRA:RHELPLAN-118462)

GitLab Runner 现在可使用 Podman 在 RHEL 上可用

从 GitLab Runner 15.1 开始，您可以在 GitLab Runner Docker 执行器中使用 Podman 作为容器运行时。如需了解更多详细信息，请参阅 GitLab 的发行说明。

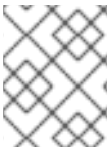
(JIRA:RHELPLAN-101140)

Podman 现在支持 --health-on-failure 选项

podman run 和 **podman create** 命令现在支持 **--health-on-failure** 选项，以确定何时容器状态不健康时要执行的操作。

--health-on-failure 选项支持四个操作：

- **none** ：不执行任何操作，这是默认操作。
- **kill** ：中断容器。
- **restart**: 重启容器。
- **stop**: 停止容器。



注意

不要将 **restart** 操作与 **--restart** 选项一起使用。在 systemd 单元内运行时，请考虑使用 **kill** 或 **stop** 操作而不是使用 systemd 的重启策略。

(BZ#2097708)

Netavark 网络堆栈现在可用

Netavark 堆栈是容器的网络配置工具。在 RHEL 9 中，Netavark 堆栈被全面支持并启用。

此网络堆栈具有以下功能：

- 使用 JSON 配置文件配置容器网络
- 创建、管理和删除网络接口，包括网桥和 MACVLAN 接口
- 配置防火墙设置，如网络地址转换(NAT)和端口映射规则
- IPv4 和 IPv6
- 改进了多个网络中容器的功能
- 使用 aardvark-dns 项目进行容器 DNS 解析

注意

您必须使用相同的 Netavark 堆栈和 **aardvark-dns** 权威 DNS 服务器。

(JIRA:RHELPLAN-132023)

新软件包：CRB 存储库中的 catatonit

现在，在 CodeReady Linux Builder (CRB) 仓库中提供了一个新的 **catatonit** 软件包。**catatonit** 软件包被用作容器的最小 init 程序，并可包含在应用容器镜像中。请注意，不支持 CodeReady Linux Builder 存储库中包含的软件包。

请注意，自 RHEL 9.0 起，**podman-catonit** 软件包可从 AppStream 存储库中获得。**podman-catatonit** 仅在 Podman 工具中使用。

(BZ#2074193)

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 9.1 一起发布的内核有显著变化的总结。这些更改包括添加或更新的 **proc** 条目、**sysctl** 和 **sysfs** 默认值、引导参数、内核配置选项或者任何可见的行为更改。

新内核参数

allow_mismatched_32bit_el0 = [ARM64]

使用这个参数，您可以允许 EL0 级别中不匹配的 32 位支持的系统运行 32 位应用程序。支持 32 位 EL0 的 CPU 集合由 **/sys/devices/system/cpu/aarch32_el0** 文件表示。另外，您可以限制热拔操作。
如需更多信息，请参阅文档 **/arm64/asymmetric-32bit.rst**。

arm64.nomte = [ARM64]

使用这个参数时，您可以无条件地禁用内存标记 (MTE) 支持。

i8042.probe_defer = [HW]

使用这个参数时，您可以允许对 **i8042** 探测错误进行延迟。

idxd.tc_override = [HW]

使用此参数（**<bool>** 格式），您可以允许覆盖该设备的默认流量类配置。
默认值为 **false** (**0**)。

kvm.eager_page_split = [KVM,X86]

通过这个参数，您可以控制 KVM 在脏日志记录期间主动分割所有巨页。Eager 页面分割通过消除写保护错误和内存管理单元 (MMU) 锁定争用而减少 vCPU 执行中断，而其他方式需要分割大页面。
很少执行写入或只写入虚拟机内存区域的虚拟机工作负载可从禁用 eager 页面拆分中受益，从而使巨页仍被用于读取。

eager 页面分割行为取决于启用或禁用 **KVM_DIRTY_LOG_INITIALLY_SET** 选项。

- 如果禁用，当脏日志在 **memslot** 中启用，， **memslot** 中的所有巨页都会积极分割。
- 如果启用，在 **KVM_CLEAR_DIRTY ioctl()** 系统调用期间执行 eager 页面分割，并且仅针对被清除的页面。
  Eager 页面分割目前只支持分割由两个维度分页(TDP) MMU 映射的巨页。

  默认值为 **Y** (**on**)。

kvm.nx_huge_pages_recovery_period_ms = [KVM]

使用这个参数时，您可以控制 KVM zaps 4 KiB 页面回巨页的时间周期。

- 如果值为非零的 **N**，KVM 会没每 **N** 毫秒的页面部分。
- 如果值为 **0**，KVM 根据比例选择一个周期，从而使页面平均在 1 小时后显示出来。
  默认值为 **0**。

l1d_flush = [X86,INTEL]

使用这个参数，您可以控制基于 L1D 的侦听漏洞的缓解方案。
某些 CPU 可能会容易受到 CPU 内部缓冲区的攻击，在某些情况下可以把信息转发到披露。在存在安全漏洞的处理器中，缓存侧频道攻击可以利用预测的数据转发，访问到应该无法直接访问到的数据。

可用选项是 **on**，这表示为**缓解措施启用接口**。

mmio_stale_data = [X86,INTEL]

使用这个参数，您可以控制 Processor Memory-mapped I/O (MMIO) Stale 数据漏洞的缓解方案。处理器 MMIO Stale 数据是一个漏洞，在 MMIO 操作后可以公开数据。公开的数据可能源自或结束与元数据服务器 (MDS) 和异步 Asynchronous Abort (TAA) 影响的相同 CPU 缓冲区。因此，与 MDS 和 TAA 类似，缓解方案是清除受影响的 CPU 缓冲区。

可用的选项有：

- **full**: 在存在安全漏洞的 CPU 上启用缓解方案
- **full,nosmt**: 在存在安全漏洞的 CPU 上启用缓解方案并禁用 SMT。
- **off**: 无条件禁用缓解方案
  在 MDS 或 TAA 受影响的机器上，活跃 MDS 或 TAA 缓解方案可以防止 **mmio_stale_data=off**，因为这些漏洞会使用相同的机制进行缓解。因此，为了禁用这个缓解方案，您还需要指定 **mds=off** 和 **tsx_async_abort=off**。

  不指定这个选项等同于 **mmio_stale_data=full**。

  如需更多信息，请参阅 **Documentation/admin-guide/hw-vuln/processor_mmio_stale_data.rst**。

random.trust_bootloader={on,off} = [KNL]

使用这个参数您可以启用或禁用引导装载程序传递的使用（如果可用）的信任，以完全看到内核的 CRNG。默认的行为由 **CONFIG_RANDOM_TRUST_BOOTLOADER** 选项控制。

rcupdate.rcu_task_collapse_lim = [KNL]

使用此参数，您可以设置在宽限期开始时提供的最大回调数，允许 RCU 任务类别折叠回使用单一回调队列。只有在 **rcupdate.rcu_task_enqueue_enqueue_lim** 选项被设置为默认值 **-1** 时，才会进行这种切换。

rcupdate.rcu_task_contend_lim = [KNL]

使用这个参数，您可以设置每个 jiffy 所需的最少回调锁定事件数量，从而导致 RCU 任务类型切换到每个 CPU 回调队列。只有在 **rcupdate.rcu_task_enqueue_enqueue_lim** 选项被设置为默认值 **-1** 时，才会进行这种切换。

rcupdate.rcu_task_enqueue_lim = [KNL]

使用此参数，您可以设置用于 RCU 类别的 RCU 任务系列的回调队列数量。您可以使用默认值 **-1** 自动调整回调队列的数量，并动态调整。
这个参数用于测试。

retbleed = [X86]

使用这个参数，您可以使用返回说明(RETBleed)漏洞来控制对 Arbitrary Speculative Code Execution 的缓解方案。可用的选项有：

- **off**: 不实施缓解方案
- **auto**: 自动选择缓解方案
- **auto,nosmt**: 会自动选择一个缓解方案，在完全缓解措施的情况下禁用 SMT （只在 Zen1 和更早的 STIBP 中）。
- **ibpb** ：减轻基本块边界上的短规范窗口。安全，对性能有最高的影响。
- **unret** ：force enable untrained return thunks, only effective on AMD f15h-f17h based systems。

- **unret,nosmt**: 与 **unret** 选项一样，当 STIBP 不可用时，将禁用 SMT。
  选择 **auto** 选项会根据 CPU，在运行时选择缓解方案。

  不指定这个选项等同于 **retbleed=auto**。

sev=option[,option...] = [X86-64]

如需更多信息，请参阅 **Documentation/x86/x86_64/boot-options.rst**。

更新的内核参数

acpi_sleep = [HW,ACPI]

格式为：{ s3_bios, s3_mode, s3_beep, s4_hwsig, s4_nohwsig, old_ordering, nonvs, sci_force_enable, nobl }

- 有关 **s3_bios** 和 **s3_mode** 的更多信息，请参见 **Documentation/power/video.rst**。
- **s3_beep** 用于调试；当内核的实际模式入口点被调用时，PC 会马上发出滴滴的声音。
- **s4_hwsig** 会导致内核在从休眠状态恢复过程中检查 ACPI 硬件签名，并在其更改后安全拒绝恢复。默认行为是允许恢复，只需在签名更改时警告，除非启用了 **s4_hwsig** 选项。
- **s4_nohwsig** 可防止在恢复过程中使用 ACPI 硬件签名，甚至警告。**old_ordering** 会导致 **_PTS** 控制方法的 ACPI 1.0 排序，以防止将设备置于低功耗状态。默认情况下会使用 **_PTS** 的 ACPI 2.0 排序。
- **nonvs** 可防止内核在挂起、休眠和恢复过程中保存和恢复 ACPI NVS 内存。
- **sci_force_enable** 可使内核直接从 S1/S3 恢复时设置 **SCI_EN**。虽然此行为与 ACPI 规范相反，但一些损坏的系统无法正常工作。
- **nobl** 不会导致已知系统的内部拒绝列表的行为在某些方面与系统挂起和恢复被忽略时不正确。请谨慎使用这个选项。
  如需更多信息，请参阅 **Documentation/power/video.rst**。

crashkernel=size[KMG],high = [KNL, X86-64, ARM64]

使用这个参数，您可以按如下方式从顶部分配物理内存区域：

- 如果系统安装了 4 GB RAM，物理内存区域可能会超过 4 GB。
- 如果系统安装的 4 GB RAM 少于 4 GB，那么将在 4 GB 下分配物理内存区域（如果可用）。
  如果指定了 **crashkernel=X** 参数，则忽略此参数。

crashkernel=size[KMG],low = [KNL, X86-64]

当您使用 **crashkernel=X,high** 时，内核可以分配超过 4 GB 的物理内存区域。这会导致在需要某种低内存的系统上出现第二个内核崩溃（例如，**swiotlb** 需要至少 64M+32K 低内存）和足够的低内存，以确保 32 位设备的 DMA 缓冲区不会被耗尽。内核会尝试自动分配比 4 GB 低 256 M。使用此参数，您可以为第二个内核指定 4 GB 的低范围。

- **0**：禁用低分配。当 **crashkernel=X,high** 未使用或保留内存低于 4 GB 时，它将会被忽略。

crashkernel=size[KMG],low = [KNL, ARM64]

使用此参数，您可以为崩溃转储内核在 DMA 区域中指定低范围。当 **crashkernel=X,high** 未使用或保留的内存位于 DMA 区域时，它将会被忽略。

kvm.nx_huge_pages_recovery_ratio = [KVM]

使用这个参数，您可以控制定期将多少 4 KiB 页面处理回巨页：

- **0** 禁用恢复
- **N** KVM 将在每个期间快速切换 4 KiB 页的 **1/Nth**。
  默认值为 **60**。

kvm-arm.mode = [KVM,ARM]

使用这个参数您可以选择 KVM 操作的模式之一：

- **none** ：强制禁用 KVM。
- **nvhe** ：基于 nVHE 的标准模式，无需对受保护的客户机提供支持。
- **protected** ：基于 **nVHE** 的模式，支持其状态从主机中私有的客户机。如果内核在 EL2 级别中运行，则无效。
  根据硬件支持，默认值被设置为 **VHE/nVHE**。

mitigations = [X86,PPC,S390,ARM64]

使用这个参数，您可以控制 CPU 漏洞的可选缓解方案。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：

- **off** ：禁用所有可选 CPU 缓解方案。这会提高系统性能，但可能会使用户暴露一些 CPU 漏洞。
  - 等同于：**nopti [X86,PPC]**, **kpti=0 [ARM64]**, **nospectre_v1 [X86,PPC]**, **nobp=0 [S390]**, **nospectre_v2 [X86,PPC,S390,ARM64]**, **spectre_v2_user=off [X86]**, **spec_store_bypass_disable=off [X86,PPC]**, **ssbd=force-off [ARM64]**, **l1tf=off [X86]**, **mds=off [X86]**, **tsx_async_abort=off [X86]**, **kvm.nx_huge_pages=off [X86]**, **no_entry_flush [PPC]**, **no_uaccess_flush [PPC]**, **mmio_stale_data=off [X86]**.
  - 例外：当指定了 **kvm.nx_huge_pages=force** 选项时，这不会影响 **kvm.nx_huge_pages=force**。
- **auto** （默认）：缓解所有 CPU 漏洞，但仍然启用 SMT，即使它存在安全漏洞。
  - 等效于：（默认行为）
- **auto,nosmt** ：缓解所有 CPU 漏洞，如果需要，禁用 SMT。
  - 等同于：**l1tf=flush,nosmt [X86]**, **mds=full,nosmt [X86]**, **tsx_async_abort=full,nosmt [X86]**, **mmio_stale_data=full,nosmt [X86]**

rcu_nocbs[=cpu-list] = [KNL]

可选参数是一个 CPU 列表。
在使用 **CONFIG_RCU_NOCB_CPU=y** 构建的内核中，您可以启用 no-callback CPU 模式，这样可防止此类 CPU 回调在 softirq 上下文调用。对这类 CPU 的 RCU callbacks 的调用会使用为这个目的创建的 **rcuox/N kthreads**，其中 **x** 是 **p**（对于 RCU-preempt），**s**（对于 RCU-sched），**g**（对于 **kthreads** 用于实现 grace 期），以及 **N** 是 CPU 号。这可减少卸载 CPU 上的 OS 存放位置，这对于 HPC 和实时工作负载非常有用。它还可提高非对称多处理器的能源效率。

- 如果将 **cpulist** 作为参数传递，则指定 CPU 列表从引导设置为 no-callback 模式。
- 如果省略 **=** 符号和 **cpulist** 参数，则不会在启动时将 CPU 设置为 no-callback 模式，但您可以使用 **cpusets** 在运行时切换模式。

rcutree.kthread_prio = [KNL,BOOT]

使用这个参数，您可以设置每个 CPU **kthreads** (**rcuc/N**) 的 RCU 的 **SCHED_FIFO** 优先级。这个值也用于 RCU 的优先级增加线程 (**rcub/N**) 和 RCU grace-period **kthreads** (**rcu_bh**、**rcu_preempt**、**rcu_sched**)。

- 如果设置了 **RCU_BOOST**，则有效值为 1-99，默认值为 **1**（最低优先级）。
- 如果没有设置 **RCU_BOOST**，则有效值为 0-99，默认值为 **0**（非实时操作）。
  当设置了 **RCU_NOCB_CPU** 时，您应该调整 **NOCB** 回调 **kthreads** 的优先级。

rcutorture.fwd_progress = [KNL]

使用这个参数，您可以指定用于 RCU grace-period forward-progress 测试的 **kthreads** 数。默认为 **1 kthread**。小于零的值或大于 CPU 数量的值会导致要使用的 CPU 的数量。

spectre_v2 = [X86]

通过这个参数，您可以控制 Spectre 变体 2（间接分支推测）漏洞的缓解。默认的操作可以防止内核免受用户空间的攻击。

- **on**: 无条件启用，代表 **spectre_v2_user=on**
- **off**: 无条件禁用，代表 **spectre_v2_user=off**
- **auto**: 内核会检测 CPU 中是否存在安全漏洞
- 选择 **on** 将会（选择 **auto** 可能会）根据 CPU，可用的 microcode， **CONFIG_RETPOLINE** 配置选项的设置，以及内核构建的编译器，在运行时选择环境方案。
- 选择 **on** 还将启用对用户空间任务攻击的缓解方案。
- 选择 **off** 将禁用内核和用户空间保护。
- 也可以手动选择特定的缓解方案：
  - **retpoline**: 替换间接分支
  - **retpoline,generic**: Retpolines
  - **retpoline,lfence**: LFENCE; 间接分支
  - **retpoline,amd**: retpoline,lfence 的别名
  - **eibrs** ：增强的 IBRS
  - **eibrs,retpoline**: 增强的 IBRS + Retpolines
  - **eibrs,lfence** ：增强的 IBRS + LFENCE
  - **ibrs**: 使用 IBRS 保护内核
    不指定这个选项等同于 **spectre_v2=auto**。

新 sysctl 参数

max_rcu_stall_to_panic

当您将 **panic_on_rcu_stall** 设置为 **1** 时，您可以确定 RCU 可在调用 **panic()** 前停止的次数。当您将 **panic_on_rcu_stall** 设置为 **0** 时，这个值不会起作用。

perf_user_access = [ARM64]

使用此参数，您可以控制用户空间的访问读取 **perf** 事件计数器。

- 当设置为 **1** 时，用户空间可以读取性能监控计数器寄存器。
- 默认值为 **0**，这表示**已禁用访问**。
  如需更多信息，请参阅文档 **/arm64/perf.rst**。

gro_normal_batch

使用此参数，您可以将片段的最大数量设置为在 GRO 的输出上批处理。当数据包退出 GRO 时，作为 coalesced 超级框架，或者作为原始数据包 (GRO) 决定不冲突，它将放置在每个NAPI 列表中。然后，当片段数量达到 **gro_normal_batch** 限制时，此列表将传递到堆栈。

high_order_alloc_disable

使用这个参数您可以选择 order-0 分配。默认情况下，页片段的分配器会尝试使用高顺序页面，即 X86 系统上的 order-3。虽然默认行为返回良好结果，但在某些情况下，在页面分配和释放发生的情况下发生。当高排序页面没有存储在每个 CPU 列表中时，在较旧的内核（版本 5.14 及更高版本）中尤其如此。这个参数现在包括大多数历史的重要性。
默认值为 **0**。

page_lock_unfairness

通过指定此参数的值，您可以确定页面锁定在等待者下可以阻止的次数。锁定被盗了在此文件中指定的次数后，会应用**公平锁定**的语义，并且等待者仅在有锁定时才会被忽略。
默认值为 **5**。

更改了 sysctl 参数

urandom_min_reseed_secs

您可以使用此参数来确定 **urandom** 池重新处理之间的最少秒数。此文件对于兼容性目的而言是可写的，但对任何 RNG 的行为都没有影响。

write_wakeup_threshold

当熵的 sink 数低于这个阈值时，您可以唤醒等待写入 **/dev/random** 文件的进程。此文件对于兼容性目的而言是可写的，但对任何 RNG 的行为都没有影响。

第 6 章 设备驱动程序

6.1. 新驱动程序

网络驱动程序

- 平台固件运行时更新遥测驱动程序 (**pfr_telemetry**)
- 平台固件运行时更新设备驱动程序 (**pfr_update**)
- MediaTek 设备的蓝牙支持超过 0.1 (**btmtk**)
- MHI 主机接口 (**mhi**)
- modem Host Interface (MHI) PCI 控制器驱动程序 (**mhi_pci_generic**)
- IDXD driver dsa_bus_type driver (**idxd_bus**)
- AMD PassThru DMA 驱动程序 (**ptdma**)
- Mellanox FAN 驱动程序 (**mlxreg-fan**)
- Mellanox LED regmap 驱动程序 (**leds-mlxreg**)
- Intel® LPSS ACPI 驱动程序 (**intel-lpss-acpi**)
- Intel® LPSS PCI 驱动程序 (**intel-lpss-pci**)
- Intel® LPSS 内核驱动程序 (**intel-lpss**)
- Maxlinear Ethernet GPY Driver (**mxl-gpy**)
- Realtek 802.11ax wireless 8852A 驱动程序 (**rtw89_8852a**)
- Realtek 802.11ax wireless 8852AE 驱动程序 (**rtw89_8852ae**)
- Intel® PMT Class 驱动程序 (**pmt_class**)
- Intel® PMT Crashlog 驱动程序 (**pmt_crashlog**)
- Intel® PMT Telemetry 驱动程序 (**pmt_telemetry**)
- Intel® speed select interface mailbox driver (**isst_if_mbox_msr**)
- Intel® speed select interface pci mailbox driver (**isst_if_mbox_pci**)
- Intel® speed select interface mmio driver (**isst_if_mmio**)
- Intel® Software Defined Silicon 驱动程序 (**intel_sdsi**)
- Intel® Extended Capabilities auxiliary bus driver (**intel_vsec**)
- ISH ISHTP eclite client opregion driver (**ishtp_eclite**)
- Acer Wireless Radio Control Driver (**acer-wireless**)
- AMD HSMP Platform Interface Driver (**amd_hsmp**)

- DESIGNWARE HS OTG Core (**dwc2**)
- Synopsys HAPS PCI Glue Layer (**dwc3-haps**)
- DesignWare USB3 PCI Glue Layer (**dwc3-pci**)
- DesignWare USB3 DRD Controller Driver (**dwc3**)
- xHCI Platform Host Controller Driver (**xhci-plat-hcd**)
- ON Semiconductor FSA4480 driver (**fsa4480**)
- Richtek RT1719 Sink Only USBPD Controller Driver (**rt1719**)
- Willsemi WUSB3801 Type-C port controller driver (**wusb3801**)
- Core driver for VFIO based PCI devices (**vfio-pci-core**)
- AMD SEV Guest Driver (**sev-guest**)
- Mellanox watchdog driver (**mlx_wdt**)

图形驱动程序和各种驱动程序

- Cirrus Logic DSP Support (**cs_dsp**)
- DRM DisplayPort helper (**drm_dp_helper**)
- DRM Buddy Allocator (**drm_buddy**)
- DRM SHMEM memory-management helpers (**drm_shmem_helper**)
- DRM driver using bochs dispi interface (**bochs**)
- Letsketch tablet driver (**hid-letsketch**)
- Intel® speed select interface driver (**isst_if_common**)
- SiGma Micro HID driver (**hid-sigmamicro**)
- Fixing side buttons of Xiaomi Mi Silent Mouse (**hid-xiaomi**)
- Driver for DEC VSXXX-AA and -GA mice and VSXXX-AB tablet (**vsxxxaa**)
- Nvidia line card platform driver (**mlxreg-lc**)
- Intel PCH Thermal driver (**intel_pch_thermal**)
- Intel LPSS UART driver (**8250_lpss**)

6.2. 更新的驱动程序

网络驱动程序更新

- VMware vmxnet3 virtual NIC driver (**vmxnet3**) 已被更新到版本 1.7.0.0-k。

存储驱动程序更新

- Emulex LightPulse Fibre Channel SCSI driver (**lpfc**) 已被更新到版本 14.2.0.5。
- MPI3 Storage Controller Device Driver (**mpi3mr**) 更新至 8.0.0.69.0。
- LSI MPT Fusion SAS 3.0 Device Driver (**mpt3sas**) 更新至 40.100.00.00。
- Microchip Smart Family Controller (**smartpqi**) 已更新至版本 2.1.18-045。

图形和各种驱动程序更新

- VMware SVGA 设备 (**vmwgfx**) 的独立 drm 驱动程序已更新至 2.20.0.0 版本。

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

Option	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 启用）
非特权用户的 JIT 的内存限值	264241152
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

Option	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	n
CONFIG_BPF_LIRC_MODE2	n
CONFIG_BPF_STREAM_PARSER	y
CONFIG_NETFILTER_XT_MATCH_BPF	m
CONFIG_BPFILTER	n
CONFIG_BPFILTER_UMH	n

Option	值
CONFIG_TEST_BPF	m
CONFIG_HZ	1000
bpf() syscall	可用
大型程序大小限制	可用

表 7.2. 可用的程序类型和支持的帮助程序

计划类型	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_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
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_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_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

计划类型	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_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_csum_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_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_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, 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_jiffies64, bpf_sk_assign, 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_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, 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

计划类型	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_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_csum_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_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_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, 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_jiffies64, bpf_sk_assign, 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_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, 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
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_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_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

计划类型	可用的帮助程序
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_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
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_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_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

计划类型	可用的帮助程序
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_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
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_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
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_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

计划类型	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_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_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, 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
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_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

计划类型	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_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
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_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
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_uid_gid, 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_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_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs

计划类型	可用的帮助程序
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_peek_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, 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_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
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_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_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

计划类型	可用的帮助程序
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_skc_lookup_tcp, 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_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
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_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_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, 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
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_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

计划类型	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_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
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_uid_gid, 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_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_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
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_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_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

计划类型	可用的帮助程序
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_uid_gid, 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_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_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
tracing	不支持

计划类型	可用的帮助程序
struct_ops	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_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_perf_event_read, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_stackid, bpf_csum_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_current_task_under_cgroup, 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_xdp_adjust_head, bpf_probe_read_str, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_setsockopt, bpf_skb_adjust_room, bpf_redirect_map, bpf_sk_redirect_map, bpf_sock_map_update, bpf_xdp_adjust_meta, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_getsockopt, bpf_override_return, bpf_sock_ops_cb_flags_set, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_bind, bpf_xdp_adjust_tail, bpf_skb_get_xfrm_state, bpf_get_stack, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_sock_hash_update, bpf_msg_redirect_hash, bpf_sk_redirect_hash, bpf_lwt_push_encap, bpf_lwt_seg6_store_bytes, bpf_lwt_seg6_adjust_srh, bpf_lwt_seg6_action, bpf_rc_repeat, bpf_rc_keydown, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_select_reuseport, 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_msg_push_data, bpf_msg_pop_data, bpf_rc_pointer_rel, 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_tcp_check_syncookie, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_send_signal, bpf_tcp_gen_syncookie, bpf_skb_output, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_tcp_send_ack, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_xdp_output, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_seq_printf, bpf_seq_write, 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_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_get_task_stack, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_inode_storage_get, bpf_inode_storage_delete, bpf_d_path, bpf_copy_from_user, bpf_snprintf_btf, bpf_seq_printf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_bprm_opts_set, bpf_ktime_get_coarse_ns, bpf_ima_inode_hash, bpf_sock_from_file, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_sys_bpf, bpf_btf_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_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_skc_to_unix_sock, bpf_kallsyms_lookup_name

计划类型	可用的帮助程序
ext	不支持
lsm	不支持
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_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, 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

表 7.3. 可用映射类型

映射类型	Available
hash	是
array	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

映射类型	Available
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
stack	是
sk_storage	是
devmap_hash	是
struct_ops	否
ringbuf	是
inode_storage	是
task_storage	是

第 8 章 程序错误修复

这部分描述了 Red Hat Enterprise Linux 9.1 中对用户有严重影响的 bug 修复。

8.1. 安装程序和镜像创建

安装程序不再安装早期版本的软件包

在以前的版本中，在安装过程中安装程序无法正确载入 DNF 配置文件。因此，安装程序有时会在 RPM 事务中安装所选软件包的早期版本。

这个 bug 已经修复，现在只安装存储仓库中最新版本的软件包。如果无法安装最新版本的软件包，安装会失败。

(BZ#2053710)

即使在 stage2 中更改网络配置，Anaconda 安装也会成功

在以前的版本中，当使用 **rd.live.ram** 引导参数时，Anaconda 不会卸载在 **initramfs** 中用来将安装镜像提取到内存中的 NFS 挂载点。因此，如果阶段 2 中更改了网络配置，安装过程可能会变得没有响应，或因为超时错误而失败。

要解决这个问题，在 switchroot 之前，在 **initramfs** 中卸载用来提取安装镜像的 NFS 挂载点。因此，安装过程会在不中断的情况下完成。

(BZ#2082132)

8.2. 订阅管理

现在，在 FIPS 模式中，virt-who 可以正确地连接到 ESX 服务器

在以前的版本中，当在 FIPS 模式的 RHEL 9 系统中使用 **virt-who** 工具时，**virt-who** 无法连接到 ESX 服务器。因此，**virt-who** 没有报告任何 ESX 服务器，即使为它们进行了配置，并记录以下出错信息：

```
ValueError: [digital envelope routines] unsupported
```

在这个版本中，**virt-who** 已被修复，以正确处理 FIPS 模式，上面描述的问题不再发生。

(BZ#2054504)

8.3. 软件管理

DNF 现在可以正确地回滚一个带有 Reason Change Action 类型的项的事务

在以前的版本中，对包含 **Reason Change** Action 类型的项的事务运行 **dnf history rollback** 命令会失败。有了这个更新，此问题已被解决，**dnf history rollback** 现在可以按预期工作。

(BZ#2053014)

8.4. SHELL 和命令行工具

ReaR 中的 vi 命令不再产生无限循环

在以前的版本中，ReaR 救援系统不包含 **vi** 可执行文件，只包含 **/bin/vi** 脚本。因此，**/bin/vi** 脚本在调用时会造成一个死循环。在这个版本中，ReaR 救援系统包含实际的 **vi** 可执行文件 **/usr/libexec/vi**，并运行 **vi** 命令不再会导致无限期循环。

(BZ#2097437)

具有 PXE 输出方法的 ReaR 不再将输出文件存储在 rsync OUTPUT_URL 位置

在以前的版本中，删除了使用 **OUTPUT=PXE** 和 **BACKUP=RSYNC** 选项的 **OUTPUT_URL** 变量处理。因此，当为 **OUTPUT_URL** 使用 rsync 位置时，ReaR 无法将 **initrd** 和内核文件复制到此位置，尽管将其上传到 **BACKUP_URL** 指定的位置。有了这个更新，RHEL 8.4 和更早版本的行为被恢复了。ReaR 使用 rsync 在指定的 **OUTPUT_URL** 目标中创建所需的文件。

(BZ#2115958)

如果没有更新 /etc/fstab 中的 UUID，ReaR 不再显示错误消息

在以前的版本中，当 ReaR 不能更新 **/etc/fstab** 中的通用唯一标识符(UUID)，来匹配在 UUID 不同的情况下新创建的分区时，ReaR 不会在恢复过程中显示一条错误消息。如果救援镜像与备份不同步，则会出现这种情况。在这个版本中，如果恢复的基本系统文件与重新创建系统不匹配，则会在恢复过程中显示错误消息。

(BZ#2083272)

ReaR 现在支持使用 NetBackup 版本 9 恢复系统

在以前的版本中，使用 NetBackup 版本 9 或更高版本的 NetBackup (NBU) 恢复系统会因为缺少库和其他文件而失败。在这个版本中，**NBU_LD_LARY_PATH** 变量包含所需的库路径，救援系统现在包含所需的文件，ReaR 可以使用 NetBackup 方法。

(BZ#2120736)

ReaR 不再显示有关丢失符号链接目标的假错误消息

在以前的版本中，ReaR 在创建救援镜像时，会对 **/usr/lib/modules/** 下的 **build** 和 **source** 符号链接缺失符号链接目标而显示不正确的错误消息。这个情况是无害的，您可以安全地忽略此错误消息。在这个版本中，ReaR 不会报告在这种情况下缺少 symlink 目标的假错误消息。

(BZ#2119501)

没有参数的 cmx 操作不再崩溃 CIM 客户端

cmx 操作调用一个方法，并返回 XML，一个参数指定调用的方法的名称。在以前的版本中，当在运行没有附加参数的 **cmx** 操作时，命令行 **sblim-wbemcli** 公共信息模型(CIM)客户端会崩溃。有了这个更新，**cmx** 操作需要定义调用的方法的名称。在没有此参数的情况下调用 **cmx** 操作会导致错误消息，CIM 客户端不再崩溃。

(BZ#2083577)

free 命令使用新的计算方法用于使用的内存

在以前的版本中，在 **free** 实用程序中计算已用内存的计算会减去了总内存的空闲空间、缓存空间和缓冲区空间。因此，当您将已用内存的值与另一个工具的结果进行比较时，会发生异常，因为 **free** 实用程序没有计算共享内存。在这个版本中，**free** 命令使用一个新的计算方法来提供可用内存的明确状态，并考虑不可声明的缓存。已用内存现在是任何不可用的内存，也包括虚拟内存中的 **tmpfs** 对象。

(BZ#2003033)

8.5. 基础架构服务

unbound 不再验证基于 SHA-1- 的 RSA 签名

在以前的版本中，OpenSSL 在 DEFAULT 系统范围的加密策略中无法验证 SHA-1- 基于 RSA 签名。因此，当 Unbound 试图验证这些签名时，OpenSSL 中的错误会导致解析失败。在这个版本中，Unbound 禁用对所有 RSA/SHA1（算法号 5）和 RSASHA1-NSEC3-SHA1（算法号 7）签名的验证支持。请注意，这会导致在所有系统范围的加密策略下不安全。

(BZ#2071543)

8.6. 安全

OpenSSH 密钥生成使用 FIPS 兼容接口

OpenSSH 使用的 OpenSSL 加密库提供了两个接口： legacy 和 modern。在以前的版本中，OpenSSH 将旧界面用于密钥生成，它不符合联邦信息处理标准 (FIPS) 要求。在这个版本中，**ssh-keygen** 工具使用 FIPS 兼容 API，而不是兼容 FIPS 的 API。因此，OpenSSH 密钥生成是 FIPS 兼容。

(BZ#2087121)

FIPS 批准的加密不再可用于 FIPS 模式

在以前的版本中，无论系统设置是什么，没有 FIPS 批准的加密程序在 OpenSSL 工具包中正常工作。因此，您可以使用在系统以 FIPS 模式运行时禁用的加密算法和密码，例如：

- 使用 RSA 密钥交换工作的 TLS 密码套件。
- 尽管使用 PKCS #1 和 SSLv23 补丁，或者使用比 2048 位更短的密钥，对公钥的加密算法进行基于 RSA 的加密算法。

在这个版本中，确保 FIPS 没有被 FIPS 批准的加密功能在 FIPS 模式的 OpenSSL 中无法正常工作。

(BZ#2053289)

指定从 OpenSSL 中删除任意 curves

在以前的版本中，检查显式 curve 参数安全是否不完整。因此，在 RHEL 中具有足够大的 **p** 值的任意 elliptic curves。在这个版本中，检查会验证显式 curve 参数是否与一个已知的受支持 curves 匹配。因此，使用显式 curve 参数来指定任意曲线的选项已从 OpenSSL 中删除。指定在 OpenSSL 中任意明确的 curves 无法正常工作的参数文件、私钥、公钥和证书。使用显式 curve 参数指定已知并支持的 curves 之一，如 P-224, P-256, P-384, P-521 和 **secp256k1**，在非 FIPS 模式中仍然被支持。

(BZ#2066412)

Openssl req 使用 AES-256-CBC 进行私钥加密

在以前的版本中，OpenSSL **req** 工具使用 3DES 算法加密私钥文件。因为 3DES 算法在当前 FIPS 140 标准中不安全且不禁止加密模块，所以 **req** 现在会生成使用 AES-256-CBC 算法加密的私钥文件。整个 PKCS#8 文件格式保持不变。

(BZ#2063947)

当使用 FFDHE 时，openssl 不再无法连接

在以前的版本中，使用基于 field 的 Diffie-Hellman ephemeral (FFDHE) 密钥交换机制的 TLS 连接有时会在从客户端处理 FFDHE 密钥共享时会失败。这是因为 OpenSSL 中的限制性检查导致。因此，OpenSSL 服务器会中止与 **internal_error** 警报的连接。在这个版本中，OpenSSL 接受较小的但仍然兼容的客户端

密钥共享。因此，在使用 FFDHE 密钥交换时，OpenSSL 和其他实现之间的连接不再随机中止。

(BZ#2004915)

基于 openssl 的应用程序现在可以使用 Turkish 区域正常工作

因为 **OpenSSL** 库使用不区分大小写的字符串比较功能，因此基于 OpenSSL 的应用程序无法使用 Turkish 区域正常工作，因此忽略的检查会导致应用程序使用这个区域设置崩溃。在这个版本中，提供一个补丁以使用可移植的操作系统接口(POSIX)区域进行不区分大小写的字符串比较。因此，基于 OpenSSL 的应用程序（如 curl）可以与 Turkish 区域正常工作。

(BZ#2071631)

在 SELinux 策略中添加 insights-client 的权限

新的 **insights-client** 服务需要权限，这些权限在以前的 **selinux-policy** 版本中。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。

(BZ#2081425, BZ#2077377, BZ#2087765, BZ#2107363)

SELinux staff_u 用户不再错误地切换到 unconfined_r

在以前的版本中，当启用了 **secure_mode** 布尔值时，**staff_u** 用户可以切换到 **unconfined_r** 角色，这不是预期的行为。因此，**staff_u** 用户可以执行影响系统安全性的特权操作。在这个版本中，SELinux 策略已被修复，**staff_u** 用户无法再错误地切换到 **unconfined_r**。

(BZ#2076681)

在检查可用内存时，OpenSCAP 不再生成错误

在以前的版本中，当评估一些 XCCDF 规则时，OpenSCAP 会错误地显示错误消息 **Failed to check available memory** 并生成无效的扫描结果。例如，这会在 **accounts_user_dot_no_world_writable_programs**, **accounts_user_dot_group_ownership** 和 **accounts_users_home_files_permissions** 规则发生。在这个版本中，错误处理中的错误已被修复，错误消息只针对真实故障显示。

(BZ#2109485)

fagenrules --load 现在可以正常工作

在以前的版本中，**fapolicyd** 服务无法正确处理信号挂起 (SIGHUP)。因此，在接收 SIGHUP 后 **fapolicyd** 终止，**fagenrules --load** 命令无法正常工作。此更新包含针对此问题的修复。因此，**fagenrules --load** 现在可以正常工作，规则更新不再需要手动重启 **fapolicyd**。

(BZ#2070655)

8.7. 网络

现在，即使在 Alibaba Cloud 中启动 nm-cloud-setup 服务后，实例也会保留主 IP 地址

在以前的版本中，在 Alibaba Cloud 中启动实例后，**nm-cloud-setup** 服务会在多个 IPv4 地址时将不正确的 IP 地址配置为主 IP 地址。因此，这会影响为出站连接选择 IPv4 源地址。在这个版本中，在手动配置辅助 IP 地址后，**NetworkManager** 软件包从 **primary-ip-address** 元数据获取主 IP 地址，并正确配置主 IP 地址。

(BZ#2079849)

NetworkManager 实用程序强制实施手动添加的 IPv6 地址的正确排序

通常，IPv6 地址的顺序会影响源地址选择的优先级。例如，当您进行传出 TCP 连接时。在以前的版本中，通过 **manual**、**dhcpv6** 和 **autoconf6** 方法添加的 IPv6 地址相对优先级不正确。在这个版本中解决了这个问题，排序优先级反映了这个逻辑： **manual** > **dhcpv6** > **autoconf6**。另外，**ipv6.addresses** 设置下的地址顺序会被反向，以便第一个添加的地址具有最高优先级。

(BZ#2097293)

8.8. 内核

网络套接字标记再次可以正常工作

某些旧的 **cgroup** v1 控制器没有对应的 **cgroup** v2，如 **net_prio** 或 **net_cls**，之前会干扰 **cgroup** v2 套接字标记，当它们与混合 **cgroup** v1/v2 环境中的其他 **cgroup** v2 控制器一起挂载时。因此，使用 **net_prio** 或 **net_cls** v1 控制器使用 **cgroup** v1/v2 的混合 cgroup v1/v2 环境禁用了带有 **cgroup** v2 的网络 socket 标记。这个版本消除了这个限制，因此可以使用混合 cgroup v1/v2 环境网络套接字标记。

(BZ#2060150)

kexec-tools 软件包现在支持默认的 crashkernel 内存保留值

kexec-tools 软件包现在维护默认的 **crashkernel** 内存保留值。**kdump** 服务使用默认值为每个内核保留崩溃内核内存。当系统小于 4 GB 的可用内存时，这个实现还会提高 **kdump** 的内存分配。

如果默认 **crashkernel** 值在系统中保留的内存不足，您可以使用 **kdumpctl estimate** 命令获得估算值而无需触发崩溃。估算的 **crashkernel=** 值可能并不准确，可作为设置适当的 **crashkernel=** 值的参考。

(BZ#1959203)

系统可以成功运行动态 LPAR 操作

在以前的版本中，如果满足以下条件之一，用户无法从硬件管理控制台(HMC)运行动态逻辑分区(DLPAR)操作：

- 启用安全引导功能，以隐形方式在完整性模式下启用内核 **lockdown** 机制。
- 过去，内核 **lockdown** 在完整性模式或机密模式下需要手动启用。

在 RHEL 9 中，内核 **lockdown** 完全阻止了 Run Time Abstraction Services(RTAS)访问通过 **/dev/mem** 字符设备文件访问系统内存。多个 RTAS 调用需要对 **/dev/mem** 进行写入访问权限才能正常工作。因此，RTAS 调用无法正确执行，用户会看到以下错误消息：

> HSCL2957 Either there is currently no RMC connection between the management console and the partition <LPAR name> or the partition does not support dynamic partitioning operations. Verify the network setup on the management console and the partition and ensure that any firewall authentication between the management console and the partition has occurred. Run the management console diagrmc command to identify problems that might be causing no RMC connection.

在这个版本中，这个问题已通过提供非常缩小的 PowerPC 异常 **lockdown** 来解决。例外允许 RTAS 访问所需的 **/dev/mem** 区域。因此，这个问题不再在上述场景中的清单。

(BZ#2046472)

在将环缓冲值从 rx 设置为 max 后没有内核警告

当内部函数预期一个干净的输入，但调用时带有一个重新使用的已初始化数据结构，则内核会生成警告消息 **Missing unregister, handled but fix driver**。在这个版本中，通过在重新注册前重新初始化结构来解决这个问题。

(BZ#2054379)

8.9. 引导加载程序

grubby 现在将参数传递给未来的内核

当安装较新版本的内核时，**grubby** 工具不会传递来自上一内核版本的内核命令行参数。因此，GRUB 引导装载程序会忽略用户设置。有了这个修复，在安装了新内核版本后，用户设置仍然有效。

(BZ#1978226)

8.10. 文件系统和存储

日志条目不再停止日志写入

在以前的版本中，在设备映射器挂起操作期间和恢复设备操作后的 VDO 驱动程序中，一些日志块仍然被标记为等待一些元数据更新，直到它们可以被重复使用，即使这些更新已经完成。当为日志制作了足够的日志条目以将其折回同一物理块时，它不可用。日志写入将停止，等待块可用，这永远不会发生。因此，当对 VDO 设备的一些操作包含挂起或恢复循环时，设备会在一些日志更新后处于冻结状态。此设备状态之前的日志更新不可预测，因为它依赖 VDO 中以前的分配模式，以及传入的写或丢弃模式。有了这个更新，在挂起或恢复将数据保存到存储后，内部数据结构状态会被重置，并不会再发生锁定。

(BZ#2064802)

添加数据设备不再触发断言失败

在以前的版本中，当在缓存中添加附加设备时，Stratis 在初始化后不会立即使用缓存。因此，当用户试图向池中添加额外的数据设备时，**stratisd** 服务会返回一个断言失败信息。在这个版本中，缓存会在初始化后立即使用，且不会发生断言失败。

(BZ#2007018)

在向加密池添加新数据设备时解决的错误

在以前的版本中，当用户使用 tang 服务器（通过 **--trust-url** 选项指定）在 tang 服务器上使用 Clevis bind 命令初始化加密池时，**stratisd** 不会将 Clevis tang配置的 thumbprint 部分包含在内部数据结构中。因此，当试图向池中添加新数据设备时，会出现故障。在这个版本中，**stratisd** 的内部数据结构包括 Clevis tang 配置的 thumbprint 部分。

(BZ#2005110)

从 AMD EPYC 系统上的广播发起端连接到 NVMe 命名空间不再需要非默认 IOMMU 设置

默认情况下，RHEL 内核在基于 AMD 的平台上启用 IOMMU。在以前的版本中，**lpfc** 驱动程序没有使用 scatter-gather 列表 accessor 宏。因此，AMD 处理器的某些服务器遇到 NVMe I/O 问题，如 I/O 故障，因为传输长度不匹配。

在这个版本中，您不需要使用内核命令行选项将 IOMMU 置于维护模式，以便从 Broadcom 启动器连接到 NVMe 命名空间。

(BZ#2073541)

8.11. 高可用性和集群

pcs 现在验证 stonith-watchdog-timeout 的值

在以前的版本中，可以将 **stonith-watchdog-timeout** 属性设为与 SBD 配置不兼容的值。这可能会导致隔离循环，或者可能导致集群将隔离操作视为成功，即使操作没有完成。在这个版本中，**pcs** 会在您设置时验证 **stonith-watchdog-property** 的值，以防止配置不正确。

(BZ#2058246)

pcs 现在识别创建新 Booth ticket 时的 mode 选项

在以前的版本中，当用户在添加新的 Booth 票据时指定了 **mode** 选项，**pcs** 会报告错误 **invalid booth ticket option 'mode'**。在这个版本中，您可以在创建 Booth ticket 时指定 **mode** 选项。

(BZ#2058243)

pcs 现在区分资源和 stonith 资源

在以前的版本中，一些 **pcs** 命令无法区分 resources 和 stonith 资源。这允许用户使用 **pcs resource** 子命令获取 stonith 资源，并将 **pcs stonith** 子命令用于非 stonith 资源的资源。这可能会导致用户混淆或资源错误配置。在这个版本中，当有资源类型不匹配时，**pcs** 会显示一个警告。

(BZ#1301204)

8.12. 编译器和开发工具

glibc 现在在载入 NSS 模块后恢复 errno

在以前的版本中，如果最后一个 NSS 模块没有提供任何数据，**glibc** 中的 Name Service Switch (NSS)实现会在数据库枚举过程中（如使用 **getpwent()**）出现错误。因此，使用这些枚举功能的应用程序会错误地观察到错误和失败。**glibc** 现在在载入 NSS 模块后恢复 errno，因此使用这些功能的应用程序不再会失败。

(BZ#2063142)

审计界面现在保存并恢复 x8 寄存器，以及 NEON 注册的完整宽度 AArch64

在以前的版本中，动态加载程序审计接口实现中的一个错误会导致 **AArch64** 保存的寄存器状态与过程调用标准相比不完整。这个程序错误已被解决，审核接口现在保存并恢复 x8 注册，以及用于 **AArch64** 的 NEON 注册的完整宽度。使用动态加载程序审核接口的应用程序现在可以检查并影响 x8 为 **AArch64** 的注册。要使用这个新的 x8 注册，且有权访问 NEON 寄存器的完整宽度，必须在 **AArch64** 中重新编译审计模块以使用接口的新版本 (LAV_CURRENT 为 2)。

(BZ#2003291)

POWER9-optimized strncpy 功能不再给出不正确的结果

在以前的版本中，POWER9 strncpy 功能没有使用正确的注册作为 padding NUL 字节的来源。因此，输出缓冲包含未初始化的寄存器内容，而不是 NUL padding。在这个版本中，strncpy 功能已被修复，输出缓冲区的末尾会正确添加 NUL 字节。

(BZ#2091549)

在 IBMz15 构架中安装的 glibc memmem 功能的 Valgrind 覆盖

在以前的版本中，**glibc memmem** 功能缺少 valgrind 覆盖会导致假的警告：

> Conditional jump or move depends on uninitialised value(s)

这个版本包括 **glibc memmem** 函数的 valgrind 覆盖，因此在 IBMz15 架构下运行的程序中使用 **memmem** 功能时不再有假的警告。

(BZ#1993976)

8.13. 身份管理

ipa user-del --preserve user_login 输出不再表示该用户已被删除

在以前的版本中，如果您运行 **ipa user-del --preserve user_login** 命令来保留用户帐户，输出会错误地返回信息 **Deleted user "user_login"**。在这个版本中，输出会返回 **Preserved user “user_login”**。

(BZ#2100227)

PKINIT 用户身份验证现在可以在 RHEL 9 Kerberos 客户端中正常工作 - Heimdal KDC 场景

在以前的版本中，在 RHEL 9 Kerberos 客户端中针对 Heimdal Kerberos 分发中心 (KDC) 的 IdM 用户进行 PKINIT 身份验证会失败。发生这种情况的原因是 Kerberos 客户端不支持 RHEL 9 中弃用 SHA-1 算法上下文中所需的 **supportedCMSTypes** 字段。

在这个版本中，在从 PKINIT 到 Heimdal KDC 的过程中，RHEL 9 Kerberos 客户端会发送了一个签名算法列表，包括 **sha512WithRSAEncryption**, 和 **sha256WithRSAEncryption** 作为 **supportedCMSTypes**。Heimdal KDC 使用 **sha512WithRSAEncryption**，因此 PKINIT 身份验证可以正常工作。

(BZ#2068935)

处理 LDAP 组成员列表中的非可读对象

在此次更新之前，SSSD 不一致地处理 LDAP 组成员列表中不可读对象，这会导致不可读对象出错或在某些情况下不可读对象被忽略。

有了这个更新，SSSD 有一个新的选项 **ldap_ignore_unreadable_references** 来修改此行为。如果 **ldap_ignore_unreadable_references** 选项被设为 **false**，则不可读对象会导致错误，如果设为 **true**，则不可读对象会被忽略。默认值被设为 **false**，因为更新后原始不一致的行为，一些组查找可能会失败。在这种情况下，在 **/etc/sssd/sssd.conf** 文件中对应的 **[domain/*name of the domain*]** 部分中设置 **ldap_ignore_unreadable_references = True**。

这允许以一致方式处理不可读对象，并使用新的 **ldap_ignore_unreadable_references** 选项调优行为。

(BZ#2069376)

8.14. DESKTOP

修复了使用激活码注册的订阅

在以前的版本中，您无法使用激活码在 Settings 中注册您的红帽订阅。在按 **Register** 后 Settings 会显示以下错误信息：

> Failed to register system; Failed to RegisterWithActivationKeys: Unknown arguments: dict_keys(['enable_content'])

在这个版本中，这个问题已被解决，现在可以使用 Settings 中的激活码注册您的订阅。

(BZ#2100467)

8.15. 图形基础结构

X.org 现在启用 X11 SECURITY 扩展

在以前的版本中，X.org 显示服务器不提供 X11 **SECURITY** 扩展。因此，使用这个扩展的应用程序会意外终止。

在这个版本中，X.org 启用 X11 **SECURITY** 扩展。因此，依赖于扩展的应用程序现在可以正常工作。

(BZ#1894612)

带有 VGA 显示器的 Matrox GPU 现在可以按预期工作

在这个版本前，如果您使用以下系统配置，显示器没有图形输出：

- Matrox MGA G200 家族中的 GPU
- 通过 VGA 控制器连接的显示
- UEFI 切换到旧模式

因此，您不能在此配置上使用或安装 RHEL。

有了这个更新，**mgag200** 驱动程序已被显著重写，因此图形输出现在可以按预期工作。

(BZ#2100898)

8.16. WEB 控制台

使用 Web 控制台删除 USB 主机设备现在可以按预期工作

在以前的版本中，当您将 USB 设备附加到虚拟机时，USB 设备的设备号和总线号会在传给虚拟机后改变。因此，由于设备和总线号关联不正确，因此使用 Web 控制台删除这样的设备会失败。有了这个更新，这个问题已被解决，您可以使用 web 控制台删除 USB 主机设备。

(JIRA:RHELPLAN-109067)

使用 Web 控制台附加多个主机设备现在可以按预期工作

在以前的版本中，当使用 web 控制台选择多个设备来附加到虚拟机(VM)时，只会附加一个设备，剩余的设备都会被忽略。有了这个更新，这个问题已被解决，现在您可以使用 web 控制台同时附加多个主机设备。

(JIRA:RHELPLAN-115603)

8.17. RED HAT ENTERPRISE LINUX 系统角色

network RHEL 角色管理配置文件中的 ansible_managed 参数

在以前的版本中，Ansible 角色无法为 **network** 角色管理的配置文件提供正确的 **ansible_managed** 标头。因此，系统管理员对哪些文件由 Ansible 管理无关。在这个版本中，角色受管文件具有正确的 **ansible_managed** 标头，系统管理员可以可靠地了解哪些文件是管理的 Ansible。

(BZ#2065382)

修复了一个拼写错误，为正确的绑定模式支持 active-backup

在以前的版本中，在指定 **active-backup** 绑定模式时，在支持 InfiniBand 端口时，**active_backup** 有一个拼写错误 。由于这个拼写错误，对于 InfiniBand 绑定端口，连接无法支持正确的绑定模式。这个更新通过将绑定模式改为 **active-backup** 解决了拼写错误。现在，连接可以成功支持 InfiniBand 绑定端口。

(BZ#2065394)

IPRouteUtils.get_route_tables_mapping() 函数现在接受任何空白序列

在以前的版本中，**iproute2** 路由表数据库的一个解析程序（如 **/etc/iproute2/rt_tables** ）断言，文件中条目的格式为 **254 main** ，只有一个空格字符将数字 ID 和名称分开。因此，解析器无法缓存路由表名称和表 ID 之间的所有映射。因此用户无法通过定义路由表名称来在路由表中添加静态路由。有了这个更新，解析器会接受表 ID 和表名称之间的任何空格序列。因此，因为解析器会缓存路由表名称和表 ID 之间的所有映射，用户可以通过定义路由表名称在路由表中添加静态路由。

(BZ#2115886)

forward_port 参数现在接受 字符串和字典选项

在以前的版本中，在 **firewall** RHEL 系统角色中， **forward_port** 参数只接受 **string** 选项。但是，该角色文档声明了 **字符串 和 字典** 选项的支持。因此，用户读取并遵循文档收到错误。这个程序错误已通过使 **forward_port** 接受这两个选项来解决。因此，用户可以安全地按照文档配置端口转发。

(BZ#2100605)

现在，由 metrics 角色配置会正确遵循符号链接

安装 **mssql pcp** 软件包后，**mssql.conf** 文件位于 **/etc/pcp/mssql/** 中，由被符号链接 **/var/lib/pcp/pmdas/mssql/mssql.conf** 定为目标。但是， **metrics** 角色覆盖了符号链接，而不是遵循它，并配置 **mssql.conf**。因此，运行 **metrics** 角色会更改到常规文件的符号链接，因此配置只影响 **/var/lib/pcp/pmdas/mssql/mssql.conf** 文件。这会导致符号链接失败，而主配置文件 **/etc/pcp/mssql.conf** 没有受到配置的影响。这个问题现已解决，遵循符号链接的 **follow: yes** 选项已添加到 **metrics** 角色中。因此， **metrics** 角色会保留符号链接并正确配置主配置文件。

(BZ#2060523)

kernel_settings configobj 在受管主机上可用

在以前的版本中，**kernel_settings** 角色没有在受管主机上安装 **python3-configobj** 软件包。因此，角色会返回一个错误，表示无法找到 **configobj** Python 模块。在这个版本中，角色可确保受管主机上存在 **python3-configobj** 软件包，**kernel_settings** 角色可以正常工作。

(BZ#2060525)

卷的 mount_options 参数现在对卷有效

在以前的版本中，该参数会从卷的有效参数列表中意外删除。因此，用户无法为卷设置 **mount_options** 参数。有了这个 bug 修复，**mount_options** 参数已被添加到有效参数列表中，并且已重构了代码来捕获错误。因此，**storage** RHEL 系统角色可以为卷设置 **mount_options** 参数。

(BZ#2083376)

storage RHEL 系统角色现在可以正确地支持 LVM 卷的条状和 raid0 级别

storage RHEL 系统角色以前错误地将 RAID 级别 **striped** 和 **raid0** 报告为 LVM 卷不支持。现在，这个问题已被解决，角色现在可以正确创建由 LVM 支持的所有 RAID 级别的 LVM 卷：**raid0**、**raid1**、**raid4**、**raid5**、**raid6**、**raid10**、**striped** 和 **mirror**。

(BZ#2083410)

metrics RHEL 系统角色 README 和文档现在明确指定角色在 RHEL 的特定版本上支持的 Redis 和 Grafana 版本

在以前的版本中，当尝试在不支持的平台上使用具有不支持的 Redis 和 Grafana 版本的 **metrics** 角色时，角色会失败。这个更新澄清了关于角色在哪个 RHEL 版本上支持哪个 Redis 和 Grafana 版本的文档。因此，您可以避免在不支持的平台上使用 Redis 和 Grafana 的不受支持的版本。

(BZ#2100286)

ssh 和 sshd RHEL 系统角色中的最小 RSA 密钥位长度选项

意外使用短 RSA 密钥可能会使系统更易受到攻击。有了此更新，您可以在 **ssh** 和 **sshd** RHEL 系统角色中使用 **RequiredRSASize** 选项为 OpenSSH 客户端和服务器设置 RSA 密钥最小位长度。

(BZ#2109998)

nbde_client RHEL 系统角色现在在指定额外的 Dracut 命令行参数时使用正确的空格

Dracut 框架需要在指定附加参数时正确启动，如内核命令行参数。如果没有通过正确的空间指定参数，则 Dracut 可能不会将指定的额外参数附加到内核命令行中。有了此更新，**nbde_client** RHEL 系统角色在创建附加组件 Dracut 配置文件时使用正确的空格。因此，该角色可以正确地设置 Dracut 命令行参数。

(BZ#2115156)

tlog RHEL 系统角色现在可以被 SSSD 正确覆盖

在以前的版本中，**tlog** RHEL 系统角色依赖系统安全服务守护进程(SSSD)文件提供者以及启用的 **authselect** 选项 **with-files-domain**，来在 **nsswitch.conf** 文件中设置正确的 **passwd** 条目。在 RHEL 9.0 中，SSSD 默认不会隐式启用文件供应商，因此 SSSD 的 **tlog-rec-session** shell 覆盖无法正常工作。有了这个修复，**tlog** 角色现在可以更新 **nsswitch.conf**，来确保**tlog-rec-session** 被 SSSD 正确覆盖。

(BZ#2071804)

metrics RHEL 系统角色在更新其配置后自动重启 pmie 和 pmlogger 服务

在以前的版本中，**pmie** 和 **pmlogger** 服务在配置更改后不会重启，并等待处理程序执行。这会导致其他 **metrics** 服务出现错误，这些服务需要 **pmie** 和 **pmlogger** 配置与其运行时行为匹配。在这个版本中，角色会在配置更新后立即重启 **pmie** 和 **pmlogger**，其配置与依赖 metrics 服务的运行时行为匹配，它们可以正常工作。

(BZ#2100294)

8.18. 虚拟化

当负载过重时，虚拟机的网络流量性能不再降低

在以前的版本中，在一些情况下，RHEL 虚拟机在处理高级别的网络流量时会降低性能。已修复底层代码，在上述情况下，网络流量性能现在可以正常工作。

(BZ#1945040)

8.19. 云环境中的 RHEL

附加到 Hyper-V 虚拟机的网络适配器的 SR-IOV 功能现在可以正常工作

在以前的版本中，当把带有单根 I/O 虚拟化(SR-IOV)的网络适配器附加到在 Microsoft Hyper-V hypervisor 上运行的 RHEL 9 虚拟机(VM)时，在某些情况下 SR-IOV 功能无法正常工作。解决了 Hyper-V 特定内存映射 I/O (MMIO)分配代码中的一个错误，SR-IOV 功能现在可以正常工作。

(BZ#2030922)

SR-IOV 不再在 Azure 上的 ARM 64 RHEL 9 虚拟机中执行子利用

在以前的版本中，SR-IOV 网络设备在 Microsoft Azure 平台上运行的 ARM 64 RHEL 9 虚拟机 (VM) 中具有明显低于预期的吞吐量和更高的延迟。这个问题已被解决，受影响的虚拟机现在如预期执行。

(BZ#2068432)

8.20. 容器

podman system connection add 和 podman image scp 不再失败

Podman 为 RSA 密钥交换使用 SHA-1 哈希。在以前的版本中，在机器间的正常 SSH 连接中使用 RSA 密钥可以正常工作，而 **podman system connection add** 和 **podman image scp** 命令使用相同的 RSA 密钥无法工作，因在 RHEL 9 上密钥交换不接受 SHA-1 哈希。有了这个更新，此问题已被解决。

(JIRA:RHELPLAN-121180)

现在，可以拉取使用 Beta GPG 密钥签名的容器镜像

在以前的版本中，当您拉取 RHEL Beta 容器镜像时，Podman 会失败，并显示错误消息：**Error: Source image rejected: None of the signatures are accepted**。由于当前构建被配置为不信任 RHEL Beta GPG 密钥，所以镜像无法被拉取 (pull) 失败。在这个版本中，**/etc/containers/policy.json** 文件支持一个新的 **keyPaths** 字段，该字段接受包含可信密钥的文件列表。因此，在默认配置中，现在接受使用 GA 和 Beta GPG 密钥签名的容器镜像。

(BZ#2094015)

Podman 不再无法拉取容器 "X509: certificate signed by unknown authority"

在以前的版本中，如果您有自己 CA 证书签名的内部 registry，则必须将证书导入到主机机器中。否则，会出现错误：

```
x509: certificate signed by unknown authority
```

在这个版本中，这个问题已被解决。

(BZ#2027576)

由于不匹配存储库 ID，DNF 和 YUM 不再失败

在以前的版本中，DNF 和 YUM 存储库 ID 与 DNF 或 YUM 的预期格式不匹配。例如，如果您运行以下示例，则会发生错误：

```
# podman run -ti ubi8-ubi
# dnf debuginfo-install dnsmasq
...
This system is not registered with an entitlement server. You can use subscription-manager to register.
```

在这个版本中，这个问题已被解决。后缀 **--debug-rpms** 被添加到所有 debug 软件仓库名称中（如 **ubi-8-appstream-debug-rpms**），以及后缀 **-rpms** 被添加到所有 UBI 存储库名称（如 **ubi-8-appstream-rpms**）。

如需更多信息，请参阅 通用基础镜像(UBI)：镜像、存储库、软件包和源代码。

(BZ#2120378)

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. SHELL 和命令行工具

ReaR 在 64 位 IBM Z 构架中作为技术预览提供

64 位 IBM Z 构架中现在作为技术预览提供了基本的 Relax 和 Recover(ReaR)功能。您只能在 z/VM 环境中的 IBM Z 上创建 ReaR 救援镜像。备份和恢复逻辑分区(LPAR)还没有测试。

当前唯一可用的输出方法是 Initial Program Load(IPL)。IPL 生成一个内核和一个初始 ramdisk(initrd)，可与 **zIPL** 引导装载程序一起使用。

警告

目前，救援过程会重新格式化连接到系统的所有 DASD（直接附加的存储设备）。如果系统存储设备中存有宝贵的数据，则不要尝试进行系统恢复。这还包括用于引导到救援环境的 **zIPL** 引导装载程序、ReaR 内核和 initrd 的设备。确保保留一个副本。

如需更多信息，请参阅在 64 位 IBM Z 架构中使用 ReaR 救援镜像 。

(BZ#2046653)

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**。

(BZ#2047161)

9.2. 安全性

gnutls 现在使用 KTLS 作为技术预览

更新的 **gnutls** 软件包可以使用 Kernel TLS (KTLS) 在加密频道上加速数据传输作为技术预览。要启用 KTLS，请使用 **modprobe** 命令添加 **tls.ko** 内核模块，并创建一个新的配置文件 **/etc/crypto-policies/local.d/gnutls-ktls.txt**，其中包含以下内容：

```
[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。

(BZ#2042009)

9.3. 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。

(BZ#1613522)

使用 NetworkManager 配置多路径 TCP 作为技术预览提供

在这个版本中，NetworkManager 工具为您提供了多路径 TCP (MPTCP) 功能。您可以使用 **nmcli** 命令控制 MPTCP，并使其设置持久。

如需更多信息，请参阅了解多路径 TCP：端点高可用性，以及未来的网络和 RFC 8684：用于多地址的多路径操作 TCP 扩展。

(BZ#2029636)

KTLS 作为技术预览提供

RHEL 作为技术预览提供内核传输层(KTLS)。KTLS 使用内核中的对称加密或者解密算法为 AES-GCM 密码处理 TLS 记录。KTLS 还包括将 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。

(BZ#1570255)

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。

(BZ#2020529)

9.4. 内核

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)。

(BZ#2030412)

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 内核部分提供了 SGX v1 和 v1.5 功能。版本 1 启用使用 **Flexible Launch Control** 机制的平台使用 SGX 技术。

(BZ#1874182)

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。

(BZ#2023416)

9.5. 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** 会把存储直接映射到应用程序的地址空间中。

(BZ#1995338)

Stratis 作为技术预览提供

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

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

Stratis 作为技术预览提供。

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

(BZ#2041558)

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/developers/nvme-specification/ 网站中的 NVM Express 2.0 Ratified TPs。

(BZ#2021672)

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**)。

(BZ#1893841)

9.6. 编译器和开发工具

jmc-core 和 owasp-java-encoder 作为技术预览

RHEL 9 带有 **jmc-core** 和 **owasp-java-encoder** 软件包作为技术预览功能。

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，以及通过 Java 发现协议(JDP)发现的 Java 虚拟机(JVM)发现库。

owasp-java-encoder 软件包提供了 Java 的高性能低后台上下文组。

(BZ#1980981)

9.7. 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。

(BZ#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)。

(BZ#2084166)

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。当这些用户与 RHEL 9.1 中的 SSSD 版本进行身份验证时，它们会在外部 IdP 执行身份验证和授权后接收到使用 Kerberos ticket 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。

(BZ#2069202)

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。这个功能只适用于 RHEL 8.7 及更高版本上的 IdM 服务器，以及 RHEL 9.1 及更高版本。

(BZ#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。这个功能只适用于 RHEL 8.7 及更高版本上的 IdM 服务器，以及 RHEL 9.1 及更高版本。

(BZ#2056482)

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
```

(BZ#2084181)

9.8. DESKTOP

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)

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）

9.9. WEB 控制台

Stratis 作为 RHEL web 控制台中的技术预览提供

有了这个更新，Red Hat Enterprise Linux web 控制台将管理 Stratis 存储作为一个技术预览提供。

要了解更多有关 Stratis 的信息，请参阅什么是 Stratis。

(JIRA:RHELPLAN-122345)

9.10. 虚拟化

RHEL 虚拟机现在可以部署到 ARM64 处理器上运行的 VMware ESXi 实例

作为技术预览，现在可以将 RHEL 虚拟机部署到在基于 64 位 ARM 的处理器上运行的 VMware ESXi hypervisor 实例。

(JIRA:RHELPLAN-95456)

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)

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)

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。

(BZ#2014487, BZ#2044162, BZ#2044172)

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，并将在以后的 RHEL 发行版本中完全删除。

(JIRA:RHELDOCS-17050)

创建嵌套虚拟机

> Nested KVM virtualization is provided as a Technology Preview for KVM virtual machines (VMs) running on Intel, AMD64, and IBM Z hosts with RHEL 9. With this feature, a RHEL 7, RHEL 8, or RHEL 9 VM that runs on a physical RHEL 9 host can act as a hypervisor, and host its own VMs.

(JIRA:RHELDOCS-17040)

9.11. 云环境中的 RHEL

RHEL secret 虚拟机现在在 Azure 上作为技术预览提供

使用更新的 RHEL 内核，现在您可以在 Microsoft Azure 上创建并运行 secret 虚拟机 (VM) 作为技术预览。但是，在 Azure 上引导时无法加密 RHEL 机密虚拟机镜像。

(JIRA:RHELPLAN-122321)

9.12. 容器

多个用于签名镜像的可信 GPG 密钥的功能作为技术预览提供

/etc/containers/policy.json 文件支持一个新的 **keyPaths** 字段，该字段接受包含可信密钥的文件列表。因此，在默认配置中，现在接受使用 GA 和 Beta GPG 密钥签名的容器镜像。

例如：

```
"registry.redhat.io": [
        {
            "type": "signedBy",
            "keyType": "GPGKeys",
            "keyPaths": ["/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release", "/etc/pki/rpm-gpg/RPM-GPG-
KEY-redhat-beta"]
        }
]
```

(JIRA:RHELPLAN-129327)

sigstore 签名现在作为技术预览提供

从 Podman 4.2 开始，您可以使用容器镜像签名的 sigstore 格式。sigstore 签名会与容器镜像一起存储在容器 registry 中，而无需具有单独的签名服务器来存储镜像签名。

(JIRA:RHELPLAN-74672)

podman-machine 命令不受支持

用于管理虚拟机的 **podman-machine** 命令仅作为技术预览提供。相反，请直接从命令行运行 Podman。

(JIRA:RHELDOCS-16861)

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。

在 Red Hat Enterprise Linux 9 中，已弃用的功能的支持状态不会改变。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。硬件驱动程序更新仅限于安全和关键修复。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*功能的信息，请参阅使用 RHEL 9 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。

(BZ#1899167)

10.2. SHELL 和命令行工具

在 ReaR 配置文件中设置 TMPDIR 变量已弃用

使用语句如 **export TMPDIR=...** 来在 **/etc/rear/local.conf** 或 **/etc/rear/site.conf** ReaR 配置文件中设置 **TMPDIR** 环境变量，其无法正常工作，并已弃用。

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

10.3. 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)

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)

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。

(BZ#1995600)

OpenSSL 弃用 MD2, MD4, MDC2, Whirlpool, RIPEMD160, Blowfish, CAST, DES, IDEA, RC2, RC4, RC5, SEED, and 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至 OpenSSL 中的旧供应商：MD2、MD4、MD4、MDC2、Whirlpool、RIPEMD160、Blowfish、CAST、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。

(BZ#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)

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。

(BZ#2034569)

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。

(BZ#2054740)

10.4. 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。

(BZ#1935544)

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文件的详情，请参考 keyfile 格式的 NetworkManager 连接配置文件。
(BZ#1894877)

firewalld 中的 iptables 后端已弃用

在 RHEL 9 中，**iptables** 框架已弃用。因此，**iptables** 后端和 **firewalld** 中的**直接接口**也已弃用。您可以使用 **firewalld** 中的原生功能来配置所需的规则，而不是**直接接口**。

(BZ#2089200)

10.5. 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 。

(BZ#2058153)

10.6. 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。

(BZ#2038183)

10.7. 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。

(BZ#1927780, BZ#1974657, JIRA:RHELPLAN-80695)

10.8. 编译器和开发工具

openssl 3.0 弃用了比 2048 小的密钥

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。

(BZ#2111072)

有些 PKCS1 v1.5 模式现已弃用

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。

(BZ#2092016)

10.9. 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。

(BZ#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)

-h 和 -p 选项在 OpenLDAP 客户端工具中被弃用。

上游 OpenLDAP 项目已在其工具中弃用了 **-h** 和 **-p** 选项，建议使用 **-H** 选项来指定 LDAP URI。因此，RHEL 9 已在所有 OpenLDAP 客户端工具中都弃用了这两个选项。**-h** 和 **-p** 选项将在以后的版本中从 RHEL 产品中删除。

(JIRA:RHELPLAN-137660)

SMB1 协议在 Samba 中被弃用

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

10.10. DESKTOP

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)

10.11. 图形基础结构

x.org Server 现已弃用

X.org 显示服务器已弃用，并将在以后的主 RHEL 发行版本中删除。现在，在大多数情形中，默认桌面会话都是 **Wayland** 会话。

X11 协议仍完全支持使用 **XWayland** 后端。因此，需要 **X11** 的应用程序可以在 **Wayland** 会话中运行。

红帽正在努力解决 **Wayland** 会话中的剩余问题。有关 **Wayland** 中的未解决的问题，请参阅已知问题部分。

您可以将用户会话切回到 **X.org** 后端。如需更多信息，请参阅选择 GNOME 环境并显示协议。

(JIRA:RHELPLAN-121048)

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)

10.12. RED HAT ENTERPRISE LINUX 系统角色

当在 RHEL 9 节点上配置 team 时， networking 系统角色显示一条弃用警告

RHEL 9 中弃用了网络协作功能。因此，在 RHEL 8 控制器上使用 **networking** RHEL 系统角色在 RHEL 9 节点上配置网络 team 时，显示一条有关其弃用的警告。

(BZ#1999770)

10.13. 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。

(BZ#1935497)

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 监控可能会被阻止，这会影响某些工作负载的 hypervisor 性能。

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:RHELPLAN-15509, BZ#1621944)

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它的替代。因此，建议您使用 web 控制台使用 GUI 管理虚拟化。但请注意，在 RHEL web 控制台中，**virt-manager** 中的一些功能可能还不可用。

（JIRA:RHELPLAN-10304）

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。有关详情请参考 RHEL 9 配置和管理虚拟化 文档。

(JIRA:RHELPLAN-113995)

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。

(BZ#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。

(BZ#1951814)

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 系列前的模型（也称为 Nehalem）

- 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'
```

(BZ#2060839)

10.14. 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)

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。

(BZ#2069279)

rhel9/pause 已被弃用

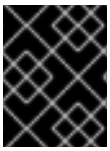
rhel9/pause 容器镜像已弃用。

(BZ#2106816)

10.15. 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：

- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services

- iptables-utils
- libdb
- mcpp
- mod_auth_mellon
- python3-pytz
- xorg-x11-server-Xorg

第 11 章 已知问题

这部分论述了 Red Hat Enterprise Linux 9.1 中已知的问题。

11.1. 安装程序和镜像创建

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。

(BZ#1697896)

当使用第三方工具创建的 USB 引导安装时，不会检测到本地介质安装源

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。
- 使用 Rufus 等第三方工具创建可引导 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 USB 设备。

有关执行任何指定临时解决方案的步骤的更多信息，请参阅安装 RHEL 8.3 过程中不会自动探测到安装介质

(BZ#1877697)

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，请确定安装程序可使用 BaseOS 和 AppStream 软件仓库，或者在安装过程中使用 **authselect** Kickstart 命令。

(BZ#1640697)

驱动程序磁盘菜单无法在控制台上显示用户输入

当您在内核命令行上将 **inst.dd** 选项与驱动程序磁盘一起使用来开始 RHEL 安装时，控制台将无法显示用户输入。因此，看起来应用程序没有对用户输入和冻结做出响应，但会显示让用户混淆的输出。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，要查看预期的结果，请忽略控制台中没有用户输入，并在完成添加输入后按 **Enter** 键。

(BZ#2109231)

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，请不要在生产环境系统上运行 Anaconda，而在临时虚拟机中执行它。因此，生产系统上的 SELinux 策略没有被修改。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。

(BZ#2050140)

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。

（BZ#1914955）

使用 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。

要解决这个问题，请在 kickstart 文件中添加以下脚本，以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。

(BZ#1929105)

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。

(BZ#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。

(BZ#1997832)

当 PReP 大小为 4 或 8 MiB 时，无法安装 RHEL

如果 PowerPC Reference Platform(PReP)分区与使用 4 kiB 扇区的磁盘上的 4 MiB 或 8 MiB 不同，RHEL 安装程序无法安装引导装载程序。因此，您无法在磁盘中安装 RHEL。

要临时解决这个问题，请确保 PReP 分区的大小为 4 MiB 或 8 MiB，且大小没有舍入到另一个值。现在，安装程序可以在磁盘中安装 RHEL。

(BZ#2026579)

安装程序使用多路径设备在自定义分区时显示不正确的磁盘空间总量

在自定义分区时，安装程序不会过滤多路径设备的独立路径。这会导致安装程序显示到多路径设备的独立路径，用户可以为创建的分区选择到多路径设备的独立路径。因此，会显示不正确的磁盘空间总和。通过向总磁盘空间添加每个独立路径的大小来计算它。

作为临时解决方案，在自定义分区时只使用多路径设备而不是单独的路径，并忽略错误计算的磁盘空间。

(BZ#2052938)

安装无法通过光纤通道设备进行 NVMe 失败

安装 RHEL 时，安装程序会显示并允许通过 Fibre Channel 设备选择 Non-volatile Memory Express (NVMe)。不支持在安装过程中使用这样的设备。因此，安装过程可能会失败，或者安装的系统可能无法正确引导。

要临时解决这个问题，在互动安装过程中请不要使用 NVMe over Fibre Channel 设备（文本或者图形模式）。在运行 Kickstart 安装时，将系统配置为使用 **ignoredisk --drives=<IGNORE_DISKS>** Kickstart 命令，将 **<IGNORE_DISKS>** 替换为 NVMe over Fibre Channel 设备。另外，您可以使用 **ignoredisk --only-use=<ONLY_USE_DISKS>** 在安装过程中定义磁盘 Kickstart 使用，将 **<ONLY_USE_DISKS>** 替换为支持的设备。

注意

只使用 Fibre Channel 设备的 NVMe 安装会失败。本地附加 NVMe 设备可以正常工作。

有关 **ignoredisk** Kickstart 命令的详细信息，请参阅执行高级 RHEL 9 安装指南中的 Kickstart 命令处理存储。

(BZ#2107346)

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，安装过程可以成功完成。

(BZ#2125542)

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，取消选择 **Connect automatically with priority**
  e. 点 **Save**。

(BZ#2115783)

11.2. 订阅管理

在完成命令后，subscription-manager 工具会在终端中保留不重要的文本

从 RHEL 9.1 开始，**subscription-manager** 工具会在处理操作时显示进度信息。对于某些语言（通常是非拉丁），在操作完成后进度消息可能没有被清除。因此，您可能会在终端中看到部分旧进度信息。

请注意，这不是 **subscription-manager** 的功能失败。

要临时解决这个问题，请执行以下步骤之一：

- 当在终端中运行 'subscription-manager' 命令时，包括 **--no-progress-messages** 选项
- 输入以下命令将 **subscription-manager** 配置为在不显示进度信息的情况下运行：

  ```
  # subscription-manager config --rhsm.progress_messages=0
  ```

(BZ#2136694)

11.3. 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。

(BZ#2073510)

在通过升级更改其架构的软件包时，安全 DNF 升级会失败

BZ#2108969, 的补丁（通过 RHBA-2022:8295 公共提供）存在以下的回归问题：对于通过升级使其架构更改为 **noarch** 或从其更改，则使用 DNF 升级安全过滤器会失败。因此，它可以使系统处于存在安全漏洞的状态。

要临时解决这个问题，在不出现安全过滤器的情况下执行常规升级。

(BZ#2108969)

11.4. 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 配置文件中设置并导出 **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同样的值，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，并在稍后中止：

> ERROR: Could not create build area

要临时解决这个问题，如果您有一个自定义临时目录，请在执行 ReaR 之前，通过在 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。

(BZ#2018112)

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，请参阅管理 systemd。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。

(BZ#2053598)

11.5. 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。

(BZ#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，可写区域文件通常用于允许由 DNSSEC 维护的动态更新、从属区域或区域的区域。

(BZ#1984982)

设置控制台 键映射 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
```

(BZ#2214130)

11.6. 安全性

openssl 不检测 PKCS #11 令牌是否支持创建原始 RSA 或 RSA-PSS 签名

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 **PKCS #11** 令牌持有密钥，使用 **OpenSSL** 库的服务器应用程序无法使用 **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。

(BZ#1681178)

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。

(BZ#1685470)

当使用特定语法时，scp 会让将文件复制到自己的文件

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**

(BZ#2056884)

PSK 密码suites 无法用于 FUTURE 加密策略

预共享密钥(PSK)密码组合不能被识别为执行完美的转发保密(PFS)密钥交换方法。因此，**ECDHE-PSK** 和 **DHE-PSK** 加密套件无法与配置为 **SECLEVEL=3** 的 OpenSSL 一起工作，例如使用 **FUTURE** 加密策略。作为临时解决方案，您可以为使用 PSK 密码的应用程序设置限制较低的加密策略或设置较低的安全级别(**SECLEVEL**)。

(BZ#2060044)

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，您将使用非安全 SHA-1 签名来防止 GnuPG 降低默认系统安全性。

(BZ#2070722)

gpg-agent 在 FIPS 模式中无法作为 SSH 代理工作

当向 **ssh-agent** 程序添加密钥时，**gpg-agent** 工具会创建 MD5 指纹，即使 FIPS 模式禁用 MD5 摘要。因此，**ssh-add** 工具无法将密钥添加到身份验证代理中。

要临时解决这个问题，请在不使用 **gpg-agent --daemon --enable-ssh-support** 命令的情况下创建 **~/.gnupg/sshcontrol** 文件。例如，您可以以 **<FINGERPRINT> 0** 格式的 **gpg --list-keys** 命令的输出粘贴到 **~/.gnupg/sshcontrol**。因此，**gpg-agent** 充当 SSH 身份验证代理。

(BZ#2073567)

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。

(BZ#2064274)

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)

修正 SCAP 审计规则失败

在修复修复时，对一些与 Audit 配置相关的 SCAP 规则的 Bash 修复不会添加 Audit 密钥。这适用于以下规则：

- **audit_rules_login_events**
- **audit_rules_login_events_faillock**
- **audit_rules_login_events_lastlog**

- **audit_rules_login_events_tallylog**
- **audit_rules_usergroup_modification**
- **audit_rules_usergroup_modification_group**
- **audit_rules_usergroup_modification_gshadow**
- **audit_rules_usergroup_modification_opasswd**
- **audit_rules_usergroup_modification_passwd**
- **audit_rules_usergroup_modification_shadow**
- **audit_rules_time_watch_localtime**
- **audit_rules_mac_modification**
- **audit_rules_networkconfig_modification**
- **audit_rules_sysadmin_actions**
- **audit_rules_session_events**
- **audit_rules_sudoers**
- **audit_rules_sudoers_d**

因此，如果相关的审计规则已经存在，但没有完全符合 OVAL 检查，补救会修复审计规则的功能部分，即路径和访问位，但不添加 Audit 密钥。因此，生成的审计规则可以正常工作，但 SCAP 规则会错误地报告 FAIL。要临时解决这个问题，请手动在审计规则中添加正确的密钥。

(BZ#2120978)

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。

(BZ#2038978)

在测试访问多个 IMAmeasured 文件的系统时，Keylime 可能会失败

如果运行 Keylime 代理的系统会在快速成功的情况下访问由 Integrity 衡量架构 (IMA) 测量的多个文件，则 Keylime 验证程序可能会错误地处理 IMA 日志添加。因此，运行的哈希值与正确的平台配置 Register (PCR) 状态不匹配，系统会在测试时失败。目前没有临时解决方案。

(BZ#2138167)

Keylime 测量的引导策略生成脚本可能会导致分段错误和内核转储

在处理 tpm2_eventlog 工具的输出时，**create_mb_refstate** 脚本可能会错误地计算 **DevicePath** 字段中的数据长度，而不是在处理 **tpm2_eventlog** 工具的输出时使用 **LengthOfDevicePath** 字段的值。因此，脚本会尝试使用错误计算的长度访问无效的内存，这会导致分段错误和核心转储。Keylime 的主要功能不受此问题的影响，但您可能无法生成测量的引导策略。

要临时解决这个问题，请不要使用测量的引导策略，或使用 **tpm2-tools** 软件包中的 **tpm2_eventlog** 工具手动从获取的数据写入策略文件。

(BZ#2140670)

有些 TPM 证书会导致 Keylime registrar 崩溃

tenant.conf 中的 **require_ek_cert** 配置选项应在生产部署中启用，决定 Keylime 租户是否需要来自 Trusted Platform 模块(TPM) 的端到端密钥 (EK) 证书。当执行启用 **require_ek_cert** 的初始身份报价时，Kelime 会尝试验证代理上的 TPM 设备是否与 Keylime TPM 证书存储中的可信证书进行比较。但是，存储中的某些证书都是格式的 x509 证书，并导致密钥精简注册崩溃。目前，这个问题还没有简单的临时解决方案，除非将 **require_ek_cert** 设置为 **false**，并在 **ek_check_script** 选项中定义自定义脚本，这将执行 EK 验证。

(BZ#2142009)

11.7. 网络

nm-cloud-setup 服务从接口中删除了手动配置的辅助 IP 地址

根据从云环境收到的信息，**nm-cloud-setup** 服务会配置网络接口。禁用 **nm-cloud-setup** 以手动配置接口。然而，在某些情况下，主机上的其他服务也可以配置接口。例如，这些服务可以添加辅助 IP 地址。为了避免 **nm-cloud-setup** 删除辅助 IP 地址：

1. 停止并禁用 **nm-cloud-setup** 服务和计时器：

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. 显示可用的连接配置文件：

```
# nmcli connection show
```

3. 激活受影响的连接配置文件：

```
# nmcli connection up "<profile_name>"
```

因此，该服务不再从接口中删除手动配置的辅助 IP 地址。

(BZ#2151040)

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。

(BZ#2013650)

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。

(BZ#2082303)

11.8. 内核

在使用 Mellanox ConnectX-5 适配器的过程中 mlx5 驱动程序失败

在以太网交换机设备驱动程序模型(**switchdev**)模式下，当使用设备管理的流转向(DMFS)参数和 **ConnectX-5** 适配器支持的硬件配置时，**mlx5** 驱动程序失败。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，您需要使用软件管理的流 转向(SMFS)参数而不是 DMFS。

(BZ#2180665)

启用了安全引导的 fadump 可能会导致 GRUB 内存不足(OOM)

在安全引导环境中，GRUB 和 PowerVM 一起分配 512 MB 内存区域，称为 Real Mode Area (RMA)，用于引导内存。区域在引导组件之间划分，如果任何一个组件超过了其分配，则会发生内存不足故障。

通常，默认安装的 **initramfs** 文件系统和 **vmlinux** 符号表在限制内，以避免出现这样的故障。但是，如果在系统中启用了固件辅助转储(FADump)，则默认的 **initramfs** 大小可能会增加，并超过 95 MB。因此，每次系统重启都会导致 GRUB OOM 状态。

要避免这个问题，请不要将安全引导和 FA 转储一起使用。有关如临时何解决这个问题的更多信息和方法，请参阅 https://www.ibm.com/support/pages/node/6846531。

(BZ#2149172)

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构建或放置额外的模块。

(BZ#2103605)

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
```

(BZ#2064708)

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，第二个内核无法在 LUKS 加密目标上捕获崩溃转储文件(**vmcore**)。

使用 **kdumpctl estimate** 命令，您可以查询 **推荐的 crashkernel 值**，这是 **kdump** 所需的内存大小。

要解决这个问题，请按照以下步骤在 LUKS 加密目标中为 **kdump** 配置所需的内存：

1. 输出估计的 **crashkernel** 值：

```
# kdumpctl estimate
```

2. 通过增大 **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。

(BZ#2017401)

在引导时分配崩溃内核内存失败

在某些 Ampere Altra 系统中，在可用内存低于 1GB 时为 **kdump** 分配崩溃内核内存会失败。因此，**kdumpctl** 命令无法启动 **kdump** 服务。

要解决这个问题，请执行以下操作之一：

- 减少 **crashkernel** 参数的值（最少 240 MB）以满足大小要求，例如 **crashkernel=240M**）。
- 使用 **crashkernel=x,high** 选项为 **kdump** 保留大于 4 GB 的崩溃内核内存。

因此，在 Ampere Altra 系统中，**kdump** 崩溃内核内存分配不会失败。

(BZ#2065013)

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** 证书，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。

- 编辑 GRUB 2 配置文件以覆盖默认参数：

a. 将 **delayacct** 选项附加到 **/etc/default/grub** 文件的 **GRUB _CMDLINE_LINUX** 条目。

b. 运行 **grub2-mkconfig** 工具以重新生成引导配置：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

如需更多信息，请参阅如何永久修改内核命令行？

c. 重启系统以使更改生效。

因此，**iotop** 应用程序会显示 **SWAPIN** 和 **IO%** 统计列。

(BZ#2132480)

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。

(BZ#2000616)

iwl7260-firmware 在 Intel Wi-Fi 6 AX200、AX210 和 Lenovo ThinkPad P1 Gen 4 上会破坏 Wi-Fi

在将 **iwl7260-firmware** 或 **iwl7260-wifi** 驱动程序更新至 RHEL 8.7 和/或 RHEL 9.1（及更高版本）后，硬件会处于不正确的内部状态。报告其状态不正确。因此，Intel Wifi 6 卡可能无法正常工作并显示出错信息：

```
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
```

一个没有限制的问题是关闭该系统并重新恢复。不重启。

(BZ#2129288)

dkms 在在 64 位 ARM CPU 上，对带有正确编译的驱动程序的程序失败提供错误的警告，

动态内核模块支持(**dkms**)工具无法识别适用于 4 KB 和 64 KB 页大小的内核的 64 位 ARM CPU 的内核标头。因此，当执行了内核更新且 **kernel-64k-devel** 软件包未安装时，**dkms** 提供一条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，其包含用于两种类型的 ARM CPU 架构的头文件，且不特定于 **dkms** 及其要求。

(JIRA:RHEL-25967)

11.9. 引导加载程序

grubby 的行为与其文档有区别

当您使用 **grubby** 工具添加了一个新内核而不指定任何参数时，**grubby** 会将默认参数传递给新条目。即使不传递 **--copy-default** 参数，也会发生此行为。使用 **--args** 和 **--copy-default** 选项确保这些参数附加到默认参数中，如 **grubby** 文档中所述。

但是，当添加额外的参数（如 **$tuned_params** ）时，**grubby** 工具不会传递这些参数，除非调用了 **--copy-default** 选项。

在这种情况下，有两个临时解决方案：

- 设置 **root=** 参数，并保留 **--args** 为空：

```
# grubby --add-kernel /boot/my_kernel --initrd /boot/my_initrd --args "root=/dev/mapper/rhel-root" --title "entry_with_root_set"
```

- 或者设置 **root=** 参数和指定的参数，但不是默认的参数：

```
# grubby --add-kernel /boot/my_kernel --initrd /boot/my_initrd --args "root=/dev/mapper/rhel-root some_args and_some_more" --title "entry_with_root_set_and_other_args_too"
```

(BZ#2127453)

11.10. 文件系统和存储

如果由 cloud-init 置备并使用 NFSv3 挂载条目配置，Azure 上的 RHEL 实例无法引导

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。

(BZ#2081114)

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。

(BZ#1983602)

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。

(BZ#2033080)

blk-availability systemd 服务取消激活了复杂的设备堆栈

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。

(BZ#2011699)

supported_speeds sysfs 属性报告不正确的速度值

在以前的版本中，因为 **qla2xxx** 驱动程序中的定义不正确，HBA 的 **supported_speeds sysfs** 属性报告 20 Gb/s 速度，而不是预期的 64 Gb/s 速度。因此，如果 HBA 支持 64 Gb/s 链接速度，则 **sysfs supported_speeds** 值不正确，这会影响报告的速度值。

但是现在 HBA 的 **supported_speeds** sysfs 属性返回 100 Gb/s 速度，而不是预期的 64 Gb/s 速度，50 Gb/s 而不是预期的 128 Gb/s 速度。这只会影响报告的速度值，而光纤通道连接上使用的实际链接率是正确的。

(BZ#2069758)

11.11. 动态编程语言、网页和数据库服务器

MySQL 和 MariaDB 中的 --ssl-fips-mode 选项不会改变 FIPS 模式

RHEL 中的 **MySQL** 和 **MariaDB** 的 **--ssl-fips-mode** 选项的工作方式与上游社区版本不同。

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**，**--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** 设置为 **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** 设置为 **OFF**，则 **mysqld** 或 **mariadbd** 服务器守护进程仍以 FIPS 模式运行。

这是因为，应该为整个 RHEL 系统启用或禁用 FIPS 模式，而不是针对特定组件。

因此，请不要在 RHEL 中的 **MySQL** 或 **MariaDB** 中使用 **--ssl-fips-mode** 选项。反之，请确保在整个 RHEL 系统中启用了 FIPS 模式：

- 最好使用启用了 FIPS 模式安装 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。有关以 FIPS 模式安装 RHEL 的详情，请参考使用 FIPS 模式安装该系统。
- 另外，您可以按照将系统切换到 FIPS 模式的步骤将整个 RHEL 系统的 FIPS 模式切换到 FIPS 模式。

(BZ#1991500)

11.12. 编译器和开发工具

某些基于符号的探测无法在 64 位 ARM 架构的 SystemTap 中工作

内核配置禁用 **SystemTap** 所需的某些功能。因此，一些基于符号的探测无法在 64 位 ARM 构架中工作。因此，受影响的 **SystemTap** 脚本可能无法运行，或者可能无法在所需探测点上收集点击。

请注意，这个程序错误已针对使用 RHBA-2022:5259 公告的剩余架构解决。

(BZ#2083727)

11.13. 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。

(BZ#2106043)

在 RHEL 9 客户端上必须将 DEFAULT:SHA1 子策略设置为 PKINIT 来针对 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 系统中启用 SHA-1 算法的支持：

```
# update-crypto-policies --set DEFAULT:SHA1
```

(BZ#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-polices --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 在 RHEL 9 客户端上必须将 DEFAULT:SHA1 子策略设置为 PKINIT 来针对 AD KDC 工作 。

(BZ#2077450)

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进程，请在启用 **AD-SUPPORT** 子策略前参考 FIPS 审核员，以允许技术测量结果支持 AES SHA-1 HMAC 加密类型，然后安装 RHEL IdM：

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

(BZ#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。

(BZ#2106296)

FIPS 模式中的 IdM 不支持使用 NTLMSSP 协议建立双向跨林信任

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 模式中的 IdM 不接受在尝试验证时 AD 域控制器使用的 RC4 NTLM 哈希。

(BZ#2124243)

IdM 到 AD 跨域 TGS 请求失败

IdM Kerberos 票据中的 Privilege Attribute 证书(PAC) 信息现在使用 AES SHA-2 HMAC 加密进行签名，这是 Active Directory (AD) 不支持的。

因此，IdM 到 AD 跨域 TGS 请求（即双向信任设置）失败，并显示以下错误：

```
"Generic error (see e-text) while getting credentials for <service principal>"
```

(BZ#2060421)

IdM Vault 加密和解密在 FIPS 模式中失败

如果启用了 FIPS 模式，则 OpenSSL RSA-PKCS1v15 填充加密会被阻止。IPvquetly, Identity Management (IdM) Vault 无法正常工作，因为 IdM 目前使用 PKCS1v15 padding 来使用传输证书嵌套会话密钥。

(BZ#2089907)

迁移的 IdM 用户可能会因为不匹配域 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)

当以引用模式启动时，目录服务器会意外终止

由于一个程序错误，全局引用模式无法在 Directory Server 中工作。如果您以 **dirsrv** 用户身份使用 **refer** 选项启动 **ns-slapd** 进程，则目录服务器会忽略端口设置并意外终止。尝试以 **root** 用户身份运行进程会更改 SELinux 标签，并可以防止服务以后以正常模式启动。没有可用的临时解决方案。

(BZ#2053204)

在 Directory 服务器中为后缀配置引用失败

如果您在 Directory 服务器中设置了后端引用，使用 **dsconf <instance_name> backend suffix set --state referral** 命令设置后端状态会失败，并显示以下错误：

```
Error: 103 - 9 - 53 - Server is unwilling to perform - [] - need to set nsslapd-referral before moving to referral state
```

因此，为后缀配置引用会失败。要临时解决这个问题：

1. 手动设置 **nsslapd-referral** 参数：

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
add: nsslapd-referral
nsslapd-referral: ldap://remote_server:389/dc=example,dc=com
```

2. 设置后端状态：

```
# dsconf <instance_name> backend suffix set --state referral
```

因此，您可以使用临时解决方案为后缀配置引用。

(BZ#2063140)

dsconf 实用程序没有为 entryUUID 插件创建修复任务的选项

dsconf 实用程序没有提供为 **entryUUID** 插件创建修复任务的选项。因此，管理员无法使用 **dsconf** 创建任务来自动将 **entryUUID** 属性添加到现有条目。作为临时解决方案，请手动创建任务：

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=entryuuid_fixup_<time_stamp>,cn=entryuuid task,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
```

```
basedn: <fixup base tree>
cn: entryuuid_fixup_<time_stamp>
filter: <filtered_entry>
```

创建了任务后，Directory 服务器会修复缺少或无效 **entryUUID** 属性的条目。

(BZ#2047175)

将对 ldap_id_use_start_tls 选项使用默认值时潜在的风险

当不使用 TLS 进行身份查找的情况下使用 **ldap://** 时，可能会对攻击向量构成风险。特别是中间人(MITM)攻击，例如，其通过更改 LDAP 搜索中返回的对象的 UID 或 GID 来允许攻击者冒充用户。

目前，用于强制 TLS 的 **ldap_id_use_start_tls** SSSD 配置选项，默认为 **false**。确保您的设置可在可信环境中操作，并决定是否可以安全地对 **id_provider = ldap** 使用未加密的通信。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是不安全的，请在 **/etc/sssd/sssd.conf** 文件中将 **ldap_id_use_start_tls** 选项设为 **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)

11.14. DESKTOP

升级到 RHEL 9 后将禁用 Firefox 附加组件

如果您从 RHEL 8 升级到 RHEL 9，则之前在 Firefox 中启用的所有附加组件都会被禁用。

要临时解决这个问题，请手动重新安装或更新附加组件。因此，附加组件会如预期启用。

(BZ#2013247)

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。

(BZ#2122636)

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。

(BZ#2060308)

11.15. 图形基础结构

Matrox G200e 在 VGA 显示器上没有显示输出

如果您使用以下系统配置，您的显示可能不会显示图形输出：

- Matrox G200e GPU
- 通过 VGA 控制器连接的显示

因此，您不能在这种配置中使用或安装 RHEL。

要临时解决这个问题，请使用以下步骤：

1. 将系统启动到引导装载程序菜单。
2. 将 **module_blacklist=mgag200** 选项添加到内核命令行中。

因此，RHEL 会按预期引导并显示图形输出，但最大分辨率限制为 16 位颜色深度的 1024x768。

(BZ#1960467)

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)

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)

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)

11.16. WEB 控制台

VNC 控制台在特定解析时无法正常工作

当使用特定显示解析下的虚拟网络计算(VNC)控制台时，您可能会遇到鼠标偏移问题，或者您可能只看到一部分接口。因此，可能无法使用 VNC 控制台。要临时解决这个问题，您可以尝试扩展 VNC 控制台的大小，或使用 Console 选项卡中的 Desktop Viewer 来启动远程查看器。

(BZ#2030836)

11.17. 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**。所描述的操作意外终止并显示 **internal error: process exited while connecting to monitor** 错误，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 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** 来启用 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。

(BZ#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)

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，请在主机的 BIOS 中手动打开 **erms** 和 **fsrm**。

(BZ#2077767)

禁用 AVX 会导致虚拟机变得无法引导

在使用高级 Vector Extensions (AVX) 支持的 CPU 中，尝试引导当前禁用 AVX 的虚拟机目前失败，而是在虚拟机中触发内核 panic。

(BZ#2005173)

迁移后，VNC 无法连接到 UEFI 虚拟机

如果您在迁移虚拟机时启用或禁用消息队列，则虚拟化网络计算 (VNC) 客户端在迁移完成后无法连接到虚拟机。

这个问题只会影响到使用 Open Virtual Machine Firmware (OVMF) 的基于 UEFI 的虚拟机。

(JIRA:RHELPLAN-135600)

在 Windows 虚拟机上不分配 IP 地址故障转移 virtio NIC

目前，当使用故障转移 virtio NIC 启动 Windows 虚拟机时，虚拟机无法为 NIC 分配 IP 地址。因此，NIC 无法设置网络连接。目前，没有临时解决方案。

(BZ#1969724)

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。

(BZ#2084003)

实时迁移后，Broadcom 网络适配器在 Windows 虚拟机上无法正常工作

目前，广播设备（如 Broadcom、Qlogic 或 Marvell）的网络适配器无法在 Windows 虚拟机实时迁移过程中热拔。因此，在迁移完成后，适配器可以正常工作。

此问题只会影响那些使用单根 I/O 虚拟化 (SR-IOV) 附加到 Windows 虚拟机的适配器。

(BZ#2090712, BZ#2091528, BZ#2111319)

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。

(BZ#2052424)

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。

(BZ#1817965)

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。

(BZ#1789206)

使用大量队列可能会导致 Windows 虚拟机失败

当启用了虚拟可信平台模块(vTPM)设备，且将 *multi-queue virtio-net* 功能配置为使用超过 250 个队列时，Windows 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 设备对于打开的文件描述符的最大数量有一个硬性的限制。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。

(BZ#2020146)

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。

(BZ#2073872)

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)

11.18. 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

(BZ#2059545)

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。

(BZ#2037657)

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。

(BZ#1750862)

11.19. 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。要相应地设置插件超时，您可以首先通过运行以下命令估算收集没有超时的插件所需的时间：

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

(BZ#1869561)

11.20. 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)

附录 A. 按组件划分的问题单列表

Bugzilla 和 JIRA ID 列在本文档中以便参考。可公开访问的 Bugzilla 程序错误包括到 ticket 的链接。

组件	票证
389-ds-base	BZ#2052527, BZ#2057063, BZ#2057066, BZ#1872451, BZ#2053204, BZ#2063140, BZ#2047175
NetworkManager	BZ#2068525, BZ#2059608, BZ#2030997, BZ#2079849, BZ#2097293, BZ#2029636, BZ#1894877, BZ#2151040
anaconda	BZ#2059414, BZ#2053710, BZ#2082132, BZ#2050140, BZ#1877697, BZ#1914955, BZ#1929105, BZ#1997832, BZ#2052938, BZ#2107346, BZ#2125542, BZ#2115783
ansible-collection-microsoft-sql	BZ#2066337
ansible-collection-redhat-rhel_mgmt	BZ#2112434
ansible-freeipa	BZ#2076567
bind	BZ#1984982
catatonit	BZ#2074193
chrony	BZ#2047415, BZ#2051441
clevis	BZ#2107078
cloud-init	BZ#1750862
cockpit-appstream	BZ#2030836
cockpit	BZ#2056786
cronie	BZ#2090691
crypto-policies	BZ#2102774, BZ#2070604
cyrus-sasl	BZ#1995600
device-mapper-multipath	BZ#2084365, BZ#2033080, BZ#2011699
distribution	BZ#2063773

组件	票证
dnf-plugins-core	BZ#2066646
dnf	BZ#2053014, BZ#2073510
dotnet7.0	BZ#2112027
dyninst	BZ#2057675
edk2	BZ#1935497
elfutils	BZ#2088774
fapolicyd	BZ#2100041, BZ#2054740, BZ#2070655
firefox	BZ#2013247
firewalld	BZ#2040689, BZ#2039542
frr	BZ#2069563
gcc-toolset-12-annobin	BZ#2077438
gcc-toolset-12-binutils	BZ#2077445
gcc-toolset-12-gcc	BZ#2077465
gcc-toolset-12-gdb	BZ#2077494
gcc	BZ#2063255
gdb	BZ#1870017
gdm	BZ#2097308
gimp	BZ#2047161
glibc	BZ#2033683, BZ#2096191, BZ#2063142, BZ#2077838, BZ#2085529, BZ#2003291, BZ#2091549
gnome-settings-daemon	BZ#2100467
gnupg2	BZ#2070722, BZ#2073567
gnutls	BZ#2042009

组件	票证
golang	BZ#2075169, BZ#2111072, BZ#2092016
grub2	BZ#2074761, BZ#2026579
grubby	BZ#1978226, BZ#1969362, BZ#2127453
httpd	BZ#2079939, BZ#2065677
ipa	BZ#747959, BZ#2091988, BZ#2083218, BZ#2100227, BZ#2084180, BZ#2084166, BZ#2069202, BZ#2057471, BZ#2124243, BZ#2089907
jmc-core	BZ#1980981
kdump-anaconda-addon	BZ#1959203, BZ#2017401
kernel-rt	BZ#2061574
内核	JIRA:RHELPLAN-117713, BZ#2027894, BZ#2066451, BZ#2079368, BZ#2065226, BZ#2013413, BZ#2069045, BZ#2001936, BZ#2097188, BZ#2096127, BZ#2054379, BZ#2073541, BZ#2030922, BZ#1945040, BZ#2100898, BZ#2068432, BZ#2046472, BZ#1613522, BZ#1874182, BZ#1995338, BZ#1570255, BZ#2023416, BZ#2021672, BZ#2000616, BZ#2013650, BZ#2132480, BZ#2060150, BZ#2059545, BZ#2069758, BZ#1960467, BZ#2005173, BZ#2129288
kexec-tools	BZ#2064708, BZ#2065013
keylime	BZ#2138167, BZ#2140670, BZ#2142009
kmod-kvdo	BZ#2064802
kmod	BZ#2103605
krb5	BZ#2068935, BZ#2106043, BZ#2060798, BZ#2077450, BZ#2106296, BZ#2060421
libdnf	BZ#2108969
libnvme	BZ#2099619
libsepol	BZ#2069718, BZ#2079276
libvirt	BZ#2064194, BZ#2014487
libvpd	BZ#2051288

组件	票证
libxcrypt	BZ#2034569
llvm-toolset	BZ#2061041
lsvpd	BZ#2051289
lvm2	BZ#2038183
maven	BZ#2083112
mysql	BZ#1991500
nfs-utils	BZ#2081114
nmstate	BZ#2084474, BZ#2082043
nodejs	BZ#2083072
nss	BZ#2091905
nvme-cli	BZ#2090121
nvme-stas	BZ#1893841
open-vm-tools	BZ#2061193, BZ#2037657
opencryptoki	BZ#2044179
openscap	BZ#2109485
openssh	BZ#2066882, BZ#2087121, BZ#2056884
openssl	BZ#2060510, BZ#2053289, BZ#2066412, BZ#2063947, BZ#2004915, BZ#2058663, BZ#1975836, BZ#1681178, BZ#1685470, BZ#2060044, BZ#2071631
pacemaker	BZ#2121838, BZ#2072108
pause-container	BZ#2106816
pcre2	BZ#2086494
pcs	BZ#2024522, BZ#2054671, BZ#2058251, BZ#2058252, BZ#2058246, BZ#2058243, BZ#1301204

组件	票证
php	BZ#2070040
pki-core	BZ#2084181
podman	BZ#2097708, BZ#2027576, BZ#2069279
policycoreutils	BZ#2115242
powerpc-utils	BZ#1920964
ppc64-diag	BZ#2051286
procps-ng	BZ#2052536, BZ#2003033
pykickstart	BZ#2083269
qemu-kvm	BZ#2044218, BZ#1965079, BZ#1951814, BZ#2060839, BZ#2014229, BZ#2052424, BZ#1817965, BZ#1789206, BZ#2090712, BZ#2020146
rear	BZ#2111059, BZ#2097437, BZ#2115958, BZ#2083272, BZ#2120736, BZ#2119501
resource-agents	BZ#1826455
rhel-system-roles	BZ#2072385, BZ#2086965, BZ#2065337, BZ#2079622, BZ#2043010, BZ#2065383, BZ#2112145, BZ#2052081, BZ#2052086, BZ#2065392, BZ#2072742, BZ#2072745, BZ#2072746, BZ#2075119, BZ#2078989, BZ#2079627, BZ#2093423, BZ#2100292, BZ#2100942, BZ#2115154, BZ#2115157, BZ#2115152, BZ#2051737, BZ#2065382, BZ#2065394, BZ#2115886, BZ#2100605, BZ#2060523, BZ#2060525, BZ#2065393, BZ#2070462, BZ#2083376, BZ#2083410, BZ#2100286, BZ#2109998, BZ#2115156, BZ#2071804, BZ#2100294, BZ#1999770
rsyslog	BZ#2064318
rust	BZ#2075337
s390utils	BZ#1870699, BZ#1932480
samba	BZ#2077487, Jira:RHELDOCS-16612
sblim-wbemcli	BZ#2083577
scap-security-guide	BZ#2070563, BZ#2120978, BZ#2038978
selinux-policy	BZ#1965013, BZ#2081425, BZ#2076681, BZ#2064274

组件	票证
scs	BZ#1869561
sssd	BZ#1978119, BZ#2065693, BZ#2056482
stalld	BZ#2107275
stratisd	BZ#1990905, BZ#2040352, BZ#2039960, BZ#2007018, BZ#2005110, BZ#2041558
subscription-manager	BZ#2092014, BZ#2136694
systemd	BZ#2018112
systemtap	BZ#2083727
tigervnc	BZ#2060308
tpm2-tools	BZ#2090748
tuned	BZ#2093847
ubi8-container	BZ#2120378
udisks2	BZ#1983602
unbound	BZ#2087120, BZ#2071543, BZ#2070495
valgrind	BZ#1993976
virt-who	BZ#2054504
virtio-win	BZ#1969724, BZ#2084003
whois	BZ#2054043
xmlstarlet	BZ#2069689
xorg-x11-server	BZ#1894612

组件	票证
其他	JIRA:RHELPLAN-92522, BZ#2125549, BZ#2128016, BZ#1937031, JIRA:RHELPLAN-121982, JIRA:RHELPLAN-95456, JIRA:RHELPLAN-122321, JIRA:RHELPLAN-118462, JIRA:RHELPLAN-101140, JIRA:RHELPLAN-132023, JIRA:RHELPLAN-123369, JIRA:RHELPLAN-117109, JIRA:RHELPLAN-130379, BZ#2049492, JIRA:RHELPLAN-130376, JIRA:RHELPLAN-122735, BZ#2070793, BZ#2122716, JIRA:RHELPLAN-123368, JIRA:RHELPLAN-135601, JIRA:RHELPLAN-135602, BZ#2139877, JIRA:RHELPLAN-122776, JIRA:RHELPLAN-121180, BZ#2094015, JIRA:RHELPLAN-109067, JIRA:RHELPLAN-115603, JIRA:RHELPLAN-65217, BZ#2020529, BZ#2030412, BZ#2046653, JIRA:RHELPLAN-103993, JIRA:RHELPLAN-122345, JIRA:RHELPLAN-129327, JIRA:RHELPLAN-74672, BZ#1927780, JIRA:RHELPLAN-110763, BZ#1935544, BZ#2089200, JIRA:RHELPLAN-15509, JIRA:RHELPLAN-99136, JIRA:RHELPLAN-103232, BZ#1899167, BZ#1979521, JIRA:RHELPLAN-100087, JIRA:RHELPLAN-100639, JIRA:RHELPLAN-10304, BZ#2058153, JIRA:RHELPLAN-113995, JIRA:RHELPLAN-121048, JIRA:RHELPLAN-98983, JIRA:RHELPLAN-131882, JIRA:RHELPLAN-137660, BZ#1640697, BZ#1697896, BZ#2047713, JIRA:RHELPLAN-96940, JIRA:RHELPLAN-117234, JIRA:RHELPLAN-119001, JIRA:RHELPLAN-119852, BZ#2077767, BZ#2053598, BZ#2082303, JIRA:RHELPLAN-121049, JIRA:RHELPLAN-109613, JIRA:RHELPLAN-135600, BZ#2149172

附录 B. 修改历史记录

0.2-6

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)。

0.2-5

2024 年 6 月 11 日星期二,Brian Angelica (bangelic@redhat.com)

- 添加了一个已知问题 JIRA:RHEL-24847 (Shells 和命令行工具)。

0.2-4

2024 年 5 月 16 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 JIRA:RHEL-10019（虚拟化）。

0.2-3

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）

0.2-2

2024 年 2 月 1 日星期四，Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 KI BZ#1834716（安全）。

0.2-1

2023 年 11 月 13 日星期一, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个技术预览 JIRA:RHELDOCS-17040 （虚拟化）

0.2-0

2023 年 11 月 10 日星期五, Gabriela Fialová(gfialova@redhat.com)

- 更新了对 RHEL 文档提供反馈的模块。

0.1-9

2023 年 11 月 10 日星期五, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个技术预览 JIRA:RHELDOCS-17050 （虚拟化）。

0.1-8

2023 年 10 月 13 日星期五, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个技术预览 JIRA:RHELDOCS-16861 （容器）。

0.1-7

2023 年 9 月 25 日, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 KI BZ#2122636 (桌面)。

0.1-6

2023 年 9 月 8 日, Marc Muehlfeld (mmuehlfeld@redhat.com)

- 添加了一个已弃用的功能发行注记 JIRA:RHELDOCS-16612 (Samba)。
- 更新了"对红帽文档提供反馈"，以在 JIRA 中反映 RHEL。

0.1-5

2023 年 8 月 17 日, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 Enh BZ#2136937 (Plumbers)。

0.1-4

2023 年 8 月 7 日, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个 KI BZ#2214130 (CS)。

0.1-3

2023 年 8 月 2 日, Marc Muehlfeld (mmuehlfeld@redhat.com)

- 更新了一个已弃用的功能发行注记 BZ#1894877 (NetworkManager)。

0.1-2

2023 年 7 月 25 日, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 BZ#2109231 （安装程序）。

0.1-1

2023 年 6 月 15 日星期四, Lucie Vařáková (lvarakova@redhat.com)

- 添加了一个新功能 BZ#2070725(引导加载程序)。
- 其他小更新。

0.1-0

2023 年 5 月 17 日, Gabriela Fialová(gfialova@redhat.com)

- 使用生命周期信息更新了 已弃用的软件包 部分。

0.0-9

2023 年 4 月 27 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 JIRA:RHELPLAN-155168（身份管理）。

0.0-8

2023 年 4 月 25 日, Lucie Vařáková (lvarakova@redhat.com)

- 添加了一个已知问题 BZ#2180665 (内核)。

0.0-7

2023 年 2 月 20 日星期一， Gabriela Fialová(gfialova@redhat.com)

- 将有关 SAP 环境的信息添加到 从 RHEL 8 原位升级到 RHEL 9 。

0.0-6

2023 年 2 月 16 日星期四，Gabriela Fialová(gfialova@redhat.com)

- 更新了一个已知问题 BZ#2132480(内核)。

0.0-5

2023 年 2 月 14 日星期二，Gabriela Fialová(gfialova@redhat.com)

- 在 对外部内核参数的重要更改 中进行了一个小的格式更改。

0.0-4

2023 年 2 月 14 日星期二, Marc Muehlfeld (mmuehlfeld@redhat.com)

- 添加了一个增强 BZ#2144898（网络）。

0.0-3

2022 年 12 月 7 日星期三, Gabriela Fialová(gfialova@redhat.com)

- 将 **nodejs:18** 模块流 BZ#2083072 从技术预览移到完全支持的功能(动态编程语言、web 和数据库服务器)。

0.0-2

Wed Nov 16, 2022, Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.1 发行注记。

0.0-1

2022 年 9 月 28 日，Gabriela Fialová(gfialova@redhat.com)

- 发布 Red Hat Enterprise Linux 9.1 Beta 发行注记。