



Red Hat Enterprise Linux 8

8.0 릴리스 노트

Red Hat Enterprise Linux 8.0 릴리스 노트

Red Hat Enterprise Linux 8 8.0 릴리스 노트

Red Hat Enterprise Linux 8.0 릴리스 노트

법적 공지

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

본 릴리스 노트에서는 Red Hat Enterprise Linux 8.0에서 구현된 개선사항 및 추가 사항에 관한 대략적인 설명과 이 릴리스의 알려진 문제점, 중요한 버그 수정, 기술 프리뷰, 사용되지 않는 기능, 제거된 기능 등의 세부 사항에 대해 설명합니다.

차례

RED HAT 문서에 관한 피드백 제공	4
1장. 개요	5
콘텐츠 배포	5
소프트웨어 관리	5
웹 서버, 데이터베이스, 동적 언어	5
데스크탑	5
설치 프로그램 및 이미지 생성	5
파일 시스템 및 스토리지	5
보안	6
네트워크 구성	6
가상화	6
컴파일러 및 개발 도구	6
고가용성 및 클러스터	7
관련 자료	7
2장. 아키텍처	8
3장. RHEL 8의 콘텐츠 배포	9
3.1. 설치	9
3.2. 리포지토리	9
3.3. APPLICATION STREAMS	9
4장. 새로운 기능	11
4.1. 웹 콘솔	11
4.2. 설치 프로그램 및 이미지 생성	12
4.3. 커널	13
4.4. 소프트웨어 관리	14
4.5. 인프라 서비스	16
4.6. 셸 및 명령행 툴	17
4.7. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버	18
4.8. 데스크탑	23
4.9. IDM (IDENTITY MANAGEMENT)	24
4.10. 컴파일러 및 개발 도구	28
4.11. 파일 시스템 및 스토리지	36
4.12. 고가용성 및 클러스터	40
4.13. 네트워크 구성	42
4.14. 보안	46
4.15. 가상화	51
4.16. 지원 관련 기능	53
5장. 기술 프리뷰	54
5.1. 커널	54
5.2. 파일 시스템 및 스토리지	55
5.3. 고가용성 및 클러스터	55
5.4. 보안	56
5.5. 가상화	56
6장. 사용되지 않는 기능	57
6.1. 설치 프로그램 및 이미지 생성	57
6.2. 파일 시스템 및 스토리지	57
6.3. 네트워크 구성	57
6.4. 보안	58

6.5. 가상화	58
7장. 알려진 문제	60
7.1. 설치 프로그램 및 이미지 생성	60
7.2. 커널	60
7.3. 소프트웨어 관리	61
7.4. 인프라 서비스	62
7.5. 셸 및 명령행 툴	62
7.6. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버	62
7.7. IDM (IDENTITY MANAGEMENT)	63
7.8. 컴파일러 및 개발 도구	64
7.9. 파일 시스템 및 스토리지	65
7.10. 네트워크 구성	65
7.11. 보안	65
8장. 컨테이너 관련 주요 변경 사항	67
부록 A. 구성 요소별 티켓 목록	68

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. 다음과 같은 방식에 따라 문서를 개선할 수 있는 방법에 대해 알려주십시오.

- 특정 문구에 대해 간단한 의견을 남기려면 문서가 Multi-page HTML 형식으로 되어 있는지 확인하십시오. 댓글을 추가하려는 텍스트 부분을 강조 표시하고 강조 표시된 텍스트 아래에 나타나는 **피드백 추가 (Add Feedback)** 팝업을 클릭하여 표시된 지침을 따르십시오.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
 1. [Bugzilla](#) 웹 사이트로 이동하십시오.
 2. Component로 **Documentation**을 선택하십시오.
 3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **Submit Bug**를 클릭하십시오.

1장. 개요

Red Hat Enterprise Linux 8.0에서는 Fedora 28 및 업스트림 커널 4.18을 기반으로 모든 수준의 워크로드를 지원하는 데 필요한 툴을 사용하여 하이브리드 클라우드 및 데이터 센터 배포 전반에 걸쳐 안정적이고 안전하며 일관된 기반을 제공합니다. 이 릴리스의 주요 내용은 다음과 같습니다.

콘텐츠 배포

- 콘텐츠는 **BaseOS** 및 Application Stream (**AppStream**) 리포지토리를 통해 사용할 수 있습니다.
- **AppStream** 리포지토리에서는 기존 RPM 형식의 새로운 확장 형식(**모듈**)을 지원합니다. 따라서 구성 요소의 주요 버전을 여러개 설치할 수 있습니다.

자세한 내용은 [3장. RHEL 8의 콘텐츠 배포](#)를 참조하십시오.

소프트웨어 관리

- 현재 **YUM** 패키지 관리자는 **DNF** 기술을 기반으로 하며, 모듈식 콘텐츠 지원, 성능 향상 및 튜링과의 통합을 위해 안정적이고 우수한 API를 제공합니다.

자세한 내용은 [4.4절. "소프트웨어 관리"](#)를 참조하십시오.

웹 서버, 데이터베이스, 동적 언어

- **Python 3.6**은 RHEL 8의 기본 Python 구현입니다. Python 2.7에 대한 지원은 제한됩니다. 기본적으로 Python 버전이 설치되어 있지 않습니다.
- RHEL 8에서는 MariaDB 10.3, MySQL 8.0, PostgreSQL 10, PostgreSQL 9.6 및 Redis 4.0과 같은 **데이터베이스 서버**를 제공합니다.

자세한 내용은 [4.7절. "동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버"](#)를 참조하십시오.

데스크탑

- **GNOME Shell**은 3.28 버전으로 업데이트되었습니다.
- GNOME 세션과 GNOME Display Manager에서는 **Wayland**를 기본 디스플레이 서버로 사용합니다. RHEL 7의 기본 디스플레이 서버인 **X.Org** 서버도 사용 가능합니다.

자세한 내용은 [4.8절. "데스크탑"](#)을 참조하십시오.

설치 프로그램 및 이미지 생성

- **Anaconda** 설치 프로그램에서는 **LUKS2** 디스크 암호화를 사용하여 **NVDIMM** 장치에 시스템을 설치할 수 있습니다.
- 새로운 **Composer** 툴을 사용하면 다양한 공급자의 클라우드에 배포 용으로 준비가 된 이미지를 포함하여 다양한 형식의 사용자 정의 시스템 이미지를 생성할 수 있습니다. **Composer**는 **기술 프리뷰**로 사용할 수 있습니다.
- **IBM Z**는 하드웨어 관리 콘솔(**HMC**) 및 지원 요소(**SE**)를 사용하여 DVD에서 설치합니다.

자세한 내용은 [4.2절. "설치 프로그램 및 이미지 생성"](#)을 참조하십시오.

파일 시스템 및 스토리지

- **Stratis** 로컬 스토리지 관리자가 도입되었습니다. **Stratis**는 통합 인터페이스를 사용하여 복잡한 스토리지 작업을 쉽게 수행하고 스토리지 스택을 관리할 수 있습니다.

- 기존의 LUKS(LUKS1) 형식은 LUKS 버전 2(LUKS2) 형식으로 교체되었습니다. **dm-crypt** 하위 시스템과 **cryptsetup** 툴에서는 이제 LUKS2를 암호화된 볼륨의 기본 형식으로 사용합니다.

자세한 내용은 [4.11절. "파일 시스템 및 스토리지"](#)를 참조하십시오.

보안

- TLS, IPSec, SSH, DNSSec, Kerberos 프로토콜을 대상으로 시스템 전체의 **암호화 정책**이 기본적으로 적용되어 코어 암호화 하위 시스템에 정책을 설정합니다. 관리자가 새로운 **update-cryptopolicies** 명령을 사용하여 default, legacy, future, fips와 같은 모드 간에 쉽게 전환할 수 있습니다.
- **PKCS #11**을 사용하여 스마트 카드 및 **HSM**(Hardware Security Modules)은 이제 시스템 전체에서 일관되게 지원됩니다.

자세한 내용은 [4.14절. "보안"](#)을 참조하십시오.

네트워크 구성

- **nftables** 프레임워크가 기본 네트워크 패킷 필터링 기능의 역할에서 **iptables**를 대체하여 사용됩니다.
- 이제 **firewalld** 데몬에서는 **nftables**를 기본 백엔드로 사용합니다.
- 여러 컨테이너의 네트워크를 연결하는 데 사용할 수 있는 **IPVLAN** 가상 네트워크 드라이버를 지원합니다.

자세한 내용은 [4.13절. "네트워크 구성"](#)을 참조하십시오.

가상화

- RHEL 8에서 생성한 가상 머신에서 최신 PCI Express 기반 머신 유형(**Q35**)이 지원되며 자동으로 설정됩니다. 이로 인해 가상 장치의 기능 및 호환성이 향상되었습니다.
- **Cockpit**이라고도 하는 RHEL 8 웹 콘솔을 사용하여 가상 머신을 생성하고 관리할 수 있습니다.
- **QEMU** 에뮬레이터에서는 QEMU에서 수행할 수 있는 시스템 호출의 구성 가능 제한사항을 제공하는 **샌드박스** 기술이 도입되어 가상 머신의 보안이 강화되었습니다.

자세한 내용은 [4.15절. "가상화"](#)를 참조하십시오.

컴파일러 및 개발 도구

- 버전 8.2를 기반으로 하는 **GCC** 컴파일러는 최신 C++ 언어 표준 버전, 향상된 최적화, 새로운 코드 강화 기술, 개선된 경고 및 새로운 하드웨어 기능을 지원합니다.
- 코드 생성, 운영 및 디버깅을 위한 다양한 도구가 이제 **DWARF5** 디버깅 정보 형식을 실험적으로 처리할 수 있습니다.
- **eBPF** 추적에 대한 커널 지원은 **BCC**, **PCP** 및 **SystemTap**과 같은 일부 툴에서 사용할 수 있습니다.
- 버전 2.28을 기반으로 하는 **glibc** 라이브러리에서는 Unicode 11 지원, 최신 Linux 시스템 호출, DNS stub 해결에서 주요 개선사항, 추가 보안 강화 및 성능 향상을 지원합니다.

자세한 내용은 [4.10절. "컴파일러 및 개발 도구"](#)을 참조하십시오.

고가용성 및 클러스터

- **Pacemaker** 클러스터 리소스 관리자가 업스트림 버전 2.0.0으로 업그레이드되어, 여러 버그 수정 및 개선 사항이 추가되었습니다.
- RHEL 8의 **pcs** 설정 시스템에서는 Corosync 3, **knet** 및 노드 이름을 완전 지원합니다.

자세한 내용은 [4.12절. "고가용성 및 클러스터"](#)를 참조하십시오.

관련 자료

- Red Hat Enterprise Linux 8.0의 기능 및 제한사항과 다른 시스템 버전과의 비교 내용은 지식베이스 문서 [Red Hat Enterprise Linux 기술의 기능 및 제한사항](#)에서 참조하십시오.
- Red Hat Enterprise Linux 라이프사이클에 관한 정보는 [Red Hat Enterprise Linux 라이프사이클](#)에서 참조하십시오.
- RHEL 8의 패키지 목록은 [패키지 매니페스트](#)에서 참조하십시오.

2장. 아키텍처

Red Hat Enterprise Linux 8.0은 다음과 같은 아키텍처를 기반으로 하는 커널 버전 4.18과 함께 배포됩니다.

- AMD 및 Intel 64비트 아키텍처
- 64비트 ARM 아키텍처
- IBM Power Systems, Little Endian
- IBM Z

3장. RHEL 8의 콘텐츠 배포

3.1. 설치

Red Hat Enterprise Linux 8은 ISO 이미지를 사용하여 설치합니다. AMD64, Intel 64비트, 64비트 ARM, IBM Power Systems, IBM Z 아키텍처에서는 두 가지 유형의 ISO 이미지를 사용할 수 있습니다.

- 바이너리 DVD ISO: BaseOS 및 AppStream 리포지토리가 포함된 전체 설치 이미지로, 추가 리포지토리 없이 설치를 완료할 수 있습니다.
- 부트 ISO: 설치 프로그램으로 부팅하는 데 사용하는 최소 부트 ISO 이미지입니다. 이 옵션을 사용하여 소프트웨어 패키지를 설치하려면 BaseOS와 AppStream 리포지토리에 액세스해야 합니다. 리포지토리는 바이너리 DVD ISO 이미지의 일부입니다.

ISO 이미지 다운로드, 설치 미디어 생성 및 RHEL 설치 완료 방법은 [표준 RHEL 설치](#) 문서를 참조하십시오. 자동화된 Kickstart 설치 및 기타 고급 주제는 [고급 RHEL 설치](#) 문서를 참조하십시오.

3.2. 리포지토리

Red Hat Enterprise Linux 8은 두 개의 리포지토리를 통해 배포됩니다.

- BaseOS
- AppStream

두 리포지토리 모두 기본 RHEL 설치에 필요하며 모든 RHEL 서브스크립션을 통해 사용할 수 있습니다.

BaseOS 리포지토리의 콘텐츠는 모든 설치의 기반이 되는 기본 OS 기능의 코어 세트를 제공하는 데 사용됩니다. 이 콘텐츠는 RPM 형식으로 사용 가능하며 이전 RHEL 릴리스와 비슷한 지원 조건이 적용됩니다. BaseOS를 통해 배포되는 패키지 목록은 [패키지 매니페스트](#)를 참조하십시오.

Application Stream 리포지토리의 콘텐츠에는 다양한 워크로드와 사용 사례를 지원하는 추가 사용자 공간 애플리케이션, 런타임 언어 및 데이터베이스가 포함되어 있습니다. AppStream의 콘텐츠는 기존의 RPM 형식 및 RPM 형식의 확장 (모듈이라고 함) 형식 중 하나로 사용 가능합니다. AppStream에서 사용 가능한 패키지 목록은 [패키지 매니페스트](#)를 참조하십시오.

또한 CodeReady Linux Builder 리포지토리는 모든 RHEL 서브스크립션을 통해 사용할 수 있습니다. 이는 개발자가 사용할 수 있는 추가 패키지를 제공합니다. CodeReady Linux Builder 리포지토리에 포함된 패키지는 실제 프로덕션용으로 지원되지 않습니다.

RHEL 8 리포지토리에 관한 자세한 내용은 [패키지 매니페스트](#)를 참조하십시오.

3.3. APPLICATION STREAMS

Red Hat Enterprise Linux 8.0에는 Application Streams 개념이 도입되었습니다. 이제 여러 버전의 사용자 공간 구성 요소가 코어 운영 체제 패키지보다 자주 업데이트됩니다. 따라서 플랫폼 또는 특정 배포의 기본 안정성에 영향을 주지 않고 Red Hat Enterprise Linux를 사용자 정의할 수 있는 유연성이 향상됩니다.

Application Stream으로 사용 가능한 구성 요소는 모듈 또는 RPM 패키지로 패키징할 수 있으며 RHEL 8의 AppStream 리포지토리를 통해 제공합니다. 각 Application Stream 구성 요소에는 지정된 라이프사이클이 있습니다. 자세한 내용은 [Red Hat Enterprise Linux 라이프사이클](#) 을 참조하십시오.

모듈은 논리 단위, 애플리케이션, 언어 스택, 데이터베이스 또는 툴 세트를 나타내는 패키지 컬렉션입니다. 이러한 패키지는 함께 빌드, 테스트, 릴리스됩니다.

모듈 스트림은 Application Stream 구성 요소의 버전을 나타냅니다. 예를 들어 postgresql 모듈은 두 개의 스트림(버전)의 PostgreSQL 데이터베이스 서버, 즉 PostgreSQL 10(기본 스트림)과 PostgreSQL 9.6를 사용할 수 있습니다. 시스템에는 하나의 모듈 스트림만 설치할 수 있습니다. 개별 컨테이너에서 서로 다른 버전을 사용할 수 있습니다.

자세한 모듈 명령은 [Application Stream 사용](#)에서 참조하십시오. AppStream에서 사용 가능한 모듈 목록은 [패키지 매니페스트](#)를 참조하십시오.

4장. 새로운 기능

이 부분에서는 Red Hat Enterprise Linux 8에 도입된 새로운 기능 및 주요 개선 사항에 대해 설명합니다.

4.1. 웹 콘솔



참고

웹 콘솔의 서브스크립션 페이지는 이제 새로운 **subscription-manager-cockpit** 패키지에서 제공됩니다.

웹 콘솔에 방화벽 인터페이스가 추가

RHEL 8 웹 콘솔의 **네트워크** 페이지에는 이제 **방화벽** 섹션이 포함되어 있습니다. 이 섹션에서는 방화벽 사용 설정 또는 사용 해제는 물론 방화벽 규칙 추가, 제거, 수정할 수 있습니다.

(BZ#1647110)

웹 콘솔을 기본적으로 사용 가능

Cockpit이라고도 하는 RHEL 8 웹 콘솔용 패키지는 이제 Red Hat Enterprise Linux 기본 리포지토리의 일부이므로 등록된 RHEL 8 시스템에 즉시 설치할 수 있습니다.

또한 최소 설치 아닌 설치에서는 RHEL 8에 웹 콘솔이 자동으로 설치되고 콘솔에 필요한 방화벽 포트가 자동으로 열립니다. 로그인하기 전에 웹 콘솔을 사용으로 설정하거나 액세스하는 방법에 관한 정보를 제공하는 시스템 메시지도 추가되었습니다.

(JIRA:RHELPLAN-10355)

웹 콘솔 용 IdM 통합 개선

시스템이 IdM(Identity Management) 도메인에 등록된 경우 이제 RHEL 8 웹 콘솔에서 기본적으로 도메인의 중앙 관리 IdM 리소스를 사용합니다. 여기에는 다음과 같은 이점이 있습니다.

- IdM 도메인 관리자가 웹 콘솔을 사용하여 로컬 시스템을 관리할 수 있습니다.
- 콘솔의 웹 서버는 IdM 인증 기관 (CA)에서 발급하여 브라우저에서 승인하는 인증서로 자동 전환합니다.
- IdM 도메인의 Kerberos 티켓이 있는 사용자는 웹 콘솔에 액세스하기 위해 로그인 인증 정보를 제공하지 않아도 됩니다.
- IdM 도메인에서 인식되는 SSH 호스트는 수동으로 SSH 연결을 추가하지 않아도 웹 콘솔에 액세스할 수 있습니다.

웹 콘솔과 IdM 통합이 제대로 작동하려면 먼저 IdM 마스터 시스템에서 **enable-admins-sudo** 옵션을 사용하여 **ipa-adviser** 유틸리티를 실행해야 합니다.

(JIRA:RHELPLAN-3010)

웹 콘솔은 모바일 브라우저와 호환 가능

이번 업데이트 통해 모바일 브라우저 변형에서 웹 콘솔 메뉴 및 페이지를 탐색할 수 있습니다. 따라서 모바일 장치에서 RHEL 8 웹 콘솔을 사용하여 시스템을 관리할 수 있습니다.

(JIRA:RHELPLAN-10352)

웹 콘솔의 첫 페이지에 누락된 업데이트 및 서브스크립션 표시

RHEL 8 웹 콘솔에서 관리하는 시스템의 패키지가 오래되었거나 서브스크립션이 만료된 경우 시스템의 웹 콘솔의 첫 페이지에 경고가 표시됩니다.

(JIRA:RHELPLAN-10353)

웹 콘솔에서 PBD 등록 지원

이번 업데이트를 통해 RHEL 8 웹 콘솔 인터페이스를 사용하여 관리되는 시스템의 디스크에 PBD(Policy-Based Decryption) 규칙을 적용할 수 있습니다. Clevis 암호 해독 클라이언트를 사용하여 LUKS 암호화 디스크 파티션의 자동 잠금 해제와 같은 웹 콘솔의 다양한 보안 관리 기능을 용이하게 합니다.

(JIRA:RHELPLAN-10354)

웹 콘솔을 사용하여 가상 머신을 관리 가능

이제 가상 머신 페이지를 RHEL 8 웹 콘솔 인터페이스에 추가할 수 있으므로, 사용자가 libvirt 기반 가상 머신을 생성하고 관리할 수 있습니다.

(JIRA:RHELPLAN-2896)

4.2. 설치 프로그램 및 이미지 생성

IBM Z에서 SE 및 HMC를 사용하여 DVD에서 RHEL을 설치하는 기능을 완전 지원

IBM Z 하드웨어에서 SE(Support Element) 및 HMC(Hardware Management Console)를 사용하여 DVD에서 Red Hat Enterprise Linux 8을 설치하는 기능은 이제 완전하게 지원됩니다. 이와 같은 기능이 추가되어 SE와 HMC를 사용하여 IBM Z에 설치하는 프로세스가 간소화되었습니다.

바이너리 DVD에서 부팅할 때, 설치 프로그램에서 사용자에게 추가 커널 매개 변수를 입력하라는 메시지가 표시됩니다. DVD를 설치 소스로 설정하려면 `inst.repo=hmc`를 커널 매개 변수에 추가하십시오. 그러면 설치 프로그램을 사용하여 SE 및 HMC 파일에 액세스할 수 있고, DVD에서 stage2 이미지를 가져와서 소프트웨어 선택을 위해 DVD에서 패키지에 액세스할 수 있게 됩니다.

새 기능을 사용하면 외부 네트워크 설정 요구사항이 제거되며 설치 옵션이 확장됩니다.

(BZ#1500792)

설치 프로그램에서 LUKS2 디스크 암호화 형식을 지원

Red Hat Enterprise Linux 8 설치 프로그램에서는 기본적으로 LUKS2 형식을 사용하지만 Anaconda의 사용자 정의 파티션 설정 창 또는 Kickstart의 `autopart`, `logvol`, `part` 및 `RAID` 명령을 사용하여 LUKS 버전을 선택할 수 있습니다.

LUKS2에서는 많은 개선사항과 기능을 제공합니다. 예를 들어, 디스크 형식의 기능을 확장하고 메타데이터를 저장하는 유연한 방법을 제공합니다.

(BZ#1547908)

Anaconda에서는 RHEL 8의 시스템 용도를 지원

이전에는 Anaconda에서 서브스크립션 관리자에게 시스템 용도 정보를 제공하지 않았습니다. Red Hat Enterprise Linux 8.0에서는 Anaconda의 **System Purpose** 창 또는 Kickstart의 `syspurpose` 명령을 사용하여 설치 중에 시스템의 용도를 설정할 수 있습니다. 설치가 완료되면 서브스크립션 관리자가 시스템을 등록할 때 시스템 용도 정보를 사용합니다.

(BZ#1612060)

Pykickstart에서는 RHEL 8의 시스템 용도를 지원

이전에는 **pykickstart** 라이브러리에서 **서브스크립션 관리자**에게 시스템 용도 정보를 제공할 수 없었습니다. Red Hat Enterprise Linux 8.0에서 **pykickstart**는 새로운 **syspurpose** 명령을 구문 분석하고 전체 자동화 또는 부분적으로 자동화된 설치 중에 시스템의 용도를 기록합니다. 그러면 정보가 **Anaconda**에 전달되고 새로 설치된 시스템에 저장되어 다음 시스템 등록 시 **서브스크립션 관리자**가 사용할 수 있게 됩니다.

(BZ#1612061)

Anaconda에서는 RHEL 8의 새로운 커널 부트 매개변수를 지원

이전에는 커널 부트 매개변수의 기본 리포지토리만 지정할 수 있습니다. Red Hat Enterprise Linux 8에서 새로운 커널 매개변수, **inst.addrepo=<name>,<url>**을 사용하면 추가 리포지토리를 지정할 수 있습니다.

이 매개변수에는 두 가지 필수 값(즉, 리포지토리의 이름과 리포지토리를 가리키는 URL)이 있습니다. 자세한 내용은 <https://anaconda-installer.readthedocs.io/en/latest/boot-options.html#inst-addrepo>를 참조하십시오.

(BZ#1595415)

Anaconda에서는 RHEL 8의 통합 ISO를 지원

Red Hat Enterprise Linux 8.0에서 통합 ISO를 통해 BaseOS 및 AppStream 설치 소스 리포지토리를 자동으로 로드합니다.

이 기능은 설치 중에 로드되는 첫 번째 기본 리포지토리에서 작동합니다. 예를 들어 리포지토리를 설정하지 않고 설치를 시작하고 GUI에 통합 ISO가 기본 리포지토리로 포함되어 있거나 통합 ISO를 가리키는 **inst.repo=** 옵션을 사용하여 설치를 시작하는 경우가 이에 해당합니다. 결과적으로 AppStream 리포지토리는 설치 소스 GUI 창의 **추가 리포지토리** 섹션에서 사용됩니다. AppStream 리포지토리를 제거하거나 해당 설정을 변경할 수 없지만, **설치 소스**에서 이를 사용하지 않게 설정 해제할 수 있습니다. 이 기능은 다른 기본 리포지토리를 사용하여 설치를 시작한 다음 통합 ISO로 변경하면 작동하지 않습니다. 이렇게 하면 기본 리포지토리가 교체되지만 AppStream 리포지토리는 교체되지 않고 원래 파일을 가리킵니다.

(BZ#1610806)

4.3. 커널

ARM 52비트 물리적 주소 지정이 가능

이번 업데이트에서는 64비트 ARM 아키텍처용 52비트 물리적 주소 지정(PA)을 사용할 수 있습니다. 따라서 이전 48비트 PA보다 큰 주소 공간을 제공합니다.

(BZ#1643522)

IOMMU 코드는 RHEL 8의 5 레벨 페이지 테이블을 지원

Linux 커널의 IOMMU(I/O Memory Management Unit)은 Red Hat Enterprise Linux 8의 5레벨 페이지 테이블을 지원하도록 업데이트되었습니다.

(BZ#1485546)

5 레벨의 페이지징을 지원

Red Hat Enterprise Linux 8에서 5 레벨의 페이지징을 지원하기 위해 새로운 **P4d_t** 소프트웨어 페이지 테이블 유형이 Linux 커널에 추가되었습니다.

(BZ#1485532)

메모리 관리에서 5 레벨 페이지 테이블 지원

Red Hat Enterprise Linux 7의 경우 기존 메모리 버스는 48/46비트의 가상/실제 메모리 주소 지정 용량이 있으며 Linux 커널은 이러한 가상 주소를 실제 주소로 관리하기 위해 4 레벨의 페이지 테이블을 구현하고 있습니다. 물리적 버스 주소 지정 라인은 물리적 메모리 제한 용량을 64TB로 설정합니다.

이러한 제한은 128PiB의 가상 주소 공간과 4PB의 실제 메모리 용량이 있는 57/52 비트의 가상/실제 메모리 주소 지정까지 확장되었습니다.

확장된 주소 범위를 통해 Red Hat Enterprise Linux 8의 메모리 관리에서는 확장된 주소 범위를 처리할 수 있도록 5 레벨 페이지 테이블 구현을 지원합니다.

(BZ#1485525)

kernel-signing-ca.cer이 RHEL 8에서 kernel-core로 이동

Red Hat Enterprise Linux 7의 모든 버전에서 **kernel-signing-ca.cer** 공개 키는 **kernel-doc** 패키지에 있었지만 Red Hat Enterprise Linux 8에서 **kernel-signing-ca.cer**은 모든 아키텍처의 **kernel-core** 패키지로 재배포되었습니다.

(BZ#1638465)

eBPF 기반 프로그램 및 맵의 검사 및 조작을 위해 bpftool 추가

eBPF(extended Berkeley Packet Filtering)를 기반으로 프로그램과 맵의 검사 및 단순 조작에 사용하는 **bpftool** 유틸리티가 Linux 커널에 추가되었습니다. **bpftool**은 커널 소스 트리의 일부이며 **kernel** 패키지의 하위 패키지로 포함된 **bpftool**을 통해 제공됩니다.

(BZ#1559607)

kernel-rt 소스 업데이트

최신 RHEL 커널 소스 트리를 사용하도록 **kernel-rt** 소스가 업데이트되었습니다. 최신 커널 소스 트리에서는 이제 업스트림 v4.18 실시간 패치 세트를 사용하고 있으며 이전 버전에 비해 많은 버그 수정 및 개선 사항이 추가되었습니다.

(BZ#1592977)

4.4. 소프트웨어 관리

YUM 성능 개선 및 모듈식 콘텐츠 지원

Red Hat Enterprise Linux 8에서는 **DNF** 기술을 기반으로 하는 새로운 버전의 **YUM** 툴을 통해 소프트웨어를 설치합니다.

DNF를 기반으로 하는 **YUM**은 RHEL 7에서 사용된 이전 **YUM v3**에 비해 다음과 같은 이점이 있습니다.

- 성능 향상
- 모듈식 콘텐츠 지원
- 틀링과의 통합을 위해 적절하게 설계된 안정된 API

새로운 **YUM** 툴과 RHEL7의 이전 버전 **YUM v3**의 차이점에 관한 자세한 내용은 http://dnf.readthedocs.io/en/latest/cli_vs_yum.html을 참조하십시오.

DNF를 기반으로 하는 **YUM**은 명령행에서 사용하거나 구성 파일을 편집 또는 생성할 때 **YUM v3**과 호환됩니다.

소프트웨어 설치 시, RHEL 7에서와 동일한 방식으로 **yum** 명령과 특정 옵션을 사용할 수 있습니다.

Provides를 사용하여 이전 이름으로 패키지를 설치할 수 있습니다. 패키지는 호환성 심볼릭 링크를 제공하기 때문에 바이너리, 구성 파일 및 디렉터리는 일반적인 위치에서 찾을 수 있습니다.

YUM v3에서 제공하는 레거시 Python API와 Libdnf C API는 불안정하므로 Red Hat Enterprise Linux 8 라이프사이클 기간 내에 변경될 수 있습니다. 사용자는 플러그인과 스크립트를 안정적으로 지원하는 새로운 DNF Python API로 마이그레이션할 것을 권장합니다. DNF Python API는 <https://dnf.readthedocs.io/en/latest/api.html>에서 사용할 수 있습니다.

YUM v3 기능 중 일부는 **DNF**를 기반으로 하는 **YUM**에서 다르게 작동할 수 있습니다. 이러한 변경 때문에 워크플로에 부정적인 영향을 미치는 경우 [How do I open and manage a support case on the Customer Portal?](#)에 설명된 대로 Red Hat 지원 케이스를 작성하십시오.

(BZ#1581198)

RHEL 8의 주요 RPM 기능

Red Hat Enterprise Linux 8은 RPM 4.14와 함께 배포됩니다. 이 버전에서는 RHEL 7에서 사용할 수 있는 RPM 4.11에 대한 여러 개선사항을 소개합니다. 주요 기능은 다음과 같습니다.

- **debuginfo** 패키지의 여러 버전을 동시에 설치 가능
- 약한 종속성 지원
- 풍부한 부울 종속성 지원
- 4GB 이상의 파일 패키지 지원
- 파일 트리거 지원

주요 변경 사항은 다음과 같습니다.

- 정확한 사양 구문 분석기
- 상세하지 않은 모드에서 출력을 확인하는 간소화된 서명
- 매크로의 추가사항 및 폐지 사항

(BZ#1581990)

RPM이 설치를 시작하기 전에 전체 패키지 콘텐츠를 확인

Red Hat Enterprise Linux 7에서는 압축을 푸는 동안 **RPM** 유틸리티를 사용하여 개별 파일의 페이로드 콘텐츠를 검증했습니다. 그러나 이 방법은 다음과 같은 이유로 충분하지 않습니다.

- 페이로드가 손상된 경우 스크립트 작업을 실행한 후에만 페이로드가 발견되며 작업을 취소할 수 없습니다.
- 페이로드가 손상되면 이전 버전의 일부 파일을 교체한 후 패키지 업그레이드가 중단되므로 작동 중인 설치가 중단됩니다.
- 개별 파일의 해시는 압축 해제된 데이터에서 수행되므로 **RPM**은 압축 풀기 프로그램의 취약성에 취약합니다.

Red Hat Enterprise Linux 8에서 전체 패키지는 설치 전에 별도의 단계에서 사용 가능한 최상의 해시를 사용하여 유효성을 검사합니다.

Red Hat Enterprise Linux 8에서 빌드된 패키지는 압축된 페이로드의 새로운 **SHA256** 해시를 사용합니

다. 서명된 패키지에서 페이로드 해시는 서명으로 추가 보호되므로, 패키지 헤더에서 서명과 기타 해시를 손상시키지 않고 변경할 수 없습니다. 이전 패키지에서는 FIPS 모드와 같은 설정을 통해 비활성화되어 있지 않으면 헤더와 페이로드의 **MD5** 해시를 사용합니다.

`%_pkgverify_level` 매크로는 설치 전에 추가로 서명 검증을 강제 시행하거나 페이로드 검증을 완전히 비활성화하는 데 사용할 수 있습니다. 또한 **%_pkgverify_flags** 매크로를 사용하여 허용되는 해시와 서명을 제한할 수 있습니다. 예를 들어 약한 **MD5** 해시 사용을 비활성화할 수 있지만 이렇게 하면 이전 패키지와 호환되지 않습니다.

(JIRA:RHELPLAN-1499)

4.5. 인프라 서비스

RHEL 8 권장 Tuned 프로파일의 주요 변경 사항

이번 업데이트를 통해 다음 규칙에 따라 (**tuned-adm recommend** 명령을 통해 보고되는) 권장 Tuned 프로파일을 선택할 수 있습니다. 일치하는 첫 번째 규칙이 적용됩니다.

- (**syspurpose show** 명령을 통해 보고되는) **syspurpose** 역할에는 **atomic**이 포함되는 동시에 다음과 같이 선택됩니다.
 - 베어 메탈에서 Tuned가 실행 중이면 **atomic-host** 프로파일이 선택됩니다.
 - 가상 머신에서 Tuned가 실행 중이면 **atomic-guest** 프로파일이 선택됩니다.
- 가상 머신에서 Tuned가 실행 중이면 **virtual-guest** 프로파일이 선택됩니다.
- **syspurpose** 역할에 **desktop** 또는 **workstation**이 포함되어 있으며 새시 유형(**dmidecode**를 통해 보고)이 **Notebook**, **Laptop** 또는 **Portable**이면 **balanced** 프로파일이 선택됩니다.
- 위의 어떤 규칙에도 일치하지 않으면 **throughput-performance** 프로파일이 선택됩니다.

(BZ#1565598)

named로 생성된 파일은 작업 디렉터리에 쓰기 가능

이전에는 **named** 데몬에서 일부 데이터를 Red Hat Enterprise Linux에서 읽기 전용인 작업 디렉터리에 저장했습니다. 이번 업데이트에서는 선택한 파일의 경로가 쓰기가 허용된 하위 디렉터리로 변경되었습니다. 이제 기본 디렉터리 Unix와 SELinux 권한을 사용하면 디렉터리에 쓸 수 있습니다. 디렉터리에서 배포된 파일은 여전히 **named**에 대해 읽기 전용입니다.

(BZ#1588592)

Geolite Database는 Geolite2 Database로 대체

Red Hat Enterprise Linux 7에 있는 Geolite Database는 Red Hat Enterprise Linux 8의 Geolite2 Database로 대체되었습니다.

Geolite Database는 **GeoIP** 패키지로 제공됩니다. 레거시 데이터베이스와 함께 이 패키지는 더 이상 업스트림에서 지원되지 않습니다.

Geolite2 Database는 여러 패키지에서 제공됩니다. **libmaxminddb** 패키지에는 주소를 수동으로 검색하는 데 사용할 수 있는 **mmdblookup** 명령행 툴과 라이브러리가 포함되어 있습니다. 레거시 **GeoIP** 패키지의 **geoipupdate** 바이너리는 이제 **geoipupdate** 패키지에서 제공되며 레거시 데이터베이스와 새로운 Geolite2 데이터베이스를 모두 다운로드할 수 있습니다.

(JIRA:RHELPLAN-6746)

CUPS 로그는 journald에서 처리

RHEL 8에서는 RHEL 7에서 사용된 `/var/log/cups` 디렉터리의 특정 파일에 CUPS 로그가 더 이상 저장되지 않습니다. RHEL 8에서는 모든 유형의 CUPS 로그가 다른 프로그램의 로그와 함께 systemd `journald` 데몬으로 중앙에 로깅됩니다. CUPS 로그에 액세스하려면 `journalctl -u cups` 명령을 사용하십시오. 자세한 내용은 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[Working with CUPS logs]에서 참조하십시오.

(JIRA:RHELPLAN-12764)

4.6. 셸 및 명령행 툴

nobody 사용자가 nfsnobody로 대체

Red Hat Enterprise Linux 7에는 다음이 설정되어 있습니다.

- ID가 99인 **nobody** 사용자 및 그룹 쌍
- ID가 65534인 **nfsnobody** 사용자와 그룹 쌍. 이 ID는 기본 커널 오버플로 ID이기도 합니다.

이 두 ID 모두 Red Hat Enterprise Linux 8에서 65534 ID를 사용하는 **nobody** 사용자와 그룹 쌍으로 병합되어 새 설치에서는 더 이상 **nfsnobody** 쌍을 생성하지 않습니다.

이와 같이 변경하면 **nobody**에서 소유한 파일에 관한 혼동은 줄어들지만 NFS와는 아무 관련이 없습니다.

(BZ#1591969)

RHEL 8의 버전 관리 시스템

RHEL 8에서는 다음과 같은 버전 관리 시스템을 제공합니다.

- **Git 2.18** - 분산 아키텍처를 사용하는 분산 버전 관리 시스템입니다.
- **Mercurial 4.8** - 대규모 프로젝트를 효율적으로 처리하도록 설계된 경량의 분산 버전 관리 시스템입니다.
- **Subversion 1.10** - 중앙 집중식 버전 관리 시스템입니다. 자세한 정보는 "[Notable changes in Subversion 1.10](#)"를 참조하십시오.

RHEL 7에서 사용할 수 있는 CVS(Concurrent Versions System)는 RHEL 8에서는 배포되지 않습니다.

(BZ#1693775)

Subversion 1.10의 주요 변경 사항

Subversion 1.10에는 RHEL 7에서 배포한 버전 1.7 이후에 추가된 새로운 기능 및 다음과 같은 호환성 변경 사항이 추가되어 있습니다.

- 언어 바인딩을 지원하는 데 사용한 **Subversion** 라이브러리의 비호환성 때문에 **Subversion 1.10**의 **Python 3** 바인딩은 사용할 수 없습니다. 결과적으로 **Subversion**에 **Python** 바인딩이 필요한 애플리케이션은 지원되지 않습니다.
- **Berkeley DB**에 기반한 리포지토리는 더 이상 지원되지 않습니다. 마이그레이션하기 전에 `svnadmin dump` 명령을 사용하여 **Subversion 1.7**에서 생성된 리포지토리를 백업합니다. RHEL 8을 설치한 후 `svnadmin load` 명령을 사용하여 리포지토리를 복원합니다.

- RHEL 7의 **Subversion 1.7** 클라이언트에서 확인한 기존 작업 사본은 새 형식으로 업그레이드해야 **Subversion 1.10**에서 사용할 수 있습니다. RHEL 8을 설치한 다음 각 작업 사본에서 **svn upgrade** 명령을 실행합니다.
- **https://**를 사용하여 리포지토리에 액세스하는 스마트 카드 인증은 더 이상 지원되지 않습니다.

(BZ#1571415)

4.7. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

Python 3은 RHEL 8의 기본 Python 구현

Red Hat Enterprise Linux 8은 **Python 3.6**과 함께 배포됩니다. 이 패키지는 기본적으로 설치되지 않습니다. **Python 3.6**을 설치하려면 **yum install python3** 명령을 사용합니다.

Python 2.7은 **python2** 패키지에서 사용할 수 있습니다. **Python 2**는 라이프사이클이 더 짧기 때문에 고객이 더 원활하게 **Python 3**으로 전환할 수 있도록 지원됩니다.

기본 **python** 패키지와 버전이 지정되지 않은 **/usr/bin/python** 실행 파일 모두 RHEL 8과 함께 배포되지 않습니다. 고객은 **python3** 또는 **python2**를 직접 사용하는 것이 좋습니다. 또는 관리자가 **alternatives** 명령을 사용하여 버전이 없는 **python** 명령을 설정할 수 있습니다.

자세한 내용은 [Using Python in Red Hat Enterprise Linux 8](#) 을 참조하십시오.

(BZ#1580387)

RPM 빌드 시 Python 스크립트에서 hashbangs에 주요 버전을 지정

RHEL 8에서 실행 가능한 Python 스크립트에서는 주요 Python 버전을 명시적으로 지정하는 hashbangs(shebangs)을 사용합니다.

RPM 패키지를 빌드할 때 **/usr/lib/rpm/redhat/brp-mangle-shebangs** BRP(BuildRoot Policy) 스크립트를 자동으로 실행합니다. 이 스크립트에서는 모든 실행 파일에서 hashbangs를 정정하려고 시도합니다. 스크립트에서 Python의 주요 버전으로 분석될 수 없는 모호한 hashbangs가 발견되면 오류를 생성합니다. 이와 같이 모호한 hashbangs의 예는 다음과 같습니다.

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

RPM 빌드 시 이러한 빌드 오류를 초래하는 Python 스크립트의 hashbangs를 수정하려면 **platform-python-devel** 패키지에서 **pathfix.py** 스크립트를 사용하십시오.

```
pathfix.py -pn -i %[_python3] PATH ...
```

여러 개의 *PATH*를 지정할 수 있습니다. *PATH*가 디렉터리이면 **pathfix.py**에서 모호한 hashbang을 포함하는 스크립트 외에도 **^[a-zA-Z0-9_]+\.\$** 패턴과 일치하는 Python 스크립트를 반복적으로 스캔합니다. **%prep** 섹션에 이 명령을 추가하거나 **%install** 섹션의 끝에 추가합니다.

자세한 내용은 [Handling hashbangs in Python scripts](#) 를 참조하십시오.

(BZ#1583620)

PHP에서 주요 변경 사항

Red Hat Enterprise Linux 8은 **PHP 7.2**와 함께 배포됩니다. 이 버전에서는 RHEL 7에서 사용할 수 있는 **PHP 5.4**에 대해 다음과 같은 주요 변경 사항이 추가되었습니다.

- **PHP**에서는 기본적으로 **FPM(FastCGI Process Manager)**을 사용합니다(스레드된 **httpd**로 안전하게 사용 가능).
- **php_value**와 **php-flag** 변수는 더 이상 **httpd** 구성 파일에서 사용되지 않으며 대신 풀 구성 (**/etc/php-fpm.d/*.conf**)에서 설정해야 합니다.
- **PHP** 스크립트 오류와 경고는 **/var/log/httpd/error.log**가 아니라 **/var/log/php-fpm/www-error.log** 파일에 기록됩니다.
- **PHP max_execution_time** 설정 변수를 변경할 때 변경한 값에 따라 **httpd ProxyTimeout** 설정을 늘려야 합니다.
- **PHP** 스크립트를 실행하는 사용자는 이제 **FPM** 풀 설정 (**/etc/php-fpm.d/www.conf** 파일, **apache** 사용자가 기본값)으로 구성되어 있습니다.
- 설정 변경 후 또는 새로운 확장 기능을 설치한 후 **php-fpm** 서비스를 다시 시작해야 합니다.

다음의 확장 기능이 제거되었습니다.

- **aspell**
- **mysql**(**php-mysqldb** 패키지에서 제공하는 **mysqli** 및 **pdo_mysql** 확장 기능을 계속 사용 가능).
- **zip**
- **memcache**

(BZ#1580430, BZ#1691688)

Ruby에서 주요 변경 사항

RHEL 8에서는 RHEL 7에서 사용 가능한 **Ruby 2.0.0**의 새로운 기능과 개선사항이 추가된 **Ruby 2.5**를 제공합니다. 주요 변경 사항은 다음과 같습니다.

- 증분식 가비지 컬렉터 (Garbage Collector)가 추가되었습니다.
- **Refinements** 구문이 추가되었습니다.
- 기호는 이제 가비지 수집됩니다.
- **SAFE=2** 및 **SAFE=3** 안전 레벨은 이제 폐지됩니다.
- **Fixnum** 및 **Bignum** 클래스는 **Integer** 클래스로 통합되었습니다.
- **Hash** 클래스를 최적화하고 인스턴스 변수에 대한 액세스를 향상시키고 **Mutex** 클래스를 더 작고 빠르게 최적화하여 성능을 향상시킬 수 있습니다.
- 이전 API 중 일부는 더 이상 사용되지 않습니다.
- **RubyGems**, **Rake**, **RDoc**, **Psych**, **Minitest** 및 **test-unit**과 같이 번들된 라이브러리가 업데이트되었습니다.
- **Ruby**를 통해 이전에 배포된 **mathn**, **DL**, **ext/tk** 및 **XMLRPC** 등의 라이브러리는 더 이상 사용되지 않거나 포함되지 않습니다.
- **SemVer** 버전 지정 스키마는 이제 **Ruby** 버전 지정에 사용됩니다.

(BZ#1648843)

Perl에서 주요 변경 사항

RHEL 8을 통해 배포된 **Perl 5.26**에서는 RHEL 7에서 제공되었던 버전에 다음과 같은 변경 사항이 추가되었습니다.

- 이제 **유니코드 9.0**이 지원됩니다.
- 새로운 **op-entry, loading-file** 및 **loaded-file SystemTap** 프로브가 제공됩니다.
- 성능 향상을 위해 스칼라를 할당할 때 COW 메커니즘이 사용됩니다.
- IPv4 및 IPv6 소켓을 투명하게 처리하는 **IO::Socket::IP** 모듈이 추가되었습니다.
- 구조화된 방식으로 **perl -V** 데이터에 액세스하는 **Config::Perl::V** 모듈이 추가되었습니다.
- CPAN(Comprehensive Perl Archive Network) 리포지토리에서 모듈을 검색, 추출, 빌드 및 설치하는 **cpanm** 유틸리티가 포함된 새로운 **perl-App-cpanminus** 패키지가 추가되었습니다.
- 보안상의 이유로 현재 디렉터리 **.i @INC** 모듈 검색 경로에서 제거되었습니다.
- 위에 설명된 동작 변경을 사용하여 파일을 로드하는 데 실패했을 때 **do** 문에서는 사용 중단 경고를 표시합니다.
- **do subroutine(LIST)** 호출은 더 이상 지원되지 않으므로 구문 오류가 발생합니다.
- 현재 해시는 기본적으로 무작위로 지정됩니다. 해시에서 반환되는 키와 값의 순서는 **perl**을 실행할 때마다 변경됩니다. 무작위 지정을 비활성화하려면 **PERL_PERTURB_KEYS** 환경 변수를 **0**으로 설정합니다.
- 정규 표현식 패턴에서 이스케이프되지 않은 리터럴 **{** 문자는 더 이상 사용되지 않습니다.
- **\$_** 변수의 어휘 범위 지원은 제거되었습니다.
- 배열이나 해시에서 **defined** 연산자를 사용하면 치명적인 오류가 발생합니다.
- **UNIVERSAL** 모듈에서 함수를 가져올 때 치명적인 오류가 발생합니다.
- **find2perl, s2p, a2p, c2ph** 및 **pstruct** 툴이 제거되었습니다.
- **\${^ENCODING}** 기능이 제거되었습니다. **encoding** pragma의 기본 모드는 더 이상 지원되지 않습니다. **UTF-8** 이외의 다른 인코딩으로 소스 코드를 작성하려면 인코딩의 **Filter** 옵션을 사용합니다.
- 이제 **perl** 패키지가 업스트림에 맞게 변경되었습니다. **perl** 패키지에서는 코어 모듈을 설치하고 **/usr/bin/perl** 인터프리터는 **perl-interpreter** 패키지로 제공됩니다. 이전 릴리스에서는 **perl** 패키지에 최소한의 인터프리터만 포함되어 있고 **perl-core** 패키지에는 인터프리터와 코어 모듈 모두 포함되어 있었습니다.

(BZ#1511131)

RHEL 8에 새로 추가된 Node.js 기능

JavaScript 프로그래밍 언어로 빠르고 확장 가능한 네트워크 애플리케이션을 구축하기 위한 소프트웨어 개발 플랫폼인 **Node.js**가 RHEL에서 처음으로 제공됩니다. 이전에는 소프트웨어 컬렉션으로만 제공되었습니다. RHEL 8에서는 **Node.js 10**이 제공됩니다.

(BZ#1622118)

SWIG에서 주요 변경 사항

RHEL 8에는 RHEL 7에서 배포된 버전 2.0에 비해 많은 새로운 기능, 향상된 기능 및 버그 수정사항이 추가된 SWIG(Simplified Wrapper and Interface Generator) 버전 3.0이 포함되어 있습니다. 특히 C++ 11 표준 지원이 구현됩니다. 이제 SWIG에서는 Go 1.6, PHP 7, Octave 4.2 및 Python 3.5도 지원합니다.

(BZ#1660051)

Apache httpd에서 주요 변경 사항

RHEL 8은 Apache HTTP Server 2.4.37과 함께 배포됩니다. 이 버전에서는 RHEL 7에서 사용되던 **httpd**에 대해 다음과 같은 변경 사항이 추가되었습니다.

- HTTP/2 지원은 이제 **httpd** 모듈의 일부인 **mod_http2** 패키지를 통해 제공합니다.
- ACME(Automatic Certificate Management Environment) 프로토콜을 사용하는 자동화된 TLS 인증서 프로비저닝 및 업데이트가 이제 **mod_md** 패키지와 함께 지원됩니다(**Let's Encrypt**와 같은 인증 기관을 통해 사용).
- 이제 Apache HTTP Server에서는 **PKCS#11** 모듈에서 직접 하드웨어 보안 토큰의 TLS 인증서 및 개인 키를 로드하는 기능을 지원합니다. 결과적으로 이제 **mod_ssl** 구성에서 **PKCS#11** URL을 사용하여 TLS 개인키를 식별할 수 있으며, 선택적으로 **SSLCertificateKeyFile** 및 **SSLCertificateFile** 지시문으로 TLS 인증서를 식별할 수 있습니다.
- 기본적으로 Apache HTTP Server와 함께 구성된 멀티 프로세싱 모듈(MPM)은 멀티 프로세스 분기 모델(**prefork**라고도 함)에서 고성능 멀티 스레드 모델 **event**로 변경되었습니다. 스레드로부터 안전하지 않은 타사 모듈은 교체하거나 제거해야 합니다. 설정된 MPM을 변경하려면 **/etc/httpd/conf.modules.d/00-mpm.conf** 파일을 편집합니다. 자세한 내용은 **httpd.conf(5)** man 페이지를 참조하십시오.

httpd에 관한 자세한 내용은 [Setting up the Apache HTTP web server](#)를 참조하십시오.

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

RHEL8에서 새로 추가된 nginx 웹 서버 기능

RHEL 8에서는 HTTP 및 기타 프로토콜을 지원하는 웹 서버 및 프록시 서버인 **nginx 1.14**를 도입하여 높은 동시성, 성능 및 낮은 메모리 사용률에 중점을 두고 있습니다. **nginx**는 이전에는 소프트웨어 컬렉션으로만 사용 가능했습니다.

이제 **nginx** 웹 서버에서는 **PKCS#11** 모듈에서 직접 하드웨어 보안 토큰의 TLS 개인 키를 로드하는 기능을 지원합니다. 결과적으로 이제 **nginx** 설정에서 **PKCS#11** URL을 사용하여 **ssl_certificate_key** 지시문으로 TLS 개인 키를 지정할 수 있습니다.

(BZ#1545526)

RHEL 8의 데이터베이스 서버

RHEL 8에서는 다음과 같은 데이터베이스 서버를 제공합니다.

- **MySQL 8.0.** 다중 사용자, 멀티 스레드 SQL 데이터베이스 서버입니다. 이 서버는 **MySQL** 서버 데몬, **mysqld** 및 여러 클라이언트 프로그램으로 구성됩니다.
- **MariaDB 10.3.** 다중 사용자, 멀티 스레드 SQL 데이터베이스 서버입니다. 실용적인 목적으로 사용하기 위해 **MariaDB**는 **MySQL**과 바이너리 호환됩니다.
- **PostgreSQL 10** 및 **PostgreSQL 9.6.** 고급 오브젝트 관계형 데이터베이스 관리 시스템(DBMS)입니다.

- **Redis 5.** 고급 키-값 저장소입니다. 키에 문자열, 해시, 목록, 세트 및 정렬된 세트가 포함될 수 있으므로 데이터 구조 서버라고 합니다. RHEL에서 처음으로 **Redis**가 제공됩니다.

SSPL(Server Side Public License)을 사용하므로 RHEL 8.0에는 NoSQL **MongoDB** 데이터베이스 서버가 포함되어 있지 않습니다.

(BZ#1647908)

MySQL 8.0에서 주요 변경 사항

RHEL 8에 포함되는 **MySQL 8.0**은 다음과 같은 향상된 기능을 제공합니다.

- **MySQL**은 이제 데이터베이스 객체에 관한 정보를 저장하는 트랜잭션 데이터 사전을 통합합니다.
- **MySQL**에서는 이제 권한 컬렉션인 역할을 지원합니다.
- 기본 문자 세트가 **latin1**에서 **utf8mb4**로 변경되었습니다.
- 채귀 및 비채귀 CTE (Common Table Expression) 지원이 추가되었습니다.
- **MySQL**에서는 이제 관련 행을 사용하여 쿼리의 각 행에 대한 계산을 수행하는 창 함수를 지원합니다.
- **InnoDB**에서는 이제 locking read 문과 함께 **NOWAIT** 및 **SKIP LOCKED** 옵션을 지원합니다.
- GIS 관련 기능이 향상되었습니다.
- JSON 함수가 강화되었습니다.
- 새로운 **mariadb-connector-c** 패키지에서는 **MySQL** 및 **MariaDB**의 공통 클라이언트 라이브러리를 제공합니다. 이 라이브러리는 **MySQL** 및 **MariaDB** 데이터베이스 서버의 모든 버전에서 사용할 수 있습니다. 결과적으로 RHEL 8을 통해 배포된 **MySQL**과 **MariaDB** 서버에 구축되는 애플리케이션 중 하나에 연결할 수 있습니다.

또한 RHEL 8의 클라이언트 툴과 라이브러리는 기본적으로 업스트림 **MySQL 8.0** 버전에서 사용되는 **caching_sha2_password** 방법과 호환되지 않으므로, RHEL 8을 통해 배포된 **MySQL 8.0** 서버는 **mysql_native_password**를 기본 인증 플러그인으로 사용하도록 구성됩니다.

기본 인증 플러그인을 **caching_sha2_password**로 변경하려면 **/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** 파일을 다음과 같이 변경합니다.

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

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

MariaDB 10.3에서 주요 변경 사항

MariaDB 10.3에서는 RHEL 7에서 배포된 버전 5.5에 비해 많은 새로운 기능을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- 동기식 멀티 마스터 클러스터인 **MariaDB Galera Cluster**는 이제 **MariaDB**에 포함되어 있음
- **XtraDB** 대신 **InnoDB**를 기본 스토리지 엔진으로 사용
- CTE (Common Table Expression)
- 시스템 버전 테이블

- **FOR** 루프
- 표시되지 않는 열
- 순서
- **InnoDB** 인스턴트 **ADD COLUMN**
- 스토리지 엔진에 의존하지 않는 컬럼 압축
- 병렬 복제
- 멀티 소스 복제

또한 새로운 **mariadb-connector-c** 패키지에서는 **MySQL** 및 **MariaDB**의 공통 클라이언트 라이브러리를 제공합니다. 이 라이브러리는 **MySQL** 및 **MariaDB** 데이터베이스 서버의 모든 버전에서 사용할 수 있습니다. 결과적으로 RHEL 8을 통해 배포된 **MySQL**과 **MariaDB** 서버에 구축되는 애플리케이션 중 하나에 연결할 수 있습니다.

[Using MariaDB on Red Hat Enterprise Linux 8](#) 도 참조하십시오.

(BZ#1637034, BZ#1519450)

4.8. 데스크탑

RHEL 8의 GNOME 셸 (버전 3.28)

RHEL(Red Hat Enterprise Linux) 8에서는 GNOME 셸 버전 3.28을 사용할 수 있습니다. 주요 개선 사항은 다음과 같습니다.

- 새로운 GNOME Boxes 기능
- 새로운 온-스크린 키보드
- 장치 지원 확장 (특히 Thunderbolt 3 인터페이스의 통합)
- GNOME 소프트웨어, dconf-editor 및 GNOME 터미널 개선

(BZ#1649404)

Wayland를 기본 디스플레이 서버로 사용

Red Hat Enterprise Linux 8에서 GNOME 세션과 GDM(GNOME Display Manager)에서 기본 디스플레이 서버로 이전 RHEL 주요 버전에서 사용된 **X.org** 대신 **Wayland**를 사용합니다.

Wayland에서는 **X.org**와 비교하여 다음과 같은 이점과 개선사항을 제공합니다.

- 보다 강력한 보안 모델
- 개선된 멀티 모니터 처리
- 향상된 사용자 인터페이스(UI) 확장
- 데스크탑에서 직접 창 처리를 제어 가능

다음 기능은 현재 사용할 수 없거나 작동하지 않는 기능입니다.

- 멀티-GPU 설정은 **Wayland**에서 지원되지 않습니다.

- **NVIDIA** 바이너리 드라이버는 **Wayland**에서 작동하지 않습니다.
- **xrandr** 유틸리티는 해상도, 회전 및 레이아웃 처리 방법이 다르기 때문에 **Wayland**에서 작동하지 않습니다. 화면을 조작하는 기타 **X.org** 유틸리티도 **Wayland**에서 작동하지 않습니다.
- 화면 녹화, 원격 데스크탑 및 접근성은 **Wayland**에서 제대로 작동하지 않습니다.
- 클립보드 관리자는 사용할 수 없습니다.
- **Wayland**에서는 가상 머신 뷰어와 같이 X11 애플리케이션의 키보드 그랩(**grab**)을 무시합니다.
- 게스트 가상 머신(VM)의 **Wayland**에는 안정성과 성능에 문제가 있으므로 가상 환경의 경우 X11 세션을 사용하는 것이 좋습니다.

X.org GNOME 세션을 사용하는 RHEL 7 시스템에서 RHEL 8로 업그레이드하는 경우 시스템은 계속 **X.org**를 사용합니다. 다음 그래픽 드라이버를 사용하는 경우 시스템에서 **X.org**로 자동 대체합니다.

- **NVIDIA** 바이너리 드라이버
- **cirrus** 드라이버
- **mga** 드라이버
- **aspeed** 드라이버

Wayland 사용을 수동으로 비활성화할 수 있습니다.

- **GDM**에서 **Wayland**를 비활성화하려면 `/etc/gdm/custom.conf` 파일에 **WaylandEnable=false** 옵션을 설정합니다.
- GNOME 세션에서 **Wayland**를 비활성화하려면 로그인 이름을 입력한 다음 로그인 화면의 설정 (톱니바퀴 모양) 메뉴에서 legacy X11 옵션을 선택합니다.

Wayland에 관한 자세한 내용은 <https://wayland.freedesktop.org/>를 참조하십시오.

(BZ#1589678)

기본적으로 사용되지 않는 리포지토리에 있는 RPM 패키지 검색

기본적으로 데스크탑의 추가 리포지토리는 사용하도록 설정되어 있지 않습니다. 비활성화는 해당 **.repo** 파일의 **enabled=0** 행에 지정됩니다. **PackageKit**을 사용하여 이러한 리포지토리에서 패키지를 설치하려고 하면 **PackageKit**에서 애플리케이션을 사용할 수 없다는 오류 메시지가 표시됩니다. 패키지를 사용 가능하게 하려면 각각의 **.repo** 파일에서 이전에 사용한 **enabled=0** 행을 **enabled=1**로 변경합니다.

(JIRA:RHELPLAN-2878)

패키지 관리용 GNOME 소프트웨어

Red Hat Enterprise Linux 7의 그래픽 환경에서 패키지 관리용 툴 컬렉션을 제공한 **gnome-packagekit** 패키지는 더 이상 제공되지 않습니다. Red Hat Enterprise Linux 8에서는 애플리케이션과 **gnome-shell** 확장 기능을 설치 및 업데이트할 수 있는 **GNOME 소프트웨어** 유틸리티에서 유사한 기능을 제공합니다. **GNOME 소프트웨어**는 **gnome-software** 패키지로 배포됩니다.

(JIRA:RHELPLAN-3001)

4.9. IDM (IDENTITY MANAGEMENT)

Directory Server에서 새 암호 구문 검사

이번 개선된 기능에는 Directory Server에 새로운 암호 구문 검사 기능이 추가되었습니다. 예를 들어 관리자는 사전 검사를 활성화하고 문자 순서 및 회문(palindrome)을 사용하여 허용 또는 거부할 수 있습니다. 따라서 이 기능이 활성화된 경우 Directory Server의 암호 정책 구문 검사를 통해 더 안전한 암호가 적용됩니다.

(BZ#1334254)

Directory Server에서 개선된 내부 작업 로그 지원 제공

서버 및 클라이언트에서 시작한 Directory Server 작업 때문에 백그라운드에서 추가 작업이 수행됩니다. 이전에는 서버에서 내부 작업을 위해 **Internal** 연결 키워드만 기록했으며 작업 ID는 항상 **-1**로 설정되었습니다. 이번 개선된 기능을 통해 Directory Server에서는 실제 연결 및 작업 ID를 기록할 수 있으므로 이 작업을 유발한 서버 또는 클라이언트 작업에 관한 내부 작업을 추적할 수 있습니다.

내부 작업 로깅에 관한 자세한 내용은 [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)를 참조하십시오.

(BZ#1358706)

tomcatjss 라이브러리에서는 AIA 확장의 응답자를 사용하여 OCSP 검사를 지원

이 향상된 기능을 통해 **tomcatjss** 라이브러리에서는 인증서의 AIA(Authority Information Access) 확장의 응답자를 사용하여 OCSP(Online Certificate Status Protocol) 검사를 지원합니다. 결과적으로 Red Hat Certificate System 관리자는 AIA 확장의 URL을 사용하여 OCSP 검사를 설정할 수 있습니다.

(BZ#1636564)

Directory Server에서는 인스턴스를 관리하는 새로운 명령행 유틸리티 제공

Red Hat Directory Server 11.0에서는 **dscreate**, **dsconf** 및 **dsctl** 유틸리티를 제공합니다. 이 유틸리티에서는 명령행을 사용하여 Directory Server 관리를 단순화합니다. 예를 들어 복잡한 LDIF 문을 서버에 보내는 대신 매개변수를 지정하고 명령을 사용하여 기능을 설정할 수 있습니다.

다음은 각 유틸리티의 사용 개요입니다.

- 대화식 모드 또는 INF 파일을 사용하여 새 Directory Server 인스턴스를 생성하려면 **dscreate** 유틸리티를 사용하십시오. INF 파일 형식은 이전 Directory Server 버전에서 사용된 설치 프로그램의 형식과 다릅니다.
- 런타임 시 **dsconf** 유틸리티를 사용하여 Directory Server 인스턴스를 관리합니다. 예를 들어, **dsconf**를 사용하여 다음을 수행합니다.
 - **cn=config** 항목을 설정합니다.
 - 플러그인을 설정합니다.
 - 복제를 설정합니다.
 - 인스턴스를 백업 및 복원합니다.
- 런타임 시 **dsctl** 유틸리티를 사용하여 오프라인 상태에서 Directory Server 인스턴스를 관리하십시오. 예를 들어, **dsctl**을 사용하여 다음을 수행합니다.
 - 인스턴스를 시작 및 중지합니다.
 - 서버 데이터베이스의 인덱스를 다시 작성합니다.

- 인스턴스를 백업 및 복원합니다.

이 유틸리티는 Directory Server 10에서 더 이상 사용되지 않는 것으로 표시된 Perl 및 셸 스크립트를 대체합니다. 지원되지 않는 **389-ds-base-legacy-tools** 패키지에서 계속 스크립트를 사용할 수 있지만 Red Hat에서는 새로운 유틸리티를 사용하여 Directory Server 관리만 지원합니다.

LDIF 명령문을 사용하여 Directory Server를 구성하는 작업은 계속 지원되지만 Red Hat에서는 유틸리티 사용을 권장합니다.

유틸리티 사용에 대한 자세한 내용은 [Red Hat Directory Server 11 문서](#) 를 참조하십시오.

(BZ#1693159)

pki subsystem-cert-find 및 **pki subsystem-cert-show** 명령은 인증서의 일련 번호를 표시

이 향상된 기능을 통해 인증서 시스템의 **pki subsystem-cert-find** 명령 및 **pki subsystem-cert-show** 명령에서 출력에 인증서의 일련 번호를 표시할 수 있게 되었습니다. 일련 번호는 중요한 정보이며 여러 다른 명령에 필요합니다. 결과적으로 인증서의 일련 번호를 쉽게 식별할 수 있습니다.

(BZ#1566360)

인증서 시스템에서 **pki user** 명령 및 **pki group** 명령은 더 이상 사용되지 않음

이번 업데이트를 통해 새로운 **pki <subsystem>-user** 및 **pki <subsystem>-group** 명령으로 인증서 시스템의 **pki user** 명령 및 **pki group** 명령이 교체되었습니다. 교체된 명령으로 계속 사용할 수 있지만 명령이 더 이상 사용되지 않으며 새 명령을 참조하라는 메시지를 표시합니다.

(BZ#1394069)

인증서 시스템에서 시스템 인증서의 오프라인 업데이트를 지원

이 향상된 기능을 통해 관리자는 오프라인 업데이트 기능을 사용하여 인증서 시스템에 구성된 시스템 인증서를 갱신할 수 있습니다. 시스템 인증서가 만료되면 인증서 시스템이 시작되지 않습니다. 이러한 기능 개선으로 관리자는 더 이상 만료된 시스템 인증서를 교체하기 위한 임시 해결방법이 필요하지 않습니다.

(BZ#1669257)

인증서 시스템에서 외부 CA 서명을 위해 SKI 확장을 사용하여 CSR을 생성 가능

이 향상된 기능을 통해 인증서 시스템에서는 외부 인증 기관(CA) 서명을 위해 SKI(Subject Key Identifier) 확장을 사용하여 인증서 서명 요청(CSR)을 생성하도록 지원합니다. 특정 CA에서는 이 확장이 특정 값을 사용하거나 CA 공용 키에서 파생되어야 합니다. 결과적으로 관리자는 이제 **pkispawn** 유틸리티에 전달된 설정 파일에서 **pki_req_ski** 매개 변수를 사용하여 SKI 확장자가 있는 CSR을 생성할 수 있습니다.

(BZ#1656856)

로컬 사용자는 SSSD를 통해 캐시하고 **nss_sss** 모듈을 통해 처리

RHEL 8의 SSSD(System Security Services Daemon)에서 기본적으로 **/etc/passwd** 및 **/etc/groups** 파일의 사용자와 그룹에 서비스를 제공합니다. **sss** nsswitch 모듈은 **/etc/nsswitch.conf** 파일의 파일 앞에 선행합니다.

SSSD를 통해 로컬 사용자에게 서비스를 제공하면 **memory-mapped cache**가 **nss_sss** 모듈에 있기 때문에 디스크에 액세스하여 NSS 요청마다 파일을 여는 것과 비교하여 NSS(Name Service Switch) 검색의 속도를 높이는 이점이 있습니다. 이전에는 Name 서비스 캐시 데몬(**nscd**)을 통해 디스크 액세스 처리 속도를 가속화했습니다. 그러나 SSSD와 **nscd** 모두 자신의 고유한 독립 캐싱을 사용하므로 SSSD와 **nscd**를 동시에 사용하는 것이 쉽지 않을 수 있습니다. 결과적으로 SSSD에서 원격 도메인의 사용자에게도 서비스를 제공하는 설정(예: LDAP 또는 Active Directory)에서 **nscd**를 사용하면 예상치 못한 동작이 발생할 수 있습니다.

이번 업데이트를 사용하면 RHEL 8에서 더 빠르게 로컬 사용자와 그룹을 분석할 수 있습니다. **root** 사용자는 SSSD에서 처리하지 않으므로 **root** 분석은 SSSD에 있는 잠재적 버그의 영향을 받을 수 없습니다. 또한 SSSD가 실행되지 않으면 **nss_sss** 모듈에서 문제를 해결하기 위해 **nss_files**로 대체하여 상황을 단계적으로 적절하게 처리합니다. 어떠한 방식으로든 SSSD를 설정할 필요가 없으며 files 도메인은 자동으로 추가됩니다.

(JIRA:RHELPLAN-10439)

KCM은 KEYRING을 기본 인증 정보 캐시 스토리지로 대체

RHEL 8에서 기본 인증 정보 캐시 스토리지는 **sssd-kcm** 데몬에서 지원하는 KCM(Kerberos Credential Manager)입니다. KCM에는 이전에 사용된 KEYRING의 제한 사항(예: 네임 스페이스가 없기 때문에 컨테이너 환경에서 사용하기가 어렵고 할당량을 표시 및 관리할 수 없음)이 해결되어 있습니다.

이번 업데이트를 통해 RHEL 8에는 컨테이너화된 환경에 더 적합하고 향후 릴리스에서 보다 많은 기능을 빌드할 수 있는 기반을 제공하는 인증 정보 캐시가 제공됩니다.

(JIRA:RHELPLAN-10440)

Active Directory 사용자가 Identity Management를 관리 가능

이번 업데이트를 통해 RHEL 8에서는 IdM(Identity Management) 그룹의 구성원으로 AD(Active Directory) 사용자에 대한 사용자 ID 재정의의 추가할 수 있습니다. ID 재정의는 특정 ID 보기(이 경우 기본 신뢰 보기) 내에서 특정 AD 사용자 또는 그룹 속성이 표시되는 방식을 설명하는 레코드입니다. 업데이트 결과 IdM LDAP 서버에서 IdM 그룹에 대한 액세스 제어 규칙을 AD 사용자에게 적용할 수 있습니다.

이제 AD 사용자가 IdM UI의 셸프 서비스 기능을 사용하여 SSH 키를 업로드하거나 개인 데이터를 변경하는 등의 작업을 수행할 수 있습니다. AD 관리자가 두 개의 서로 다른 계정 및 암호 없이도 IdM을 완전히 관리할 수 있습니다. 현재 AD 사용자는 IdM에서 선택된 일부 기능을 사용할 수 없다는 점에 유의하십시오.

(JIRA:RHELPLAN-10442)

sssctl을 통해 IdM 도메인의 HBAC 규칙 보고서 출력

이번 업데이트에서 SSSD(System Security Services Daemon)의 **sssctl** 유틸리티를 통해 IdM(Identity Management) 도메인의 액세스 제어 보고서를 출력할 수 있습니다. 이는 특정 환경에서 규제상의 이유로 특정 클라이언트 컴퓨터에 액세스할 수 있는 사용자 및 그룹 목록을 표시하는 기능으로 사용될 수 있습니다. IdM 클라이언트에서 **sssctl access-report domain_name**을 실행하면 클라이언트 시스템에 적용되는 IdM 도메인의 호스트 기반 액세스 제어(HBAC) 규칙의 구문 분석된 하위 세트를 표시합니다.

IdM 이외의 모든 공급자는 이 기능을 지원하지 않습니다.

(JIRA:RHELPLAN-10443)

Identity Management 패키지는 모듈로 사용 가능

RHEL 8에서는 IdM(Identity Management) 서버 및 클라이언트 설치에 필요한 패키지가 모듈로 제공됩니다. **client** 스트림은 **idm** 모듈의 기본 스트림이며 스트림을 활성화하지 않고 클라이언트 설치에 필요한 패키지를 다운로드할 수 있습니다.

IdM 서버 모듈 스트림은 **DL1** 스트림이라고 합니다. 스트림에는 다양한 유형의 IdM 서버에 해당하는 여러 프로파일이 포함되어 있습니다(서버, dns, adtrust, 클라이언트 및 기본값). **DL1** 스트림의 특정 프로파일에 있는 패키지를 다운로드하려면 스트림을 사용하도록 설정하여 스트림을 통해 전달된 RPM으로 전환한 후 **yum module install idm:DL1/profile_name** 명령을 실행합니다.

(JIRA:RHELPLAN-10438)

RHEL 8의 세션 레코딩 솔루션 추가

세션 레코딩 솔루션이 RHEL 8(Red Hat Enterprise Linux 8)에 추가되었습니다. 새로운 **tlog** 패키지와 관련 웹 콘솔 세션 플레이어를 사용하여 사용자 터미널 세션을 레코딩하고 재생할 수 있습니다. SSSD(System Security Services Daemon) 서비스를 통해 사용자 또는 사용자 그룹별로 레코딩을 구성할 수 있습니다. 모든 터미널 입력 및 출력은 캡처되어 시스템 저널에 텍스트 기반 형식으로 저장됩니다. 보안상의 이유로 원시 암호 및 기타 중요한 정보를 가로채지 않도록 입력은 기본적으로 비활성화됩니다.

이 솔루션은 보안에 민감한 시스템에서 사용자 세션을 감사하는 데 사용할 수 있습니다. 보안 위반이 발생한 경우 기록된 세션을 법의학 분석의 일부로 검토할 수 있습니다. 이제 시스템 관리자는 세션 레코딩을 로컬로 설정하고 **tlog-play** 유틸리티를 사용하여 명령행 인터페이스 또는 RHEL 8 웹 콘솔 인터페이스의 결과를 볼 수 있습니다.

(JIRA:RHELPLAN-1473)

authselect를 통해 사용자 인증 설정 간소화

이번 업데이트에서는 RHEL 8 호스트에서 사용자 인증 설정을 간소화하고 **authconfig** 유틸리티를 대신하여 **authselect** 유틸리티를 도입하고 있습니다. **authselect**에는 시스템 관리자가 PAM 설정을 보다 간단하게 변경할 수 있도록 PAM 스택을 더 안전하게 관리하는 접근 방식이 포함되어 있습니다. **authselect**는 암호, 인증서, 스마트 카드, 지문과 같은 인증 방법을 설정하는데 사용할 수 있습니다. **authselect**를 통해서만 원격 도메인에 가입하는 데 필요한 서비스를 설정하지 않습니다. 이 작업은 **realmd** 또는 **ipa-client-install**과 같은 특수화된 툴을 통해 수행합니다.

(JIRA:RHELPLAN-10445)

SSSD를 사용하여 여러 스마트 카드 인증 장치 중 하나를 선택 가능

이 업데이트를 사용하면 스마트 카드 인증 장치를 선택하도록 PKCS#11 URI를 설정할 수 있습니다.

기본적으로 SSSD에서는 스마트 카드 인증에 사용할 장치를 자동으로 검색합니다. 여러 장치가 연결되어 있는 경우 SSSD에서 첫 번째로 발견한 장치를 선택하고 특정 장치를 선택할 수 없으므로 실패할 가능성도 있습니다.

따라서 **sssd.conf**의 **[pam]** 섹션에 맞게 새로운 **p11_uri**를 설정할 수 있습니다. 이 옵션을 사용하면 스마트 카드 인증에 사용할 장치를 정의할 수 있습니다.

예를 들어 OpenSC PKCS#11 모듈에서 감지한 슬롯 ID가 '2'인 리더를 선택하려면

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

sssd.conf의 **[pam]** 섹션에 추가합니다.

자세한 내용은 **man sssd-conf**를 참조하십시오.

(BZ#1620123)

4.10. 컴파일러 및 개발 도구

Boost가 1.66 버전으로 업데이트

Boost C++ 라이브러리가 업스트림 버전 1.66으로 업데이트되었습니다. Red Hat Enterprise Linux 7에 포함된 **Boost**는 1.53 버전입니다. 자세한 내용은 업스트림 변경 로그 <https://www.boost.org/users/history/>를 참조하십시오.

이번 업데이트에서는 이전 버전과의 호환성을 손상시키는 다음과 같은 사항이 변경되었습니다.

- **bs_set_hook()** 함수, **splay** 컨테이너의 **splay_set_hook()** 함수 및 **Intrusive** 라이브러리에 있는 **splaytree_algorithms()** 함수의 **bool splay = true** 추가 매개변수가 제거되었습니다.

- JSON 파일의 주석 또는 문자열 연결은 **Property Tree** 라이브러리의 구문 분석기에서 더 이상 지원하지 않습니다.
- **Math** 라이브러리의 일부 배포 및 특수 기능은 문서화된 대로 동작하며 최대 유효 값을 반환하지 않고 **overflow_error**를 전송하도록 수정되었습니다.
- **Math** 라이브러리의 일부 헤더가 **libs/math/include_private** 디렉터리로 이동되었습니다.
- **Regex** 라이브러리의 **basic_regex<>::mark_count()** 및 **basic_regex<>::subexpression(n)** 함수의 동작이 문서에 맞게 변경되었습니다.
- **Variant** 라이브러리에서 가변성 템플릿을 사용하면 메타 프로그래밍 기능이 손상될 수 있습니다.
- **boost::python::numeric** API가 제거되었습니다. 대신 **boost::python::numpy**를 사용할 수 있습니다.
- 비오브젝트 유형의 포인터에 대한 연산은 더 이상 **Atomic** 라이브러리에서 제공되지 않습니다.

(BZ#1494495)

유니코드 11.0.0 지원

Red Hat Enterprise Linux 코어 C 라이브러리인 **glibc**는 유니코드 표준 버전 11.0.0을 지원하도록 업데이트되었습니다. 결과적으로 문자 세트 간 음역 및 변환을 포함한 모든 와이드 문자 및 멀티 바이트 문자 API는 이 표준을 준수하는 정확한 정보를 제공합니다.

(BZ#1512004)

boost 패키지는 Python에서 독립

이번 업데이트에서는 **boost** 패키지를 설치해도 더 이상 **Boost.Python** 라이브러리가 종속 항목으로 설치되지 않습니다. **Boost.Python**을 사용하려면 **boost-python3** 또는 **boost-python3-devel** 패키지를 설치해야 합니다.

(BZ#1616244)

새로운 compat-libgfortran-48 패키지 사용 가능

Fortran 라이브러리를 사용하는 Red Hat Enterprise Linux 6 및 7 애플리케이션과의 호환성을 위해 이제 **libgfortran.so.3** 라이브러리를 제공하는 새로운 **compat-libgfortran-48** 호환성 패키지가 사용 가능합니다.

(BZ#1607227)

GCC에서 Retpoline 지원

이번 업데이트에서는 GCC에 retpolines 지원이 추가되었습니다. retpoline은 CVE-2017-5715에 설명된 Spectre Variant 2 공격을 완화하는 오버헤드를 줄이기 위해 커널에서 사용하는 소프트웨어 구조입니다.

(BZ#1535774)

툴체인 구성 요소에서 64비트 ARM 아키텍처의 지원 강화

툴체인 구성 요소인 **GCC** 및 **binutils**에서는 이제 64비트 ARM 아키텍처에 대한 확장된 지원을 제공합니다. 예를 들어 다음과 같습니다.

- **GCC**와 **binutils**에서는 이제 SVE(Scalable Vector Extension)를 지원합니다.

- ARM v8.2에서 제공하는 **FP16** 데이터 유형에 대한 지원이 **GCC**에 추가되었습니다. **FP16** 데이터 유형을 사용하면 특정 알고리즘의 성능이 향상됩니다.
- **binutils**의 툴에서는 이제 포인터 인증을 포함하는 ARM v8.3 아키텍처 정의를 지원합니다. 포인터 인증 기능을 사용하면 악성 코드를 통해 고유 함수 포인터를 작성하여 프로그램 또는 커널의 정상 실행을 손상시키는 것을 방지할 수 있습니다. 따라서 코드에서 다른 위치로 분기할 때 신뢰할 수 있는 주소만 사용하므로 보안이 향상됩니다.

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

IBM POWER 시스템에 맞게 **glibc** 최적화

이번 업데이트에서는 IBM POWER 8 및 IBM POWER 9 아키텍처에 최적화된 새 버전의 **glibc**를 제공합니다. 결과적으로 IBM POWER 8 및 IBM POWER 9 시스템은 이제 런타임 시 최적화된 적절한 **glibc** 변형으로 자동 전환합니다.

(BZ#1376834)

GNU C 라이브러리 2.28 버전으로 업데이트

Red Hat Enterprise Linux 8에는 2.28 버전의 GNU C 라이브러리(**glibc**)가 포함되어 있습니다. 주요 개선 사항은 다음과 같습니다.

- 보안 강화 기능:
 - **AT_SECURE** 플래그로 표시된 보안 바이너리 파일은 **LD_LIBRARY_PATH** 환경 변수를 무시합니다.
 - 시스템 종료 속도를 높이고 손상된 환경에서 추가 코드를 실행하지 않도록 스택 검사 실패에 대한 역추적은 더 이상 표시되지 않습니다.
- 성능 향상:
 - 스레드 로컬 캐시로 **malloc()** 함수의 성능이 향상되었습니다.
 - 라이브러리 성능 특성을 변경하기 위해 **GLIBC_TUNABLES** 환경 변수가 추가되었습니다.
 - 스레드 세마포어의 구현이 개선되고 확장 가능한 새로운 **pthread_rwlock_xxx()** 함수가 추가되었습니다.
 - **math** 라이브러리의 성능이 향상되었습니다.
- 유니코드 11.0.0에 대한 지원이 추가되었습니다.
- ISO/IEC/IEEE 60559:2011, IEEE 754-2008 및 ISO/IEC TS 18661-3:2015 표준에 정의된 대로 128 비트 부동 소수점 지원이 추가되었습니다.
- **/etc/resolv.conf** 설정 파일과 관련하여 DNS(Domain Name Service) 스텝 해결 조치가 개선되었습니다.
 - 파일이 변경되면 설정이 자동으로 다시 로드됩니다.
 - 모든 검색 도메인에 대한 지원이 추가되었습니다.
 - **rotate** 옵션에 대한 적절한 무작위 선택이 추가되었습니다.
- 다음과 같은 개발을 위한 새로운 기능이 추가되었습니다.
 - **preadv2** 및 **pwritev2** 커널 호출의 Linux 래퍼 함수

- `reallocarray()` 및 `explicit_bzero()` 등의 새로운 기능
- `POSIX_SPAWN_SETSID`와 같은 `posix_spawnattr_setflags()` 함수의 새로운 플래그

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

RHEL에서 CMake 사용 가능

CMake 빌드 시스템 버전 3.11은 Red Hat Enterprise Linux 8에서 **cmake** 패키지로 사용 가능합니다.

(BZ#1590139, BZ#1502802)

make 버전 4.2.1

Red Hat Enterprise Linux 8은 **make** 빌드 툴 버전 4.2.1과 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 레시피에 실패하면 makefile의 이름과 레시피의 행 번호가 표시됩니다.
- 대상 추적을 사용하기 위해 `--trace` 옵션이 추가되었습니다. 이 옵션을 사용하면 비활성화된 경우에도 호출 전에 모든 레시피가 출력됩니다. 이때 레시피가 들어있는 파일 이름, 행 번호 및 이 기능을 호출하기 위한 전제 조건도 함께 출력됩니다.
- 명시적 규칙과 암시적 규칙을 혼합해도 더 이상 **make**의 실행이 종료되지 않습니다. 대신 경고가 표시됩니다. 이 구문은 더 이상 사용되지 않으며 향후에 완전히 제거될 수 있습니다.
- 파일에 텍스트를 쓰기 위해 `$(file ...)` 함수가 추가되었습니다. 텍스트 인수 없이 호출하면 파일을 연 후 바로 닫습니다.
- 새로운 옵션인 `--output-sync` 또는 `-O`를 사용하면 작업별로 여러 작업의 출력이 그룹화되며 병렬 빌드를 디버깅하기가 쉬워집니다.
- `--debug` 옵션은 현재 사용된 모든 디버깅 설정을 비활성화하기 위해 이제 `n(none)` 플래그도 허용합니다.
- BSD makefile과의 호환성을 높이기 위해 `$(shell ...)` 함수 대신 `!=` 셸 대입 연산자가 추가되었습니다. 연산자와 함수 사이의 차이점 및 자세한 내용은 GNU make 매뉴얼을 참조하십시오.

결과적으로, `variable!=value`와 같이 이름이 느낌표로 끝나고 바로 뒤에 대입 연산자가 오는 변수는 이제 새로운 구문으로 해석됩니다. 이전 동작을 복원하려면 `variable! =value`와 같이 느낌표 뒤에 공백을 추가하십시오.

+

- POSIX 표준에 정의된 `::=` 대입 연산자가 추가되었습니다.
- **.POSIX** 변수가 지정되면 **make**에서 백 슬래시와 개행을 처리하기 위한 POSIX 표준 요구사항을 준수합니다. 이 모드에서는 백슬래시 앞에 있는 후행 공백이 보존되고 새로운 백슬래시 뒤에 새 줄과 공백 문자가 하나의 공백 문자로 변환됩니다.
- **MAKEFLAGS** 및 **MFLAGS** 변수의 동작이 보다 정확하게 정의되었습니다.
- **make** 플래그의 새로운 변수인 **GNUMAKEFLAGS**는 **MAKEFLAGS**와 동일하게 구문 분석합니다. 결과적으로, GNU **make** 특정 플래그는 **MAKEFLAGS** 외부에 저장할 수 있으며 makefile의 이 동성이 향상됩니다.
- 호스트 아키텍처를 포함하는 새로운 변수인 **MAKE_HOST**가 추가되었습니다.

- 새로운 변수인 **MAKE_TERMOUT**과 **MAKE_TERMERR**을 통해 **make**에서 표준 출력 및 오류를 터미널에 쓰고 있는지 여부가 표시됩니다.
- makefile의 **MAKEFLAGS** 변수에 **-r** 및 **-R** 옵션을 설정하면 제대로 작동하고 모든 기본 규칙과 변수가 적절하게 제거되었습니다.
- 이제 레시피별로 **.RECIPEPREFIX** 설정이 기억됩니다. 또한 해당 레시피에서 확장된 변수에서도 레시피 접두사 설정을 사용합니다.
- **.RECIPEPREFIX** 설정과 모든 대상별 변수는 주석이 아니라 makefile에 있는 것처럼 **-p** 옵션의 출력에 표시됩니다.

(BZ#1641015)

Go Toolset로 빌드된 Go 프로그램은 FIPS와 호환 가능

호스트 시스템이 FIPS 모드로 구성된 경우 **Go Toolset**에서 사용 가능한 암호화 라이브러리가 **OpenSSL** 라이브러리 버전 1.1.0을 사용하도록 변경되었습니다. 결과적으로 이 버전의 **Go Toolset**으로 빌드된 프로그램은 FIPS와 호환됩니다.

Go 프로그램에서 인증되지 않은 표준 암호화 루틴만 사용하게 하려면 빌드 시 **go** 컴파일러의 **-tags no_openssl** 옵션을 사용하십시오.

(BZ#1512570)

SystemTap 버전 4.0

Red Hat Enterprise Linux 8은 **SystemTap** 측정 툴 버전 4.0과 함께 배포됩니다. 주요 개선사항은 다음과 같습니다.

- eBPF(extended Berkeley Packet Filter) 백엔드에서 특히 문자열과 함수가 향상되었습니다. 이 백엔드를 사용하려면 **SystemTap**을 시작할 때 **--runtime=bpf** 옵션을 사용합니다.
- Prometheus 모니터링 시스템과 함께 사용할 새로운 내보내기 네트워크 서비스가 추가되었습니다.
- 필요에 따라 커널 추적 포인트를 사용하도록 시스템 호출 검사 구현이 개선되었습니다.

(BZ#1641032)

binutils 버전 2.30의 개선사항

Red Hat Enterprise Linux 8에는 2.30 버전의 **binutils** 패키지가 포함되어 있습니다. 주요 개선사항은 다음과 같습니다.

- 새로운 s390x 아키텍처 확장에 대한 지원이 개선되었습니다.

어셈블러:

- WebAssembly 파일 형식에 대한 지원 및 WebAssembly를 wasm32 ELF 파일 형식으로 변환하는 기능이 추가되었습니다.
- ARMv8-R 아키텍처 및 Cortex-R52, Cortex-M23 및 Cortex-M33 프로세서에 대한 지원이 추가되었습니다.
- RISC-V 아키텍처 지원이 추가되었습니다.

링커:

- 링커는 기본적으로 코드와 읽기 전용 데이터를 별도의 세그먼트에 배치할 수 있게 되었습니다. 동적 로더는 읽기 전용 데이터가 포함된 메모리 페이지의 실행을 비활성화할 수 있으므로 결과적으로 생성된 실행 파일은 더 커지고 더 안전하게 실행할 수 있게 되었습니다.
- 바이너리 파일에 관한 동적 로더에 힌트를 제공하는 GNU Property 노트 지원이 추가되었습니다.
- 이전에는 링커에서 Intel IBT(Indirect Branch Tracking) 기술의 잘못된 실행 코드를 생성했기 때문에 생성된 실행 파일을 시작하지 못했습니다. 이 버그가 수정되었습니다.
- 이전에는 **gold** 링커에서 부적절하게 속성 노트를 병합했습니다. 따라서 생성된 코드에서 잘못된 하드웨어 기능이 활성화되고 코드가 예기치 않게 종료될 수 있었습니다. 이 버그가 수정되었습니다.
- 이전에는 **gold** 링커에서 아키텍처에 따라 정렬을 수행하기 위해 끝에 패딩 바이트가 있는 노트 섹션을 생성했습니다. 동적 로더가 패딩을 예상하지 않았기 때문에 로드 중인 프로그램이 예기치 않게 종료될 수 있었습니다. 이 버그가 수정되었습니다.

기타 툴:

- **readelf** 및 **objdump** 툴에는 별도의 디버그 정보 파일로 연결된 링크를 따라 이동하여 해당 정보를 표시하는 옵션이 추가되었습니다.
- 새로운 **--inlines** 옵션은 **objdump** 툴의 기존 **--line-numbers** 옵션을 확장하여 인라인 함수의 중첩 정보를 표시합니다.
- **nm** 툴에는 기호의 이름 뒤에 해당 버전 정보를 표시하는 새로운 옵션 **--with-version-strings**가 추가되었습니다.

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

Performance co-pilot 버전 4.1.3

Red Hat Enterprise Linux 8은 PCP(Performance co-pilot) 버전 4.1.3과 함께 배포됩니다. 주요 개선사항은 다음과 같습니다.

- **pcp-dstat** 툴에는 이제 기록 분석과 쉽표로 구분된 값(CSV) 형식 출력이 추가되어 있습니다.
- 로그 유틸리티에서 메트릭 레이블과 도움말 텍스트 레코드를 사용할 수 있습니다.
- **pmdaperfevent** 툴에서는 낮은 SMT(Simultaneous Multi Threading) 수준에서 올바른 CPU 번호를 보고합니다.
- **pmdapostgresql** 툴에서는 **Postgres** 시리즈 10.x를 지원합니다.
- **pmdaredis** 툴에서는 **Redis** 시리즈 5.x를 지원합니다.
- **pmdabcc** 툴은 동적 프로세스 필터링 및 프로세스별 syscalls, ucalls 및 ustat로 향상되었습니다.
- **pmdammv** 툴에서는 메트릭 레이블을 내보내낼 수 있으며 형식 버전은 3으로 증가했습니다.
- **pmdagfs2** 툴에서는 추가 glock과 glock 홀더 메크릭을 지원합니다.
- SELinux 정책과 관련하여 몇 가지 항목이 수정되었습니다.

(BZ#1641034)

메모리 보호 키

이 업데이트를 사용하면 스레드마다 페이지 보호 플래그를 변경할 수 있는 하드웨어 기능을 사용할 수 있습니다. **pkey_alloc()**, **pkey_free()** 및 **pkey_mprotect()** 함수의 새로운 **glibc** 시스템 호출 래퍼가 추가되었습니다. 또한 스레드별 보호 플래그에 액세스할 수 있도록 **pkey_set()** 및 **pkey_get()** 함수도 추가되었습니다.

(BZ#1304448)

elfutils가 버전 0.174로 업데이트

Red Hat Enterprise Linux 8에서 **elfutils** 패키지는 버전 0.174에서 사용 가능합니다. 주요 변경 사항은 다음과 같습니다.

- 이전에는 **eu-readelf** 툴에서 값이 음수인 변수를 부호 없는 큰 값 처럼 표시하거나 부호 없는 큰 값을 음수 값으로 표시했습니다. 이 문제는 수정되어 **eu-readelf**에서 값을 올바르게 표시하기 위해 상수 값 형태의 크기와 부호 특성을 확인하여 올바르게 표시할 수 있게 되었습니다.
- CU가 없는 **.debug_line** 데이터를 읽는 데 사용하는 새로운 함수 **dwarf_next_lines()**가 **libdw** 라이브러리에 추가되었습니다. 이 함수는 **dwarf_getsrclines()** 및 **dwarf_getsrcfiles()** 함수 대신 사용할 수 있습니다.
- 이전에는 섹션이 65280개가 넘는 파일이 있으면 **libelf** 및 **libdw** 라이브러리와 해당 라이브러리를 사용하는 모든 툴에 오류가 발생했습니다. 이 버그는 수정되어 결과적으로 ELF 파일 헤더에서 확장된 **shnum** 및 **shstrndx** 값이 올바르게 처리됩니다.

(BZ#1641007)

Valgrind가 3.14 버전으로 업데이트

Red Hat Enterprise Linux 8은 Valgrind 실행 코드 분석 툴 버전 3.14와 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 로드되지 않은 코드에 관한 디버그 정보를 보존할 수 있도록 새로운 **--keep-debuginfo** 옵션이 추가되었습니다. 결과적으로 저장된 스택 추적에는 더 이상 메모리에 없는 코드의 파일 및 행 정보가 포함될 수 있습니다.
- 소스 파일 이름 및 행 번호를 기반으로 한 비활성화가 추가되었습니다.
- **Helgrind** 툴은 전체 히스토리 스택 추적의 계산을 지정하기 위해 **--delta-stacktrace** 옵션으로 확장되었습니다. **--history-level=full**과 함께 이 옵션을 사용하면 **Helgrind** 성능이 최대 25% 향상될 수 있습니다.
- Intel과 AMD 64비트 아키텍처 및 ARM 64비트 아키텍처에서 최적화된 코드의 **Memcheck** 툴에서 오검출 비율이 감소되었습니다. **--expensive-definedness-checks**를 사용하여 정의 검사 처리를 제어하여 비율을 향상시킬 수 있지만 성능은 저하됩니다.
- Valgrind에서는 IBM Power Systems의 little-endian 변형의 명령어를 더 많이 인식할 수 있습니다.
- Valgrind에서는 IBM Z 아키텍처 z13 프로세서의 정수 및 문자열 벡터 명령어를 부분적으로 처리할 수 있습니다.

새로운 옵션과 알려진 제한사항에 관한 자세한 내용은 **valgrind(1)** 매뉴얼 페이지를 참조하십시오.

(BZ#1641029, BZ#1501419)

GDB 버전 8.2

Red Hat Enterprise Linux 8은 GDB 디버거 버전 8.2와 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- GDB 및 **gdbserver**에서 원격 디버깅을 수행하는 데 IPv6 프로토콜이 지원됩니다.
- 디버그 정보 없이 디버깅 기능이 향상되었습니다.
- GDB 사용자 인터페이스의 기호는 ABI 태그나 네임 스페이스와 같은 구문 구조를 사용하여 더 나은 제안사항을 제공하도록 개선되었습니다.
- 이제 백그라운드에서 명령을 실행할 수 있습니다.
- Rust 프로그래밍 언어로 작성된 디버깅 프로그램을 사용할 수 있습니다.
- **_Alignof** 및 **alignof** 연산자, C++ rvalue 참조 및 C99 변수-길이 자동 배열을 지원하는 구문 분석기를 통해 C와 C++ 언어 디버깅이 향상되었습니다.
- GDB 확장 스크립트는 이제 Guile 스크립팅 언어를 사용할 수 있습니다.
- 확장을 위한 Python 스크립팅 언어 인터페이스가 새로운 API 함수, 프레임 데코레이터, 필터 및 unwinder로 향상되었습니다. 또한 GDB 구성의 **.debug_gdb_scripts** 섹션에 있는 스크립트가 자동으로 로드됩니다.
- GDB에서는 Python 버전 3을 사용하여 프린터, 프레임 데코레이터, 필터 및 unwinder 등의 스크립트를 실행합니다.
- ARM 및 64비트 ARM 아키텍처는 Thumb 32비트 및 시스템 호출 명령어를 포함하여 프로세스 실행 레코드 및 재생이 추가되었습니다.
- Intel MPX 레지스터 및 바인딩 위반, PKU 레지스터 및 Intel Processor Trace에 대한 지원이 추가되었습니다.
- Intel 기반 시스템에서 **rdrand** 및 **rdseed** 명령어를 포함하도록 레코드 및 재생 기능이 확장되었습니다.
- IBM Z 아키텍처에서 GDB의 기능은 추적점과 빠른 추적점, 벡터 레지스터와 ABI, **Catch** 시스템 호출을 지원하도록 확장되었습니다. 또한 GDB는 아키텍처에 대한 최신 명령어를 지원합니다.
- GDB는 64bit ARM 아키텍처에서 SystemTap 정적 사용자 공간 프로브(SDT)를 사용할 수 있습니다.

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

RHEL의 현지화는 여러 패키지로 배포

RHEL 8에서는 단일 **glibc-common** 패키지로 로케일과 번역을 더 이상 제공하지 않습니다. 대신, 모든 로케일과 언어는 **glibc-langpack-CODE** 패키지에서 사용 가능합니다. 또한 모든 로케일이 기본적으로 설치되는 것은 아니며 설치 프로그램에서 선택한 로케일만 설치됩니다. 사용자는 필요한 추가 로케일 패키지를 별도로 설치해야 합니다.

langpacks 사용에 대한 자세한 내용은 [langpacks 설치 및 사용](#) 을 참조하십시오.

(BZ#1512009)

strace는 버전 4.24로 업데이트

Red Hat Enterprise Linux 8은 **strace** 툴 버전 4.24와 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 시스템 호출 조작 기능이 **-e inject=** 옵션으로 추가되었습니다. 여기에는 오류, 반환 값, 지연 및 신호 삽입이 포함됩니다.

- 시스템 호출 자격이 개선되었습니다.
 - 정규 표현식을 사용하여 시스템 호출을 필터링하기 위해 **-e trace=/regex** 옵션이 추가되었습니다.
 - 자격이 시스템 호출과 일치하지 않는 경우에도 **-e trace=** 옵션에서 시스템 호출 자격 앞에 물음표를 추가하면 **strace**가 계속됩니다.
 - **-e trace** 옵션의 시스템 호출 자격에 개성 (Personality) 지정이 추가되었습니다.
- **kvm vcpu** 종료 이유의 디코딩이 추가되었습니다. 여기에는 **-e kvm=vcpu** 옵션을 사용합니다.
- **-k** 옵션 사용 시 이제 **libdw** 라이브러리가 스택 풀기에 사용됩니다. 또한 **libiberty** 라이브러리가 시스템에 설치되어 있을 때 기호 디맹글링(demangling)이 가능합니다.
- 이전에는 **-r** 옵션을 지정하면 **strace**가 **-t** 옵션을 무시했습니다. 이 문제가 해결되어 현재 두 옵션은 서로 간섭받지 않게 되었습니다.
- 침부 모드에서 출력 파일을 여는 경우 [option]`-A 옵션이 추가되었습니다.
- **xlat** 출력 형식을 구성하기 위해 **-X** 옵션이 추가되었습니다.
- **-yy** 옵션으로 소켓 주소를 디코딩하는 기능이 향상되었습니다. 또한 **-yy** 모드에서 블록 및 문자 장치 번호 출력이 추가되었습니다.

또한 다음 항목에 대한 디코딩이 추가, 개선, 업데이트되었습니다.

- **netlink** 프로토콜, 메시지 및 속성
- **arch_prctl, bpf, getsockopt, io_pgetevent, kern_features, keyctl, prctl, pkey_alloc, pkey_free, pkey_mprotect, ptrace, rseq, setsockopt, socket, statx** 및 기타 시스템 호출
- **ioctl** 시스템 호출을 위한 다수의 명령
- 다양한 유형의 상수
- 간접 인수가 있는 **execveat, inotify_add_watch, inotify_init, select, symlink, symlinkat** 시스템 호출 및 **mmap** 시스템 호출 경로 추적
- ARM 아키텍처 **__ARM_NR_***과 관련된 시스템 호출
- 신호 코드 목록

(BZ#1641014)

4.11. 파일 시스템 및 스토리지

XFS는 공유 COW(Copy-On-Write) 데이터 Extent 기능 지원

XFS 파일 시스템은 공유 copy-on-write(COW) 데이터 Extent 기능을 지원합니다. 이 기능을 사용하면 두 개 이상의 파일이 공통 데이터 블록 집합을 공유할 수 있습니다. 공통 블록을 공유하는 파일 중 하나가 변경되면 XFS는 링크를 공통 블록으로 분리하고 새 파일을 만듭니다. 이는 다른 파일 시스템에 있는 COW(Copy-On-Write) 기능과 유사합니다.

공유 COW(Copy-On-Write) 데이터 Extent 특징은 다음과 같습니다.

신속 (Fast)

공유 복사본 생성 시 디스크 I/O를 사용하지 않습니다.

공간 효율적 (Space-efficient)

공유 블록이 추가 디스크 공간을 소비하지 않습니다.

투명 (Transparent)

공통 블록을 공유하는 파일은 일반 파일과 동일하게 작동합니다.

사용자 공간 유틸리티는 다음과 같은 목적으로 공유 COW(Copy-On-Write) 데이터 Extent를 사용할 수 있습니다.

- 효율적인 파일 복제(예: `cp --reflink` 명령)
- 파일 별 스냅샷

이 기능은 보다 더욱 효율적인 운영을 위해 Overlayfs 및 NFS와 같은 커널 하위 시스템에서도 사용됩니다.

공유 COW(Copy-On-Write) 데이터 Extent는 **xfsprogs** 패키지 버전 **4.17.0-2.el8** 이상에서 XFS 파일 시스템을 만들 때 기본적으로 활성화됩니다.

현재 Direct Access(DAX) 장치는 공유 COW(Copy-On-Write) 데이터 Extent가 있는 XFS를 지원하지 않습니다. 이 기능을 사용하지 않고 XFS 파일 시스템을 만들려면 다음 명령을 사용하십시오.

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

Red Hat Enterprise Linux 7은 읽기 전용 모드에서만 공유 COW(Copy-On-Write) 데이터 Extent로 XFS 파일 시스템을 마운트할 수 있습니다.

(BZ#1494028)

XFS 파일 시스템 최대 크기는 1024TiB

XFS 파일 시스템의 최대 지원 크기가 500TiB에서 1024TiB로 증가했습니다.

500TiB 이상의 파일 시스템에는 다음이 필요합니다.

- 메타데이터 CRC 기능과 free inode btree 기능은 모두 파일 시스템 형식으로 활성화되어야 합니다.
- 할당 그룹 크기는 최소 512GiB입니다.

RHEL 8에서 **mkfs.xfs** 유틸리티는 기본적으로 이러한 요구 사항을 충족하는 파일 시스템을 생성합니다.

이러한 요구 사항을 충족하지 않는 소규모 파일 시스템을 500TiB 이상의 새로운 크기로 확장하는 기능은 지원되지 않습니다.

(BZ#1563617)

모든 아키텍처를 지원하는 VDO

VDO(Virtual Data Optimizer)는 RHEL 8이 지원하는 모든 아키텍처에서 사용할 수 있습니다.

지원되는 아키텍처 목록은 [2장. 아키텍처](#)를 참조하십시오.

(BZ#1534087)

BOOM 부팅 관리자가 부팅 항목을 생성하는 과정을 단순화

BOOM은 부팅 항목 설정을 위해 BootLoader Specification을 지원하는 부팅 로더를 사용하는 Linux 시스템용 부팅 관리자입니다. 유연한 부팅 설정이 가능하며 LVM을 사용하여 생성된 시스템의 스냅샷 이미지를 부팅하는 것과 같이 새롭거나 수정된 부팅 항목을 간단하게 만들 수 있습니다.

BOOM은 기존 부팅 로더 구성을 변경하지 않고 추가 항목만 삽입합니다. 기존 설정이 유지되며 커널 설치 및 업데이트 스크립트와 같은 배포 통합은 이전처럼 계속 작동합니다.

BOOM에는 부팅 항목을 쉽게 생성할 수 있는 CLI(Command-Line Interface) 및 API가 있습니다.

(BZ#1649582)

LUKS2는 볼륨 암호화를 위한 기본 형식

RHEL 8에서는 LUKS 버전 2(LUKS2) 형식이 기존 LUKS(LUKS1) 형식 대신 사용됩니다. 이제 **dm-crypt** 하위 시스템과 **cryptsetup** 툴은 LUKS2를 암호화된 볼륨의 기본 형식으로 사용합니다. LUKS2는 부분적인 메타데이터 손상 이벤트가 발생했을 경우에 대비하여 메타데이터 중복 및 자동 복구 기능을 갖춘 암호화된 볼륨을 제공합니다.

LUKS2는 내부의 유연한 레이아웃을 사용하여 향후 제공되는 기능을 사용할 수 있습니다.

libcryptsetup에 내장된 일반 커널-키링 토큰을 통해 자동 잠금 해제 기능을 지원하며, 사용자는 커널 키링 보존 서비스에 저장된 암호를 사용하여 LUKS2 볼륨의 잠금을 해제할 수 있습니다.

기타 주요 기능 개선 사항:

- 랩된 키 암호화 스키마 사용하여 보안키 설정
- 정책 기반 암호 해독(Clevis)과 보다 쉽게 통합
- 최대 32개의 키 슬롯 - LUKS1은 8개의 키 슬롯만 제공

자세한 내용은 **cryptsetup(8)** 및 **cryptsetup-reencrypt(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1564540)

Broadcom Emulex Fibre Channel 어댑터에서 NVMe/FC 완전 지원

Broadcom Emulex Fibre Channel 32Gbit 어댑터와 함께 사용하는 경우 개시자 모드에서 NVMe/FC(NVMe over Fibre Channel) 전송 유형이 완전히 지원됩니다.

NVMe/FC(NVMe over Fibre Channel)는 Red Hat Enterprise Linux에서 이전에 도입된 RDMA(Remote Direct Memory Access) 프로토콜 외에도 NVMe(Nonvolatile Memory Express) 프로토콜을 위한 추가적인 패브릭 전송 유형입니다.

lpfc 드라이버에서 NVMe/FC를 활성화하려면 **/etc/modprobe.d/lpfc.conf** 파일을 편집하고 다음 옵션을 추가합니다.

```
lpfc_enable_fc4_type=3
```

/etc/modprobe.d/lpfc.conf 이외의 드라이버는 기술 프리뷰에 남아있습니다.

추가 제한 사항:

- NVMe/FC에서 멀티패스가 지원되지 않습니다.
- NVMe/FC에서 NVMe 클러스터링이 지원되지 않습니다.
- 현재 Red Hat Enterprise Linux는 개시자 포트에서 NVMe/FC 및 SCSI/FC의 동시 사용을 지원하지 않습니다.

- `kernel-alt` 패키지는 NVMe/FC를 지원하지 않습니다.
- NVMe/FC에서 `kdump`가 지원되지 않습니다.
- SAN(Storage Area Network)의 NVMe/FC에서 시스템 부팅이 지원되지 않습니다.

(BZ#1649497)

DM Multipath 설정 파일의 새로운 `overrides` 섹션 추가

`/etc/multipath.conf` 파일에는 모든 장치에 대한 구성 값을 설정할 수 있는 `overrides` 섹션이 포함되어 있습니다. DM Multipath는 장치를 포함하는 경로의 `/etc/multipath.conf` 파일에 있는 `multipaths` 섹션에서 지정한 속성으로 덮어쓰지 않는 한 모든 장치에 대해 이러한 속성을 사용합니다. 이 기능은 더 이상 지원되지 않는 구성 파일의 `devices` 섹션에 있는 `all_devs` 매개변수를 대체합니다.

(BZ#1643294)

NVDIMM 장치의 설치 및 부팅 지원

이전에는 설치 프로그램이 모든 모드에서 NVDIMM(Nonvolatile Dual Inline Memory Module) 장치를 무시했습니다.

이번 업데이트를 통해 NVDIMM 장치를 지원하도록 커널이 개선되어 향상된 시스템 성능, 데이터베이스 또는 분석 워크로드와 같은 쓰기 집약적인 애플리케이션에 대한 파일 시스템 액세스가 강화되어 CPU 오버헤드가 감소됩니다.

이번 업데이트를 통해 다음을 지원할 수 있습니다.

- `nvdimm` Kickstart 명령과 GUI를 사용하여 설치 시 NVDIMM 장치를 사용하면 설치 중에 섹터 모드에서 NVDIMM 장치를 설치 및 부팅하고 NVDIMM 장치를 섹터 모드로 재설정할 수 있습니다.
- NVDIMM 장치를 다루는 명령은 `Anaconda`에 대한 `Kickstart` 스크립트를 확장합니다.
- `grub2`, `efibootmgr` 및 `efivar` 시스템 구성 요소로 NVDIMM 장치를 처리 및 부팅할 수 있습니다.

(BZ#1499442)

DM Multipath의 경계 경로 검색 개선

`multipathd` 서비스의 경계 경로 검색이 개선되었습니다. 이는 멀티패스 장치가 반복적으로 실패할 가능성이 있는 경로를 피하고 성능을 향상시킬 수 있습니다. 경계 경로는 지속적이지만 일시적인 I/O 오류가 있는 경로입니다.

`/etc/multipath.conf` 파일의 다음 옵션은 경계 경로 동작을 제어합니다.

- `marginal_path_double_failed_time`
- `marginal_path_err_sample_time`
- `marginal_path_err_rate_threshold`
- `marginal_path_err_recheck_gap_time`

DM Multipath는 다음과 같은 경우 경로를 비활성화하고 구성된 샘플 기간 동안 I/O를 반복 테스트합니다.

- 나열된 `multipath.conf` 옵션으로 설정된 경우
- 구성된 기간 내에 경로가 두 번 실패한 경우

- 기타 경로를 사용할 수 있는 경우

테스트 중에 경로에 구성된 오류 비율보다 큰 오류 비율이 있는 경우 DM Multipath는 설정된 간격 시간 동안 이 비율을 무시한 다음 복원할 수 있을 정도로 제대로 작동하고 있는지 여부를 다시 확인합니다.

자세한 내용은 **multipath.conf** 매뉴얼 페이지를 참조하십시오.

(BZ#1643550)

Multi-Queue 기본 동작

Red Hat Enterprise Linux 8에서 블록 장치는 Multi-Queue 스케줄링을 사용합니다. 이를 통해 솔리드 스테이트 드라이브(SSD) 및 멀티 코어 시스템으로 블록 계층 성능을 확장할 수 있습니다.

SCSI Multiqueue (**scsi-mq**) 드라이버가 기본적으로 활성화되고 커널은 **scsi_mod.use_blk_mq=Y** 옵션으로 부팅됩니다. 이러한 변화는 업스트림 Linux 커널과 일치합니다.

DM Multipath(Device Mapper Multipath)를 활성화하려면 **scsi-mq** 드라이버가 필요합니다.

(BZ#1647612)

Stratis 사용 가능

Stratis는 새로운 로컬 스토리지 관리자이며 추가 기능을 사용하여 사용자에게 스토리지 풀 상단에서 관리되는 파일 시스템을 제공합니다.

Stratis를 사용하면 다음과 같은 스토리지 작업을 보다 쉽게 수행할 수 있습니다.

- 스냅샷 및 썬 프로비저닝 관리
- 필요에 따라 파일 시스템 크기 자동 확장
- 파일 시스템 관리

Stratis 스토리지를 관리하려면 **stratisd** 백그라운드 서비스와 통신하는 **stratis** 유틸리티를 사용합니다.

자세한 내용은 Stratis 설명서 [Managing layered local storage with Stratis](#) 를 참조하십시오.

(JIRA:RHELPLAN-1212)

4.12. 고가용성 및 클러스터

새로운 pcs 명령으로 사용 가능한 위치독 장치 나열 및 테스트

Pacemaker로 SBD를 구성하려면 위치독 장치가 작동해야 합니다. 이 릴리스에서는 로컬 노드에서 사용할 수 있는 위치독 장치를 나열하기 위해 **pcs stonith sbd watchdog list** 명령을 지원하며 위치독 장치를 테스트하기 위해 **pcs stonith sbd watchdog test** 명령을 지원합니다. **sbd** 명령행 틀에 대한 자세한 내용은 **sbd(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1578891)

pcs 명령으로 작업 및 해당 간격 기준으로 리소스 오류 필터링 지원

Pacemaker는 리소스 이름과 노드 상단에서 리소스 작업별로 리소스 오류를 추적합니다. **pcs resource failcount show** 명령은 리소스, 노드, 작업, 간격별로 오류를 필터링할 수 있으며 리소스 및 노드별로 수집된 오류를 표시하거나 리소스, 노드, 작업, 간격별로 상세한 오류를 표시할 수 있는 옵션을 제공합니다. 또한 **pcs resource failcount reset** 명령을 사용하여 리소스, 노드, 작업, 간격별로 오류를 필터링할 수 있습니다.

(BZ#1591308)

corosync 로그에서 타임스탬프 활성화

이전에는 **corosync** 로그에 타임스탬프가 없었으므로 다른 노드 및 데몬의 로그에 연결하기가 어려웠습니다. 이 릴리스에서는 **corosync** 로그에 타임스탬프가 기록됩니다.

(BZ#1615420)

pcs cluster setup, pcs cluster node add 및 pcs cluster node remove 명령의 새로운 형식

Red Hat Enterprise Linux 8에서 **pcs**는 Corosync 3, **knet** 및 노드 이름을 완벽하게 지원합니다. 노드 이름은 필수 사항이며 노드 식별자에서 노드 주소를 교체합니다. 노드 주소는 임의로 제공됩니다.

- **pcs host auth** 명령에서 노드 주소가 노드 이름으로 기본 설정됩니다.
- **pcs cluster setup** 명령 및 **pcs cluster node add** 명령에서 노드 주소가 **pcs host auth** 명령에 지정된 노드 주소로 기본 설정됩니다.

이러한 변경으로 인해 클러스터 설정, 클러스터에 노드 추가, 클러스터에서 노드 제거를 위한 명령 형식이 변경되었습니다. 이러한 새로운 명령 형식에 대한 자세한 내용은 **pcs cluster setup, pcs cluster node add** 및 **pcs cluster node remove** 명령의 도움말 보기를 참조하십시오.

(BZ#1158816)

RHEL 8에서 Pacemaker 2.0.0

pacemaker 패키지가 Pacemaker 2.0.0 업스트림 버전으로 업그레이드되었으며 이전 버전에 비해 여러 버그 수정 및 개선 사항을 제공합니다.

- Pacemaker 세부 로그는 기본적으로 **/var/log/pacemaker/pacemaker.log**가 됩니다 (**/var/log/cluster** 아래에서 **/var/log**에 직접 표시되거나 **corosync**와 결합되지 않음).
- Pacemaker 데몬 프로세스의 이름이 로그를 더욱 직관적으로 읽을 수 있도록 변경되었습니다. 예를 들어, **pengine**은 **pacemaker-schedulerd**로 이름이 변경되었습니다.
- 더 이상 사용하지 않는 **default-resource-stickiness** 및 **is-managed-default** 클러스터 속성에 대한 지원이 중단되었습니다. 대신 **resource-stickiness** 및 **is-managed** 속성을 리소스 기본값으로 설정해야 합니다. 더 이상 사용하지 않는 구문이 있는 기존 구성(새롭게 생성된 구문이 아님)은 지원되는 구문을 사용하도록 자동 업데이트됩니다.
- 변경 사항 전체 목록은 <https://access.redhat.com/articles/3681151>에서 확인할 수 있습니다.

Red Hat Enterprise Linux 7 또는 이전 버전을 사용하여 기존 클러스터를 업그레이드하는 사용자는 모든 클러스터 노드에서 RHEL을 업그레이드하기 전과 후에 모든 클러스터 노드에서 **pcs cluster cib-upgrade**를 실행하는 것이 좋습니다.

(BZ#1543494)

마스터 리소스의 이름이 승격 가능한 복제 리소스로 변경

RHEL(Red Hat Enterprise Linux) 8은 Pacemaker 2.0을 지원합니다. 마스터/슬레이브 리소스는 더 이상 별도의 리소스 유형이 아니지만 **promotable** 메타 속성이 **true**로 설정된 표준 복제 리소스입니다. 이번 업데이트에는 다음과 같은 변경 사항이 구현되었습니다.

- **pcs** 명령으로 더 이상 마스터 리소스를 생성할 수 없습니다. 대신 **promotable** 복제 리소스를 생성할 수 있습니다. 관련 키워드와 명령이 **master**에서 **promotable**로 변경되었습니다.
- 기존의 모든 마스터 리소스가 승격 가능한 복제 리소스로 표시됩니다.

- RHEL7 클러스터는 승격 가능한 복제를 지원하지 않으므로 웹 UI에서 RHEL7 클러스터를 관리할 때 마스터 리소스를 계속 마스터라고 부릅니다.

(BZ#1542288)

클러스터에서 노드를 인증하기 위한 새로운 명령

RHEL(Red Hat Enterprise Linux) 8에서 클러스터 노드 인증에 사용되는 명령에 다음과 같은 변경 사항이 적용됩니다.

- 인증을 위한 새로운 명령은 **pcs host auth**입니다. 이 명령으로 사용자는 호스트 이름, 주소 및 **pcsd** 포트를 지정할 수 있습니다.
- **pcs cluster auth** 명령은 로컬 클러스터의 노드만 인증하고 노드 목록을 허용하지 않습니다.
- 각 노드에 대해 주소를 지정할 수 있습니다. **pcs/pcsd**는 지정된 주소를 사용하여 각 노드와 통신합니다. 이러한 주소는 내부적으로 **corosync**가 사용하는 주소와 다를 수 있습니다.
- **pcs pcsd clear-auth** 명령은 **pcs pcsd deauth** 명령과 **pcs host deauth** 명령으로 교체되었습니다. 사용자는 새 명령으로 단일 호스트뿐 아니라 모든 호스트의 인증을 해제할 수 있습니다.
- 이전에는 노드 인증이 양방향이었으며 **pcs cluster auth** 명령을 실행하면 지정된 모든 노드가 서로 인증되었습니다. 그러나 **pcs host auth** 명령은 지정된 노드에 대해 로컬 호스트만 인증됩니다. 따라서 이 명령을 실행할 때 다른 노드에 대해 인증되는 노드를 보다 효과적으로 제어할 수 있습니다. 클러스터를 설정하거나 노드 추가 시 **pcs**가 클러스터의 토큰을 자동으로 동기화하므로 클러스터의 모든 노드는 이전과 마찬가지로 자동으로 인증되고 클러스터 노드는 서로 통신할 수 있습니다.

이러한 변경 사항은 이전 버전과 호환되지 않습니다. RHEL 7 시스템에서 인증된 노드는 다시 인증해야 합니다.

(BZ#1549535)

pcs 명령으로 펜싱 기록의 표시, 정리 및 동기화 지원

Pacemaker의 fence 데몬은 실행한 모든 fence 작업 내역(보류 중, 성공 및 실패)을 추적합니다. 이번 릴리스에서는 **pcs** 명령을 사용하여 사용자가 다음과 같은 방법으로 펜싱 기록에 액세스할 수 있습니다.

- **pcs status** 명령으로 실패한 펜싱 작업과 보류 중인 펜싱 작업 표시
- **pcs status --full** 명령으로 전체 펜싱 작업 표시
- **pcs stonith history** 명령으로 펜싱 기록을 표시 및 정리할 수 있는 옵션 제공
- 펜싱 기록이 자동으로 동기화되지만 **pcs stonith history** 명령은 필요에 따라 사용자가 펜싱 기록을 수동으로 동기화할 수 있는 **update** 옵션 지원

(BZ#1620190, BZ#1615891)

4.13. 네트워크 구성

nftables가 기본 네트워크 패킷 필터링 프레임워크로 iptables 대체

nftables 프레임워크는 패킷 분류 기능을 제공하며 **iptables**, **ip6tables**, **arptables**, **eatables** 툴을 대체합니다. 이전의 패킷 필터링 툴에 비해 편의성, 기능 및 성능이 크게 향상되었으며 주요 개선 사항은 다음과 같습니다.

- 선형 처리를 대신하는 룩업 테이블
- IPv4 및 IPv6 프로토콜 모두를 위한 단일 프레임워크
- 전체 규칙 세트를 가져오기, 업데이트 및 저장하는 대신 모두 정밀하게 적용되는 규칙
- 규칙 세트(nfttrace) 및 모니터링 추적 이벤트(nft 툴 내부)에서 디버깅 및 추적 지원
- 프로토콜별 확장 없이 보다 일관되고 컴팩트한 구문
- 타사 애플리케이션용 Netlink API

iptables 과 마찬가지로 **nftables** 는 체인을 저장하기 위해 테이블을 사용합니다. 체인에는 작업 수행을 위한 개별 규칙이 있습니다. **nft** 툴은 이전 패킷 필터링 프레임워크의 모든 툴을 교체합니다. **libnftables** 라이브러리는 **libmnl** 라이브러리를 통해 **nftables** Netlink API와 낮은 수준의 상호 작용에 사용될 수 있습니다.

iptables, **ip6tables**, **ebtables** 및 **arptables** 툴은 같은 이름을 가진 nftables 기반 드롭인 툴로 교체됩니다. 외부 동작은 기존 대응 동작과 일치하지만, 내부적으로는 필요에 따라 호환성 인터페이스를 통해 기존 **netfilter** 커널 모듈과 함께 **nftables** 를 사용합니다.

nftables 규칙 세트에서 모듈의 영향은 **nft list ruleset** 명령을 사용하여 확인할 수 있습니다. 이러한 툴은 테이블, 체인 및 규칙을 **nftables** 규칙 세트에 추가하기 때문에 **nft flush ruleset** 명령과 같은 **nftables** 규칙 세트 작업이 이전의 별도 기존 명령을 사용하여 설치된 규칙 세트에 영향을 줄 수 있습니다.

어떤 종류의 툴이 존재하는지 신속하게 확인하기 위해 백엔드 이름을 포함하도록 버전 정보가 업데이트되었습니다. RHEL 8에서는 nftables 기반의 **iptables** 툴로 다음 버전 문자열이 출력됩니다.

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

기존 **iptables** 툴이 있는 경우 다음 버전 정보가 출력됩니다.

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

(BZ#1644030)

RHEL 8에서 주요 TCP 기능

Red Hat Enterprise Linux 8은 고성능, 뛰어난 확장성 및 안정성을 제공하는 TCP 네트워킹 스택 버전 4.16 과 함께 배포됩니다. 특히 접속률이 높고 사용량이 많은 TCP 서버의 경우 성능이 향상됩니다.

또한 새로운 TCP 혼잡 알고리즘인 **BBR** 및 **NV** 두 가지를 사용할 수 있으므로 대부분 시나리오에서 Cubic 보다 대기 시간이 짧아지고 처리량이 향상됩니다.

(BZ#1562998)

firewalld는 기본적으로 nftables를 사용

이번 업데이트를 통해 **nftables** 필터링 하위 시스템이 **firewalld** 데몬의 기본 방화벽 백엔드가 됩니다. 백엔드를 변경하려면 **/etc/firewalld.conf** 파일에서 **FirewallBackend** 옵션을 사용하십시오.

이 변경으로 **nftables** 를 사용할 때 동작에서 다음과 같은 차이가 발생합니다.

1. **iptables** 규칙 실행이 항상 **firewalld** 규칙 이전에 발생

- **iptables**의 **DROP**은 패킷이 **firewalld**에 절대 표시되지 않음을 의미함
 - **iptables**의 **ACCEPT**는 패킷이 **firewalld** 규칙에 여전히 종속된다는 것을 의미함
2. **firewalld** 직접 규칙은 여전히 **iptables**를 통해 구현되지만 다른 **firewalld** 기능은 **nftables**를 사용함
 3. **firewalld**가 설정된 연결을 일반적으로 수락하기 전에 직접 규칙 실행이 이루어짐

(BZ#1509026)

RHEL 8 **wpa_supplicant**에서 주요 변경 사항

RHEL(Red Hat Enterprise Linux) 8에서는 **CONFIG_DEBUG_SYSLOG**가 활성화된 상태에서 **wpa_supplicant** 패키지가 구축됩니다. 이렇게 하면 **/var/log/wpa_supplicant.log** 파일의 콘텐츠를 검사하는 대신 **journalctl** 유틸리티를 사용하여 **wpa_supplicant** 로그를 읽을 수 있습니다.

(BZ#1582538)

NetworkManager가 SR-IOV 가상 함수 지원

Red Hat Enterprise Linux 8.0에서는 **NetworkManager**를 사용하여 SR-IOV (Single Root I/O virtualization)를 지원하는 인터페이스에 대해 가상 함수(VF)를 설정할 수 있습니다. 또한 **NetworkManager**는 MAC 주소, VLAN, **spoof checking** 설정 및 허용된 비트레이트와 같은 VF의 일부 속성을 설정할 수 있습니다. SR-IOV와 관련된 모든 속성은 **sriov** 연결 설정에서 사용할 수 있습니다. 자세한 내용은 **nm-settings(5)** 매뉴얼 페이지를 참조하십시오.

(BZ#1555013)

IPVLAN 가상 네트워크 드라이버 지원

Red Hat Enterprise Linux 8.0에서 커널은 IPVLAN 가상 네트워크 드라이버를 지원합니다. 이번 업데이트를 통해 IPVLAN 가상 NIC(네트워크 인터페이스 카드)는 단일 MAC 주소를 로컬 네트워크에 노출하는 여러 컨테이너에 대한 네트워크 연결을 가능하게 합니다. 이제 단일 호스트는 피어 네트워킹 장비가 지원하는 MAC 주소 개수 제한을 개선하기 위해 다수의 컨테이너를 보유할 수 있게 됩니다.

(BZ#1261167)

NetworkManager가 연결에 대한 와일드카드 인터페이스 이름 일치 지원

이전에는 인터페이스 이름의 정확한 일치만 사용하여 주어진 인터페이스에 대한 연결을 제한할 수 있었습니다. 이번 업데이트에서는 연결에 와일드카드를 지원하는 새로운 **match.interface-name** 속성이 포함되어 있습니다. 이제 사용자는 와일드카드 패턴을 사용하여 보다 유연한 방식으로 연결에 대한 인터페이스를 선택할 수 있습니다.

(BZ#1555012)

네트워킹 스택 4.18 개선 사항

Red Hat Enterprise Linux 8.0에는 업스트림 버전 4.18로 업그레이드된 네트워킹 스택이 포함되어 있으며, 일부 버그 수정 및 개선사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- **UDP_GSO**와 같은 새로운 오프로드 기능이 도입되었으며 일부 장치 드라이버의 경우 **GRO_HW**를 도입
- UDP(User Datagram Protocol) 확장성이 크게 개선
- 일반적으로 사용하는 폴링 코드 개선

- IPv6 프로토콜 확장성 개선
- 라우팅 코드 확장성 개선
- 새로운 기본 전송 큐 스케줄링 알고리즘인 **fq_codel**을 추가하여 전송 지연 개선
- 일부 전송 큐 스케줄링 알고리즘의 확장성 개선(예: **pfifo_fast**의 잠금 해제)

(BZ#1562987)

iptables를 nftables로 변환하는 새로운 툴

이번 업데이트에서는 **iptables-translate** 및 **ip6tables-translate** 툴을 추가하여 기존의 **iptables** 또는 **ip6tables** 규칙을 **nftables**에 해당하는 규칙으로 변환했습니다. 일부 확장 기능에는 해당 지원이 누락되어 있는 경우도 있습니다. 이러한 해당 기능이 누락된 확장 기능이 있는 경우 툴은 **#** 기호가 앞에 있는 미 변환 규칙을 출력합니다. 예:

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

또한 사용자는 규칙 덤프를 변환하기 위해 **iptables-restore-translate** 및 **ip6tables-restore-translate** 툴을 사용할 수 있습니다. 이전에는 **iptables-save** 또는 **ip6tables-save** 명령을 사용하여 현재 규칙 덤프를 출력할 수 있습니다. 예:

```
| % 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)

NetworkManager를 사용하여 VPN에 새 기능 추가

Red Hat Enterprise Linux 8.0의 **NetworkManager**는 VPN에 다음과 같은 새로운 기능을 제공합니다.

- IKEv2(Internet Key Exchange version 2) 프로토콜 지원
- **rightid**, **leftcert**, **narrowing**, **rekey**, **fragmentation** 옵션과 같은 다양한 **Libreswan** 옵션이 추가되었습니다. 지원되는 옵션에 대한 자세한 내용은 **nm-settings-libreswan** 매뉴얼 페이지를 참조하십시오.
- 기본 암호가 업데이트되었습니다. 즉, 사용자가 암호를 지정하지 않으면 **NetworkManager-libreswan** 플러그인은 **NetworkManager-libreswan** 애플리케이션에서 시스템 기본 암호를 선택합니다. 유일한 예외는 사용자가 IKEv1 공격적인 모드(**aggressive mode**) 설정을 선택할 때입니다. 이 경우 **ike = aes256-sha1;modp1536** and **eps = aes256-sha1** 값이 **Libreswan**에 전달됩니다.

(BZ#1557035)

새로운 데이터 체크 유형 I-DATA가 SCTP에 추가

이번 업데이트에는 SCTP(Stream Control Transmission Protocol)에 새로운 데이터 체크 유형, **I-DATA** 및 스트림 스케줄러를 추가되었습니다. 이전에는 SCTP가 사용자가 보낸 것과 동일한 순서로 사용자 메시지를 전송했습니다. 그 결과, 용량이 큰 SCTP 사용자 메시지는 완전히 전송될 때까지 스트림의 다른 모든 메시지를 차단했습니다. **I-DATA** 체크를 사용하면 TSN(Transmission Sequence Number) 필드가 오버로드되지 않으므로 SCTP는 스트림을 다른 방식으로 스케줄링할 수 있고, **I-DATA**는 사용자의 메시지 인터리브를 허용합니다(RFC 8260). 2개의 피어 모두 **I-DATA** 체크 유형을 지원해야 합니다.

(BZ#1273139)

4.14. 보안

SCAP Security Guide PCI-DSS 프로파일이 버전 3.2.1과 일치

SCAP Security Guide 프로젝트는 Red Hat Enterprise Linux 8에 대해 PCI-DSS (Payment Card Industry Data Security Standard) 프로파일을 제공하며 최신 PCI-DSS 버전 3.2.1과 일치하도록 업데이트되었습니다.

(BZ#1618528)

OpenSSH가 버전 7.8p1로 업데이트

openssh 패키지가 업스트림 버전 7.8p1로 업그레이드되었습니다. 주요 변경 사항은 다음과 같습니다.

- **SSH version 1** 프로토콜에 대한 지원 삭제
- **hmac-ripemd160** 메시지 인증 코드에 대한 지원 삭제
- **RC4(arcfour)** 암호화 지원 삭제
- **Blowfish** 암호화 지원 삭제
- **CAST** 암호화 지원 삭제
- **UseDNS** 옵션의 기본값을 **no**로 변경
- **DSA** 공개 키 알고리즘이 기본적으로 비활성화됨
- **Diffie-Hellman** 매개변수의 최소 모듈러스 크기가 2048bit로 변경됨
- **ExposeAuthInfo** 설정 옵션의 의미가 변경됨
- **UsePrivilegeSeparation=sandbox** 옵션은 필수 사항이며 비활성화할 수 없음
- 허용되는 최소의 **RSA** 키 크기를 1024bit로 설정

(BZ#1622511)

RSA-PSS가 OpenSC에서 지원

이번 업데이트에서 RSA-PSS 암호화 서명 체계에 대한 지원이 **OpenSC** 스마트 카드 드라이버에 추가되었습니다. 이러한 새로운 체계는 클라이언트 소프트웨어에서 TLS 1.3 지원에 필요한 보안 암호화 알고리즘을 사용 가능하게 합니다.

(BZ#1595626)

RHEL 8 rsyslog에서 주요 변경 사항

rsyslog 패키지가 업스트림 버전 8.37.0으로 업그레이드되었으며 이전 버전에 대해 여러 버그 수정 및 개선 사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- **rsyslog** 내부 메시지의 처리 기능 강화; 속도 제한 가능; 가능한 교착 상태 수정
- 일반적인 속도 제한 개선; 실제 스캔 출처가 기록됨

- 크기를 초과한 메시지 처리 개선 - 사용자가 별도의 작업으로 코어 및 특정 모듈에서 메시지를 처리하는 방법을 설정할 수 있음
- **mmnormalize** 규칙 기반에 대해 별도의 파일을 생성하는 대신 **config** 파일에 내장 가능
- 사용자가 **imtcp**에 대해 **GnuTLS** 우선순위 문자열을 설정하여 암호화를 세밀하게 제어 가능
- JSON의 변수를 포함한 모든 **config** 변수가 대소문자를 구분하지 않음
- PostgreSQL 출력에 다양한 개선 사항 추가
- 셸 변수를 사용하여 추가 구성 파일의 조건부 로딩, 명령 실행 또는 **config**에 텍스트 포함 등과 같은 **config** 처리를 제어할 수 있는 기능을 추가했습니다. 이 기능을 과도하게 사용하면 **rsyslog**가 있는 문제를 디버그하는 데 어려움을 겪을 수 있습니다.
- 4자리 파일 생성 모드를 **config**에 지정할 수 있음
- RELP(Reliable Event Logging Protocol) 입력을 지정된 주소에서만 바인딩할 수 있음
- 이메일 출력의 **enable.body** 옵션 기본값이 설명서에 맞게 변경됨
- 사용자가 **MongoDB** 출력에서 무시해야 하는 삽입 오류 코드를 지정할 수 있음
- 병렬 TCP(pTCP) 입력이 더욱 효과적인 로드 밸런싱에 설정 가능한 백로그를 보유함

(BZ#1613880)

새로운 rsyslog 모듈: omkafka

kafka 중앙 데이터 스토리지 시나리오를 활성화하기 위해 새로운 **omkafka** 모듈을 사용하여 로그를 **kafka** 인프라로 전송할 수 있습니다.

(BZ#1542497)

libssh는 코어 암호화 구성 요소로서 SSH를 구현

이 변경으로 인해 Red Hat Enterprise Linux 8의 코어 암호화 구성 요소인 **libssh**가 도입되었습니다. **libssh** 라이브러리가 SSH(Secure SHell) 프로토콜을 구현합니다.

libssh가 전체 시스템의 암호화 정책을 따르지 않는다는 점에 유의하십시오.

(BZ#1485241)

스마트 카드 및 HSM에 대한 PKCS #11 지원이 시스템 전반에서 일관성 유지

이번 업데이트를 통해 PKCS #11 암호화 토큰 인터페이스와 함께 스마트 카드 및 HSM(Hardware Security Module) 사용 방법이 시스템 전반에서 일관성을 유지합니다. 즉, 사용자와 관리자는 시스템의 모든 관련 톨에 대해 동일한 구문을 사용할 수 있습니다. 주요 개선 사항은 다음과 같습니다.

- 관리자와 애플리케이션 작성자 모두를 위해 RHEL 서버에서 토큰 활성화를 단순화하는 PKCS #11 URI(Uniform Resource Identifier) 체계 지원
- 전체 시스템에서 시스템 카드 및 HSM 등록에 **pkcs11.conf**를 사용
- NSS, GnuTLS 및 OpenSSL(**openssl-pkcs11** 엔진을 통해) 애플리케이션에서 HSM 및 스마트 카드에 대한 일관된 지원 가능
- Apache HTTP 서버(**httpd**)가 HSM을 완벽하게 지원

자세한 내용은 **pkcs11.conf(5)** 매뉴얼 페이지를 참조하십시오.

(BZ#1516741)

전체 시스템의 암호화 정책이 기본적으로 적용

암호화 정책은 TLS, IPSec, SSH, DNSSEC 및 Kerberos 프로토콜을 다루는 코어 암호화 하위 시스템을 구성하는 Red Hat Enterprise Linux 8의 구성 요소입니다. 이는 관리자가 **update-crypto-policies** 명령을 사용하여 선택할 수 있는 소규모 정책 세트를 제공합니다.

DEFAULT 전체 시스템의 암호화 정책은 현재의 위협 모델에 대한 보안 설정을 제공합니다. 이 보안 설정은 TLS 1.2 및 1.3 프로토콜과 IKEv2 및 SSH2 프로토콜을 허용합니다. RSA 키와 Diffie-Hellman 매개변수는 2047bit보다 큰 경우 허용됩니다.

자세한 내용은 Red Hat 블로그의 [Consistent security by crypto policies in Red Hat Enterprise Linux 8](#) 문서 및 **update-crypto-policies(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1591620)

SCAP Security Guide가 OSPP 4.2 지원

SCAP Security Guide는 Red Hat Enterprise Linux 8에 대해 OSPP(Protection Profile for General Purpose Operating Systems) 프로파일 버전 4.2 초안을 제공합니다. 이 프로파일은 NIAP Configuration Annex에 포함되어 있는 필수 구성 제어를 범용 운영 체제(Protection Profile Version 4.2)용 보호 프로파일에 반영합니다. SCAP Security Guide는 사용자가 OSPP에 정의된 요구 사항을 충족할 수 있도록 자동 검사 및 스크립트를 제공합니다.

(BZ#1618518)

OpenSCAP 명령줄 인터페이스 개선

상세 모드를 모든 **oscap** 모듈과 하위 모듈에서 사용할 수 있습니다. 툴 출력의 포맷이 개선되었습니다.

명령줄 인터페이스의 유용성이 개선되어 더 이상 사용되지 않는 옵션이 제거되었습니다.

다음 옵션은 더 이상 사용할 수 없습니다.

- **oscap xccdf generate report**의 **--show**가 완전히 삭제됨
- **oscap oval eval**의 **--probe-root**가 완전히 삭제됨. 환경 변수인 **OSCAP_PROBE_ROOT**를 설정하여 교체 가능.
- **oscap xccdf eval**의 **--sce-results**가 **--check-engine-results**로 교체됨
- CPE, OVAL 및 XCCDF 모듈에서 **validate-xml** 하위 모듈이 삭제됨. 대신 **validate** 하위 모듈을 사용하여 XML 스키마 및 XSD schematron에 대해 SCAP 콘텐츠의 유효성을 검증할 수 있음
- **oscap oval list-probes** 명령이 제거되었으므로 대신 **oscap --version**을 사용하여 사용 가능한 프로브 목록을 표시할 수 있음

OpenSCAP은 **--profile '(all)'**을 사용하여 프로파일과 관계없이 주어진 XCCDF 벤치마크의 모든 규칙을 평가할 수 있습니다.

(BZ#1618484)

mmap syscall에서 새로운 맵 권한 검사 지원

SELinux **map** 권한이 추가되어 파일, 디렉터리, 소켓 등에 대한 메모리 매핑 액세스를 제어할 수 있습니다. 이를 통해 SELinux 정책은 다양한 파일 시스템 개체에 대한 직접 메모리 액세스를 방지하고 이러한 모든 액세스를 재검증합니다.

(BZ#1592244)

SELinux가 systemd No New Privileges 지원

이번 업데이트에서는 이전 컨텍스트와 신규 컨텍스트 간에 **nnp_nosuid_transition**이 허용된 경우 **nnp_nosuid_transition** 정책 기능을 도입하여 **No New Privileges (NNP)** 또는 **nosuid**에서 SELinux 도메인 전환을 가능하게 합니다. 이제 **selinux-policy** 패키지에는 **NNP** 보안 기능을 사용하는 **systemd** 서비스를 위한 정책이 포함되어 있습니다.

다음 규칙은 이 기능을 서비스에 허용하는 방법에 대해 설명합니다.

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

예:

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

배포 정책에 m4 매크로 인터페이스가 포함되어 **init_nnp_daemon_domain()** 함수를 사용하는 서비스의 SELinux 보안 정책에서 사용할 수 있습니다.

(BZ#1594111)

SELinux가 process 클래스에서 getrlimit 권한 지원

이번 업데이트에는 새로운 SELinux 액세스 제어 검사인 **process:getrlimit**를 도입하여 **prlimit()** 함수에 추가되었습니다. 이것은 SELinux 정책 개발자가 어떤 프로세스가 **process:setrlimit** 권한을 사용하여 다른 프로세스의 리소스 제한을 읽고 수정하려고 할 때 제어할 수 있게 합니다. SELinux는 프로세스에서 **prlimit()**를 통해 자체 리소스 제한을 처리하는 작업을 제한하지 않습니다. 자세한 내용은 **prlimit(2)** 및 **getrlimit(2)** 매뉴얼 페이지를 참조하십시오.

(BZ#1549772)

암호화 라이브러리에서 TLS 1.3 지원

이번 업데이트에는 모든 주요 백엔드 암호화 라이브러리에서 기본적으로 TLS(Transport Layer Security) 1.3을 사용하도록 설정되어 있습니다. 이를 통해 운영 체제 통신 계층 전반에서 낮은 대기 시간을 단축하고 RSA-PSS 또는 X25519와 같은 새로운 알고리즘을 사용하여 애플리케이션의 개인정보 보호 및 보안을 강화합니다.

(BZ#1516728)

RHEL 8 OpenSCAP의 새로운 기능

OpenSCAP 제품군이 업스트림 버전 1.3.0으로 업그레이드되어 이전 버전에 비해 다양한 개선 사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- API 및 ABI가 통합 및 업데이트되었으며, 더 이상 사용하지 않는 기호가 삭제됨
- 프로브가 별도의 프로세스로 실행되지 않고 **oscap** 프로세스 내에서 스레드로 실행됨
- 명령줄 인터페이스가 업데이트됨
- **Python 2** 바인딩이 **Python 3** 바인딩으로 교체됨

(BZ#1614273)

Audit 3.0에서 audispd를 auditd로 교체

이번 업데이트에서 **audispd** 기능이 **auditd**로 변경되었습니다. 결과적으로 **audispd** 설정 옵션이 이제 **auditd.conf**의 일부가 되었습니다. 또한 **plugins.d** 디렉터리가 **/etc/audit**로 이동했습니다. **service auditd state** 명령을 실행하여 **auditd**의 현재 상태와 해당 플러그인을 확인할 수 있습니다.

(BZ#1616428)

rsyslog imfile이 symlinks 지원

이번 업데이트에서는 더 나은 성능과 구성 옵션을 갖춘 **rsyslog imfile** 모듈이 제공됩니다. 이제 더 복잡한 파일 모니터링 사용 사례에 모듈을 사용할 수 있습니다. 예를 들어, 구성된 경로를 따라 어디에서나 Glob 패턴이 있는 파일 모니터를 사용할 수 있으며 데이터 처리량이 증가하는 경우 symlink 대상을 순환시킬 수 있습니다.

(BZ#1614179)

자동 OpenSSH 서버 키 생성을 sshd-keygen@.service에서 처리

OpenSSH는 RSA, ECDSA 및 ED25519 서버 호스트 키가 누락된 경우 자동으로 생성합니다. RHEL 8에서 호스트 키 생성을 구성하려면 **sshd-keygen@.service** 인스턴스화 서비스를 사용하십시오.

예를 들어, RSA 키 유형의 자동 생성을 비활성화하려면 다음을 실행합니다.

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

자세한 내용은 **/etc/sysconfig/sshd** 파일을 참조하십시오.

(BZ#1228088)

기본 rsyslog 설정 파일은 비-레거시 형식 사용

rsyslog 패키지에 있는 설정 파일은 기본적으로 비-레거시 형식을 사용합니다. 레거시 형식은 계속 사용할 수 있지만, 현재 및 기존 설정 문을 혼합하는 데 몇 가지 제약이 있습니다. 이전 RHEL 릴리스에서 적용된 설정을 수정해야 합니다. 자세한 내용은 **rsyslog.conf(5)** 매뉴얼 페이지를 참조하십시오.

(BZ#1619645)

새로운 SELinux 부울값

이번 SELinux 시스템 정책 업데이트로 다음과 같은 부울 값이 도입되었습니다.

- `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`

자세한 내용은 다음 명령의 출력을 참조하십시오.

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

4.15. 가상화

KVM이 5 레벨 페이징 지원

Red Hat Enterprise Linux 8에서 KVM 가상화는 5단계 페이징을 지원합니다. 이 기능은 호스트 및 게스트 시스템이 사용할 수 있는 실제 및 가상 주소 공간을 많이 증가시킵니다.

(BZ#1485229)

KVM이 RHEL 8에서 UMIP 지원

이제 KVM 가상화가 UMIP(User-Mode Instruction Prevention) 기능을 지원하므로 사용자 공간 애플리케이션이 시스템 차원 설정에 액세스하는 것을 방지할 수 있습니다. 이렇게 하면 권한 확대 공격의 잠재적인 요소가 줄어들기 때문에 KVM 하이퍼바이저와 게스트 시스템을 보다 안전하게 보호할 수 있습니다.

(BZ#1494651)

KVM 게스트 충돌 보고서의 추가 정보

게스트가 예기치 않게 종료되거나 응답하지 않는 경우 KVM 하이퍼바이저가 생성하는 충돌 정보가 상세하게 제공됩니다. 따라서 KVM 가상화 배포에서 문제를 간편하게 진단하고 수정할 수 있습니다.

(BZ#1508139)

RHEL 8의 qemu-kvm 2.12

Red Hat Enterprise Linux 8은 **qemu-kvm** 2.12와 함께 배포됩니다. 이 버전은 Red Hat Enterprise Linux 7에서 사용할 수 있는 버전 1.5.3에 비해 여러 버그가 수정되고 개선 사항이 추가되었습니다.

특히 다음과 같은 기능이 도입되었습니다.

- Q35 게스트 시스템 유형
- UEFI 게스트 부팅
- NUMA 게스트 튜닝 및 고정
- vCPU 핫 플러그 및 핫 언플러그
- 게스트 I/O 스테딩

qemu-kvm 2.12에서 사용할 수 있는 기능 중 일부는 Red Hat Enterprise Linux 8에서 지원되지 않습니다. 자세한 내용은 Red Hat 고객 포털의 "Feature support and limitations in RHEL 8 virtualization"을 참조하십시오.

(BZ#1559240)

NVIDIA vGPU가 VNC 콘솔과 호환

NVIDIA vGPU(virtual GPU) 기능을 사용할 때 VNC 콘솔을 사용하여 게스트의 시각적 출력을 표시할 수 있습니다.

(BZ#1497911)

가상화에서 Ceph 지원

이번 업데이트를 통해 KVM 가상화가 Red Hat에서 지원하는 모든 CPU 아키텍처에서 Ceph 스토리지를 지원합니다.

(BZ#1578855)

가상화에서 Q35 시스템 유형 지원

Red hat Enterprise Linux 8은 최신 PCI Express 기반 시스템 유형인 **Q35**에 대한 지원을 도입했습니다. 이를 통해 가상 장치의 기능 및 성능이 다양하게 개선되고 광범위한 최신 장치가 가상화와 호환될 수 있습니다. 또한 Red Hat Enterprise Linux 8에서 생성된 가상 머신은 기본적으로 **Q35**를 사용하도록 설정되어 있습니다.

이전의 기본 **PC** 시스템 유형은 더 이상 사용되지 않으며 Q35를 지원하지 않는 이전 운영 체제를 가상화할 때만 사용해야 합니다.

(BZ#1599777)

IBM Z의 가상 머신에 대한 대화형 부트로더

IBM Z 호스트에서 가상 머신(VM)을 부팅할 때 QEMU 부팅 로더 펌웨어가 대화형 콘솔 인터페이스를 제공합니다. 이렇게 하면 호스트 환경에 액세스하지 않고도 게스트 OS 부팅 문제를 해결할 수 있습니다.

(BZ#1508137)

QEMU 샌드박스 추가

Red Hat Enterprise Linux 8에서 QEMU 에뮬레이터(Emulator)는 샌드박스 기능을 도입하고 있습니다. QEMU 샌드박스 기능은 QEMU를 호출하는 시스템에 설정 가능한 제한을 제공하므로 가상 머신의 보안을 더욱 강화합니다. 이 기능은 기본적으로 활성화되어 있습니다.

(JIRA:RHELPLAN-10628)

Intel Xeon SnowRidge에서 GFNI 및 CLDEMOT 명령 집합 활성화

Intel Xeon SnowRidge 시스템의 RHEL 8 호스트에서 실행 중인 가상 머신(VM)은 이제 GFNI 및 CLDEMOT 명령 집합을 사용할 수 있습니다. 이제 특정 시나리오에서 VM 등의 성능을 크게 향상시킬 수 있습니다.

(BZ#1494705)

NVMe 장치용 VFIO 기반 블록 드라이버 추가

QEMU 에뮬레이터(Emulator)는 NVMe(Non-volatile memory express) 장치용 VFIO(가상 함수 I/O) 기반 드라이버를 제공합니다. 이 드라이버는 가상 머신에 연결된 NVMe 장치와 직접 통신하고 커널 시스템 계층과 해당 NVMe 드라이버를 사용하지 않습니다. 결과적으로 가상 머신에서 NVMe 장치의 성능이 향상됩니다.

(BZ#1519004)

대규모 페이지 지원 개선

RHEL 8을 가상화 호스트로 사용하는 경우 사용자는 가상 머신(VM) 메모리를 지원하는 페이지 크기를 CPU에서 지원하는 크기로 수정할 수 있습니다. 이렇게 하면 VM의 성능을 크게 향상시킬 수 있습니다.

VM 메모리 페이지의 크기를 설정하려면 VM의 XML 구성을 편집하고 <hugepages> 요소를 <memoryBacking> 섹션에 추가합니다.

(JIRA:RHELPLAN-14607)

4.16. 지원 관련 기능

sosreport가 eBPF 기반 프로그램 및 맵 보고

Red Hat Enterprise Linux 8에서 로드된 eBPF(extended Berkeley Packet Filtering) 프로그램과 맵을 보고 하도록 **sosreport** 틀이 향상되었습니다.

(BZ#1559836)

5장. 기술 프리뷰

다음 부분에서는 Red Hat Enterprise Linux 8.0에서 사용 가능한 모든 기술 프리뷰 목록을 설명합니다.

기술 프리뷰 기능에 대한 Red Hat 지원 범위 내용은 [기술 프리뷰 기능 지원 범위](#)를 참조하십시오.

5.1. 커널

XDP를 기술 프리뷰로 이용 가능

기술 프리뷰로 이용할 수 있는 **eXpress Data Path(XDP)** 기능은 커널의 고성능 패킷 처리를 위해 BPF(Berkeley Packet Filter) 프로그램을 업로드할 수 있는 방법을 제공하여 커널의 네트워킹 데이터 경로를 프로그래밍할 수 있도록 합니다.

(BZ#1503672)

eBPF를 기술 프리뷰로 이용 가능

eBPF(extended Berkeley Packet Filtering) 기능은 네트워킹 및 추적을 위해 기술 프리뷰로 사용할 수 있습니다. **eBPF**는 사용자 지정 프로그램을 다양한 지점(소켓, 추적 지점, 패킷 수신)에 연결한 다음 데이터를 수신 및 처리할 수 있도록 사용자 공간을 활성화합니다. 이 기능에는 다양한 유형의 맵을 생성하고 다양한 유형의 프로그램을 커널에 삽입할 수 있는 새로운 시스템 호출 **bpf()**가 포함되어 있습니다. 자세한 내용은 **bpf(2)** 매뉴얼 페이지를 참조하십시오.

(BZ#1559616)

BCC를 기술 프리뷰로 이용 가능

BCC(BPF Compiler Collection)은 Red Hat Enterprise Linux 8에서 기술 프리뷰로 사용 가능한 효율적인 커널 추적 및 조작 프로그램을 생성하기 위한 사용자 공간 툴킷입니다. **BCC**는 **eBPF(extended Berkeley Packet Filtering)**를 사용하여 Linux 운영 체제의 I/O 분석, 네트워킹 및 모니터링을 위한 툴을 제공합니다.

(BZ#1548302)

Control Group v2를 RHEL 8에서 기술 프리뷰로 이용 가능

Control Group v2 메커니즘은 통합된 계층 컨트롤 그룹입니다. **Control Group v2**는 프로세스를 계층적으로 구성하고 제어 및 설정 가능한 방식으로 계층에 따라 시스템 리소스를 배포합니다.

이전 버전과 달리 **Control Group v2**에는 하나의 계층만 있습니다. 이 단일 계층을 통해 Linux 커널에서는 다음을 수행할 수 있습니다.

- 소유자의 역할에 따라 프로세스 분류
- 여러 계층에서 충돌하는 정책 문제 해결

Control Group v2에서는 다양한 컨트롤러가 지원되고 있습니다.

- CPU 컨트롤러가 CPU 사이클의 배분을 조정합니다. 이 컨트롤러는 다음을 구현합니다.
 - 일반적인 스케줄링 정책에 대한 가중치 및 절대 대역폭 제한 모델
 - 실시간 스케줄링 정책에 대한 절대 대역폭 할당 모델
- 메모리 컨트롤러가 메모리 할당을 조정합니다. 현재 다음과 같은 유형의 메모리 사용을 추적할 수 있습니다.
 - Userland 메모리 - 페이지 캐시 및 익명 메모리

- dentries 및 inode와 같은 커널 데이터 구조
- TCP 소켓 버퍼
- I/O 컨트롤러가 I/O 리소스 배분을 제한합니다.
- 쓰기 저장 컨트롤러는 메모리 및 I/O 컨트롤러와 상호작용하며 **Control Group v2** 고유의 것입니다.

위의 정보는 <https://www.kernel.org/doc/Documentation/cgroup-v2.txt> 링크를 기반으로 합니다. 동일한 링크를 참조하여 특정 **Control Group v2** 컨트롤러에 대한 자세한 정보를 얻을 수 있습니다.

(BZ#1401552)

early kdump는 Red Hat Enterprise Linux 8에서 기술 프리뷰로 이용 가능

early kdump 기능은 크래시 커널 및 initramfs가 초기 충돌에서도 **vmcore** 정보를 캡처할 수 있을 만큼 충분히 일찍 로드할 수 있게 되었습니다. **early kdump**에 대한 자세한 내용은 `/usr/share/doc/kexec-tools/early-kdump-howto.txt` 파일을 참조하십시오.

(BZ#1520209)

5.2. 파일 시스템 및 스토리지

VDO는 LVM 논리 볼륨 유형으로 사용 가능

LVM을 사용하여 **VDO(Virtual Data Optimizer)** 유형의 논리 볼륨을 생성할 수 있습니다. VDO는 데이터를 압축하고 중복 제거할 수 있는 기능을 갖춘 가상 블록 장치입니다.

이는 기술 프리뷰 기능입니다.

(BZ#1643553)

DIF/DIX(데이터 무결성 필드/데이터 무결성 확장) 지원

DIF/DIX가 SCSI 표준에 추가되었습니다. 지원 대상으로 특별히 나열된 항목을 제외하고는 모든 HBA 및 스토리지 배열의 경우 기술 프리뷰로 남아 있습니다.

DIF/DIX는 일반적으로 사용되는 512바이트 디스크 블록의 크기를 512에서 520바이트로 늘려 DIF(데이터 무결성 필드)를 추가합니다. DIF는 쓰기가 발생할 때 HBA(Host Bus Adapter)에 의해 계산된 데이터 블록의 체크섬 값을 저장합니다. 스토리지 장치는 수신 시 체크섬을 확인한 다음 데이터와 체크섬을 모두 저장합니다. 반대로, 읽기가 발생하면 스토리지 장치와 수신 HBA에서 체크섬을 확인할 수 있습니다.

(BZ#1649493)

NVMe/FC는 qla2xxx를 사용하여 Qlogic 어댑터에서 기술 프리뷰로 사용 가능

NVMe/FC(NVMe over Fibre Channel) 전송 유형은 **qla2xxx** 드라이버를 사용하여 Qlogic 어댑터에서 기술 프리뷰로 사용할 수 있습니다.

(BZ#1649922)

5.3. 고가용성 및 클러스터

Pacemaker podman 번들을 기술 프리뷰로 이용 가능

이제 Pacemaker 컨테이너 번들은 기술 프리뷰로 사용할 수 있는 컨테이너 번들 기능과 함께 **podman** 컨테이너 플랫폼에서 실행됩니다. 이 기능은 기술 프리뷰로 사용할 수 있지만 한 가지 예외가 있습니다. Red Hat은 Red Hat Openstack에 대한 Pacemaker 번들의 사용을 완전하게 지원합니다.

(BZ#1619620)

5.4. 보안

RHEL 8.0 릴리스의 SWID 태그

ISO/IEC 19770-2:2015 메커니즘을 사용하여 RHEL 8.0 설치를 식별할 수 있도록 SWID(소프트웨어 식별) 태그가 `/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8-<architecture>.swidtag` 및 `/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.swidtag`에 설치됩니다. 또한 이러한 태그의 상위 디렉터리는 `/etc/swid/swidtags.d/redhat.com` 심볼릭 링크를 통해 찾을 수 있습니다.

SWID 태그 파일의 XML 서명은 **xmlsec1 verify** 명령을 사용하여 확인할 수 있습니다. 예:

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

코드 서명 인증 기관의 인증서는 고객 포털의 [제품 서명 키](#) 페이지에서도 확인할 수 있습니다.

(BZ#1636338)

5.5. 가상화

KVM 가상 머신용 AMD SEV

RHEL 8은 KVM 하이퍼바이저를 사용하는 AMD EPYC 호스트 시스템을 위한 SEV(Secure Encrypted Virtualization) 기능을 기술 프리뷰로 도입하고 있습니다. 가상 머신(VM)에서 활성화된 경우 SEV는 VM 메모리를 암호화하여 호스트가 VM의 데이터에 액세스할 수 없도록 합니다. 이제 호스트가 악성 코드에 의해 감염된 경우 VM의 보안이 향상됩니다.

단일 호스트에서 이 기능을 한 번에 사용할 수 있는 VM의 개수는 호스트 하드웨어에 의해 결정됩니다. 현재 AMD EPYC 프로세서는 SEV를 사용하여 실행 중인 15개 이하의 VM을 지원합니다.

(BZ#1501618, BZ#1501607)

IBM POWER 9에서 중첩된 가상화 사용 가능

이제 IBM POWER 9 시스템에서 실행 중인 RHEL 8 호스트 시스템에서 기술 프리뷰로 중첩된 가상화 기능을 사용할 수 있습니다. 중첩된 가상화를 통해 KVM 가상 머신(VM)이 하이퍼바이저로 작동하여 VM 내에서 VM을 실행할 수 있습니다.

중첩된 가상화가 IBM POWER 9에서 작동하려면 호스트, 게스트 및 중첩된 게스트가 현재 다음 운영 체제 중 하나를 실행해야 합니다.

- RHEL 8
- POWER 9용 RHEL 7

(BZ#1505999)

6장. 사용되지 않는 기능

다음 부분에서는 Red Hat Enterprise Linux 8.0에서 더 이상 사용되지 않는 기능에 대해 설명합니다.

더 이상 사용되지 않는 기능도 Red Hat Enterprise Linux 8의 서비스가 종료될 때까지 계속 지원됩니다. 사용되지 않는 기능은 이 제품의 향후 주요 릴리스에서 지원되지 않을 가능성이 높으며 새로운 배포에 구현하는 것은 권장되지 않습니다. 특정 주요 릴리스 내에서 더 이상 사용되지 않는 기능의 최신 목록은 최신 릴리스 노트를 참조하십시오.

사용되지 않는 하드웨어 구성 요소는 현재 또는 향후 주요 릴리스의 새로운 배포에 구현하는 것을 권장하지 않습니다. 하드웨어 드라이버 업데이트는 보안 및 중요 수정 사항으로만 제한됩니다. Red Hat은 최대한 빠른 시일 내에 이 하드웨어를 교체할 것을 권장합니다.

패키지가 더 이상 사용되지 않으며 향후 사용이 권장되지 않는 경우가 있습니다. 경우에 따라 패키지가 제품에서 삭제될 수 있습니다. 제품 설명서에 더 이상 사용되지 않는 기능과 유사 또는 동일하거나 보다 고급 기능을 제공하는 최근 패키지가 지정된 권장 사항이 기재됩니다.

6.1. 설치 프로그램 및 이미지 생성

ignoredisk Kickstart 명령의 `--interactive` 옵션이 더 이상 사용되지 않음

Red Hat Enterprise Linux의 향후 릴리스에서 `--interactive option`을 사용하면 치명적인 설치 오류가 발생합니다. 옵션을 제거하려면 Kickstart 파일을 수정하는 것이 좋습니다.

(BZ#1637872)

6.2. 파일 시스템 및 스토리지

UDP를 통한 NFSv3이 비활성화

NFS 서버는 기본적으로 더 이상 UDP(User Datagram Protocol) 소켓을 열거나 수신하지 않습니다. 버전 4는 TCP(Transmission Control Protocol)가 필요하기 때문에 이 변경은 NFS 버전 3에만 영향을 미칩니다.

RHEL 8에서는 UDP를 통한 NFS가 더 이상 지원되지 않습니다.

(BZ#1592011)

NVMe/FC 대상 모드가 더 이상 사용되지 않음

이전에는 NVMe/FC(Nonvolatile Memory Express over Fibre Channel) 전송 프로토콜의 대상 모드를 RHEL 7에서 기술 프리뷰로 사용할 수 있었습니다. RHEL 8에서는 NVMe/FC 대상 모드가 더 이상 사용되지 않습니다.

NVMe 대상 모드에서 FC HBA(Host Bus Adapter) 포트를 활성화하면 다음과 같은 오류 메시지가 표시됩니다.

경고 : NVMe/FC 대상 - 이 드라이버는 이 릴리스에 대해 Red Hat에서 충분한 테스트를 거치지 않았기 때문에 프로덕션 시스템에서 사용할 수 없습니다.

(BZ#1664838)

6.3. 네트워크 구성

RHEL 8에서 네트워크 스크립트가 더 이상 사용되지 않음

네트워크 스크립트가 Red Hat Enterprise Linux 8에서 더 이상 사용되지 않으므로 이제 기본적으로 제공되지 않습니다. 기본 설치에는 **nmcli** 툴을 통해 **NetworkManager** 서비스를 호출하는 **ifup** 및 **ifdown** 스크립트의 새 버전을 제공합니다. Red Hat Enterprise Linux 8에서 **ifup** 및 **ifdown** 스크립트를 실행하려면 **NetworkManager**를 실행해야 합니다.

/sbin/ifup-local, **ifdown-pre-local** 및 **ifdown-local** 스크립트에서 사용자 지정 명령이 실행되지 않습니다.

이러한 스크립트가 필요한 경우 다음 명령을 사용하여 시스템에서 더 이상 사용되지 않는 네트워크 스크립트를 설치할 수 있습니다.

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

ifup 및 **ifdown** 스크립트는 설치된 레거시 네트워크 스크립트에 연결됩니다.

레거시 네트워크 스크립트를 호출하면 사용 중단을 알리는 경고 메시지가 표시됩니다.

(BZ#1647725)

6.4. 보안

Red Hat Enterprise Linux 8에서 DSA 사용 중단

Red Hat Enterprise Linux 8에서는 DSA(Digital Signature Algorithm)가 더 이상 사용되지 않습니다. DSA 키에 의존하는 인증 메커니즘은 기본 구성에서 작동하지 않습니다. **OpenSSH** 클라이언트는 LEGACY 정책에서도 DSA 호스트 키를 허용하지 않습니다.

(BZ#1646541)

SSL2 Client Hello가 NSS에서 더 이상 사용되지 않음

TLS(Transport Layer Security) 프로토콜 버전 1.2 이전의 SSL(Secure Sockets Layer) 프로토콜 버전 2와 역호환되는 방식으로 서식이 지정된 **Client Hello** 메시지와 협상을 시작할 수 있습니다. NSS(Network Security Services) 라이브러리에서 이 기능에 대한 지원은 더 이상 사용되지 않으며 기본적으로 비활성화되어 있습니다.

이 기능에 대한 지원이 필요한 애플리케이션은 새로운 **SSL_ENABLE_V2_COMPATIBLE_HELLO** API를 사용하여 활성화해야 합니다. 이 기능에 대한 지원은 Red Hat Enterprise Linux 8의 이후 릴리스에서 완전히 삭제될 수 있습니다.

(BZ#1645153)

6.5. 가상화

RHEL 8에서 가상 머신 스냅샷이 올바르게 지원되지 않음

가상 머신(VM) 스냅샷을 생성하는 현재 메커니즘이 안정적으로 작동하지 않기 때문에 더 이상 사용되지 않습니다. 따라서 RHEL 8에서 VM 스냅샷을 사용하지 않는 것이 좋습니다.

새로운 VM 스냅샷 메커니즘이 개발 중이며 향후 RHEL 8의 마이너 릴리스에서 완전히 구현될 예정입니다.

(BZ#1686057)

Cirrus VGA 가상 GPU 유형이 더 이상 사용되지 않음

향후 Red Hat Enterprise Linux에 대한 주요 업데이트가 있을 경우 KVM 가상 머신에서 **Cirrus VGA GPU** 장치가 더 이상 지원되지 않습니다. 따라서 Red Hat은 Cirrus VGA 대신 **stdvga**, **virtio-vga** 또는 **qxl** 장치 사용을 권장합니다.

(BZ#1651994)

virt-manager가 더 이상 사용되지 않음

virt-manager라고도 하는 Virtual Machine Manager 애플리케이션은 더 이상 사용되지 않습니다.

Cockpit라고도 하는 RHEL 8 웹 콘솔은 후속 릴리스에서 교체될 예정입니다. 따라서 GUI에서 가상화를 관리하기 위해 웹 콘솔을 사용하는 것이 좋습니다. 그러나 Red Hat Enterprise Linux 8.0의 일부 기능은 **virt-manager** 또는 명령줄에서만 액세스할 수 있습니다.

(JIRA:RHELPLAN-10304)

7장. 알려진 문제

다음 부분에서는 Red Hat Enterprise Linux 8의 알려진 문제에 대해 설명합니다.

7.1. 설치 프로그램 및 이미지 생성

auth 및 authconfig Kickstart 명령에는 AppStream 리포지토리가 필요

authselect-compat 패키지는 설치하는 동안 **auth** 및 **authconfig** Kickstart 명령이 필요합니다. 이 패키지가 없으면 **auth** 또는 **authconfig**가 사용되는 경우 설치에 실패합니다. 설계에 따라 **authselect-compat** 패키지는 AppStream 리포지토리에서만 사용할 수 있습니다.

이 문제를 해결하려면 설치 프로그램에 BaseOS 및 AppStream 리포지토리를 사용할 수 있는지 확인하거나 설치 중에 **authselect** Kickstart 명령을 사용합니다.

(BZ#1640697)

Binary DVD.iso 파일의 내용을 파티션에 복사해도 .treeinfo 및 .discinfo 파일이 복사되지 않음

RHEL 8.0 Binary DVD.iso 이미지 파일의 내용을 로컬 설치를 위해 파티션에 복사할 때 **cp <path>/^* <mounted partition>/dir** 명령의 '*'로 인해 설치에 필요한 **.treeinfo** 및 **.discinfo** 파일이 복사되지 않습니다. 결과적으로, BaseOS 및 AppStream 리포지토리가 로딩되지 않으며, **anaconda.log** 파일에 있는 디버그 로그 메시지에서만 이 문제를 확인할 수 있습니다.

이 문제를 해결하려면 누락된 **.treeinfo** 및 **.discinfo** 파일을 파티션에 복사하십시오.

(BZ#1692746)

7.2. 커널

부팅 시 i40iw 모듈이 자동으로 로드되지 않음

많은 i40e NIC에서 iWarp를 지원하지 않고 **i40iw** 모듈이 일시 중지/재개를 완전히 지원하지 않기 때문에 이 모듈이 기본적으로 자동 로드되지 않으므로 일시 중지/재개 동작이 제대로 작동하지 않게 됩니다. 이 문제를 해결하려면 수동으로 **/lib/udev/rules.d/90-rdma-hw-modules.rules** 파일을 편집하여 **i40iw**의 자동 로드를 활성화해야 합니다.

또한 동일한 시스템에 i40e 장치와 함께 설치된 다른 RDMA 장치가 있는 경우 비 i40e RDMA 장치는 **rdma** 서비스를 트리거하여 **i40iw** 모듈을 포함하여 활성화된 모든 RDMA 스택 모듈을 로드합니다.

(BZ#1623712)

많은 장치가 연결되어 있을 때 시스템이 응답하지 않는 경우가 있음

Red Hat Enterprise Linux 8이 다량의 장치를 구성하면, 시스템 콘솔에서 많은 수의 콘솔 메시지가 발생합니다. 예를 들어, 다수의 LUN(Logical Unit Number)이 있고 각 LUN에 대한 다중 경로가 있는 경우에 이러한 상황이 발생합니다. 커널이 수행하고 있는 다른 작업 외에도 콘솔 메시지가 많아 커널이 중지된 것처럼 보이기 때문에 커널 watchdog가 커널 패닉을 일으킬 수 있습니다.

부팅 사이클 초기에 스캔이 이루어지기 때문에 장치가 많이 연결되어 있으면 시스템이 응답하지 않게 됩니다. 일반적으로 부팅 시 이런 문제가 발생합니다.

부팅 후 장치 스캔 이벤트 중에 **kdump**가 시스템에 활성화되면 hard lockup으로 인해 **vmcore** 이미지가 캡처됩니다.

이 문제를 해결하려면 watchdog 잠금 타이머를 늘려야 합니다. 이렇게 하려면 커널 명령행에 **watchdog_thresh=N** 옵션을 추가하고 **N**을 초로 교체하십시오.

- 장치가 1,000개 미만인 경우 **30**을 사용하십시오.
- 장치가 1,000개 이상인 경우 **60**을 사용하십시오.

스토리지의 경우 장치 개수는 모든 LUN에 대한 경로 개수입니다. 즉 일반적으로 이는 **/dev/sd*** 장치 개수입니다.

해결방법을 적용하면 여러 개의 장치를 구성할 때 시스템이 더 이상 응답하지 않는 문제가 발생하지 않습니다.

(BZ#1598448)

KSM이 NUMA 메모리 정책을 무시하는 경우가 있음

"merge_across_nodes=1" 매개변수로 커널 공유 메모리(KSM) 기능을 활성화하면 KSM은 mbind() 함수로 설정된 메모리 정책을 무시하고 일부 메모리 영역의 페이지를 정책과 일치하지 않는 NUMA(Non-Uniform Memory Access) 노드에 병합할 수 있습니다.

이 문제를 해결하려면 QEMU와 함께 NUMA 메모리 바인딩을 사용하는 경우 KSM을 비활성화하거나 merge_across_nodes 매개변수를 "0"으로 설정합니다. 결과적으로 KVM VM에 대해 구성된 NUMA 메모리 정책이 예상대로 작동되게 됩니다.

(BZ#1153521)

7.3. 소프트웨어 관리

root가 아닌 사용자에서 yum list를 실행하면 YUM 충돌 발생

libdnf 패키지가 업데이트된 후 root가 아닌 사용자에서 **yum list** 명령을 실행하면 YUM가 예기치 않게 종료될 수 있습니다. 이 버그가 발견되면 root에서 **yum list**를 실행하여 문제를 해결할 수 있습니다. 그 결과 root가 아닌 사용자에서 **yum list**를 실행해도 더 이상 YUM 충돌이 발생하지 않습니다.

(BZ#1642458)

yum(8) 매뉴얼 페이지에 yum module profile 명령이 잘못 기재되어 있음

yum(8) 매뉴얼 페이지에 YUM 패키지 관리 틀에 모듈 프로필에 대한 상세 정보를 제공하는 **yum module profile** 명령이 포함되어 있다고 잘못 기술되어 있습니다. 그러나 이 명령은 더 이상 사용할 수 없으며, 만약 명령을 사용한다면 YUM이 잘못된 명령에 대한 오류 메시지를 표시합니다. 모듈 프로필에 대한 자세한 내용은 **yum module info --profile** 명령을 사용하십시오.

(BZ#1622580)

yum-plugin-aliases는 현재 사용할 수 없음

사용자 지정 yum 별칭을 추가하기 위한 **alias** 명령을 제공하는 **yum-plugin-aliases** 패키지는 현재 사용할 수 없습니다. 따라서 현재 별칭을 사용할 수 없습니다.

(BZ#1647760)

yum-plugin-changelog는 현재 사용할 수 없음

패키지 업데이트 이전과 이후에 패키지 변경 로그를 볼 수 있는 **yum-plugin-changelog** 패키지는 현재 사용할 수 없습니다.

(BZ#1581191)

7.4. 인프라 서비스

Tuned가 커널 부팅 명령줄 매개 변수를 설정하지 않음

Tuned 툴이 기본적으로 활성화되어 있는 BLS(Boot Loader Specification)를 지원하지 않습니다. 결과적으로 **Tuned**는 특정 커널 부팅 명령줄 매개 변수를 설정하지 않기 때문에 성능 저하 또는 CPU 코어가 분리되지 않는 것과 같은 몇 가지 문제가 발생합니다. 이 문제를 해결하려면 BLS를 비활성화하고 **Tuned**를 다시 시작하십시오.

1. `grubby` 패키지를 설치합니다.
2. `/etc/default/grub` 파일에서 다음 줄을 제거합니다.

```
GRUB_ENABLE_BLSCFG=true
```

3. 비 EFI 시스템에 대해 다음을 실행하여 `grub2.cfg` 파일을 다시 생성합니다.

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

or for EFI systems:

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

4. 다음을 실행하여 **Tuned**를 다시 시작합니다.

```
systemctl restart tuned
```

결과적으로 **Tuned**가 커널 부팅 매개 변수를 예상대로 설정합니다.

(BZ#1576435)

7.5. 셸 및 명령행 툴

net-snmp 패키지의 Python 바인딩을 사용할 수 없음

Net-SNMP 툴 제품군은 RHEL 8의 기본 **Python** 구현인 **Python 3**에 대한 바인딩을 제공하지 않습니다. 그 결과, `python-net-snmp`, `python2-net-snmp` 또는 `python3-net-snmp` 패키지는 RHEL 8에서 사용할 수 없습니다.

(BZ#1584510)

7.6. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

데이터베이스 서버를 병렬로 설치할 수 없음

충돌하는 RPM 패키지로 인해 RHEL 8.0에서 `mariadb` 및 `mysql` 모듈을 병렬로 설치할 수 없습니다.

설계상 동일한 모듈의 두 개 이상 버전(스트림)을 병렬로 설치할 수 없습니다. 예를 들어 `postgresql` 모듈에서 사용할 수 있는 스트림 중 하나를 **10**(기본) 또는 **9.6** 중에서 선택해야 합니다. RHEL 6 및 RHEL 7용 Red Hat Software Collections에서는 구성 요소의 병렬 설치가 가능합니다. RHEL 8에서는 컨테이너에서 다른 버전의 데이터베이스 서버를 사용할 수 있습니다.

(BZ#1566048)

mod_cgid 로깅 문제

RHEL 8에서 기본으로 **mod_cgid** Apache httpd 모듈을 스레드 MPM(Multi-processing module)에서 사용하는 경우 다음과 같은 로깅 문제가 발생합니다.

- CGI 스크립트의 **stderr** 출력 시 표준 타임스탬프 정보가 포함되지 않음
- CGI 스크립트의 **stderr** 출력이 **VirtualHost**에 대한 로그 파일로 올바르게 리디렉션되지 않음

(BZ#1633224)

IO::Socket::SSL Perl 모듈이 TLS 1.3을 지원하지 않음

세션 재개 또는 포스트 핸드셰이크(Post-Handshake) 인증과 같은 TLS 1.3 프로토콜의 새로운 기능이 RHEL 8 **OpenSSL** 라이브러리에서 구현되었지만 **Net::SSLeay** Perl 모듈에서는 구현되지 않았기 때문에 **IO::Socket::SSL** Perl 모듈에서 사용할 수 없습니다. 결과적으로 클라이언트 인증서 인증에 실패하고 세션 재연결이 TLS 1.2 프로토콜보다 느려질 수 있습니다.

이 문제를 해결하려면 **IO::Socket::SSL** 개체를 생성할 때 **SSL_version** 옵션을 **!TLSv1_3** 값으로 설정하여 TLS 1.3의 사용을 비활성화합니다.

(BZ#1632600)

7.7. IDM (IDENTITY MANAGEMENT)**KCM 인증 정보 캐시가 단일 인증 정보 캐시의 많은 인증 정보에 적합하지 않음**

인증 정보 캐시에 포함된 인증 정보가 너무 많으면 **kinit**와 같은 Kerberos 작업은 **sssd-kcm** 구성 요소와 기본 데이터베이스 간의 데이터 전송에 사용되는 버퍼의 하드코딩 제한으로 인해 실패합니다.

이 문제를 해결하려면 **/etc/sss/sss.conf** 파일의 **kcm** 섹션에 **ccache_storage = memory** 옵션을 추가합니다. 이렇게 하면 **kcm** 응답자가 인증 정보 캐시를 영구적으로 저장하지 않고 메모리에 저장하도록 지시합니다. 이렇게 하는 경우, 시스템을 다시 시작하거나 **sssd-kcm**으로 인증 정보 캐시를 지웁니다. KCM 은 최대 64KB의 캐시 크기를 처리할 수 있습니다.

(BZ#1448094)

시간 초과 값의 충돌로 인해 SSSD가 서버에 연결되지 않음

SSSD(System Security Services Daemon)에서 사용하는 장애 조치 작업과 관련된 기본 시간 초과 값 일부가 충돌합니다. 결과적으로 SSSD가 단일 서버와 통신하도록 예약된 시간 초과 값은 전체 시간 초과로 SSSD가 연결 작업 이전에 다른 서버를 시도하지 못하도록 합니다. 이 문제를 해결하려면 **dns_resolver_timeout** 매개변수 값보다 큰 **ldap_opt_timeout** 시간 초과 매개변수 값을 설정하고, **dns_resolver_op_timeout** 매개변수 값보다 큰 **dns_resolver_timeout** 매개변수 값을 설정합니다.

(BZ#1382750)

스마트 카드를 사용한 IdM 웹 UI 로그인 작동하지 않음

사용자가 스마트 카드에 저장된 인증서를 사용하여 IdM(Identity Management) 웹 UI에 로그인하려고 하면 SSSD(System Security Services Daemon) D-Bus 인터페이스 코드가 잘못된 콜백을 사용하여 사용자를 조회합니다. 그 결과 사용자 확인 조회가 충돌합니다. 문제를 해결하려면 다른 인증 방법을 사용하십시오.

(BZ#1642508)

IdM 서버가 FIPS에서 작동하지 않음

Tomcat용 SSL 커넥터 구현이 불완전하기 때문에 인증서 서버가 설치된 IdM 서버는 FIPS 모드가 활성화된 시스템에서 작동하지 않습니다.

(BZ#1673296)

nuxwdog 서비스가 HSM 환경에서 작동하지 않기 때문에 비 HSM 환경에서 keyutils 패키지를 설치해야 함

nuxwdog watchdog 서비스가 인증서 시스템에 통합되었습니다. 그 결과, nuxwdog은 더 이상 별도의 패키지로 제공되지 않습니다. watchdog 서비스를 사용하려면 pki-server 패키지를 설치하십시오.

nuxwdog 서비스에는 다음과 같은 알려진 문제점이 있습니다.

- HSM(하드웨어 스토리지 모듈)을 사용하는 경우 nuxwdog 서비스가 작동하지 않습니다. 이 문제의 해결방법은 없습니다.
- 비 HSM 환경에서 Red Hat Enterprise Linux 8.0은 keyutils를 패키지 종속성으로 자동 설치하지 않습니다. 패키지를 수동으로 설치하려면 `dnf install keyutils` 명령을 사용하십시오.

(BZ#1652269)

7.8. 컴파일러 및 개발 도구

GCC에서 생성된 합성 함수가 SystemTap에 혼란 가중

GCC 최적화는 다른 함수의 부분 인라인 사본에 대해 합성 함수를 생성할 수 있습니다. SystemTap 및 GDB와 같은 툴은 이러한 합성 함수를 실제 함수와 구분할 수 없습니다. 결과적으로 SystemTap은 합성 및 실제 함수 진입점 모두에 프로브를 배치할 수 있으므로 단일 실제 함수 호출에 대해 여러 개의 프로브 히트를 등록합니다.

이 문제를 해결하려면 SystemTap 스크립트를 인라인 부분 함수와 관련된 재귀 탐지 및 프로브 억제와 같은 조치로 수정해야 합니다. 스크립트 예시:

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

다음과 같이 설명된 문제를 피할 수 있습니다.

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

이 예시 스크립트는 kprobes 또는 kretprobes 누락, 의도된 재귀와 같이 가능한 모든 시나리오를 고려하지 않습니다.

(BZ#1169184)

ltrace 툴이 함수 호출을 보고하지 않음

모든 RHEL 구성 요소에 적용된 바이너리 강화 기능이 개선되었으므로 **ltrace** 툴은 RHEL 구성 요소의 바이너리 파일에서 함수 호출을 더 이상 감지할 수 없습니다. 따라서 이러한 바이너리 파일에 사용될 때 감지된 모든 호출을 보고하지 않으므로 **ltrace** 출력이 비어 있습니다. 현재 해결방법은 없습니다.

참고로 **ltrace**는 해당 강화 플래그없이 구축된 사용자 정의 바이너리 파일에서 호출을 올바르게 보고할 수 있습니다.

(BZ#1618748, BZ#1655368)

7.9. 파일 시스템 및 스토리지

iscsiuio 패키지를 사용하여 iSCSI 대상을 찾을 수 없음

Red Hat Enterprise Linux 8은 PCI 레지스터 영역에 대한 동시 액세스를 허용하지 않습니다. 결과적으로 **could not set host net params (err 29)** 오류가 설정된 검색 포털에 대한 연결이 실패했습니다. 이 문제를 해결하려면 iSCSI 오프로드에 대한 커널 명령줄에 커널 매개변수 **iomem=relaxed**를 설정합니다. 특히 **bnx2i** 드라이버를 사용하는 모든 오프로드를 포함합니다. 그 결과 검색 포털에 성공적으로 연결되고 **iscsiuio** 패키지가 올바르게 작동합니다.

(BZ#1626629)

XFS DAX 마운트 옵션이 공유 COW(Copy-On-Write) 데이터 Extent와 호환되지 않음

공유 COW(Copy-On-Write) 데이터 Extent 기능으로 포맷화된 XFS 파일 시스템이 **-o dax** 마운트 옵션과 호환되지 않습니다. 그 결과 **-o dax**와 파일 시스템 마운트에 실패합니다.

이 문제를 해결하려면 **reflink=0** 메타데이터 옵션을 사용하여 파일 시스템을 포맷한 뒤 공유 COW(Copy-On-Write) 데이터 Extent를 비활성화하십시오.

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

그 결과 **-o dax**를 사용한 파일 시스템 마운트에 성공합니다.

(BZ#1620330)

7.10. 네트워크 구성

nftables이 다차원 IP 집합 유형을 지원하지 않음

nftables 패킷 필터링 프레임워크가 연결 및 간격이 있는 집합 유형을 지원하지 않습니다. 그 결과, **hash:net,port**와 같은 다차원 IP 집합 유형을 **nftables**와 함께 사용할 수 없습니다.

이 문제를 해결하려면 다차원 IP 집합 유형이 필요한 경우 **iptables** 프레임워크를 **ipset** 툴과 함께 사용하십시오.

(BZ#1593711)

7.11. 보안

OpenSCAP **rpmverifypackage**가 올바르게 작동하지 않음

chdir 및 **chroot** 시스템 호출은 **rpmverifypackage** 프로브에 의해 두 번 호출됩니다. 그 결과, 사용자 지정 OVAL(Open Vulnerability and Assessment Language) 콘텐츠로 **OpenSCAP** 스캔 중에 프로브를 사용할 때 오류가 발생합니다.

이 문제를 해결하려면 **rpmverifypackage_test** OVAL 테스트를 콘텐츠에서 사용하지 마십시오. 또는 **rpmverifypackage_test**가 사용되지 않은 **scap-security-guide** 패키지에서만 콘텐츠를 사용하십시오.

(BZ#1646197)

libssh가 시스템 차원의 암호화 정책을 따르지 않음

libssh 라이브러리는 시스템 차원의 암호화 정책 설정을 따르지 않습니다. 그 결과, 관리자가 **update-crypto-policies** 명령을 사용하여 암호 정책 수준을 변경할 때 지원되는 알고리즘 세트가 변경되지 않습니다.

이 문제를 해결하기 위해 **libssh**를 사용하는 모든 애플리케이션별로 공개된 알고리즘 세트를 각각 설정해야 합니다. 그 결과, 시스템을 LEGACY 또는 FUTURE 정책 수준으로 설정할 때 **libssh**를 사용하는 애플리케이션이 **OpenSSH**와 비교했을 때 일관성 없이 작동합니다.

(BZ#1646563)

SCAP Workbench가 사용자 정의 프로파일에서 결과를 기반으로 한 수정을 생성하지 못함

SCAP Workbench 툴을 사용하여 사용자 정의된 프로파일에서 결과 기반 수정 역할을 생성하려고 할 때 다음과 같은 오류가 발생합니다.

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

이 문제를 해결하려면 **--tailoring-file** 옵션과 함께 **oscap** 명령을 사용하십시오.

(BZ#1640715)

OpenSCAP rpmverifyfile이 작동하지 않음

OpenSCAP 스캐너가 오프라인 모드에서 현재 작업 중인 디렉터리를 올바르게 변경하지 않으며 **fchdir** 함수가 **OpenSCAP rpmverifyfile** 프로브에서 올바른 인수와 함께 호출되지 않습니다. 그 결과, **oscap-chroot** 명령을 사용하여 임의의 파일 시스템을 스캔할 때 SCAP 콘텐츠에서 **rpmverifyfile_test**가 사용되면 실패합니다. 따라서 설명된 시나리오에서 **oscap-chroot**이 중단됩니다.

(BZ#1636431)

컨테이너 보안 및 준수 검사를 위한 유틸리티를 사용할 수 없음

Red Hat Enterprise Linux 7에서 **oscap-docker** 유틸리티는 Atomic 기술을 기반으로 한 Docker 컨테이너 스캔에 사용할 수 있었습니다. Red Hat Enterprise Linux 8에서는 Docker 및 Atomic 관련 **OpenSCAP** 명령을 사용할 수 없습니다. 그 결과, 현재 RHEL 8에서는 컨테이너 보안 및 준수 검사를 위한 **oscap-docker** 또는 이와 동등한 유틸리티를 사용할 수 없습니다.

(BZ#1642373)

8장. 컨테이너 관련 주요 변경 사항

Red Hat Enterprise Linux (RHEL) 8.0에서 컨테이너 이미지 집합을 사용할 수 있습니다. 주요 변경 사항은 다음과 같습니다.

- Docker가 RHEL 8.0에 포함되어 있지 않습니다. 컨테이너 작업을 할 때는 **podman**, **buildah**, **skopeo** 및 **runc** 툴을 사용하십시오.
RHEL 8에서 이러한 툴 및 컨테이너 사용에 대한 자세한 내용은 [Building, running, and managing containers](#)를 참조하십시오.
- **podman** 툴의 기능이 완전하게 지원됩니다.
podman 툴은 단일 노드에서 pod, 컨테이너 이미지 및 컨테이너를 관리합니다. Pod라고 하는 컨테이너 및 컨테이너 그룹을 관리할 수 있는 **libpod** 라이브러리를 기반으로 합니다.

podman 사용 방법에 대한 자세한 내용은 [Building, running, and managing containers](#)를 참조하십시오.
- RHEL 8 GA에서는 Red Hat Universal Base Images (UBI)가 새롭게 제공됩니다. UBI는 표준 및 최소 RHEL 기반 이미지와 같이 Red Hat에서 이전에 제공한 이미지의 일부를 교체합니다.
이전 Red Hat 이미지와 달리 UBI는 자유롭게 재배포가 가능합니다. 즉, 모든 환경에서 사용하고 어디에서나 공유할 수 있습니다. Red Hat 고객이 아니어도 사용할 수 있습니다.

UBI에 대한 자세한 내용은 [Building, running, and managing containers](#)를 참조하십시오.
- RHEL 8 GA에서는 AppStream 구성 요소를 제공하는 추가 컨테이너 이미지를 사용할 수 있으며, RHEL 7에서는 컨테이너 이미지가 **Red Hat Software Collections**와 함께 배포됩니다. 이러한 모든 RHEL 8 이미지는 **ubi8** 기본 이미지를 기반으로 합니다.
- 64bit ARM 아키텍처용 컨테이너 이미지는 RHEL 8에서 완벽하게 지원됩니다.
- RHEL 8에서는 **rhel-tools** 컨테이너가 제거되었습니다. **sos** 및 **redhat-support-tool** 툴은 **support-tools** 컨테이너에서 제공됩니다. 시스템 관리자는 이 이미지를 시스템 도구 컨테이너 이미지를 구축하는 데 기본으로 사용할 수도 있습니다.
- 루트리스(rootless) 컨테이너에 대한 지원은 RHEL 8에서 기술 프리뷰로 사용할 수 있습니다. 루트리스(rootless) 컨테이너는 관리자 권한 없이 일반 시스템 사용자가 생성 및 관리하는 컨테이너입니다.

부록 A. 구성 요소별 티켓 목록

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

구성 요소	티켓
glibc	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
go-toolset	BZ#1512570
httpd	BZ#1633224, BZ#1632754
iproute	BZ#1640991
iptables	BZ#1644030, BZ#1564596
iscsi-initiator-utils	BZ#1626629, BZ#1582099
kernel-rt	BZ#1592977
kernel	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
kexec-tools	BZ#1520209
kmod-kvdo	BZ#1534087, BZ#1639512
libdnf	BZ#1642458
libreswan	BZ#1657854
libssh	BZ#1485241
ltrace	BZ#1618748, BZ#1584322
lvm2	BZ#1643553, BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
mariadb	BZ#1637034
net-snmp	BZ#1584510
nfs-utils	BZ#1592011, BZ#1639432
nftables	BZ#1593711

구성 요소	티켓
nginx	BZ#1545526
nodejs-10-module	BZ#1622118
nss	BZ#1645153
nuxwdog	BZ#1652269
openldap	BZ#1570056
opensc	BZ#1595626
openscap	BZ#1646197, BZ#1636431, BZ#1642373, BZ#1618484, BZ#1614273, BZ#1618464
openssh	BZ#1622511, BZ#1228088
pacemaker	BZ#1543494
pcs	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
perl	BZ#1511131
pki-core	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, BZ#1669257 , BZ#1656856, BZ#1673296
pykickstart	BZ#1637872, BZ#1612061
qemu-kvm	BZ#1508139, BZ#1559240, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1519004
redhat-release	BZ#1636338
rsyslog	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645
scap-security-guide	BZ#1618528, BZ#1618518
scap-workbench	BZ#1640715
selinux-policy	BZ#1592244, BZ#1594111, BZ#1549772, BZ#1626446

구성 요소	티켓
setup	BZ#1591969
sos	BZ#1559836
sssd	BZ#1448094, BZ#1382750, BZ#1642508, BZ#1620123
subversion	BZ#1571415
swig-3.0-module	BZ#1660051
tomcatjss	BZ#1424966, BZ#1636564
tuned	BZ#1576435, BZ#1565598
valgrind	BZ#1500481, BZ#1538009
virt-manager	BZ#1599777, BZ#1643609
wpa_supplcaant	BZ#1582538
기타	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, BZ#1686057 , BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, BZ#1650701 , 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