Red Hat Enterprise Linux 8

8.0 Release Notes

Release Notes for Red Hat Enterprise Linux 8.0

Last Updated: 2024-05-10

Red Hat Enterprise Linux 8 8.0 Release Notes

Release Notes for Red Hat Enterprise Linux 8.0

Legal Notice

Abstract

The Release Notes provide high-level coverage of the improvements and additions that have been implemented in Red Hat Enterprise Linux 8.0 and document known problems in this release, as well as notable bug fixes, Technology Previews, deprecated functionalities, and other details.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. OVERVIEW

Based on Fedora 28 and the upstream kernel 4.18, Red Hat Enterprise Linux 8.0 provides users with a stable, secure, consistent foundation across hybrid cloud deployments with the tools needed to support traditional and emerging workloads. Highlights of the release include:

Distribution

- Content is available through the **BaseOS** and Application Stream (**AppStream**) repositories.
- The **AppStream** repository supports a new extension of the traditional RPM format - *modules*. This allows for multiple major versions of a component to be available for install.

See Chapter 3, *Distribution of content in RHEL 8* for more information.

Software Management

- The **YUM** package manager is now based on the **DNF** technology and it provides support for modular content, increased performance, and a well-designed stable API for integration with tooling.

See Section 5.1.4, "Software management" for more details.

Shells and command-line tools

- RHEL 8 provides the following **version control systems**: **Git 2.18**, **Mercurial 4.8**, and **Subversion 1.10**.

See Section 5.1.6, "Shells and command-line tools" for details.

Dynamic programming languages, web and database servers

- **Python 3.6** is the default **Python** implementation in RHEL 8; limited support for **Python 2.7** is provided. No version of Python is installed by default.
- **Node.js** is new in RHEL. Other **dynamic programming languages** have been updated since RHEL 7: **PHP 7.2**, Ruby **2.5**, **Perl 5.26**, **SWIG 3.0** are now available.
- The following **database servers** are distributed with RHEL 8: **MariaDB 10.3**, **MySQL 8.0**, **PostgreSQL 10**, **PostgreSQL 9.6**, and **Redis 5**.
- RHEL 8 provides the **Apache HTTP Server 2.4** and introduces a new **web server**, **nginx 1.14**.
- **Squid** has been updated to version 4.4, and a new **proxy caching server** is now included: **Varnish Cache 6.0**.

See Section 5.1.7, "Dynamic programming languages, web and database servers" for more information.

Desktop

- **GNOME Shell** has been rebased to version 3.28.
- The GNOME session and the GNOME Display Manager use **Wayland** as their default display server. The **X.Org** server, which is the default display server in RHEL 7, is available as well.

See Section 5.1.8, "Desktop" for more information.

Installer and image creation

- The **Anaconda** installer can utilize **LUKS2** disk encryption, and install the system on **NVDIMM** devices.
- The **Image Builder** tool enables users to create customized system images in a variety of formats, including images prepared for deployment on clouds of various providers.
- Installation from a DVD using Hardware Management Console (**HMC**) and Support Element (**SE**) on **IBM Z** are available in RHEL 8.

See Section 5.1.2, "Installer and image creation" for further details.

Kernel

- The extended Berkeley Packet Filtering (**eBPF**) feature enables the user space to attach custom programs onto a variety of points (sockets, trace points, packet reception) to receive and process data. This feature is available as a **Technology Preview**.
- BPF Compiler Collection (**BCC**), a tool for creating efficient kernel tracing and manipulation programs, is available as a **Technology Preview**.

See Section 5.3.1, "Kernel" for more information.

File systems and storage

- The LUKS version 2 (**LUKS2**) format replaces the legacy LUKS (LUKS1) format. The **dm-crypt** subsystem and the **cryptsetup** tool now uses LUKS2 as the default format for encrypted volumes.

See Section 5.1.12, "File systems and storage" for more information.

Security

- System-wide **cryptographic policies**, which configures the core cryptographic subsystems, covering the TLS, IPsec, SSH, DNSSEC, and Kerberos protocols, are applied by default. With the new **update-crypto-policies** command, the administrator can easily switch between modes: default, legacy, future, and fips.
- Support for **smart cards** and Hardware Security Modules (**HSM**) with **PKCS #11** is now consistent across the system.

See Section 5.1.15, "Security" for more information.

Networking

- The **nftables** framework replaces **iptables** in the role of the default network packet filtering facility.
- The **firewalld** daemon now uses **nftables** as its default backend.
- Support for **IPVLAN** virtual network drivers that enable the network connectivity for multiple containers has been introduced.
- The eXpress Data Path (**XDP**), XDP for Traffic Control (**tc**), and Address Family eXpress Data Path (**AF_XDP**), as parts of the extended Berkeley Packet Filtering (**eBPF**) feature, are available as **Technology Previews**. For more details, see Section 5.3.7, "Networking" in Technology Previews.

See Section 5.1.14, "Networking" in New features for additional features.

Virtualization

- A more modern PCI Express-based machine type (**Q35**) is now supported and automatically configured in virtual machines created in RHEL 8. This provides a variety of improvements in features and compatibility of virtual devices.
- Virtual machines can now be created and managed using the RHEL 8 web console, also known as **Cockpit**.
- The **QEMU** emulator introduces the **sandboxing** feature, which provides configurable limitations to what systems calls QEMU can perform, and thus makes virtual machines more secure.

See Section 5.1.16, "Virtualization" for more information.

Compilers and development tools

- The **GCC** compiler based on version 8.2 brings support for more recent C++ language standard versions, better optimizations, new code hardening techniques, improved warnings, and new hardware features.
- Various tools for code generation, manipulation, and debugging can now experimentally handle the **DWARF5** debugging information format.
- Kernel support for **eBPF** tracing is available for some tools, such as **BCC**, **PCP**, and **SystemTap**.
- The **glibc** libraries based on version 2.28 add support for Unicode 11, newer Linux system calls, key improvements in the DNS stub resolver, additional security hardening, and improved performance.
- RHEL 8 provides OpenJDK 11, OpenJDK 8, IcedTea-Web, and various **Java** tools, such as **Ant**, **Maven**, or **Scala**.

See Section 5.1.11, "Compilers and development tools" for additional details.

High availability and clusters

- The **Pacemaker** cluster resource manager has been upgraded to upstream version 2.0.0, which provides a number of bug fixes and enhancements.
- In RHEL 8, the **pcs** configuration system fully supports Corosync 3, **knet**, and node names.

See Section 5.1.13, "High availability and clusters" for more information.

Additional resources

- **Capabilities and limits** of Red Hat Enterprise Linux 8 as compared to other versions of the system are available in the Knowledgebase article Red Hat Enterprise Linux technology capabilities and limits.
- Information regarding the Red Hat Enterprise Linux **life cycle** is provided in the Red Hat Enterprise Linux Life Cycle document.
- The Package manifest document provides a **package listing** for RHEL 8.

- Major **differences between RHEL 7 and RHEL 8** are documented in Considerations in adopting RHEL 8.
- Instructions on how to perform an **in-place upgrade from RHEL 7 to RHEL 8** are provided by the document Upgrading from RHEL 7 to RHEL 8 .
- Crrently supported upgrade paths are listed in Supported in-place upgrade paths for Red Hat Enterprise Linux.
- The **Red Hat Insights** service, which enables you to proactively identify, examine, and resolve known technical issues, is now available with all RHEL subscriptions. For instructions on how to install the Red Hat Insights client and register your system to the service, see the Red Hat Insights Get Started page.

Red Hat Customer Portal Labs

Red Hat Customer Portal Labs is a set of tools in a section of the Customer Portal available at https://access.redhat.com/labs/. The applications in Red Hat Customer Portal Labs can help you improve performance, quickly troubleshoot issues, identify security problems, and quickly deploy and configure complex applications. Some of the most popular applications are:

- Registration Assistant
- Kickstart Generator
- Product Life Cycle Checker
- Red Hat Product Certificates
- Red Hat Satellite Upgrade Helper
- Red Hat CVE Checker
- JVM Options Configuration Tool
- Load Balancer Configuration Tool
- Red Hat Code Browser
- Yum Repository Configuration Helper

CHAPTER 2. ARCHITECTURES

Red Hat Enterprise Linux 8.0 is distributed with the kernel version 4.18.0-80, which provides support for the following architectures:

- AMD and Intel 64-bit architectures
- The 64-bit ARM architecture
- IBM Power Systems, Little Endian
- 64-bit IBM Z

Make sure you purchase the appropriate subscription for each architecture. For more information, see Get Started with Red Hat Enterprise Linux - additional architectures . For a list of available subscriptions, see Subscription Utilization on the Customer Portal.

CHAPTER 3. DISTRIBUTION OF CONTENT IN RHEL 8

3.1. INSTALLATION

Red Hat Enterprise Linux 8 is installed using ISO images. Two types of ISO image are available for the AMD64, Intel 64-bit, 64-bit ARM, IBM Power Systems, and IBM Z architectures:

- Binary DVD ISO: A full installation image that contains the BaseOS and AppStream repositories and allows you to complete the installation without additional repositories.

NOTE

The Binary DVD ISO image is larger than 4.7 GB, and as a result, it might not fit on a single-layer DVD. A dual-layer DVD or USB key is recommended when using the Binary DVD ISO image to create bootable installation media. You can also use the Image Builder tool to create customized RHEL images. For more information about Image Builder, see the *Composing a customized RHEL system image* document.

- Boot ISO: A minimal boot ISO image that is used to boot into the installation program. This option requires access to the BaseOS and AppStream repositories to install software packages. The repositories are part of the Binary DVD ISO image.

See the Performing a standard RHEL 8 installation document for instructions on downloading ISO images, creating installation media, and completing a RHEL installation. For automated Kickstart installations and other advanced topics, see the Performing an advanced 8 RHEL installation document.

3.2. REPOSITORIES

Red Hat Enterprise Linux 8 is distributed through two main repositories:

- BaseOS
- AppStream

Both repositories are required for a basic RHEL installation, and are available with all RHEL subscriptions.

Content in the BaseOS repository is intended to provide the core set of the underlying OS functionality that provides the foundation for all installations. This content is available in the RPM format and is subject to support terms similar to those in previous releases of RHEL. For a list of packages distributed through BaseOS, see the Package manifest.

Content in the Application Stream repository includes additional user space applications, runtime languages, and databases in support of the varied workloads and use cases. Application Streams are available in the familiar RPM format, as an extension to the RPM format called *modules*, or as Software Collections. For a list of packages available in AppStream, see the Package manifest.

In addition, the CodeReady Linux Builder repository is available with all RHEL subscriptions. It provides additional packages for use by developers. Packages included in the CodeReady Linux Builder repository are unsupported.

For more information about RHEL 8 repositories, see the Package manifest.

3.3. APPLICATION STREAMS

Red Hat Enterprise Linux 8 introduces the concept of Application Streams. Multiple versions of user space components are now delivered and updated more frequently than the core operating system packages. This provides greater flexibility to customize Red Hat Enterprise Linux without impacting the underlying stability of the platform or specific deployments.

Components made available as Application Streams can be packaged as modules or RPM packages and are delivered through the AppStream repository in RHEL 8. Each Application Stream component has a given life cycle, either the same as RHEL 8 or shorter. For details, see Red Hat Enterprise Linux Life Cycle.

Modules are collections of packages representing a logical unit: an application, a language stack, a database, or a set of tools. These packages are built, tested, and released together.

Module streams represent versions of the Application Stream components. For example, several streams (versions) of the PostgreSQL database server are available in the **postgresql** module with the default **postgresql:10** stream. Only one module stream can be installed on the system. Different versions can be used in separate containers.

Detailed module commands are described in the Installing, managing, and removing user-space components document. For a list of modules available in AppStream, see the Package manifest.

3.4. PACKAGE MANAGEMENT WITH YUM/DNF

On Red Hat Enterprise Linux 8, installing software is ensured by the **YUM** tool, which is based on the **DNF** technology. We deliberately adhere to usage of the **yum** term for consistency with previous major versions of RHEL. However, if you type **dnf** instead of **yum**, the command works as expected because **yum** is an alias to **dnf** for compatibility.

For more details, see the following documentation:

- Installing, managing, and removing user-space components
- Considerations in adopting RHEL 8

CHAPTER 4. RHEL 8.0.1 RELEASE

4.1. NEW FEATURES

RHEL System Roles updated

The **rhel-system-roles** packages, which provide a configuration interface for RHEL subsystems, have been updated. Notable changes include:

- Handling of absent profiles in the **network** role has been improved. When deleting an existing NetworkManager on-disk profile configuration by setting the persistent state to **absent**, only the persistent configuration for the profile is now removed, and the current runtime configuration remains unchanged. As a result, the corresponding network device is no longer brought down in the described situation.
- Specifying a Maximum Transmission Unit (MTU) size for VLAN and MACVLAN interfaces in the **network** role has been fixed. As a result, setting MTU size on VLAN and MACVLAN interfaces using the **network** role no longer fails with the following error message:

  ```
  failure: created connection failed to normalize: nm-connection-error-quark:
  connection.type: property is missing (6)
  ```

- The **selinux** and **timesync** roles now include all their documented input variables in their defaults files (**defaults/main.yml**). This makes it easy to determine what input variables are supported by the roles by examining the content of their respective defaults files.
- The **kdump** and **timesync** roles have been fixed to not fail in check mode.

(BZ#1685902, BZ#1674004, BZ#1685904)

sos-collector rebased to version 1.7

The **sos-collector** packages have been updated to version 1.7 in RHEL 8.0.1. Notable changes include:

- **sos-collector** can now collect sosreports from Red Hat Enterprise Linux CoreOS (RHCOS) nodes in the same way as from regular RHEL nodes. Users do not need to make any changes to the way they run **sos-collector**. Identification of when a node is RHCOS or RHEL is automatic.
- When collecting from RHCOS nodes, **sos-collector** will create a temporary container on the node and use the **support-tools** container to generate a sosreport. This container will be removed after completion.
- Using the **--cluster-type=none** option allows users to skip all cluster-related checks or modifications to the **sosreport** command that gets run on the nodes, and simply collect from a static list of nodes passed through the **--nodes** parameter.
- Red Hat Satellite is now a supported cluster type to allow collecting sosreports from the Satellite and any Capsules.

(BZ#1695764)

Upgraded compiler toolsets

The following compiler toolsets, distributed as Application Streams, have been upgraded with RHEL 8.0.1:

- Rust Toolset, which provides the Rust programming language compiler **rustc**, the **cargo** build tool and dependency manager, and required libraries, to version 1.35
- Go Toolset, which provides the Go (**golang**) programming language tools and libraries, to version 1.11.6.

(BZ#1731500)

Enabling and disabling SMT

Simultaneous Multi-Threading (SMT) configuration is now available in RHEL 8. Disabling SMT in the web console allows you to mitigate a class of CPU security vulnerabilities such as:

- Microarchitectural Data Sampling
- L1 Terminal Fault Attack

(BZ#1713186)

4.2. KNOWN ISSUES

Performance deterioration in IPSec tunnels

Using the **aes256_sha2** or the **aes-gcm256** IPSec cipher set in RHEL 8.0.1 has a negative performance impact on IPSec tunnels. Users with specific VPN settings will experience 10% performance deterioration for IPSec tunnels. This regression is not caused by Microarchitectural Data Sampling (MDS) mitigations; it can be observed with the mitigations both on and off.

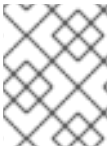
(BZ#1731362)

CHAPTER 5. RHEL 8.0.0 RELEASE

5.1. NEW FEATURES

This part describes new features and major enhancements introduced in Red Hat Enterprise Linux 8.

5.1.1. The web console



NOTE

The web console's Subscriptions page is now provided by the new **subscription-manager-cockpit** package.

A firewall interface has been added to the web console

The **Networking** page in the RHEL 8 web console now includes a **Firewall** section. In this section, users can enable or disable the firewall, as well as add, remove, and modify firewall rules.

(BZ#1647110)

The web console is now available by default

Packages for the RHEL 8 web console, also known as Cockpit, are now part of Red Hat Enterprise Linux default repositories, and can therefore be immediately installed on a registered RHEL 8 system.

In addition, on a non-minimal installation of RHEL 8, the web console is automatically installed and firewall ports required by the console are automatically open. A system message has also been added prior to login that provides information about how to enable or access the web console.

(JIRA:RHELPLAN-10355)

Better IdM integration for the web console

If your system is enrolled in an Identity Management (IdM) domain, the RHEL 8 web console now uses the domain's centrally managed IdM resources by default. This includes the following benefits:

- The IdM domain's administrators can use the web console to manage the local machine.
- The console's web server automatically switches to a certificate issued by the IdM certificate authority (CA) and accepted by browsers.
- Users with a Kerberos ticket in the IdM domain do not need to provide login credentials to access the web console.
- SSH hosts known to the IdM domain are accessible to the web console without manually adding an SSH connection.

Note that for IdM integration with the web console to work properly, the user first needs to run the **ipa-advise** utility with the **enable-admins-sudo** option in the IdM master system.

(JIRA:RHELPLAN-3010)

The web console is now compatible with mobile browsers

With this update, the web console menus and pages can be navigated on mobile browser variants. This makes it possible to manage systems using the RHEL 8 web console from a mobile device.

(JIRA:RHELPLAN-10352)

The web console front page now displays missing updates and subscriptions

If a system managed by the RHEL 8 web console has outdated packages or a lapsed subscription, a warning is now displayed on the web console front page of the system.

(JIRA:RHELPLAN-10353)

The web console now supports PBD enrollment

With this update, you can use the the RHEL 8 web console interface to apply Policy-Based Decryption (PBD) rules to disks on managed systems. This uses the Clevis decryption client to facilitate a variety of security management functions in the web console, such as automatic unlocking of LUKS-encrypted disk partitions.

(JIRA:RHELPLAN-10354)

Virtual Machines can now be managed using the web console

The **Virtual Machines** page can now be added to the RHEL 8 web console interface, which enables the user to create and manage libvirt-based virtual machines.

(JIRA:RHELPLAN-2896)

5.1.2. Installer and image creation

Installing RHEL from a DVD using SE and HMC is now fully supported on IBM Z

The installation of Red Hat Enterprise Linux 8 on IBM Z hardware from a DVD using the **Support Element (SE)** and **Hardware Management Console (HMC)** is now fully supported. This addition simplifies the installation process on IBM Z with **SE** and **HMC**.

When booting from a binary DVD, the installer prompts the user to enter additional kernel parameters. To set the DVD as an installation source, append **inst.repo=hmc** to the kernel parameters. The installer then enables **SE** and **HMC** file access, fetches the images for stage2 from the DVD, and provides access to the packages on the DVD for software selection.

The new feature eliminates the requirement of an external network setup and expands the installation options.

(BZ#1500792)

Installer now supports the LUKS2 disk encryption format

Red Hat Enterprise Linux 8 installer now uses the LUKS2 format by default but you can select a LUKS version from **Anaconda's** Custom Partitioning window or by using the new options in Kickstart's **autopart**, **logvol**, **part**, and **RAID** commands.

LUKS2 provides many improvements and features, for example, it extends the capabilities of the on-disk format and provides flexible ways of storing metadata.

(BZ#1547908)

Anaconda supports System Purpose in RHEL 8

Previously, **Anaconda** did not provide system purpose information to **Subscription Manager**. In Red Hat Enterprise Linux 8.0, you can set the intended purpose of the system during installation by using **Anaconda's System Purpose** window or Kickstart's **syspurpose** command. When the installation

completes, **Subscription Manager** uses the system purpose information when subscribing the system.

(BZ#1612060)

Pykickstart supports System Purpose in RHEL 8

Previously, it was not possible for the **pykickstart** library to provide system purpose information to **Subscription Manager**. In Red Hat Enterprise Linux 8.0, **pykickstart** parses the new **syspurpose** command and records the intended purpose of the system during automated and partially-automated installation. The information is then passed to **Anaconda**, saved on the newly-installed system, and available for **Subscription Manager** when subscribing the system.

(BZ#1612061)

Anaconda supports a new kernel boot parameter in RHEL 8

Previously, you could only specify a base repository from the kernel boot parameters. In Red Hat Enterprise Linux 8, a new kernel parameter, **inst.addrepo=<name>,<url>**, allows you to specify an additional repository during installation.

This parameter has two mandatory values: the name of the repository and the URL that points to the repository. For more information, see https://anaconda-installer.readthedocs.io/en/latest/boot-options.html#inst-addrepo

(BZ#1595415)

Anaconda supports a unified ISO in RHEL 8

In Red Hat Enterprise Linux 8.0, a unified ISO automatically loads the BaseOS and AppStream installation source repositories.

This feature works for the first base repository that is loaded during installation. For example, if you boot the installation with no repository configured and have the unified ISO as the base repository in the GUI, or if you boot the installation using the **inst.repo=** option that points to the unified ISO. As a result, the AppStream repository is enabled under the **Additional Repositories** section of the **Installation Source** GUI window. You cannot remove the AppStream repository or change its settings but you can disable it in **Installation Source**. This feature does not work if you boot the installation using a different base repository and then change it to the unified ISO. If you do that, the base repository is replaced. However, the AppStream repository is not replaced and points to the original file.

(BZ#1610806)

Anaconda can install modular packages in Kickstart scripts

The Anaconda installer has been extended to handle all features related to application streams: modules, streams and profiles. Kickstart scripts can now enable module and stream combinations, install module profiles, and install modular packages. For more information, see Performing an advanced RHEL installation.

(JIRA:RHELPLAN-1943)

The nosmt boot option is now available in the RHEL 8 installation options

The **nosmt** boot option is available in the installation options that are passed to a newly-installed RHEL 8 system.

(BZ#1677411)

RHEL 8 supports installing from a repository on a local hard drive

Previously, installing RHEL from a hard drive required an ISO image as the installation source. However, the RHEL 8 ISO image might be too large for some file systems; for example, the FAT32 file system cannot store files larger than 4 GiB.

In RHEL 8, you can enable installation from a repository on a local hard drive. You only need to specify the directory instead of the ISO image. For example:`inst.repo=hd:<device>:<path to the repository>`

(BZ#1502323)

Custom system image creation with Image Builder is available in RHEL 8

The Image Builder tool enables users to create customized RHEL images. Image Builder is available in AppStream in the **lorax-composer** package.

With Image Builder, users can create custom system images which include additional packages. Image Builder functionality can be accessed through:

- a graphical user interface in the web console
- a command line interface in the **composer-cli** tool.

Image Builder output formats include, among others:

- live ISO disk image
- qcow2 file for direct use with a virtual machine or OpenStack
- file system image file
- cloud images for Azure, VMWare and AWS

To learn more about Image Builder, see the documentation title Composing a customized RHEL system image.

(JIRA:RHELPLAN-7291, BZ#1628645, BZ#1628646, BZ#1628647, BZ#1628648)

Added new kickstart commands: authselect and modules

With this release, the following kickstart commands are added:

- **authselect**: Use the **authselect** command to set up the system authentication options during installation. You can use **authselect** as a replacement for deprecated **auth** or **authconfig** Kickstart commands. For more information, see the **authselect** section in the Performing an advanced installation guide.
- **module**: Use the **module** command to enable a package module stream within the kickstart script. For more information, see the **module** section in the Performing an advanced installation guide.

(BZ#1972210)

5.1.3. Kernel

Kernel version in RHEL 8.0

Red Hat Enterprise Linux 8.0 is distributed with the kernel version 4.18.0-80.

(BZ#1797671)

ARM 52-bit physical addressing is now available

With this update, support for 52-bit physical addressing (PA) for the 64-bit ARM architecture is available. This provides larger address space than previous 48-bit PA.

(BZ#1643522)

The IOMMU code supports 5-level page tables in RHEL 8

The I/O memory management unit (IOMMU) code in the Linux kernel has been updated to support 5-level page tables in Red Hat Enterprise Linux 8.

(BZ#1485546)

Support for 5-level paging

New **P4d_t** software page table type has been added into the Linux kernel in order to support 5-level paging in Red Hat Enterprise Linux 8.

(BZ#1485532)

Memory management supports 5-level page tables

With Red Hat Enterprise Linux 7, existing memory bus had 48/46 bit of virtual/physical memory addressing capacity, and the Linux kernel implemented 4 levels of page tables to manage these virtual addresses to physical addresses. The physical bus addressing line put the physical memory upper limit capacity at 64 TB.

These limits have been extended to 57/52 bit of virtual/physical memory addressing with 128 PiB of virtual address space and 4 PB of physical memory capacity.

With the extended address range, the memory management in Red Hat Enterprise Linux 8 adds support for 5-level page table implementation, to be able to handle the expanded address range.

(BZ#1485525)

kernel-signing-ca.cer is moved to **kernel-core** in RHEL 8

In all versions of Red Hat Enterprise Linux 7, the **kernel-signing-ca.cer** public key was located in the **kernel-doc** package. However, in Red Hat Enterprise Linux 8, **kernel-signing-ca.cer** has been relocated to the **kernel-core** package for every architecture.

(BZ#1638465)

Spectre V2 mitigation default changed from IBRS to Retpolines

The default mitigation for the Spectre V2 vulnerability (CVE-2017-5715) for systems with the 6th Generation Intel Core Processors and its close derivatives [1] has changed from Indirect Branch Restricted Speculation (IBRS) to Retpolines in Red Hat Enterprise Linux 8. Red Hat has implemented this change as a result of Intel's recommendations to align with the defaults used in the Linux community and to restore lost performance. However, note that using Retpolines in some cases may not fully mitigate Spectre V2. Intel's Retpoline document [2] describes any cases of exposure. This document also states that the risk of an attack is low.

For use cases where complete Spectre V2 mitigation is desired, a user can select IBRS through the kernel boot line by adding the **spectre_v2=ibrs** flag.

If one or more kernel modules were not built with the Retpoline support, the **/sys/devices/system/cpu/vulnerabilities/spectre_v2** file will indicate vulnerability and the

/var/log/messages file will identify the offending modules. See How to determine which modules are responsible for spectre_v2 returning "Vulnerable: Retpoline with unsafe module(s)"? for further information.

[1] "6th generation Intel Core Processors and its close derivatives" are what the Intel's Retpolines document refers to as "Skylake-generation".

[2] Retpoline: A Branch Target Injection Mitigation - White Paper

(BZ#1651806)

Intel® Omni-Path Architecture (OPA) Host Software

Intel Omni-Path Architecture (OPA) host software is fully supported in Red Hat Enterprise Linux 8.

Intel OPA provides Host Fabric Interface (HFI) hardware with initialization and setup for high performance data transfers (high bandwidth, high message rate, low latency) between compute and I/O nodes in a clustered environment.

For instructions on installing Intel Omni-Path Architecture documentation, see: https://www.intel.com/content/dam/support/us/en/documents/network-and-i-o/fabric-products/Intel_OP_Software_RHEL_8_RN_K51383.pdf

(BZ#1683712)

NUMA supports more nodes in RHEL 8

With this update, the Non-Uniform Memory Access (NUMA) node count has been increased from 4 NUMA nodes to 8 NUMA nodes in Red Hat Enterprise Linux 8 on systems with the 64-bit ARM architecture.

(BZ#1550498)

IOMMU passthrough is now enabled by default in RHEL 8

The Input/Output Memory Management Unit (IOMMU) passthrough has been enabled by default. This provides improved performance for AMD systems because Direct Memory Access (DMA) remapping is disabled for the host. This update brings consistency with Intel systems where DMA remapping is also disabled by default. Users may disable such behavior (and enable DMA remapping) by specifying either **iommu.passthrough=off** or **iommu=nopt** parameters on the kernel command line, including the hypervisor.

(BZ#1658391)

RHEL8 kernel now supports 5-level page tables

Red Hat Enterprise Linux kernel now fully supports future Intel processors with up to 5 levels of page tables. This enables the processors to support up to 4PB of physical memory and 128PB of virtual address space. Applications that utilize large amounts of memory can now use as much memory as possible as provided by the system without the constraints of 4-level page tables.

(BZ#1623590)

RHEL8 kernel supports enhanced IBRS for future Intel CPUs

Red Hat Enterprise Linux kernel now supports the use of enhanced Indirect Branch Restricted Speculation (IBRS) capability to mitigate the Spectre V2 vulnerability. When enabled, IBRS will perform better than Retpolines (default) to mitigate Spectre V2 and will not interfere with Intel Control-flow

Enforcement technology. As a result, the performance penalty of enabling the mitigation for Spectre V2 will be smaller on future Intel CPUs.

(BZ#1614144)

bpftool for inspection and manipulation of eBPF-based programs and maps added

The **bpftool** utility that serves for inspection and simple manipulation of programs and maps based on extended Berkeley Packet Filtering (eBPF) has been added into the Linux kernel. **bpftool** is a part of the kernel source tree, and is provided by the **bpftool** package, which is included as a sub-package of the **kernel** package.

(BZ#1559607)

The kernel-rt sources have been updated

The **kernel-rt** sources have been updated to use the latest RHEL kernel source tree. The latest kernel source tree is now using the upstream v4.18 realtime patch set, which provides a number of bug fixes and enhancements over the previous version.

(BZ#1592977)

5.1.4. Software management

YUM performance improvement and support for modular content

On Red Hat Enterprise Linux 8, installing software is ensured by the new version of the **YUM** tool, which is based on the **DNF** technology (**YUM v4**).

YUM v4 has the following advantages over the previous **YUM v3** used on RHEL 7:

- Increased performance
- Support for modular content
- Well-designed stable API for integration with tooling

For detailed information about differences between the new **YUM v4** tool and the previous version **YUM v3** from RHEL 7, see Changes in DNF CLI compared to YUM .

YUM v4 is compatible with **YUM v3** when using from the command line, editing or creating configuration files.

For installing software, you can use the **yum** command and its particular options in the same way as on RHEL 7.

Selected yum plug-ins and utilities have been ported to the new DNF back end, and can be installed under the same names as in RHEL 7. They also provide compatibility symlinks, so the binaries, configuration files and directories can be found in usual locations.

Note that the legacy Python API provided by **YUM v3** is no longer available. Users are advised to migrate their plug-ins and scripts to the new API provided by **YUM v4** (DNF Python API), which is stable and fully supported. The DNF Python API is available at DNF API Reference.

The Libdnf and Hawkey APIs (both C and Python) are unstable, and will likely change during Red Hat Enterprise Linux 8 life cycle.

For more details on changes of **YUM** packages and tools availability, see Considerations in adopting RHEL 8.

Some of the **YUM v3** features may behave differently in **YUM v4**. If any such change negatively impacts your workflows, please open a case with Red Hat Support, as described in How do I open and manage a support case on the Customer Portal?

(BZ#1581198)

Notable RPM features in RHEL 8

Red Hat Enterprise Linux 8 is distributed with RPM 4.14. This version introduces many enhancements over RPM 4.11, which is available in RHEL 7. The most notable features include:

- The **debuginfo** packages can be installed in parallel
- Support for weak dependencies
- Support for rich or boolean dependencies
- Support for packaging files above 4 GB in size
- Support for file triggers

Also, the most notable changes include:

- Stricter spec-parser
- Simplified signature checking the output in non-verbose mode
- Additions and deprecation in macros

(BZ#1581990)

RPM now validates the entire package contents before starting an installation

On Red Hat Enterprise Linux 7, the **RPM** utility verified payload contents of individual files while unpacking. However, this is insufficient for multiple reasons:

- If the payload is damaged, it is only noticed after executing script actions, which are irreversible.
- If the payload is damaged, upgrade of a package aborts after replacing some files of the previous version, which breaks a working installation.
- The hashes on individual files are performed on uncompressed data, which makes **RPM** vulnerable to decompressor vulnerabilities.

On Red Hat Enterprise Linux 8, the entire package is validated prior to the installation in a separate step, using the best available hash.

Packages built on Red Hat Enterprise Linux 8 use a new **SHA-256** hash on the compressed payload. On signed packages, the payload hash is additionally protected by the signature, and thus cannot be altered without breaking a signature and other hashes on the package header. Older packages use the **MD5** hash of the header and payload unless it is disabled by configuration.

The **%_pkgverify_level** macro can be used to additionally enable enforcing signature verification before installation or disable the payload verification completely. In addition, the **%_pkgverify_flags** macro can be used to limit which hashes and signatures are allowed. For example, it is possible to disable the use of the weak **MD5** hash at the cost of compatibility with older packages.

(JIRA:RHELPLAN-10596)

5.1.5. Infrastructure services

Notable changes in the recommended Tuned profile in RHEL 8

With this update, the recommended Tuned profile (reported by the **tuned-adm recommend** command) is now selected based on the following rules - the first rule that matches takes effect:

- If the **syspurpose** role (reported by the **syspurpose show** command) contains **atomic**, and at the same time:
  - if Tuned is running on bare metal, the **atomic-host** profile is selected
  - if Tuned is running in a virtual machine, the **atomic-guest** profile is selected
- If Tuned is running in a virtual machine, the **virtual-guest** profile is selected
- If the **syspurpose** role contains **desktop** or **workstation** and the chassis type (reported by **dmidecode**) is **Notebook**, **Laptop**, or **Portable**, then the **balanced** profile is selected
- If none of the above rules matches, the **throughput-performance** profile is selected

(BZ#1565598)

Files produced by named can be written in the working directory

Previously, the **named** daemon stored some data in the working directory, which has been read-only in Red Hat Enterprise Linux. With this update, paths have been changed for selected files into subdirectories, where writing is allowed. Now, default directory Unix and SELinux permissions allow writing into the directory. Files distributed inside the directory are still read-only to **named**.

(BZ#1588592)

Geolite Databases have been replaced by Geolite2 Databases

Geolite Databases that were present in Red Hat Enterprise Linux 7 were replaced by Geolite2 Databases on Red Hat Enterprise Linux 8.

Geolite Databases were provided by the **GeoIP** package. This package together with the legacy database is no longer supported in the upstream.

Geolite2 Databases are provided by multiple packages. The **libmaxminddb** package includes the library and the **mmdblookup** command line tool, which enables manual searching of addresses. The **geoipupdate** binary from the legacy **GeoIP** package is now provided by the **geoipupdate** package, and is capable of downloading both legacy databases and the new Geolite2 databases.

(JIRA:RHELPLAN-6746)

CUPS logs are handled by journald

In RHEL 8, the CUPS logs are no longer stored in specific files within the **/var/log/cups** directory, which was used in RHEL 7. In RHEL 8, all types of CUPS logs are centrally-logged in the systemd **journald** daemon together with logs from other programs. To access the CUPS logs, use the **journalctl -u cups** command. For more information, see Accessing the CUPS logs in the systemd journal .

(JIRA:RHELPLAN-12764)

Notable BIND features in RHEL 8

RHEL 8 includes BIND (Berkeley Internet Name Domain) in version 9.11. This version of the DNS server introduces multiple new features and feature changes compared to version 9.10.

New features:

- A new method of provisioning secondary servers called **Catalog Zones** has been added.
- Domain Name System Cookies are now sent by the **named** service and the **dig** utility.
- The **Response Rate Limiting** feature can now help with mitigation of DNS amplification attacks.
- Performance of response-policy zone (RPZ) has been improved.
- A new zone file format called **map** has been added. Zone data stored in this format can be mapped directly into memory, which enables zones to load significantly faster.
- A new tool called **delv** (domain entity lookup and validation) has been added, with dig-like semantics for looking up DNS data and performing internal DNS Security Extensions (DNSSEC) validation.
- A new **mdig** command is now available. This command is a version of the `dig` command that sends multiple pipelined queries and then waits for responses, instead of sending one query and waiting for the response before sending the next query.
- A new **prefetch** option, which improves the recursive resolver performance, has been added.
- A new **in-view** zone option, which allows zone data to be shared between views, has been added. When this option is used, multiple views can serve the same zones authoritatively without storing multiple copies in memory.
- A new **max-zone-ttl** option, which enforces maximum TTLs for zones, has been added. When a zone containing a higher TTL is loaded, the load fails. Dynamic DNS (DDNS) updates with higher TTLs are accepted but the TTL is truncated.
- New quotas have been added to limit queries that are sent by recursive resolvers to authoritative servers experiencing denial-of-service attacks.
- The **nslookup** utility now looks up both IPv6 and IPv4 addresses by default.
- The **named** service now checks whether other name server processes are running before starting up.
- When loading a signed zone, **named** now checks whether a Resource Record Signature's (RSIG) inception time is in the future, and if so, it regenerates the RRSIG immediately.
- Zone transfers now use smaller message sizes to improve message compression, which reduces network usage.

Feature changes:

- The version **3 XML** schema for the statistics channel, including new statistics and a flattened XML tree for faster parsing, is provided by the HTTP interface. The legacy version **2 XML** schema is no longer supported.
- The **named** service now listens on both IPv6 and IPv4 interfaces by default.

- The **named** service no longer supports GeoIP. Access control lists (ACLs) defined by presumed location of query sender are unavailable.

(JIRA:RHELPLAN-1820)

5.1.6. Shells and command-line tools

The nobody user replaces nfsnobody

In Red Hat Enterprise Linux 7, there was:

- the **nobody** user and group pair with the ID of 99, and
- the **nfsnobody** user and group pair with the ID of 65534, which is the default kernel overflow ID, too.

Both of these have been merged into the **nobody** user and group pair, which uses the 65534 ID in Red Hat Enterprise Linux 8. New installations no longer create the **nfsnobody** pair.

This change reduces the confusion about files that are owned by **nobody** but have nothing to do with NFS.

(BZ#1591969)

Version control systems in RHEL 8

RHEL 8 provides the following version control systems:

- **Git 2.18**, a distributed revision control system with a decentralized architecture.
- **Mercurial 4.8**, a lightweight distributed version control system, designed for efficient handling of large projects.
- **Subversion 1.10**, a centralized version control system.

Note that the Concurrent Versions System (CVS) and Revision Control System (RCS), available in RHEL 7, are not distributed with RHEL 8.

(BZ#1693775)

Notable changes in Subversion 1.10

Subversion 1.10 introduces a number of new features since the version 1.7 distributed in RHEL 7, as well as the following compatibility changes:

- Due to incompatibilities in the **Subversion** libraries used for supporting language bindings, **Python 3** bindings for **Subversion 1.10** are unavailable. As a consequence, applications that require **Python** bindings for **Subversion** are unsupported.
- Repositories based on **Berkeley DB** are no longer supported. Before migrating, back up repositories created with **Subversion 1.7** by using the **svnadmin dump** command. After installing RHEL 8, restore the repositories using the **svnadmin load** command.
- Existing working copies checked out by the **Subversion 1.7** client in RHEL 7 must be upgraded to the new format before they can be used from **Subversion 1.10**. After installing RHEL 8, run the **svn upgrade** command in each working copy.
- Smartcard authentication for accessing repositories using **https://** is no longer supported.

(BZ#1571415)

Notable changes in dstat

RHEL 8 is distributed with a new version of the **dstat** tool. This tool is now a part of the Performance Co-Pilot (PCP) toolkit. The **/usr/bin/dstat** file and the **dstat** package name is now provided by the **pcp-system-tools** package.

The new version of **dstat** introduces the following enhancements over **dstat** available in RHEL 7:

- **python3** support
- Historical analysis
- Remote host analysis
- Configuration file plugins
- New performance metrics

(BZ#1684947)

5.1.7. Dynamic programming languages, web and database servers

Python 3 is the default Python implementation in RHEL 8

Red Hat Enterprise Linux 8 is distributed with **Python 3.6**. The package might not be installed by default. To install **Python 3.6**, use the **yum install python3** command.

Python 2.7 is available in the **python2** package. However, **Python 2** will have a shorter life cycle and its aim is to facilitate a smoother transition to **Python 3** for customers.

Neither the default **python** package nor the unversioned **/usr/bin/python** executable is distributed with RHEL 8. Customers are advised to use **python3** or **python2** directly. Alternatively, administrators can configure the unversioned **python** command using the **alternatives** command.

For more information, see Introduction to Python.

(BZ#1580387)

Python scripts must specify major version in interpreter directives at RPM build time

In RHEL 8, executable Python scripts are expected to use interpreter directives (hashbangs) specifying explicitly at least the major Python version.

The **/usr/lib/rpm/redhat/brp-mangle-shebangs** buildroot policy (BRP) script is run automatically when building any RPM package. This script attempts to correct interpreter directives in all executable files. When the script encounters ambiguous Python interpreter directives that do not specify the major version of Python, it generates errors and the RPM build fails. Examples of such ambiguous interpreter directives include:

- **#! /usr/bin/python**
- **#! /usr/bin/env python**

To modify interpreter directives in the Python scripts causing these build errors at RPM build time, use the **pathfix.py** script from the **platform-python-devel** package:

```
pathfix.py -pn -i %{__python3} PATH ...
```

Multiple *PATH*s can be specified. If a *PATH* is a directory, **pathfix.py** recursively scans for any Python scripts matching the pattern **^[a-zA-Z0-9_]+\.py$**, not only those with an ambiguous hashbang. Add the command for running **pathfix.py** to the **%prep** section or at the end of the **%install** section.

For more information, see Handling interpreter directives in Python scripts .

(BZ#1583620)

Notable changes in PHP

Red Hat Enterprise Linux 8 is distributed with **PHP 7.2**. This version introduces the following major changes over **PHP 5.4**, which is available in RHEL 7:

- **PHP** uses FastCGI Process Manager (FPM) by default (safe for use with a threaded **httpd**)
- The **php_value** and **php-flag** variables should no longer be used in the **httpd** configuration files; they should be set in pool configuration instead: **/etc/php-fpm.d/*.conf**
- **PHP** script errors and warnings are logged to the **/var/log/php-fpm/www-error.log** file instead of **/var/log/httpd/error.log**
- When changing the PHP **max_execution_time** configuration variable, the **httpd ProxyTimeout** setting should be increased to match
- The user running **PHP** scripts is now configured in the FPM pool configuration (the **/etc/php-fpm.d/www.conf** file; the **apache** user is the default)
- The **php-fpm** service needs to be restarted after a configuration change or after a new extension is installed
- The **zip** extension has been moved from the **php-common** package to a separate package, **php-pecl-zip**

The following extensions have been removed:

- **aspell**
- **mysql** (note that the **mysqli** and **pdo_mysql** extensions are still available, provided by **php-mysqlnd** package)
- **memcache**

(BZ#1580430, BZ#1691688)

Notable changes in Ruby

RHEL 8 provides **Ruby 2.5**, which introduces numerous new features and enhancements over **Ruby 2.0.0** available in RHEL 7. Notable changes include:

- Incremental garbage collector has been added.
- The **Refinements** syntax has been added.
- Symbols are now garbage collected.
- The **$SAFE=2** and **$SAFE=3** safe levels are now obsolete.

- The **Fixnum** and **Bignum** classes have been unified into the **Integer** class.
- Performance has been improved by optimizing the **Hash** class, improved access to instance variables, and the **Mutex** class being smaller and faster.
- Certain old APIs have been deprecated.
- Bundled libraries, such as **RubyGems**, **Rake**, **RDoc**, **Psych**, **Minitest**, and **test-unit**, have been updated.
- Other libraries, such as **mathn**, **DL**, **ext/tk**, and **XMLRPC**, which were previously distributed with **Ruby**, are deprecated or no longer included.
- The **SemVer** versioning scheme is now used for **Ruby** versioning.

(BZ#1648843)

Notable changes in Perl

Perl 5.26, distributed with RHEL 8, introduces the following changes over the version available in RHEL 7:

- **Unicode 9.0** is now supported.
- New **op-entry**, **loading-file**, and **loaded-file SystemTap** probes are provided.
- Copy-on-write mechanism is used when assigning scalars for improved performance.
- The **IO::Socket::IP** module for handling IPv4 and IPv6 sockets transparently has been added.
- The **Config::Perl::V** module to access **perl -V** data in a structured way has been added.
- A new **perl-App-cpanminus** package has been added, which contains the **cpanm** utility for getting, extracting, building, and installing modules from the Comprehensive Perl Archive Network (CPAN) repository.
- The current directory **.** has been removed from the **@INC** module search path for security reasons.
- The **do** statement now returns a deprecation warning when it fails to load a file because of the behavioral change described above.
- The **do subroutine(LIST)** call is no longer supported and results in a syntax error.
- Hashes are randomized by default now. The order in which keys and values are returned from a hash changes on each **perl** run. To disable the randomization, set the **PERL_PERTURB_KEYS** environment variable to **0**.
- Unescaped literal **{** characters in regular expression patterns are no longer permissible.
- Lexical scope support for the **$_** variable has been removed.
- Using the **defined** operator on an array or a hash results in a fatal error.
- Importing functions from the **UNIVERSAL** module results in a fatal error.
- The **find2perl**, **s2p**, **a2p**, **c2ph**, and **pstruct** tools have been removed.

- The **${^ENCODING}** facility has been removed. The **encoding** pragma's default mode is no longer supported. To write source code in other encoding than **UTF-8**, use the encoding's **Filter** option.

- The **perl** packaging is now aligned with upstream. The **perl** package installs also core modules, while the **/usr/bin/perl** interpreter is provided by the **perl-interpreter** package. In previous releases, the **perl** package included just a minimal interpreter, whereas the **perl-core** package included both the interpreter and the core modules.

- The **IO::Socket::SSL** Perl module no longer loads a certificate authority certificate from the **./certs/my-ca.pem** file or the **./ca** directory, a server private key from the **./certs/server-key.pem** file, a server certificate from the **./certs/server-cert.pem** file, a client private key from the **./certs/client-key.pem** file, and a client certificate from the **./certs/client-cert.pem** file. Specify the paths to the files explicitly instead.

(BZ#1511131)

Node.js new in RHEL

Node.js, a software development platform for building fast and scalable network applications in the JavaScript programming language, is provided for the first time in RHEL. It was previously available only as a Software Collection. RHEL 8 provides **Node.js 10**.

(BZ#1622118)

Notable changes in SWIG

RHEL 8 includes the Simplified Wrapper and Interface Generator (SWIG) version 3.0, which provides numerous new features, enhancements, and bug fixes over the version 2.0 distributed in RHEL 7. Most notably, support for the C++11 standard has been implemented. **SWIG** now supports also **Go 1.6**, **PHP 7**, **Octave 4.2**, and **Python 3.5**.

(BZ#1660051)

Notable changes in Apache httpd

RHEL 8 is distributed with the Apache HTTP Server 2.4.37. This version introduces the following changes over **httpd** available in RHEL 7:

- HTTP/2 support is now provided by the **mod_http2** package, which is a part of the **httpd** module.

- Automated TLS certificate provisioning and renewal using the Automatic Certificate Management Environment (ACME) protocol is now supported with the **mod_md** package (for use with certificate providers such as **Let's Encrypt**)

- The Apache HTTP Server now supports loading TLS certificates and private keys from hardware security tokens directly from **PKCS#11** modules. As a result, a **mod_ssl** configuration can now use **PKCS#11** URLs to identify the TLS private key, and, optionally, the TLS certificate in the **SSLCertificateKeyFile** and **SSLCertificateFile** directives.

- The multi-processing module (MPM) configured by default with the Apache HTTP Server has changed from a multi-process, forked model (known as **prefork**) to a high-performance multi-threaded model, **event**. Any third-party modules that are not thread-safe need to be replaced or removed. To change the configured MPM, edit the **/etc/httpd/conf.modules.d/00-mpm.conf** file. See the **httpd.conf(5)** man page for more information.

For more information about changes in **httpd** and its usage, see Setting up the Apache HTTP web server.

(BZ#1632754, BZ#1527084, BZ#1581178)

The nginx web server new in RHEL

RHEL 8 introduces **nginx 1.14**, a web and proxy server supporting HTTP and other protocols, with a focus on high concurrency, performance, and low memory usage. **nginx** was previously available only as a Software Collection.

The **nginx** web server now supports loading TLS private keys from hardware security tokens directly from **PKCS#11** modules. As a result, an **nginx** configuration can use **PKCS#11** URLs to identify the TLS private key in the **ssl_certificate_key** directive.

(BZ#1545526)

Database servers in RHEL 8

RHEL 8 provides the following database servers:

- **MySQL 8.0**, a multi-user, multi-threaded SQL database server. It consists of the **MySQL** server daemon, **mysqld**, and many client programs.
- **MariaDB 10.3**, a multi-user, multi-threaded SQL database server. For all practical purposes, **MariaDB** is binary-compatible with **MySQL**.
- **PostgreSQL 10** and **PostgreSQL 9.6**, an advanced object-relational database management system (DBMS).
- **Redis 5**, an advanced key-value store. It is often referred to as a data structure server because keys can contain strings, hashes, lists, sets, and sorted sets. **Redis** is provided for the first time in RHEL.

Note that the NoSQL **MongoDB** database server is not included in RHEL 8.0 because it uses the Server Side Public License (SSPL).

(BZ#1647908)

Notable changes in MySQL 8.0

RHEL 8 is distributed with **MySQL 8.0**, which provides, for example, the following enhancements:

- **MySQL** now incorporates a transactional data dictionary, which stores information about database objects.
- **MySQL** now supports roles, which are collections of privileges.
- The default character set has been changed from **latin1** to **utf8mb4**.
- Support for common table expressions, both nonrecursive and recursive, has been added.
- **MySQL** now supports window functions, which perform a calculation for each row from a query, using related rows.
- **InnoDB** now supports the **NOWAIT** and **SKIP LOCKED** options with locking read statements.
- GIS-related functions have been improved.

- JSON functionality has been enhanced.
- The new **mariadb-connector-c** packages provide a common client library for **MySQL** and **MariaDB**. This library is usable with any version of the **MySQL** and **MariaDB** database servers. As a result, the user is able to connect one build of an application to any of the **MySQL** and **MariaDB** servers distributed with RHEL 8.

In addition, the **MySQL 8.0** server distributed with RHEL 8 is configured to use **mysql_native_password** as the default authentication plug-in because client tools and libraries in RHEL 8 are incompatible with the **caching_sha2_password** method, which is used by default in the upstream **MySQL 8.0** version.

To change the default authentication plug-in to **caching_sha2_password**, edit the **/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** file as follows:

```
[mysqld]
default_authentication_plugin=caching_sha2_password
```

See also Using MySQL.

(BZ#1649891, BZ#1519450, BZ#1631400)

Notable changes in MariaDB 10.3

MariaDB 10.3 provides numerous new features over the version 5.5 distributed in RHEL 7, such as:

- Common table expressions
- System-versioned tables
- **FOR** loops
- Invisible columns
- Sequences
- Instant **ADD COLUMN** for **InnoDB**
- Storage-engine independent column compression
- Parallel replication
- Multi-source replication

In addition, the new **mariadb-connector-c** packages provide a common client library for **MySQL** and **MariaDB**. This library is usable with any version of the **MySQL** and **MariaDB** database servers. As a result, the user is able to connect one build of an application to any of the **MySQL** and **MariaDB** servers distributed with RHEL 8.

Other notable changes include:

- **MariaDB Galera Cluster**, a synchronous multi-master cluster, is now a standard part of **MariaDB**.
- **InnoDB** is used as the default storage engine instead of **XtraDB**.
- The mariadb-bench subpackage has been removed.

- The default allowed level of the plug-in maturity has been changed to one level less than the server maturity. As a result, plug-ins with a lower maturity level that were previously working, will no longer load.

See also Using MariaDB.

(BZ#1637034, BZ#1519450, BZ#1688374)

Notable changes in PostgreSQL

RHEL 8.0 provides two versions of the **PostgreSQL** database server, distributed in two streams of the **postgresql** module: **PostgreSQL 10** (the default stream) and **PostgreSQL 9.6**. RHEL 7 includes **PostgreSQL** version 9.2.

Notable changes in **PostgreSQL 9.6** are, for example:

- Parallel execution of the sequential operations: **scan**, **join**, and **aggregate**
- Enhancements to synchronous replication
- Improved full-text search enabling users to search for phrases
- The **postgres_fdw** data federation driver now supports remote **join**, **sort**, **UPDATE**, and **DELETE** operations
- Substantial performance improvements, especially regarding scalability on multi-CPU-socket servers

Major enhancements in **PostgreSQL 10** include:

- Logical replication using the **publish** and **subscribe** keywords
- Stronger password authentication based on the **SCRAM-SHA-256** mechanism
- Declarative table partitioning
- Improved query parallelism
- Significant general performance improvements
- Improved monitoring and control

See also Using PostgreSQL.

(BZ#1660041)

Notable changes in Squid

RHEL 8.0 is distributed with **Squid 4.4**, a high-performance proxy caching server for web clients, supporting FTP, Gopher, and HTTP data objects. This release provides numerous new features, enhancements, and bug fixes over the version 3.5 available in RHEL 7.

Notable changes include:

- Configurable helper queue size
- Changes to helper concurrency channels

- Changes to the helper binary
- Secure Internet Content Adaptation Protocol (ICAP)
- Improved support for Symmetric Multi Processing (SMP)
- Improved process management
- Removed support for SSL
- Removed Edge Side Includes (ESI) custom parser
- Multiple configuration changes

(BZ#1656871)

Varnish Cache new in RHEL

Varnish Cache, a high-performance HTTP reverse proxy, is provided for the first time in RHEL. It was previously available only as a Software Collection. **Varnish Cache** stores files or fragments of files in memory that are used to reduce the response time and network bandwidth consumption on future equivalent requests. RHEL 8.0 is distributed with **Varnish Cache 6.0**.

(BZ#1633338)

5.1.8. Desktop

GNOME Shell, version 3.28 in RHEL 8

GNOME Shell, version 3.28 is available in Red Hat Enterprise Linux (RHEL) 8. Notable enhancements include:

- New GNOME Boxes features
- New on-screen keyboard
- Extended devices support, most significantly integration for the Thunderbolt 3 interface
- Improvements for GNOME Software, dconf-editor and GNOME Terminal

(BZ#1649404)

Wayland is the default display server

With Red Hat Enterprise Linux 8, the GNOME session and the GNOME Display Manager (GDM) use **Wayland** as their default display server instead of the **X.org** server, which was used with the previous major version of RHEL.

Wayland provides multiple advantages and improvements over **X.org**. Most notably:

- Stronger security model
- Improved multi-monitor handling
- Improved user interface (UI) scaling
- The desktop can control window handling directly.

Note that the following features are currently unavailable or do not work as expected:

- Multi-GPU setups are not supported under **Wayland**.
- The **NVIDIA** binary driver does not work under **Wayland**.
- The **xrandr** utility does not work under **Wayland** due to its different approach to handling, resolutions, rotations, and layout. Note that other **X.org** utilities for manipulating the screen do not work under **Wayland**, either.
- Screen recording, remote desktop, and accessibility do not always work correctly under **Wayland**.
- No clipboard manager is available.
- **Wayland** ignores keyboard grabs issued by X11 applications, such as virtual machines viewers.
- **Wayland** inside guest virtual machines (VMs) has stability and performance problems, so it is recommended to use the X11 session for virtual environments.

If you upgrade to RHEL 8 from a RHEL 7 system where you used the **X.org** GNOME session, your system continues to use **X.org**. The system also automatically falls back to **X.org** when the following graphics drivers are in use:

- The NVIDIA binary driver
- The **cirrus** driver
- The **mga** driver
- The **aspeed** driver

You can disable the use of **Wayland** manually:

- To disable **Wayland** in **GDM**, set the **WaylandEnable=false** option in the **/etc/gdm/custom.conf** file.
- To disable **Wayland** in the GNOME session, select the legacy X11 option by using the cogwheel menu on the login screen after entering your login name.

For more details on **Wayland**, see https://wayland.freedesktop.org/.

(BZ#1589678)

Locating RPM packages that are in repositories not enabled by default

Additional repositories for desktop are not enabled by default. The disablement is indicated by the **enabled=0** line in the corresponding **.repo** file. If you attempt to install a package from such repository using PackageKit, PackageKit shows an error message announcing that the application is not available. To make the package available, replace previously used **enabled=0** line in the respective **.repo** file with **enabled=1**.

(JIRA:RHELPLAN-2878)

GNOME Sofware for package management

The **gnome-packagekit** package that provided a collection of tools for package management in graphical environment on Red Hat Enterprise Linux 7 is no longer available. On Red Hat Enterprise Linux

8, similar functionality is provided by the **GNOME Software** utility, which enables you to install and update applications and gnome-shell extensions. **GNOME Software** is distributed in the **gnome-software** package.

(JIRA:RHELPLAN-3001)

Fractional scaling available for GNOME Shell on Wayland

On a **GNOME Shell on Wayland** session, the fractional scaling feature is available. The feature makes it possible to scale the GUI by fractions, which improves the appearance of scaled GUI on certain displays.

Note that the feature is currently considered experimental and is, therefore, disabled by default.

To enable fractional scaling, run the following command:

```
# gsettings set org.gnome.mutter experimental-features "['scale-monitor-framebuffer']"
```

(BZ#1668883)

5.1.9. Hardware enablement

Firmware updates using **fwupd** are available

RHEL 8 supports firmware updates, such as UEFI capsule, Device Firmware Upgrade (DFU), and others, using the **fwupd** daemon. The daemon allows session software to update device firmware on a local machine automatically.

To view and apply updates, you can use:

- A GUI software manager, such as GNOME Software
- The **fwupdmgr** command-line tool

The metadata files are automatically downloaded from the Linux Vendor Firmware Service (LVFS) secure portal, and submitted into **fwupd** over D-Bus. The updates that need to be applied are downloaded displaying user notifications and update details. The user must explicitly agree with the firmware update action before the update is performed.

Note that the access to LVFS is disabled by default.

To enable the access to LVFS, either click the slider in the **sources** dialog in GNOME Software, or run the **fwupdmgr enable-remote lvfs** command. If you use **fwupdmgr** to get the updates list, you will be asked if you want to enable LVFS.

With access to LVFS, you will get firmware updates directly from the hardware vendor. Note that such updates have not been verified by Red Hat QA.

(BZ#1504934)

Memory Mode for Optane DC Persistent Memory technology is fully supported

Intel Optane DC Persistent Memory storage devices provide data center-class persistent memory technology, which can significantly increase transaction throughput.

To use the Memory Mode technology, your system does not require any special drivers or specific certification. Memory Mode is transparent to the operating system.

(BZ#1718422)

5.1.10. Identity Management

New password syntax checks in Directory Server

This enhancement adds new password syntax checks to Directory Server. Administrators can now, for example, enable dictionary checks, allow or deny using character sequences and palindromes. As a result, if enabled, the password policy syntax check in Directory Server enforces more secure passwords.

(BZ#1334254)

Directory Server now provides improved internal operations logging support

Several operations in Directory Server, initiated by the server and clients, cause additional operations in the background. Previously, the server only logged for internal operations the **Internal** connection keyword, and the operation ID was always set to **-1**. With this enhancement, Directory Server logs the real connection and operation ID. You can now trace the internal operation to the server or client operation that caused this operation.

(BZ#1358706)

The **tomcatjss** library supports OCSP checking using the responder from the AIA extension

With this enhancement, the **tomcatjss** library supports Online Certificate Status Protocol (OCSP) checking using the responder from the Authority Information Access (AIA) extension of a certificate. As a result, administrators of Red Hat Certificate System can now configure OCSP checking that uses the URL from the AIA extension.

(BZ#1636564)

The **pki subsystem-cert-find** and **pki subsystem-cert-show** commands now show the serial number of certificates

With this enhancement, the **pki subsystem-cert-find** and **pki subsystem-cert-show** commands in Certificate System show the serial number of certificates in their output. The serial number is an important piece of information and often required by multiple other commands. As a result, identifying the serial number of a certificate is now easier.

(BZ#1566360)

The **pki user** and **pki group** commands have been deprecated in Certificate System

With this update, the new **pki *<subsystem>*-user** and **pki *<subsystem>*-group** commands replace the **pki user** and **pki group** commands in Certificate System. The replaced commands still works, but they display a message that the command is deprecated and refer to the new commands.

(BZ#1394069)

Certificate System now supports offline renewal of system certificates

With this enhancement, administrators can use the offline renewal feature to renew system certificates configured in Certificate System. When a system certificate expires, Certificate System fails to start. As a result of the enhancement, administrators no longer need workarounds to replace an expired system certificate.

(BZ#1669257)

Certificate System can now create CSRs with SKI extension for external CA signing

With this enhancement, Certificate System supports creating a certificate signing request (CSR) with the Subject Key Identifier (SKI) extension for external certificate authority (CA) signing. Certain CAs require this extension either with a particular value or derived from the CA public key. As a result, administrators can now use the **pki_req_ski** parameter in the configuration file passed to the **pkispawn** utility to create a CSR with SKI extension.

(BZ#1656856)

SSSD no longer uses the fallback_homedir value from the [nss] section as fallback for AD domains

Prior to RHEL 7.7, the SSSD **fallback_homedir** parameter in an Active Directory (AD) provider had no default value. If **fallback_homedir** was not set, SSSD used instead the value from the same parameter from the **[nss]** section in the **/etc/sssd/sssd.conf** file. To increase security, SSSD in RHEL 7.7 introduced a default value for **fallback_homedir**. As a consequence, SSSD no longer falls back to the value set in the **[nss]** section. If you want to use a different value than the default for the **fallback_homedir** parameter in an AD domain, you must manually set it in the domain's section.

(BZ#1652719)

SSSD now allows you to select one of the multiple Smartcard authentication devices

By default, the System Security Services Daemon (SSSD) tries to detect a device for Smartcard authentication automatically. If there are multiple devices connected, SSSD selects the first one it detects. Consequently, you cannot select a particular device, which sometimes leads to failures.

With this update, you can configure a new **p11_uri** option for the **[pam]** section of the **sssd.conf** configuration file. This option enables you to define which device is used for Smartcard authentication.

For example, to select a reader with the slot id **2** detected by the OpenSC PKCS#11 module, add:

```
p11_uri = library-description=OpenSC%20smartcard%20framework;slot-id=2
```

to the **[pam]** section of **sssd.conf**.

For details, see the **man sssd.conf** page.

(BZ#1620123)

Local users are cached by SSSD and served through the nss_sss module

In RHEL 8, the System Security Services Daemon (SSSD) serves users and groups from the **/etc/passwd** and **/etc/groups** files by default. The **sss** nsswitch module precedes files in the **/etc/nsswitch.conf** file.

The advantage of serving local users through SSSD is that the **nss_sss** module has a fast **memory-mapped cache** that speeds up Name Service Switch (NSS) lookups compared to accessing the disk and opening the files on each NSS request. Previously, the Name service cache daemon (**nscd**) helped accelerate the process of accessing the disk. However, using **nscd** in parallel with SSSD is cumbersome, as both SSSD and **nscd** use their own independent caching. Consequently, using **nscd** in setups where SSSD is also serving users from a remote domain, for example LDAP or Active Directory, can cause unpredictable behavior.

With this update, the resolution of local users and groups is faster in RHEL 8. Note that the **root** user is never handled by SSSD, therefore **root** resolution cannot be impacted by a potential bug in SSSD. Note also that if SSSD is not running, the **nss_sss** module handles the situation gracefully by falling back to

nss_files to avoid problems. You do not have to configure SSSD in any way, the files domain is added automatically.

(JIRA:RHELPLAN-10439)

KCM replaces KEYRING as the default credential cache storage

In RHEL 8, the default credential cache storage is the Kerberos Credential Manager (KCM) which is backed by the **sssd-kcm** deamon. KCM overcomes the limitations of the previously used KEYRING, such as its being difficult to use in containerized environments because it is not namespaced, and to view and manage quotas.

With this update, RHEL 8 contains a credential cache that is better suited for containerized environments and that provides a basis for building more features in future releases.

(JIRA:RHELPLAN-10440)

Active Directory users can now administer Identity Management

With this update, RHEL 8 allows adding a user ID override for an Active Directory (AD) user as a member of an Identity Management (IdM) group. An ID override is a record describing what a specific AD user or group properties should look like within a specific ID view, in this case the Default Trust View. As a consequence of the update, the IdM LDAP server is able to apply access control rules for the IdM group to the AD user.

AD users are now able to use the self service features of IdM UI, for example to upload their SSH keys, or change their personal data. An AD administrator is able to fully administer IdM without having two different accounts and passwords. Note that currently, selected features in IdM may still be unavailable to AD users.

(JIRA:RHELPLAN-10442)

sssctl prints an HBAC rules report for an IdM domain

With this update, the **sssctl** utility of the System Security Services Daemon (SSSD) can print an access control report for an Identity Management (IdM) domain. This feature meets the need of certain environments to see, for regulatory reasons, a list of users and groups that can access a specific client machine. Running **sssctl access-report domain_name** on an IdM client prints the parsed subset of host-based access control (HBAC) rules in the IdM domain that apply to the client machine.

Note that no other providers than IdM support this feature.

(JIRA:RHELPLAN-10443)

Identity Management packages are available as a module

In RHEL 8, the packages necessary for installing an Identity Management (IdM) server and client are shipped as a module. The **client** stream is the default stream of the **idm** module and you can download the packages necessary for installing the client without enabling the stream.

The IdM server module stream is called the **DL1** stream. The stream contains multiple profiles corresponding to different types of IdM servers: server, dns, adtrust, client, and default. To download the packages in a specific profile of the **DL1** stream:

1. Enable the stream.
2. Switch to the RPMs delivered through the stream.
3. Run the **yum module install idm:DL1/profile_name** command.

To switch to a new module stream once you have already enabled a specific stream and downloaded packages from it:

1. Remove all the relevant installed content and disable the current module stream.
2. Enable the new module stream.

(JIRA:RHELPLAN-10438)

Session recording solution for RHEL 8 added

A session recording solution has been added to Red Hat Enterprise Linux 8 (RHEL 8). A new **tlog** package and its associated web console session player enable to record and playback the user terminal sessions. The recording can be configured per user or user group via the System Security Services Daemon (SSSD) service. All terminal input and output is captured and stored in a text-based format in a system journal. The input is inactive by default for security reasons not to intercept raw passwords and other sensitive information.

The solution can be used for auditing of user sessions on security-sensitive systems. In the event of a security breach, the recorded sessions can be reviewed as a part of a forensic analysis. The system administrators are now able to configure the session recording locally and view the result from the RHEL 8 web console interface or from the Command-Line Interface using the **tlog-play** utility.

(JIRA:RHELPLAN-1473)

authselect simplifies the configuration of user authentication

This update introduces the **authselect** utility that simplifies the configuration of user authentication on RHEL 8 hosts, replacing the **authconfig** utility. **authselect** comes with a safer approach to PAM stack management that makes the PAM configuration changes simpler for system administrators. **authselect** can be used to configure authentication methods such as passwords, certificates, smart cards, and fingerprint. Note that **authselect** does not configure services required to join remote domains. This task is performed by specialized tools, such as **realmd** or **ipa-client-install**.

(JIRA:RHELPLAN-10445)

SSSD now enforces AD GPOs by default

The default setting for the SSSD option **ad_gpo_access_control** is now **enforcing**. In RHEL 8, SSSD enforces access control rules based on Active Directory Group Policy Objects (GPOs) by default.

Red Hat recommends ensuring GPOs are configured correctly in Active Directory before upgrading from RHEL 7 to RHEL 8. If you would not like to enforce GPOs, change the value of the **ad_gpo_access_control** option in the **/etc/sssd/sssd.conf** file to **permissive**.

(JIRA:RHELPLAN-51289)

5.1.11. Compilers and development tools

Boost updated to version 1.66

The **Boost** C++ library has been updated to upstream version 1.66. The version of **Boost** included in Red Hat Enterprise Linux 7 is 1.53. For details, see the upstream changelogs: https://www.boost.org/users/history/

This update introduces the following changes breaking compatibility with previous versions:

- The **bs_set_hook()** function, the **splay_set_hook()** function from splay containers, and the **bool splay = true** extra parameter in the **splaytree_algorithms()** function in the **Intrusive** library have been removed.
- Comments or string concatenation in JSON files are no longer supported by the parser in the **Property Tree** library.
- Some distributions and special functions from the **Math** library have been fixed to behave as documented and raise an **overflow_error** instead of returning the maximum finite value.
- Some headers from the **Math** library have been moved into the directory **libs/math/include_private**.
- Behavior of the **basic_regex<>::mark_count()** and **basic_regex<>::subexpression(n)** functions from the **Regex** library has been changed to match their documentation.
- Use of variadic templates in the **Variant** library may break metaprogramming functions.
- The **boost::python::numeric** API has been removed. Users can use **boost::python::numpy** instead.
- Arithmetic operations on pointers to non-object types are no longer provided in the Atomic library.

(BZ#1494495)

Unicode 11.0.0 support

The Red Hat Enterprise Linux core C library, **glibc**, has been updated to support the Unicode standard version 11.0.0. As a result, all wide character and multi-byte character APIs including transliteration and conversion between character sets provide accurate and correct information conforming to this standard.

(BZ#1512004)

The boost package is now independent of Python

With this update, installing the **boost** package no longer installs the **Boost.Python** library as a dependency. In order to use **Boost.Python**, you need to explicitly install the **boost-python3** or **boost-python3-devel** packages.

(BZ#1616244)

A new compat-libgfortran-48 package available

For compatibility with Red Hat Enterprise Linux 6 and 7 applications using the Fortran library, a new **compat-libgfortran-48** compatibility package is now available, which provides the **libgfortran.so.3** library.

(BZ#1607227)

Retpoline support in GCC

This update adds support for retpolines to GCC. A retpoline is a software construct used by the kernel to reduce overhead of mitigating Spectre Variant 2 attacks described in CVE-2017-5715.

(BZ#1535774)

Enhanced support for the 64-bit ARM architecture in toolchain components

Toolchain components, **GCC** and **binutils**, now provide extended support for the 64-bit ARM architecture. For example:

- **GCC** and **binutils** now support Scalable Vector Extension (SVE).
- Support for the **FP16** data type, provided by ARM v8.2, has been added to **GCC**. The **FP16** data type improves performance of certain algorithms.
- Tools from **binutils** now support the ARM v8.3 architecture definition, including Pointer Authentication. The Pointer Authentication feature prevents malicious code from corrupting the normal execution of a program or the kernel by crafting their own function pointers. As a result, only trusted addresses are used when branching to different places in the code, which improves security.

(BZ#1504980, BZ#1550501, BZ#1504995, BZ#1504993, BZ#1504994)

Optimizations to glibc for IBM POWER systems

This update provides a new version of **glibc** that is optimized for both IBM POWER 8 and IBM POWER 9 architectures. As a result, IBM POWER 8 and IBM POWER 9 systems now automatically switch to the appropriate, optimized **glibc** variant at run time.

(BZ#1376834)

GNU C Library updated to version 2.28

Red Hat Enterprise Linux 8 includes version 2.28 of the GNU C Library (glibc). Notable improvements include:

- Security hardening features:
  - Secure binary files marked with the **AT_SECURE** flag ignore the **LD_LIBRARY_PATH** environment variable.
  - Backtraces are no longer printed for stack checking failures to speed up shutdown and avoid running more code in a compromised environment.
- Performance improvements:
  - Performance of the **malloc()** function has been improved with a thread local cache.
  - Addition of the **GLIBC_TUNABLES** environment variable to alter library performance characteristics.
  - Implementation of thread semaphores has been improved and new scalable **pthread_rwlock_xxx()** functions have been added.
  - Performance of the math library has been improved.
- Support for Unicode 11.0.0 has been added.
- Improved support for 128-bit floating point numbers as defined by the ISO/IEC/IEEE 60559:2011, IEEE 754-2008, and ISO/IEC TS 18661-3:2015 standards has been added.
- Domain Name Service (DNS) stub resolver improvements related to the **/etc/resolv.conf** configuration file:
  - Configuration is automatically reloaded when the file is changed.

- Support for an arbitrary number of search domains has been added.
- Proper random selection for the **rotate** option has been added.

- New features for development have been added, including:
  - Linux wrapper functions for the **preadv2** and **pwritev2** kernel calls
  - New functions including **reallocarray()** and **explicit_bzero()**
  - New flags for the **posix_spawnattr_setflags()** function such as **POSIX_SPAWN_SETSID**

(BZ#1512010, BZ#1504125, BZ#506398)

CMake available in RHEL

The CMake build system version 3.11 is available in Red Hat Enterprise Linux 8 as the **cmake** package.

(BZ#1590139, BZ#1502802)

make version 4.2.1

Red Hat Enterprise Linux 8 is distributed with the **make** build tool version 4.2.1. Notable changes include:

- When a recipe fails, the name of the makefile and line number of the recipe are shown.
- The **--trace** option has been added to enable tracing of targets. When this option is used, every recipe is printed before invocation even if it would be suppressed, together with the file name and line number where this recipe is located, and also with the prerequisites causing it to be invoked.
- Mixing explicit and implicit rules no longer cause **make** to terminate execution. Instead, a warning is printed. Note that this syntax is deprecated and may be completely removed in the future.
- The **$(file ...)** function has been added to write text to a file. When called without a text argument, it only opens and immediately closes the file.
- A new option, **--output-sync** or **-O**, causes an output from multiple jobs to be grouped per job and enables easier debugging of parallel builds.
- The **--debug** option now accepts also the **n** (none) flag to disable all currently enabled debugging settings.
- The **!=** shell assignment operator has been added as an alternative to the **$(shell ...)** function to increase compatibility with BSD makefiles. For more details and differences between the operator and the function, see the GNU make manual.
  Note that as a consequence, variables with a name ending in exclamation mark and immediately followed by assignment, such as **variable!=value**, are now interpreted as the new syntax. To restore the previous behavior, add a space after the exclamation mark, such as **variable! =value**.
- The **::=** assignment operator defined by the POSIX standard has been added.
- When the **.POSIX** variable is specified, **make** observes the POSIX standard requirements for handling backslash and new line. In this mode, any trailing space before the backslash is preserved, and each backslash followed by a new line and white space characters is converted to a single space character.

- Behavior of the **MAKEFLAGS** and **MFLAGS** variables is now more precisely defined.
- A new variable, **GNUMAKEFLAGS**, is parsed for **make** flags identically to **MAKEFLAGS**. As a consequence, GNU **make**-specific flags can be stored outside **MAKEFLAGS** and portability of makefiles is increased.
- A new variable, **MAKE_HOST**, containing the host architecture has been added.
- The new variables, **MAKE_TERMOUT** and **MAKE_TERMERR**, indicate whether **make** is writing standard output and error to a terminal.
- Setting the **-r** and **-R** options in the **MAKEFLAGS** variable inside a makefile now works correctly and removes all built-in rules and variables, respectively.
- The **.RECIPEPREFIX** setting is now remembered per recipe. Additionally, variables expanded in that recipe also use that recipe prefix setting.
- The **.RECIPEPREFIX** setting and all target-specific variables are displayed in the output of the **-p** option as if in a makefile, instead of as comments.

(BZ#1641015)

SystemTap version 4.0

Red Hat Enterprise Linux 8 is distributed with the **SystemTap** instrumentation tool version 4.0. Notable improvements include:

- The extended Berkeley Packet Filter (eBPF) backend has been improved, especially strings and functions. To use this backend, start **SystemTap** with the **--runtime=bpf** option.
- A new export network service for use with the Prometheus monitoring system has been added.
- The system call probing implementation has been improved to use the kernel tracepoints if necessary.

(BZ#1641032)

Improvements in binutils version 2.30

Red Hat Enterprise Linux 8 includes version 2.30 of the **binutils** package. Notable improvements include:

- Support for new IBM Z architecture extensions has been improved.

Linkers:

- The linker now puts code and read-only data into separate segments by default. As a result, the created executable files are bigger and more safe to run, because the dynamic loader can disable execution of any memory page containing read-only data.
- Support for GNU Property notes which provide hints to the dynamic loader about the binary file has been added.
- Previously, the linker generated invalid executable code for the Intel Indirect Branch Tracking (IBT) technology. As a consequence, the generated executable files could not start. This bug has been fixed.

- Previously, the **gold** linker merged property notes improperly. As a consequence, wrong hardware features could be enabled in the generated code, and the code could terminate unexpectedly. This bug has been fixed.

- Previously, the **gold** linker created note sections with padding bytes at the end to achieve alignment according to architecture. Because the dynamic loader did not expect the padding, it coud terminate unexpectedly the program it was loading. This bug has been fixed.

Other tools:

- The **readelf** and **objdump** tools now have options to follow links into separate debug information files and display information in them, too.

- The new **--inlines** option extends the existing **--line-numbers** option of the **objdump** tool to display nesting information for inlined functions.

- The **nm** tool gained a new option **--with-version-strings** to display version information of a symbol after its name, if present.

- Support for the ARMv8-R architecture and Cortex-R52, Cortex-M23, and Cortex-M33 processors has been added to the assembler.

(BZ#1641004, BZ#1637072, BZ#1501420, BZ#1504114, BZ#1614908, BZ#1614920)

Performance Co-Pilot version 4.3.0

Red Hat Enterprise Linux 8 is distributed with **Performance Co-Pilot** (PCP) version 4.3.0. Notable improvements include:

- The **pcp-dstat** tool now includes historical analysis and Comma-separated Values (CSV) format output.

- The log utilities can use metric labels and help text records.

- The **pmdaperfevent** tool now reports the correct CPU numbers at the lower Simultaneous Multi Threading (SMT) levels.

- The **pmdapostgresql** tool now supports **Postgres** series 10.x.

- The **pmdaredis** tool now supports **Redis** series 5.x.

- The **pmdabcc** tool has been enhanced with dynamic process filtering and per-process syscalls, ucalls, and ustat.

- The **pmdammv** tool now exports metric labels, and the format version is increased to 3.

- The **pmdagfs2** tool supports additional glock and glock holder metrics.

- Several fixes have been made to the SELinux policy.

(BZ#1641034)

Memory Protection Keys

This update enables hardware features which allow per-thread page protection flag changes. The new **glibc** system call wrappers have been added for the **pkey_alloc()**, **pkey_free()**, and **pkey_mprotect()** functions. In addition, the **pkey_set()** and **pkey_get()** functions have been added to allow access to the per-thread protection flags.

(BZ#1304448)

GCC now defaults to z13 on IBM Z

With this update, by default GCC on the IBM Z architecture builds code for the z13 processor, and the code is tuned for the z14 processor. This is equivalent to using the **-march=z13** and **-mtune=z14** options. Users can override this default by explicitly using options for target architecture and tuning.

(BZ#1571124)

elfutils updated to version 0.174

In Red Hat Enterprise Linux 8, the **elfutils** package is available in version 0.174. Notable changes include:

- Previously, the **eu-readelf** tool could show a variable with a negative value as if it had a large unsigned value, or show a large unsigned value as a negative value. This has been corrected and **eu-readelf** now looks up the size and signedness of constant value types to display them correctly.
- A new function **dwarf_next_lines()** for reading **.debug_line** data lacking CU has been added to the **libdw** library. This function can be used as alternative to the **dwarf_getsrclines()** and **dwarf_getsrcfiles()** functions.
- Previously, files with more than 65280 sections could cause errors in the **libelf** and **libdw** libraries and all tools using them. This bug has been fixed. As a result, extended **shnum** and **shstrndx** values in ELF file headers are handled correctly.

(BZ#1641007)

Valgrind updated to version 3.14

Red Hat Enterprise Linux 8 is distributed with the Valgrind executable code analysis tool version 3.14. Notable changes include:

- A new **--keep-debuginfo** option has been added to enable retention of debug info for unloaded code. As a result, saved stack traces can include file and line information for code that is no longer present in memory.
- Suppressions based on source file name and line number have been added.
- The **Helgrind** tool has been extended with an option **--delta-stacktrace** to specify computation of full history stack traces. Notably, using this option together with **--history-level=full** can improve **Helgrind** performance by up to 25%.
- False positive rate in the **Memcheck** tool for optimised code on the Intel and AMD 64-bit arcitectures and the ARM 64-bit architecture has been reduced. Note that you can use the **--expensive-definedness-checks** to control handling of definedness checks and improve the rate at the expense of performance.
- Valgrind can now recognize more instructions of the little-endian variant of IBM Power Systems.
- Valgrind can now process most of the integer and string vector instructions of the IBM Z architecture z13 processor.

For more information about the new options and their known limitations, see the **valgrind(1)** manual page.

(BZ#1641029, BZ#1501419)

GDB version 8.2

Red Hat Enterprise Linux 8 is distributed with the GDB debugger version 8.2 Notable changes include:

- The IPv6 protocol is supported for remote debugging with GDB and **gdbserver**.
- Debugging without debug information has been improved.
- Symbol completion in the GDB user interface has been improved to offer better suggestions by using more syntactic constructions such as ABI tags or namespaces.
- Commands can now be executed in the background.
- Debugging programs created in the Rust programming language is now possible.
- Debugging C and C++ languages has been improved with parser support for the **_Alignof** and **alignof** operators, C++ rvalue references, and C99 variable-length automatic arrays.
- GDB extension scripts can now use the Guile scripting language.
- The Python scripting language interface for extensions has been improved with new API functions, frame decorators, filters, and unwinders. Additionally, scripts in the **.debug_gdb_scripts** section of GDB configuration are loaded automatically.
- GDB now uses Python version 3 to run its scripts, including pretty printers, frame decorators, filters, and unwinders.
- The ARM and 64-bit ARM architectures have been improved with process execution record and replay, including Thumb 32-bit and system call instructions.
- GDB now supports the Scalable Vector Extension (SVE) on the 64-bit ARM architecture.
- Support for Intel PKU register and Intel Processor Trace has been added.
- Record and replay functionality has been extended to include the **rdrand** and **rdseed** instructions on Intel based systems.
- Functionality of GDB on the IBM Z architecture has been extended with support for tracepoints and fast tracepoints, vector registers and ABI, and the **Catch** system call. Additionally, GDB now supports more recent instructions of the architecture.
- GDB can now use the SystemTap static user space probes (SDT) on the 64-bit ARM architecture.

(BZ#1641022, BZ#1497096, BZ#1505346, BZ#1592332, BZ#1550502)

glibc localization for RHEL is distributed in multiple packages

In RHEL 8, **glibc** locales and translations are no longer provided by the single **glibc-common** package. Instead, every locale and language is available in a **glibc-langpack-*CODE*** package. Additionally, in most cases not all locales are installed by default, only these selected in the installer. Users must install all further locale packages that they need separately, or if they wish they can install **glibc-all-langpacks** to get the locales archive containing all the **glibc** locales installed as before.

For more information, see Using langpacks.

(BZ#1512009)

GCC version 8.2

In Red Hat Enterprise Linux 8, the GCC toolchain is based on the GCC 8.2 release series. Notable changes include:

- Numerous general optimizations have been added, such as alias analysis, vectorizer improvements, identical code folding, inter-procedural analysis, store merging optimization pass, and others.
- The Address Sanitizer has been improved. The Leak Sanitizer and Undefined Behavior Sanitizer have been added.
- Debug information can now be produced in the DWARF5 format. This capability is experimental.
- The source code coverage analysis tool GCOV has been extended with various improvements.
- New warnings and improved diagnostics have been added for static detection of more programming errors.
- GCC has been extended to provide tools to ensure additional hardening of the generated code. Improvements related to security include built-ins for overflow checking, additional protection against stack clash, checking target addresses of control-flow instructions, warnings for bounded string manipulation functions, and warnings to detect out-of-bounds array indices.

Improvements to architecture and processor support include:

- Multiple new architecture-specific options for the Intel AVX-512 architecture, a number of its microarchitectures, and Intel Software Guard Extensions (SGX) have been added.
- Code generation can now target the 64-bit ARM architecture LSE extensions, ARMv8.2-A 16-bit Floating-Point Extensions (FPE), and ARMv8.2-A, ARMv8.3-A, and ARMv8.4-A architecture versions.
- Support for the z13 and z14 processors of the IBM Z architecture has been added.

Notable changes related to languages and standards include:

- The default standard used when compiling code in the C language has changed to C17 with GNU extensions.
- The default standard used when compiling code in the C++ language has changed to C++14 with GNU extensions.
- The C++ runtime library now supports the C++11 and C++14 standards.
- The C++ compiler now implements the C++14 standard.
- Support for the C language standard C11 has been improved.
- The new **__auto_type** GNU C extension provides a subset of the functionality of C++11 **auto** keyword in the C language.
- The **_FloatN** and **_FloatNx** type names specified by the ISO/IEC TS 18661-3:2015 standard are now recognized by the C front end.
- Passing an empty class as an argument now takes up no space on the Intel 64 and AMD64 architectures, as required by the platform ABI.

- The value returned by the C++11 **alignof** operator has been corrected to match the C **_Alignof** operator and return minimum alignment. To find the preferred alignment, use the GNU extension **__alignof__**.
- The main version of the **libgfortran** library for Fortran language code has been changed to 5.
- Support for the Ada (GNAT), GCC Go, and Objective C/C++ languages has been removed. Use the Go Toolset for Go code development.

(JIRA:RHELPLAN-7437, BZ#1512593, BZ#1512378)

The Go cryptographic library FIPS mode now honors system settings

Previously, the Go standard cryptographic library always used its FIPS mode unless it was explicitly disabled at build time of the application using the library. As a consequence, users of Go-based applications could not control whether the FIPS mode was used. With this change, the library does not default to FIPS mode when the system is not configured in FIPS mode. As a result, users of Go-based applications on RHEL systems have more control over the use of the FIPS mode of the Go cryptographic library.

(BZ#1633351)

strace updated to version 4.24

Red Hat Enterprise Linux 8 is distributed with the **strace** tool version 4.24. Notable changes include:

- System call tampering features have been added with the **-e inject=** option. This includes injection of errors, return values, delays, and signals.
- System call qualification syntax has been improved:
  - The **-e trace=/regex** option has been added to filter system calls with regular expressions.
  - Prepending a question mark to a system call qualification in the **-e trace=** option lets **strace** continue, even if the qualification does not match any system call.
  - Personality designation has been added to system call qualifications in the **-e trace** option.
- Decoding of **kvm vcpu** exit reason has been added. To do so, use the **-e kvm=vcpu** option.
- The **libdw** library from **elfutils** is now used for stack unwinding when the **-k** option is used. Additionally, symbol demangling is performed using the **libiberty** library.
- Previously, the **-r** option caused **strace** to ignore the **-t** option. This has been fixed, and the two options are now independent.
- The **-A** option has been added for opening output files in append mode.
- The **-X** option has been added for configuring **xlat** output formatting.
- Decoding of socket addresses with the **-yy** option has been improved. Additionally, block and character device number printing in **-yy** mode has been added.
- It is now possible to trace both 64-bit and 32-bit binaries with a single **strace** tool on the IBM Z architecture. As a consequence, the separate **strace32** package no longer exists in RHEL 8.

Additionally, decoding of the following items has been added, improved or updated:

- **netlink** protocols, messages and attributes
- **arch_prctl**, **bpf**, **getsockopt**, **io_pgetevent**, **keyctl**, **prctl**, **pkey_alloc**, **pkey_free**, **pkey_mprotect**, **ptrace**, **rseq**, **setsockopt**, **socket**, **statx** and other system calls
- Multiple commands for the **ioctl** system call
- Constants of various types
- Path tracing for **execveat**, **inotify_add_watch**, **inotify_init**, **select**, **symlink**, **symlinkat** system calls and **mmap** system calls with indirect arguments
- Lists of signal codes

(BZ#1641014)

Compiler toolsets in RHEL 8

RHEL 8.0 provides the following compiler toolsets as Application Streams:

- Clang and LLVM Toolset 7.0.1, which provides the LLVM compiler infrastructure framework, the Clang compiler for the C and C++ languages, the LLDB debugger, and related tools for code analysis. See the Using Clang and LLVM Toolset document.
- Rust Toolset 1.31, which provides the Rust programming language compiler **rustc**, the **cargo** build tool and dependency manager, the **cargo-vendor** plugin, and required libraries. See the Using Rust Toolset document.
- Go Toolset 1.11.5, which provides the Go programming language tools and libraries. Go is alternatively known as **golang**. See the Using Go Toolset document.

(BZ#1695698, BZ#1613515, BZ#1613516, BZ#1613518)

Java implementations and Java tools in RHEL 8

The RHEL 8 AppStream repository includes:

- The **java-11-openjdk** packages, which provide the OpenJDK 11 Java Runtime Environment and the OpenJDK 11 Java Software Development Kit.
- The **java-1.8.0-openjdk** packages, which provide the OpenJDK 8 Java Runtime Environment and the OpenJDK 8 Java Software Development Kit.
- The **icedtea-web** packages, which provide an implementation of Java Web Start.
- The **ant** module, providing a Java library and command-line tool for compiling, assembling, testing, and running Java applications. **Ant** has been updated to version 1.10.
- The **maven** module, providing a software project management and comprehension tool. **Maven** was previously available only as a Software Collection or in the unsupported Optional channel.
- The **scala** module, providing a general purpose programming language for the Java platform. **Scala** was previously available only as a Software Collection.

In addition, the **java-1.8.0-ibm** packages are distributed through the Supplementary repository. Note that packages in this repository are unsupported by Red Hat.

(BZ#1699535)

C++ ABI change in std::string and std::list

The Application Binary Interface (ABI) of the **std::string** and **std::list** classes from the **libstdc++** library changed between RHEL 7 (GCC 4.8) and RHEL 8 (GCC 8) to conform to the C++11 standard. The **libstdc++** library supports both the old and new ABI, but some other C++ system libraries do not. As a consequence, applications that dynamically link against these libraries will need to be rebuilt. This affects all C++ standard modes, including C++98. It also affects applications built with Red Hat Developer Toolset compilers for RHEL 7, which kept the old ABI to maintain compatibility with the system libraries.

(BZ#1704867)

5.1.12. File systems and storage

Support for Data Integrity Field/Data Integrity Extension (DIF/DIX)

DIF/DIX is supported on configurations where the hardware vendor has qualified it and provides full support for the particular host bus adapter (HBA) and storage array configuration on RHEL.

DIF/DIX is not supported on the following configurations:

- It is not supported for use on the boot device.
- It is not supported on virtualized guests.
- Red Hat does not support using the Automatic Storage Management library (ASMLib) when DIF/DIX is enabled.

DIF/DIX is enabled or disabled at the storage device, which involves various layers up to (and including) the application. The method for activating the DIF on storage devices is device-dependent.

For further information on the DIF/DIX feature, see What is DIF/DIX.

(BZ#1649493)

XFS now supports shared copy-on-write data extents

The XFS file system supports shared copy-on-write data extent functionality. This feature enables two or more files to share a common set of data blocks. When either of the files sharing common blocks changes, XFS breaks the link to common blocks and creates a new file. This is similar to the copy-on-write (COW) functionality found in other file systems.

Shared copy-on-write data extents are:

Fast

Creating shared copies does not utilize disk I/O.

Space-efficient

Shared blocks do not consume additional disk space.

Transparent

Files sharing common blocks act like regular files.

Userspace utilities can use shared copy-on-write data extents for:

- Efficient file cloning, such as with the **cp --reflink** command
- Per-file snapshots

This functionality is also used by kernel subsystems such as Overlayfs and NFS for more efficient operation.

Shared copy-on-write data extents are now enabled by default when creating an XFS file system, starting with the **xfsprogs** package version **4.17.0-2.el8**.

Note that Direct Access (DAX) devices currently do not support XFS with shared copy-on-write data extents. To create an XFS file system without this feature, use the following command:

```
# mkfs.xfs -m reflink=0 block-device
```

Red Hat Enterprise Linux 7 can mount XFS file systems with shared copy-on-write data extents only in the read-only mode.

(BZ#1494028)

Maximum XFS file system size is 1024 TiB

The maximum supported size of an XFS file system has been increased from 500 TiB to 1024 TiB.

File systems larger than 500 TiB require that:

- the metadata CRC feature and the free inode btree feature are both enabled in the file system format, and
- the allocation group size is at least 512 GiB.

In RHEL 8, the **mkfs.xfs** utility creates file systems that meet these requirements by default.

Growing a smaller file system that does not meet these requirements to a new size greater than 500 TiB is not supported.

(BZ#1563617)

ext4 file system now supports metadata checksum

With this update, **ext4** metadata is protected by **checksums**. This enables the file system to recognize the corrupt metadata, which avoids damage and increases the file system resilience.

(BZ#1695584)

VDO now supports all architectures

Virtual Data Optimizer (VDO) is now available on all of the architectures supported by RHEL 8.

For the list of supported architectures, see Chapter 2, *Architectures*.

(BZ#1534087)

The BOOM boot manager simplifies the process of creating boot entries

BOOM is a boot manager for Linux systems that use boot loaders supporting the BootLoader Specification for boot entry configuration. It enables flexible boot configuration and simplifies the creation of new or modified boot entries: for example, to boot snapshot images of the system created using LVM.

BOOM does not modify the existing boot loader configuration, and only inserts additional entries. The existing configuration is maintained, and any distribution integration, such as kernel installation and update scripts, continue to function as before.

BOOM has a simplified command-line interface (CLI) and API that ease the task of creating boot entries.

(BZ#1649582)

LUKS2 is now the default format for encrypting volumes

In RHEL 8, the LUKS version 2 (LUKS2) format replaces the legacy LUKS (LUKS1) format. The **dm-crypt** subsystem and the **cryptsetup** tool now uses LUKS2 as the default format for encrypted volumes. LUKS2 provides encrypted volumes with metadata redundancy and auto-recovery in case of a partial metadata corruption event.

Due to the internal flexible layout, LUKS2 is also an enabler of future features. It supports auto-unlocking through the generic kernel-keyring token built in **libcryptsetup** that allow users unlocking of LUKS2 volumes using a passphrase stored in the kernel-keyring retention service.

Other notable enhancements include:

- The protected key setup using the wrapped key cipher scheme.
- Easier integration with Policy-Based Decryption (Clevis).
- Up to 32 key slots - LUKS1 provides only 8 key slots.

For more details, see the **cryptsetup(8)** and **cryptsetup-reencrypt(8)** man pages.

(BZ#1564540)

NVMe/FC is fully supported on Broadcom Emulex and Marvell Qlogic Fibre Channel adapters

The NVMe over Fibre Channel (NVMe/FC) transport type is now fully supported in Initiator mode when used with Broadcom Emulex and Marvell Qlogic Fibre Channel 32Gbit adapters that feature NVMe support.

NVMe over Fibre Channel is an additional fabric transport type for the Nonvolatile Memory Express (NVMe) protocol, in addition to the Remote Direct Memory Access (RDMA) protocol that was previously introduced in Red Hat Enterprise Linux.

Enabling NVMe/FC:

- To enable NVMe/FC in the **lpfc** driver, edit the **/etc/modprobe.d/lpfc.conf** file and add the following option:

```
lpfc_enable_fc4_type=3
```

- To enable NVMe/FC in the **qla2xxx** driver, edit the **/etc/modprobe.d/qla2xxx.conf** file and add the following option:

```
qla2xxx.ql2xnvmeenable=1
```

Additional restrictions:

- Multipath is not supported with NVMe/FC.
- NVMe clustering is not supported with NVMe/FC.
- **kdump** is not supported with NVMe/FC.
- Booting from Storage Area Network (SAN) NVMe/FC is not supported.

(BZ#1649497)

New scan_lvs configuration setting

A new **lvm.conf** configuration file setting, **scan_lvs**, has been added and set to 0 by default. The new default behavior stops LVM from looking for PVs that may exist on top of LVs; that is, it will not scan active LVs for more PVs. The default setting also prevents LVM from creating PVs on top of LVs.

Layering PVs on top of LVs can occur by way of VM images placed on top of LVs, in which case it is not safe for the host to access the PVs. Avoiding this unsafe access is the primary reason for the new default behavior. Also, in environments with many active LVs, the amount of device scanning done by LVM can be significantly decreased.

The previous behavior can be restored by changing this setting to 1.

(BZ#1676598)

New overrides section of the DM Multipath configuration file

The **/etc/multipath.conf** file now includes an **overrides** section that allows you to set a configuration value for all of your devices. These attributes are used by DM Multipath for all devices unless they are overwritten by the attributes specified in the **multipaths** section of the **/etc/multipath.conf** file for paths that contain the device. This functionality replaces the **all_devs** parameter of the **devices** section of the configuration file, which is no longer supported.

(BZ#1643294)

Installing and booting from NVDIMM devices is now supported

Prior to this update, Nonvolatile Dual Inline Memory Module (NVDIMM) devices in any mode were ignored by the installer.

With this update, kernel improvements to support NVDIMM devices provide improved system performance capabilities and enhanced file system access for write-intensive applications like database or analytic workloads, as well as reduced CPU overhead.

This update introduces support for:

- The use of NVDIMM devices for installation using the **nvdimm** Kickstart command and the GUI, making it possible to install and boot from NVDIMM devices in sector mode and reconfigure NVDIMM devices into sector mode during installation.
- The extension of **Kickstart** scripts for **Anaconda** with commands for handling NVDIMM devices.
- The ability of **grub2**, **efibootmgr**, and **efivar** system components to handle and boot from NVDIMM devices.

(BZ#1499442)

The detection of marginal paths in DM Multipath has been improved

The **multipathd** service now supports improved detection of marginal paths. This helps multipath devices avoid paths that are likely to fail repeatedly, and improves performance. Marginal paths are paths with persistent but intermittent I/O errors.

The following options in the **/etc/multipath.conf** file control marginal paths behavior:

- **marginal_path_double_failed_time**,
- **marginal_path_err_sample_time**,
- **marginal_path_err_rate_threshold**, and
- **marginal_path_err_recheck_gap_time**.

DM Multipath disables a path and tests it with repeated I/O for the configured sample time if:

- the listed **multipath.conf** options are set,
- a path fails twice in the configured time, and
- other paths are available.

If the path has more than the configured err rate during this testing, DM Multipath ignores it for the configured gap time, and then retests it to see if it is working well enough to be reinstated.

For more information, see the **multipath.conf** man page.

(BZ#1643550)

Multiqueue scheduling on block devices

Block devices now use multiqueue scheduling in Red Hat Enterprise Linux 8. This enables the block layer performance to scale well with fast solid-state drives (SSDs) and multi-core systems.

The traditional schedulers, which were available in RHEL 7 and earlier versions, have been removed. RHEL 8 supports only multiqueue schedulers.

(BZ#1647612)

5.1.13. High availability and clusters

New pcs commands to list available watchdog devices and test watchdog devices

In order to configure SBD with Pacemaker, a functioning watchdog device is required. This release supports the **pcs stonith sbd watchdog list** command to list available watchdog devices on the local node, and the **pcs stonith sbd watchdog test** command to test a watchdog device. For information on the **sbd** command line tool, see the **sbd**(8) man page.

(BZ#1578891)

The pcs command now supports filtering resource failures by an operation and its interval

Pacemaker now tracks resource failures per a resource operation on top of a resource name, and a node. The **pcs resource failcount show** command now allows filtering failures by a resource, node, operation, and interval. It provides an option to display failures aggregated per a resource and node or detailed per a resource, node, operation, and its interval. Additionally, the **pcs resource cleanup** command now allows filtering failures by a resource, node, operation, and interval.

(BZ#1591308)

Timestamps enabled in corosync log

The **corosync** log did not previously contain timestamps, which made it difficult to relate it to logs from other nodes and daemons. With this release, timestamps are present in the **corosync** log.

(BZ#1615420)

New formats for pcs cluster setup, pcs cluster node add and pcs cluster node remove commands

In Red Hat Enterprise Linux 8, **pcs** fully supports Corosync 3, **knet**, and node names. Node names are now required and replace node addresses in the role of node identifier. Node addresses are now optional.

- In the **pcs host auth** command, node addresses default to node names.
- In the **pcs cluster setup** and **pcs cluster node add** commands, node addresses default to the node addresses specified in the **pcs host auth** command.

With these changes, the formats for the commands to set up a cluster, add a node to a cluster, and remove a node from a cluster have changed. For information on these new command formats, see the help display for the **pcs cluster setup**, **pcs cluster node add** and **pcs cluster node remove** commands.

(BZ#1158816)

New pcs commands

Red Hat Enterprise Linux 8 introduces the following new commands.

- RHEL 8 introduces a new command, **pcs cluster node add-guest | remove-guest**, which replaces the **pcs cluster remote-node add | remove** command in RHEL 7.
- RHEL 8 introduces a new command, **pcs quorum unblock**, which replaces the **pcs cluster quorum unblock** command in RHEL 7.
- The **pcs resource failcount reset** command has been removed as it duplicates the functionality of the **pcs resource cleanup** command.
- RHEL 8 introduces new commands which replace the **pcs resource [show]** command in RHEL 7:
  - The **pcs resource [status]** command in RHEL 8 replaces the **pcs resource [show]** command in RHEL 7.
  - The **pcs resource config** command in RHEL 8 replaces the **pcs resource [show] --full** command in RHEL 7.
  - The **pcs resource config *resource id*** command in RHEL 8 replaces the **pcs resource show *resource id*** command in RHEL 7.
- RHEL 8 introduces new commands which replace the **pcs stonith [show]** command in RHEL 7:
  - The **pcs stonith [status]** command in RHEL 8 replaces the **pcs stonith [show]** command in RHEL 7.

- The **pcs stonith config** command in RHEL 8 replaces the **pcs stonith [show] --full** command in RHEL 7.
- The **pcs stonith config** ***resource id*** command in RHEL 8 replaces the **pcs stonith show** ***resource id*** command in RHEL 7.

(BZ#1654280)

Pacemaker 2.0.0 in RHEL 8

The **pacemaker** packages have been upgraded to the upstream version of Pacemaker 2.0.0, which provides a number of bug fixes and enhancements over the previous version:

- The Pacemaker detail log is now **/var/log/pacemaker/pacemaker.log** by default (not directly in **/var/log** or combined with the **corosync** log under **/var/log/cluster**).
- The Pacemaker daemon processes have been renamed to make reading the logs more intuitive. For example, **pengine** has been renamed to **pacemaker-schedulerd**.
- Support for the deprecated **default-resource-stickiness** and **is-managed-default** cluster properties has been dropped. The **resource-stickiness** and **is-managed** properties should be set in resource defaults instead. Existing configurations (though not newly created ones) with the deprecated syntax will automatically be updated to use the supported syntax.
- For a more complete list of changes, see Pacemaker 2.0 upgrade in Red Hat Enterprise Linux 8 .

It is recommended that users who are upgrading an existing cluster using Red Hat Enterprise Linux 7 or earlier, run **pcs cluster cib-upgrade** on any cluster node before and after upgrading RHEL on all cluster nodes.

(BZ#1543494)

Master resources renamed to promotable clone resources

Red Hat Enterprise Linux (RHEL) 8 supports Pacemaker 2.0, in which a master/slave resource is no longer a separate type of resource but a standard clone resource with a **promotable** meta-attribute set to **true**. The following changes have been implemented in support of this update:

- It is no longer possible to create master resources with the **pcs** command. Instead, it is possible to create **promotable** clone resources. Related keywords and commands have been changed from **master** to **promotable**.
- All existing master resources are displayed as promotable clone resources.
- When managing a RHEL7 cluster in the Web UI, master resources are still called master, as RHEL7 clusters do not support promotable clones.

(BZ#1542288)

New commands for authenticating nodes in a cluster

Red Hat Enterprise Linux (RHEL) 8 incorporates the following changes to the commands used to authenticate nodes in a cluster.

- The new command for authentication is **pcs host auth**. This command allows users to specify host names, addresses and **pcsd** ports.

- The **pcs cluster auth** command authenticates only the nodes in a local cluster and does not accept a node list
- It is now possible to specify an address for each node. **pcs**/**pcsd** will then communicate with each node using the specified address. These addresses can be different than the ones **corosync** uses internally.
- The **pcs pcsd clear-auth** command has been replaced by the **pcs pcsd deauth** and **pcs host deauth** commands. The new commands allow users to deauthenticate a single host as well as all hosts.
- Previously, node authentication was bidirectional, and running the **pcs cluster auth** command caused all specified nodes to be authenticated against each other. The **pcs host auth** command, however, causes only the local host to be authenticated against the specified nodes. This allows better control of what node is authenticated against what other nodes when running this command. On cluster setup itself, and also when adding a node, **pcs** automatically synchronizes tokens on the cluster, so all nodes in the cluster are still automatically authenticated as before and the cluster nodes can communicate with each other.

Note that these changes are not backward compatible. Nodes that were authenticated on a RHEL 7 system will need to be authenticated again.

(BZ#1549535)

The pcs commands now support display, cleanup, and synchronization of fencing history

Pacemaker's fence daemon tracks a history of all fence actions taken (pending, successful, and failed). With this release, the **pcs** commands allow users to access the fencing history in the following ways:

- The **pcs status** command shows failed and pending fencing actions
- The **pcs status --full** command shows the entire fencing history
- The **pcs stonith history** command provides options to display and clean up fencing history
- Although fencing history is synchronized automatically, the **pcs stonith history** command now supports an **update** option that allows a user to manually synchronize fencing history should that be necessary

(BZ#1620190, BZ#1615891)

5.1.14. Networking

nftables replaces iptables as the default network packet filtering framework

The **nftables** framework provides packet classification facilities and it is the designated successor to the **iptables**, **ip6tables**, **arptables**, and **ebtables** tools. It offers numerous improvements in convenience, features, and performance over previous packet-filtering tools, most notably:

- lookup tables instead of linear processing
- a single framework for both the **IPv4** and **IPv6** protocols
- rules all applied atomically instead of fetching, updating, and storing a complete ruleset
- support for debugging and tracing in the ruleset (**nftrace**) and monitoring trace events (in the **nft** tool)

- more consistent and compact syntax, no protocol-specific extensions
- a Netlink API for third-party applications

Similarly to **iptables**, **nftables** use tables for storing chains. The chains contain individual rules for performing actions. The **nft** tool replaces all tools from the previous packet-filtering frameworks. The **libnftables** library can be used for low-level interaction with **nftables** Netlink API over the **libmnl** library.

The **iptables**, **ip6tables**, **ebtables** and **arptables** tools are replaced by nftables-based drop-in replacements with the same name. While external behavior is identical to their legacy counterparts, internally they use **nftables** with legacy **netfilter** kernel modules through a compatibility interface where required.

Effect of the modules on the **nftables** ruleset can be observed using the **nft list ruleset** command. Since these tools add tables, chains, and rules to the **nftables** ruleset, be aware that **nftables** rule-set operations, such as the **nft flush ruleset** command, might affect rule sets installed using the formerly separate legacy commands.

To quickly identify which variant of the tool is present, version information has been updated to include the back-end name. In RHEL 8, the nftables-based **iptables** tool prints the following version string:

```
$ iptables --version
iptables v1.8.0 (nf_tables)
```

For comparison, the following version information is printed if legacy **iptables** tool is present:

```
$ iptables --version
iptables v1.8.0 (legacy)
```

(BZ#1644030)

Notable TCP features in RHEL 8

Red Hat Enterprise Linux 8 is distributed with TCP networking stack version 4.18, which provides higher performances, better scalability, and more stability. Performances are boosted especially for busy TCP server with a high ingress connection rate.

Additionally, two new TCP congestion algorithms, **BBR** and **NV**, are available, offering lower latency, and better throughput than cubic in most scenarios.

(BZ#1562998)

firewalld uses nftables by default

With this update, the **nftables** filtering subsystem is the default firewall backend for the **firewalld** daemon. To change the backend, use the **FirewallBackend** option in the **/etc/firewalld/firewalld.conf** file.

This change introduces the following differences in behavior when using **nftables**:

1. **iptables** rule executions always occur before **firewalld** rules
   - **DROP** in **iptables** means a packet is never seen by **firewalld**
   - **ACCEPT** in **iptables** means a packet is still subject to **firewalld** rules

2. **firewalld** direct rules are still implemented through **iptables** while other **firewalld** features use **nftables**

3. direct rule execution occurs before **firewalld** generic acceptance of established connections

(BZ#1509026)

Notable change in wpa_supplicant in RHEL 8

In Red Hat Enterprise Linux (RHEL) 8, the **wpa_supplicant** package is built with **CONFIG_DEBUG_SYSLOG** enabled. This allows reading the **wpa_supplicant** log using the **journalctl** utility instead of checking the contents of the **/var/log/wpa_supplicant.log** file.

(BZ#1582538)

NetworkManager now supports SR-IOV virtual functions

In Red Hat Enterprise Linux 8.0, **NetworkManager** allows configuring the number of virtual functions (VF) for interfaces that support single-root I/O virtualization (SR-IOV). Additionally, **NetworkManager** allows configuring some attributes of the VFs, such as the MAC address, VLAN, the **spoof checking** setting and allowed bitrates. Note that all properties related to SR-IOV are available in the **sriov** connection setting. For more details, see the **nm-settings(5)** man page.

(BZ#1555013)

IPVLAN virtual network drivers are now supported

In Red Hat Enterprise Linux 8.0, the kernel includes support for IPVLAN virtual network drivers. With this update, IPVLAN virtual Network Interface Cards (NICs) enable the network connectivity for multiple containers exposing a single MAC address to the local network. This allows a single host to have a lot of containers overcoming the possible limitation on the number of MAC addresses supported by the peer networking equipment.

(BZ#1261167)

NetworkManager supports a wildcard interface name match for connections

Previously, it was possible to restrict a connection to a given interface using only an exact match on the interface name. With this update, connections have a new **match.interface-name** property which supports wildcards. This update enables users to choose the interface for a connection in a more flexible way using a wildcard pattern.

(BZ#1555012)

Improvements in the networking stack 4.18

Red Hat Enterprise Linux 8.0 includes the networking stack upgraded to upstream version 4.18, which provides several bug fixes and enhancements. Notable changes include:

- Introduced new offload features, such as **UDP_GSO**, and, for some device drivers, **GRO_HW**.
- Improved significant scalability for the User Datagram Protocol (UDP).
- Improved the generic busy polling code.
- Improved scalability for the IPv6 protocol.
- Improved scalability for the routing code.

- Added a new default transmit queue scheduling algorithm,**fq_codel**, which improves a transmission delay.
- Improved scalability for some transmit queue scheduling algorithms. For example, **pfifo_fast** is now lockless.
- Improved scalability of the IP reassembly unit by removing the garbage collection kernel thread and ip fragments expire only on timeout. As a result, CPU usage under DoS is much lower, and the maximum sustainable fragments drop rate is limited by the amount of memory configured for the IP reassembly unit.

(BZ#1562987)

New tools to convert iptables to nftables

This update adds the **iptables-translate** and **ip6tables-translate** tools to convert the existing **iptables** or **ip6tables** rules into the equivalent ones for **nftables**. Note that some extensions lack translation support. If such an extension exists, the tool prints the untranslated rule prefixed with the **#** sign. For example:

```
| % iptables-translate -A INPUT -j CHECKSUM --checksum-fill
| nft # -A INPUT -j CHECKSUM --checksum-fill
```

Additionally, users can use the **iptables-restore-translate** and **ip6tables-restore-translate** tools to translate a dump of rules. Note that before that, users can use the **iptables-save** or **ip6tables-save** commands to print a dump of current rules. For example:

```
| % sudo iptables-save >/tmp/iptables.dump
| % iptables-restore-translate -f /tmp/iptables.dump
| # Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
| add table ip nat
| ...
```

(BZ#1564596)

New features added to VPN using NetworkManager

In Red Hat Enterprise Linux 8.0, **NetworkManager** provides the following new features to VPN:

- Support for the Internet Key Exchange version 2 (IKEv2) protocol.
- Added some more **Libreswan** options, such as the **rightid**, **leftcert**, **narrowing**, **rekey**, **fragmentation** options. For more details on the supported options, see the **nm-settings-libreswan** man page.
- Updated the default ciphers. This means that when the user does not specify the ciphers, the **NetworkManager-libreswan** plugin allows the **Libreswan** application to choose the system default cipher. The only exception is when the user selects an IKEv1 aggressive mode configuration. In this case, the **ike = aes256-sha1;modp1536** and **eps = aes256-sha1** values are passed to **Libreswan**.

(BZ#1557035)

A new data chunk type, I-DATA, added to SCTP

This update adds a new data chunk type, **I-DATA**, and stream schedulers to the Stream Control Transmission Protocol (SCTP). Previously, SCTP sent user messages in the same order as they were

sent by a user. Consequently, a large SCTP user message blocked all other messages in any stream until completely sent. When using **I-DATA** chunks, the Transmission Sequence Number (TSN) field is not overloaded. As a result, SCTP now can schedule the streams in different ways, and **I-DATA** allows user messages interleaving (RFC 8260). Note that both peers must support the **I-DATA** chunk type.

(BZ#1273139)

NetworkManager supports configuring ethtool offload features

With this enhancement, **NetworkManager** supports configuring **ethtool** offload features, and users no longer need to use init scripts or a **NetworkManager** dispatcher script. As a result, users can now configure the offload feature as a part of the connection profile using one of the following methods:

- By using the **nmcli** utility
- By editing key files in the **/etc/NetworkManager/system-connections/** directory
- By editing the **/etc/sysconfig/network-scripts/ifcfg-*** files

Note that this feature is currently not supported in graphical interfaces and in the **nmtui** utility.

(BZ#1335409)

TCP BBR support in RHEL 8

A new TCP congestion control algorithm, Bottleneck Bandwidth and Round-trip time (BBR) is now supported in Red Hat Enterprise Linux (RHEL) 8. BBR attempts to determine the bandwidth of the bottleneck link and the Round-trip time (RTT). Most congestion algorithms are based on packet loss (including CUBIC, the default Linux TCP congestion control algorithm), which have problems on high-throughput links. BBR does not react to loss events directly, it adjusts the TCP pacing rate to match it with the available bandwidth. Users of TCP BBR should switch to the **fq** queueing setting on all the involved interfaces.

Note that users should explicitly use **fq** and not **fq_codel**.

For more details, see the **tc-fq** man page.

(BZ#1515987)

lksctp-tools, version 1.0.18 in RHEL 8

The **lksctp-tools** package, version 3.28 is available in Red Hat Enterprise Linux (RHEL) 8. Notable enhancements and bug fixes include:

- Integration with Travis CI and Coverity Scan
- Support for the **sctp_peeloff_flags** function
- Indication of which kernel features are available
- Fixes on Coverity Scan issues

(BZ#1568622)

Blacklisting SCTP module by default in RHEL 8

To increase security, a set of kernel modules have been moved to the **kernel-modules-extra** package. These are not installed by default. As a consequence, non-root users cannot load these components as they are blacklisted by default. To use one of these kernel modules, the system administrator must

install **kernel-modules-extra** and explicitly remove the module blacklist. As a result, non-root users will be able to load the software component automatically.

(BZ#1642795)

Notable changes in driverctl 0.101

Red Hat Enterprise Linux 8.0 is distributed with **driverctl** 0.101. This version includes the following bug fixes:

- The **shellcheck** warnings have been fixed.
- The bash-completion is installed as **driverctl** instead of **driverctl-bash-completion.sh**.
- The **load_override** function for non-PCI buses has been fixed.
- The **driverctl** service loads all overrides before it reaches the **basic.target** systemd target.

(BZ#1648411)

Added rich rules priorities to firewalld

The **priority** option has been added to rich rules. This allows users to define the desirable priority order during the rule execution and provides more advanced control over rich rules.

(BZ#1648497)

NVMe over RDMA is supported in RHEL 8

In Red Hat Enterprise Linux (RHEL) 8, Nonvolatile Memory Express (NVMe) over Remote Direct Memory Access (RDMA) supports Infiniband, RoCEv2, and iWARP only in initiator mode.

Note that Multipath is supported in failover mode only.

Additional restrictions:

- Kdump is not supported with NVMe/RDMA.
- Booting from NVMe device over RDMA is not supported.

(BZ#1680177)

The nf_tables back end does not support debugging using dmesg

Red Hat Enterprise Linux 8.0 uses the **nf_tables** back end for firewalls that does not support debugging the firewall using the output of the **dmesg** utility. To debug firewall rules, use the **xtables-monitor -t** or **nft monitor trace** commands to decode rule evaluation events.

(BZ#1645744)

Red Hat Enterprise Linux supports VRF

The kernel in RHEL 8.0 supports virtual routing and forwarding (VRF). VRF devices, combined with rules set using the **ip** utility, enable administrators to create VRF domains in the Linux network stack. These domains isolate the traffic on layer 3 and, therefore, the administrator can create different routing tables and reuse the same IP addresses within different VRF domains on one host.

(BZ#1440031)

iproute, version 4.18 in RHEL 8

The **iproute** package is distributed with the version 4.18 in Red Hat Enterprise Linux (RHEL) 8. The most notable change is that the interface alias marked as ethX:Y, such as eth0:1, is no longer supported. To work around this problem, users should remove the alias suffix, which is the colon and the following number before entering **ip link show**.

(BZ#1589317)

5.1.15. Security

SWID tag of the RHEL 8.0 release

To enable identification of RHEL 8.0 installations using the ISO/IEC 19770-2:2015 mechanism, software identification (SWID) tags are installed in files **/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8-<architecture>.swidtag** and **/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.swidtag**. The parent directory of these tags can also be found by following the **/etc/swid/swidtags.d/redhat.com** symbolic link.

The XML signature of the SWID tag files can be verified using the **xmlsec1 verify** command, for example:

```
xmlsec1 verify --trusted-pem /etc/pki/swid/CA/redhat.com/redhatcodesignca.cert
/usr/share/redhat.com/com.redhat.RHEL-8-x86_64.swidtag
```

The certificate of the code signing certification authority can also be obtained from the Product Signing Keys page on the Customer Portal.

(BZ#1636338)

System-wide cryptographic policies are applied by default

Crypto-policies is a component in Red Hat Enterprise Linux 8, which configures the core cryptographic subsystems, covering the TLS, IPsec, DNSSEC, Kerberos, and SSH protocols. It provides a small set of policies, which the administrator can select using the **update-crypto-policies** command.

The **DEFAULT** system-wide cryptographic policy offers secure settings for current threat models. It allows the TLS 1.2 and 1.3 protocols, as well as the IKEv2 and SSH2 protocols. The RSA keys and Diffie-Hellman parameters are accepted if larger than 2047 bits.

See the Consistent security by crypto policies in Red Hat Enterprise Linux 8 article on the Red Hat Blog and the **update-crypto-policies(8)** man page for more information.

(BZ#1591620)

OpenSSH rebased to version 7.8p1

The **openssh** packages have been upgraded to upstream version 7.8p1. Notable changes include:

- Removed support for the **SSH version 1** protocol.
- Removed support for the **hmac-ripemd160** message authentication code.
- Removed support for RC4 (**arcfour**) ciphers.
- Removed support for **Blowfish** ciphers.

- Removed support for **CAST** ciphers.
- Changed the default value of the **UseDNS** option to **no**.
- Disabled **DSA** public key algorithms by default.
- Changed the minimal modulus size for **Diffie-Hellman** parameters to 2048 bits.
- Changed semantics of the **ExposeAuthInfo** configuration option.
- The **UsePrivilegeSeparation=sandbox** option is now mandatory and cannot be disabled.
- Set the minimal accepted **RSA** key size to 1024 bits.

(BZ#1622511)

The automatic OpenSSH server keys generation is now handled by sshd-keygen@.service

OpenSSH creates RSA, ECDSA, and ED25519 server host keys automatically if they are missing. To configure the host key creation in RHEL 8, use the **sshd-keygen@.service** instantiated service.

For example, to disable the automatic creation of the RSA key type:

```
# systemctl mask sshd-keygen@rsa.service
```

See the **/etc/sysconfig/sshd** file for more information.

(BZ#1228088)

ECDSA keys are supported for SSH authentication

This release of the **OpenSSH** suite introduces support for ECDSA keys stored on PKCS #11 smart cards. As a result, users can now use both RSA and ECDSA keys for SSH authentication.

(BZ#1645038)

libssh implements SSH as a core cryptographic component

This change introduces **libssh** as a core cryptographic component in Red Hat Enterprise Linux 8. The **libssh** library implements the Secure Shell (SSH) protocol.

Note that the client side of **libssh** follows the configuration set for **OpenSSH** through system-wide crypto policies, but the configuration of the server side cannot be changed through system-wide crypto policies.

(BZ#1485241)

TLS 1.3 support in cryptographic libraries

This update enables Transport Layer Security (TLS) 1.3 by default in all major back-end crypto libraries. This enables low latency across the operating system communications layer and enhances privacy and security for applications by taking advantage of new algorithms, such as RSA-PSS or X25519.

(BZ#1516728)

NSS now use SQL by default

The Network Security Services (NSS) libraries now use the SQL file format for the trust database by default. The DBM file format, which was used as a default database format in previous releases, does not

support concurrent access to the same database by multiple processes and it has been deprecated in upstream. As a result, applications that use the NSS trust database to store keys, certificates, and revocation information now create databases in the SQL format by default. Attempts to create databases in the legacy DBM format fail. The existing DBM databases are opened in read-only mode, and they are automatically converted to the SQL format. Note that NSS support the SQL file format since Red Hat Enterprise Linux 6.

(BZ#1489094)

PKCS #11 support for smart cards and HSMs is now consistent across the system

With this update, using smart cards and Hardware Security Modules (HSM) with PKCS #11 cryptographic token interface becomes consistent. This means that the user and the administrator can use the same syntax for all related tools in the system. Notable enhancements include:

- Support for the PKCS #11 Uniform Resource Identifier (URI) scheme that ensures a simplified enablement of tokens on RHEL servers both for administrators and application writers.
- A system-wide registration method for smart cards and HSMs using the **pkcs11.conf**.
- Consistent support for HSMs and smart cards is available in NSS, GnuTLS, and OpenSSL (through the **openssl-pkcs11** engine) applications.
- The Apache HTTP server (**httpd**) now seamlessly supports HSMs.

For more information, see the **pkcs11.conf(5)** man page.

(BZ#1516741)

Firefox now works with system-wide registered PKCS #11 drivers

The Firefox web browser automatically loads the **p11-kit-proxy** module and every smart card that is registered system-wide in **p11-kit** through the **pkcs11.conf** file is automatically detected. For using TLS client authentication, no additional setup is required and keys from a smart card are automatically used when a server requests them.

(BZ#1595638)

RSA-PSS is now supported in OpenSC

This update adds support for the RSA-PSS cryptographic signature scheme to the **OpenSC** smart card driver. The new scheme enables a secure cryptographic algorithm required for the TLS 1.3 support in the client software.

(BZ#1595626)

Notable changes in Libreswan in RHEL 8

The **libreswan** packages have been upgraded to upstream version 3.27, which provides many bug fixes and enhancements over the previous versions. Most notable changes include:

- Support for RSA-PSS (RFC 7427) through **authby=rsa-sha2**, ECDSA (RFC 7427) through **authby=ecdsa-sha2**, CURVE25519 using the **dh31** keyword, and CHACHA20-POLY1305 for IKE and ESP through the **chacha20_poly1305** encryption keyword has been added for the IKEv2 protocol.
- Support for the alternative KLIPS kernel module has been removed from **Libreswan**, as upstream has deprecated KLIPS entirely.

- The Diffie-Hellman groups DH22, DH23, and DH24 are no longer supported (as per RFC 8247).

Note that the **authby=rsasig** has been changed to always use the RSA v1.5 method, and the **authby=rsa-sha2** option uses the RSASSA-PSS method. The **authby=rsa-sha1** option is not valid as per RFC 8247. That is the reason **Libreswan** no longer supports SHA-1 with digital signatures.

(BZ#1566574)

System-wide cryptographic policies change the default IKE version in Libreswan to IKEv2

The default IKE version in the Libreswan IPsec implementation has been changed from IKEv1 (RFC 2409) to IKEv2 (RFC 7296). The default IKE and ESP/AH algorithms for use with IPsec have been updated to comply with system-wide crypto policies, RFC 8221, and RFC 8247. Encryption key sizes of 256 bits are now preferred over key sizes of 128 bits.

The default IKE and ESP/AH ciphers now include AES-GCM, CHACHA20POLY1305, and AES-CBC for encryption. For integrity checking, they provide AEAD and SHA-2. The Diffie-Hellman groups now contain DH19, DH20, DH21, DH14, DH15, DH16, and DH18.

The following algorithms have been removed from the default IKE and ESP/AH policies: AES_CTR, 3DES, SHA1, DH2, DH5, DH22, DH23, and DH24. With the exceptions of DH22, DH23, and DH24, these algorithms can be enabled by the **ike=** or **phase2alg=/esp=/ah=** option in IPsec configuration files.

To configure IPsec VPN connections that still require the IKEv1 protocol, add the **ikev2=no** option to connection configuration files. See the **ipsec.conf(5)** man page for more information.

(BZ#1645606)

IKE version-related changes in Libreswan

With this enhancement, Libreswan handles internet key exchange (IKE) settings differently:

- The default internet key exchange (IKE) version has been changed from 1 to 2.
- Connections can now either use the IKEv1 or IKEv2 protocol, but not both.
- The interpretation of the **ikev2** option has been changed:
  - The values **insist** is interpreted as IKEv2-only.
  - The values **no** and **never** are interpreted as IKEv1-only.
  - The values **propose**, **yes** and, **permit** are no longer valid and result in an error, because it was not clear which IKE versions resulted from these values

(BZ#1648776)

New features in OpenSCAP in RHEL 8

The **OpenSCAP** suite has been upgraded to upstream version 1.3.0, which introduces many enhancements over the previous versions. The most notable features include:

- API and ABI have been consolidated - updated, deprecated and/or unused symbols have been removed.
- The probes are not run as independent processes, but as threads within the **oscap** process.
- The command-line interface has been updated.

- **Python 2** bindings have been replaced with **Python 3** bindings.

(BZ#1614273)

SCAP Security Guide now supports system-wide cryptographic policies

The **scap-security-guide** packages have been updated to use predefined system-wide cryptographic policies for configuring the core cryptographic subsystems. The security content that conflicted with or overrode the system-wide cryptographic policies has been removed.

Note that this change applies only on the security content in **scap-security-guide**, and you do not need to update the OpenSCAP scanner or other SCAP components.

(BZ#1618505)

OpenSCAP command-line interface has been improved

The verbose mode is now available in all **oscap** modules and submodules. The tool output has improved formatting.

Deprecated options have been removed to improve the usability of the command-line interface.

The following options are no longer available:

- **--show** in **oscap xccdf generate report** has been completely removed.
- **--probe-root** in **oscap oval eval** has been removed. It can be replaced by setting the environment variable, **OSCAP_PROBE_ROOT**.
- **--sce-results** in **oscap xccdf eval** has been replaced by **--check-engine-results**
- **validate-xml** submodule has been dropped from CPE, OVAL, and XCCDF modules. **validate** submodules can be used instead to validate SCAP content against XML schemas and XSD schematrons.
- **oscap oval list-probes** command has been removed, the list of available probes can be displayed using **oscap --version** instead.

OpenSCAP allows to evaluate all rules in a given XCCDF benchmark regardless of the profile by using **--profile '(all)'**.

(BZ#1618484)

SCAP Security Guide PCI-DSS profile aligns with version 3.2.1

The **scap-security-guide** packages provide the PCI-DSS (Payment Card Industry Data Security Standard) profile for Red Hat Enterprise Linux 8 and this profile has been updated to align with the latest PCI-DSS version - 3.2.1.

(BZ#1618528)

SCAP Security Guide supports OSPP 4.2

The **scap-security-guide** packages provide a draft of the OSPP (Protection Profile for General Purpose Operating Systems) profile version 4.2 for Red Hat Enterprise Linux 8. This profile reflects mandatory configuration controls identified in the NIAP Configuration Annex to the Protection Profile for General Purpose Operating Systems (Protection Profile Version 4.2). SCAP Security Guide provides automated checks and scripts that help users to meet requirements defined in the OSPP.

(BZ#1618518)

Notable changes in rsyslog in RHEL 8

The **rsyslog** packages have been upgraded to upstream version 8.37.0, which provides many bug fixes and enhancements over the previous versions. Most notable changes include:

- Enhanced processing of **rsyslog** internal messages; possibility of rate-limiting them; fixed possible deadlock.
- Enhanced rate-limiting in general; the actual *spam source* is now logged.
- Improved handling of oversized messages - the user can now set how to treat them both in the core and in certain modules with separate actions.
- **mmnormalize** rule bases can now be embedded in the **config** file instead of creating separate files for them.
- All **config** variables, including variables in JSON, are now case-insensitive.
- Various improvements of PostgreSQL output.
- Added a possibility to use shell variables to control **config** processing, such as conditional loading of additional configuration files, executing statements, or including a text in **config**. Note that an excessive use of this feature can make it very hard to debug problems with **rsyslog**.
- 4-digit file creation modes can be now specified in **config**.
- Reliable Event Logging Protocol (RELP) input can now bind also only on a specified address.
- The default value of the **enable.body** option of mail output is now aligned to documentation
- The user can now specify insertion error codes that should be ignored in **MongoDB** output.
- Parallel TCP (pTCP) input has now the configurable backlog for better load-balancing.
- To avoid duplicate records that might appear when **journald** rotated its files, the **imjournal** option has been added. Note that use of this option can affect performance.

Note that the system with **rsyslog** can be configured to provide better performance as described in the Configuring system logging without journald or with minimized journald usage Knowledgebase article.

(BZ#1613880)

New rsyslog module: omkafka

To enable **kafka** centralized data storage scenarios, you can now forward logs to the **kafka** infrastructure using the new **omkafka** module.

(BZ#1542497)

rsyslog imfile now supports symlinks

With this update, the **rsyslog imfile** module delivers better performance and more configuration options. This allows you to use the module for more complicated file monitoring use cases. For example, you can now use file monitors with glob patterns anywhere along the configured path and rotate symlink targets with increased data throughput.

(BZ#1614179)

The default rsyslog configuration file format is now non-legacy

The configuration files in the **rsyslog** packages now use the non-legacy format by default. The legacy format can be still used, however, mixing current and legacy configuration statements has several constraints. Configurations carried from previous RHEL releases should be revised. See the **rsyslog.conf(5)** man page for more information.

(BZ#1619645)

Audit 3.0 replaces audispd with auditd

With this update, functionality of **audispd** has been moved to **auditd**. As a result, **audispd** configuration options are now part of **auditd.conf**. In addition, the **plugins.d** directory has been moved under **/etc/audit**. The current status of **auditd** and its plug-ins can now be checked by running the **service auditd state** command.

(BZ#1616428)

tangd_port_t allows changes of the default port for Tang

This update introduces the **tangd_port_t** SELinux type that allows the **tangd** service run as confined with SELinux enforcing mode. That change helps to simplify configuring a Tang server to listen on a user-defined port and it also preserves the security level provided by SELinux in enforcing mode.

See the Configuring automated unlocking of encrypted volumes using policy-based decryption section for more information.

(BZ#1664345)

New SELinux booleans

This update of the SELinux system policy introduces the following booleans:

- colord_use_nfs
- mysql_connect_http
- pdns_can_network_connect_db
- ssh_use_tcpd
- sslh_can_bind_any_port
- sslh_can_connect_any_port
- virt_use_pcscd

To get a list of booleans including their meaning, and to find out if they are enabled or disabled, install the **selinux-policy-devel** package and use:

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

SELinux now supports systemd No New Privileges

This update introduces the **nnp_nosuid_transition** policy capability that enables SELinux domain transitions under **No New Privileges** (NNP) or **nosuid** if **nnp_nosuid_transition** is allowed between the old and new contexts. The **selinux-policy** packages now contain a policy for **systemd** services that use the **NNP** security feature.

The following rule describes allowing this capability for a service:

```
allow source_domain  target_type:process2 { nnp_transition nosuid_transition };
```

For example:

```
allow init_t fprintd_t:process2 { nnp_transition nosuid_transition };
```

The distribution policy now also contains an m4 macro interface, which can be used in SELinux security policies for services that use the **init_nnp_daemon_domain()** function.

(BZ#1594111)

Support for a new map permission check on the mmap syscall

The SELinux **map** permission has been added to control memory mapped access to files, directories, sockets, and so on. This allows the SELinux policy to prevent direct memory access to various file system objects and ensure that every such access is revalidated.

(BZ#1592244)

SELinux now supports getrlimit permission in the process class

This update introduces a new SELinux access control check, **process:getrlimit**, which has been added for the **prlimit()** function. This enables SELinux policy developers to control when one process attempts to read and then modify the resource limits of another process using the **process:setrlimit** permission. Note that SELinux does not restrict a process from manipulating its own resource limits through **prlimit()**. See the **prlimit(2)** and **getrlimit(2)** man pages for more information.

(BZ#1549772)

selinux-policy now supports VxFS labels

This update introduces support for Veritas File System (VxFS) security extended attributes (xattrs). This enables to store proper SELinux labels with objects on the file system instead of the generic vxfs_t type. As a result, systems with VxFS with full support for SELinux are more secure.

(BZ#1483904)

Compile-time security hardening flags are applied more consistently

Compile-time security hardening flags are applied more consistently on RPM packages in the RHEL 8 distribution, and the **redhat-rpm-config** package now automatically provides security hardening flags. The applied compile-time flags also help to meet Common Criteria (CC) requirements. The following security hardening flags are applied:

- For detection of buffer-overflow errors: **D_FORTIFY_SOURCE=2**
- Standard library hardening that checks for C++ arrays, vectors, and strings: **D_GLIBCXX_ASSERTIONS**
- For Stack Smashing Protector (SSP): **fstack-protector-strong**

- For exception hardening: **fexceptions**
- For Control-Flow Integrity (CFI): **fcf-protection=full** (only on AMD and Intel 64-bit architectures)
- For Address Space Layout Randomization (ASLR): **fPIE** (for executables) or **fPIC** (for libraries)
- For protection against the Stack Clash vulnerability: **fstack-clash-protection** (except ARM)
- Link flags to resolve all symbols on startup: **-Wl**, **-z,now**

See the **gcc(1)** man page for more information.

(JIRA:RHELPLAN-2306)

5.1.16. Virtualization

qemu-kvm 2.12 in RHEL 8

Red Hat Enterprise Linux 8 is distributed with **qemu-kvm** 2.12. This version fixes multiple bugs and adds a number of enhancements over the version 1.5.3, available in Red Hat Enterprise Linux 7.

Notably, the following features have been introduced:

- Q35 guest machine type
- UEFI guest boot
- NUMA tuning and pinning in the guest
- vCPU hot plug and hot unplug
- guest I/O threading

Note that some of the features available in **qemu-kvm** 2.12 are not supported on Red Hat Enterprise Linux 8. For detailed information, see "Feature support and limitations in RHEL 8 virtualization" on the Red Hat Customer Portal.

(BZ#1559240)

The Q35 machine type is now supported by virtualization

Red hat Enterprise Linux 8 introduces the support for **Q35**, a more modern PCI Express-based machine type. This provides a variety of improvements in features and performance of virtual devices, and ensures that a wider range of modern devices are compatible with virtualization. In addition, virtual machines created in Red Hat Enterprise Linux 8 are set to use **Q35** by default.

Also note that the previously default **PC** machine type has become deprecated and should only be used when virtualizing older operating systems that do not support Q35.

(BZ#1599777)

Post-copy virtual machine migration

RHEL 8 makes it possible to perform a post-copy migration of KVM virtual machines (VMs). When used, post-copy migration pauses the migrating VM's vCPUs on the source host, transfers only a minimum of memory pages, activates the VM's vCPUs on the destination host, and transfers the remaining memory pages while the VM is running on the destination.

This significantly reduces the downtime of the migrated VM, and also guarantees that the migration finishes regardless of how rapidly the memory pages of the source VM change. As such, it is optimal for migrating VMs in heavy continuous use, which would not be possible to migrate with the standard pre-copy migration.

(JIRA:RHELPLAN-14323)

virtio-gpu is now supported by KVM virtualization

The **virtio-gpu** display device has been introduced for KVM virtual machines (VMs). **virtio-gpu** improves VM graphical performance and also enables various enhancements for virtual GPU devices to be implemented in the future.

(JIRA:RHELPLAN-14329)

KVM supports UMIP in RHEL 8

KVM virtualization now supports the User-Mode Instruction Prevention (UMIP) feature, which can help prevent user-space applications from accessing to system-wide settings. This reduces the potential vectors for privilege escalation attacks, and thus makes the KVM hypervisor and its guest machines more secure.

(BZ#1494651)

Additional information in KVM guest crash reports

The crash information that KVM hypervisor generates if a guest terminates unexpectedly or becomes unresponsive has been expanded. This makes it easier to diagnose and fix problems in KVM virtualization deployments.

(BZ#1508139)

NVIDIA vGPU is now compatible with the VNC console

When using the NVIDIA virtual GPU (vGPU) feature, it is now possible to use the VNC console to display the visual output of the guest.

(BZ#1497911)

Ceph is supported by virtualization

With this update, Ceph storage is supported by KVM virtualization on all CPU architectures supported by Red Hat.

(BZ#1578855)

Interactive boot loader for KVM virtual machines on IBM Z

When booting a KVM virtual machine on an IBM Z host, the QEMU boot loader firmware can now present an interactive console interface of the guest OS. This makes it possible to troubleshoot guest OS boot problems without access to the host environment.

(BZ#1508137)

IBM z14 ZR1 supported in virtual machines

The KVM hypervisor now supports the CPU model of the IBM z14 ZR1 server. This enables using the features of this CPU in KVM virtual machines that run on an IBM Z system.

(BZ#1592337)

KVM supports Telnet 3270 on IBM Z

When using RHEL 8 as a host on an IBM Z system, it is now possible to connect to virtual machines on the host using **Telnet 3270** clients.

(BZ#1570029)

QEMU sandboxing has been added

In Red Hat Enterprise Linux 8, the QEMU emulator introduces the sandboxing feature. QEMU sandboxing provides configurable limitations to what systems calls QEMU can perform, and thus makes virtual machines more secure. Note that this feature is enabled and configured by default.

(JIRA:RHELPLAN-10628)

PV TLB Flush Hyper-V enlightenment

RHEL 8 adds the **PV TLB Flush** Hyper-V Enlightenment feature. This improves the performance of Windows virtual machines (VMs) that run in overcommitted environments on the KVM hypervisor.

(JIRA:RHELPLAN-14330)

New machine types for KVM virtual machines on IBM POWER

Multiple new rhel-pseries machine types have been enabled for KVM hypervisors running on IBM POWER 8 and IBM POWER 9 systems. This makes it possible for virtual machines (VMs) hosted on RHEL 8 on an IBM POWER system to correctly use the CPU features of these machine types. In addition, this allows for migrating VMs on IBM POWER to a more recent version of the KVM hypervisor.

(BZ#1585651, BZ#1595501)

GFNI and CLDEMOT instruction sets enabled for Intel Xeon SnowRidge

Virtual machines (VMs) running in a RHEL 8 host on an Intel Xeon SnowRidge system are now able to use the GFNI and CLDEMOT instruction sets. This may significantly increase the performance of such VMs in certain scenarios.

(BZ#1494705)

IPv6 enabled for OVMF

The IPv6 protocol is now enabled on Open Virtual Machine Firmware (OVMF). This makes it possible for virtual machines that use OVMF to take advantage of a variety of network boot improvements that IPv6 provides.

(BZ#1536627)

A VFIO-based block driver for NVMe devices has been added

The QEMU emulator introduces a driver based on virtual function I/O (VFIO) for Non-volatile Memory Express (NVMe) devices. The driver communicates directly with NVMe devices attached to virtual machines (VMs) and avoids using the kernel system layer and its NVMe drivers. As a result, this enhances the performance of NVMe devices in virtual machines.

(BZ#1519004)

Multichannel support for the Hyper-V Generic UIO driver

RHEL 8 now supports the multichannel feature for the Hyper-V Generic userspace I/O (UIO) driver. This makes it possible for RHEL 8 VMs running on the Hyper-V hypervisor to use the Data Plane

Development Kit (DPDK) Netvsc Poll Mode driver (PMD), which enhances the networking capabilities of these VMs.

Note, however, that the Netvsc interface status currently displays as Down even when it is running and usable.

(BZ#1650149)

Improved huge page support

When using RHEL 8 as a virtualization host, users can modify the size of pages that back memory of a virtual machine (VM) to any size that is supported by the CPU. This can significantly improve the performance of the VM.

To configure the size of VM memory pages, edit the VM's XML configuration and add the <hugepages> element to the <memoryBacking> section.

(JIRA:RHELPLAN-14607)

VMs on POWER 9 hosts can use THP

In RHEL 8 hosts running on the IBM POWER 9 architecture, virtual machines (VMs) benefit from the transparent huge pages (THP) feature. THP enables the host kernel to dynamically assign huge memory pages to processes and thus improves the performance of VMs with large amounts of memory.

(JIRA:RHELPLAN-13440)

5.1.17. Supportability

sosreport can report eBPF-based programs and maps

The **sosreport** tool has been enhanced to report any loaded extended Berkeley Packet Filtering (eBPF) programs and maps in Red Hat Enterprise Linux 8.

(BZ#1559836)

5.2. BUG FIXES

This part describes bugs fixed in Red Hat Enterprise Linux 8.0 that have a significant impact on users.

5.2.1. Desktop

PackageKit can now operate on rpm packages

With this update, the support for operating on **rpm** packages has been added into **PackageKit**.

(BZ#1559414)

5.2.2. Graphics infrastructures

QEMU does not handle 8-byte ggtt entries correctly

QEMU occasionally splits an 8-byte **ggtt** entry write to two consecutive 4-byte writes. Each of these partial writes can trigger a separate host **ggtt** write. Sometimes the two **ggtt** writes are combined incorrectly. Consequently, translation to a machine address fails, and an error log occurs.

(BZ#1598776)

5.2.3. Identity Management

The Enterprise Security Client uses the opensc library for token detection

Red Hat Enterprise Linux 8.0 only supports the **opensc** library for smart cards. With this update, the Enterprise Security Client (ESC) use **opensc** for token detection instead of the removed **coolkey** library. As a result, applications correctly detect supported tokens.

(BZ#1538645)

Certificate System now supports rotating debug logs

Previously, Certificate System used a custom logging framework, which did not support log rotation. As a consequence, debug logs such as **/var/log/pki/*instance_name*/ca/debug** grew indefinitely. With this update, Certificate System uses the **java.logging.util** framework, which supports log rotation. As a result, you can configure log rotation in the **/var/lib/pki/*instance_name*/conf/logging.properties** file.

For further information on log rotation, see documentation for the **java.util.logging** package.

(BZ#1565073)

Certificate System no longer logs SetAllPropertiesRule operation warnings when the service starts

Previously, Certificate System logged warnings on the **SetAllPropertiesRule** operation in the **/var/log/messages** log file when the service started. The problem has been fixed, and the mentioned warnings are no longer logged.

(BZ#1424966)

The Certificate System KRA client parses Key Request responses correctly

Previously, Certificate System switched to a new JSON library. As a consequence, serialization for certain objects differed, and the Python key recovery authority (KRA) client failed to parse **Key Request** responses. The client has been modified to support responses using both the old and the new JSON library. As a result, the Python KRA client parses **Key Request** responses correctly.

(BZ#1623444)

5.2.4. Compilers and development tools

GCC no longer produces false positive warnings about out-of-bounds access

Previously, when compiling with the **-O3** optimization level option, the GNU Compiler Collection (GCC) occasionally returned a false positive warning about an out-of-bounds access, even if the compiled code did not contain it. The optimization has been fixed and GCC no longer displays the false positive warning.

(BZ#1246444)

ltrace displays large structures correctly

Previously, the **ltrace** tool could not correctly print large structures returned from functions. Handling of large structures in **ltrace** has been improved and they are now printed correctly.

(BZ#1584322)

GCC built-in function __builtin_clz returns correct values on IBM Z

Previously, the **FLOGR** instruction of the IBM Z architecture was incorrectly folded by the GCC compiler. As a consequence, the **__builtin_clz** function using this instruction could return wrong results when the code was compiled with the **-funroll-loops** GCC option. This bug has been fixed and the function now provides correct results.

(BZ#1652016)

GDB provides nonzero exit status when last command in batch mode fails

Previously, GDB always exited with status **0** when running in batch mode, regardless of errors in the commands. As a consequence, it was not possible to determine whether the commands succeeded. This behavior has been changed and GDB now exits with status **1** when an error occurs in the last command. This preserves compatibility with the previous behavior where all commands are executed. As a result, it is now possible to determine if GDB batch mode execution is successful.

(BZ#1491128)

5.2.5. File systems and storage

Higher print levels no longer cause iscsiadm to terminate unexpectedly

Previously, the **iscsiadm** utility terminated unexpectedly when the user specified a print level higher than 0 with the **--print** or **-P** option. This problem has been fixed, and all print levels now work as expected.

(BZ#1582099)

multipathd no longer disables the path when it fails to get the WWID of a path

Previously, the **multipathd** service treated a failed attempt at getting a path's WWID as getting an empty WWID. If **multipathd** failed to get the WWID of a path, it sometimes disabled that path.

With this update, **multipathd** continues to use the old WWID if it fails to get the WWID when checking to see if it has changed.

As a result, **multipathd** no longer disables paths when it fails to get the WWID, when checking if the WWID has changed.

(BZ#1673167)

5.2.6. High availability and clusters

New /etc/sysconfig/pcsd option to reject client-initiated SSL/TLS renegotiation

When TLS renegotiation is enabled on the server, a client is allowed to send a renegotiation request, which initiates a new handshake. Computational requirements of a handshake are higher on a server than on a client. This makes the server vulnerable to DoS attacks. With this fix, the setting **PCSD_SSL_OPTIONS** in the **/etc/sysconfig/pcsd** configuration file accepts the **OP_NO_RENEGOTIATION** option to reject renegotiations. Note that the client can still open multiple connections to a server with a handshake performed in all of them.

(BZ#1566430)

A removed cluster node is no longer displayed in the cluster status

Previously, when a node was removed with the **pcs cluster node remove** command, the removed node remained visible in the output of a **pcs status** display. With this fix, the removed node is no longer displayed in the cluster status.

(BZ#1595829)

Fence agents can now be configured using either newer, preferred parameter names or deprecated parameter names

A large number of fence agent parameters have been renamed while the old parameter names are still supported as deprecated. Previously, **pcs** was not able to set the new parameters unless used with the **--force** option. With this fix, **pcs** now supports the renamed fence agent parameters while maintaining support for the deprecated parameters.

(BZ#1436217)

The pcs command now correctly reads the XML status of a cluster for display

The **pcs** command runs the **crm_mon** utility to get the status of a cluster in XML format. The **crm_mon** utility prints XML to standard output and warnings to standard error output. Previously **pcs** mixed XML and warnings into one stream and was then unable to parse it as XML. With this fix, standard and error outputs are separated in **pcs** and reading the XML status of a cluster works as expected.

(BZ#1578955)

Users no longer advised to destroy clusters when creating new clusters with nodes from existing clusters

Previously, when a user specified nodes from an existing cluster when running the **pcs cluster setup** command or when creating a cluster with the **pcsd** Web UI, pcs reported that as an error and suggested that the user destroy the cluster on the nodes. As a result, users would destroy the cluster on the nodes, breaking the cluster the nodes were part of as the remaining nodes would still consider the destroyed nodes to be part of the cluster. With this fix, users are instead advised to remove nodes from their cluster, better informing them of how to address the issue without breaking their clusters.

(BZ#1596050)

pcs commands no longer interactively ask for credentials

When a non-root user runs a **pcs** command that requires root permission, **pcs** connects to the locally running **pcsd** daemon and passes the command to it, since the **pcsd** daemon runs with root permissions and is capable of running the command. Previously, if the user was not authenticated to the local **pcsd** daemon, **pcs** asked for a user name and a password interactively. This was confusing to the user and required special handling in scripts running **pcs**. With this fix, if the user is not authenticated then **pcs** exits with an error advising what to do: Either run **pcs** as root or authenticate using the new **pcs client local-auth** command. As a result, **pcs** commands do not interactively ask for credentials, improving the user experience.

(BZ#1554310)

The pcsd daemon now starts with its default self-generated SSL certificate when crypto-policies is set to FUTURE.

A **crypto-policies** setting of **FUTURE** requires RSA keys in SSL certificates to be at least 3072b long. Previously, the **pcsd** daemon would not start when this policy was set since it generates SSL certificates with a 2048b key. With this update, the key size of **pcsd** self-generated SSL certificates has been increased to 3072b and **pcsd** now starts with its default self-generated SSL certificate.

(BZ#1638852)

The pcsd service now starts when the network is ready

Previously, When a user configured **pcsd** to bind to a specific IP address and the address was not ready during boot when **pcsd** attempted to start up, then **pcsd** failed to start and a manual intervention was required to start **pcsd**. With this fix, **pcsd.service** depends on **network-online.target**. As a result, **pcsd** starts when the network is ready and is able to bind to an IP address.

(BZ#1640477)

5.2.7. Networking

Weak TLS algorithms are no longer allowed for glib-networking

Previously, the **glib-networking** package was not compatible with RHEL 8 System-wide Crypto Policy. As a consequence, applications using the **glib** library for networking might allow Transport Layer Security (TLS) connections using weak algorithms than the administrator intended. With this update, the system-wide crypto policy is applied, and now applications using **glib** for networking allow only TLS connections that are acceptable according to the policy.

(BZ#1640534)

5.2.8. Security

SELinux policy now allows iscsiuio processes to connect to the discovery portal

Previously, SELinux policy was too restrictive for **iscsiuio** processes and these processes were not able to access **/dev/uio*** devices using the **mmap** system call. As a consequence, connection to the discovery portal failed. This update adds the missing rules to the SELinux policy and **iscsiuio** processes work as expected in the described scenario.

(BZ#1626446)

5.2.9. Subscription management

dnf and yum can now access the repos regardless of subscription-manager values

Previously, the **dnf** or **yum** commands ignored the **https://** prefix from a URL added by the **subscription-manager** service. The updated **dnf** or **yum** commands do not ignore invalid **https://** URLs. As a consequence, **dnf** and **yum** failed to access the repos. To fix the problem, a new configuration variable, **proxy_scheme** has been added to the **/etc/rhsm/rhsm.conf** file and the value can be set to either **http** or **https**. If no value is specified, **subscription-manager** set **http** by default which is more commonly used.

Note that if the proxy uses **http**, most users should not change anything in the configuration in **/etc/rhsm/rhsm.conf**. If the proxy uses **https**, users should update the value of **proxy_scheme** to **https**. Then, in both cases, users need to run the **subscription-manager repos --list** command or wait for the **rhsmcertd** daemon process to regenerate the **/etc/yum.repos.d/redhat.repo** properly.

(BZ#1654531)

5.2.10. Virtualization

Mounting ephemeral disks on Azure now works more reliably

Previously, mounting an ephemeral disk on a virtual machine (VM) running on the Microsoft Azure platform failed if the VM was "stopped(deallocated)" and then started. This update ensures that reconnecting disks is handled correctly in the described circumstances, which prevents the problem from occurring.

(BZ#1615599)

5.3. TECHNOLOGY PREVIEWS

This part provides a list of all Technology Previews available in Red Hat Enterprise Linux 8.0.

For information on Red Hat scope of support for Technology Preview features, see Technology Preview Features Support Scope.

5.3.1. Kernel

eBPF available as a Technology Preview

The **extended Berkeley Packet Filtering (eBPF)** feature is available as a Technology Preview for both networking and tracing. **eBPF** enables the user space to attach custom programs onto a variety of points (sockets, trace points, packet reception) to receive and process data. The feature includes a new system call **bpf()**, which supports creating various types of maps, and also to insert various types of programs into the kernel. Note that the **bpf()** syscall can be successfully used only by a user with the **CAP_SYS_ADMIN** capability, such as a root user. See the **bpf**(2) man page for more information.

(BZ#1559616)

BCC is available as a Technology Preview

BPF Compiler Collection (BCC) is a user space tool kit for creating efficient kernel tracing and manipulation programs that is available as a Technology Preview in Red Hat Enterprise Linux 8. **BCC** provides tools for I/O analysis, networking, and monitoring of Linux operating systems using the **extended Berkeley Packet Filtering (eBPF)**.

(BZ#1548302)

Control Group v2 available as a Technology Preview in RHEL 8

Control Group v2 mechanism is a unified hierarchy control group. **Control Group v2** organizes processes hierarchically and distributes system resources along the hierarchy in a controlled and configurable manner.

Unlike the previous version, **Control Group v2** has only a single hierarchy. This single hierarchy enables the Linux kernel to:

- Categorize processes based on the role of their owner.
- Eliminate issues with conflicting policies of multiple hierarchies.

Control Group v2 supports numerous controllers:

- CPU controller regulates the distribution of CPU cycles. This controller implements:
  - Weight and absolute bandwidth limit models for normal scheduling policy.
  - Absolute bandwidth allocation model for real time scheduling policy.

- Memory controller regulates the memory distribution. Currently, the following types of memory usages are tracked:
  - Userland memory - page cache and anonymous memory.
  - Kernel data structures such as dentries and inodes.
  - TCP socket buffers.
- I/O controller regulates the distribution of I/O resources.
- Writeback controller interacts with both Memory and I/O controllers and is **Control Group v2** specific.

The information above was based on link: https://www.kernel.org/doc/Documentation/cgroup-v2.txt. You can refer to the same link to obtain more information about particular **Control Group v2** controllers.

(BZ#1401552)

early kdump available as a Technology Preview in Red Hat Enterprise Linux 8

The **early kdump** feature allows the crash kernel and initramfs to load early enough to capture the **vmcore** information even for early crashes. For more details about **early kdump**, see the **/usr/share/doc/kexec-tools/early-kdump-howto.txt** file.

(BZ#1520209)

The ibmvnic device driver available as a Technology Preview

With Red Hat Enterprise Linux 8.0, the IBM Virtual Network Interface Controller (vNIC) driver for IBM POWER architectures, **ibmvnic**, is available as a Technology Preview. vNIC is a PowerVM virtual networking technology that delivers enterprise capabilities and simplifies network management. It is a high-performance, efficient technology that when combined with SR-IOV NIC provides bandwidth control Quality of Service (QoS) capabilities at the virtual NIC level. vNIC significantly reduces virtualization overhead, resulting in lower latencies and fewer server resources, including CPU and memory, required for network virtualization.

(BZ#1524683)

Soft-RoCE available as a Technology Preview

Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) is a network protocol which implements RDMA over Ethernet. Soft-RoCE is the software implementation of RoCE which supports two protocol versions, RoCE v1 and RoCE v2. The Soft-RoCE driver, **rdma_rxe**, is available as an unsupported Technology Preview in RHEL 8.

(BZ#1605216)

5.3.2. Graphics infrastructures

VNC remote console available as a Technology Preview for the 64-bit ARM architecture

On the 64-bit ARM architecture, the Virtual Network Computing (VNC) remote console is available as a Technology Preview. Note that the rest of the graphics stack is currently unverified for the 64-bit ARM architecture.

(BZ#1698565)

5.3.3. Hardware enablement

The cluster-aware MD RAID1 is available as a technology preview.

RAID1 cluster is not enabled by default in the kernel space. If you want to have a try with RAID1 cluster, you need to build the kernel with RAID1 cluster as a module first, use the following steps:

1. Enter the **make menuconfig** command.
2. Enter the **make && make modules && make modules_install && make install** command.
3. Enter the **reboot** command.

(BZ#1654482)

5.3.4. Identity Management

DNSSEC available as Technology Preview in IdM

Identity Management (IdM) servers with integrated DNS now support DNS Security Extensions (DNSSEC), a set of extensions to DNS that enhance security of the DNS protocol. DNS zones hosted on IdM servers can be automatically signed using DNSSEC. The cryptographic keys are automatically generated and rotated.

Users who decide to secure their DNS zones with DNSSEC are advised to read and follow these documents:

- DNSSEC Operational Practices, Version 2: http://tools.ietf.org/html/rfc6781#section-2
- Secure Domain Name System (DNS) Deployment Guide: http://dx.doi.org/10.6028/NIST.SP.800-81-2
- DNSSEC Key Rollover Timing Considerations: http://tools.ietf.org/html/rfc7583

Note that IdM servers with integrated DNS use DNSSEC to validate DNS answers obtained from other DNS servers. This might affect the availability of DNS zones that are not configured in accordance with recommended naming practices.

(BZ#1664718)

Identity Management JSON-RPC API available as Technology Preview

An API is available for Identity Management (IdM). To view the API, IdM also provides an API browser as Technology Preview.

In Red Hat Enterprise Linux 7.3, the IdM API was enhanced to enable multiple versions of API commands. Previously, enhancements could change the behavior of a command in an incompatible way. Users are now able to continue using existing tools and scripts even if the IdM API changes. This enables:

- Administrators to use previous or later versions of IdM on the server than on the managing client.
- Developers to use a specific version of an IdM call, even if the IdM version changes on the server.

In all cases, the communication with the server is possible, regardless if one side uses, for example, a newer version that introduces new options for a feature.

For details on using the API, see Using the Identity Management API to Communicate with the IdM Server (TECHNOLOGY PREVIEW).

(BZ#1664719)

5.3.5. File systems and storage

Aero adapters available as a Technology Preview

The following Aero adapters are available as a Technology Preview:

- PCI ID 0x1000:0x00e2 and 0x1000:0x00e6, controlled by the **mpt3sas** driver
- PCI ID 0x1000:Ox10e5 and 0x1000:0x10e6, controlled by the **megaraid_sas** driver

(BZ#1663281)

Stratis is now available

Stratis is a new local storage manager. It provides managed file systems on top of pools of storage with additional features to the user.

Stratis enables you to more easily perform storage tasks such as:

- Manage snapshots and thin provisioning
- Automatically grow file system sizes as needed
- Maintain file systems

To administer Stratis storage, use the **stratis** utility, which communicates with the **stratisd** background service.

Stratis is provided as a Technology Preview.

For more information, see the Stratis documentation: Setting up Stratis file systems.

(JIRA:RHELPLAN-1212)

OverlayFS

OverlayFS is a type of union file system. It enables you to overlay one file system on top of another. Changes are recorded in the upper file system, while the lower file system remains unmodified. This allows multiple users to share a file-system image, such as a container or a DVD-ROM, where the base image is on read-only media.

OverlayFS remains a Technology Preview under most circumstances. As such, the kernel logs warnings when this technology is activated.

Full support is available for OverlayFS when used with supported container engines (**podman**, **cri-o**, or **buildah**) under the following restrictions:

- OverlayFS is supported for use only as a container engine graph driver or other specialized use cases, such as squashed **kdump** initramfs. Its use is supported primarily for container COW content, not for persistent storage. You must place any persistent storage on non-OverlayFS volumes. You can use only the default container engine configuration: one level of overlay, one lowerdir, and both lower and upper levels are on the same file system.

- Only XFS is currently supported for use as a lower layer file system.

Additionally, the following rules and limitations apply to using OverlayFS:

- The OverlayFS kernel ABI and user-space behavior are not considered stable, and might change in future updates.
- OverlayFS provides a restricted set of the POSIX standards. Test your application thoroughly before deploying it with OverlayFS. The following cases are not POSIX-compliant:
  - Lower files opened with **O_RDONLY** do not receive **st_atime** updates when the files are read.
  - Lower files opened with **O_RDONLY**, then mapped with **MAP_SHARED** are inconsistent with subsequent modification.
  - Fully compliant **st_ino** or **d_ino** values are not enabled by default on RHEL 8, but you can enable full POSIX compliance for them with a module option or mount option.
    To get consistent inode numbering, use the **xino=on** mount option.

    You can also use the **redirect_dir=on** and **index=on** options to improve POSIX compliance. These two options make the format of the upper layer incompatible with an overlay without these options. That is, you might get unexpected results or errors if you create an overlay with **redirect_dir=on** or **index=on**, unmount the overlay, then mount the overlay without these options.
- To determine whether an existing XFS file system is eligible for use as an overlay, use the following command and see if the **ftype=1** option is enabled:

  ```
  # xfs_info /mount-point | grep ftype
  ```
- SELinux security labels are enabled by default in all supported container engines with OverlayFS.
- Several known issues are associated with OverlayFS in this release. For details, see *Non-standard behavior* in the Linux kernel documentation.

For more information about OverlayFS, see the Linux kernel documentation.

(BZ#1690207)

File system DAX is now available for ext4 and XFS as a Technology Preview

In Red Hat Enterprise Linux 8.0, file system DAX is available as a Technology Preview. DAX provides a means for an application to directly map persistent memory into its address space. To use DAX, a system must have some form of persistent memory available, usually in the form of one or more Non-Volatile Dual In-line Memory Modules (NVDIMMs), and a file system that supports DAX must be created on the NVDIMM(s). Also, the file system must be mounted with the **dax** mount option. Then, an **mmap** of a file on the dax-mounted file system results in a direct mapping of storage into the application's address space.

(BZ#1627455)

5.3.6. High availability and clusters

Pacemaker **podman** bundles available as a Technology Preview

Pacemaker container bundles now run on the **podman** container platform, with the container bundle feature being available as a Technology Preview. There is one exception to this feature being Technology Preview: Red Hat fully supports the use of Pacemaker bundles for Red Hat Openstack.

(BZ#1619620)

5.3.7. Networking

XDP available as a Technology Preview

The eXpress Data Path (XDP) feature, which is available as a Technology Preview, provides a means to attach extended Berkeley Packet Filter (eBPF) programs for high-performance packet processing at an early point in the kernel ingress data path, allowing efficient programmable packet analysis, filtering, and manipulation.

(BZ#1503672)

eBPF for tc available as a Technology Preview

As a Technology Preview, the Traffic Control (tc) kernel subsystem and the **tc** tool can attach extended Berkeley Packet Filtering (eBPF) programs as packet classifiers and actions for both ingress and egress queueing disciplines. This enables programmable packet processing inside the kernel network data path.

(BZ#1699825)

AF_XDP available as a Technology Preview

Address Family eXpress Data Path (**AF_XDP**) socket is designed for high-performance packet processing. It accompanies **XDP** and grants efficient redirection of programmatically selected packets to user space applications for further processing.

(BZ#1633143)

KTLS available as a Technology Preview

In Red Hat Enterprise Linux 8, Kernel Transport Layer Security (KTLS) is provided as a Technology Preview. KTLS handles TLS records using the symmetric encryption or decryption algorithms in the kernel for the AES-GCM cipher. KTLS also provides the interface for offloading TLS record encryption to Network Interface Controllers (NICs) that support this functionality.

(BZ#1570255)

TIPC available as a Technology Preview

The Transparent Inter Process Communication (**TIPC**) is a protocol specially designed for efficient communication within clusters of loosely paired nodes. It works as a kernel module and provides a **tipc** tool in **iproute2** package to allow designers to create applications that can communicate quickly and reliably with other applications regardless of their location within the cluster. This feature is available as a Technology Preview.

(BZ#1581898)

The systemd-resolved service is now available as a Technology Preview

The **systemd-resolved** service provides name resolution to local applications. The service implements a caching and validating DNS stub resolver, an Link-Local Multicast Name Resolution (LLMNR), and Multicast DNS resolver and responder.

Note that, even if the **systemd** package provides **systemd-resolved**, this service is an unsupported Technology Preview.

(BZ#1906489)

5.3.8. Red Hat Enterprise Linux System Roles

The postfix role of RHEL System Roles available as a Technology Preview

Red Hat Enterprise Linux System Roles provides a configuration interface for Red Hat Enterprise Linux subsystems, which makes system configuration easier through the inclusion of Ansible Roles. This interface enables managing system configurations across multiple versions of Red Hat Enterprise Linux, as well as adopting new major releases.

The **rhel-system-roles** packages are distributed through the AppStream repository.

The **postfix** role is available as a Technology Preview.

The following roles are fully supported:

- **kdump**
- **network**
- **selinux**
- **timesync**

For more information, see the Knowledgebase article about RHEL System Roles.

(BZ#1812552)

5.3.9. Virtualization

AMD SEV for KVM virtual machines

As a Technology Preview, RHEL 8 introduces the Secure Encrypted Virtualization (SEV) feature for AMD EPYC host machines that use the KVM hypervisor. If enabled on a virtual machine (VM), SEV encrypts VM memory so that the host cannot access data on the VM. This increases the security of the VM if the host is successfully infected by malware.

Note that the number of VMs that can use this feature at a time on a single host is determined by the host hardware. Current AMD EPYC processors support up to 15 running VMs using SEV.

Also note that for VMs with SEV configured to be able to boot, you must also configure the VM with a hard memory limit. To do so, add the following to the VM's XML configuration:

```
<memtune>
  <hard_limit unit='KiB'>N</hard_limit>
</memtune>
```

The recommended value for N is equal to or greater then the guest RAM + 256 MiB. For example, if the guest is assigned 2 GiB RAM, N should be 2359296 or greater.

(BZ#1501618, BZ#1501607)

Intel vGPU

As a Technology Preview, it is now possible to divide a physical Intel GPU device into multiple virtual devices referred to as **mediated devices**. These mediated devices can then be assigned to multiple virtual machines (VMs) as virtual GPUs. As a result, these VMs share the performance of a single physical Intel GPU.

Note that only selected Intel GPUs are compatible with the vGPU feature. In addition, assigning a physical GPU to VMs makes it impossible for the host to use the GPU, and may prevent graphical display output on the host from working.

(BZ#1528684)

Nested virtualization now available on IBM POWER 9

As a Technology Preview, it is now possible to use the nested virtualization features on RHEL 8 host machines running on IBM POWER 9 systems. Nested virtualization enables KVM virtual machines (VMs) to act as hypervisors, which allows for running VMs inside VMs.

Note that nested virtualization also remains a Technology Preview on AMD64 and Intel 64 systems.

Also note that for nested virtualization to work on IBM POWER 9, the host, the guest, and the nested guests currently all need to run one of the following operating systems:

- RHEL 8
- RHEL 7 for POWER 9

(BZ#1505999, BZ#1518937)

KVM virtualization is usable in RHEL 8 Hyper-V virtual machines

As a Technology Preview, nested KVM virtualization can now be used on the Microsoft Hyper-V hypervisor. As a result, you can create virtual machines on a RHEL 8 guest system running on a Hyper-V host.

Note that currently, this feature only works on Intel systems. In addition, nested virtualization is in some cases not enabled by default on Hyper-V. To enable it, see the following Microsoft documentation:

https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/nested-virtualization

(BZ#1519039)

5.3.10. Containers

The **podman-machine** command is unsupported

The **podman-machine** command for managing virtual machines, is available only as a Technology Preview. Instead, run Podman directly from the command line.

(JIRA:RHELDOCS-16861)

5.4. DEPRECATED FUNCTIONALITY

This part provides an overview of functionality that has been *deprecated* in Red Hat Enterprise Linux 8.0.

Deprecated functionality will likely not be supported in future major releases of this product and is not recommended for new deployments. For the most recent list of deprecated functionality within a particular major release, refer to the latest version of release documentation.

The support status of deprecated functionality remains unchanged within Red Hat Enterprise Linux 8. For information about the length of support, see Red Hat Enterprise Linux Life Cycle and Red Hat Enterprise Linux Application Streams Life Cycle.

Deprecated hardware components are not recommended for new deployments on the current or future major releases. Hardware driver updates are limited to security and critical fixes only. Red Hat recommends replacing this hardware as soon as reasonably feasible.

A package can be deprecated and not recommended for further use. Under certain circumstances, a package can be removed from a product. Product documentation then identifies more recent packages that offer functionality similar, identical, or more advanced to the one deprecated, and provides further recommendations.

For information regarding functionality that is present in RHEL 7 but has been *removed* in RHEL 8, see Considerations in adopting RHEL 8 .

5.4.1. Installer and image creation

The --interactive option of the ignoredisk Kickstart command has been deprecated

Using the **--interactive option** in future releases of Red Hat Enterprise Linux will result in a fatal installation error. It is recommended that you modify your Kickstart file to remove the option.

(BZ#1637872)

Several Kickstart commands and options have been deprecated

Using the following commands and options in RHEL 8 Kickstart files will print a warning in the logs.

- **auth** or **authconfig**
- **device**
- **deviceprobe**
- **dmraid**
- **install**
- **lilo**
- **lilocheck**
- **mouse**
- **multipath**
- **bootloader --upgrade**
- **ignoredisk --interactive**
- **partition --active**
- **reboot --kexec**

Where only specific options are listed, the base command and its other options are still available and not deprecated.

For more details and related changes in Kickstart, see the Kickstart changes section of the *Considerations in adopting RHEL 8* document.

(BZ#1642765)

5.4.2. File systems and storage

NFSv3 over UDP has been disabled

The NFS server no longer opens or listens on a User Datagram Protocol (UDP) socket by default. This change affects only NFS version 3 because version 4 requires the Transmission Control Protocol (TCP).

NFS over UDP is no longer supported in RHEL 8.

(BZ#1592011)

The elevator kernel command line parameter is deprecated

The **elevator** kernel command line parameter was used in earlier RHEL releases to set the disk scheduler for all devices. In RHEL 8, the parameter is deprecated.

The upstream Linux kernel has removed support for the **elevator** parameter, but it is still available in RHEL 8 for compatibility reasons.

Note that the kernel selects a default disk scheduler based on the type of device. This is typically the optimal setting. If you require a different scheduler, Red Hat recommends that you use **udev** rules or the Tuned service to configure it. Match the selected devices and switch the scheduler only for those devices.

For more information, see the following article: Why does the 'elevator=' parameter no longer work in RHEL8.

(BZ#1665295)

The VDO Ansible module in VDO packages

The VDO Ansible module is currently provided by the **vdo** RPM package. In a future release, the VDO Ansible module will be moved to the Ansible RPM packages.

(BZ#1669537)

5.4.3. Networking

Network scripts are deprecated in RHEL 8

Network scripts are deprecated in Red Hat Enterprise Linux 8 and they are no longer provided by default. The basic installation provides a new version of the **ifup** and **ifdown** scripts which call the **NetworkManager** service through the **nmcli** tool. In Red Hat Enterprise Linux 8, to run the **ifup** and the **ifdown** scripts, NetworkManager must be running.

Note that custom commands in **/sbin/ifup-local**, **ifdown-pre-local** and **ifdown-local** scripts are not executed.

If any of these scripts are required, the installation of the deprecated network scripts in the system is still possible with the following command:

```
~]# yum install network-scripts
```

The **ifup** and **ifdown** scripts link to the installed legacy network scripts.

Calling the legacy network scripts shows a warning about their deprecation.

(BZ#1647725)

5.4.4. Kernel

The rdma_rxe Soft-RoCE driver is deprecated

Software Remote Direct Memory Access over Converged Ethernet (Soft-RoCE), also known as RXE, is a feature that emulates Remote Direct Memory Access (RDMA). In RHEL 8, the Soft-RoCE feature is available as an unsupported Technology Preview. However, due to stability issues, this feature has been deprecated and will be removed in RHEL 9.

(BZ#1878207)

5.4.5. Security

DSA is deprecated in RHEL 8

The Digital Signature Algorithm (DSA) is considered deprecated in Red Hat Enterprise Linux 8. Authentication mechanisms that depend on DSA keys do not work in the default configuration. Note that **OpenSSH** clients do not accept DSA host keys even in the **LEGACY** system-wide cryptographic policy level.

(BZ#1646541)

SSL2 Client Hello has been deprecated in NSS

The Transport Layer Security (**TLS**) protocol version 1.2 and earlier allow to start a negotiation with a **Client Hello** message formatted in a way that is backward compatible with the Secure Sockets Layer (**SSL**) protocol version 2. Support for this feature in the Network Security Services (**NSS**) library has been deprecated and it is disabled by default.

Applications that require support for this feature need to use the new **SSL_ENABLE_V2_COMPATIBLE_HELLO** API to enable it. Support for this feature may be removed completely in future releases of Red Hat Enterprise Linux 8.

(BZ#1645153)

TLS 1.0 and TLS 1.1 are deprecated

The TLS 1.0 and TLS 1.1 protocols are disabled in the **DEFAULT** system-wide cryptographic policy level. If your scenario, for example, a video conferencing application in the Firefox web browser, requires using the deprecated protocols, switch the system-wide cryptographic policy to the **LEGACY** level:

```
# update-crypto-policies --set LEGACY
```

For more information, see the Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms Knowledgebase article on the Red Hat Customer Portal and the **update-crypto-policies(8)** man page.

(BZ#1660839)

5.4.6. Virtualization

Virtual machine snapshots are not properly supported in RHEL 8

The current mechanism of creating virtual machine (VM) snapshots has been deprecated, as it is not working reliably. As a consequence, it is recommended not to use VM snapshots in RHEL 8.

Note that a new VM snapshot mechanism is under development and will be fully implemented in a future minor release of RHEL 8.

(BZ#1686057)

The Cirrus VGA virtual GPU type has been deprecated

With a future major update of Red Hat Enterprise Linux, the **Cirrus VGA** GPU device will no longer be supported in KVM virtual machines. Therefore, Red Hat recommends using the **stdvga**, **virtio-vga**, or **qxl** devices instead of Cirrus VGA.

(BZ#1651994)

virt-manager has been deprecated

The Virtual Machine Manager application, also known as **virt-manager**, has been deprecated. The RHEL 8 web console, also known as **Cockpit**, is intended to become its replacement in a subsequent release. It is, therefore, recommended that you use the web console for managing virtualization in a GUI. However, in Red Hat Enterprise Linux 8.0, some features may only be accessible from either **virt-manager** or the command line.

(JIRA:RHELPLAN-10304)

5.4.7. Deprecated packages

The following packages have been deprecated and will probably not be included in a future major release of Red Hat Enterprise Linux:

- 389-ds-base-legacy-tools
- authd
- custodia
- hostname
- libidn
- net-tools
- network-scripts
- nss-pam-ldapd
- sendmail

- yp-tools
- ypbind
- ypserv

5.5. KNOWN ISSUES

This part describes known issues in Red Hat Enterprise Linux 8.

5.5.1. The web console

Logging to RHEL web console with session_recording shell is not possible

Currently, the RHEL web console logins will fail for tlog recording-enabled users. RHEL web console requires a user's shell to be present in the **/etc/shells** directory to allow a successful login. However, if **tlog-rec-session** is added to **/etc/shells**, a recorded user is able to disable recording by changing the shell from **tlog-rec-session** to another shell from **/etc/shells**, using the "chsh" utility. Red Hat does not recommend adding **tlog-rec-session** to **/etc/shells** for this reason.

(BZ#1631905)

5.5.2. Installer and image creation

The auth and authconfig Kickstart commands require the AppStream repository

The **authselect-compat** package is required by the **auth** and **authconfig** Kickstart commands during installation. Without this package, the installation fails if **auth** or **authconfig** are used. However, by design, the **authselect-compat** package is only available in the AppStream repository.

To work around this problem, verify that the BaseOS and AppStream repositories are available to the installer or use the **authselect** Kickstart command during installation.

(BZ#1640697)

The xorg-x11-drv-fbdev, xorg-x11-drv-vesa, and xorg-x11-drv-vmware video drivers are not installed by default

Workstations with specific models of NVIDIA graphics cards and workstations with specific AMD accelerated processing units will not display the graphical login window after a RHEL 8.0 Server installation.

To work around this problem, perform a RHEL 8.0 **Workstation** installation on a workstation machine. If a RHEL 8.0 **Server** installation is required on the workstation, manually install the **base-x** package group after installation by running the **yum -y groupinstall base-x** command.

In addition, virtual machines relying on EFI for graphics support, such as Hyper-V, are also affected. If you selected the **Server with GUI** base environment on Hyper-V, you might be unable to log in due to a black screen displayed on reboot. To work around this problem on Hyper-v, enable multi- or single-user mode using the following steps:

1. Reboot the virtual machine.
2. During the booting process, select the required kernel using the up and down arrow keys on your keyboard.

3. Press the **e** key on your keyboard to edit the kernel command line.
4. Add **systemd.unit=multi-user.target** to the kernel command line in GRUB.
5. Press **Ctrl-X** to start the virtual machine.
6. After logging in, run the **yum -y groupinstall base-x** command.
7. Reboot the virtual machine to access the graphical mode.

(BZ#1687489)

Installation fails when using the reboot --kexec command

The RHEL 8 installation fails when using a Kickstart file that contains the **reboot --kexec** command. To avoid the problem, use the **reboot** command instead of **reboot --kexec** in your Kickstart file.

(BZ#1672405)

Copying the content of the Binary DVD.iso file to a partition omits the .treeinfo and .discinfo files

During local installation, while copying the content of the RHEL 8 Binary DVD.iso image file to a partition, the * in the **cp <path>/* <mounted partition>/dir** command fails to copy the **.treeinfo** and **.discinfo** files. These files are required for a successful installation. As a result, the BaseOS and AppStream repositories are not loaded, and a debug-related log message in the **anaconda.log** file is the only record of the problem.

To work around the problem, copy the missing **.treeinfo** and **.discinfo** files to the partition.

(BZ#1692746)

Anaconda installation includes low limits of minimal resources setting requirements

Anaconda initiates the installation on systems with minimal resource settings required available and do not provide previous message warning about the required resources for performing the installation successfully. As a result, the installation can fail and the output errors do not provide clear messages for possible debug and recovery. To work around this problem, make sure that the system has the minimal resources settings required for installation: 2GB memory on PPC64(LE) and 1GB on x86_64. As a result, it should be possible to perform a successful installation.

(BZ#1696609)

The reboot --kexec and inst.kexec commands do not provide a predictable system state

Performing a RHEL installation with the **reboot --kexec** Kickstart command or the **inst.kexec** kernel boot parameters do not provide the same predictable system state as a full reboot. As a consequence, switching to the installed system without rebooting can produce unpredictable results.

Note that the **kexec** feature is deprecated and will be removed in a future release of Red Hat Enterprise Linux.

(BZ#1697896)

5.5.3. Kernel

The i40iw module does not load automatically on boot

Due to many i40e NICs not supporting iWarp and the **i40iw** module not fully supporting suspend/resume, this module is not automatically loaded by default to ensure suspend/resume works properly. To work around this problem, manually edit the **/lib/udev/rules.d/90-rdma-hw-modules.rules** file to enable automated load of **i40iw**.

Also note that if there is another RDMA device installed with a i40e device on the same machine, the non-i40e RDMA device triggers the **rdma** service, which loads all enabled RDMA stack modules, including the **i40iw** module.

(BZ#1623712)

The system sometimes becomes unresponsive when many devices are connected

When Red Hat Enterprise Linux 8 configures a large number of devices, a large number of console messages occurs on the system console. This happens, for example, when there are a large number of logical unit numbers (LUNs), with multiple paths to each LUN. The flood of console messages, in addition to other work the kernel is doing, might cause the kernel watchdog to force a kernel panic because the kernel appears to be hung.

Because the scan happens early in the boot cycle, the system becomes unresponsive when many devices are connected. This typically occurs at boot time.

If **kdump** is enabled on your machine during the device scan event after boot, the hard lockup results in a capture of a **vmcore** image.

To work around this problem, increase the watchdog lockup timer. To do so, add the **watchdog_thresh=*N*** option to the kernel command line. Replace ***N*** with the number of seconds:

- If you have less than a thousand devices, use **30**.
- If you have more than a thousand devices, use **60**.

For storage, the number of device is the number of paths to all the LUNs: generally, the number of **/dev/sd*** devices.

After applying the workaround, the system no longer becomes unresponsive when configuring a large amount of devices.

(BZ#1598448)

KSM sometimes ignores NUMA memory policies

When the kernel shared memory (KSM) feature is enabled with the **merge_across_nodes=1** parameter, KSM ignores memory policies set by the mbind() function, and may merge pages from some memory areas to Non-Uniform Memory Access (NUMA) nodes that do not match the policies.

To work around this problem, disable KSM or set the **merge_across_nodes** parameter to **0** if using NUMA memory binding with QEMU. As a result, NUMA memory policies configured for the KVM VM will work as expected.

(BZ#1153521)

The qede driver hangs the NIC and makes it unusable

Due to a bug, the **qede** driver for the 41000 and 45000 QLogic series NICs can cause Firmware upgrade and debug data collection operations to fail and make the NIC unusable or in hung state until reboot (PCI reset) of the host makes the NIC operational again.

This issue has been detected in all of the following scenarios:

- when upgrading Firmware of the NIC using the inbox driver
- when collecting debug data running the **ethtool -d ethx** command
- running the **sosreport** command as it includes **ethtool -d ethx.**
- when the inbox driver initiates automatic debug data collection, such as IO timeout, Mail Box Command timeout and a Hardware Attention.

A future erratum from Red Hat will be released via Red Hat Bug Advisory (RHBA) to address this issue. To work around this problem, create a case in https://access.redhat.com/support to request a supported fix for the issue until the RHBA is released.

(BZ#1697310)

Radix tree symbols were added to kernel-abi-whitelists

The following radix tree symbols have been added to the **kernel-abi-whitelists** package in Red Hat Enterprise Linux 8:

- **__radix_tree_insert**
- **__radix_tree_next_slot**
- **radix_tree_delete**
- **radix_tree_gang_lookup**
- **radix_tree_gang_lookup_tag**
- **radix_tree_next_chunk**
- **radix_tree_preload**
- **radix_tree_tag_set**

The symbols above were not supposed to be present and will be removed from the RHEL8 whitelist.

(BZ#1695142)

podman fails to checkpoint a container in RHEL 8

The version of the Checkpoint and Restore In Userspace (CRIU) package is outdated in Red Hat Enterprise Linux 8. As a consequence, CRIU does not support container checkpoint and restore functionality and the **podman** utility fails to checkpoint containers. When running the **podman container checkpoint** command, the following error message is displayed: 'checkpointing a container requires at least CRIU 31100'

(BZ#1689746)

early-kdump and standard kdump fail if the add_dracutmodules+=earlykdump option is used in dracut.conf

Currently, an inconsistency occurs between the kernel version being installed for **early-kdump** and the kernel version **initramfs** is generated for. As a consequence, booting with **early-kdump** enabled, **early-kdump** fails. In addition, if **early-kdump** detects that it is being included in a standard **kdump** initramfs

image, it forces an exit. Therefore the standard **kdump** service also fails when trying to rebuild **kdump** initramfs if **early-kdump** is added as a default **dracut** module. As a consequence, **early-kdump** and standard **kdump** both fail. To work around this problem, do not add **add_dracutmodules+=earlykdump** or any equivalent configuration in the **dracut.conf** file. As a result, **early-kdump** is not included by **dracut** by default, which prevents the problem from occuring. However, if an **early-kdump** image is required, it has to be created manually.

(BZ#1662911)

Debug kernel fails to boot in crash capture environment in RHEL 8

Due to memory-demanding nature of the debug kernel, a problem occurs when the debug kernel is in use and a kernel panic is triggered. As a consequence, the debug kernel is not able to boot as the capture kernel, and a stack trace is generated instead. To work around this problem, increase the crash kernel memory accordingly. As a result, the debug kernel successfully boots in the crash capture environment.

(BZ#1659609)

Network interface is renamed to kdump-<interface-name> when fadump is used

When firmware-assisted dump (**fadump**) is utilized to capture a vmcore and store it to a remote machine using SSH or NFS protocol, the network interface is renamed to **kdump-<interface-name>** if **<interface-name>** is generic, for example, *eth#, or net#. This problem occurs because the vmcore capture scripts in the initial RAM disk (**initrd**) add the kdump- prefix to the network interface name to secure persistent naming. The same **initrd** is used also for a regular boot, so the interface name is changed for the production kernel too.

(BZ#1745507)

5.5.4. Software management

Running yum list under a non-root user causes YUM crash

When running the **yum list** command under a non-root user after the **libdnf** package has been updated, **YUM** can terminate unexpectedly. If you hit this bug, run **yum list** under root to resolve the problem. As a result, subsequent attempts to run **yum list** under a non-root user no longer cause **YUM** crash.

(BZ#1642458)

YUM v4 skips unavailable repositories by default

YUM v4 defaults to the "skip_if_unavailable=True" setting for all repositories. As a consequence, if the required repository is not available, the packages from the repository are not considered in the install, search, or update operations. Subsequently, some **yum** commands and yum-based scripts succeed with exit code 0 even if there are unavailable repositories.

Currently, there is no other workaround available than updating the **libdnf** package.

(BZ#1679509)

5.5.5. Infrastructure services

The nslookup and host utilities ignore replies from name servers with recursion not available

If more name servers are configured and recursion is not available for a name server, the **nslookup** and

host utilities ignore replies from such name server unless it is the one that is last configured. In case of the last configured name server, answer is accepted even without the **recursion available** flag. However, if the last configured name server is not responding or unreachable, name resolution fails.

To work around the problem:

- Ensure that configured name servers always reply with the **recursion available** flag set.
- Allow recursion for all internal clients.

To troubleshoot the problem, you can also use the **dig** utility to detect whether recursion is available or not.

(BZ#1599459)

5.5.6. Shells and command-line tools

Python binding of the net-snmp package is unavailable

The **Net-SNMP** suite of tools does not provide binding for **Python 3**, which is the default **Python** implementation in RHEL 8. Consequently, **python-net-snmp**, **python2-net-snmp**, or **python3-net-snmp** packages are unavailable in RHEL 8.

(BZ#1584510)

systemd in debug mode produces unnecessary log messages

The **systemd** system and service manager in debug mode produces unnecessary log messages that start with:

```
"Failed to add rule for system call ..."
```

List the messages by running:

```
journalctl -b _PID=1
```

These debug messages are harmless, and you can safely ignore them.

Currently, there is no workaround available.

(BZ#1658691)

ksh with the KEYBD trap mishandles multibyte characters

The Korn Shell (KSH) is unable to correctly handle multibyte characters when the **KEYBD** trap is enabled. Consequently, when the user enters, for example, Japanese characters, **ksh** displays an incorrect string. To work around this problem, disable the **KEYBD** trap in the **/etc/kshrc** file by commenting out the following line:

```
trap keybd_trap KEYBD
```

For more details, see a related Knowledgebase solution.

(BZ#1503922)

5.5.7. Dynamic programming languages, web and database servers

Database servers are not installable in parallel

The **mariadb** and **mysql** modules cannot be installed in parallel in RHEL 8.0 due to conflicting RPM packages.

By design, it is impossible to install more than one version (stream) of the same module in parallel. For example, you need to choose only one of the available streams from the **postgresql** module, either **10** (default) or **9.6**. Parallel installation of components is possible in Red Hat Software Collections for RHEL 6 and RHEL 7. In RHEL 8, different versions of database servers can be used in containers.

(BZ#1566048)

Problems in mod_cgid logging

If the **mod_cgid** Apache httpd module is used under a threaded multi-processing module (MPM), which is the default situation in RHEL 8, the following logging problems occur:

- The **stderr** output of the CGI script is not prefixed with standard timestamp information.
- The **stderr** output of the CGI script is not correctly redirected to a log file specific to the **VirtualHost**, if configured.

(BZ#1633224)

The IO::Socket::SSL Perl module does not support TLS 1.3

New features of the TLS 1.3 protocol, such as session resumption or post-handshake authentication, were implemented in the RHEL 8 **OpenSSL** library but not in the **Net::SSLeay** Perl module, and thus are unavailable in the **IO::Socket::SSL** Perl module. Consequently, client certificate authentication might fail and reestablishing sessions might be slower than with the TLS 1.2 protocol.

To work around this problem, disable usage of TLS 1.3 by setting the **SSL_version** option to the **!TLSv1_3** value when creating an **IO::Socket::SSL** object.

(BZ#1632600)

Generated Scala documentation is unreadable

When generating documentation using the **scaladoc** command, the resulting HTML page is unusable due to missing JavaScript resources.

(BZ#1641744)

5.5.8. Desktop

qxl does not work on VMs based on Wayland

The **qxl** driver is not able to provide kernel mode setting features on certain hypervisors. Consequently, the graphics based on the Wayland protocol are not available to virtual machines (VMs) that use **qxl**, and the Wayland-based login screen does not start.

To work around the problem, use either :

- The **Xorg** display server instead of **GNOME Shell on Wayland** on VMs based on QuarkXpress Element Library (QXL) graphics.

Or

- The **virtio** driver instead of the **qxl** driver for your VMs.

(BZ#1641763)

The console prompt is not displayed when running systemctl isolate multi-user.target

When running the **systemctl isolate multi-user.target** command from GNOME Terminal in a GNOME Desktop session, only a cursor is displayed, and not the console prompt. To work around the problem, press the **Ctrl+Alt+F2** keys. As a result, the console prompt appears.

The behavior applies both to **GNOME Shell on Wayland** and **X.Org** display server.

(BZ#1678627)

5.5.9. Graphics infrastructures

Desktop running on X.Org hangs when changing to low screen resolutions

When using the GNOME desktop with the **X.Org** display server, the desktop becomes unresponsive if you attempt to change the screen resolution to low values. To work around the problem, do not set the screen resolution to a value lower than 800 × 600 pixels.

(BZ#1655413)

radeon fails to reset hardware correctly

The **radeon** kernel driver currently does not reset hardware in the kexec context correctly. Instead, **radeon** falls over, which causes the rest of the **kdump** service to fail.

To work around this problem, blacklist **radeon** in **kdump** by adding the following line to the **/etc/kdump.conf** file:

```
dracut_args --omit-drivers "radeon"
force_rebuild 1
```

Restart the machine and **kdump**. After starting **kdump**, the **force_rebuild 1** line may be removed from the configuration file.

Note that in this scenario, no graphics will be available during **kdump**, but **kdump** will work successfully.

(BZ#1694705)

5.5.10. Hardware enablement

Backup slave MII status does not work when using the ARP link monitor

By default, devices managed by the i40e driver, do source pruning, which drops packets that have the source Media Access Control (MAC) address that matches one of the receive filters. As a consequence, backup slave Media Independent Interface (MII) status does not work when using the Address Resolution Protocol (ARP) monitoring in channel bonding. To work around this problem, disable source pruning by the following command:

```
# ethtool --set-priv-flags <ethX> disable-source-pruning on
```

As a result, the backup slave MII status will work as expected.

(BZ#1645433)

The HP NMI watchdog in some cases does not generate a crash dump

The **hpwdt** driver for the HP NMI watchdog is sometimes not able to claim a non-maskable interrupt (NMI) generated by the HPE watchdog timer because the NMI was instead consumed by the **perfmon** driver.

As a consequence, **hpwdt** in some cases cannot call a panic to generate a crash dump.

(BZ#1602962)

5.5.11. Identity Management

The KCM credential cache is not suitable for a large number of credentials in a single credential cache

The Kerberos Credential Manager (KCM) can handle ccache sizes of up to 64 kB. If it contains too many credentials, Kerberos operations, such as **kinit**, fail due to a hardcoded limit on the buffer used to transfer data between the **sssd-kcm** component and the underlying database.

To work around this problem, add the **ccache_storage = memory** option in the **kcm** section of the **/etc/sssd/sssd.conf** file. This instructs the **kcm** responder to only store the credential caches in-memory, not persistently. If you do this, restarting the system or **sssd-kcm** clears the credential caches.

(BZ#1448094)

Changing /etc/nsswitch.conf requires a manual system reboot

Any change to the **/etc/nsswitch.conf** file, for example running the **authselect select profile_id** command, requires a system reboot so that all relevant processes use the updated version of the **/etc/nsswitch.conf** file. If a system reboot is not possible, restart the service that joins your system to Active Directory, which is the **System Security Services Daemon** (SSSD) or **winbind**.

(BZ#1657295)

Conflicting timeout values prevent SSSD from connecting to servers

Some of the default timeout values related to the failover operations used by the System Security Services Daemon (SSSD) are conflicting. Consequently, the timeout value reserved for SSSD to talk to a single server prevents SSSD from trying other servers before the connecting operation as a whole time out. To work around the problem, set the value of the **ldap_opt_timeout** timeout parameter higher than the value of the **dns_resolver_timeout** parameter, and set the value of the **dns_resolver_timeout** parameter higher than the value of the **dns_resolver_op_timeout** parameter.

(BZ#1382750)

SSSD can look up only unique certificates in ID overrides

When multiple ID overrides contain the same certificate, the System Security Services Daemon (SSSD) is unable to resolve queries for the users that match the certificate. An attempt to look up these users does not return any user. Note that looking up users by using their user name or UID works as expected.

(BZ#1446101)

SSSD does not correctly handle multiple certificate matching rules with the same priority

If a given certificate matches multiple certificate matching rules with the same priority, the System Security Services Daemon (SSSD) uses only one of the rules. As a workaround, use a single certificate matching rule whose LDAP filter consists of the filters of the individual rules concatenated with the | (or) operator. For examples of certificate matching rules, see the sss-certamp(5) man page.

(BZ#1447945)

SSSD returns incorrect LDAP group membership for local users

If the System Security Services Daemon (SSSD) serves users from the local files, the files provider does not include group memberships from other domains. As a consequence, if a local user is a member of an LDAP group, the **id local_user** command does not return the user's LDAP group membership. To work around the problem, either revert the order of the databases where the system is looking up the group membership of users in the **/etc/nsswitch.conf** file, replacing **sss files** with **files sss**, or disable the implicit **files** domain by adding

```
enable_files_domain=False
```

to the **[sssd]** section in the **/etc/sssd/sssd.conf** file.

As a result, **id local_user** returns correct LDAP group membership for local users.

(BZ#1652562)

Sudo rules might not work with id_provider=ad if sudo rules reference group names

System Security Services Daemon (SSSD) does not resolve Active Directory group names during the **initgroups** operation because of an optimization of communication between AD and SSSD by using a cache. The cache entry contains only a Security Identifiers (SID) and not group names until the group is requested by name or ID. Therefore, sudo rules do not match the AD group unless the groups are fully resolved prior to running sudo.

To work around this problem, you need to disable the optimization: Open the **/etc/sssd/sssd.conf** file and add the **ldap_use_tokengroups = false** parameter in the **[domain/example.com]** section.

(BZ#1659457)

Default PAM settings for systemd-user have changed in RHEL 8 which may influence SSSD behavior

The Pluggable authentication modules (PAM) stack has changed in Red Hat Enterprise Linux 8. For example, the **systemd** user session now starts a PAM conversation using the **systemd-user** PAM service. This service now recursively includes the **system-auth** PAM service, which may include the **pam_sss.so** interface. This means that the SSSD access control is always called.

Be aware of the change when designing access control rules for RHEL 8 systems. For example, you can add the **systemd-user** service to the allowed services list.

Please note that for some access control mechanisms, such as IPA HBAC or AD GPOs, the **systemd-user** service is has been added to the allowed services list by default and you do not need to take any action.

(BZ#1669407)

IdM server does not work in FIPS

Due to an incomplete implementation of the SSL connector for Tomcat, an Identity Management (IdM) server with a certificate server installed does not work on machines with the FIPS mode enabled.

(BZ#1673296)

Samba denies access when using the sss ID mapping plug-in

To use Samba as a file server on a RHEL host joined to an Active Directory (AD) domain, the Samba Winbind service must be running even if SSSD is used to manage user and groups from AD. If you join the domain using the **realm join --client-software=sssd** command or without specifying the **--client-software** parameter in this command, **realm** creates only the **/etc/sssd/sssd.conf** file. When you run Samba on the domain member with this configuration and add a configuration that uses the **sss** ID mapping back end to the **/etc/samba/smb.conf** file to share directories, changes in the ID mapping back end can cause errors. Consequently, Samba denies access to files in certain cases, even if the user or group exists and it is known by SSSD.

If you plan to upgrade from a previous RHEL version and the **ldap_id_mapping** parameter in the **/etc/sssd/sssd.conf** file is set to **True**, which is the default, no workaround is available. In this case, do not upgrade the host to RHEL 8 until the problem has been fixed.

Possible workarounds in other scenarios:

- For new installations, join the domain using the **realm join --client-software=winbind** command. This configures the system to use Winbind instead of SSSD for all user and group lookups. In this case, Samba uses the **rid** or **ad** ID mapping plug-in in **/etc/samba/smb.conf** depending on whether you set the **--automatic-id-mapping** option to **yes** (default) or **no**. If you plan to use SSSD in future or on other systems, using **--automatic-id-mapping=no** allows an easier migration but requires that you store POSIX UIDs and GIDs in AD for all users and groups.

- When upgrading from a previous RHEL version, and if the **ldap_id_mapping** parameter in the **/etc/sssd/sssd.conf** file is set to **False** and the system uses the **uidNumber** and **gidNumber** attributes from AD for ID mapping:

    1. Change the **idmap config <domain> : backend = sss** entry in the **/etc/samba/smb.conf** file to **idmap config <domain> : backend = ad**

    2. Use the **systemctl status winbind** command to restart the Winbind.

(BZ#1657665)

The nuxwdog service fails in HSM environments and requires to install the keyutils package in non-HSM environments

The **nuxwdog** watchdog service has been integrated into Certificate System. As a consequence, **nuxwdog** is no longer provided as a separate package. To use the watchdog service, install the **pki-server** package.

Note that the **nuxwdog** service has following known issues:

- The **nuxwdog** service does not work if you use a hardware storage module (HSM). For this issue, no workaround is available.

- In a non-HSM environment, Red Hat Enterprise Linux 8.0 does not automatically install the **keyutils** package as a dependency. To install the package manually, use the **dnf install keyutils** command.

(BZ#1652269)

Adding ID overrides of AD users works only in the IdM CLI

Currently, adding ID overrides of Active Directory (AD) users to Identity Management (IdM) groups for the purpose of granting access to management roles fails in the IdM Web UI. To work around the problem, use the IdM command-line interface (CLI) instead.

Note that if you installed the **ipa-idoverride-memberof-plugin** package on the IdM server after previously performing certain operations using the **ipa** utility, Red Hat recommends cleaning up the **ipa** utility's cache to force it to refresh its view about the IdM server metadata.

To do so, remove the content of the **~/.cache/ipa** directory for the user under which the **ipa** utility is executed. For example, for root:

```
# rm -r /root/.cache/ipa
```

(BZ#1651577)

No information about required DNS records displayed when enabling support for AD trust in IdM

When enabling support for Active Directory (AD) trust in Red Hat Enterprise Linux Identity Management (IdM) installation with external DNS management, no information about required DNS records is displayed. Forest trust to AD is not successful until the required DNS records are added. To work around this problem, run the 'ipa dns-update-system-records --dry-run' command to obtain a list of all DNS records required by IdM. When external DNS for IdM domain defines the required DNS records, establishing forest trust to AD is possible.

(BZ#1665051)

Potential risk when using the default value for ldap_id_use_start_tls option

When using **ldap://** without TLS for identity lookups, it can pose a risk for an attack vector. Particularly a man-in-the-middle (MITM) attack which could allow an attacker to impersonate a user by altering, for example, the UID or GID of an object returned in an LDAP search.

Currently, the SSSD configuration option to enforce TLS, **ldap_id_use_start_tls**, defaults to **false**. Ensure that your setup operates in a trusted environment and decide if it is safe to use unencrypted communication for **id_provider = ldap**. Note **id_provider = ad** and **id_provider = ipa** are not affected as they use encrypted connections protected by SASL and GSSAPI.

If it is not safe to use unencrypted communication, enforce TLS by setting the **ldap_id_use_start_tls** option to **true** in the **/etc/sssd/sssd.conf** file. The default behavior is planned to be changed in a future release of RHEL.

(JIRA:RHELPLAN-155168)

5.5.12. Compilers and development tools

Synthetic functions generated by GCC confuse SystemTap

GCC optimization can generate synthetic functions for partially inlined copies of other functions. Tools such as SystemTap and GDB can not distinguish these synthetic functions from real functions. As a consequence, SystemTap can place probes on both synthetic and real function entry points, and thus register multiple probe hits for a single real function call.

To work around this problem, SystemTap scripts must be adapted with measures such as detecting recursion and suppressing probes related to inlined partial functions. For example, a script

```
probe kernel.function("can_nice").call { }
```

-

can try to avoid the described problem as follows:

```
global in_can_nice%

probe kernel.function("can_nice").call {
  in_can_nice[tid()] ++;
  if (in_can_nice[tid()] > 1) { next }
  /* code for real probe handler */
}

probe kernel.function("can_nice").return {
  in_can_nice[tid()] --;
}
```

Note that this example script does not take into account all possible scenarios, such as missed kprobes or kretprobes, or genuine intended recursion.

(BZ#1169184)

The ltrace tool does not report function calls

Because of improvements to binary hardening applied to all RHEL components, the **ltrace** tool can no longer detect function calls in binary files coming from RHEL components. As a consequence, **ltrace** output is empty because it does not report any detected calls when used on such binary files. There is no workaround currently available.

As a note, **ltrace** can correctly report calls in custom binary files built without the respective hardening flags.

(BZ#1618748, BZ#1655368)

5.5.13. File systems and storage

Unable to discover an iSCSI target using the iscsiuio package

Red Hat Enterprise Linux 8 does not allow concurrent access to PCI register areas. As a consequence, a **could not set host net params (err 29)** error was set and the connection to the discovery portal failed. To work around this problem, set the kernel parameter **iomem=relaxed** in the kernel command line for the iSCSI offload. This specifically involves any offload using the **bnx2i** driver. As a result, connection to the discovery portal is now successful and **iscsiuio** package now works correctly.

(BZ#1626629)

VDO volumes lose deduplication advice after moving to a different-endian platform

Virtual Data Optimizer (VDO) writes the Universal Deduplication Service (UDS) index header in the endian format native to your platform. VDO considers the UDS index corrupt and overwrites it with a new, blank index if you move your VDO volume to a platform that uses a different endian.

As a consequence, any deduplication advice stored in the UDS index prior to being overwritten is lost. VDO is then unable to deduplicate newly written data against the data that was stored before you moved the volume, leading to lower space savings.

(BZ#1696492)

The XFS DAX mount option is incompatible with shared copy-on-write data extents

An XFS file system formatted with the shared copy-on-write data extents feature is not compatible with the **-o dax** mount option. As a consequence, mounting such a file system with **-o dax** fails.

To work around the problem, format the file system with the **reflink=0** metadata option to disable shared copy-on-write data extents:

```
# mkfs.xfs -m reflink=0 block-device
```

As a result, mounting the file system with **-o dax** is successful.

For more information, see Creating a file system DAX namespace on an NVDIMM .

(BZ#1620330)

Certain SCSI drivers might sometimes use an excessive amount of memory

Certain SCSI drivers use a larger amount of memory than in RHEL 7. In certain cases, such as vPort creation on a Fibre Channel host bus adapter (HBA), the memory usage might be excessive, depending upon the system configuration.

The increased memory usage is caused by memory preallocation in the block layer. Both the multiqueue block device scheduling (BLK-MQ) and the multiqueue SCSI stack (SCSI-MQ) preallocate memory for each I/O request in RHEL 8, leading to the increased memory usage.

(BZ#1733278)

5.5.14. Networking

nftables does not support multi-dimensional IP set types

The **nftables** packet-filtering framework does not support set types with concatenations and intervals. Consequently, you cannot use multi-dimensional IP set types, such as **hash:net,port**, with **nftables**.

To work around this problem, use the **iptables** framework with the **ipset** tool if you require multi-dimensional IP set types.

(BZ#1593711)

The TRACE target in the iptables-extensions(8) man page does not refer to the nf_tables variant

The description of the **TRACE** target in the **iptables-extensions(8)** man page refers only to the **compat** variant, but Red Hat Enterprise Linux (RHEL) 8.0 uses the **nf_tables** variant. The **nftables**-based **iptables** utility in RHEL uses the **meta nftrace** expression internally. Therefore, the kernel does not print **TRACE** events in the kernel log but sends them to the user space instead. However, the man page does not reference the **xtables-monitor** command-line utility to display these events.

(BZ#1658734)

RHEL 8 shows the status of an 802.3ad bond as "Churned" after a switch was unavailable for an extended period of time

Currently, when you configure an 802.3ad network bond and the switch is down for an extended period of time, Red Hat Enterprise Linux properly shows the status of the bond as "Churned", even after the connection returns to a working state. However, this is the intended behavior, as the "Churned" status aims to tell the administrator that a significant link outage occurred. To clear this status, restart the network bond or reboot the host.

(BZ#1708807)

The ebtables command does not support broute table

The **nftables**-based **ebtables** command in Red Hat Enterprise Linux 8.0 does not support the **broute** table. Consequently, users can not use this feature.

(BZ#1649790)

IPsec network traffic fails during IPsec offloading when GRO is disabled

IPsec offloading is not expected to work when Generic Receive Offload (GRO) is disabled on the device. If IPsec offloading is configured on a network interface and GRO is disabled on that device, IPsec network traffic fails.

To work around this problem, keep GRO enabled on the device.

(BZ#1649647)

NetworkManager now uses the internal DHCP plug-in by default

NetworkManager supports the **internal** and **dhclient** DHCP plug-ins. By default, **NetworkManager** in Red Hat Enterprise Linux (RHEL) 7 uses the **dhclient** and RHEL 8 the **internal** plug-in. In certain situations, the plug-ins behave differently. For example, **dhclient** can use additional settings specified in the **/etc/dhcp/** directory.

If you upgrade from RHEL 7 to RHEL 8 and **NetworkManager** behaves different, add the following setting to the **[main]** section in the **/etc/NetworkManager/NetworkManager.conf** file to use the **dhclient** plug-in:

```
[main]
dhcp=dhclient
```

(BZ#1571655)

Advanced options of IPsec based VPN cannot be changed using gnome-control-center

When configuring an **IPsec based VPN** connection using the **gnome-control-center** application, the **Advanced** dialog will only display the configuration, but will not allow doing any change. As a consequence, users cannot change any advanced IPsec options. To work around this problem, use the **nm-connection-editor** or **nmcli** tools to perform configuration of the advanced properties.

(BZ#1697326)

The /etc/hosts.allow and /etc/hosts.deny files contain inaccurate information

The tcp_wrappers package is removed in Red Hat Enterprise Linux (RHEL) 8, but not its files, /etc/hosts.allow and /etc/hosts.deny. As a consequence, these files contain outdated information, which is not applicable for RHEL 8.

To work around this problem, use firewall rules for filtering access to the services. For filtering based on usernames and hostnames, use the application-specific configuration.

(BZ#1663556)

IP defragmentation cannot be sustainable under network traffic overload

In Red Hat Enterprise Linux 8, the garbage collection kernel thread has been removed and IP fragments expire only on timeout. As a result, CPU usage under Denial of Service (DoS) is much lower, and the

maximum sustainable fragments drop rate is limited by the amount of memory configured for the IP reassembly unit. With the default settings workloads requiring fragmented traffic in presence of packet drops, packet reorder or many concurrent fragmented flows may incur in relevant performance regression.

In this case, users can use the appropriate tuning of the IP fragmentation cache in the **/proc/sys/net/ipv4** directory setting the **ipfrag_high_thresh** variable to limit the amount of memory and the **ipfrag_time** variable to keep per seconds an IP fragment in memory. For example,

echo 419430400 > /proc/sys/net/ipv4/ipfrag_high_thresh echo 1 > /proc/sys/net/ipv4/ipfrag_time

The above applies to IPv4 traffic. For IPv6 the relevant tunables are: **ip6frag_high_thresh** and **ip6frag_time** in the **/proc/sys/net/ipv6/** directory.

Note that any workload relying on high-speed fragmented traffic can cause stability and performance issues, especially with packet drops, and such kind of deployments are highly discouraged in production.

(BZ#1597671)

Network interface name changes in RHEL 8

In Red Hat Enterprise Linux 8, the same consistent network device naming scheme is used by default as in RHEL 7. However, some kernel drivers, such as **e1000e**, **nfp**, **qede**, **sfc**, **tg3** and **bnxt_en** changed their consistent name on a fresh installation of RHEL 8. However, the names are preserved on upgrade from RHEL 7.

(BZ#1701968)

5.5.15. Security

libselinux-python is available only through its module

The **libselinux-python** package contains only Python 2 bindings for developing SELinux applications and it is used for backward compatibility. For this reason, **libselinux-python** is no longer available in the default RHEL 8 repositories through the **dnf install libselinux-python** command.

To work around this problem, enable both the **libselinux-python** and **python27** modules, and install the **libselinux-python** package and its dependencies with the following commands:

```
# dnf module enable libselinux-python
# dnf install libselinux-python
```

Alternatively, install **libselinux-python** using its install profile with a single command:

```
# dnf module install libselinux-python:2.8/common
```

As a result, you can install **libselinux-python** using the respective module.

(BZ#1666328)

libssh does not comply with the system-wide crypto policy

The **libssh** library does not follow system-wide cryptographic policy settings. As a consequence, the set of supported algorithms is not changed when the administrator changes the crypto policies level using the **update-crypto-policies** command.

To work around this problem, the set of advertised algorithms needs to be set individually by every application that uses **libssh**. As a result, when the system is set to the LEGACY or FUTURE policy level, applications that use **libssh** behave inconsistently when compared to **OpenSSH**.

(BZ#1646563)

Certain rsyslog priority strings do not work correctly

Support for the **GnuTLS** priority string for **imtcp** that allows fine-grained control over encryption is not complete. Consequently, the following priority strings do not work properly in **rsyslog**:

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+DHE-RSA:+AES-256-GCM:+SIGN-RSA-SHA384:+COMP-ALL:+GROUP-ALL
```

To work around this problem, use only correctly working priority strings:

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+ECDHE-RSA:+AES-128-CBC:+SIGN-RSA-SHA1:+COMP-ALL:+GROUP-ALL
```

As a result, current configurations must be limited to the strings that work correctly.

(BZ#1679512)

Negative effects of the default logging setup on performance

The default logging environment setup might consume 4 GB of memory or even more and adjustments of rate-limit values are complex when **systemd-journald** is running with **rsyslog**.

See the Negative effects of the RHEL default logging setup on performance and their mitigations Knowledgebase article for more information.

(JIRA:RHELPLAN-10431)

OpenSCAP rpmverifypackage does not work correctly

The **chdir** and **chroot** system calls are called twice by the **rpmverifypackage** probe. Consequently, an error occurs when the probe is utilized during an **OpenSCAP** scan with custom Open Vulnerability and Assessment Language (OVAL) content.

To work around this problem, do not use the **rpmverifypackage_test** OVAL test in your content or use only the content from the **scap-security-guide** package where **rpmverifypackage_test** is not used.

(BZ#1646197)

SCAP Workbench fails to generate results-based remediations from tailored profiles

The following error occurs when trying to generate results-based remediation roles from a customized profile using the **SCAP Workbench** tool:

```
Error generating remediation role .../remediation.sh: Exit code of oscap was 1: [output truncated]
```

To work around this problem, use the **oscap** command with the **--tailoring-file** option.

(BZ#1640715)

Kickstart uses org_fedora_oscap instead of com_redhat_oscap in RHEL 8

The Kickstart references the Open Security Content Automation Protocol (OSCAP) Anaconda add-on as **org_fedora_oscap** instead of **com_redhat_oscap** which might cause confusion. That is done to preserve backward compatibility with Red Hat Enterprise Linux 7.

(BZ#1665082)

OpenSCAP rpmverifyfile does not work

The **OpenSCAP** scanner does not correctly change the current working directory in offline mode, and the **fchdir** function is not called with the correct arguments in the **OpenSCAP rpmverifyfile** probe. Consequently, scanning arbitrary file systems using the **oscap-chroot** command fails if **rpmverifyfile_test** is used in an SCAP content. As a result, **oscap-chroot** aborts in the described scenario.

(BZ#1636431)

OpenSCAP does not provide offline scanning of virtual machines and containers

Refactoring of **OpenSCAP** codebase caused certain RPM probes to fail to scan VM and containers file systems in offline mode. For that reason, the following tools were removed from the **openscap-utils** package: **oscap-vm** and **oscap-chroot**. Also, the **openscap-containers** package was completely removed.

(BZ#1618489)

A utility for security and compliance scanning of containers is not available

In Red Hat Enterprise Linux 7, the **oscap-docker** utility can be used for scanning of Docker containers based on Atomic technologies. In Red Hat Enterprise Linux 8, the Docker- and Atomic-related **OpenSCAP** commands are not available. As a result, **oscap-docker** or an equivalent utility for security and compliance scanning of containers is not available in RHEL 8 at the moment.

(BZ#1642373)

The OpenSSL TLS library does not detect if the PKCS#11 token supports creation of raw RSA or RSA-PSS signatures

The **TLS-1.3** protocol requires the support for **RSA-PSS** signature. If the **PKCS#11** token does not support **raw RSA** or **RSA-PSS** signatures, the server applications which use **OpenSSL TLS** library will fail to work with the **RSA** key if it is held by the **PKCS#11** token. As a result, **TLS** communication will fail.

To work around this problem, configure server or client to use the **TLS-1.2** version as the highest **TLS** protocol version available.

(BZ#1681178)

Apache httpd fails to start if it uses an RSA private key stored in a PKCS#11 device and an RSA-PSS certificate

The PKCS#11 standard does not differentiate between RSA and RSA-PSS key objects and uses the **CKK_RSA** type for both. However, OpenSSL uses different types for RSA and RSA-PSS keys. As a consequence, the **openssl-pkcs11** engine cannot determine which type should be provided to OpenSSL for PKCS#11 RSA key objects. Currently, the engine sets the key type as RSA keys for all PKCS#11 **CKK_RSA** objects. When OpenSSL compares the types of an RSA-PSS public key obtained from the certificate with the type contained in an RSA private key object provided by the engine, it concludes that the types are different. Therefore, the certificate and the private key do not match. The check performed in the **X509_check_private_key()** OpenSSL function returns an error in this scenario. The **httpd** web server calls this function in its startup process to check if the provided certificate and key

match. Since this check always fails for a certificate containing an RSA-PSS public key and a RSA private key stored in the PKCS#11 module, **httpd** fails to start using this configuration. There is no workaround available for this issue.

(BZ#1664802)

httpd fails to start if it uses an ECDSA private key without corresponding public key stored in a PKCS#11 device

Unlike RSA keys, ECDSA private keys do not necessarily contain public key information. In this case, you cannot obtain the public key from an ECDSA private key. For this reason, a PKCS#11 device stores public key information in a separate object whether it is a public key object or a certificate object. OpenSSL expects the **EVP_PKEY** structure provided by an engine for a private key to contain the public key information. When filling the **EVP_PKEY** structure to be provided to OpenSSL, the engine in the **openssl-pkcs11** package tries to fetch the public key information only from matching public key objects and ignores the present certificate objects.

When OpenSSL requests an ECDSA private key from the engine, the provided **EVP_PKEY** structure does not contain the public key information if the public key is not present in the PKCS#11 device, even when a matching certificate that contains the public key is available. As a consequence, since the Apache **httpd** web server calls the **X509_check_private_key()** function, which requires the public key, in its start-up process, **httpd** fails to start in this scenario. To work around the problem, store both the private and public key in the PKCS#11 device when using ECDSA keys. As a result, **httpd** starts correctly when ECDSA keys are stored in the PKCS#11 device.

(BZ#1664807)

OpenSSH does not handle PKCS #11 URIs for keys with mismatching labels correctly

The OpenSSH suite can identify key pairs by a label. The label might differ on private and public keys stored on a smart card. Consequently, specifying PKCS #11 URIs with the object part (key label) can prevent OpenSSH from finding appropriate objects in PKCS #11.

To work around this problem, specify PKCS #11 URIs without the object part. As a result, OpenSSH is able to use keys on smart cards referenced using PKCS #11 URIs.

(BZ#1671262)

Output of iptables-ebtables is not 100% compatible with ebtables

In RHEL 8, the **ebtables** command is provided by the **iptables-ebtables** package, which contains an **nftables**-based reimplementation of the tool. This tool has a different code base, and its output deviates in aspects, which are either negligible or deliberate design choices.

Consequently, when migrating your scripts parsing some **ebtables** output, adjust the scripts to reflect the following:

- MAC address formatting has been changed to be fixed in length. Where necessary, individual byte values contain a leading zero to maintain the format of two characters per octet.
- Formatting of IPv6 prefixes has been changed to conform with RFC 4291. The trailing part after the slash character no longer contains a netmask in the IPv6 address format but a prefix length. This change applies to valid (left-contiguous) masks only, while others are still printed in the old formatting.

(BZ#1674536)

curve25519-sha256 is not supported by default in OpenSSH

The **curve25519-sha256** SSH key exchange algorithm is missing in the system-wide crypto policies configurations for the OpenSSH client and server even though it is compliant with the default policy level. As a consequence, if a client or a server uses **curve25519-sha256** and this algorithm is not supported by the host, the connection might fail.

To work around this problem, you can manually override the configuration of system-wide crypto policies by modifying the **openssh.config** and **opensshserver.config** files in the **/etc/crypto-policies/back-ends/** directory for the OpenSSH client and server. Note that this configuration is overwritten with every change of system-wide crypto policies. See the **update-crypto-policies(8)** man page for more information.

(BZ#1678661)

OpenSSL incorrectly handles PKCS #11 tokens that does not support raw RSA or RSA-PSS signatures

The **OpenSSL** library does not detect key-related capabilities of PKCS #11 tokens. Consequently, establishing a TLS connection fails when a signature is created with a token that does not support raw RSA or RSA-PSS signatures.

To work around the problem, add the following lines after the **.include** line at the end of the **crypto_policy** section in the **/etc/pki/tls/openssl.cnf** file:

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

As a result, a TLS connection can be established in the described scenario.

(BZ#1685470)

SSH connections with VMware-hosted systems do not work

The current version of the **OpenSSH** suite introduces a change of the default IP Quality of Service (IPQoS) flags in SSH packets, which is not correctly handled by the VMware virtualization platform. Consequently, it is not possible to establish an SSH connection with systems on VMware.

To work around this problem, include the **IPQoS=throughput** in the **ssh_config** file. As a result, SSH connections with VMware-hosted systems work correctly.

See the RHEL 8 running in VMWare Workstation unable to connect via SSH to other hosts Knowledgebase solution article for more information.

(BZ#1651763)

5.5.16. Subscription management

No message is printed for the successful setting and unsetting of service-level

When the **candlepin** service does not have a 'syspurpose' functionality, subscription manager uses a different code path to set the **service-level** argument. This code path does not print the result of the operation. As a consequence, no message is displayed when the service level is set by subscription manager. This is especially problematic when the **service-level** set has a typo or is not truly available.

(BZ#1661414)

syspurpose addons have no effect on the subscription-manager attach --auto output.

In Red Hat Enterprise Linux 8, four attributes of the **syspurpose** command-line tool have been added: **role**,**usage**, **service_level_agreement** and **addons**. Currently, only **role**, **usage** and **service_level_agreement** affect the output of running the **subscription-manager attach --auto** command. Users who attempt to set values to the **addons** argument will not observe any effect on the subscriptions that are auto-attached.

(BZ#1687900)

5.5.17. Virtualization

ESXi virtual machines that were customized using cloud-init and cloned boot very slowly

Currently, if the **cloud-init** service is used to modify a virtual machine (VM) that runs on the VMware ESXi hypervisor to use static IP and the VM is then cloned, the new cloned VM in some cases takes a very long time to reboot. This is caused **cloud-init** rewriting the VM's static IP to DHCP and then searching for an available datasource.

To work around this problem, you can uninstall **cloud-init** after the VM is booted for the first time. As a result, the subsequent reboots will not be slowed down.

(BZ#1666961, BZ#1706482)

Enabling nested virtualization blocks live migration

Currently, the nested virtualization feature is incompatible with live migration. Therefore, enabling nested virtualization on a RHEL 8 host prevents migrating any virtual machines (VMs) from the host, as well as saving VM state snapshots to disk.

Note that nested virtualization is currently provided as a Technology Preview in RHEL 8, and is therefore not supported. In addition, nested virtualization is disabled by default. If you want to enable it, use the **kvm_intel.nested** or **kvm_amd.nested** module parameters.

(BZ#1689216)

Using cloud-init to provision virtual machines on Microsoft Azure fails

Currently, it is not possible to use the **cloud-init** utility to provision a RHEL 8 virtual machine (VM) on the Microsoft Azure platform. To work around this problem, use one of the following methods:

- Use the **WALinuxAgent** package instead of **cloud-init** to provision VMs on Microsoft Azure.
- Add the following setting to the **[main]** section in the **/etc/NetworkManager/NetworkManager.conf** file:

```
[main]
dhcp=dhclient
```

(BZ#1641190)

Generation 2 RHEL 8 virtual machines sometimes fail to boot on Hyper-V Server 2016 hosts

When using RHEL 8 as the guest operating system on a virtual machine (VM) running on a Microsoft Hyper-V Server 2016 host, the VM in some cases fails to boot and returns to the GRUB boot menu. In addition, the following error is logged in the Hyper-V event log:

```
The guest operating system reported that it failed with the following error code: 0x1E
```

This error occurs due to a UEFI firmware bug on the Hyper-V host. To work around this problem, use Hyper-V Server 2019 as the host.

(BZ#1583445)

virsh iface-* commands do not work consistently

Currently, **virsh iface-*** commands, such as **virsh iface-start** and **virsh iface-destroy**, frequently fail due to configuration dependencies. Therefore, it is recommended not to use **virsh iface-*** commands for configuring and managing host network connections. Instead, use the NetworkManager program and its related management applications.

(BZ#1664592)

Linux virtual machine extensions for Azure sometimes do not work

RHEL 8 does not include the **python2** package by default. As a consequence, running Linux virtual machine extensions for Azure, also known as **azure-linux-extensions**, on a RHEL 8 VM in some cases fails.

To increase the probability that **azure-linux-extensions** will work as expected, install **python2** on the RHEL 8 VM manually:

```
# yum install python2
```

(BZ#1561132)

5.5.18. Supportability

redhat-support-tool does not collect sosreport automatically from opencase

The **redhat-support-tool** command cannot create a **sosreport** archive. To work around this problem, run the **sosreport** command separately and then enter the **redhat-support-tool addattachment -c** command to upload the archive or use web UI on the Customer Portal. As a result, a case will be created and **sosreport** will be uploaded.

Note that the **findkerneldebugs**, **btextract**, **analyze diagnose** commands do not work as expected and will be fixed in future releases.

(BZ#1688274)

CHAPTER 6. NOTABLE CHANGES TO CONTAINERS

A set of container images is available for Red Hat Enterprise Linux (RHEL) 8.0. Notable changes include:

- Docker is not included in RHEL 8.0. For working with containers, use the **podman**, **buildah**, **skopeo**, and **runc** tools.
  For information on these tools and on using containers in RHEL 8, see Building, running, and managing containers.

- The **podman** tool has been released as a fully supported feature.
  The **podman** tool manages pods, container images, and containers on a single node. It is built on the **libpod** library, which enables management of containers and groups of containers, called pods.

  To learn how to use **podman**, see Building, running, and managing containers.

- In RHEL 8 GA, Red Hat Universal Base Images (UBI) are newly available. UBIs replace some of the images Red Hat previously provided, such as the standard and the minimal RHEL base images.
  Unlike older Red Hat images, UBIs are freely redistributable. This means they can be used in any environment and shared anywhere. You can use them even if you are not a Red Hat customer.

  For UBI documentation, see Building, running, and managing containers.

- In RHEL 8 GA, additional container images are available that provide AppStream components, for which container images are distributed with **Red Hat Software Collections** in RHEL 7. All of these RHEL 8 images are based on the **ubi8** base image.

- Container images ARM for the 64-bit ARM architecture are fully supported in RHEL 8.

- The **rhel-tools** container has been removed in RHEL 8. The **sos** and **redhat-support-tool** tools are provided in the **support-tools** container. System administrators can also use this image as a base for building system tools container image.

- The support for rootless containers is available as a technology preview in RHEL 8.
  Rootless containers are containers that are created and managed by regular system users without administrative permissions.

CHAPTER 7. INTERNATIONALIZATION

7.1. RED HAT ENTERPRISE LINUX 8 INTERNATIONAL LANGUAGES

Red Hat Enterprise Linux 8 supports the installation of multiple languages and the changing of languages based on your requirements.

- East Asian Languages - Japanese, Korean, Simplified Chinese, and Traditional Chinese.
- European Languages - English, German, Spanish, French, Italian, Portuguese, and Russian.

The following table lists the fonts and input methods provided for various major languages.

Language	Default Font (Font Package)	Input Methods
English	dejavu-sans-fonts	
French	dejavu-sans-fonts	
German	dejavu-sans-fonts	
Italian	dejavu-sans-fonts	
Russian	dejavu-sans-fonts	
Spanish	dejavu-sans-fonts	
Portuguese	dejavu-sans-fonts	
Simplified Chinese	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libpinyin, libpinyin
Traditional Chinese	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libzhuyin, libzhuyin
Japanese	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-kkc, libkkc
Korean	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-hangul, libhangu

7.2. NOTABLE CHANGES TO INTERNATIONALIZATION IN RHEL 8

RHEL 8 introduces the following changes to internationalization compared to RHEL 7:

- Support for the **Unicode 11** computing industry standard has been added.
- Internationalization is distributed in multiple packages, which allows for smaller footprint installations.

For more information, see **glibc** localization for RHEL is distributed in multiple packages .

- The **glibc** package updates for multiple locales are now synchronized with the Common Locale Data Repository (CLDR).

APPENDIX A. LIST OF TICKETS BY COMPONENT

Component	Tickets
389-ds-base	BZ#1334254, BZ#1358706
NetworkManager	BZ#1555013, BZ#1555012, BZ#1557035, BZ#1335409, BZ#1571655
PackageKit	BZ#1559414
WALinuxAgent	BZ#1561132
anaconda	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904, BZ#1672405, JIRA:RHELPLAN-1943, BZ#1677411, BZ#1502323, BZ#1696609
audit	BZ#1616428
authselect	BZ#1657295
bcc	BZ#1548302
bind	BZ#1588592
boom-boot	BZ#1649582
boost	BZ#1494495, BZ#1616244
cloud-init	BZ#1615599, BZ#1641190
cmake	BZ#1590139
cockpit	BZ#1619993, BZ#1631905
criu	BZ#1689746
crypto-policies	BZ#1591620, BZ#1645606, BZ#1678661, BZ#1660839
cryptsetup	BZ#1564540
device-mapper-multipath	BZ#1643550, BZ#1673167
distribution	BZ#1516728, BZ#1516741, BZ#1566048
dnf	BZ#1622580, BZ#1647760, BZ#1581191
driverctl	BZ#1648411

Component	Tickets
edk2	BZ#1536627
esc	BZ#1538645
firewalld	BZ#1509026, BZ#1648497
gcc	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1571124, BZ#1246444, JIRA:RHELPLAN-7437, BZ#1652016
gdb	BZ#1491128
gdm	BZ#1589678, BZ#1641763, BZ#1678627
glib-networking	BZ#1640534
glibc	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
gnome-control-center	BZ#1697326
go-toolset-1.10-golang	BZ#1633351
grub2	BZ#1583445
httpd	BZ#1633224, BZ#1632754
ipa-idoverride-memberof	BZ#1651577
ipa	BZ#1664718, BZ#1664719, BZ#1665051
iproute	BZ#1640991, BZ#1589317
iptables	BZ#1644030, BZ#1564596, BZ#1646159, BZ#1658734, BZ#1649790, BZ#1674536
iscsi-initiator-utils	BZ#1626629, BZ#1582099
kernel-rt	BZ#1592977

Component	Tickets
kernel	BZ#1598448, BZ#1559607, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1273139, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1633143, BZ#1596240, BZ#1534870, BZ#1153521, BZ#1515987, BZ#1642795, BZ#1570255, BZ#1645744, BZ#1440031, BZ#1649647, BZ#1494705, BZ#1650149, BZ#1655413, BZ#1651806, BZ#1620330, BZ#1665295, BZ#1505999, BZ#1645433, BZ#1663281, BZ#1695142, BZ#1627455, BZ#1581898, BZ#1597671, BZ#1550498, BZ#1658391, BZ#1623590, BZ#1614144, BZ#1519039, BZ#1524683, BZ#1694705
kexec-tools	BZ#1520209, BZ#1662911
kmod-kvdo	BZ#1534087, BZ#1639512, BZ#1696492
ksh	BZ#1503922
libdnf	BZ#1642458, BZ#1679509
libreswan	BZ#1566574, BZ#1648776, BZ#1657854
libssh	BZ#1485241
libvirt	BZ#1528684
lksctp-tools	BZ#1568622
ltrace	BZ#1618748, BZ#1584322
lvm2	BZ#1676598, BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
mariadb	BZ#1637034
mdadm	BZ#1654482
mutter	BZ#1668883
net-snmp	BZ#1584510
nfs-utils	BZ#1592011, BZ#1639432
nftables	BZ#1593711
nginx	BZ#1545526

Component	Tickets
nodejs-10-module	BZ#1622118
nss	BZ#1489094, BZ#1645153
nuxwdog	BZ#1652269
openldap	BZ#1570056
opensc	BZ#1595638, BZ#1595626
openscap	BZ#1614273, BZ#1618484, BZ#1646197, BZ#1636431, BZ#1618489, BZ#1642373, BZ#1618464
openssh	BZ#1622511, BZ#1228088, BZ#1645038, BZ#1671262, BZ#1651763
openssl-pkcs11	BZ#1664802, BZ#1664807
openssl	BZ#1685470
oscap-anaconda-addon	BZ#1665082
pacemaker	BZ#1543494
pcs	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
perl	BZ#1511131
pki-core	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, BZ#1669257, BZ#1656856, BZ#1673296
postgresql-9.6-module	BZ#1660041
pykickstart	BZ#1637872, BZ#1612061
python-rtslib	BZ#1666377
qemu-kvm	BZ#1559240, BZ#1508139, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1592337, BZ#1570029, BZ#1689216, BZ#1585651, BZ#1519004
redhat-release	BZ#1636338

Component	Tickets
redhat-support-tool	BZ#1688274
rsyslog	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645, BZ#1679512, JIRA:RHELPLAN-10431
scala-2.10-module	BZ#1641744
scap-security-guide	BZ#1618505, BZ#1618528, BZ#1618518
scap-workbench	BZ#1640715
selinux-policy	BZ#1664345, BZ#1594111, BZ#1592244, BZ#1549772, BZ#1483904, BZ#1626446
setup	BZ#1591969, BZ#1663556
sos	BZ#1559836
squid	BZ#1656871
sssd	BZ#1448094, BZ#1382750, BZ#1446101, BZ#1447945, BZ#1620123, BZ#1652562, BZ#1659457, BZ#1669407, BZ#1657665
subscription-manager	BZ#1654531, BZ#1661414
subversion	BZ#1571415
swig-3.0-module	BZ#1660051
systemd	BZ#1658691
tomcatjss	BZ#1424966, BZ#1636564
tuned	BZ#1565598
valgrind	BZ#1500481, BZ#1538009
varnish	BZ#1633338
vdo	BZ#1669537
virt-manager	BZ#1599777, BZ#1643609
wpa_supplicant	BZ#1582538, BZ#1537143

Component	Tickets
xorg-x11-server	BZ#1687489, BZ#1698565
other	JIRA:RHELPLAN-10347, BZ#1646563, JIRA:RHELPLAN-2306, BZ#1640697, BZ#1623712, BZ#1649404, BZ#1581198, BZ#1581990, BZ#1649497, BZ#1695584, BZ#1654280, BZ#1643294, BZ#1647612, BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1647110, BZ#1641007, BZ#1641029, BZ#1641022, JIRA:RHELPLAN-1212, BZ#1649493, BZ#1559616, BZ#1699825, BZ#1646541, BZ#1647725, BZ#1686057, BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, BZ#1650701, JIRA:RHELPLAN-10439, JIRA:RHELPLAN-10440, JIRA:RHELPLAN-10442, JIRA:RHELPLAN-10443, JIRA:RHELPLAN-10438, JIRA:RHELPLAN-2878, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-3010, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-1473, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-3001, JIRA:RHELPLAN-6746, JIRA:RHELPLAN-10354, JIRA:RHELPLAN-2896, JIRA:RHELPLAN-10304, JIRA:RHELPLAN-10628, JIRA:RHELPLAN-10441, JIRA:RHELPLAN-10444, JIRA:RHELPLAN-1842, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-7291, JIRA:RHELPLAN-12764, BZ#1680177, JIRA:RHELPLAN-14607, JIRA:RHELPLAN-1820, BZ#1684947, BZ#1683712, BZ#1659609, BZ#1504934, BZ#1642765, BZ#1641014, BZ#1692746, BZ#1687900, BZ#1690207, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#1649891, BZ#1695698, BZ#1697896, BZ#1698613, BZ#1699535, BZ#1701968, BZ#1704867

ACKNOWLEDGEMENTS

Thank you to everyone who provided feedback as part of the RHEL 8 Readiness Challenge. The top 3 winners are:

- Sterling Alexander
- John Pittman
- Jake Hunsaker

APPENDIX B. REVISION HISTORY

0.1-7

Thu May 9 2024, Brian Angelica (bangelic@redhat.com)

- Updated Tech Preview in BZ#1690207.

0.1-6

Fri Nov 10 2023, Gabriela Fialová (gfialova@redhat.com)

- Updated the module on Providing Feedback on RHEL Documentation.

0.1-5

Fri Oct 13 2023, Gabriela Fialová (gfialova@redhat.com)

- Added a Tech Preview JIRA:RHELDOCS-16861 (Containers).

0.1-4

Thu Apr 27 2023, Gabriela Fialová (gfialova@redhat.com)

- Added a known issue JIRA:RHELPLAN-155168 (Identity Management).

0.1-3

Fri Apr 29 2022, Lenka Špačková (lspackova@redhat.com)

- Updated Deprecated functionality introduction.
- Fixed typo in BZ#1605216.
- Fixed broken links.

0.1-2

Thu Mar 17 2022, Lucie Maňásková (lmanasko@redhat.com)

Added JIRA:RHELPLAN-14323, JIRA:RHELPLAN-14329, and JIRA:RHELPLAN-14330 to the New features section (Virtualization).

0.1-1

Thu Dec 23 2021, Lenka Špačková (lspackova@redhat.com)

- Added information about the Soft-RoCE driver, **rdma_rxe**, to Technology Previews BZ#1605216 and Deprecated functionality BZ#1878207 (Kernel).

0.1-0

Thu Sep 23 2021, Lucie Maňásková (lmanasko@redhat.com)

- Removed an invalid new feature description (Virtualization).

0.0-9

Thu Aug 19 2021, Lucie Maňásková (lmanasko@redhat.com)

- Added the Package management with YUM/DNF to the Distribution chapter.

0.0-8

Wed Jun 23 2021, Lucie Maňásková (lmanasko@redhat.com)

- Updated the New features section (Installer).

0.0-7

Tue Apr 06 2021, Lenka Špačková (lspackova@redhat.com)

- Improved the list of supported architectures.

0.0-6

Thu Jan 28 2021, Lucie Maňásková (lmanasko@redhat.com)

- Updated the Technology Previews chapter.

0.0-5

Thu Dec 10 2020, Lenka Špačková (lspackova@redhat.com)

- Added information about handling AD GPOs in SSSD to New features (Identity Management).

0.0-4

Tue Apr 28 2020, Lenka Špačková (lspackova@redhat.com)

- Updated information about in-place upgrades in Overview.

0.0-3

Thu Mar 12 2020, Lenka Špačková (lspackova@redhat.com)

- Added the missing **postfix** RHEL system role to Technology Previews.

0.0-2

Wed Feb 12 2020, Jaroslav Klech (jklech@redhat.com)

- Provided a complete kernel version to Architectures and New Features chapters.

0.0-1

Tue Jul 30 2019, Lucie Maňásková (lmanasko@redhat.com)

- Release of the Red Hat Enterprise Linux 8.0.1 Release Notes.

0.0-0

Tue May 07 2019, Ioanna Gkioka (igkioka@redhat.com)

- Release of the Red Hat Enterprise Linux 8.0 Release Notes.