



# Red Hat Enterprise Linux 8

## 8.0 Freigabemitteilungen (Maschinenübersetzung)

Versionshinweise für Red Hat Enterprise Linux 8.0 (Maschinenübersetzung)



# Red Hat Enterprise Linux 8 8.0 Freigabemitteilungen (Maschinenübersetzung)

---

Versionshinweise für Red Hat Enterprise Linux 8.0 (Maschinenübersetzung)

## Rechtlicher Hinweis

Copyright © 2019 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

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

All other trademarks are the property of their respective owners.

## Zusammenfassung

Die Versionshinweise bieten einen umfassenden Überblick über die Verbesserungen und Ergänzungen, die in Red Hat Enterprise Linux 8.0 implementiert wurden, und dokumentieren bekannte Probleme in dieser Version sowie bemerkenswerte Bugfixes, Technologievorschaue, veraltete Funktionalitäten, entfernte Funktionalitäten und andere Details.

## Inhaltsverzeichnis

<b>FEEDBACK ZUR RED HAT-DOKUMENTATION GEBEN (MASCHINENÜBERSETZUNG)</b> .....	<b>4</b>
<b>KAPITEL 1. ÜBERSICHT (MASCHINENÜBERSETZUNG)</b> .....	<b>5</b>
Vertrieb	5
Software-Management	5
Webserver, Datenbanken, dynamische Sprachen	5
Desktop	5
Installationsprogramm und Image-Erstellung	5
Dateisysteme und Speicher	6
Sicherheit	6
Vernetzung	6
Virtualisierung	6
Compiler und Entwicklungswerkzeuge	6
Hochverfügbarkeit und Cluster	7
Zusätzliche Ressourcen	7
<b>KAPITEL 2. ARCHITEKTUREN (MASCHINENÜBERSETZUNG)</b> .....	<b>8</b>
<b>KAPITEL 3. VERTEILUNG VON INHALTEN IN RHEL 8 (MASCHINENÜBERSETZUNG)</b> .....	<b>9</b>
3.1. INSTALLATION (MASCHINENÜBERSETZUNG)	9
3.2. REPOSITORIEN (MASCHINENÜBERSETZUNG)	9
3.3. ANWENDUNGS-STREAMS (MASCHINENÜBERSETZUNG)	9
<b>KAPITEL 4. NEUE FUNKTIONEN (MASCHINENÜBERSETZUNG)</b> .....	<b>11</b>
4.1. DIE WEB-KONSOLE (MASCHINENÜBERSETZUNG)	11
4.2. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)	12
4.3. KERNEL (MASCHINENÜBERSETZUNG)	14
4.4. SOFTWARE-MANAGEMENT (MASCHINENÜBERSETZUNG)	15
4.5. INFRASTRUKTURDIENSTLEISTUNGEN (MASCHINENÜBERSETZUNG)	16
4.6. SHELLS UND BEFEHLSZEILEN-TOOLS (MASCHINENÜBERSETZUNG)	17
4.7. DYNAMISCHE PROGRAMMIERSPRACHEN, WEB- UND DATENBANKSERVER (MASCHINENÜBERSETZUNG)	18
4.8. DESKTOP (MASCHINENÜBERSETZUNG)	24
4.9. IDENTITÄTSMANAGEMENT (MASCHINENÜBERSETZUNG)	26
4.10. COMPILER UND ENTWICKLUNGSWERKZEUGE (MASCHINENÜBERSETZUNG)	31
4.11. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)	39
4.12. HOCHVERFÜGBARKEIT UND CLUSTER (MASCHINENÜBERSETZUNG)	44
4.13. VERNETZUNG (MASCHINENÜBERSETZUNG)	46
4.14. SICHERHEIT (MASCHINENÜBERSETZUNG)	50
4.15. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)	55
4.16. SUPPORTFÄHIGKEIT (MASCHINENÜBERSETZUNG)	58
<b>KAPITEL 5. TECHNOLOGIEVORSCHAUEN (MASCHINENÜBERSETZUNG)</b> .....	<b>59</b>
5.1. KERNEL (MASCHINENÜBERSETZUNG)	59
5.2. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)	60
5.3. HOCHVERFÜGBARKEIT UND CLUSTER (MASCHINENÜBERSETZUNG)	61
5.4. SICHERHEIT (MASCHINENÜBERSETZUNG)	61
5.5. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)	61
<b>KAPITEL 6. VERALTETE FUNKTIONALITÄT (MASCHINENÜBERSETZUNG)</b> .....	<b>63</b>
6.1. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)	63
6.2. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)	63
6.3. VERNETZUNG (MASCHINENÜBERSETZUNG)	64

6.4. SICHERHEIT (MASCHINENÜBERSETZUNG)	64
6.5. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)	64
<b>KAPITEL 7. BEKANNTE PROBLEME (MASCHINENÜBERSETZUNG) .....</b>	<b>66</b>
7.1. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)	66
7.2. KERNEL (MASCHINENÜBERSETZUNG)	66
7.3. SOFTWARE-MANAGEMENT (MASCHINENÜBERSETZUNG)	67
7.4. INFRASTRUKTURDIENSTLEISTUNGEN (MASCHINENÜBERSETZUNG)	68
7.5. SHELLS UND BEFEHLSZEILEN-TOOLS (MASCHINENÜBERSETZUNG)	68
7.6. DYNAMISCHE PROGRAMMIERSPRACHEN, WEB- UND DATENBANKSERVER (MASCHINENÜBERSETZUNG)	69
7.7. IDENTITÄTSMANAGEMENT (MASCHINENÜBERSETZUNG)	69
7.8. COMPILER UND ENTWICKLUNGSWERKZEUGE (MASCHINENÜBERSETZUNG)	70
7.9. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)	71
7.10. VERNETZUNG (MASCHINENÜBERSETZUNG)	72
7.11. SICHERHEIT (MASCHINENÜBERSETZUNG)	72
<b>KAPITEL 8. BEMERKENSWERTE ÄNDERUNGEN AN CONTAINERN (MASCHINENÜBERSETZUNG) .....</b>	<b>74</b>
<b>ANHANG A. LISTE DER TICKETS NACH KOMPONENTEN .....</b>	<b>75</b>



## FEEDBACK ZUR RED HAT-DOKUMENTATION GEBEN (MASCHINENÜBERSETZUNG)

Wir freuen uns über Ihren Input zu unserer Dokumentation. Bitte lasst uns wissen, wie wir es besser machen können. Um dies zu tun:

- Für einfache Kommentare zu bestimmten Passagen stellen Sie sicher, dass Sie die Dokumentation im mehrseitigen HTML-Format anzeigen. Markieren Sie den Teil des Textes, den Sie kommentieren möchten. Klicken Sie dann auf das Popup-Fenster **Feedback hinzufügen**, das unterhalb des markierten Textes angezeigt wird, und folgen Sie den angezeigten Anweisungen.
- Um komplexeres Feedback zu senden, erstellen Sie ein Bugzilla-Ticket:
  1. Gehen Sie auf die [Bugzilla](#)Website.
  2. Verwenden Sie als Komponente die **Dokumentation**.
  3. Füllen Sie das Feld **Beschreibung** mit Ihrem Verbesserungsvorschlag aus. Fügen Sie einen Link zu den relevanten Teilen der Dokumentation ein.
  4. Klicken Sie auf **Fehler senden**.



# KAPITEL 1. ÜBERSICHT (MASCHINENÜBERSETZUNG)

Basierend auf Fedora 28 und dem Upstream-Kernel 4.18 bietet Red Hat Enterprise Linux 8.0 Benutzern eine stabile, sichere und konsistente Grundlage für hybride Cloud-Implementierungen mit den Tools, die sie zur Unterstützung traditioneller und neuer Workloads benötigen. Zu den Highlights des Releases gehören:

## Vertrieb

- Inhalte sind über die Repositories **BaseOS** und Application Stream (**AppStream**) verfügbar.
- Das **AppStream-Repository** unterstützt eine neue Erweiterung der traditionellen RPM-Format - *Module*. Dies ermöglicht es, dass mehrere Hauptversionen einer Komponente für die Installation zur Verfügung stehen.

Siehe [Kapitel 3, Verteilung von Inhalten in RHEL 8 \(Maschinenübersetzung\)](#) für weitere Informationen.

## Software-Management

- Der **YUM-Paketmanager** basiert nun auf der **DNF-Technologie** und bietet Unterstützung für modulare Inhalte, erhöhte Leistung und eine gut durchdachte, stabile API für die Integration mit Tools.

Siehe [Abschnitt 4.4, »Software-Management \(Maschinenübersetzung\)«](#) für weitere Details.

## Webserver, Datenbanken, dynamische Sprachen

- **Python 3.6** ist die Standard-Python-Implementierung in RHEL 8; die Unterstützung für Python 2.7 ist begrenzt. Standardmäßig ist keine Version von Python installiert.
- RHEL 8 stellt die folgenden **Datenbankserver** zur Verfügung: MariaDB 10.3, MySQL 8.0, PostgreSQL 10, PostgreSQL 9.6 und Redis 4.0.

Siehe [Abschnitt 4.7, »Dynamische Programmiersprachen, Web- und Datenbankserver \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Desktop

- **GNOME Shell** wurde auf die Version 3.28 umgestellt.
- Die GNOME-Sitzung und der GNOME Display Manager verwenden **Wayland** als Standardanzeige-Server. Der **X.Org-Server**, der in RHEL 7 der Standardanzeige-Server ist, ist ebenfalls verfügbar.

Siehe [Abschnitt 4.8, »Desktop \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Installationsprogramm und Image-Erstellung

- Der **Anaconda-Installer** kann die **LUKS2-Festplattenverschlüsselung** verwenden und das System auf **NVDIMM-Geräten** installieren.
- Mit dem neuen **Composer-Tool** können Benutzer benutzerdefinierte Systemabbilder in einer Vielzahl von Formaten erstellen, einschließlich Bilder, die für den Einsatz in Clouds verschiedener Anbieter vorbereitet sind. Composer ist als *Technologievorschau* verfügbar.
- Installation von einer DVD mit Hardware Management Console (**HMC**) und Support Element (**SE**) auf **IBM Z**.

Siehe [Abschnitt 4.2, »Installationsprogramm und Image-Erstellung \(Maschinenübersetzung\)«](#) für weitere Details.

## Dateisysteme und Speicher

- Der lokale Speichermanager **Stratis** wurde eingeführt. Stratis ermöglicht es Ihnen, komplexe Speicheraufgaben einfach durchzuführen und Ihren Speicherstapel über eine einheitliche Schnittstelle zu verwalten.
- Das Format LUKS Version 2 (**LUKS2**) ersetzt das bisherige Format LUKS (LUKS1). Das **dm-crypt**Subsystem und das **cryptsetup**Tool verwenden nun LUKS2 als Standardformat für verschlüsselte Datenträger.

Siehe [Abschnitt 4.11, »Dateisysteme und Speicher \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Sicherheit

- Systemweite **kryptografische Richtlinien**, die die wichtigsten kryptografischen Subsysteme konfigurieren, die die Protokolle TLS, IPsec, SSH, DNSSEC und Kerberos abdecken, werden standardmäßig angewendet. Mit dem neuen **update-crypto-policies**Befehl kann der Administrator ganz einfach zwischen den Modi Standard, Legacy, Future und Fips wechseln.
- Die Unterstützung von **Smart Cards** und Hardware Security Modules (**HSM**) mit **PKCS #11** ist nun systemweit einheitlich.

Siehe [Abschnitt 4.14, »Sicherheit \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Vernetzung

- Das **nftables**Framework ersetzt **iptables**in der Rolle der Standard-Netzwerk-Paketfilterfunktion.
- Der **firewalld**Daemon verwendet **nftables**nun als Standard-Backend.
- Unterstützung für virtuelle **IPVLAN-Netztreiber**, die die Netzwerkkonnektivität für mehrere Container ermöglichen.

Siehe [Abschnitt 4.13, »Vernetzung \(Maschinenübersetzung\)«](#) für weitere Details.

## Virtualisierung

- Ein modernerer PCI Express-basierter Maschinentyp (**Q35**) wird nun unterstützt und automatisch in virtuellen Maschinen konfiguriert, die in RHEL 8 erstellt wurden. Dies bietet eine Vielzahl von Verbesserungen in den Funktionen und der Kompatibilität virtueller Geräte.
- Virtuelle Maschinen können nun über die RHEL 8 Webkonsole, auch bekannt als **Cockpit**, erstellt und verwaltet werden.
- Der **QEMU-Emulator** führt die **Sandboxing-Funktion** ein, die konfigurierbare Einschränkungen dessen bietet, was QEMU als Systeme ausführen kann, und so virtuelle Maschinen sicherer macht.

Siehe [Abschnitt 4.15, »Virtualisierung \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Compiler und Entwicklungswerkzeuge

- Der **GCC-Compiler** auf Basis der Version 8.2 bietet Unterstützung für neuere C++-Sprachstandard-Versionen, bessere Optimierungen, neue Code-Härtetechniken, verbesserte Warnungen und neue Hardware-Features.
- Verschiedene Werkzeuge zur Codegenerierung, -manipulation und -debugging können nun experimentell mit dem **DWARF5** Debugginginformationsformat umgehen.
- Kernel-Unterstützung für **eBPF-Tracing** ist für einige Tools verfügbar, wie z.B. **BCC**, **PCP**, und **SystemTap**.
- Die **glibc**Bibliotheken, die auf Version 2.28 basieren, bieten Unterstützung für Unicode 11, neuere Linux-Systemaufrufe, wichtige Verbesserungen im DNS-Stub-Resolver, zusätzliche Sicherheitsvorkehrungen und verbesserte Leistung.

Siehe [Abschnitt 4.10, »Compiler und Entwicklungswerkzeuge \(Maschinenübersetzung\)«](#) für weitere Details.

## Hochverfügbarkeit und Cluster

- Der Cluster-Ressourcenmanager von **Pacemaker** wurde auf die Upstream-Version 2.0.0 aktualisiert, die eine Reihe von Bugfixes und Verbesserungen bietet.
- In RHEL 8 unterstützt das **pcs-Konfigurationssystem** Corosync 3, **knet**, und Knotennamen vollständig.

Siehe [Abschnitt 4.12, »Hochverfügbarkeit und Cluster \(Maschinenübersetzung\)«](#) für weitere Informationen.

## Zusätzliche Ressourcen

- Die Möglichkeiten und Grenzen von Red Hat Enterprise Linux 8.0 im Vergleich zu anderen Versionen des Systems sind im Knowledgebase-Artikel [Red Hat Enterprise Linux technology capabilities and limits](#) verfügbar.
- Informationen zum Lebenszyklus von Red Hat Enterprise Linux finden Sie im [Red Hat Enterprise Linux Life Cycle](#) Dokument.
- Das [Package manifest](#) Dokument enthält eine Paketauflistung für RHEL 8.

## KAPITEL 2. ARCHITEKTUREN (MASCHINENÜBERSETZUNG)

Red Hat Enterprise Linux 8.0 wird mit der Kernelversion 4.18 ausgeliefert, die Unterstützung für die folgenden Architekturen bietet:

- AMD- und Intel 64-Bit-Architekturen
- Die 64-Bit-ARM-Architektur
- IBM Stromversorgungssysteme, Little Endian
- IBM Z

## KAPITEL 3. VERTEILUNG VON INHALTEN IN RHEL 8 (MASCHINENÜBERSETZUNG)

### 3.1. INSTALLATION (MASCHINENÜBERSETZUNG)

Red Hat Enterprise Linux 8 wird mit ISO-Images installiert. Zwei Arten von ISO-Images sind für die AMD64-, Intel 64-Bit-, 64-Bit-ARM-, IBM Power Systems- und IBM Z-Architekturen verfügbar:

- Binary DVD ISO: Ein vollständiges Installationsabbild, das die Repositories BaseOS und AppStream enthält und es Ihnen ermöglicht, die Installation ohne zusätzliche Repositories abzuschließen.
- Boot ISO: Ein minimales Boot-ISO-Image, das zum Booten in das Installationsprogramm verwendet wird. Diese Option erfordert den Zugriff auf die Repositories BaseOS und AppStream, um Softwarepakete zu installieren. Die Repositories sind Teil des Binary DVD ISO-Images.

Im [Performing a standard RHEL installation](#) Dokument finden Sie Anweisungen zum Herunterladen von ISO-Images, zum Erstellen von Installationsmedien und zum Abschluss einer RHEL-Installation. Für automatisierte Kickstartinstallationen und andere fortgeschrittene Themen lesen Sie bitte das [Performing an advanced RHEL installation](#) Dokument.

### 3.2. REPOSITORIEN (MASCHINENÜBERSETZUNG)

Red Hat Enterprise Linux 8 wird über zwei Repositories verteilt:

- BaseOS
- AppStream

Beide Repositories werden für eine RHEL-Grundinstallation benötigt und sind mit allen RHEL-Abonnements verfügbar.

Der Inhalt im BaseOS-Repository soll den Kernsatz der zugrunde liegenden Betriebssystemfunktionalität liefern, der die Grundlage für alle Installationen bildet. Dieser Inhalt ist im RPM-Format verfügbar und unterliegt ähnlichen Supportbedingungen wie in früheren Releases von RHEL. Eine Liste der über BaseOS vertriebenen Pakete finden Sie im [Package manifest](#).

Die Inhalte im Application Stream Repository umfassen zusätzliche Userspace-Anwendungen, Laufzeitsprachen und Datenbanken zur Unterstützung der unterschiedlichen Workloads und Anwendungsfälle. Inhalte in AppStream sind in einem von zwei Formaten verfügbar - dem bekannten RPM-Format und einer Erweiterung des RPM-Formats namens *Module*. Eine Liste der in AppStream verfügbaren Pakete finden Sie im [Package manifest](#).

Darüber hinaus ist das CodeReady Linux Builder Repository mit allen RHEL-Abonnements verfügbar. Es stellt zusätzliche Pakete für die Verwendung durch Entwickler bereit. Pakete, die im CodeReady Linux Builder Repository enthalten sind, werden für den produktiven Einsatz nicht unterstützt.

Weitere Informationen zu den RHEL 8-Repositories finden Sie im [Package manifest](#).

### 3.3. ANWENDUNGS-STREAMS (MASCHINENÜBERSETZUNG)

Red Hat Enterprise Linux 8.0 führt das Konzept der Application Streams ein. Mehrere Versionen von Userspace-Komponenten werden nun häufiger ausgeliefert und aktualisiert als die

Kernbetriebssystempakete. Dies bietet eine größere Flexibilität bei der Anpassung von Red Hat Enterprise Linux, ohne die grundlegende Stabilität der Plattform oder bestimmter Implementierungen zu beeinträchtigen.

Komponenten, die als Application Streams zur Verfügung gestellt werden, können als Module oder RPM-Pakete verpackt werden und werden über das AppStream-Repository in RHEL 8 bereitgestellt. Jede Komponente des Anwendungsstroms hat einen bestimmten Lebenszyklus. Weitere Informationen finden Sie unter [Red Hat Enterprise Linux Life Cycle](#) .

Module sind Sammlungen von Paketen, die eine logische Einheit darstellen: eine Anwendung, ein Sprachstapel, eine Datenbank oder eine Reihe von Werkzeugen. Diese Pakete werden gemeinsam erstellt, getestet und freigegeben.

Modulströme stellen Versionen der Anwendungsstromkomponenten dar. So sind beispielsweise zwei Streams (Versionen) des PostgreSQL-Datenbankservers im postgresql-Modul verfügbar: PostgreSQL 10 (der Standard-Stream) und PostgreSQL 9.6. Es kann nur ein Modulstrom auf dem System installiert werden. Verschiedene Versionen können in separaten Behältern verwendet werden.

Detaillierte Modulbefehle sind im [Using Application Stream](#) Dokument beschrieben. Eine Liste der in AppStream verfügbaren Module finden Sie im [Package manifest](#).

## KAPITEL 4. NEUE FUNKTIONEN (MASCHINENÜBERSETZUNG)

Dieser Teil beschreibt neue Funktionen und wichtige Verbesserungen, die in Red Hat Enterprise Linux 8 eingeführt wurden.

### 4.1. DIE WEB-KONSOLE (MASCHINENÜBERSETZUNG)



#### ANMERKUNG

Die Seite Abonnements der Webkonsole wird nun durch das neue **subscription-manager-cockpit** Paket bereitgestellt.

#### Die Webkonsole wurde um eine Firewall-Schnittstelle erweitert

Die Seite **Networking** in der RHEL 8 Webkonsole enthält nun auch einen **Firewall-Bereich**. In diesem Abschnitt können Benutzer die Firewall aktivieren oder deaktivieren sowie Firewall-Regeln hinzufügen, entfernen und ändern.

(BZ#1647110)

#### Die Webkonsole ist nun standardmäßig verfügbar

Pakete für die RHEL 8 Webkonsole, auch bekannt als Cockpit, sind nun Teil der Red Hat Enterprise Linux Standard-Repositories und können daher sofort auf einem registrierten RHEL 8-System installiert werden.

Darüber hinaus wird bei einer nicht-minimalen Installation von RHEL 8 die Webkonsole automatisch installiert und die von der Konsole benötigten Firewall-Ports werden automatisch geöffnet. Vor der Anmeldung wurde außerdem eine Systemmeldung hinzugefügt, die Informationen darüber enthält, wie Sie die Webkonsole aktivieren oder darauf zugreifen können.

(JIRA:RHELPLAN-10355)

#### Bessere IdM-Integration für die Web-Konsole

Wenn Ihr System in einer Identity Management (IdM)-Domäne registriert ist, verwendet die RHEL 8-Webkonsole nun standardmäßig die zentral verwalteten IdM-Ressourcen der Domäne. Dazu gehören die folgenden Vorteile:

- Die Administratoren der IdM-Domäne können die Webkonsole verwenden, um den lokalen Rechner zu verwalten.
- Der Webserver der Konsole wechselt automatisch zu einem Zertifikat, das von der IdM-Zertifizierungsstelle (CA) ausgestellt und von Browsern akzeptiert wird.
- Benutzer mit einem Kerberos-Ticket in der IdM-Domäne müssen für den Zugriff auf die Webkonsole keine Anmeldeinformationen angeben.
- SSH-Hosts, die der IdM-Domäne bekannt sind, sind für die Webkonsole zugänglich, ohne manuell eine SSH-Verbindung hinzuzufügen.

Beachten Sie, dass der Benutzer, damit die IdM-Integration mit der Webkonsole ordnungsgemäß funktioniert, zuerst das **ipa-advise** Dienstprogramm mit der **enable-admins-sudo** Option im IdM-Mastersystem ausführen muss.

(JIRA:RHELPLAN-3010)

### Die Webkonsole ist nun mit mobilen Browsern kompatibel

Mit diesem Update können die Menüs und Seiten der Webkonsole über mobile Browser-Varianten navigiert werden. Damit ist es möglich, Systeme über die RHEL 8 Webkonsole von einem mobilen Gerät aus zu verwalten.

(JIRA:RHELPLAN-10352)

### Die Titelseite der Webkonsole zeigt nun fehlende Updates und Abonnements an

Wenn ein von der RHEL 8 Webkonsole verwaltetes System veraltete Pakete oder ein abgelaufenes Abonnement hat, wird nun eine Warnung auf der Titelseite der Webkonsole des Systems angezeigt.

(JIRA:RHELPLAN-10353)

### Die Webkonsole unterstützt nun die PBD-Anmeldung

Mit diesem Update können Sie die RHEL 8 Webkonsolenschnittstelle verwenden, um PBD-Regeln (Policy-Based Decryption) auf Festplatten auf verwalteten Systemen anzuwenden. Dabei wird der Clevis-Entschlüsselungsclient verwendet, um eine Vielzahl von Sicherheitsmanagementfunktionen in der Webkonsole zu ermöglichen, wie z.B. die automatische Freischaltung von LUKS-verschlüsselten Festplattenpartitionen.

(JIRA:RHELPLAN-10354)

### Virtuelle Maschinen können nun über die Webkonsole verwaltet werden

Die **Virtual Machines** Seite kann nun der RHEL 8 Webkonsolenoberfläche hinzugefügt werden, die es dem Benutzer ermöglicht, libvirtbasierte virtuelle Maschinen zu erstellen und zu verwalten.

(JIRA:RHELPLAN-2896)

## 4.2. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)

### Die Installation von RHEL von einer DVD mit SE und HMC wird nun auf IBM Z vollständig unterstützt

Die Installation von Red Hat Enterprise Linux 8 auf IBM Z-Hardware von einer DVD mit dem **Support Element (SE)** und der **Hardware Management Console (HMC)** wird nun vollständig unterstützt. Dieser Zusatz vereinfacht den Installationsprozess auf IBM Z mit **SE** und **HMC**.

Wenn Sie von einer binären DVD booten, fordert das Installationsprogramm den Benutzer auf, zusätzliche Kernelparameter einzugeben. Um die DVD als Installationsquelle festzulegen, fügen **inst.repo=hmc** Sie sie an die Kernelparameter an. Der Installer ermöglicht dann den Zugriff auf **SE-** und **HMC-Dateien**, holt die Images für stage2 von der DVD und ermöglicht den Zugriff auf die Pakete auf der DVD zur Softwareauswahl.

Die neue Funktion erübrigt die Notwendigkeit einer externen Netzwerkeinrichtung und erweitert die Installationsoptionen.

(BZ#1500792)

### Der Installer unterstützt nun das LUKS2 Festplattenverschlüsselungsformat

Das Installationsprogramm von Red Hat Enterprise Linux 8 verwendet nun standardmäßig das LUKS2-



Format, aber Sie können eine LUKS-Version aus dem Fenster Benutzerdefinierte Partitionierung von **Anaconda** auswählen oder die neuen Optionen in den **RAID**Befehlen Kickstart**autopart,logvol,,part,,** und Kickstart verwenden.

LUKS2 bietet viele Verbesserungen und Funktionen, z.B. erweitert es die Möglichkeiten des On-Disk-Formats und bietet flexible Möglichkeiten zur Speicherung von Metadaten.

(BZ#1547908)

### Anaconda unterstützt Systemzweck in RHEL 8

Bisher hat **Anaconda** dem **Subscription Manager** keine Informationen über den Systemzweck zur Verfügung gestellt. In Red Hat Enterprise Linux 8.0 können Sie den Verwendungszweck des Systems während der Installation mit dem **System Purpose**Fenster von **Anaconda** oder dem **syspurpose**Befehl von Kickstart festlegen. Wenn die Installation abgeschlossen ist, verwendet der **Subscription Manager** die Informationen zum Systemzweck, wenn er das System abonniert.

(BZ#1612060)

### Pykickstart unterstützt Systemzweck in RHEL 8

Bisher war es der **pykickstart**Bibliothek nicht möglich, dem **Subscription Manager** Systeminformationen zur Verfügung zu stellen. In Red Hat Enterprise Linux 8.0 wird der neue **syspurpose**Befehl **pykickstart** analysiert und der Verwendungszweck des Systems während der automatisierten und teilautomatisierten Installation aufgezeichnet. Die Informationen werden dann an **Anaconda** weitergeleitet, auf dem neu installierten System gespeichert und stehen dem **Subscription Manager** bei der Anmeldung des Systems zur Verfügung.

(BZ#1612061)

### Anaconda unterstützt einen neuen Kernel-Bootparameter in RHEL 8

Bisher konnten Sie ein Basis-Repository nur über die Bootparameter des Kernels angeben. In Red Hat Enterprise Linux 8, einem neuen Kernelparameter, **inst.addrepo=<name>,<url>** können Sie während der Installation ein zusätzliches Repository angeben.

Dieser Parameter hat zwei Pflichtwerte: den Namen des Repositories und die URL, die auf das Repository zeigt. Weitere Informationen finden Sie unter <https://anaconda-installer.readthedocs.io/en/latest/boot-options.html#inst-addrepo>

(BZ#1595415)

### Anaconda unterstützt eine einheitliche ISO in RHEL 8

In Red Hat Enterprise Linux 8.0 lädt ein vereinheitlichter ISO automatisch die Quell-Repositorys von BaseOS und AppStream.

Diese Funktion funktioniert für das erste Basis-Repository, das während der Installation geladen wird. Wenn Sie beispielsweise die Installation ohne Repository starten und das vereinheitlichte ISO als Basis-Repository in der GUI haben, oder wenn Sie die Installation mit der **inst.repo=**Option starten, die auf das vereinheitlichte ISO zeigt. Dadurch wird das AppStream-Repository unter dem Abschnitt **Zusätzliche Repositories** im Fenster **Installation Source** GUI aktiviert. Sie können das AppStream-Repository nicht entfernen oder seine Einstellungen ändern, aber Sie können es in der **Installationsquelle** deaktivieren. Diese Funktion funktioniert nicht, wenn Sie die Installation mit einem anderen Basis-Repository starten und dann auf das einheitliche ISO wechseln. Wenn Sie dies tun, wird das Basis-Repository ersetzt. Das AppStream-Repository wird jedoch nicht ersetzt und zeigt auf die Originaldatei.

(BZ#1610806)

## 4.3. KERNEL (MASCHINENÜBERSETZUNG)

### ARM 52-Bit physische Adressierung ist jetzt verfügbar

Mit diesem Update ist die Unterstützung von 52-Bit Physical Addressing (PA) für die 64-Bit-ARM-Architektur verfügbar. Dies bietet einen größeren Adressraum als die bisherige 48-Bit-PA.

(BZ#1643522)

### Der IOMMU-Code unterstützt 5-stufige Seitentabellen in RHEL 8

Der IOMMU-Code (I/O Memory Management Unit) im Linux-Kernel wurde aktualisiert, um 5-stufige Seitentabellen in Red Hat Enterprise Linux 8 zu unterstützen.

(BZ#1485546)

### Unterstützung für 5-stufiges Paging

Der Linux-Kernel wurde um einen neuen **P4d\_t**Software-Seiten-Tabellentyp erweitert, um 5-Level-Paging in Red Hat Enterprise Linux 8 zu unterstützen.

(BZ#1485532)

### Die Speicherverwaltung unterstützt 5-stufige Seitentabellen

Mit Red Hat Enterprise Linux 7 verfügte der vorhandene Speicherbus über 48/46 Bit virtueller/physischer Speicheradressierungskapazität, und der Linux-Kernel implementierte 4 Ebenen von Seitentabellen, um diese virtuellen Adressen an physische Adressen zu verwalten. Die physikalische Busadressierungsleitung setzt die Obergrenze für den physikalischen Speicher auf 64 TB.

Diese Grenzen wurden auf 57/52 Bit virtueller/physikalischer Speicheradressierung mit 128 PiB virtuellem Adressraum und 4 PB physikalischer Speicherkapazität erweitert.

Mit dem erweiterten Adressbereich bietet die Speicherverwaltung in Red Hat Enterprise Linux 8 Unterstützung für die Implementierung von 5-stufigen Seitentabellen, um den erweiterten Adressbereich bewältigen zu können.

(BZ#1485525)

### kernel-signing-ca.cer wird kernel-corein RHEL 8 verschoben

In allen Versionen von Red Hat Enterprise Linux 7 befand sich der **kernel-signing-ca.cer** öffentliche Schlüssel im **kernel-doc** Paket. Allerdings wurde in Red Hat Enterprise Linux 8, **kernel-signing-ca.cer** wurde für jede Architektur in das **kernel-core** Paket verschoben.

(BZ#1638465)

### bpftool zur Überprüfung und Manipulation von eBPF-basierten Programmen und Karten hinzugefügt

Das **bpftool** Dienstprogramm, das der Inspektion und einfachen Manipulation von Programmen und Maps auf Basis des erweiterten Berkeley Packet Filtering (eBPF) dient, wurde in den Linux-Kernel aufgenommen. **bpftool** Es ist Teil des Kernel-Quellbaums und wird vom **bpftool-Paket** bereitgestellt, das als Teilpaket des **Kernelpakets** enthalten ist.

(BZ#1559607)

### Die kernel-rtQuellen wurden aktualisiert

Die **kernel-rt**Quellen wurden aktualisiert, um den neuesten RHEL-Kernel-Quellbaum zu verwenden. Der neueste Kernel-Source-Tree verwendet nun das Upstream v4.18 Echtzeit-Patch-Set, das eine Reihe von Bugfixes und Verbesserungen gegenüber der Vorgängerversion bietet.

(BZ#1592977)

## 4.4. SOFTWARE-MANAGEMENT (MASCHINENÜBERSETZUNG)

### YUM Leistungssteigerung und Unterstützung für modulare Inhalte

Auf Red Hat Enterprise Linux 8 wird die Softwareinstallation durch die neue Version des **YUM-Tools** gewährleistet, das auf der **DNF-Technologie** basiert.

**YUM** auf **DNF-Basis** hat folgende Vorteile gegenüber dem bisherigen **YUM v3** auf RHEL 7:

- Leistungssteigerung
- Unterstützung für modulare Inhalte
- Gut durchdachte, stabile API zur Integration mit Werkzeugen

Detaillierte Informationen über die Unterschiede zwischen dem neuen **YUM-Tool** und der Vorgängerversion **YUM v3** von RHEL 7 finden Sie unter [http://dnf.readthedocs.io/en/latest/cli\\_vs\\_yum.html](http://dnf.readthedocs.io/en/latest/cli_vs_yum.html).

**YUM** basierend auf **DNF** ist kompatibel mit **YUM v3**, wenn Sie von der Befehlszeile aus arbeiten, Konfigurationsdateien bearbeiten oder erstellen.

Für die Installation von Software können Sie den **yum**Befehl und seine einzelnen Optionen wie bei RHEL 7 verwenden. Pakete können unter den bisherigen Namen mit **Provides**. Pakete bieten auch Kompatibilitäts-Symlinks, so dass die Binärdateien, Konfigurationsdateien und Verzeichnisse an üblichen Stellen zu finden sind.

Beachten Sie, dass die von **YUM v3** bereitgestellte alte Python-API und die Libdnf C-API instabil sind und sich während des Lebenszyklus von Red Hat Enterprise Linux 8 wahrscheinlich ändern werden. Benutzern wird empfohlen, ihre Plugins und Skripte auf die neue DNF Python API zu migrieren, die stabil und vollständig unterstützt wird. Die DNF Python API ist verfügbar unter <https://dnf.readthedocs.io/en/latest/api.html>

Einige der **YUM v3-Funktionen** können sich in **YUM** basierend auf **DNF** unterschiedlich verhalten. Wenn sich eine solche Änderung negativ auf Ihre Arbeitsabläufe auswirkt, öffnen Sie bitte einen Fall mit Red Hat Support, wie unter [How do I open and manage a support case on the Customer Portal?](#)

(BZ#1581198)

### Bemerkenswerte Drehzahlmerkmale in RHEL 8

Red Hat Enterprise Linux 8 wird mit RPM 4.14 ausgeliefert. Diese Version enthält viele Verbesserungen gegenüber RPM 4.11, das in RHEL 7 verfügbar ist. Zu den bemerkenswertesten Merkmalen gehören:

- Die **debuginfo**Pakete können parallel installiert werden
- Unterstützung bei schwachen Abhängigkeiten
- Unterstützung von reichen oder booleschen Abhängigkeiten
- Unterstützung für das Paketieren von Dateien mit einer Größe von mehr als 4 GB

- Unterstützung von Datei-Triggern

Zu den wichtigsten Änderungen gehören auch:

- Strengerer Spec-Parser
- Vereinfachte Signatur zur Überprüfung der Ausgabe im non-verbosen Modus
- Zu- und Abschreibung in Makros

(BZ#1581990)

### RPM validiert nun den gesamten Paketinhalt, bevor mit der Installation begonnen wird

Auf Red Hat Enterprise Linux 7 verifizierte das **RPM-Dienstprogramm** beim Entpacken den Inhalt der Nutzlast einzelner Dateien. Dies ist jedoch aus mehreren Gründen nicht ausreichend:

- Wenn die Payload beschädigt ist, wird sie erst nach der Ausführung von Skript-Aktionen bemerkt, die irreversibel sind.
- Wenn die Nutzlast beschädigt ist, bricht das Upgrade eines Pakets ab, nachdem einige Dateien der vorherigen Version ersetzt wurden, was eine funktionierende Installation unterbricht.
- Die Hashes für einzelne Dateien werden auf unkomprimierten Daten ausgeführt, was die **RPM** anfällig für Dekompressorschwachstellen macht.

Bei Red Hat Enterprise Linux 8 wird das gesamte Paket vor der Installation in einem separaten Schritt mit dem besten verfügbaren Hash validiert.

Pakete, die auf Red Hat Enterprise Linux 8 basieren, verwenden einen neuen **SHA256**Hash auf der komprimierten Nutzlast. Bei signierten Paketen ist der Payload-Hash zusätzlich durch die Signatur geschützt und kann daher nicht geändert werden, ohne eine Signatur und andere Hashes auf dem Paketkopf zu brechen. Ältere Pakete verwenden den **MD5**Hash des Headers und der Payload, es sei denn, er wird durch die Konfiguration deaktiviert, z.B. im FIPS-Modus.

Mit dem Makro ``%_pkgverify_level`` kann zusätzlich die Signaturprüfung vor der Installation erzwungen oder die Nutzlastverifizierung vollständig deaktiviert werden. Darüber hinaus kann das **%\_pkgverify\_flags**Makro verwendet werden, um einzuschränken, welche Hashes und Signaturen erlaubt sind. So ist es beispielsweise möglich, die Verwendung des schwachen **MD5**Hash auf Kosten der Kompatibilität mit älteren Paketen zu deaktivieren.

(JIRA:RHELPLAN-1499)

## 4.5. INFRASTRUKTURDIENSTLEISTUNGEN (MASCHINENÜBERSETZUNG)

### Bemerkenswerte Änderungen am empfohlenen Tuning-Profil in RHEL 8

Mit diesem Update wird nun das empfohlene Tuned-Profil (das vom **tuned-adm recommend**Befehl gemeldet wird) nach den folgenden Regeln ausgewählt - die erste Regel, die übereinstimmt, wird wirksam:

- Wenn die **syspurpose**Rolle (die durch den **syspurpose show**Befehl gemeldet wird) enthält **atomic**, und zwar gleichzeitig:
  - wenn Tuned auf blankem Metall läuft, wird das **atomic-host**Profil ausgewählt

- wenn Tuned in einer virtuellen Maschine läuft, wird das **atomic-guest** Profil ausgewählt
- Wenn Tuned in einer virtuellen Maschine läuft, wird das **virtual-guest** Profil ausgewählt
- Wenn die **syspurpose** Rolle **desktop** oder **workstation** enthält und der Fahrgestelltyp (gemeldet von **dmidecode**) **Notebook**, **Laptop**, oder **Portable**, ist, dann wird das **balanced** Profil ausgewählt
- Wenn keine der obigen Regeln zutrifft, wird das **throughput-performance** Profil ausgewählt

(BZ#1565598)

### Die von **named** erzeugten Dateien können in das Arbeitsverzeichnis geschrieben werden

Bisher hat der **benannte** Daemon einige Daten im Arbeitsverzeichnis gespeichert, das in Red Hat Enterprise Linux schreibgeschützt ist. Mit diesem Update wurden die Pfade für ausgewählte Dateien in Unterverzeichnisse geändert, in denen das Schreiben erlaubt ist. Nun können die Unix- und SELinux-Berechtigungen des Standardverzeichnisses das Schreiben in das Verzeichnis erlauben. Dateien, die innerhalb des Verzeichnisses verteilt werden, sind weiterhin schreibgeschützt und **benannt**.

(BZ#1588592)

### Geolite-Datenbanken wurden durch Geolite2-Datenbanken ersetzt

Geolite-Datenbanken, die in Red Hat Enterprise Linux 7 vorhanden waren, wurden durch Geolite2-Datenbanken in Red Hat Enterprise Linux 8 ersetzt.

Geolite-Datenbanken wurden vom **GeoIP** Paket zur Verfügung gestellt. Dieses Paket wird zusammen mit der Legacy-Datenbank im Upstream nicht mehr unterstützt.

Geolite2 Datenbanken werden von mehreren Paketen bereitgestellt. Das **libmaxminddb** Paket enthält die Bibliothek und das **mmdblookup** Kommandozeilen-Tool, das die manuelle Suche nach Adressen ermöglicht. Die **geoipupdate** Binärdatei aus dem Legacy-Paket **GeoIP** wird nun vom **geoipupdate** Paket bereitgestellt und kann sowohl Legacy-Datenbanken als auch die neuen Geolite2-Datenbanken herunterladen.

(JIRA:RHELPLAN-6746)

### CUPS-Protokolle werden von **journald** verwaltet

In RHEL 8 werden die CUPS-Protokolle nicht mehr in bestimmten Dateien innerhalb des **/var/log/cups** Verzeichnisses gespeichert, das in RHEL 7 verwendet wurde. In RHEL 8 werden alle Arten von CUPS-Protokollen zentral im Systemdämon **journald** zusammen mit Protokollen aus anderen Programmen protokolliert. Um auf die CUPS-Protokolle zuzugreifen, verwenden Sie den **journalctl -u cups** Befehl. Weitere Informationen finden Sie unter [Arbeiten [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html/single/deploying\\_different\\_types\\_of\\_servers/working-with-cups-logs\\_configuring-printing](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/single/deploying_different_types_of_servers/working-with-cups-logs_configuring-printing) mit CUPS-Protokollen].

(JIRA:RHELPLAN-12764)

## 4.6. SHELLS UND BEFEHLSZEILEN-TOOLS (MASCHINENÜBERSETZUNG)

### Der **nobody** Benutzer ersetzt **nfsnobody**

In Red Hat Enterprise Linux 7 gab es:

- das **nobody**Benutzer- und Gruppenpaar mit der ID 99 und
- das **nfsnobody**Benutzer- und Gruppenpaar mit der ID 65534, die auch die Standard-Überlauf-ID des Kernels ist.

Beide wurden in das **nobody**Benutzer- und Gruppenpaar zusammengeführt, das die 65534-ID in Red Hat Enterprise Linux 8 verwendet. Neue Installationen erzeugen das **nfsnobody**Paar nicht mehr.

Diese Änderung reduziert die Verwirrung über Dateien, die Eigentum von NFS sind, **nobody**aber nichts mit NFS zu tun haben.

(BZ#1591969)

## Versionskontrollsysteme in RHEL 8

RHEL 8 bietet die folgenden Versionskontrollsysteme an:

- **Git 2.18**ein verteiltes Revisionskontrollsystem mit einer dezentralen Architektur.
- **Mercurial 4.8**ist ein leichtgewichtiges, verteiltes Versionskontrollsystem, das für die effiziente Abwicklung großer Projekte entwickelt wurde.
- **Subversion 1.10**, ein zentralisiertes Versionskontrollsystem. Weitere Informationen finden Sie unter [Notable changes in Subversion 1.10](#).

Beachten Sie, dass das in RHEL 7 verfügbare Concurrent Versions System (CVS) nicht mit RHEL 8 verteilt wird.

(BZ#1693775)

## Bemerkenswerte Änderungen in der Subversion 1.10

**Subversion 1.10** führt seit der in RHEL 7 vertriebenen Version 1.7 eine Reihe von neuen Funktionen ein, sowie die folgenden Kompatibilitätsänderungen:

- Aufgrund von Inkompatibilitäten in den **Subversion**Bibliotheken, die für die Unterstützung von Sprachbindungen verwendet werden, sind **Python 3**Bindungen für **Subversion 1.10**diese nicht verfügbar. Infolgedessen sind Anwendungen, für **Subversion**die **Python**Bindungsanforderungen bestehen, nicht unterstützt.
- Repositories, die auf **Berkeley DB**Repositories basieren, werden nicht mehr unterstützt. Sichern Sie vor der Migration Repositories, die mit **Subversion 1.7**dem **svnadmin dump**Befehl. Nach der Installation von RHEL 8 stellen Sie die Repositories mit dem **svnadmin load**Befehl wieder her.
- Bestehende Arbeitskopien, die vom **Subversion 1.7**Kunden in RHEL 7 ausgecheckt wurden, müssen auf das neue Format aktualisiert werden, bevor sie verwendet werden können **Subversion 1.10**. Nach der Installation von RHEL 8 führen Sie den **svn upgrade**Befehl in jeder Arbeitskopie aus.
- Die Smartcard-Authentifizierung für den Zugriff auf Repositories über die Verwendung **https://**von Smartcards wird nicht mehr unterstützt.

(BZ#1571415)

## 4.7. DYNAMISCHE PROGRAMMIERSPRACHEN, WEB- UND DATENBANKSERVER (MASCHINENÜBERSETZUNG)

## Python 3 ist die Standardimplementierung Python in RHEL 8

Red Hat Enterprise Linux 8 wird verteilt mit **Python 3.6**. Das Paket ist standardmäßig nicht installiert. Zur Installation **Python 3.6** verwenden Sie den **yum install python3** Befehl.

**Python 2.7** ist im **python2** Paket enthalten. Allerdings **Python 2** wird ein kürzerer Lebenszyklus haben und sein Ziel ist es, einen reibungsloseren Übergang zu **Python 3** den Kunden zu erleichtern.

Weder das Standardpaket **python** noch die nicht versionierte **/usr/bin/python** ausführbare Datei werden mit RHEL 8 ausgeliefert. Den Kunden wird empfohlen, diese zu nutzen **python3** oder **python2** direkt zu verwenden. Alternativ können Administratoren den nicht versionierten **python** Befehl mit dem **alternatives** Befehl konfigurieren.

Weitere Informationen finden Sie unter [Using Python in Red Hat Enterprise Linux 8](#).

(BZ#1580387)

## Python-Skripte müssen die Hauptversion in Hashbangs zur RPM-Buildzeit angeben

In RHEL 8 wird von ausführbaren Python-Skripten erwartet, dass sie Hashbangs (Shebangs) verwenden, die explizit mindestens die Hauptversion von Python angeben.

Das **/usr/lib/rpm/redhat/brp-mangle-shebangs** Buildroot Policy (BRP)-Skript wird automatisch ausgeführt, wenn ein RPM-Paket erstellt wird. Dieses Skript versucht, Hashbangs in allen ausführbaren Dateien zu korrigieren. Wenn das Skript auf mehrdeutige Hashbangs stößt, die nicht in eine größere Version von Python aufgelöst werden können, erzeugt es Fehler. Beispiele für solche mehrdeutigen Hash-Bangs sind:

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

Um Hashbangs in den Python-Skripten zu ändern, die diese Build-Fehler zur RPM-Buildzeit verursachen, verwenden Sie das **pathfix.py** Skript aus dem **Platform-python-devel** Paket:

```
pathfix.py -pn -i %(__python3) PATH ...
```

Es können mehrere *PATHs* angegeben werden. Wenn ein *PATH* ein Verzeichnis ist, wird **pathfix.py** rekursiv nach Python-Skripten gesucht, die dem Muster **^[a-zA-Z0-9\_]+\.py\$** entsprechen, nicht nur nach solchen mit einem mehrdeutigen Hashbang. Fügen Sie diesen Befehl dem **%prep** Abschnitt oder am Ende des **%install** Abschnitts hinzu.

Weitere Informationen finden Sie unter [Handling hashbangs in Python scripts](#).

(BZ#1583620)

## Bemerkenswerte Änderungen in der PHP

Red Hat Enterprise Linux 8 wird verteilt mit **PHP 7.2**. Diese Version führt die folgenden wesentlichen Änderungen ein **PHP 5.4**, die in RHEL 7 verfügbar sind:

- **PHP** verwendet standardmäßig den FastCGI Process Manager (FPM) (sicher für die Verwendung mit einem Gewinde **httpd**)
- Die Variablen **php\_value** und **php\_flags** sollten nicht mehr in den **httpd** Konfigurationsdateien verwendet werden, sondern in der Pool-Konfiguration: **/etc/php-fpm.d/\*.conf**

- **PHP** Skriptfehler und Warnungen werden in der `/var/log/php-fpm/www-error.log` Datei protokolliert, anstelle von `/var/log/httpd/error.log`
- Beim Ändern der `max_execution_time` PHP-Konfigurationsvariablen sollte die `httpd ProxyTimeout` Einstellung entsprechend erhöht werden
- Der Benutzer, der **PHP** Skripte ausführt, ist nun in der FPM-Pool-Konfiguration konfiguriert (die `/etc/php-fpm/d/www.conf` Datei; der `apache` Benutzer ist der Standard)
- Der `php-fpm` Dienst muss nach einer Konfigurationsänderung oder nach der Installation einer neuen Erweiterung neu gestartet werden

Die folgenden Erweiterungen wurden entfernt:

- **aspell**
- **mysql** (beachten Sie, dass die `mysqli` und `pdo_mysql` Erweiterungen noch verfügbar sind, die vom `php-mysqlnd` Paket zur Verfügung gestellt werden.)
- **zip**
- **memcache**

(BZ#1580430, [BZ#1691688](#))

## Bemerkenswerte Änderungen in der Ruby

RHEL 8 bietet **Ruby 2.5** zahlreiche neue Funktionen und Verbesserungen gegenüber **Ruby 2.0.0** RHEL 7. Zu den bemerkenswerten Änderungen gehören:

- Inkrementeller Garbage Collector wurde hinzugefügt.
- Die **Refinements** Syntax wurde hinzugefügt.
- Symbole werden nun als Garbage Collection bezeichnet.
- Die `SAFE=2` und die `SAFE=3` sicheren Stufen sind nun veraltet.
- Die **Fixnum** und **Bignum** Klassen wurden in der **Integer** Klasse vereinheitlicht.
- Die Leistung wurde durch die Optimierung der **Hash** Klasse, den verbesserten Zugriff auf Instanzvariablen und die kleinere und schnellere **Mutex** Klasse verbessert.
- Bestimmte alte APIs wurden veraltet.
- Die gebündelten Bibliotheken wie **RubyGems**, **Rake**, **RDoc**, **Psych**, **Minitest**, und **test-unit**, wurden aktualisiert.
- Andere Bibliotheken wie **mathn**, **DL**, **ext/tk**, und **XMLRPC**, die zuvor mit **Ruby** verteilt wurden, sind veraltet oder nicht mehr enthalten.
- Das **SemVer** Versionierungsschema wird nun für die **Ruby** Versionierung verwendet.

(BZ#1648843)

## Bemerkenswerte Änderungen in der Perl



**Perl 5.26**, vertrieben mit RHEL 8, führt die folgenden Änderungen gegenüber der in RHEL 7 verfügbaren Version ein:

- **Unicode 9.0** wird nun unterstützt.
- Es werden neue **op-entry**, **loading-file** und **loaded-file SystemTap** Sonden bereitgestellt.
- Der Kopier-/Schreibmechanismus wird bei der Zuweisung von Skalaren zur Leistungssteigerung verwendet.
- Das **IO::Socket::IP** Modul zur transparenten Handhabung von IPv4- und IPv6-Sockeln wurde hinzugefügt.
- Das **Config::Perl::V** Modul zum strukturierten Zugriff **perl -V** auf Daten wurde hinzugefügt.
- Ein neues **perl-App-cpanminus** Paket wurde hinzugefügt, das das **cpanm** Dienstprogramm zum Abrufen, Extrahieren, Erstellen und Installieren von Modulen aus dem CPAN-Repository (Comprehensive Perl Archive Network) enthält.
- Das aktuelle Verzeichnis `.` wurde aus Sicherheitsgründen aus dem **@INC** Modulsuchpfad entfernt.
- Die **do** Anweisung gibt nun eine Verfallswarnung zurück, wenn sie eine Datei aufgrund der oben beschriebenen Verhaltensänderung nicht lädt.
- Der **do subroutine(LIST)** Aufruf wird nicht mehr unterstützt und führt zu einem Syntaxfehler.
- Hashes sind jetzt standardmäßig randomisiert. Die Reihenfolge, in der Schlüssel und Werte aus einem Hash zurückgegeben werden, ändert sich bei jedem **perl** Durchlauf. Um die Randomisierung zu deaktivieren, setzen Sie die **PERL\_PERTURB\_KEYS** Umgebungsvariable auf **0**.
- Ungeformte literale `{` Zeichen in regulären Ausdrucksmodellen sind nicht mehr zulässig.
- Die Unterstützung des Lexikalbereichs für die **\$\_** Variable wurde entfernt.
- Die Verwendung des **defined** Operators auf einem Array oder einem Hash führt zu einem fatalen Fehler.
- Der Import von Funktionen aus dem **UNIVERSAL** Modul führt zu einem fatalen Fehler.
- Die Symbole **find2perl**, **s2p**, **a2p**, **c2ph**, und **pstruct** Werkzeuge wurden entfernt.
- Die **\${^ENCODING}** Anlage wurde entfernt. Der Standardmodus des **encoding** Pragma wird nicht mehr unterstützt. Um Quellcode in eine andere Codierung als **UTF-8** die, verwenden Sie die **Filter** Option der Codierung.
- Die **perl** Verpackung ist nun mit dem Upstream ausgerichtet. Das **perl** Paket installiert auch Kernmodule, während der **/usr/bin/perl** Interpreter vom **perl-interpreter** Paket bereitgestellt wird. In früheren Versionen enthielt das **perl** Paket nur einen minimalen Interpreter, während das **perl-core** Paket sowohl den Interpreter als auch die Kernmodule enthielt.

(BZ#1511131)

**Node.js neu in RHEL 8**

**Node.js** seine Software-Entwicklungsplattform zum Aufbau schneller und skalierbarer Netzwerkanwendungen in der Programmiersprache JavaScript, wird erstmals in RHEL angeboten. Sie war bisher nur als Software-Sammlung erhältlich. RHEL 8 bietet **Node.js 10**.

(BZ#1622118)

### Bemerkenswerte Änderungen in SWIG

RHEL 8 enthält den Simplified Wrapper and Interface Generator (SWIG) Version 3.0, der zahlreiche neue Funktionen, Verbesserungen und Bugfixes gegenüber der in RHEL 7 vertriebenen Version 2.0 bietet. Insbesondere wurde die Unterstützung des C++11-Standards implementiert. SWIG unterstützt nun auch Go 1.6, PHP 7, Octave 4.2 und Python 3.5.

(BZ#1660051)

### Bemerkenswerte Änderungen im Apache httpd

RHEL 8 wird mit dem Apache HTTP Server 2.4.37 ausgeliefert. Diese Version führt die folgenden Änderungen gegenüber **httpd** der in RHEL 7 verfügbaren Version ein:

- Die HTTP/2-Unterstützung wird nun vom **mod\_http2** Paket bereitgestellt, das Teil des **httpd** Moduls ist.
- Die automatisierte TLS-Zertifikatsbereitstellung und -erneuerung mit dem Protokoll Automatic Certificate Management Environment (ACME) wird nun vom **mod\_md** Paket unterstützt (zur Verwendung mit Zertifikatsanbietern wie z.B. **Let's Encrypt**)
- Der Apache HTTP Server unterstützt nun das Laden von TLS-Zertifikaten und privaten Schlüsseln aus Hardware-Sicherheitstoken direkt aus **PKCS#11** Modulen. Infolgedessen kann eine **mod\_ssl** Konfiguration nun **PKCS#11** URLs verwenden, um den privaten TLS-Schlüssel und optional das TLS-Zertifikat in den **SSLCertificateKeyFile** und **SSLCertificateFile** Direktiven zu identifizieren.
- Das Multi-Processing-Modul (MPM), das standardmäßig mit dem Apache HTTP Server konfiguriert ist, hat sich von einem Multi-Prozess-Gabelmodell (bekannt als **prefork**) zu einem leistungsstarken Multi-Thread-Modell gewandelt. **event** Alle Module von Drittanbietern, die nicht threadsicher sind, müssen ersetzt oder entfernt werden. Um das konfigurierte MPM zu ändern, bearbeiten Sie die **/etc/httpd/conf.modules.d/00-mpm.conf** Datei. Weitere Informationen finden Sie in der **httpd.conf(5)** Man Page.

Weitere Informationen zu **httpd** den folgenden Themen finden Sie unter [Setting up the Apache HTTP web server](#).

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

### Der nginx Webserver neu in RHEL 8

RHEL 8 stellt einen Web- und Proxy-Server vor **nginx 1.14**, der HTTP und andere Protokolle unterstützt, wobei der Schwerpunkt auf hoher Parallelität, Leistung und geringer Speicherauslastung liegt. Bisher **nginx** war er nur als Software Collection erhältlich.

Der **nginx** Webserver unterstützt nun das Laden von TLS-Privatschlüsseln aus Hardware-Sicherheitstoken direkt aus **PKCS#11** Modulen. Dadurch kann eine **nginx** Konfiguration URLs verwenden **PKCS#11**, um den privaten TLS-Schlüssel in der **ssl\_certificate\_key** Direktive zu identifizieren.

(BZ#1545526)

## Datenbankserver in RHEL 8

RHEL 8 stellt die folgenden Datenbankserver zur Verfügung:

- **MySQL 8.0**, ein Mehrbenutzer- und Multithread-SQL-Datenbank-Server. Es besteht aus dem **MySQL** Server-Daemon, **mysqld**, und vielen Client-Programmen.
- **MariaDB 10.3**, ein Mehrbenutzer- und Multithread-SQL-Datenbank-Server. Für alle praktischen Zwecke **MariaDB** ist es binärkompatibel mit **MySQL**.
- **PostgreSQL 10** und **PostgreSQL 9.6**, ein fortschrittliches objektrelationales Datenbankmanagementsystem (DBMS).
- **Redis 5**, ein erweiterter Key-Value-Speicher. Er wird oft als Datenstrukturserver bezeichnet, da Schlüssel Zeichenketten, Hashes, Listen, Sets und sortierte Sets enthalten können. **Redis** wird erstmals in RHEL bereitgestellt.

Beachten Sie, dass der NoSQL-Datenbankserver **MongoDB** in RHEL 8.0 nicht enthalten ist, da er die Server Side Public License (SSPL) verwendet.

(BZ#1647908)

## Bemerkenswerte Änderungen in der MySQL 8.0

RHEL 8 wird mit **MySQL 8.0** ausgeliefert, was z.B. die folgenden Erweiterungen beinhaltet:

- **MySQL** enthält nun ein Transaktionsdatendictionary, das Informationen über Datenbankobjekte speichert.
- **MySQL** unterstützt nun Rollen, die Sammlungen von Berechtigungen sind.
- Der Standardzeichensatz wurde von **latin1** auf **utf8mb4** geändert.
- Die Unterstützung für allgemeine Tabellenausdrücke, sowohl nicht rekursive als auch rekursive, wurde hinzugefügt.
- **MySQL** unterstützt nun Fensterfunktionen, die eine Berechnung für jede Zeile einer Query unter Verwendung von Bezugszeilen durchführen.
- **InnoDB** unterstützt nun die **NOWAIT** und **SKIP LOCKED** Optionen mit Sperren von Leseanweisungen.
- Die GIS-bezogenen Funktionen wurden verbessert.
- Die JSON-Funktionalität wurde erweitert.
- Die neuen **mariadb-connector-c** Pakete bieten eine gemeinsame Client-Bibliothek für **MySQL** und **MariaDB**. Diese Bibliothek ist mit jeder Version des **MySQLMariaDB** Datenbankservers verwendbar. Dadurch ist der Benutzer in der Lage, einen Build einer Anwendung mit einem der **MySQL** mit RHEL 8 verteilten **MariaDB** Server zu verbinden.

Darüber hinaus ist der mit RHEL 8 verteilte **MySQL 8.0** Server so konfiguriert, dass er als Standard-Authentifizierungs-Plugin verwendet **mysql\_native\_password** wird, da Client-Tools und -Bibliotheken in RHEL 8 mit der **caching\_sha2\_password** Methode, die standardmäßig in der Upstream-Version **MySQL 8.0** verwendet wird, nicht kompatibel sind.

Um das Standard-Authentifizierungs-Plugin zu **caching\_sha2\_password** ändern, bearbeiten Sie die **/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** Datei wie folgt:

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

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

## Bemerkenswerte Änderungen in der MariaDB 10.3

**MariaDB 10.3** bietet zahlreiche neue Funktionen gegenüber der in RHEL 7 verteilten Version 5.5. Einige der bemerkenswertesten Änderungen sind:

- **MariaDB Galera Cluster**, ein synchroner Multi-Master-Cluster, ist nun Standard in **MariaDB**.
- **InnoDB** wird als Standard-Speicher-Engine verwendet, anstatt von **XtraDB**.
- Allgemeine Tabellenausdrücke
- Systemversionierte Tabellen
- **FOR** schleifen
- Unsichtbare Spalten
- Sequenzen
- Sofort **ADD COLUMN** für **InnoDB**
- Speichermaschinenunabhängige Spaltenkompression
- Parallele Replikation
- Multi-Source-Replikation

Darüber hinaus bieten die neuen **mariadb-connector-c** Pakete eine gemeinsame Client-Bibliothek für **MySQL** und **MariaDB**. Diese Bibliothek ist mit jeder Version des **MySQLMariaDB** Datenbanksservers verwendbar. Dadurch ist der Benutzer in der Lage, einen Build einer Anwendung mit einem der **MySQL** mit RHEL 8 verteilten **MariaDB** Server zu verbinden.

Siehe auch [Using MariaDB on Red Hat Enterprise Linux 8](#) .

(BZ#1637034, BZ#1519450)

## 4.8. DESKTOP (MASCHINENÜBERSETZUNG)

### GNOME Shell, Version 3.28 in RHEL 8

GNOME Shell, Version 3.28 ist in Red Hat Enterprise Linux (RHEL) 8 verfügbar. Zu den bemerkenswerten Verbesserungen gehören:

- Neue Funktionen der GNOME-Boxen
- Neue Bildschirmtastatur
- Erweiterte Geräteunterstützung, vor allem die Integration für die Thunderbolt 3-Schnittstelle

- Verbesserungen für GNOME-Software, dconf-editor und GNOME-Terminal

(BZ#1649404)

## Wayland ist der Standardanzeigeserver

Mit Red Hat Enterprise Linux 8 verwenden die GNOME-Sitzung und der GNOME Display Manager (GDM) **Wayland** als Standardanzeige-Server anstelle des **X.org-Servers**, der mit der vorherigen Hauptversion von RHEL verwendet wurde.

**Wayland** bietet mehrere Vorteile und Verbesserungen gegenüber **X.org**. Vor allem aber

- Stärkeres Sicherheitsmodell
- Verbessertes Multi-Monitor-Handling
- Verbesserte Skalierung der Benutzeroberfläche (UI)
- Der Desktop kann die Fensterbehandlung direkt steuern.

Beachten Sie, dass die folgenden Funktionen derzeit nicht verfügbar sind oder nicht wie erwartet funktionieren:

- Multi-GPU-Setups werden unter **Wayland** nicht unterstützt.
- Der **NVIDIA-Binär-Treiber** funktioniert unter **Wayland** nicht.
- Das **xrandr**Dienstprogramm funktioniert unter **Wayland** nicht, da es unterschiedliche Ansätze für die Handhabung, Auflösungen, Rotationen und das Layout hat. Beachten Sie, dass auch andere **X.org-Dienstprogramme** zur Manipulation des Bildschirms unter **Wayland** nicht funktionieren.
- Bildschirmaufzeichnung, Remote-Desktop und Zugänglichkeit funktionieren unter **Wayland** nicht immer korrekt.
- Es ist kein Zwischenablage-Manager verfügbar.
- **Wayland** ignoriert Tastaturgreifer, die von X11-Anwendungen, wie z.B. Viewern virtueller Maschinen, ausgegeben werden.
- **Wayland** innerhalb von virtuellen Gastmaschinen (VMs) hat Stabilitäts- und Leistungsprobleme, daher wird empfohlen, die X11-Sitzung für virtuelle Umgebungen zu verwenden.

Wenn Sie von einem RHEL 7-System, in dem Sie die **X.org** GNOME-Sitzung verwendet haben, auf RHEL 8 aktualisieren, verwendet Ihr System weiterhin **X.org**. Das System greift auch automatisch auf **X.org** zurück, wenn die folgenden Grafiktreiber verwendet werden:

- Der NVIDIA-Binär-Treiber
- Der **cirrus**Fahrer
- Der **mga**Fahrer
- Der **aspeed**Fahrer

Du kannst die Nutzung von **Wayland** manuell deaktivieren:

- Um **Wayland** in **GDM** zu deaktivieren, setzen Sie die **WaylandEnable=false** Option in der **/etc/gdm/custom.conf** Datei.
- Um **Wayland** in der GNOME-Sitzung zu deaktivieren, wählen Sie nach Eingabe Ihres Anmeldenamens die Option Legacy X11 über das Zahnradmenü auf dem Anmeldebildschirm.

Für weitere Details zu **Wayland** siehe <https://wayland.freedesktop.org/>.

(BZ#1589678)

### Auffinden von RPM-Paketen, die sich in nicht standardmäßig aktivierten Repositories befinden

Zusätzliche Repositories für den Desktop sind standardmäßig nicht aktiviert. Die Deaktivierung wird durch die **enabled=0** Zeile in der entsprechenden **.repo** Datei angezeigt. Wenn Sie versuchen, ein Paket aus einem solchen Repository mit PackageKit zu installieren, zeigt PackageKit eine Fehlermeldung an, die anzeigt, dass die Anwendung nicht verfügbar ist. Um das Paket zur Verfügung zu stellen, ersetzen Sie die zuvor verwendete **enabled=0** Zeile in der jeweiligen **.repo** Datei durch **enabled=1**.

(JIRA:RHELPLAN-2878)

### GNOME-Software für die Paketverwaltung

Das **gnome-packagekit** Paket, das eine Sammlung von Tools für die Paketverwaltung in der grafischen Umgebung auf Red Hat Enterprise Linux 7 bereitstellte, ist nicht mehr verfügbar. Auf Red Hat Enterprise Linux 8 bietet das **GNOME-Software-Dienstprogramm** ähnliche Funktionen, mit denen Sie Anwendungen und gnome-shell-Erweiterungen installieren und aktualisieren können. Die **GNOME-Software** wird im **gnome-software** Paket mitgeliefert.

(JIRA:RHELPLAN-3001)

## 4.9. IDENTITÄTSMANAGEMENT (MASCHINENÜBERSETZUNG)

### Syntaxprüfungen für neue Passwörter im Directory Server

Diese Erweiterung fügt dem Directory Server neue Syntaxprüfungen für Passwörter hinzu. Administratoren können nun z.B. Dictionary-Prüfungen aktivieren, die Verwendung von Zeichenfolgen und Palindromen erlauben oder verweigern. Wenn diese Option aktiviert ist, erzwingt die Syntaxprüfung der Kennwortrichtlinien im Directory Server die Verwendung von sichereren Kennwörtern.

(BZ#1334254)

### Directory Server bietet jetzt eine verbesserte Unterstützung für die interne Betriebsaufzeichnung

Mehrere Operationen im Directory Server, die vom Server und den Clients initiiert werden, führen zu zusätzlichen Operationen im Hintergrund. Bisher hat der Server nur für interne Operationen das **Internal** Verbindungsschlüsselwort protokolliert, und die Operations-ID wurde immer auf **-1** gesetzt. Mit dieser Erweiterung protokolliert Directory Server die tatsächliche Verbindungs- und Betriebs-ID. Sie können nun die interne Operation auf den Server- oder Client-Betrieb zurückverfolgen, der diese Operation verursacht hat.

Weitere Details zur Protokollierung interner Operationen finden Sie unter dem Link: [https://access.redhat.com/documentation/en-us/red\\_hat\\_directory\\_server/11/html-single/administration\\_guide/#logging\\_internal\\_operations](https://access.redhat.com/documentation/en-us/red_hat_directory_server/11/html-single/administration_guide/#logging_internal_operations).

(BZ#1358706)

## Die tomcatjssBibliothek unterstützt die OCSP-Prüfung mit dem Responder der AIA-Erweiterung

Mit dieser Erweiterung unterstützt die **tomcatjss**Bibliothek die Überprüfung des Online Certificate Status Protocol (OCSP) unter Verwendung des Responders der Authority Information Access (AIA) Erweiterung eines Zertifikats. Dadurch können Administratoren des Red Hat Certificate System nun die OCSP-Prüfung konfigurieren, die die URL der AIA-Erweiterung verwendet.

(BZ#1636564)

## Directory Server stellt neue Befehlszeilen-Dienstprogramme zur Verwaltung von Instanzen vor

Red Hat Directory Server 11.0 führt die **dsctl**Dienstprogramme **dscreate**, **dsconf**, und ein. Diese Dienstprogramme vereinfachen die Verwaltung von Directory Server über die Befehlszeile. Beispielsweise können Sie nun mit einem Befehl mit Parametern eine Funktion konfigurieren, anstatt komplexe LDIF-Anweisungen an den Server zu senden.

Im Folgenden erhalten Sie einen Überblick über den Zweck der einzelnen Dienstprogramme:

- Verwenden Sie das **dscreate**Dienstprogramm, um neue Directory Server-Instanzen im interaktiven Modus oder in einer INF-Datei zu erstellen. Beachten Sie, dass sich das INF-Dateiformat von demjenigen unterscheidet, das das Installationsprogramm in früheren Directory Server-Versionen verwendet hat.
- Verwenden Sie das **dsconf**Dienstprogramm, um Directory Server-Instanzen während der Laufzeit zu verwalten. Verwenden **dsconf**Sie zum Beispiel:
  - Konfigurieren Sie die Einstellungen im **cn=config**Eintrag
  - Plug-Ins konfigurieren
  - Replikation konfigurieren
  - Sichern und Wiederherstellen einer Instanz
- Verwenden Sie das **dsctl**Dienstprogramm, um Directory Server-Instanzen zu verwalten, während sie offline sind. Verwenden **dsctl**Sie zum Beispiel:
  - Starten und Stoppen einer Instanz
  - Neuindizierung der Serverdatenbank
  - Sichern und Wiederherstellen einer Instanz

Diese Dienstprogramme ersetzen die Perl- und Shell-Skripte, die in Directory Server 10 als veraltet gekennzeichnet sind. Die Skripte sind weiterhin im nicht unterstützten **389-ds-base-legacy-tools**Paket verfügbar, jedoch unterstützt Red Hat nur die Verwaltung von Directory Server mit den neuen Dienstprogrammen.

Beachten Sie, dass die Konfiguration von Directory Server mit LDIF-Anweisungen weiterhin unterstützt wird, aber Red Hat empfiehlt die Verwendung der Dienstprogramme.

Weitere Informationen zur Verwendung der Hilfsmittel finden Sie im Abschnitt [Red Hat Directory Server 11 Documentation](#).

(BZ#1693159)

## Die Befehle `pki subsystem-cert-find` und `pki subsystem-cert-show` zeigen nun die Seriennummer der Zertifikate an

Mit dieser Erweiterung zeigen die Befehle `pki subsystem-cert-find` und `pki subsystem-cert-show` im Zertifikatssystem die Seriennummer der Zertifikate in ihrer Ausgabe an. Die Seriennummer ist eine wichtige Information und wird oft von mehreren anderen Befehlen benötigt. Dadurch wird die Identifizierung der Seriennummer eines Zertifikats vereinfacht.

(BZ#1566360)

## Die Befehle `pki user` und `pki group` wurden im Zertifikatssystem veraltet

Mit diesem Update ersetzen die neuen `pki <subsystem>-user` und `pki <subsystem>-group` Befehle die Befehle `pki user` und `pki group` im Zertifikatssystem. Die ersetzten Befehle funktionieren weiterhin, aber sie zeigen eine Meldung an, dass der Befehl veraltet ist und beziehen sich auf die neuen Befehle.

(BZ#1394069)

## Das Zertifikatssystem unterstützt nun die Offline-Verlängerung von Systemzertifikaten

Mit dieser Erweiterung können Administratoren die Offline-Verlängerung nutzen, um im Zertifikatssystem konfigurierte Systemzertifikate zu verlängern. Wenn ein Systemzertifikat abläuft, kann das Zertifikatssystem nicht gestartet werden. Durch die Erweiterung benötigen Administratoren keine Umgehungsmöglichkeiten mehr, um ein abgelaufenes Systemzertifikat zu ersetzen.

(BZ#1669257)

## Das Zertifikatssystem kann nun CSRs mit SKI-Erweiterung für die externe CA-Signatur erstellen

Mit dieser Erweiterung unterstützt das Zertifikatssystem die Erstellung einer Zertifikats-Signaturanforderung (CSR) mit der Erweiterung Subject Key Identifier (SKI) für die Signatur einer externen Zertifizierungsstelle (CA). Bestimmte CAs benötigen diese Erweiterung entweder mit einem bestimmten Wert oder abgeleitet vom öffentlichen Schlüssel der CA. Dadurch können Administratoren nun den `pki_req_ski` Parameter in der an das `pki spawn` Versorgungsunternehmen übergebenen Konfigurationsdatei verwenden, um einen CSR mit SKI-Erweiterung zu erstellen.

(BZ#1656856)

## Lokale Benutzer werden von SSSD zwischengespeichert und über das `nss_sss` Modul bedient

In RHEL 8 bedient der System Security Services Daemon (SSSD) standardmäßig Benutzer und Gruppen aus den `/etc/passwd` und `/etc/groups` Dateien. Das `sss` Modul geht Dateien in der `/etc/nsswitch.conf` Datei voraus.

Der Vorteil der Bedienung lokaler Benutzer über SSSD ist, dass das `nss_sss` Modul eine schnelle **memory-mapped cache**, beschleunigte Suche nach Name Service Switch (NSS) hat, verglichen mit dem Zugriff auf die Festplatte und dem Öffnen der Dateien bei jedem NSS-Request. Bisher half der Name Service Cache Daemon (`nscd`), den Prozess des Zugriffs auf die Festplatte zu beschleunigen. Die parallele Verwendung `nscd` mit SSSD ist jedoch umständlich, da sowohl SSSD als auch `nscd` ein eigenes unabhängiges Caching verwendet werden. Daher kann die Verwendung `nscd` in Setups, in denen SSSD auch Benutzer aus einer entfernten Domäne, z.B. LDAP oder Active Directory, bedient, zu unvorhersehbarem Verhalten führen.

Mit diesem Update ist die Auflösung von lokalen Benutzern und Gruppen in RHEL 8 schneller. Beachten Sie, dass der `root` Benutzer nie von SSSD behandelt wird, daher kann die `root` Auflösung nicht durch einen möglichen Fehler in SSSD beeinträchtigt werden. Beachten Sie auch, dass das `nss_sss` Modul,



wenn SSSD nicht ausgeführt wird, die Situation problemlos handhabt, indem es zurückgreift, **nss\_files** um Probleme zu vermeiden. Sie müssen SSSD in keiner Weise konfigurieren, die Datei-Domäne wird automatisch hinzugefügt.

(JIRA:RHELPLAN-10439)

### KCM ersetzt KEYRING als Standard-Cache-Speicher für Anmeldeinformationen

In RHEL 8 ist der standardmäßige Speicher für den Credential-Cache der Kerberos Credential Manager (KCM), der durch den **sssd-kcm** Daemon unterstützt wird. KCM überwindet die Einschränkungen des zuvor verwendeten SCHLÜSSELRINGS, wie z.B. die Schwierigkeit, ihn in containerisierten Umgebungen zu verwenden, da er nicht namensgesteuert ist, und um Quoten anzuzeigen und zu verwalten.

Mit diesem Update enthält RHEL 8 einen Credential-Cache, der besser für containerisierte Umgebungen geeignet ist und die Grundlage für den Aufbau weiterer Funktionen in zukünftigen Versionen bildet.

(JIRA:RHELPLAN-10440)

### Active Directory-Benutzer können nun Identitätsmanagement verwalten

Mit diesem Update ermöglicht RHEL 8 das Hinzufügen einer Übersteuerung der Benutzer-ID für einen Active Directory (AD)-Benutzer als Mitglied einer Identity Management (IdM)-Gruppe. Ein ID-Override ist ein Datensatz, der beschreibt, wie ein bestimmter AD-Benutzer oder eine bestimmte Gruppeneigenschaft innerhalb einer bestimmten ID-Ansicht, in diesem Fall der Standard-Trust-Ansicht, aussehen sollte. Als Folge des Updates ist der IdM-LDAP-Server in der Lage, Zugriffskontrollregeln für die IdM-Gruppe auf den AD-Benutzer anzuwenden.

AD-Benutzer können nun die Self-Service-Funktionen der IdM-Benutzeroberfläche nutzen, z.B. um ihre SSH-Schlüssel hochzuladen oder ihre persönlichen Daten zu ändern. Ein AD-Administrator ist in der Lage, IdM vollständig zu verwalten, ohne zwei verschiedene Konten und Passwörter zu haben. Beachten Sie, dass derzeit ausgewählte Funktionen in IdM für AD-Benutzer möglicherweise noch nicht verfügbar sind.

(JIRA:RHELPLAN-10442)

### sssdctl druckt einen HBAC-Regelbericht für eine IdM-Domain aus

Mit diesem Update kann das **sssdctl** Dienstprogramm des System Security Services Daemon (SSSD) einen Zugriffskontrollbericht für eine Identity Management (IdM)-Domäne drucken. Diese Funktion erfüllt die Anforderungen bestimmter Umgebungen, aus rechtlichen Gründen eine Liste von Benutzern und Gruppen zu sehen, die auf einen bestimmten Client-Computer zugreifen können. Wenn es auf einem IdM-Client ausgeführt **sssdctl access-report domain\_name** wird, wird die geparte Teilmenge der HBAC-Regeln (Host-based Access Control) in der IdM-Domäne gedruckt, die für den Client-Computer gelten.

Beachten Sie, dass keine anderen Anbieter als IdM diese Funktion unterstützen.

(JIRA:RHELPLAN-10443)

### Identitätsmanagement-Pakete sind als Modul erhältlich

In RHEL 8 werden die für die Installation eines Identity Management (IdM)-Servers und -Clients erforderlichen Pakete als Modul ausgeliefert. Der **client** Stream ist der Standard-Stream des **idm** Moduls und Sie können die für die Installation des Clients erforderlichen Pakete herunterladen, ohne den Stream zu aktivieren.

Der IdM-Server-Modul-Stream wird als **DL1** Stream bezeichnet. Der Stream enthält mehrere Profile, die

verschiedenen Typen von IdM-Servern entsprechen: Server, DNS, Adtrust, Client und Standard. Um die Pakete in einem bestimmten Profil des **DL1**Streams herunterzuladen: . Aktivieren Sie den Stream. Wechseln Sie zu den RPMs, die durch den Stream geliefert werden. Führen Sie den **yum module install idm:DL1/profile\_name**Befehl aus.

(JIRA:RHELPLAN-10438)

### Sitzungsaufzeichnungslösung für RHEL 8 hinzugefügt

Red Hat Enterprise Linux 8 (RHEL 8) wurde um eine Sitzungsaufzeichnungslösung erweitert. Ein neues **tlog**Paket und der zugehörige Webkonsolen-Sitzungsplayer ermöglichen die Aufzeichnung und Wiedergabe der Benutzer-Terminalsitzungen. Die Aufzeichnung kann pro Benutzer oder Benutzergruppe über den Service System Security Services Daemon (SSSD) konfiguriert werden. Alle Ein- und Ausgänge der Terminals werden erfasst und in einem Systemjournal in einem textbasierten Format gespeichert. Die Eingabe ist aus Sicherheitsgründen standardmäßig inaktiv, um Rohpasswörter und andere sensible Informationen nicht abzufangen.

Die Lösung kann für die Auditierung von Benutzersitzungen auf sicherheitsrelevanten Systemen eingesetzt werden. Im Falle eines Sicherheitsverstoßes können die aufgezeichneten Sitzungen im Rahmen einer forensischen Analyse überprüft werden. Die Systemadministratoren können nun die Sitzungsaufzeichnung lokal konfigurieren und das Ergebnis über die RHEL 8 Webkonsolenschnittstelle oder über die Kommandozeilenschnittstelle mit dem **tlog-play**Dienstprogramm anzeigen.

(JIRA:RHELPLAN-1473)

### authselect vereinfacht die Konfiguration der Benutzerauthentifizierung

Dieses Update stellt das **authselect**Dienstprogramm vor, das die Konfiguration der Benutzerauthentifizierung auf RHEL 8-Hosts vereinfacht und das **authconfig**Dienstprogramm ersetzt. **authselect**Es enthält einen sichereren Ansatz für das PAM-Stapelmanagement, der die Änderungen der PAM-Konfiguration für Systemadministratoren einfacher macht. **authselect**kann zur Konfiguration von Authentifizierungsmethoden wie Passwörter, Zertifikate, Smartcards und Fingerabdrücke verwendet werden. Beachten Sie, dass **authselect**keine Dienste konfiguriert werden, die für die Verbindung von Remote-Domänen erforderlich sind. Diese Aufgabe wird von spezialisierten Werkzeugen übernommen, wie z.B. **realmd**oder **ipa-client-install**.

(JIRA:RHELPLAN-10445)

### SSSD ermöglicht es Ihnen nun, eines der mehreren Smartcard-Authentifizierungsgeräte auszuwählen

Mit diesem Update können Sie den PKCS#11 URI für die Auswahl von Smartcard-Authentifizierungsgeräten konfigurieren.

Standardmäßig versucht SSSD, ein Gerät für die Smartcard-Authentifizierung automatisch zu erkennen. Wenn mehrere Geräte verbunden sind, wählt SSSD das erste gefundene Gerät aus und Sie können das entsprechende Gerät nicht auswählen. Es kann zu Ausfällen führen.

Daher können Sie nun eine neue **p11\_uri**Option für den **[pam]**Abschnitt von **sssd.conf**. Mit dieser Option können Sie festlegen, welches Gerät für die Smartcard-Authentifizierung verwendet wird.

Um beispielsweise den Leser mit der vom OpenSC PKCS#11-Modul erkannten Slot-ID'2' auszuwählen, fügen Sie Folgendes hinzu

**p11\_uri = library-description=OpenSC%20smartcard%20framework;slot-id=2**

in den **[pam]**Abschnitt von **sssd.conf**.

Bitte beachten **man sssd-conf**Sie die Details.

(BZ#1620123)

## 4.10. COMPILER UND ENTWICKLUNGSWERKZEUGE (MASCHINENÜBERSETZUNG)

### Boost aktualisiert auf Version 1.66

Die **Boost** C++ Bibliothek wurde auf die Upstream-Version 1.66 aktualisiert. Die in Red Hat Enterprise Linux 7 enthaltene Version von **Boost** ist 1.53. Weitere Informationen finden Sie in den vorgelagerten Changelogs: <https://www.boost.org/users/history/>

Dieses Update führt die folgenden Änderungen ein, die die Kompatibilität mit früheren Versionen beeinträchtigen:

- Die **bs\_set\_hook()**Funktion, die **splay\_set\_hook()**Funktion aus Splay-Containern und der **bool splay = true**zusätzliche Parameter in der **splaytree\_algorithms()**Funktion in der **Intrusive Library** wurden entfernt.
- Kommentare oder Zeichenkettenverknüpfungen in JSON-Dateien werden vom Parser in der **Eigenschaftsbaum-Bibliothek** nicht mehr unterstützt.
- Einige Distributionen und spezielle Funktionen aus der **Math-Bibliothek** wurden so korrigiert, dass sie sich wie dokumentiert verhalten und einen **overflow\_error**statt den maximalen endlichen Wert zurückgeben.
- Einige Header aus der **Math-Bibliothek** wurden in das Verzeichnis **libs/math/include\_private**verschoben.
- Das Verhalten der **basic\_regex<>::mark\_count()**und **basic\_regex<>::subexpression(n)**der Funktionen aus der **Regex-Bibliothek** wurde entsprechend ihrer Dokumentation geändert.
- Die Verwendung von variablen Vorlagen in der **Variantenbibliothek** kann Metaprogrammierungsfunktionen beeinträchtigen.
- Die **boost::python::numeric**API wurde entfernt. Benutzer können **boost::python::numpy**stattdessen verwenden.
- Arithmetische Operationen an Zeigern auf Nicht-Objekttypen werden in der **Atom**bibliothek nicht mehr angeboten.

(BZ#1494495)

### Unicode 11.0.0.0 Unterstützung

Die Red Hat Enterprise Linux Core C Library, **glibc**, wurde aktualisiert, um den Unicode-Standard Version 11.0.0.0 zu unterstützen. Infolgedessen liefern alle Breitzichen- und Multibyte-Zeichen-APIs, einschließlich der Transliteration und Konvertierung zwischen Zeichensätzen, genaue und korrekte Informationen, die dieser Norm entsprechen.

(BZ#1512004)

### Das boostPaket ist nun unabhängig von Python

Mit diesem Update installiert die Installation des **boost**Pakets die **Boost.Python**Bibliothek nicht mehr als Abhängigkeit. Um dies zu nutzen **Boost.Python**, müssen Sie die **boost-python3**oder **boost-python3-devel**die Pakete explizit installieren.

(BZ#1616244)

### Ein neues **compat-libfortran-48**Paket verfügbar

Für die Kompatibilität mit Red Hat Enterprise Linux 6 und 7 Anwendungen, die die Fortran-Bibliothek verwenden, ist nun ein neues **compat-libfortran-48**Kompatibilitätspaket verfügbar, das die **libfortran.so.3**Bibliothek bereitstellt.

(BZ#1607227)

### Retpoline-Unterstützung in GCC

Dieses Update bietet Unterstützung für Retpoline in GCC. Ein Repolin ist ein Softwarekonstrukt, das vom Kernel verwendet wird, um den Overhead für die Reduzierung von Spectre Variant 2-Angriffen zu reduzieren, wie in CVE-2017-5715 beschrieben.

(BZ#1535774)

### Verbesserte Unterstützung der 64-Bit-ARM-Architektur in Toolchain-Komponenten

Toolchain-Komponenten **GCC**und**binutils**, bieten nun erweiterte Unterstützung für die 64-Bit-ARM-Architektur. Zum Beispiel:

- **GCC** und unterstützt **binutils**nun die Scalable Vector Extension (SVE).
- Die Unterstützung für den **FP16**Datentyp, die von ARM v8.2 bereitgestellt wird, wurde zu **GCC**. Der **FP16**Datentyp verbessert die Leistung bestimmter Algorithmen.
- Tools unterstützen ab **binutils**sofort die ARM v8.3 Architekturdefinition, einschließlich Pointer Authentication. Die Pointer-Authentifizierungsfunktion verhindert, dass bösartiger Code die normale Ausführung eines Programms oder des Kernels beeinträchtigt, indem er eigene Funktionszeiger erstellt. Daher werden beim Verzweigen an verschiedene Stellen im Code nur vertrauenswürdige Adressen verwendet, was die Sicherheit erhöht.

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

### Optimierungen **glibc**für IBM POWER Systeme

Dieses Update bietet eine neue Version **glibc**, die sowohl für IBM POWER 8 als auch für IBM POWER 9 Architekturen optimiert ist. Dadurch wechseln IBM POWER 8 und IBM POWER 9 Systeme nun zur Laufzeit automatisch auf die entsprechende, optimierte **glibc**Variante.

(BZ#1376834)

### GNU C Library aktualisiert auf Version 2.28

Red Hat Enterprise Linux 8 enthält die Version 2.28 der GNU C Library (glibc). Zu den bemerkenswerten Verbesserungen gehören:

- Sicherheitshärtende Eigenschaften:
  - Sichere Binärdateien, die mit dem **AT\_SECURE**Flag markiert sind, ignorieren die **LD\_LIBRARY\_PATH**Umgebungsvariable.

- Backtraces werden nicht mehr gedruckt, wenn es zu Fehlern bei der Stapelprüfung kommt, um das Herunterfahren zu beschleunigen und mehr Code in einer kompromittierten Umgebung zu vermeiden.
- Leistungssteigerung:
  - Die Performance der **malloc()**Funktion wurde mit einem lokalen Thread-Cache verbessert.
  - Hinzufügen der **GLIBC\_TUNABLES**Umgebungsvariablen zum Ändern der Leistungsmerkmale der Bibliothek.
  - Die Implementierung von Thread-Semaphoren wurde verbessert und um neue skalierbare **pthread\_rwlock\_xxx()**Funktionen erweitert.
  - Die Leistung der mathematischen Bibliothek wurde verbessert.
- Die Unterstützung für Unicode 11.0.0.0 wurde hinzugefügt.
- Die Unterstützung für 128-Bit Fließkommazahlen gemäß den Normen ISO/IEC/IEEE 60559:2011, IEEE 754-2008 und ISO/IEC TS 18661-3:2015 wurde verbessert.
- Verbesserungen des Domain Name Service (DNS) Stub-Resolvers in Bezug auf die **/etc/resolv.conf**Konfigurationsdatei:
  - Die Konfiguration wird automatisch neu geladen, wenn die Datei geändert wird.
  - Es wurde die Unterstützung für eine beliebige Anzahl von Suchdomains hinzugefügt.
  - Die richtige Zufallsauswahl für die **rotate**Option wurde hinzugefügt.
- Neue Funktionen für die Entwicklung wurden hinzugefügt, darunter:
  - Linux-Wrapper-Funktionen für die **preadv2**und **pwritev2**Kernel-Aufrufe
  - Neue Funktionen einschließlich **reallocarray()**und **explicit\_bzero()**
  - Neue Kennzeichen für die **posix\_spawnattr\_setflags()**Funktion wie z.B **POSIX\_SPAWN\_SETSID**

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

### CMake in RHEL verfügbar

Das CMake Build-System Version 3.11 ist in Red Hat Enterprise Linux 8 als **cmake**Paket verfügbar.

(BZ#1590139, BZ#1502802)

### make version 4.2.1.1

Red Hat Enterprise Linux 8 wird mit dem **make**Build-Tool Version 4.2.1 ausgeliefert. Zu den bemerkenswerten Änderungen gehören:

- Wenn ein Rezept fehlschlägt, werden der Name des Makefiles und die Zeilennummer des Rezepts angezeigt.
- Die **--trace**Option wurde hinzugefügt, um die Verfolgung von Zielen zu ermöglichen. Wenn diese Option verwendet wird, wird jedes Rezept vor dem Aufruf gedruckt, auch wenn es unterdrückt würde, zusammen mit dem Dateinamen und der Zeilennummer, in der sich dieses Rezept befindet, sowie mit den Voraussetzungen, unter denen es aufgerufen wird.

- Das Mischen von expliziten und impliziten Regeln führt **make** nicht mehr zum Abbruch der Ausführung. Stattdessen wird eine Warnung ausgegeben. Beachten Sie, dass diese Syntax veraltet ist und in Zukunft möglicherweise vollständig entfernt werden kann.
- Die **\$(file ...)**Funktion zum Schreiben von Text in eine Datei wurde hinzugefügt. Beim Aufruf ohne Textargument wird die Datei nur geöffnet und sofort geschlossen.
- Eine neue Option **--output-syncoder-O**, bewirkt, dass eine Ausgabe von mehreren Aufträgen pro Auftrag gruppiert wird und ermöglicht ein einfacheres Debuggen von parallelen Builds.
- Die **--debug**Option akzeptiert nun auch das (none **n**) Flag, um alle aktuell aktivierten Debugging-Einstellungen zu deaktivieren.
- Der **!=**Shell Assignment Operator wurde als Alternative zur **\$(shell ...)**Funktion hinzugefügt, um die Kompatibilität mit BSD Makefiles zu erhöhen. Weitere Details und Unterschiede zwischen dem Bediener und der Funktion finden Sie im GNU make Handbuch.

Beachten Sie, dass Variablen mit einem Namen, der mit einem Ausrufezeichen endet und unmittelbar gefolgt von einer Zuweisung, wie z.B. **variable!=value**, interpretiert werden, nun als die neue Syntax. Um das vorherige Verhalten wiederherzustellen, fügen Sie ein Leerzeichen nach dem Ausrufezeichen hinzu, z.B. **variable! =value**.

+

- Der im POSIX-Standard definierte **::=**Zuweisungsoperator wurde hinzugefügt.
- Wenn die **.POSIX**Variable angegeben wird, **make** beachtet man die POSIX-Standardanforderungen für den Umgang mit Backslash und neuer Zeile. In diesem Modus wird jedes Leerzeichen vor dem Backslash beibehalten, und jeder Backslash gefolgt von einer neuen Zeile und Leerzeichen wird in ein einzelnes Leerzeichen umgewandelt.
- Das Verhalten der **MAKEFLAGS** und **MFLAGS** Variablen ist nun genauer definiert.
- Eine neue Variable, **GNUMAKEFLAGS**, wird für **make**Flags analysiert, die identisch sind mit **MAKEFLAGS**. Dadurch können GNU-spezifische **make**Flags außerhalb **MAKEFLAGS** gespeichert werden und die Portierbarkeit von Makefiles wird erhöht.
- Eine neue Variable, **MAKE\_HOST** die die Host-Architektur enthält, wurde hinzugefügt.
- Die neuen Variablen **MAKE\_TERMOUT** und **MAKE\_TERMERR**, geben an, ob es sich um das Schreiben von Standardausgabe und Fehler in eine Klemme **make** handelt.
- Das Setzen der **-rund -R**Optionen in der **MAKEFLAGS** Variablen innerhalb eines Makefiles funktioniert nun korrekt und entfernt alle eingebauten Regeln bzw. Variablen.
- Die **.RECIPEPREFIX** Einstellung wird nun für jedes Rezept gespeichert. Zusätzlich verwenden Variablen, die in diesem Rezept erweitert wurden, auch diese Einstellung des Rezeptpräfixes.
- Die **.RECIPEPREFIX** Einstellung und alle zielspezifischen Variablen werden in der Ausgabe der **-p** Option wie in einem Makefile und nicht wie Kommentare angezeigt.

(BZ#1641015)

### Go-Programme, die mit Go Toolset erstellt wurden, sind FIPS-konform

Die in **Go Toolset** verfügbare kryptografische Bibliothek wurde so geändert, dass sie die **OpenSSL-Bibliothek** Version 1.1.0 verwendet, wenn das Hostsystem im FIPS-Modus konfiguriert ist. Infolgedessen sind Programme, die mit dieser Version von **Go Toolset** erstellt wurden, FIPS-konform.

Um Go-Programme nur die nicht zertifizierten kryptographischen Standardroutinen verwenden zu lassen, verwenden Sie die **-tags no\_openssl** Option des **go** Compilers zur Buildzeit.

(BZ#1512570)

### SystemTap Version 4.0

Red Hat Enterprise Linux 8 wird mit dem **SystemTap** Instrumentation Tool Version 4.0 ausgeliefert. Zu den bemerkenswerten Verbesserungen gehören:

- Das erweiterte Berkeley Packet Filter (eBPF) Backend wurde verbessert, insbesondere Zeichenketten und Funktionen. Um dieses Backend zu nutzen, starten Sie **SystemTap** mit der **--runtime=bpf** Option.
- Ein neuer Exportnetzwerkdienst für das Prometheus-Überwachungssystem wurde hinzugefügt.
- Die Implementierung der Systemaufrufsondierung wurde verbessert, um bei Bedarf die Kernel-Tracepoints zu verwenden.

(BZ#1641032)

### Verbesserungen in der binutils Version 2.30

Red Hat Enterprise Linux 8 enthält die Version 2.30 des **binutils** Pakets. Zu den bemerkenswerten Verbesserungen gehören:

- Die Unterstützung für neue s390x Architekturereweiterungen wurde verbessert.

Assembler:

- Unterstützung für das Dateiformat WebAssembly und die Konvertierung von WebAssembly in das Dateiformat wasm32 ELF wurde hinzugefügt.
- Unterstützung für die ARMv8-R-Architektur und die Prozessoren Cortex-R52, Cortex-M23 und Cortex-M33 wurde hinzugefügt.
- Die Unterstützung für die RISC-V-Architektur wurde hinzugefügt.

Linkers:

- Der Linker legt nun standardmäßig Code und schreibgeschützte Daten in separate Segmente. Dadurch sind die erstellten ausführbaren Dateien größer und sicherer, da der Dynamic Loader die Ausführung jeder Speicherseite mit schreibgeschützten Daten deaktivieren kann.
- Unterstützung für GNU Property Notes, die Hinweise für den Dynamic Loader über die Binärdatei enthalten, wurde hinzugefügt.
- Zuvor erzeugte der Linker ungültigen ausführbaren Code für die Intel Indirect Branch Tracking (IBT) Technologie. Infolgedessen konnten die erzeugten ausführbaren Dateien nicht gestartet werden. Dieser Fehler wurde behoben.
- Zuvor hat der **gold** Linker Eigenschaftsnotizen unsachgemäß zusammengeführt. Infolgedessen konnten falsche Hardwaremerkmale im generierten Code aktiviert werden und der Code konnte unerwartet abbrechen. Dieser Fehler wurde behoben.
- Bisher hat der **gold** Linker Notenabschnitte mit Padding-Bytes am Ende erstellt, um eine Ausrichtung nach der Architektur zu erreichen. Da der Dynamic Loader die Polsterung nicht erwartet hat, konnte er das geladene Programm unerwartet beenden. Dieser Fehler wurde

behoben.

Andere Werkzeuge:

- Die **readelf** und **objdump** Tools haben nun Optionen, um Links in separate Debug-Informationsdateien zu folgen und auch Informationen darin anzuzeigen.
- Die neue **--inlines** Option erweitert die bestehende **--line-numbers** Option des **objdump** Werkzeugs, um Schachtelinformationen für Inline-Funktionen anzuzeigen.
- Das **nm** Tool hat eine neue Option **--with-version-strings** erhalten, um Versionsinformationen eines Symbols nach seinem Namen anzuzeigen, falls vorhanden.

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

### Performance Co-Pilot Version 4.1.3

Red Hat Enterprise Linux 8 wird mit **Performance Co-Pilot** (pcp) Version 4.1.3 ausgeliefert. Zu den bemerkenswerten Verbesserungen gehören:

- Das **pcp-dstat** Tool enthält nun auch historische Analysen und die Ausgabe im CSV-Format (Comma-separated Values).
- Die Protokolldienstprogramme können metrische Bezeichnungen und Hilfstexte verwenden.
- Das **pmdaperfevent** Tool meldet nun die korrekten CPU-Nummern auf den unteren Ebenen des Simultaneous Multi Threading (SMT).
- Das **pmdapostgresql** Tool unterstützt nun **Postgres** Serie 10.x.
- Das **pmdaredis** Tool unterstützt nun **Redis** Serie 5.x.
- Das **pmdabcc** Tool wurde um dynamische Prozessfilterung und prozessbezogene Systemaufrufe, Ucalls und ustat erweitert.
- Das **pmdammv** Tool exportiert nun metrische Etiketten und die Formatversion wird auf 3 erhöht.
- Das **pmdagfs2** Tool unterstützt zusätzliche Kennzahlen für Glock und Glockhalter.
- Es wurden mehrere Korrekturen an der SELinux-Richtlinie vorgenommen.

(BZ#1641034)

### Speicherschutzschlüssel

Dieses Update aktiviert Hardwarefunktionen, die Änderungen an den Schutzflags für einzelne Threads ermöglichen. Die neuen **glibc** Systemaufruf-Verschaltungen für die **pkey\_mprotect()** Funktionen **pkey\_alloc()**, **pkey\_free()**, „ und, wurden hinzugefügt. Darüber hinaus wurden die **pkey\_set()** und **pkey\_get()** Funktionen hinzugefügt, um den Zugriff auf die Per-Thread-Schutzflags zu ermöglichen.

(BZ#1304448)

### elfutils aktualisiert auf Version 0.174

In Red Hat Enterprise Linux 8 ist das Paket **elfutils** in der Version 0.174 verfügbar. Zu den bemerkenswerten Änderungen gehören:



- Bisher konnte das **eu-readelf** Tool eine Variable mit einem negativen Wert anzeigen, als ob sie einen großen vorzeichenlosen Wert hätte, oder einen großen vorzeichenlosen Wert als negativen Wert. Dies wurde korrigiert und sucht **eu-readelf** nun nach der Größe und Signatur von konstanten Werttypen, um sie korrekt anzuzeigen.
- Die **libdw**-Bibliothek wurde um eine neue Funktion **dwarf\_next\_lines()** zum Lesen **.debug\_line** von Daten ohne CU erweitert. Diese Funktion kann als Alternative zu den Funktionen **dwarf\_getsrclines()** und **dwarf\_getsrcfiles()** verwendet werden.
- Bisher konnten Dateien mit mehr als 65280 Abschnitten Fehler in den Bibliotheken **libelf** und **libdw** und allen Werkzeugen, die sie verwenden, verursachen. Dieser Fehler wurde behoben. Dadurch werden erweiterte **shnum** und **shstrndx** Werte in ELF-Dateiheadern korrekt behandelt.

(BZ#1641007)

### Valgrind auf Version 3.14 aktualisiert

Red Hat Enterprise Linux 8 wird mit dem ausführbaren Codeanalysetool Valgrind Version 3.14 ausgeliefert. Zu den bemerkenswerten Änderungen gehören:

- Eine neue **--keep-debuginfo** Option wurde hinzugefügt, um die Speicherung von Debug-Informationen für entladenen Code zu ermöglichen. Infolgedessen können gespeicherte Stapelbahnen Datei- und Zeileninformationen für Code enthalten, der nicht mehr im Speicher vorhanden ist.
- Unterdrückungen basierend auf dem Namen der Quelldatei und der Zeilennummer wurden hinzugefügt.
- Das **Helgrind** Tool wurde um eine Option **--delta-stacktrace** erweitert, mit der die Berechnung der vollständigen Stack-Traces der Historie spezifiziert werden kann. Insbesondere **--history-level=full** kann die Verwendung dieser Option in Kombination mit einer Leistungssteigerung **Helgrind** von bis zu 25% führen.
- Die Falschpositivrate im **Memcheck** Tool für optimierten Code auf den Intel und AMD 64-Bit Architekturen und der ARM 64-Bit Architektur wurde reduziert. Beachten Sie, dass Sie mit der **--expensive-definedness-checks** Funktion die Behandlung von Definitionskontrollen steuern und die Rate auf Kosten der Performance verbessern können.
- Valgrind kann nun weitere Anweisungen der Little-Endian-Variante von IBM Power Systems erkennen.
- Valgrind kann nun Ganzzahl- und Zeichenkettenvektorbefehle des IBM Z-Architektur z13-Prozessors teilweise verarbeiten.

Weitere Informationen zu den neuen Optionen und ihren bekannten Einschränkungen finden Sie auf der **valgrind(1)** Handbuchseite.

(BZ#1641029, BZ#1501419)

### GDB Version 8.2

Red Hat Enterprise Linux 8 wird mit dem GDB-Debugger Version 8.2 ausgeliefert. Zu den wichtigsten Änderungen gehören:

- Das IPv6-Protokoll wird für das Remote-Debugging mit GDB und **gdbserver** unterstützt.
- Das Debugging ohne Debug-Informationen wurde verbessert.

- Die Symbolvervollständigung in der GDB-Benutzeroberfläche wurde verbessert, um bessere Vorschläge zu machen, indem syntaktischere Konstruktionen wie ABI-Tags oder Namensräume verwendet werden.
- Befehle können nun im Hintergrund ausgeführt werden.
- Debugging-Programme, die in der Programmiersprache Rust erstellt wurden, sind nun möglich.
- Das Debuggen von C- und C++-Sprachen wurde verbessert, mit Parserunterstützung für die Operatoren **\_Alignof** und **alignof** Operatoren, C++ rvalue Referenzen und C99 automatische Arrays mit variabler Länge.
- GDB-Erweiterungsskripte können nun die Skriptsprache Guile verwenden.
- Die Python-Skriptsprache-Schnittstelle für Erweiterungen wurde um neue API-Funktionen, Frame-Dekoratoren, Filter und Abwickler erweitert. Zusätzlich werden Skripte im **.debug\_gdb\_scripts** Abschnitt der GDB-Konfiguration automatisch geladen.
- GDB verwendet nun Python Version 3, um seine Skripte auszuführen, darunter hübsche Drucker, Rahmendekorateure, Filter und Abwickler.
- Die ARM- und 64-Bit-ARM-Architekturen wurden mit der Aufzeichnung und Wiedergabe von Prozessausführungen verbessert, einschließlich Thumb 32-Bit- und Systemanweisungen.
- Unterstützung für Intel MPX-Register und gebundene Verletzungen, das PKU-Register und den Intel Processor Trace wurde hinzugefügt.
- Die Aufzeichnungs- und Wiedergabefunktionalität wurde um die **rdrand** und **rdseed** Anweisungen auf Intel-basierten Systemen erweitert.
- Die Funktionalität von GDB auf der IBM Z-Architektur wurde um die Unterstützung von Tracepoints und Fast Tracepoints, Vektorregistern und ABI sowie dem **Catch** Systemaufruf erweitert. Zusätzlich unterstützt GDB nun neuere Anweisungen der Architektur.
- GDB kann nun die SystemTap static user space probes (SDT) auf der 64-Bit-ARM-Architektur verwenden.

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

### Die Lokalisierung für RHEL ist in mehreren Paketen verteilt

In RHEL 8 werden Sprachumgebungen und Übersetzungen nicht mehr durch das einzelne **glibc-common** Paket bereitgestellt. Stattdessen ist jedes Gebietsschema und jede Sprache in einem **glibc-langpack-CODE** Paket verfügbar. Außerdem werden standardmäßig nicht alle Sprachumgebungen installiert, sondern nur die im Installationsprogramm ausgewählten. Benutzer müssen alle weiteren Gebietsschema-Pakete, die sie benötigen, separat installieren.

Weitere Informationen zur Verwendung von Langpacks finden Sie unter [Installing and using langpacks](#).

(BZ#1512009)

### strace aktualisiert auf Version 4.24

Red Hat Enterprise Linux 8 wird mit der **strace** Tool-Version 4.24 ausgeliefert. Zu den bemerkenswerten Änderungen gehören:

- Mit der **-e inject=** Option wurden Manipulationsfunktionen für Systemaufrufe hinzugefügt. Dazu gehört die Einspeisung von Fehlern, Rückgabewerten, Verzögerungen und Signalen.

- Die Qualifikation des Systemaufrufs wurde verbessert:
  - Es wurde eine **-e trace=/regex**Option hinzugefügt, um Systemaufrufe mit regulären Ausdrücken zu filtern.
  - Wenn Sie einer Systemaufruf-Qualifikation in der **-e trace=**Option ein Fragezeichen voranstellen, können Sie **strace**fortfahren, auch wenn die Qualifikation keinem Systemaufruf entspricht.
  - Die Qualifikationen für Systemaufrufe wurden in der **-e trace**Option um eine Persönlichkeitsbezeichnung erweitert.
- Die Dekodierung des **kvm vcpu**Ausgangsgrundes wurde hinzugefügt. Verwenden Sie dazu die **-e kvm=vcpu**Option.
- Die **libdw**Bibliothek wird nun für die Stapelabwicklung verwendet, wenn die **-k**Option verwendet wird. Zusätzlich ist eine Symbolentflechtung möglich, wenn die **libiberty**Bibliothek auf dem System installiert ist.
- Bisher hat die **-r**Option dazu geführt **strace**, dass sie die **-t**Option ignoriert hat. Dies wurde behoben und die beiden Optionen sind nun unabhängig voneinander.
- Die Option `[option]`-A` wurde hinzugefügt, um Ausgabedateien im Append-Modus zu öffnen.
- Die **-X**Option zur Konfiguration **xlat**der Ausgabeformatierung wurde hinzugefügt.
- Die Dekodierung von Socket-Adressen mit der **-yy**Option wurde verbessert. Zusätzlich wurde der Druck von Block- und Zeichengerätenummern im **-yy**Modus hinzugefügt.

Zusätzlich wurde die Dekodierung der folgenden Elemente hinzugefügt, verbessert oder aktualisiert:

- **netlink** protokolle, Meldungen und Attribute
- **arch\_prctl,bpf,getsockopt,,io\_pgetevent,kern\_features,keyctl,prctl,pkey\_alloc,pkey\_free,pkey\_mprotect,,ptrace,rseq,setsockopt,,socket,,,, statx**und andere Systemaufrufe
- Viele Befehle für den **ioctl**Systemaufruf
- Konstanten verschiedener Typen
- Pfadverfolgung für **execveatsymlinkat**Systemaufrufe, **inotify\_add\_watch,inotify\_init,,select,symlink**, Systemaufrufe und **mmap**Systemaufrufe mit indirekten Argumenten
- Systemaufrufe spezifisch für die ARM-Architekturen **\_\_ARM\_NR\_\***
- Listen der Signalcodes

(BZ#1641014)

## 4.11. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)

### XFS unterstützt nun gemeinsame Datenerweiterungen beim Kopieren beim Schreiben

Das XFS-Dateisystem unterstützt die Funktionalität des gemeinsamen Kopierens beim Schreiben von Datenerweiterungen. Diese Funktion ermöglicht es zwei oder mehr Dateien, einen gemeinsamen Satz von Datenblöcken gemeinsam zu nutzen. Wenn sich eine der Dateien ändert, die gemeinsame Blöcke

teilen, unterbricht XFS die Verknüpfung zu gemeinsamen Blöcken und erstellt eine neue Datei. Dies ist vergleichbar mit der Copy-on-Write (COW)-Funktionalität in anderen Dateisystemen.

Gemeinsame Copy-on-Write-Datenerweiterungen sind:

### Schnell

Das Erstellen von gemeinsamen Kopien verwendet keine Festplatten-I/Os.

### Platzsparend

Gemeinsame Blöcke verbrauchen keinen zusätzlichen Festplattenspeicher.

### Transparent

Dateien, die gemeinsame Blöcke teilen, verhalten sich wie normale Dateien.

Userspace-Utilities können gemeinsame Copy-on-Write Datenerweiterungen verwenden für:

- Effizientes Klonen von Dateien, z.B. mit dem **cp --reflink**Befehl
- Snapshots pro Datei

Diese Funktionalität wird auch von Kernel-Subsystemen wie Overlayfs und NFS für einen effizienteren Betrieb genutzt.

Gemeinsame Datenerweiterungen beim Kopieren beim Schreiben sind nun standardmäßig aktiviert, wenn ein XFS-Dateisystem erstellt wird, beginnend mit der **xfsprogs**Paketversion **4.17.0-2.el8**.

Beachten Sie, dass Direct Access (DAX)-Geräte derzeit XFS mit gemeinsamen Copy-on-Write Datenerweiterungen nicht unterstützen. Um ein XFS-Dateisystem ohne diese Funktion zu erstellen, verwenden Sie den folgenden Befehl:

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

Red Hat Enterprise Linux 7 kann XFS-Dateisysteme mit gemeinsamen Copy-on-Write-Datenerweiterungen nur im schreibgeschützten Modus mounten.

(BZ#1494028)

## Die maximale Größe des XFS-Dateisystems beträgt 1024 TiB

Die maximal unterstützte Größe eines XFS-Dateisystems wurde von 500 TiB auf 1024 TiB erhöht.

Dateisysteme mit mehr als 500 TiB erfordern dies:

- die Metadaten-CRC-Funktion und die freie Inode-Baum-Funktion sind beide im Dateisystemformat aktiviert und
- die Zuordnungsgruppengröße beträgt mindestens 512 GiB.

In RHEL 8 erstellt das **mkfs.xfs**Dienstprogramm Dateisysteme, die diese Anforderungen standardmäßig erfüllen.

Ein kleineres Dateisystem, das diese Anforderungen nicht erfüllt, auf eine neue Größe von mehr als 500 TiB zu erweitern, wird nicht unterstützt.

(BZ#1563617)

## VDO unterstützt nun alle Architekturen

Virtual Data Optimizer (VDO) ist nun auf allen von RHEL 8 unterstützten Architekturen verfügbar.

Die Liste der unterstützten Architekturen finden Sie unter [Kapitel 2, Architekturen \(Maschinenübersetzung\)](#).

(BZ#1534087)

### Der BOOM Bootmanager vereinfacht die Erstellung von Boot-Einträgen

BOOM ist ein Bootmanager für Linux-Systeme, die Bootloader verwenden, die die BootLoader-Spezifikation für die Konfiguration des Boot-Eintrags unterstützen. Es ermöglicht eine flexible Boot-Konfiguration und vereinfacht die Erstellung neuer oder geänderter Boot-Einträge: z.B. um Snapshot-Images des mit LVM erstellten Systems zu starten.

BOOM ändert die bestehende Bootloader-Konfiguration nicht und fügt nur zusätzliche Einträge ein. Die bestehende Konfiguration wird beibehalten, und jede Distributionsintegration, wie Kernel-Installations- und Aktualisierungsskripte, funktioniert weiterhin wie bisher.

BOOM hat eine vereinfachte Befehlszeilenschnittstelle (CLI) und eine API, die die Erstellung von Boot-Einträgen erleichtert.

(BZ#1649582)

### LUKS2 ist nun das Standardformat für die Verschlüsselung von Volumes

In RHEL 8 ersetzt das Format LUKS Version 2 (LUKS2) das alte Format LUKS (LUKS1). Das **dm-crypt**Subsystem und das **cryptsetup**Tool verwenden nun LUKS2 als Standardformat für verschlüsselte Datenträger. LUKS2 bietet verschlüsselte Datenträger mit Metadatenredundanz und automatischer Wiederherstellung im Falle eines teilweisen Metadatenkorruptionsereignisses.

Aufgrund des internen flexiblen Layouts ist LUKS2 auch für zukünftige Funktionen geeignet. Es unterstützt die automatische Entriegelung durch das eingebaute generische Kernel-Keyring-Token, **libcryptsetup**das es Benutzern ermöglicht, LUKS2-Volumes mithilfe einer im Kernel-Keyring-Retentionsdienst gespeicherten Passphrase zu entsperren.

Weitere bemerkenswerte Verbesserungen sind:

- Die Einrichtung des geschützten Schlüssels unter Verwendung des verpackten Schlüsselverschlüsselungsschemas.
- Einfachere Integration mit Policy-Based Decryption (Clevis).
- Bis zu 32 Key-Slots - LUKS1 bietet nur 8 Key-Slots.

Weitere Informationen finden Sie auf den Seiten **cryptsetup(8)**und **cryptsetup-reencrypt(8)**in der Dokumentation.

(BZ#1564540)

### NVMe/FC wird von Broadcom Emulex Fibre Channel Adaptern vollständig unterstützt

Der Transporttyp NVMe over Fibre Channel (NVMe/FC) wird nun im Initiator-Modus vollständig unterstützt, wenn er mit Broadcom Emulex Fibre Channel 32Gbit Adaptern verwendet wird.

NVMe over Fibre Channel ist ein zusätzlicher Fabric-Transporttyp für das Nonvolatile Memory Express (NVMe) Protokoll, zusätzlich zum Remote Direct Memory Access (RDMA) Protokoll, das zuvor in Red Hat Enterprise Linux eingeführt wurde.

Um NVMe/FC im **lpfc** Treiber zu aktivieren, bearbeiten Sie die `/etc/modprobe.d/lpfc.conf` Datei und fügen Sie die folgende Option hinzu:

```
lpfc_enable_fc4_type=3
```

Andere Treiber **lpfc** bleiben in der Technologievorschau.

Zusätzliche Einschränkungen:

- Multipath wird bei NVMe/FC nicht unterstützt.
- NVMe-Clustering wird von NVMe/FC nicht unterstützt.
- Derzeit unterstützt Red Hat Enterprise Linux nicht die gleichzeitige Verwendung von NVMe/FC und SCSI/FC an einem Initiatorport.
- Das `kernel-alt` Paket unterstützt NVMe/FC nicht.
- **kdump** wird von NVMe/FC nicht unterstützt.
- Das Booten vom Storage Area Network (SAN) NVMe/FC wird nicht unterstützt.

(BZ#1649497)

### Neuer **overrides** Abschnitt der DM-Multipath-Konfigurationsdatei

Die `/etc/multipath.conf` Datei enthält nun einen **overrides** Abschnitt, in dem Sie einen Konfigurationswert für alle Ihre Geräte festlegen können. Diese Attribute werden von DM Multipath für alle Geräte verwendet, es sei denn, sie werden durch die Attribute überschrieben, die im **multipaths** Abschnitt der `/etc/multipath.conf` Datei für Pfade angegeben sind, die das Gerät enthalten. Diese Funktionalität ersetzt den **all\_devs** Parameter des **devices** Abschnitts der Konfigurationsdatei, der nicht mehr unterstützt wird.

(BZ#1643294)

### Die Installation und das Booten von NVDIMM-Geräten wird nun unterstützt

Vor diesem Update wurden nichtflüchtige Dual Inline Memory Module (NVDIMM)-Geräte in jedem Modus vom Installateur ignoriert.

Mit diesem Update bieten Kernelverbesserungen zur Unterstützung von NVDIMM-Geräten verbesserte Systemleistungsmerkmale und verbesserten Dateisystemzugriff für schreibintensive Anwendungen wie Datenbank- oder Analyse-Workloads sowie reduzierten CPU-Overhead.

Dieses Update bietet Unterstützung für:

- Verwendung von NVDIMM-Geräten für die Installation unter Verwendung des Befehls **nvdimm** Kickstart und der Benutzeroberfläche, wodurch es möglich ist, NVDIMM-Geräte im Sektormodus zu installieren und zu starten und NVDIMM-Geräte während der Installation in den Sektormodus umzukonfigurieren.
- Die Erweiterung der **Kickstart** Skripte für **Anaconda** um Befehle zur Handhabung von NVDIMM-Geräten.
- Die Fähigkeit von **grub2**, **efibootmgr** und **efivar** Systemkomponenten, mit NVDIMM-Geräten umzugehen und von ihnen zu booten.

(BZ#1499442)

## Die Erkennung von Randpfaden in DM-Multipath wurde verbessert

Der **multipathd**Dienst unterstützt nun eine verbesserte Erkennung von Randpfaden. Dies hilft Mehrwegegeräten, Pfade zu vermeiden, die wahrscheinlich immer wieder ausfallen, und verbessert die Leistung. Randpfade sind Pfade mit persistenten, aber intermittierenden I/O-Fehlern.

Die folgenden Optionen im Verhalten der **/etc/multipath.conf**Datei steuern Randpfade:

- **marginal\_path\_double\_failed\_time**,
- **marginal\_path\_err\_sample\_time**,
- **marginal\_path\_err\_rate\_threshold**und
- **marginal\_path\_err\_recheck\_gap\_time**.

DM Multipath deaktiviert einen Pfad und testet ihn mit wiederholten I/Os für die konfigurierte Abtastzeit `if`:

- die aufgelisteten **multipath.conf**Optionen sind eingestellt,
- ein Pfad in der konfigurierten Zeit zweimal ausfällt, und
- andere Pfade sind verfügbar.

Wenn der Pfad während dieses Tests mehr als die konfigurierte Fehlerrate aufweist, ignoriert DM Multipath ihn für die konfigurierte Gap-Zeit und testet ihn dann erneut, um zu sehen, ob er gut genug funktioniert, um wiederhergestellt zu werden.

Weitere Informationen finden Sie in der **multipath.conf**Man Page.

(BZ#1643550)

## Standardverhalten für mehrere Warteschlangen

Blockgeräte verwenden nun die Multiqueue-Planung in Red Hat Enterprise Linux 8. Dies ermöglicht eine gute Skalierung der Blocklayer-Performance mit schnellen Solid-State-Laufwerken (SSDs) und Multicore-Systemen.

Der SCSI Multiqueue (**scsi-mq**) Treiber ist nun standardmäßig aktiviert, und der Kernel bootet mit der **scsi\_mod.use\_blk\_mq=Y**Option. Diese Änderung steht im Einklang mit dem vorgelagerten Linux-Kernel.

Device Mapper Multipath (DM Multipath) erfordert, dass der **scsi-mq**Treiber aktiv ist.

(BZ#1647612)

## Stratis ist jetzt verfügbar

Stratis ist ein neuer lokaler Speichermanager. Es bietet verwaltete Dateisysteme auf Basis von Speicherpools mit zusätzlichen Funktionen für den Benutzer.

Mit Stratis können Sie Speicheraufgaben wie:

- Verwaltung von Snapshots und Thin Provisioning
- Automatische Vergrößerung der Dateisystemgrößen bei Bedarf
- Dateisysteme pflegen

Um den Stratis-Speicher zu verwalten, verwenden Sie das **stratis**Dienstprogramm, das mit dem **stratisd**Hintergrunddienst kommuniziert.

Weitere Informationen finden Sie in der Dokumentation Stratis: [Managing layered local storage with Stratis](#).

(JIRA:RHELPLAN-1212)

## 4.12. HOCHVERFÜGBARKEIT UND CLUSTER (MASCHINENÜBERSETZUNG)

### Neue **pcs**Befehle zum Auflisten verfügbarer Watchdog-Geräte und zum Testen von Watchdog-Geräten

Um SBD mit Herzschrittmacher zu konfigurieren, ist ein funktionierendes Watchdog-Gerät erforderlich. Diese Version unterstützt den **pcs stonith sbd watchdog list**Befehl zum Auflisten verfügbarer Watchdog-Geräte auf dem lokalen Knoten und den **pcs stonith sbd watchdog test**Befehl zum Testen eines Watchdog-Geräts. Informationen zum **sbd**Befehlszeilentool finden Sie in der Man Page **sbd**(8).

(BZ#1578891)

### Der **pcs**Befehl unterstützt nun das Filtern von Ressourcenausfällen durch eine Operation und deren Intervall

Der Herzschrittmacher verfolgt nun Ressourcenausfälle pro Ressourcenoperation über einem Ressourcennamen und einem Knoten. Der **pcs resource failcount show**Befehl ermöglicht nun das Filtern von Fehlern nach Ressource, Knoten, Operation und Intervall. Es bietet die Möglichkeit, Fehler anzuzeigen, die pro Ressource und Knoten aggregiert oder detailliert pro Ressource, Knoten, Operation und Intervall angezeigt werden. Zusätzlich erlaubt der **pcs resource failcount reset**Befehl nun das Filtern von Fehlern nach Ressource, Knoten, Operation und Intervall.

(BZ#1591308)

### Zeitstempel im **corosync**Protokoll aktiviert

Das **corosync**Protokoll enthielt bisher keine Zeitstempel, was es schwierig machte, es mit Protokollen anderer Knoten und Dämonen in Verbindung zu bringen. Ab diesem Release sind im **corosync**Protokoll Zeitstempel vorhanden.

(BZ#1615420)

### Neue Formate für **pcs cluster setup**, **pcs cluster node add**und **pcs cluster node remove**Befehle

In Red Hat Enterprise Linux 8 werden Corosync 3, **knet**, und Knotennamen **pcs**vollständig unterstützt. Knotennamen werden nun benötigt und ersetzen Knotenadressen in der Rolle des Knotenidentifikators. Knotenadressen sind nun optional.

- Im **pcs host auth**Befehl werden die Knotenadressen standardmäßig auf Knotennamen festgelegt
- In den **pcs cluster node add**Befehlen **pcs cluster setup**und werden die Knotenadressen standardmäßig auf die im **pcs host auth**Befehl angegebenen Knotenadressen festgelegt.

Mit diesen Änderungen haben sich die Formate für die Befehle zum Einrichten eines Clusters, Hinzufügen eines Knotens zu einem Cluster und Entfernen eines Knotens aus einem Cluster geändert. Informationen zu diesen neuen Befehlsformaten finden Sie in der Hilfeanzeige für die **pcs cluster node remove**Befehle**pcs cluster setup**, **pcs cluster node add**und.



(BZ#1158816)

## Herzschrittmacher 2.0.0.0 in RHEL 8

Die **pacemaker**Pakete wurden auf die Upstream-Version des Pacemaker 2.0.0 aktualisiert, die eine Reihe von Bugfixes und Verbesserungen gegenüber der Vorgängerversion bietet:

- Das Detailprotokoll des Herzschrittmachers ist nun **/var/log/pacemaker/pacemaker.log** standardmäßig vorhanden (nicht direkt in **/var/log** oder kombiniert mit dem **corosync** Protokoll unter **/var/log/cluster**).
- Die Daemonprozesse des Pacemaker wurden umbenannt, um das Lesen der Protokolle intuitiver zu gestalten. Zum Beispiel wurde **penginesie** umbenannt in **pacemaker-schedulerd**.
- Die Unterstützung für die Eigenschaften deprecated **default-resource-stickiness** und **is-managed-default** cluster wurde eingestellt. Die **resource-stickiness** und **is-managed** Eigenschaften sollten stattdessen in den Ressourcenvorgaben festgelegt werden. Bestehende Konfigurationen (wenn auch nicht neu erstellte) mit der veralteten Syntax werden automatisch aktualisiert, um die unterstützte Syntax zu verwenden.
- Eine vollständigere Liste der Änderungen finden Sie unter <https://access.redhat.com/articles/3681151>.

Es wird empfohlen, dass Benutzer, die einen bestehenden Cluster mit Red Hat Enterprise Linux 7 oder früher aktualisieren, auf einem beliebigen Clusterknoten vor und nach dem Upgrade von RHEL auf allen Clusterknoten laufen **pcs cluster cib-upgrade**.

(BZ#1543494)

## Master-Ressourcen, die in promotable Klon-Ressourcen umbenannt wurden

Red Hat Enterprise Linux (RHEL) 8 unterstützt Pacemaker 2.0, bei dem eine Master/Slave-Ressource nicht mehr ein separater Ressourcentyp, sondern eine Standard-Klonressource mit einem **promotable** Meta-Attribut ist, das auf **true**. Die folgenden Änderungen wurden zur Unterstützung dieses Updates vorgenommen:

- Es ist nicht mehr möglich, mit dem **pcs** Befehl Masterressourcen anzulegen. Stattdessen ist es möglich, **promotable** Klonressourcen zu erstellen. Verwandte Keywords und Befehle wurden von **master** auf **promotable** geändert.
- Alle vorhandenen Master-Ressourcen werden als promotable Klon-Ressourcen angezeigt.
- Bei der Verwaltung eines RHEL7-Clusters im Web-UI werden Master-Ressourcen weiterhin als Master bezeichnet, da RHEL7-Cluster keine promotablen Klone unterstützen.

(BZ#1542288)

## Neue Befehle zur Authentifizierung von Knoten in einem Cluster

Red Hat Enterprise Linux (RHEL) 8 enthält die folgenden Änderungen an den Befehlen zur Authentifizierung von Knoten in einem Cluster.

- Der neue Befehl für die Authentifizierung lautet **pcs host auth**. Mit diesem Befehl können Benutzer Hostnamen, Adressen und **pcsd** Ports angeben.
- Der **pcs cluster auth** Befehl authentifiziert nur die Knoten in einem lokalen Cluster und akzeptiert keine Knotenliste

- Es ist nun möglich, für jeden Knoten eine Adresse anzugeben **pcs/pcsd** wird dann mit jedem Knoten über die angegebene Adresse kommunizieren. Diese Adressen können sich von denen unterscheiden, die intern **corosync** verwendet werden.
- Der **pcs pcsd clear-auth** Befehl wurde durch die Befehle **pcs pcsd deauth** und **pcs host deauth** ersetzt. Mit den neuen Befehlen können Benutzer sowohl einen einzelnen Host als auch alle Hosts abmelden.
- Bisher war die Knotenauthentifizierung bidirektional, und durch die Ausführung des **pcs cluster auth** Befehls wurden alle angegebenen Knoten gegeneinander authentifiziert. Der **pcs host auth** Befehl bewirkt jedoch, dass nur der lokale Host gegenüber den angegebenen Knoten authentifiziert wird. Dies ermöglicht eine bessere Kontrolle darüber, welcher Knoten gegenüber welchen anderen Knoten authentifiziert wird, wenn dieser Befehl ausgeführt wird. Beim Cluster-Setup selbst und auch beim Hinzufügen eines Knotens werden Token auf dem Cluster **pcs** automatisch synchronisiert, so dass alle Knoten im Cluster weiterhin automatisch wie bisher authentifiziert werden und die Clusterknoten miteinander kommunizieren können.

Beachten Sie, dass diese Änderungen nicht abwärtskompatibel sind. Knoten, die auf einem RHEL 7-System authentifiziert wurden, müssen erneut authentifiziert werden.

(BZ#1549535)

### Die **pcs** Befehle unterstützen nun die Anzeige, Bereinigung und Synchronisierung des Zaunverlaufs

Der **zand** daemon des Herzschrittmachers verfolgt eine Historie aller Zaunaktionen (ausstehend, erfolgreich und fehlgeschlagen). Mit dieser Version ermöglichen die **pcs** Befehle den Benutzern den Zugriff auf die Geschichte der Umzäunung auf folgende Weise:

- Der **pcs status** Befehl zeigt fehlgeschlagene und anstehende Fechtaktionen an
- Der **pcs status --full** Befehl zeigt die gesamte Geschichte des Fechtens an
- Der **pcs stonith history** Befehl bietet Optionen zum Anzeigen und Bereinigen des Zaunverlaufs
- Obwohl der Fechtverlauf automatisch synchronisiert wird, unterstützt der **pcs stonith history** Befehl nun eine **update** Option, die es einem Benutzer ermöglicht, den Verlauf des Fechtvorgangs manuell zu synchronisieren, falls dies erforderlich sein sollte

(BZ#1620190, BZ#1615891)

## 4.13. VERNETZUNG (MASCHINENÜBERSETZUNG)

### **nftables** ersetzt **iptables** als Standard-Framework für die Paketfilterung im Netzwerk

Das **nftables** Framework bietet Paketklassifizierungsfunktionen und ist der designierte Nachfolger der **ebtables** Tools **iptables**, **ip6tables**, **arptables**, und. Es bietet zahlreiche Verbesserungen in Bezug auf Komfort, Funktionen und Leistung gegenüber früheren Tools zur Paketfilterung, insbesondere:

- nachschlagetabellen statt linearer Verarbeitung
- ein einheitlicher Rahmen für das Protokoll **IPv4** und **IPv6** die Protokolle
- regeln, die alle atomar angewendet werden, anstatt einen kompletten Regelsatz zu holen, zu aktualisieren und zu speichern

- Unterstützung für Debugging und Tracing im Regelsatz (**nfttrace**) und Überwachung von Trace-Ereignissen (im **nftTool**)
- konsistentere und kompaktere Syntax, keine protokollspezifischen Erweiterungen
- eine Netlink-API für Drittanwendungen

Ähnlich wie **iptables** bei der **nftables** Verwendung von Tabellen zur Speicherung von Ketten. Die Ketten enthalten individuelle Regeln für die Durchführung von Aktionen. Das **nftTool** ersetzt alle Tools aus den vorherigen Paketfilter-Frameworks. Die **libnftables** Bibliothek kann für die Low-Level-Interaktion mit der **nftables** Netlink-API über die **libmnl** Bibliothek verwendet werden.

Die **iptables**, **ip6tables**, **etables** und **arptables** Werkzeuge werden durch **nftables**-basierte Drop-In-Ersatzteile mit dem gleichen Namen ersetzt. Während das externe Verhalten identisch mit dem ihrer Vorgänger ist, verwenden sie intern **nftables** mit **netfilter** Legacy-Kernel-Modulen über eine Kompatibilitätsschnittstelle, falls erforderlich.

Die Wirkung der Module auf den **nftables** Regelsatz kann mit dem **nft list ruleset** Befehl beobachtet werden. Da diese Werkzeuge dem **nftables** Regelsatz Tabellen, Ketten und Regeln hinzufügen, ist zu beachten, dass **nftables** Regelsatzoperationen, wie z.B. der **nft flush ruleset** Befehl, Auswirkungen auf Regelsätze haben können, die mit den zuvor separaten Legacy-Befehlen installiert wurden.

Um schnell zu erkennen, welche Variante des Tools vorhanden ist, wurden die Versionsinformationen aktualisiert, um den Backend-Namen aufzunehmen. In RHEL 8 druckt das **nftables**-basierte **iptables** Tool die folgende Versionszeichenkette aus:

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

Zum Vergleich werden die folgenden Versionsinformationen gedruckt, wenn ein **iptables** Legacy-Tool vorhanden ist:

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

(BZ#1644030)

### Bemerkenswerte TCP-Funktionen in RHEL 8

Red Hat Enterprise Linux 8 wird mit dem TCP-Netzwerkstapel Version 4.16 ausgeliefert, der höhere Leistung, bessere Skalierbarkeit und Stabilität bietet. Insbesondere bei stark ausgelasteten TCP-Servern mit hoher Verbindungsrate werden die Leistungen gesteigert.

Zusätzlich sind zwei neue TCP-Kongestionsalgorithmen **BBR** und **NV**, verfügbar, die in den meisten Szenarien eine geringere Latenz und einen besseren Durchsatz als **cubic** bieten.

(BZ#1562998)

### firewalld verwendet nftables standardmäßig

Mit diesem Update ist das **nftables** Filter-Subsystem das Standard-Firewall-Backend für den **firewalld** Daemon. Um das Backend zu ändern, verwenden Sie die **FirewallBackend** Option in der **/etc/firewalld.conf** Datei.

Diese Änderung führt zu folgenden Unterschieden im Verhalten bei der Verwendung **nftables** von :

1. **iptables** regelausführungen finden immer vor **firewalld** Regeln statt

- **DROP** d.**iptables**. ein Paket wird nie gesehen von **firewalld**
  - **ACCEPT** d.**iptables**. ein Paket unterliegt noch den **firewalld**Regeln
2. **firewalld** direkte Regeln werden weiterhin durchgesetzt **iptables**, während andere **firewalld**Funktionen verwendet werden **nftables**
  3. die direkte Ausführung der Regeln erfolgt vor der **firewalld**generischen Akzeptanz der etablierten Verbindungen

(BZ#1509026)

### Bemerkenswerte Veränderung in **wpa\_supplicant**RHEL 8

In Red Hat Enterprise Linux (RHEL) 8 wird das **wpa\_supplicant**Paket mit **CONFIG\_DEBUG\_SYSLOG**aktiviertem. Dies ermöglicht das Lesen des **wpa\_supplicant**Protokolls mit dem **journalctl**Dienstprogramm, anstatt den Inhalt der **/var/log/wpa\_supplicant.log**Datei zu überprüfen.

(BZ#1582538)

### NetworkManager unterstützt jetzt virtuelle SR-IOV-Funktionen

In Red Hat Enterprise Linux 8.0 ermöglicht **NetworkManager** die Konfiguration der Anzahl der virtuellen Funktionen (VF) für Schnittstellen, die Single Root I/O Virtualization (SR-IOV) unterstützen. Darüber hinaus ermöglicht **NetworkManager** die Konfiguration einiger Attribute der VFs, wie MAC-Adresse, VLAN, **spoof checking**Einstellung und zulässige Bitraten. Beachten Sie, dass alle Eigenschaften von SR-IOV in der **sriov**Verbindungseinstellung verfügbar sind. Weitere Informationen finden Sie in der **nm-settings(5)**Man Page.

(BZ#1555013)

### IPVLAN virtuelle Netzwerktreiber werden nun unterstützt

In Red Hat Enterprise Linux 8.0 bietet der Kernel Unterstützung für virtuelle IPVLAN-Netzwerktreiber. Mit diesem Update ermöglichen virtuelle IPVLAN-Netzwerkschnittstellenkarten (NICs) die Netzwerkkonnektivität für mehrere Container, die eine einzige MAC-Adresse für das lokale Netzwerk freigeben. Dies ermöglicht es einem einzelnen Host, viele Container zu haben, die die mögliche Begrenzung der Anzahl der MAC-Adressen, die von den Peer-Netzwerkgeräten unterstützt werden, überwinden.

(BZ#1261167)

### NetworkManager unterstützt eine Übereinstimmung des Namens der Wildcard-Schnittstelle für Verbindungen

Bisher war es möglich, eine Verbindung zu einer bestimmten Schnittstelle nur mit einer genauen Übereinstimmung des Schnittstellennamens einzuschränken. Mit diesem Update haben Verbindungen eine neue **match.interface-name**Eigenschaft, die Wildcards unterstützt. Mit diesem Update können Benutzer die Schnittstelle für eine Verbindung flexibler über ein Wildcardmuster auswählen.

(BZ#1555012)

### Verbesserungen im Netzwerkstapel 4.18

Red Hat Enterprise Linux 8.0 enthält den auf die Upstream-Version 4.18 aktualisierten Netzwerkstapel, der mehrere Bugfixes und Verbesserungen bietet. Zu den bemerkenswerten Änderungen gehören:

- Es wurden neue Entladefunktionen eingeführt, wie z.B. **UDP\_GSO**, und, für einige Gerätetreiber, **GRO\_HW**.
- Verbesserte signifikante Skalierbarkeit für das User Datagram Protocol (UDP).
- Der generische Abfragecode für beschäftigte Benutzer wurde verbessert.
- Verbesserte Skalierbarkeit des IPv6-Protokolls.
- Verbesserte Skalierbarkeit des Routing-Codes.
- Es wurde ein neuer Standard-Sendewarteschlangenplanungsalgorithmus hinzugefügt, **fq\_codel** der eine Übertragungsverzögerung verbessert.
- Verbesserte Skalierbarkeit für einige Algorithmen zur Sendewarteschlangenplanung. Zum Beispiel, **pfifo\_fast** ist jetzt ohne Sperre.

(BZ#1562987)

### Neue Tools zur Umstellung iptables auf nftables

Dieses Update fügt die **iptables-translate** und **ip6tables-translate** Werkzeuge hinzu, um die bestehenden **iptables** oder **ip6tables** Regeln in die entsprechenden Regeln für **nftables**. Beachten Sie, dass einigen Erweiterungen die Übersetzungsunterstützung fehlt. Wenn eine solche Erweiterung existiert, druckt das Tool die unübersetzte Regel, der das **#** Vorzeichen vorangestellt ist. Zum Beispiel:

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

Darüber hinaus können Benutzer die **iptables-restore-translate** und **ip6tables-restore-translate** Werkzeuge verwenden, um einen Dump von Regeln zu übersetzen. Beachten Sie, dass Benutzer zuvor mit den Befehlen **iptables-save** oder **ip6tables-save** einen Speicherauszug der aktuellen Regeln drucken können. Zum Beispiel:

```
| % 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)

### Neue Funktionen in VPN mit NetworkManager hinzugefügt

In Red Hat Enterprise Linux 8.0 bietet **NetworkManager** die folgenden neuen Funktionen für VPN:

- Unterstützung des Protokolls Internet Key Exchange Version 2 (IKEv2).
- Es wurden weitere **Libreswan-Optionen** hinzugefügt, wie z.B. die **fragmentation** Optionen **rightid**, **leftcert**, **narrowing**, **rekey**, „. Weitere Informationen zu den unterstützten Optionen finden Sie in der **nm-settings-libreswan** Man Page.
- Die Standard-Chiffren wurden aktualisiert. Das bedeutet, dass, wenn der Benutzer die Verschlüsselungen nicht angibt, das **NetworkManager-libreswan-Plugin** es der **Libreswan-Anwendung** ermöglicht, die Standardverschlüsselung des Systems auszuwählen. Die einzige

Ausnahme ist, wenn der Benutzer eine IKEv1-Aggressionsmodus-Konfiguration auswählt. In diesem Fall werden die **ike = aes256-sha1;modp1536** und **eps = aes256-sha1** Werte an **Libreswan** weitergegeben.

(BZ#1557035)

### Ein neuer Daten-Chunk-Typ, I-DATA hinzugefügt zu SCTP

Dieses Update fügt dem Stream Control Transmission Protocol (SCTP) einen neuen Daten-Chunk-Typ und **I-DATA** Stream-Scheduler hinzu. Bisher hat SCTP Benutzernachrichten in der gleichen Reihenfolge gesendet, in der sie von einem Benutzer gesendet wurden. Folglich blockierte eine große SCTP-Benutzernachricht alle anderen Nachrichten in jedem Stream, bis sie vollständig gesendet wurde. Bei der Verwendung von **I-DATA** Chunks ist das Feld Transmission Sequence Number (TSN) nicht überladen. Infolgedessen kann SCTP nun die Streams auf verschiedene Arten planen und **I-DATA** ermöglicht die Verschachtelung von Benutzernachrichten (RFC 8260). Beachten Sie, dass beide Peers den **I-DATA** Chunk-Typ unterstützen müssen.

(BZ#1273139)

## 4.14. SICHERHEIT (MASCHINENÜBERSETZUNG)

### SCAP Sicherheitsleitfaden PCI-DSS-Profil stimmt mit Version 3.2.1 überein

Das **SCAP Security Guide** Projekt stellt das PCI-DSS-Profil (Payment Card Industry Data Security Standard) für Red Hat Enterprise Linux 8 bereit und wurde an die neueste PCI-DSS-Version - 3.2.1 - angepasst.

(BZ#1618528)

### OpenSSH auf Version 7.8p1 umgestellt

Die **openssh** Pakete wurden auf die Upstream-Version 7.8p1 aktualisiert. Zu den bemerkenswerten Änderungen gehören:

- Die Unterstützung für das **SSH version 1** Protokoll wurde entfernt.
- Die Unterstützung für den **hmac-ripemd160** Nachrichten-Authentifizierungscode wurde entfernt.
- Die Unterstützung für RC4 (**arcfour**) Verschlüsselungen wurde entfernt.
- Die Unterstützung für **Blowfish** Verschlüsselungen wurde entfernt.
- Die Unterstützung für **CAST** Verschlüsselungen wurde entfernt.
- Der Standardwert der **UseDNS** Option wurde geändert auf **no**.
- Standardmäßig sind die Algorithmen für **DSA** öffentliche Schlüssel deaktiviert.
- Die minimale Modulgröße für **Diffie-Hellman** Parameter wurde auf 2048 Bit geändert.
- Die Semantik der **ExposeAuthInfo** Konfigurationsoption wurde geändert.
- Die **UsePrivilegeSeparation=sandbox** Option ist nun obligatorisch und kann nicht mehr deaktiviert werden.
- Setzen Sie die minimal akzeptierte **RSA** Schlüsselgröße auf 1024 Bit.

(BZ#1622511)

**RSA-PSS wird nun unterstützt von OpenSC**

Dieses Update bietet dem **OpenSC** Smartcard-Treiber Unterstützung für das kryptografische Signaturschema RSA-PSS. Das neue Schema ermöglicht einen sicheren kryptographischen Algorithmus, der für die TLS 1.3-Unterstützung in der Client-Software erforderlich ist.

(BZ#1595626)

**Bemerkenswerte Änderungen in rsyslogRHEL 8**

Die **rsyslog** Pakete wurden auf die Upstream-Version 8.37.0 aktualisiert, die viele Bugfixes und Verbesserungen gegenüber den Vorgängerversionen bietet. Zu den wichtigsten Änderungen gehören:

- Verbesserte Verarbeitung von internen **rsyslog-Nachrichten**; Möglichkeit der Geschwindigkeitsbegrenzung; mögliche Blockierung behoben.
- Verbesserte Geschwindigkeitsbegrenzung im Allgemeinen; die tatsächliche *Spam-Quelle* wird nun protokolliert.
- Verbessertes Handling von übergroßen Nachrichten - der Benutzer kann nun einstellen, wie er sie sowohl im Kern als auch in bestimmten Modulen mit separaten Aktionen behandeln soll.
- **mmnormalize** regelbasen können nun in die **config** Datei eingebettet werden, anstatt separate Dateien für sie zu erstellen.
- Der Benutzer kann nun die **GnuTLS-Prioritätszeichenfolge** festlegen, **imtcp** die eine fein abgestufte Kontrolle über die Verschlüsselung ermöglicht.
- Alle **config** Variablen, einschließlich der Variablen in JSON, sind nun case-insensitiv.
- Verschiedene Verbesserungen der PostgreSQL-Ausgabe.
- Es wurde die Möglichkeit zugefügt, Shell-Variablen zur Steuerung der **config** Verarbeitung zu verwenden, wie z.B. bedingtes Laden zusätzlicher Konfigurationsdateien, Ausführen von Anweisungen oder Einfügen eines Textes in **config**. Beachten Sie, dass eine übermäßige Nutzung dieser Funktion es sehr schwierig machen kann, Probleme mit **rsyslog** zu beheben.
- 4-stellige Dateierstellungsmodi können nun in **config**.
- Der Eingang des Reliable Event Logging Protocol (RELP) kann nun auch nur noch an eine bestimmte Adresse gebunden werden.
- Der Standardwert der **enable.body** Option Mailausgabe ist nun auf die Dokumentation abgestimmt
- Der Benutzer kann nun Einfügefehlercodes angeben, die in der **MongoDB-Ausgabe** ignoriert werden sollen.
- Der parallele TCP (pTCP)-Eingang hat nun den konfigurierbaren Backlog für einen besseren Lastausgleich.

(BZ#1613880)

**Neues rsyslog-Modul: omkafka**

Um **kafka-zentralisierte** Datenspeicherszenarien zu ermöglichen, können Sie nun mit dem neuen **omkafka** Modul Protokolle an die **kafka-Infrastruktur** weiterleiten.

(BZ#1542497)

### **libssh implementiert SSH als zentrale kryptographische Komponente**

Diese Änderung führt als eine zentrale kryptografische Komponente in Red Hat Enterprise Linux 8 ein **libssh**. Die **libssh** Bibliothek implementiert das Secure SHell (SSH)-Protokoll.

Beachten Sie, dass **libssh** dies nicht mit der systemweiten Krypto-Richtlinie übereinstimmt.

(BZ#1485241)

### **Die PKCS #11-Unterstützung für Smart Cards und HSMs ist nun systemweit einheitlich**

Mit diesem Update wird die Verwendung von Smart Cards und Hardware Security Modules (HSM) mit PKCS #11 kryptographischer Token-Schnittstelle konsistent. Dies bedeutet, dass der Benutzer und der Administrator die gleiche Syntax für alle zugehörigen Werkzeuge im System verwenden können. Zu den bemerkenswerten Verbesserungen gehören:

- Unterstützung für das PKCS #11 Uniform Resource Identifier (URI)-Schema, das eine vereinfachte Aktivierung von Token auf RHEL-Servern sowohl für Administratoren als auch für Anwendungsautoren gewährleistet.
- Ein systemweites Registrierungsverfahren für Smart Cards und HSMs unter Verwendung der **pkcs11.conf**.
- Konsistente Unterstützung für HSMs und Smartcards ist in NSS-, GnuTLS- und OpenSSL-Anwendungen (through the **openssl-pkcs11** engine) verfügbar.
- Der Apache HTTP-Server (**httpd**) unterstützt nun nahtlos HSMs.

Weitere Informationen finden Sie in der **pkcs11.conf(5)** Man Page.

(BZ#1516741)

### **Systemweite kryptografische Richtlinien werden standardmäßig angewendet**

Krypto-Richtlinien sind eine Komponente in Red Hat Enterprise Linux 8, die die wichtigsten kryptographischen Subsysteme konfiguriert, die die Protokolle TLS, IPSec, SSH, DNSSEC und Kerberos abdecken. Es bietet einen kleinen Satz von Richtlinien, die der Administrator mit dem **update-crypto-policies** Befehl auswählen kann.

Die **DEFAULT** systemweite kryptografische Richtlinie bietet sichere Einstellungen für aktuelle Bedrohungsmodelle. Es erlaubt die Protokolle TLS 1.2 und 1.3 sowie die Protokolle IKEv2 und SSH2. Die RSA-Schlüssel und Diffie-Hellman-Parameter werden akzeptiert, wenn sie größer als 2047 Bit sind.

Siehe den [Consistent security by crypto policies in Red Hat Enterprise Linux 8](#) Artikel auf dem Red Hat Blog und der **update-crypto-policies(8)** Man Page für weitere Informationen.

(BZ#1591620)

### **SCAP Security Guide unterstützt OSPP 4.2**

**SCAP Security Guide** enthält einen Entwurf des OSPP-Profiles (Protection Profile for General Purpose Operating Systems) Version 4.2 für Red Hat Enterprise Linux 8. Dieses Profil spiegelt die obligatorischen Konfigurationskontrollen wider, die im NIAP-Konfigurationsanhang des Schutzprofils für



allgemeine Betriebssysteme (Schutzprofil Version 4.2) aufgeführt sind. Der SCAP Security Guide bietet automatisierte Prüfungen und Skripte, mit denen Benutzer die im OSPP definierten Anforderungen erfüllen können.

(BZ#1618518)

### Die OpenSCAP-Befehlszeilenschnittstelle wurde verbessert

Der ausführliche Modus ist nun in allen **oscap**-Modulen und Submodulen verfügbar. Die Werkzeugausgabe hat eine verbesserte Formatierung.

Veraltete Optionen wurden entfernt, um die Benutzerfreundlichkeit der Befehlszeilenschnittstelle zu verbessern.

Die folgenden Optionen sind nicht mehr verfügbar:

- **--show** in **oscap xccdf generate report** wurde vollständig entfernt.
- **--probe-root** in **oscap oval eval** wurde entfernt. Sie kann durch Setzen der Umgebungsvariablen ersetzt werden, **OSCAP\_PROBE\_ROOT**.
- **--sce-results** in **oscap xccdf eval** wurde ersetzt durch **--check-engine-results**
- **validate-xml** submodul wurde aus CPE-, OVAL- und XCCDF-Modulen entfernt. **validate** Submodule können stattdessen verwendet werden, um SCAP-Inhalte anhand von XML-Schemata und XSD-Schemata zu validieren.
- **oscap oval list-probes** befehl entfernt wurde, kann **oscap --version** stattdessen die Liste der verfügbaren Sonden angezeigt werden.

OpenSCAP ermöglicht es, alle Regeln in einem bestimmten XCCDF-Benchmark unabhängig vom Profil zu bewerten, indem es **--profile '(all)'**.

(BZ#1618484)

### Unterstützung für eine neue Berechtigungsprüfung der Karte auf dem **mmapsyscall**

Die **map** SELinux-Berechtigung wurde hinzugefügt, um den Speicherabbildungszugriff auf Dateien, Verzeichnisse, Sockets usw. zu steuern. Dadurch kann die SELinux-Richtlinie den direkten Speicherzugriff auf verschiedene Dateisystemobjekte verhindern und sicherstellen, dass jeder dieser Zugriffe erneut validiert wird.

(BZ#1592244)

### SELinux unterstützt jetzt **systemd No New Privileges**

Dieses Update stellt die **nnp\_nosuid\_transition** Richtlinienfunktion vor, die SELinux-Domänenübergänge unter (**NNP No New Privileges**) oder **nosuid** wenn **nnp\_nosuid\_transition** zwischen dem alten und dem neuen Kontext erlaubt ist. Die **selinux-policy** Pakete enthalten nun eine Richtlinie für **Systemdienste**, die die **NNP** Sicherheitsfunktion verwenden.

Die folgende Regel beschreibt das Zulassen dieser Fähigkeit für einen Dienst:

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

Zum Beispiel:

■

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

Die Distributionsrichtlinie enthält nun auch die m4-Makroschnittstelle, die in SELinux-Sicherheitsrichtlinien für Dienste, die die **init\_nnp\_daemon\_domain()**Funktion nutzen, verwendet werden kann.

(BZ#1594111)

### SELinux unterstützt nun **getrlimit**Berechtigungen in der **process**Klasse

Mit diesem Update wird eine neue SELinux-Zugangskontrollprüfung eingeführt, **process:getrlimit**die für die **prlimit()**Funktion hinzugefügt wurde. Auf diese Weise können SELinux-Richtlinienentwickler steuern, wann ein Prozess versucht, die Ressourcenbegrenzungen eines anderen Prozesses mit der **process:setrlimit**Berechtigung zu lesen und dann zu ändern. Beachten Sie, dass SELinux einen Prozess nicht daran hindert, seine eigenen Ressourcengrenzen durch **prlimit()**. Weitere Informationen finden Sie auf den Seiten **prlimit(2)**und **getrlimit(2)**in der Bedienungsanleitung.

(BZ#1549772)

### TLS 1.3-Unterstützung in kryptographischen Bibliotheken

Dieses Update ermöglicht Transport Layer Security (TLS) 1.3 standardmäßig in allen wichtigen Backend-Kryptobibliotheken. Dies ermöglicht eine geringe Latenzzeit auf der Kommunikationsschicht des Betriebssystems und erhöht die Privatsphäre und Sicherheit von Anwendungen durch die Nutzung neuer Algorithmen wie RSA-PSS oder X25519.

(BZ#1516728)

### Neue Funktionen in **OpenSCAP**RHEL 8

Die **OpenSCAP**Suite wurde auf die Upstream-Version 1.3.0 aktualisiert, die viele Verbesserungen gegenüber den Vorgängerversionen enthält. Zu den bemerkenswertesten Merkmalen gehören:

- API und ABI wurden konsolidiert - aktualisierte, veraltete und/oder unbenutzte Symbole wurden entfernt.
- Die Sonden werden nicht als unabhängige Prozesse, sondern als Threads innerhalb des **oscap**Prozesses ausgeführt.
- Die Befehlszeilenschnittstelle wurde aktualisiert.
- **Python 2** bindungen wurden durch **Python 3** Bindungen ersetzt.

(BZ#1614273)

### **Audit 3.0** ersetzt **audispd**durch **auditd**

Mit diesem Update wurde die Funktionalität von **audispd**.... nach **auditd**.... verschoben. Infolgedessen sind **audispd**Konfigurationsoptionen nun Teil von **auditd.conf**. Außerdem wurde das **plugins.d**Verzeichnis verschoben unter **/etc/audit**. Der aktuelle Status von **auditd**und seiner Plugins kann nun durch Ausführen des **service auditd state**Befehls überprüft werden.

(BZ#1616428)

### **rsyslogimfile** unterstützt jetzt **Symlinks**

Mit diesem Update bietet das **rsyslog-Modulimfile** eine bessere Leistung und mehr Konfigurationsmöglichkeiten. Dies ermöglicht es Ihnen, das Modul für komplexere Anwendungsfälle der

Dateiüberwachung zu verwenden. So können Sie beispielsweise Datei-Monitore mit Glob-Mustern an beliebiger Stelle entlang des konfigurierten Pfades verwenden und Symlink-Ziele mit erhöhtem Datendurchsatz drehen.

(BZ#1614179)

### Die automatische OpenSSHGenerierung von Serverschlüsseln wird nun von folgenden Personen durchgeführt sshd-keygen@.service

**OpenSSH** erstellt automatisch RSA-, ECDSA- und ED25519-Server-Hostschlüssel, wenn sie fehlen. Um die Hostschlüsselerstellung in RHEL 8 zu konfigurieren, verwenden Sie den **sshd-keygen@.service**instanziierten Service.

Zum Beispiel, um die automatische Erstellung des RSA-Schlüsseltyps zu deaktivieren:

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

Weitere Informationen finden Sie in der **/etc/sysconfig/sshd** Datei.

(BZ#1228088)

### Das rsyslogStandard-Konfigurationsdateiformat ist nun nicht mehr veraltet

Die Konfigurationsdateien in den **rsyslog** Paketen verwenden nun standardmäßig das Non-Legacy-Format. Das Legacy-Format kann weiterhin verwendet werden, jedoch hat die Mischung von aktuellen und älteren Konfigurationsanweisungen mehrere Einschränkungen. Konfigurationen, die aus früheren RHEL-Releases übernommen wurden, sollten überarbeitet werden. Weitere Informationen finden Sie in der **rsyslog.conf(5)** Man Page.

(BZ#1619645)

### Neue SELinux-Booleans

Dieses Update der SELinux-Systemrichtlinie führt die folgenden Booleans ein:

- `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_pcsd`

Weitere Informationen finden Sie in der Ausgabe des folgenden Befehls:

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

## 4.15. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)

## KVM unterstützt 5-stufiges Paging

Mit Red Hat Enterprise Linux 8 unterstützt die KVM-Virtualisierung die 5-stufige Paging-Funktion, die den physischen und virtuellen Adressraum, den die Host- und Gastsysteme nutzen können, erheblich erweitert.

(BZ#1485229)

## KVM unterstützt UMIP in RHEL 8

Die KVM-Virtualisierung unterstützt nun die Funktion User-Mode Instruction Prevention (UMIP), die verhindern kann, dass Benutzerraumanwendungen auf systemweite Einstellungen zugreifen. Dies reduziert die potenziellen Vektoren für Privilegien-Eskalationsangriffe und macht so den KVM-Hypervisor und seine Gastmaschinen sicherer.

(BZ#1494651)

## Zusätzliche Informationen in KVM-Gast Crash-Berichten

Die Absturzinformationen, die der KVM-Hypervisor erzeugt, wenn ein Gast unerwartet beendet oder nicht mehr reagiert, wurden erweitert. Dies erleichtert die Diagnose und Behebung von Problemen in KVM-Virtualisierungsimplementierungen.

(BZ#1508139)

## qemu-kvm 2.12 in RHEL 8

Red Hat Enterprise Linux 8 wird mit **2qemu-kvm.12** ausgeliefert. Diese Version behebt mehrere Fehler und bietet eine Reihe von Verbesserungen gegenüber der Version 1.5.3, die in Red Hat Enterprise Linux 7 verfügbar ist.

Insbesondere wurden die folgenden Funktionen eingeführt:

- Q35 Gastmaschine Typ
- UEFI Gastboot
- NUMA-Tuning und Pinning im Gast
- vCPU hot plug und hot unplugging
- gast I/O Threading

Beachten Sie, dass einige der in **2qemu-kvm.12** verfügbaren Funktionen von Red Hat Enterprise Linux 8 nicht unterstützt werden. Detaillierte Informationen finden Sie unter "Funktionsunterstützung und Einschränkungen bei der RHEL 8-Virtualisierung" im Red Hat Kundenportal.

(BZ#1559240)

## NVIDIA vGPU ist jetzt kompatibel mit der VNC-Konsole

Bei Verwendung der NVIDIA Virtual GPU (vGPU) Funktion ist es nun möglich, über die VNC-Konsole die visuelle Ausgabe des Gastes anzuzeigen.

(BZ#1497911)

## Ceph wird durch Virtualisierung unterstützt

Mit diesem Update wird der Ceph-Speicher durch KVM-Virtualisierung auf allen von Red Hat unterstützten CPU-Architekturen unterstützt.

(BZ#1578855)

### **Der Q35-Maschinentyp wird nun durch Virtualisierung unterstützt**

Red hat Enterprise Linux 8 stellt die Unterstützung für **Q35** vor, einen moderneren PCI Express-basierten Maschinentyp. Dies bietet eine Vielzahl von Verbesserungen in den Funktionen und der Leistung virtueller Geräte und stellt sicher, dass eine größere Bandbreite moderner Geräte mit der Virtualisierung kompatibel ist. Darüber hinaus sind virtuelle Maschinen, die in Red Hat Enterprise Linux 8 erstellt wurden, standardmäßig auf **Q35** eingestellt.

Beachten Sie auch, dass der zuvor voreingestellte **PC-Maschinentyp** veraltet ist und nur noch bei der Virtualisierung älterer Betriebssysteme verwendet werden sollte, die Q35 nicht unterstützen.

(BZ#1599777)

### **Interaktiver Bootloader für virtuelle Maschinen auf IBM Z**

Beim Booten einer virtuellen Maschine (VM) auf einem IBM Z-Host präsentiert die QEMU-Bootloader-Firmware nun eine interaktive Konsolenschnittstelle. Dies ermöglicht es, Probleme beim Booten des Gastbetriebssystems ohne Zugriff auf die Host-Umgebung zu beheben.

(BZ#1508137)

### **QEMU Sandboxing wurde hinzugefügt**

In Red Hat Enterprise Linux 8 führt der QEMU-Emulator die Sandboxing-Funktion ein. QEMU Sandboxing bietet konfigurierbare Einschränkungen für die Leistung von QEMU-Systemen und macht so virtuelle Maschinen sicherer. Beachten Sie, dass diese Funktion aktiviert und standardmäßig konfiguriert ist.

(JIRA:RHELPLAN-10628)

### **GFNI und CLDEMOT Befehlsätze für Intel Xeon SnowRidge aktiviert**

Virtuelle Maschinen (VMs), die auf einem RHEL 8-Host auf einem Intel Xeon SnowRidge-System laufen, können nun die Befehlsätze GFNI und CLDEMOT verwenden. Dies kann in bestimmten Szenarien die Performance solcher VMs deutlich erhöhen.

(BZ#1494705)

### **Ein VFIO-basierter Blocktreiber für NVMe-Geräte wurde hinzugefügt**

Der QEMU-Emulator stellt einen VFIO-basierten Treiber für nichtflüchtige Speicher-Express (NVMe)-Geräte vor. Der Treiber kommuniziert direkt mit NVMe-Geräten, die an virtuelle Maschinen (VMs) angeschlossen sind, und vermeidet die Verwendung der Kernelsystemschiicht und ihrer NVMe-Treiber. Dadurch wird die Leistung von NVMe-Geräten in virtuellen Maschinen verbessert.

(BZ#1519004)

### **Verbesserte Unterstützung für große Seiten**

Bei Verwendung von RHEL 8 als Virtualisierungshost können Benutzer die Größe der Seiten, die den Speicher einer virtuellen Maschine (VM) zurücksetzen, auf jede Größe ändern, die von der CPU unterstützt wird. Dies kann die Performance der VM deutlich verbessern.

Um die Größe der VM-Speicherseiten zu konfigurieren, bearbeiten Sie die XML-Konfiguration der VM und fügen Sie das Element `<hugepages>` dem Abschnitt `<memoryBacking>` hinzu.

(JIRA:RHELPLAN-14607)

## 4.16. SUPPORTFÄHIGKEIT (MASCHINENÜBERSETZUNG)

### **sosreport kann eBPF-basierte Programme und Karten melden**

Das **sosreport** Tool wurde erweitert, um alle geladenen erweiterten Berkeley Packet Filtering (eBPF) Programme und Karten in Red Hat Enterprise Linux 8 zu melden.

(BZ#1559836)

## KAPITEL 5. TECHNOLOGIEVORSCHAUEN (MASCHINENÜBERSETZUNG)

Dieser Teil enthält eine Liste aller in Red Hat Enterprise Linux 8.0 verfügbaren Technologievorschau.

Informationen zum Umfang der Unterstützung von Red Hat für Funktionen der Technologievorschau finden Sie unter [Technology Preview Features Support Scope](#).

### 5.1. KERNEL (MASCHINENÜBERSETZUNG)

#### XDP als Technologievorschau verfügbar

Die Funktion **eXpress Data Path (XDP)**, die als Technologievorschau verfügbar ist, bietet die Möglichkeit, Berkeley Packet Filter (BPF)-Programme für eine leistungsstarke Paketverarbeitung im Kernel hochzuladen, wodurch der Netzwerkdatenpfad des Kernels programmierbar wird.

(BZ#1503672)

#### eBPF als Technologievorschau verfügbar

Die **erweiterte Berkeley Packet Filtering (eBPF)** Funktion ist als Technologievorschau sowohl für die Vernetzung als auch für das Tracing verfügbar. **eBPF** ermöglicht es dem Benutzer, benutzerdefinierte Programme an eine Vielzahl von Punkten (Sockets, Tracing Points, Paketempfang) anzuhängen, um Daten zu empfangen und zu verarbeiten. Die Funktion beinhaltet einen neuen Systemaufruf **bpf()**, der es ermöglicht, verschiedene Arten von Maps zu erstellen und auch verschiedene Arten von Programmen in den Kernel einzufügen. Weitere Informationen finden Sie in der Man Page **bpf(2)**.

(BZ#1559616)

#### BCC ist als Technologievorschau verfügbar

**BPF Compiler Collection (BCC)** ist ein User Space Toolkit zur Erstellung effizienter Kernel-Tracing- und Manipulationsprogramme, das als Technologievorschau in Red Hat Enterprise Linux 8 verfügbar ist. Es **BCC** bietet Werkzeuge zur I/O-Analyse, Vernetzung und Überwachung von Linux-Betriebssystemen mit dem **extended Berkeley Packet Filtering (eBPF)**.

(BZ#1548302)

#### Kontrollgruppe v2 als Technologievorschau in RHEL 8 verfügbar

Der Mechanismus der **Kontrollgruppe v2** ist eine einheitliche Hierarchiekontrollgruppe. Die **Kontrollgruppe v2** organisiert Prozesse hierarchisch und verteilt die Systemressourcen entlang der Hierarchie kontrolliert und konfigurierbar.

Im Gegensatz zur Vorgängerversion hat die **Kontrollgruppe v2** nur eine einzige Hierarchie. Diese einzige Hierarchie ermöglicht es dem Linux-Kernel,:

- Kategorisieren Sie Prozesse basierend auf der Rolle ihres Eigentümers.
- Beseitigen Sie Probleme mit widersprüchlichen Richtlinien mehrerer Hierarchien.

Die **Kontrollgruppe v2** unterstützt zahlreiche Controller:

- Die CPU-Steuerung regelt die Verteilung der CPU-Zyklen. Diese Steuerung implementiert:
  - Gewichts- und absolute Bandbreitenbegrenzungsmodelle für normale Planungsrichtlinien.

- Absolutes Bandbreitenzuweisungsmodell für Echtzeit-Planungsrichtlinien.
- Die Speichersteuerung regelt die Speicherverteilung. Derzeit werden die folgenden Arten von Speicherauslastungen verfolgt:
  - Userland-Speicher - Seiten-Cache und anonymer Speicher.
  - Kernel-Datenstrukturen wie Dentries und Inodes.
  - TCP-Socket-Puffer.
- Die I/O-Steuerung regelt die Verteilung der I/O-Ressourcen.
- Der Writeback-Controller interagiert sowohl mit Speicher- als auch mit I/O-Controllern und ist **Control Group v2-spezifisch**.

Die obigen Informationen basieren auf dem Link: <https://www.kernel.org/doc/Documentation/cgroup-v2.txt>. Sie können auf den gleichen Link verweisen, um weitere Informationen über bestimmte Controller der Kontrollgruppe v2 zu erhalten.

(BZ#1401552)

### **early kdump verfügbar als Technologievorschau in Red Hat Enterprise Linux 8**

Die **early kdump** Funktion ermöglicht es dem Absturzkern und initsramfs, früh genug zu laden, um die **vmcore** Informationen auch bei frühen Abstürzen zu erfassen. Weitere Informationen dazu **early kdump** finden Sie in der `/usr/share/doc/kexec-tools/early-kdump-howto.txt` Datei.

(BZ#1520209)

## **5.2. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)**

### **VDO als logischer LVM-Volume-Typ verfügbar**

LVM kann nun verwendet werden, um logische Volumes vom Typ *Virtual Data Optimizer* (VDO) zu erstellen. VDO ist ein virtuelles Blockgerät mit der Fähigkeit, Daten zu komprimieren und zu deduplizieren.

Dies ist eine Technologievorschau-Funktion.

(BZ#1643553)

### **Unterstützung für Datenintegritätsfeld/Datenintegritätsenerweiterung (DIF/DIX)**

DIF/DIX ist eine Ergänzung zum SCSI-Standard. Es bleibt in der Technologievorschau für alle HBAs und Speicher-Arrays, mit Ausnahme derjenigen, die speziell als unterstützt aufgeführt sind.

DIF/DIX vergrößert die Größe des häufig verwendeten 512-Byte-Festplattenblocks von 512 auf 520 Bytes und fügt das Data Integrity Field (DIF) hinzu. Das DIF speichert einen Prüfsummenwert für den Datenblock, der vom Host Bus Adapter (HBA) berechnet wird, wenn ein Schreiben erfolgt. Das Speichermedium bestätigt dann beim Empfang die Prüfsumme und speichert sowohl die Daten als auch die Prüfsumme. Umgekehrt kann die Prüfsumme beim Lesen durch das Speichermedium und durch den empfangenden HBA verifiziert werden.

(BZ#1649493)

### **NVMe/FC ist als Technologievorschau in Qlogic-Adaptoren verfügbar qla2xxx**



Der Transporttyp NVMe over Fibre Channel (NVMe/FC) ist als Technologievorschau in Qlogic-Adaptern mit dem **qla2xxx**Treiber verfügbar.

(BZ#1649922)

### 5.3. HOCHVERFÜGBARKEIT UND CLUSTER (MASCHINENÜBERSETZUNG)

#### Herzschrittmacher-Bundles **podman**als Technologievorschau verfügbar

Herzschrittmacher-Containerbündel laufen nun auf der **podman**Containerplattform, wobei die Containerbündelfunktion als Technologievorschau verfügbar ist. Es gibt eine Ausnahme von dieser Funktion, nämlich die Technologievorschau: Red Hat unterstützt die Verwendung von Pacemaker-Bundles für Red Hat Openstack vollständig.

(BZ#1619620)

### 5.4. SICHERHEIT (MASCHINENÜBERSETZUNG)

#### SWID-Tag der RHEL 8.0-Version

Um die Identifizierung von RHEL 8.0-Installationen mit Hilfe des ISO/IEC 19770-2:2015-Mechanismus zu ermöglichen, werden SWID-Tags (Software Identification) in Dateien **/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8-<architecture>.swidtag** und **/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.swidtag**. Das übergeordnete Verzeichnis dieser Tags kann auch über den **/etc/swid/swidtags.d/redhat.com**symbolischen Link gefunden werden.

Die XML-Signatur der SWID-Tag-Dateien kann z.B. mit dem **xmlsec1 verify**Befehl verifiziert werden:

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

Das Zertifikat der Zertifizierungsstelle für Code Signing kann auch über die [Product Signing Keys](#)Seite im Kundenportal bezogen werden.

(BZ#1636338)

### 5.5. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)

#### AMD SEV für KVM virtuelle Maschinen

Als Technologievorschau stellt RHEL 8 die Funktion Secure Encrypted Virtualization (SEV) für AMD EPYC-Hostmaschinen vor, die den KVM-Hypervisor verwenden. Wenn auf einer virtuellen Maschine (VM) aktiviert, verschlüsselt SEV den VM-Speicher, so dass der Host nicht auf Daten auf der VM zugreifen kann. Dies erhöht die Sicherheit der VM, wenn der Host erfolgreich mit Malware infiziert ist.

Beachten Sie, dass die Anzahl der VMs, die diese Funktion gleichzeitig auf einem einzelnen Host nutzen können, von der Host-Hardware bestimmt wird. Aktuelle AMD EPYC-Prozessoren unterstützen 15 oder weniger laufende VMs mit SEV.

(BZ#1501618, BZ#1501607)

Verschachtelte Virtualisierung jetzt auch auf IBM POWER 9 verfügbar

Als Technologievorschau ist es nun möglich, die verschachtelten Virtualisierungsfunktionen auf RHEL 8-Hostmaschinen auf IBM POWER 9-Systemen zu nutzen. Verschachtelte Virtualisierung ermöglicht es KVM Virtual Machines (VMs), als Hypervisor zu fungieren, was es ermöglicht, VMs innerhalb von VMs auszuführen.

Beachten Sie, dass der Host, der Gast und die verschachtelten Gäste derzeit alle eines der folgenden Betriebssysteme ausführen müssen, damit verschachtelte Virtualisierung mit IBM POWER 9 funktioniert:

- RHEL 8
- RHEL 7 für POWER 9

(BZ#1505999)

## KAPITEL 6. VERALTETE FUNKTIONALITÄT (MASCHINENÜBERSETZUNG)

Dieser Teil gibt einen Überblick über die Funktionalität, die in Red Hat Enterprise Linux 8.0 veraltet ist.

Veraltete Funktionen werden bis zum Ende der Lebensdauer von Red Hat Enterprise Linux 8 weiterhin unterstützt. Veraltete Funktionen werden in zukünftigen Hauptversionen dieses Produkts wahrscheinlich nicht mehr unterstützt und werden für neue Implementierungen nicht empfohlen. Die aktuellste Liste der veralteten Funktionen innerhalb eines bestimmten Hauptrelease finden Sie in der neuesten Version der Release-Dokumentation.

Veraltete Hardwarekomponenten werden für neue Implementierungen in den aktuellen oder zukünftigen Hauptversionen nicht empfohlen. Hardware-Treiber-Updates sind auf Sicherheit und kritische Korrekturen beschränkt. Red Hat empfiehlt, diese Hardware so schnell wie möglich zu ersetzen.

Ein Paket kann veraltet sein und nicht für die weitere Verwendung empfohlen werden. Unter bestimmten Umständen kann eine Verpackung aus einem Produkt entnommen werden. Die Produktdokumentation identifiziert dann neuere Pakete, die eine ähnliche, identische oder fortgeschrittene Funktionalität wie die veraltete bieten, und liefert weitere Empfehlungen.

### 6.1. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)

#### Die `--interactive` Option des Befehls `ignorediskKickstart` wurde verworfen

Die Verwendung der `--interactive option` zukünftigen Versionen von Red Hat Enterprise Linux führt zu einem fatalen Installationsfehler. Es wird empfohlen, dass Sie Ihre Kickstart-Datei ändern, um die Option zu entfernen.

(BZ#1637872)

### 6.2. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)

#### NFSv3 über UDP wurde deaktiviert

Der NFS-Server öffnet oder hört standardmäßig nicht mehr auf einem User Datagram Protocol (UDP)-Sockel. Diese Änderung betrifft nur NFS Version 3, da Version 4 das Transmission Control Protocol (TCP) erfordert.

NFS über UDP wird in RHEL 8 nicht mehr unterstützt.

(BZ#1592011)

#### Der NVMe/FC-Zielmodus ist veraltet

Der Zielmodus des Nonvolatile Memory Express over Fibre Channel (NVMe/FC) Transportprotokolls war bisher als Technologievorschau in RHEL 7 verfügbar. In RHEL 8 ist der NVMe/FC-Zielmodus veraltet.

Das Aktivieren von FC-Hostbus-Adapter-(HBA)-Ports im NVMe-Zielmodus führt zu folgender Fehlermeldung:

**Warning: NVMe over FC Target - This driver has not undergone sufficient testing by Red Hat for this release and therefore cannot be used in production systems.**

(BZ#1664838)

## 6.3. VERNETZUNG (MASCHINENÜBERSETZUNG)

### Netzwerkskripte sind in RHEL 8 veraltet

Netzwerkskripte sind in Red Hat Enterprise Linux 8 veraltet und werden standardmäßig nicht mehr angeboten. Die Basisinstallation enthält eine neue Version des **ifup** und der **ifdown** Skripte, die den **NetworkManager-Dienst** über das **nmcli-Tool** aufrufen. In Red Hat Enterprise Linux 8 muss NetworkManager ausgeführt werden, um die **ifup** und die **ifdown** Skripte auszuführen.

Beachten Sie, dass benutzerdefinierte Befehle in **/sbin/ifup-local**, **ifdown-pre-local** und **ifdown-local** Skripte nicht ausgeführt werden.

Wenn eines dieser Skripte benötigt wird, ist die Installation der veralteten Netzwerkskripte im System mit dem folgenden Befehl weiterhin möglich:

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

Die **ifup** und **ifdown** Skripte verweisen auf die installierten älteren Netzwerkskripte.

Der Aufruf der alten Netzwerkskripte zeigt eine Warnung über ihre Verwerfung an.

(BZ#1647725)

## 6.4. SICHERHEIT (MASCHINENÜBERSETZUNG)

### DSA ist in Red Hat Enterprise Linux 8 veraltet

Der Digital Signature Algorithmus (DSA) gilt in Red Hat Enterprise Linux 8 als veraltet. Authentifizierungsmechanismen, die von DSA-Schlüsseln abhängen, funktionieren in der Standardkonfiguration nicht. Beachten Sie, dass **OpenSSH** Clients auch in der LEGACY-Richtlinie keine DSA-Hostschlüssel akzeptieren.

(BZ#1646541)

### SSL2Client Hello wurde in NSS veraltet

Das **TLS-Protokoll** (Transport Layer Security) Version 1.2 und früher ermöglicht es, eine Verhandlung mit einer **Client Hello** Nachricht zu beginnen, die so formatiert ist, dass sie mit dem **SSL-Protokoll** (Secure Sockets Layer) Version 2 rückwärts kompatibel ist. Die Unterstützung für diese Funktion in der Bibliothek der Network Security Services (**NSS**) wurde veraltet und ist standardmäßig deaktiviert.

Anwendungen, die Unterstützung für diese Funktion benötigen, müssen die neue **SSL\_ENABLE\_V2\_COMPATIBLE\_HELLO** API verwenden, um sie zu aktivieren. Die Unterstützung für diese Funktion kann in zukünftigen Versionen von Red Hat Enterprise Linux 8 vollständig entfernt werden.

(BZ#1645153)

## 6.5. VIRTUALISIERUNG (MASCHINENÜBERSETZUNG)

Snapshots von virtuellen Maschinen werden in RHEL 8 nicht richtig unterstützt

Der aktuelle Mechanismus zum Erstellen von Snapshots für virtuelle Maschinen (VM) wurde veraltet, da er nicht zuverlässig funktioniert. Daher wird empfohlen, in RHEL 8 keine VM-Snapshots zu verwenden.

Beachten Sie, dass ein neuer VM-Snapshot-Mechanismus in Entwicklung ist und in einem zukünftigen Minor-Release von RHEL 8 vollständig implementiert wird.

([BZ#1686057](#))

### **Der virtuelle Cirrus VGA-GPU-Typ wurde veraltet**

Mit einem zukünftigen größeren Update von Red Hat Enterprise Linux wird das **Cirrus VGA-GPU-Gerät** in virtuellen KVM-Maschinen nicht mehr unterstützt. Red Hat empfiehlt daher, die Geräte **stdvga**, **virtio-vga** oder **qxl** anstelle von Cirrus VGA zu verwenden.

(BZ#1651994)

### **virt-manager wurde veraltet**

Die Anwendung Virtual Machine Manager, auch bekannt als **virt-manager**, wurde veraltet. Die RHEL 8 Webkonsole, auch bekannt als **Cockpit**, soll in einem späteren Release zu ihrem Ersatz werden. Es wird daher empfohlen, die Webkonsole zur Verwaltung der Virtualisierung in einer GUI zu verwenden. In Red Hat Enterprise Linux 8.0 sind einige Funktionen jedoch möglicherweise nur über den **virt-manager** oder die Befehlszeile zugänglich.

(JIRA:RHELPLAN-10304)

## KAPITEL 7. BEKANNTE PROBLEME (MASCHINENÜBERSETZUNG)

Dieser Teil beschreibt bekannte Probleme in Red Hat Enterprise Linux 8.

### 7.1. INSTALLATIONSPROGRAMM UND IMAGE-ERSTELLUNG (MASCHINENÜBERSETZUNG)

#### Die Befehle **auth** und **authconfig** Kickstart erfordern das AppStream-Repository

Das **authselect-compat** Paket wird von den Befehlen **auth** und **authconfig** Kickstart während der Installation benötigt. Ohne dieses Paket schlägt die Installation fehl, wenn **auth** oder **authconfig** werden sie verwendet. Das **authselect-compat** Paket ist jedoch per Design nur im AppStream-Repository verfügbar.

Um dieses Problem zu umgehen, stellen Sie sicher, dass die Repositories BaseOS und AppStream für das Installationsprogramm verfügbar sind, oder verwenden Sie während der Installation den Befehl **authselect** Kickstart.

(BZ#1640697)

#### Das Kopieren des Inhalts der **Binary DVD.iso** Datei auf eine Partition lässt die **.treeinfo** und **.discinfo** Dateien weg

Wenn Sie den Inhalt der RHEL 8.0 Binär-DVD.iso-Image-Datei auf eine Partition für eine lokale Installation kopieren, kann das '\*' im Befehl **cp <path>/^\* <mounted partition>/dir** die Dateien **.treeinfo**, **.discinfo** die für eine erfolgreiche Installation erforderlich sind, nicht kopieren. Infolgedessen werden die Repositories BaseOS und AppStream nicht geladen, und eine Debug-Logmeldung in der **anaconda.log** Datei ist der einzige Datensatz des Problems.

Um das Problem zu umgehen, kopieren Sie die fehlenden **.treeinfo** Dateien und **.discinfo** Dateien auf die Partition.

(BZ#1692746)

### 7.2. KERNEL (MASCHINENÜBERSETZUNG)

#### Das **i40iw**-Modul wird nicht automatisch beim Booten geladen

Da viele i40e-NICs iWarp nicht unterstützen und das **i40iw-Modul** nicht vollständig suspend/resume unterstützt, wird dieses Modul standardmäßig nicht automatisch geladen, um sicherzustellen, dass suspend/resume ordnungsgemäß funktioniert. Um dieses Problem zu umgehen, bearbeiten Sie die **/lib/udev/rules.d/90-rdma-hw-modules.rules** Datei manuell, um das automatische Laden von **i40iw** zu ermöglichen.

Beachten Sie auch, dass, wenn ein anderes RDMA-Gerät mit einem i40e Gerät auf derselben Maschine installiert ist, das nicht-i40e RDMA-Gerät den **rdma-Dienst** auslöst, der alle aktivierten RDMA-Stapelmodule lädt, einschließlich des **i40iw-Moduls**.

(BZ#1623712)

#### Das System reagiert manchmal nicht mehr, wenn viele Geräte angeschlossen sind

Wenn Red Hat Enterprise Linux 8 eine große Anzahl von Geräten konfiguriert, tritt eine große Anzahl von Konsolenmeldungen auf der Systemkonsole auf. Dies geschieht beispielsweise, wenn es eine große

Anzahl von Logical Unit Numbers (LUNs) mit mehreren Pfaden zu jeder LUN gibt. Die Flut von Konsolenmeldungen, zusätzlich zu anderer Arbeit, die der Kernel leistet, kann dazu führen, dass der Kernel-Watchdog eine Kernelpanik erzwingt, weil der Kernel scheinbar hängen bleibt.

Da der Scan zu Beginn des Bootzyklus stattfindet, reagiert das System nicht mehr, wenn viele Geräte angeschlossen sind. Dies geschieht typischerweise beim Booten.

Wenn **kdump** auf Ihrem Computer während des Geräte-Scan-Ereignisses nach dem Booten aktiviert ist, führt die Hardlockup zu einer Aufnahme eines **vmcore**Bildes.

Um dieses Problem zu umgehen, erhöhen Sie den Watchdog-Lockup-Timer. Um dies zu tun, fügen Sie die **watchdog\_thresh=N** Option der Kernel-Befehlszeile hinzu. Ersetzen **N** Sie diese durch die Anzahl der Sekunden:

- Wenn Sie weniger als tausend Geräte haben, verwenden Sie **30**.
- Wenn Sie mehr als tausend Geräte haben, verwenden Sie **60**.

Für die Speicherung ist die Anzahl der Geräte die Anzahl der Pfade zu allen LUNs: im Allgemeinen die Anzahl der **/dev/sd\***Geräte.

Nach der Anwendung des Workarounds reagiert das System nicht mehr, wenn es eine große Anzahl von Geräten konfiguriert.

(BZ#1598448)

### KSM ignoriert manchmal NUMA-Speicherrichtlinien

Wenn die Kernel Shared Memory (KSM)-Funktion mit dem Parameter "merge\_across\_nodes=1" aktiviert ist, ignoriert KSM die von der Funktion mbind() gesetzten Speicherrichtlinien und kann Seiten aus einigen Speicherbereichen mit NUMA-Knoten (Non-Uniform Memory Access) zusammenführen, die nicht den Richtlinien entsprechen.

Um dieses Problem zu umgehen, deaktivieren Sie KSM oder setzen Sie den Parameter merge\_across\_nodes auf "0", wenn Sie die NUMA-Speicherbindung mit QEMU verwenden. Infolgedessen funktionieren die für die KVM-VM konfigurierten NUMA-Speicherrichtlinien wie erwartet.

(BZ#1153521)

## 7.3. SOFTWARE-MANAGEMENT (MASCHINENÜBERSETZUNG)

### Das Ausführen yum list unter einem Nicht-Root-Benutzer verursacht einen YUM-Absturz

Wenn der **yum list** Befehl unter einem Nicht-Root-Benutzer ausgeführt wird, nachdem das **libdnf** Paket aktualisiert wurde, kann **YUM** unerwartet beendet werden. Wenn Sie diesen Fehler gefunden haben, führen Sie ihn **yum list** unter root aus, um das Problem zu beheben. Folglich verursachen nachfolgende Versuche, unter einem Nicht-Root-Benutzer zu laufen **yum list**, keinen **YUM-Absturz** mehr.

(BZ#1642458)

### Die yum(8) Man Page erwähnt den yum module profile Befehl falsch

Auf der **yum(8)** Handbuchseite wird fälschlicherweise angegeben, dass das **YUM-Paketverwaltungswerkzeug** den **yum module profile** Befehl zum Bereitstellen von Details zu Modulprofilen enthält. Dieser Befehl ist jedoch nicht mehr verfügbar, und bei Verwendung zeigt **YUM** eine Fehlermeldung über einen ungültigen Befehl an. Für Details zu Modulprofilen verwenden Sie stattdessen den neuen **yum module info --profile** Befehl.

(BZ#1622580)

### yum-plugin-aliases derzeit nicht verfügbar

Das **yum-plugin-aliases** Paket, das den **alias** Befehl zum Hinzufügen benutzerdefinierter Yum-Aliase bereitstellt, ist derzeit nicht verfügbar. Daher ist es derzeit nicht möglich, Aliase zu verwenden.

(BZ#1647760)

### yum-plugin-changelog derzeit nicht verfügbar

Das **yum-plugin-changelog** Paket, das die Anzeige von Paketänderungsprotokollen vor und nach der Aktualisierung des Pakets ermöglicht, ist derzeit nicht verfügbar.

(BZ#1581191)

## 7.4. INFRASTRUKTURDIENSTLEISTUNGEN (MASCHINENÜBERSETZUNG)

### Tuned setzt die Kommandozeilenparameter für den Kernel-Boot nicht

Das **Tuned-Tool** unterstützt die Boot Loader Specification (BLS) nicht, die standardmäßig aktiviert ist. Folglich setzt **Tuned** bestimmte Kernel-Boot-Befehlszeilenparameter nicht, was einige Probleme verursacht, wie z.B. Leistungsabfall oder nicht isolierte CPU-Kerne. Um dieses Problem zu umgehen, deaktivieren Sie BLS und starten Sie **Tuned** neu.

1. Installiere das *grubby*-Paket.
2. Entfernen Sie die folgende Zeile aus der `/etc/default/grub` Datei:

```
GRUB_ENABLE_BLSCFG=true
```

3. Generieren Sie die **grub2.cfg** Datei erneut, indem Sie sie für Nicht-EFI-Systeme ausführen:

```
grub2-mkconfig -o /etc/grub2.cfg
```

oder für EFI-Systeme:

```
grub2-mkconfig -o /etc/grub2-efi.cfg
```

4. Neustart **Tuned** durch Ausführen:

```
systemctl Neustart abgestimmt
```

Infolgedessen setzt **Tuned** die Bootparameter des Kernels wie erwartet.

(BZ#1576435)

## 7.5. SHELLS UND BEFEHLSZEILEN-TOOLS (MASCHINENÜBERSETZUNG)

Python bindung des **net-snmp** Pakets ist nicht verfügbar



Die **Net-SNMP**Werkzeugsuite sieht keine Bindung vor **Python 3**, was die Standardimplementierung **Python** in RHEL 8 ist. Infolgedessen sind in RHEL 8 **python-net-snmp**, **python2-net-snmp**, oder **python3-net-snmp** Pakete nicht verfügbar.

(BZ#1584510)

## 7.6. DYNAMISCHE PROGRAMMIERSPRACHEN, WEB- UND DATENBANKSERVER (MASCHINENÜBERSETZUNG)

### Datenbankserver sind nicht parallel installierbar

Die **mariadb** und **mysql** Module können in RHEL 8.0 aufgrund widersprüchlicher RPM-Pakete nicht parallel installiert werden.

Konstruktiv ist es unmöglich, mehr als eine Version (Stream) desselben Moduls parallel zu installieren. Beispielsweise müssen Sie nur einen der verfügbaren Streams aus dem **postgresql** Modul auswählen, entweder (Standard **10**) oder **9.6**. Die parallele Installation von Komponenten ist in den Red Hat Software Collections für RHEL 6 und RHEL 7 möglich. In RHEL 8 können verschiedene Versionen von Datenbankservern in Containern verwendet werden.

(BZ#1566048)

### Probleme bei der **mod\_cgid** Protokollierung

Wenn das **mod\_cgid** Apache httpd-Modul unter einem threaded multi-processing module (MPM) verwendet wird, was die Standardsituation in RHEL 8 ist, treten folgende Protokollierungsprobleme auf:

- Die **stderr** Ausgabe des CGI-Skripts wird nicht mit Standardzeitstempelinformationen versehen.
- Die **stderr** Ausgabe des CGI-Skripts wird nicht korrekt in eine Protokolldatei umgeleitet, die spezifisch für das **VirtualHost**, wenn konfiguriert, ist.

(BZ#1633224)

### Das **IO::Socket::SSLPerl**-Modul unterstützt TLS 1.3 nicht

Neue Funktionen des TLS 1.3-Protokolls, wie z.B. Sitzungswiederaufnahme oder Post-Handshake-Authentifizierung, wurden in der RHEL 8-Bibliothek **OpenSSL**, aber nicht im **Net::SSLeayPerl**-Modul implementiert und sind daher im **IO::Socket::SSLPerl**-Modul nicht verfügbar. Infolgedessen kann die Authentifizierung von Client-Zertifikaten fehlschlagen und das Wiederherstellen von Sitzungen kann langsamer sein als mit dem TLS 1.2-Protokoll.

Um dieses Problem zu umgehen, deaktivieren Sie die Verwendung von TLS 1.3, indem Sie die **SSL\_version** Option beim Erstellen eines **IO::Socket::SSL** Objekts auf den **!TLSv1\_3** Wert setzen.

(BZ#1632600)

## 7.7. IDENTITÄTSMANAGEMENT (MASCHINENÜBERSETZUNG)

### Der KCM Credential Cache ist nicht für eine große Anzahl von Anmeldeinformationen in einem einzigen Credential Cache geeignet

Wenn der Credential-Cache zu viele Anmeldeinformationen enthält, schlagen Kerberos-Operationen, wie z.B. **kinit**, fehl, da der Puffer für den Datentransfer zwischen der **sssd-kcm-Komponente** und der zugrunde liegenden Datenbank fest programmiert ist.

Um dieses Problem zu umgehen, fügen Sie die **ccache\_storage = memory**Option im **kcm-Bereich** der **/etc/sss/sss.conf** Datei hinzu. Dadurch wird der **kcm-Responder** angewiesen, die Credential-Caches nur im Speicher zu speichern, nicht dauerhaft. Wenn Sie dies tun, werden durch einen Neustart des Systems oder **sss-kcm** die Credential-Caches gelöscht. Beachten Sie, dass KCM Cachegrößen von bis zu 64 kB verarbeiten kann.

(BZ#1448094)

### Widersprüchliche Timeout-Werte verhindern, dass sich SSSD mit Servern verbindet

Einige der Standard-Timeout-Werte, die sich auf die Failover-Operationen beziehen, die vom System Security Services Daemon (SSSD) verwendet werden, sind widersprüchlich. Folglich verhindert der Timeout-Wert, der für SSSD für die Kommunikation mit einem einzelnen Server reserviert ist, dass SSSD vor dem Verbindungsvorgang andere Server als Ganzes ausprobiert. Um das Problem zu umgehen, setzen Sie den Wert des **ldap\_opt\_timeout** Timeout-Parameters höher als den Wert des **dns\_resolver\_timeout** Parameters und den Wert des **dns\_resolver\_timeout** Parameters höher als den Wert des **dns\_resolver\_op\_timeout** Parameters.

(BZ#1382750)

### Die Verwendung einer Smartcard zur Anmeldung am IdM-Web-UI funktioniert nicht

Wenn ein Benutzer versucht, sich mit einem auf seiner Smartcard gespeicherten Zertifikat in die Web-Benutzeroberfläche des Identity Management (IdM) einzuloggen, verwendet der D-Bus-Schnittstellencode des System Security Services Daemon (SSSD) einen falschen Callback, um den Benutzer nachzuschlagen. Infolgedessen stürzt das Lookup ab. Um das Problem zu umgehen, verwenden Sie andere Methoden der Authentifizierung.

(BZ#1642508)

### IdM-Server funktioniert nicht in FIPS

Aufgrund einer unvollständigen Implementierung des SSL-Konnektors für Tomcat funktioniert ein IdM-Server mit installiertem Zertifikatsserver auf Maschinen mit aktiviertem FIPS-Modus nicht.

(BZ#1673296)

### Der nuxwdog Dienst schlägt in HSM-Umgebungen fehl und erfordert die Installation des keyutils Pakets in Nicht-HSM-Umgebungen

Der **nuxwdog** Watchdog-Service wurde in das Zertifikatssystem integriert. Infolgedessen **nuxwdog** wird es nicht mehr als separates Paket angeboten. Um den Watchdog-Dienst zu nutzen, installieren Sie das **pki-server** Paket.

Beachten Sie, dass der **nuxwdog** Dienst folgende bekannte Probleme hat:

- Der **nuxwdog** Dienst funktioniert nicht, wenn Sie ein Hardware-Speichermodul (HSM) verwenden. Für dieses Problem ist kein Workaround verfügbar.
- In einer Nicht-HSM-Umgebung installiert Red Hat Enterprise Linux 8.0 das **keyutils** Paket nicht automatisch als Abhängigkeit. Um das Paket manuell zu installieren, verwenden Sie den **dnf install keyutils** Befehl.

(BZ#1652269)

## 7.8. COMPILER UND ENTWICKLUNGSWERKZEUGE (MASCHINENÜBERSETZUNG)

## Synthetische Funktionen, die von GCC generiert werden, verwirren SystemTap

Die GCC-Optimierung kann synthetische Funktionen für teilweise inlinierte Kopien anderer Funktionen erzeugen. Tools wie SystemTap und GDB können diese synthetischen Funktionen nicht von realen Funktionen unterscheiden. Infolgedessen kann SystemTap Sonden sowohl an synthetischen als auch an realen Funktionseintrittspunkten platzieren und so mehrere Sondentreffer für einen einzigen realen Funktionsaufruf registrieren.

Um dieses Problem zu umgehen, müssen SystemTap-Skripte mit Maßnahmen wie dem Erkennen von Rekursionen und dem Unterdrücken von Sonden, die sich auf inlinierte Teilfunktionen beziehen, angepasst werden. Zum Beispiel ein Skript

```
probe kernel.function("can_nice").call { }
```

kann versuchen, das beschriebene Problem wie folgt zu vermeiden:

```
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()] --;
}
```

Beachten Sie, dass dieses Beispielskript nicht alle möglichen Szenarien berücksichtigt, wie z.B. verpasste Kprobes oder Kretprobes oder wirklich beabsichtigte Rekursionen.

(BZ#1169184)

## Das ItraceTool meldet keine Funktionsaufrufe

Aufgrund der Verbesserungen der binären Härtung, die auf alle RHEL-Komponenten angewendet werden, kann das **Itrace**Tool Funktionsaufrufe in Binärdateien von RHEL-Komponenten nicht mehr erkennen. Infolgedessen ist die **Itrace**Ausgabe leer, da sie bei Verwendung in solchen Binärdateien keine erkannten Aufrufe meldet. Es ist derzeit kein Workaround verfügbar.

Als Hinweis **Itrace**kann man Aufrufe in benutzerdefinierten Binärdateien, die ohne die entsprechenden Härtungsflags erstellt wurden, korrekt melden.

(BZ#1618748, BZ#1655368)

## 7.9. DATEISYSTEME UND SPEICHER (MASCHINENÜBERSETZUNG)

### Es ist nicht möglich, ein iSCSI-Ziel mithilfe des **iscsiuio**Pakets zu erkennen

Red Hat Enterprise Linux 8 erlaubt keinen gleichzeitigen Zugriff auf PCI-Registerbereiche. In der Folge wurde ein **could not set host net params (err 29)**Fehler gesetzt und die Verbindung zum Discovery-Portal fehlgeschlagen. Um dieses Problem zu umgehen, setzen Sie den Kernelparameter **iomem=relaxed**in der Kernel-Befehlszeile für das iSCSI-Offload. Dabei handelt es sich insbesondere um eine Entladung mit dem **bnx2i**Fahrer. Dadurch ist die Verbindung zum Discovery-Portal nun erfolgreich und das **iscsiuio**Paket funktioniert nun korrekt.

(BZ#1626629)

## Die XFS DAX-Mount Option ist nicht kompatibel mit gemeinsamen Copy-on-Write Datenerweiterungen

Ein XFS-Dateisystem, das mit der Funktion zum gemeinsamen Kopieren beim Schreiben von Datenerweiterungen formatiert ist, ist mit der **-o dax**Mount-Option nicht kompatibel. Infolgedessen ist das Mounten eines solchen Dateisystems mit **-o dax**Fehlern verbunden.

Um das Problem zu umgehen, formatieren Sie das Dateisystem mit der Option **reflink=0**Metadaten, um freigegebene Copy-on-Write Datenerweiterungen zu deaktivieren:

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

Daher **-o dax**ist das Mounten des Dateisystems erfolgreich.

(BZ#1620330)

## 7.10. VERNETZUNG (MASCHINENÜBERSETZUNG)

### nftables unterstützt keine mehrdimensionalen IP-Settypen

Das **nftables**Paketfilter-Framework unterstützt keine Mengentypen mit Verkettungen und Intervallen. Daher können Sie keine multidimensionalen IP-Settypen verwenden, wie z.B. **hash:net,port**, mit **nftables**.

Um dieses Problem zu umgehen, verwenden Sie das **iptables**Framework mit dem **ipset**Tool, wenn Sie mehrdimensionale IP-Settypen benötigen.

(BZ#1593711)

## 7.11. SICHERHEIT (MASCHINENÜBERSETZUNG)

### OpenSCAPrpmverifypackage funktioniert nicht richtig

Die **chdir**und **chroot**Systemaufrufe werden von der **rpmverifypackage**Sonde zweimal aufgerufen. Folglich tritt ein Fehler auf, wenn die Sonde während eines **OpenSCAP-Scans** mit benutzerdefiniertem Inhalt der Open Vulnerability and Assessment Language (OVAL) verwendet wird.

Um dieses Problem zu umgehen, verwenden Sie den **rpmverifypackage\_test**OVAL-Test nicht in Ihren Inhalten oder verwenden Sie nur die Inhalte aus dem **scap-security-guide**Paket, in dem **rpmverifypackage\_test**sie nicht verwendet werden.

(BZ#1646197)

### libssh nicht mit der systemweiten Krypto-Richtlinie übereinstimmt

Die **libssh**Bibliothek folgt nicht den systemweiten kryptographischen Richtlinieneinstellungen. Infolgedessen wird der Satz der unterstützten Algorithmen nicht geändert, wenn der Administrator die Ebene der Krypto-Richtlinien mit dem **update-crypto-policies**Befehl ändert.

Um dieses Problem zu umgehen, muss der Satz der beworbenen Algorithmen von jeder Anwendung, die diese verwendet **libssh**, individuell eingestellt werden. Wenn das System auf die Richtlinienebene LEGACY oder FUTURE eingestellt ist, **libssh**verhalten sich Anwendungen, die verwenden, daher inkonsistent im Vergleich zu **OpenSSH**.

(BZ#1646563)

### SCAP Workbench kann keine ergebnisbasierten Korrekturen aus maßgeschneiderten Profilen generieren

Der folgende Fehler tritt auf, wenn versucht wird, mit dem Werkzeug **SCAP Workbench** aus einem benutzerdefinierten Profil ergebnisbasierte Korrekturrollen zu generieren:

```
Error generating remediation role ../remediation.sh: Exit code of oscap was 1: [output truncated]
```

Um dieses Problem zu umgehen, verwenden Sie den **oscap**Befehl mit der **--tailoring-file**Option.

(BZ#1640715)

### OpenSCAPrpmverifyfile funktioniert nicht

Der **OpenSCAP-Scanner** ändert das aktuelle Arbeitsverzeichnis nicht korrekt im Offlinemodus, und die **fchdir**Funktion wird nicht mit den richtigen Argumenten in der **OpenSCAP-Sonderrpmverifyfile** aufgerufen. Folglich schlägt das Scannen beliebiger Dateisysteme mit dem **oscap-chroot**Befehl fehl, wenn **rpmverifyfile\_testes** in einem SCAP-Inhalt verwendet wird. Infolgedessen wird das beschriebene Szenario **oscap-chroot**abgebrochen.

(BZ#1636431)

### Ein Dienstprogramm für die Sicherheits- und Konformitätsprüfung von Containern ist nicht verfügbar

In Red Hat Enterprise Linux 7 kann das **oscap-docker**Dienstprogramm zum Scannen von Docker-Containern auf Basis von Atomic-Technologien verwendet werden. In Red Hat Enterprise Linux 8 sind die Docker- und Atom-bezogenen **OpenSCAP-Befehle** nicht verfügbar. Infolgedessen ist in RHEL 8 derzeit kein gleichwertiges Dienstprogramm für die Sicherheits- und Konformitätsprüfung von Containern verfügbar **oscap-docker**.

(BZ#1642373)

## KAPITEL 8. BEMERKENSWERTE ÄNDERUNGEN AN CONTAINERN (MASCHINENÜBERSETZUNG)

Eine Reihe von Container-Images ist für Red Hat Enterprise Linux (RHEL) 8.0 verfügbar. Zu den bemerkenswerten Änderungen gehören:

- Docker ist in RHEL 8.0 nicht enthalten. Verwenden Sie für die Arbeit mit Containern die Tools **podman**, **buildah**, **skopeo** und **runc**. Informationen zu diesen Werkzeugen und zur Verwendung von Containern in RHEL 8 finden Sie unter [Building, running, and managing containers](#).
- Das **Podman-Tool** wurde als voll unterstütztes Feature veröffentlicht. Das **Podman-Tool** verwaltet Pods, Container-Images und Container auf einem einzigen Knoten. Es basiert auf der **libpod-Bibliothek**, die die Verwaltung von Containern und Containergruppen, den sogenannten Pods, ermöglicht.

Um zu lernen, wie man **podman** verwendet, siehe [Building, running, and managing containers](#).

- In RHEL 8 GA sind Red Hat Universal Base Images (UBI) neu verfügbar. UBIs ersetzen einige der zuvor von Red Hat bereitgestellten Bilder, wie beispielsweise die Standard- und die minimalen RHEL-Basisbilder. Im Gegensatz zu älteren Red Hat-Bildern sind UBIs frei weiterverteilbar. Das bedeutet, dass sie in jeder Umgebung verwendet und überall gemeinsam genutzt werden können. Sie können sie auch verwenden, wenn Sie kein Red Hat-Kunde sind.

Zur UBI-Dokumentation siehe [Building, running, and managing containers](#).

- In RHEL 8 GA sind zusätzliche Container-Images verfügbar, die AppStream-Komponenten bereitstellen, für die Container-Images mit **Red Hat Software Collections** in RHEL 7 verteilt werden. Alle diese RHEL 8 Bilder basieren auf dem **ubi8** Basisbild.
- Container-Images ARM für die 64-Bit-ARM-Architektur werden in RHEL 8 vollständig unterstützt.
- Der **rhel-tools** Behälter wurde in RHEL 8 entfernt. Die **sos** und die **redhat-support-tool** Werkzeuge sind im **support-tools** Container enthalten. Systemadministratoren können dieses Image auch als Grundlage für den Aufbau eines Container-Images für Systemtools verwenden.
- Die Unterstützung für wurzellose Behälter ist als Technologievorschau in RHEL 8 verfügbar. Rootless-Container sind Container, die von regulären Systembenutzern ohne Administratorrechte erstellt und verwaltet werden.

## ANHANG A. LISTE DER TICKETS NACH KOMPONENTEN

Komponente	Tickets
<b>389-ds-base</b>	BZ#1334254, BZ#1358706, <a href="#">BZ#1693159</a>
<b>NetworkManager</b>	BZ#1555013, BZ#1555012, BZ#1557035
<b>PackageKit</b>	BZ#1559414
<b>anaconda</b>	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904
<b>audit</b>	BZ#1616428
<b>bcc</b>	BZ#1548302
<b>bind</b>	BZ#1588592
<b>boom-boot</b>	BZ#1649582
<b>boost</b>	BZ#1494495, BZ#1616244
<b>cloud-init</b>	BZ#1615599
<b>cmake</b>	BZ#1590139
<b>cockpit</b>	BZ#1619993
<b>crypto-policies</b>	BZ#1591620
<b>cryptsetup</b>	BZ#1564540
<b>device-mapper-multipath</b>	BZ#1643550
<b>distribution</b>	BZ#1566048, BZ#1516741, BZ#1516728
<b>dnf</b>	BZ#1622580, BZ#1647760, BZ#1581191
<b>esc</b>	BZ#1538645
<b>firewalld</b>	BZ#1509026
<b>gcc</b>	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1246444
<b>gdm</b>	BZ#1589678

Komponente	Tickets
<b>glibc</b>	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
<b>go-toolset</b>	BZ#1512570
<b>httpd</b>	BZ#1633224, BZ#1632754
<b>iproute</b>	BZ#1640991
<b>iptables</b>	BZ#1644030, BZ#1564596
<b>iscsi-initiator-utils</b>	BZ#1626629, BZ#1582099
<b>kernel-rt</b>	BZ#1592977
<b>kernel</b>	BZ#1598448, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1485229, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1273139, BZ#1559607, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1664838, BZ#1596240, BZ#1534870, BZ#1501618, BZ#1153521, BZ#1494705, BZ#1620330, BZ#1505999
<b>kexec-tools</b>	BZ#1520209
<b>kmod-kvdo</b>	BZ#1534087, BZ#1639512
<b>libdnf</b>	BZ#1642458
<b>libreswan</b>	<a href="#">BZ#1657854</a>
<b>libssh</b>	BZ#1485241
<b>ltrace</b>	BZ#1618748, BZ#1584322
<b>lvm2</b>	BZ#1643553, BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
<b>mariadb</b>	BZ#1637034
<b>net-snmp</b>	BZ#1584510
<b>nfs-utils</b>	BZ#1592011, BZ#1639432
<b>nftables</b>	BZ#1593711



Komponente	Tickets
<b>nginx</b>	BZ#1545526
<b>nodejs-10-module</b>	BZ#1622118
<b>nss</b>	BZ#1645153
<b>nuxwdog</b>	<a href="#">BZ#1652269</a>
<b>openldap</b>	BZ#1570056
<b>opensc</b>	BZ#1595626
<b>openscap</b>	BZ#1646197, BZ#1636431, BZ#1642373, BZ#1618484, BZ#1614273, BZ#1618464
<b>openssh</b>	BZ#1622511, BZ#1228088
<b>pacemaker</b>	<a href="#">BZ#1543494</a>
<b>pcs</b>	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
<b>perl-IO-Socket-SSL</b>	BZ#1632600
<b>perl</b>	BZ#1511131
<b>pki-core</b>	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, <a href="#">BZ#1669257</a> , BZ#1656856, <a href="#">BZ#1673296</a>
<b>pykickstart</b>	BZ#1637872, BZ#1612061
<b>qemu-kvm</b>	BZ#1508139, BZ#1559240, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1519004
<b>redhat-release</b>	BZ#1636338
<b>rsyslog</b>	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645
<b>scap-security-guide</b>	BZ#1618528, BZ#1618518
<b>scap-workbench</b>	BZ#1640715
<b>selinux-policy</b>	BZ#1592244, BZ#1594111, BZ#1549772, BZ#1626446

Komponente	Tickets
<b>setup</b>	BZ#1591969
<b>sos</b>	BZ#1559836
<b>sssd</b>	BZ#1448094, BZ#1382750, BZ#1642508, BZ#1620123
<b>subversion</b>	BZ#1571415
<b>swig-3.0-module</b>	BZ#1660051
<b>tomcatjss</b>	BZ#1424966, BZ#1636564
<b>tuned</b>	BZ#1576435, BZ#1565598
<b>valgrind</b>	BZ#1500481, BZ#1538009
<b>virt-manager</b>	BZ#1599777, BZ#1643609
<b>wpa_supplicant</b>	BZ#1582538
andere	BZ#1646563, BZ#1640697, BZ#1623712, BZ#1649404, BZ#1581198, BZ#1581990, BZ#1649497, BZ#1643294, BZ#1647612, BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1647110, BZ#1641007, BZ#1641029, BZ#1641022, BZ#1649493, BZ#1649922, BZ#1559616, BZ#1646541, BZ#1647725, <a href="#">BZ#1686057</a> , BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, <a href="#">BZ#1650701</a> , JIRA:RHELPLAN-10347, 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-1212, JIRA:RHELPLAN-1473, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-1499, 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-12764, JIRA:RHELPLAN-14607, BZ#1641014, BZ#1692746, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#1649891