



Red Hat Enterprise Linux 8

8.0 릴리스 노트

Release Notes for Red Hat Enterprise Linux 8.0

Red Hat Enterprise Linux 8 8.0 릴리스 노트

Release Notes for Red Hat Enterprise Linux 8.0

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/8.0_Release_Notes.ent file |.

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

릴리스 노트에서는 Red Hat Enterprise Linux 8.0에서 구현된 개선 사항 및 추가 사항 및 이 릴리스의 알려진 문제뿐만 아니라 주요 버그 수정, 기술 프리뷰, 사용되지 않는 기능 및 기타 세부 사항에 대해 설명합니다.

차례

RED HAT 문서에 관한 피드백 제공	5
1장. 개요	6
콘텐츠 배포	6
소프트웨어 관리	6
셸 및 명령행 툴	6
동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버	6
데스크탑	6
설치 프로그램 및 이미지 생성	6
커널	7
파일 시스템 및 스토리지	7
보안	7
네트워킹	7
가상화	7
컴파일러 및 개발 도구	8
고가용성 및 클러스터	8
추가 리소스	8
Red Hat Customer Portal 랩	9
2장. 아키텍처	11
3장. RHEL 8의 콘텐츠 배포	12
3.1. 설치	12
3.2. 리포지토리	12
3.3. APPLICATION STREAMS	13
3.4. YUM/DNF를 사용한 패키지 관리	14
4장. RHEL 8.0.1 릴리스	15
4.1. 새로운 기능	15
4.2. 확인된 문제	17
5장. RHEL 8.0.0 릴리스	18
5.1. 새로운 기능	18
5.1.1. 웹 콘솔	18
5.1.2. 설치 프로그램 및 이미지 생성	20
5.1.3. 커널	24
5.1.4. 소프트웨어 관리	27
5.1.5. 인프라 서비스	30
5.1.6. 셸 및 명령행 툴	34
5.1.7. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버	36
5.1.8. 데스크탑	48
5.1.9. 하드웨어 지원	51
5.1.10. IdM (Identity Management)	52
5.1.11. 컴파일러 및 개발 도구	58
5.1.12. 파일 시스템 및 스토리지	75
5.1.13. 고가용성 및 클러스터	83
5.1.14. 네트워킹	87
5.1.15. 보안	96
5.1.16. 가상화	109
5.1.17. 지원 관련 기능	114
5.2. 버그 수정	115
5.2.1. 데스크탑	115

5.2.2. 그래픽 인프라	115
5.2.3. IdM (Identity Management)	115
5.2.4. 컴파일러 및 개발 도구	116
5.2.5. 파일 시스템 및 스토리지	117
5.2.6. 고가용성 및 클러스터	118
5.2.7. 네트워킹	120
5.2.8. 보안	120
5.2.9. 서브스크립션 관리	120
5.2.10. 가상화	121
5.3. 기술 프리뷰	121
5.3.1. 커널	121
5.3.2. 그래픽 인프라	124
5.3.3. 하드웨어 지원	124
5.3.4. IdM (Identity Management)	124
5.3.5. 파일 시스템 및 스토리지	125
5.3.6. 고가용성 및 클러스터	129
5.3.7. 네트워킹	129
5.3.8. Red Hat Enterprise Linux System Roles	130
5.3.9. 가상화	131
5.4. 사용되지 않는 기능	133
5.4.1. 설치 프로그램 및 이미지 생성	134
5.4.2. 파일 시스템 및 스토리지	135
5.4.3. 네트워킹	136
5.4.4. 커널	137
5.4.5. 보안	137
5.4.6. 가상화	138
5.4.7. 더 이상 사용되지 않는 패키지	139
5.5. 확인된 문제	140
5.5.1. 웹 콘솔	140
5.5.2. 설치 프로그램 및 이미지 생성	140
5.5.3. 커널	142
5.5.4. 소프트웨어 관리	146
5.5.5. 인프라 서비스	147
5.5.6. 셸 및 명령행 툴	148
5.5.7. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버	149
5.5.8. 데스크탑	150
5.5.9. 그래픽 인프라	151
5.5.10. 하드웨어 지원	151
5.5.11. IdM (Identity Management)	152
5.5.12. 컴파일러 및 개발 도구	157
5.5.13. 파일 시스템 및 스토리지	158
5.5.14. 네트워킹	159
5.5.15. 보안	162
5.5.16. 서브스크립션 관리	168
5.5.17. 가상화	169
5.5.18. 지원 관련 기능	171
6장. 컨테이너 관련 주요 변경 사항	172
7장. 국제화	174
7.1. RED HAT ENTERPRISE LINUX 8 INTERNATIONAL LANGUAGES	174
7.2. RHEL 8 국제화에 대한 주요 변경 사항	174
부록 A. 구성 요소별 티켓 목록	176

감사 인사	182
부록 B. 개정 내역	183

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오. 이렇게 하려면 다음을 수행합니다.

- 특정 문구에 대해 간단한 의견을 남기려면 문서가 Multi-page HTML 형식으로 되어 있는지 확인하십시오. 주석하려는 텍스트 부분을 강조 표시합니다. 그런 다음 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭하고 표시된 지침을 따릅니다.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
 1. [Bugzilla](#) 웹 사이트로 이동합니다.
 2. 구성 요소로 **문서**를 사용합니다.
 3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **버그 제출**을 클릭합니다.

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 툴링과의 통합을 위해 안정적이고 우수한 API를 제공합니다.

자세한 내용은 [5.1.4절. "소프트웨어 관리"](#) 을 참조하십시오.

셸 및 명령행 툴

- RHEL 8에서는 **Git 2.18**, **Mercurial 4.8** 및 **Subversion 1.10** 의 버전 관리 시스템을 제공합니다.

자세한 내용은 [???](#) 를 참조하십시오.

동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

- **Python 3.6** 은 RHEL 8의 기본 **Python** 구현이며 **Python 2.7** 에 대한 제한된 지원이 제공됩니다. Python 2.7에 대한 지원은 제한됩니다. 기본적으로 Python 버전이 설치되어 있지 않습니다.
- **Node.js** 는 RHEL에서 새로운 기능입니다. 기타 동적 프로그래밍 언어는 RHEL 7: **PHP 7.2**, **Ruby 2.5**, **Perl 5.26**, **SWIG 3.0** 부터 업데이트되었습니다.
- 다음 데이터베이스 서버는 RHEL 8: **MariaDB 10.3**, **MySQL 8.0**, **PostgreSQL 10**, **PostgreSQL 9.6** 및 **Redis 5** 와 함께 배포됩니다.
- RHEL 8에서는 **Apache HTTP Server 2.4** 를 제공하며 새로운 웹 서버, **nginx 1.14** 를 도입합니다.
- **squid** 가 버전 4.4로 업데이트되었으며 새로운 프록시 캐싱 서버가 포함되어 있습니다. **Varnish Cache 6.0**.

자세한 내용은 [5.1.7절. "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 [5.1.8절. "데스크탑"](#) 를 참조하십시오.

설치 프로그램 및 이미지 생성

- **Anaconda** 설치 프로그램에서는 **LUKS2** 디스크 암호화를 사용하여 **NVDIMM** 장치에 시스템을 설치할 수 있습니다.
- **Image Builder** 도구를 사용하면 다양한 공급자의 클라우드에 배포할 수 있는 이미지를 포함하여 다양한 형식으로 사용자 지정 시스템 이미지를 만들 수 있습니다.
- RHEL 8에서는 **HMC**(Hardware Management Console) 및 **IBM Z**의 **SE**(Hardware Management Console)를 사용하여 DVD에서 설치할 수 있습니다.

자세한 내용은 [5.1.2절. “설치 프로그램 및 이미지 생성”](#) 을 참조하십시오.

커널

- **eBPF**(extended Berkeley Packet Filtering) 기능을 통해 사용자 지정 프로그램을 다양한 지점(소켓, 추적 포인트, 패킷 수신)에 연결하여 데이터를 수신하고 처리할 수 있습니다. 이 기능은 **기술 프리뷰** 로 제공됩니다.
- 효율적인 커널 추적 및 조작 프로그램을 만드는 툴인 **BCC**(BPF Compiler Collection)는 **기술 프리뷰** 로 사용할 수 있습니다.

자세한 내용은 [5.3.1절. “커널”](#) 을 참조하십시오.

파일 시스템 및 스토리지

- **LUKS2**(**LUKS2**버전) 형식은 기존 **LUKS**(**LUKS1**) 형식을 대체합니다. 이제 **dm-crypt** 하위 시스템과 **cryptsetup** 툴은 **LUKS2**를 암호화된 볼륨의 기본 형식으로 사용합니다.

자세한 내용은 [5.1.12절. “파일 시스템 및 스토리지”](#) 을 참조하십시오.

보안

- TLS, IPsec, SSH, DNSSEC 및 Kerberos 프로토콜을 대상으로 하는 코어 암호화 하위 시스템을 구성하는 시스템 전체 암호화 **정책**이 기본적으로 적용됩니다. 관리자가 새로운 **update-crypto-policies** 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 [5.1.15절. “보안”](#) 을 참조하십시오.

네트워킹

- **nftables** 프레임워크가 기본 네트워크 패킷 필터링 기능의 역할에서 **iptables**를 대체하여 사용됩니다.
- 이제 **firewalld** 데몬에서는 **nftables**를 기본 백엔드로 사용합니다.
- 여러 컨테이너에 대한 네트워크 연결을 가능하게 하는 **IPVLAN** 가상 네트워크 드라이버 지원이 도입되었습니다.
- eBPF(extended Berkeley Packet Filtering) 기능의 일부로 eXpress Data Path(**XDP**) 및 Address Family eXDP(Address Family eXDP)는 **eBPF**(extended Berkeley Packet Filtering)기능의 일부로 사용할 수 있습니다. 자세한 내용은 **기술 프리뷰**의 [5.3.7절. “네트워킹”](#) 을 참조하십시오.

추가 기능을 위한 새로운 기능은 [5.1.14절. “네트워킹”](#) 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.
- Cockpit이라고도 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 [5.1.16절. “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 SystemTap과 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.
- RHEL 8에서는 OpenJDK 11, OpenJDK 8 IcedTea-Web 및 `aka`, `aka`, `Maven` 또는 `Scala` 와 같은 다양한 Java 툴을 제공합니다.

자세한 내용은 [5.1.11절. “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 [5.1.13절. “고가용성 및 클러스터”](#)를 참조하십시오.

추가 리소스

- Red Hat Enterprise Linux 8의 기능 및 제한사항은 지식베이스 문서 [Red Hat Enterprise Linux 기술 기능 및 제한](#)에서 사용할 수 있습니다.
- Red Hat Enterprise Linux 라이프 사이클에 대한 정보는 [Red Hat Enterprise Linux 라이프 사이클](#) 문서를 참조하십시오.

- 패키지 매니페스트 문서는 **RHEL 8**의 패키지 목록을 제공합니다.
- **RHEL 7**과 **RHEL 8**의 주요 차이점은 **RHEL 8 도입에 대한 고려** 사항에 설명되어 있습니다.
- **RHEL 7**에서 **RHEL 8**로의 인플레이스 업그레이드를 수행하는 방법에 대한 지침은 **RHEL 7**에서 **RHEL 8로 업그레이드하여** 제공합니다.
- **Currently supported upgrade paths are listed in-place upgrade paths for Red Hat Enterprise Linux.**
- 알려진 기술 문제를 사전에 확인, 검토 및 해결할 수 있는 **Red Hat Insights** 서비스는 모든 **RHEL** 서브스크립션을 통해 사용할 수 있습니다. **Red Hat Insights** 클라이언트를 설치하고 시스템을 서비스에 등록하는 방법에 대한 자세한 내용은 **Red Hat Insights Get Started** 페이지를 참조하십시오.

Red Hat Customer Portal 랩

Red Hat 고객 포털 랩은 <https://access.redhat.com/labs/>에서 제공되는 고객 포털의 섹션에 있는 일련의 틀입니다. **Red Hat Customer Portal Labs**의 애플리케이션을 통해 성능 향상, 신속한 문제 해결, 보안 문제 확인, 복잡한 애플리케이션을 신속하게 배포 및 구성할 수 있습니다. 가장 많이 사용되는 앱 중 일부는 다음과 같습니다.

- [등록 도우미](#)
- [Kickstart 생성기](#)
- [제품 라이프 사이클 검사기](#)
- [Red Hat 제품 인증서](#)
- [Red Hat Satellite 업그레이드 도우미](#)
- [Red Hat CVE Checker](#)

- [JVM 옵션 구성 툴](#)
- [로드 밸런서 구성 툴](#)
- [Red Hat Code 브라우저](#)
- [yum Repository Configuration Helper](#)

2장. 아키텍처

Red Hat Enterprise Linux 8.0은 다음 아키텍처를 지원하는 커널 버전 **4.18.0-80**과 함께 배포됩니다.

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

각 아키텍처에 적합한 서브스크립션을 구매해야 합니다. 자세한 내용은 [Red Hat Enterprise Linux 시작하기 - 추가 아키텍처](#)를 참조하십시오. 사용 가능한 서브스크립션 목록은 고객 포털의 [서브스크립션 사용률](#)을 참조하십시오.

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.



참고

바이너리 DVD ISO 이미지는 4.7GB보다 크므로 단일 계층 DVD에 맞지 않을 수 있습니다. 바이너리 DVD ISO 이미지를 사용하여 부팅 가능한 설치 미디어를 생성할 때 이중 계층 DVD 또는 USB 키를 사용하는 것이 좋습니다. Image Builder 툴을 사용하여 사용자 지정 RHEL 이미지를 생성할 수도 있습니다. 이미지 빌더에 대한 자세한 내용은 [사용자 지정된 RHEL 시스템 이미지 컴파일](#) 문서를 참조하십시오.

- 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 8 설치 문서를 참조하십시오. 자동화된 Kickstart 설치 및 기타 고급 주제는 고급 8 RHEL 설치 문서를 참조하십시오.

3.2. 리포지토리

Red Hat Enterprise Linux 8은 다음 두 가지 주요 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. **BaseOS**를 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. **Application Streams**는 친숙한 **RPM** 형식으로, 모듈이라는 **RPM** 형식의 확장 또는 **Software Collections**로 사용할 수 있습니다. **AppStream**에서 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 지원되지 않습니다.

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

3.3. APPLICATION STREAMS

Red Hat Enterprise Linux 8에는 **Application Streams**의 개념이 도입되어 있습니다. 이제 여러 버전의 사용자 공간 구성 요소가 핵심 운영 체제 패키지보다 더 자주 제공되고 업데이트됩니다. 이는 플랫폼 또는 특정 배포의 기본 안정성에 영향을 주지 않고 **Red Hat Enterprise Linux**를 사용자 지정할 수 있는 유연성을 향상시킵니다.

Application Stream으로 사용 가능한 구성 요소는 모듈 또는 **RPM** 패키지로 패키징할 수 있으며 **RHEL 8**의 **AppStream** 리포지토리를 통해 제공합니다. 각 **Application Stream** 구성 요소에는 **RHEL 8**과 동일하거나 더 짧은 라이프 사이클이 있습니다. 자세한 내용은 [Red Hat Enterprise Linux 라이프 사이클](#)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** 모듈에서 **PostgreSQL** 데이터베이스 서버의 여러 스트림(버전)을 기본 **postgresql:10** 스트림을 사용하여 사용할 수 있습니다. 시스템에는 하나의 모듈 스트림만 설치할 수 있습니다. 개별 컨테이너에서 서로 다른 버전을 사용할 수 있습니다.

자세한 모듈 명령은 [사용자 공간 구성 요소 설치, 관리 및 제거](#) 문서에 설명되어 있습니다. **AppStream**에서 사용 가능한 모듈 목록은 [패키지 매니페스트](#)를 참조하십시오.

3.4. YUM/DNF를 사용한 패키지 관리

Red Hat Enterprise Linux 8에서 소프트웨어 설치에 DNF 기술을 기반으로 하는 YUM 틀에 의해 확인됩니다. 이전 RHEL 주요 버전과의 일관성을 위해 **yum** 용어 사용을 의도적으로 준수합니다. 그러나 **yum** 대신 **dnf** 를 입력하면 **yum** 이 호환성을 위해 **dnf** 의 별칭이기 때문에 명령이 예상대로 작동합니다.

자세한 내용은 다음 문서를 참조하십시오.

- [사용자 공간 구성 요소 설치, 관리 및 제거](#)
- [RHEL 8 도입의 고려 사항](#)

4장. RHEL 8.0.1 릴리스

4.1. 새로운 기능

RHEL System Roles 업데이트

RHEL 하위 시스템의 구성 인터페이스를 제공하는 **rhel-system-roles** 패키지가 업데이트되었습니다. 주요 변경 사항은 다음과 같습니다.

- 네트워크 역할에서 **absent** 프로필 처리가 향상되었습니다. 영구 상태를 **absent**로 설정하여 기존 **NetworkManager** 온-디스크 프로필 구성을 삭제할 때 프로필의 영구 구성만 제거되고 현재 런타임 구성은 변경되지 않습니다. 그 결과, 설명된 상황에서 해당 네트워크 장치가 더 이상 중단되지 않습니다.
- VLAN 및 MACVLAN 인터페이스에 대해 최대 전송 단위(MTU) 크기를 지정하면 네트워크 역할의 MACVLAN 인터페이스가 수정되었습니다. 결과적으로 네트워크 역할을 사용하여 VLAN 및 MACVLAN 인터페이스에서 MTU 크기를 설정하면 더 이상 다음 오류 메시지와 함께 실패하지 않습니다.


```
failure: created connection failed to normalize: nm-connection-error-quark:
connection.type: property is missing (6)
```
- selinux 및 timesync 역할에는 문서화된 모든 입력 변수가 기본값 파일(기본값/main.yml)에 포함됩니다. 이렇게 하면 각 기본값 파일의 콘텐츠를 검사하여 역할에서 지원하는 입력 변수를 쉽게 확인할 수 있습니다.
- kdump 및 timesync 역할이 점검 모드에서 실패하지 않도록 수정되었습니다.

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

SOS-collector 버전 1.7으로 다시 기반

sos-collector 패키지는 RHEL 8.0.1에서 버전 1.7로 업데이트되었습니다. 주요 변경 사항은 다음과 같습니다.

- SOS-collector 는 이제 일반 RHEL 노드에서와 동일한 방식으로 RHCOS(Red Hat Enterprise Linux CoreOS) 노드에서 sosreport를 수집할 수 있습니다. 사용자는 sos-collector를 실행하는 방식을 변경할 필요가 없습니다. 노드가 RHCOS 또는 RHEL인 경우 자동으로 식별됩니다.

- **RHCOS** 노드에서 수집할 때 **sos-collector** 는 노드에 임시 컨테이너를 생성하고 **support-tools** 컨테이너를 사용하여 **sosreport**를 생성합니다. 이 컨테이너는 완료 후 제거됩니다.
- **--cluster-type=none** 옵션을 사용하면 사용자가 노드에서 실행되는 **sosreport** 명령에 대한 모든 클러스터 관련 검사 또는 수정 사항을 건너뛰고 **--nodes** 매개변수를 통해 전달되는 정적 노드 목록에서만 수집할 수 있습니다.
- **Red Hat Satellite**는 이제 **Satellite** 및 모든 **Capsule**에서 **sosreport**를 수집할 수 있도록 지원되는 클러스터 유형입니다.

(BZ#1695764)

업그레이드된 컴파일러 툴 세트

Application Streams로 배포된 다음 컴파일러 툴셋이 **RHEL 8.0.1**로 업그레이드되었습니다.

- **Rust** 프로그래밍 언어 컴파일러 **rustc**, **loading** 빌드 도구 및 종속성 관리자를 제공하는 **Rust Toolset**, 버전 1.35에 필요한 라이브러리
- **Go (golang)** 프로그래밍 언어 도구 및 라이브러리를 버전 1.11.6에 제공하는 **go Toolset**.

(BZ#1731500)

SMT 활성화 및 비활성화

이제 **RHEL 8**에서 동시 멀티 스레딩(**SMT**) 구성을 사용할 수 있습니다. 웹 콘솔에서 **SMT**를 비활성화 하면 다음과 같은 **CPU** 보안 취약점을 완화할 수 있습니다.

- [마이크로 아키텍처 데이터 샘플링](#)
- [L1 터미널 공격 \(Fault Attack\)](#)

(BZ#1713186)

4.2. 확인된 문제

IPSec 터널의 성능 저하

aes256_sha2 또는 **RHEL 8.0.1**에 설정된 **aes-gcm256 IPSec** 암호를 사용하면 **IPSec** 터널에 부정적인 영향을 미칩니다. 특정 **VPN** 설정을 가진 사용자는 **IPSec** 터널의 성능 저하를 경험할 수 있습니다. 이 회귀는 **Microarchitectural Data Sampling (MDS)** 완화 조치로 인해 발생하지 않습니다. 이 완화 방법은 **on** 및 **off**에서 확인할 수 있습니다.

(BZ#1731362)

5장. RHEL 8.0.0 릴리스

5.1. 새로운 기능

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

5.1.1. 웹 콘솔



참고

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

웹 콘솔에 방화벽 인터페이스가 추가

RHEL 8 웹 콘솔의 네트워킹 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 설치가 아닌 설치에서는 **RHEL 8**에 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-advise** 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)**5.1.2. 설치 프로그램 및 이미지 생성**

IBM Z에서 **SE** 및 **HMC**를 사용하여 **DVD**에서 **RHEL**을 설치하는 기능을 완전 지원

IBM Z 하드웨어에서 **SE**(Support Element) 및 **HMC**(Hardware Management Console)를 사용하여 **DVD**에서 **Red Hat Enterprise Linux 8**을 설치하는 기능은 이제 완전하게 지원됩니다. 이와 같은 기능이 추가되어 **SE**와 **HMC**를 사용하여 **IBM Z**에 설치하는 프로세스가 간소화되었습니다.

바이너리 **DVD**에서 부팅할 때, 설치 프로그램에서 사용자에게 추가 커널 매개 변수를 입력하라는 메시지가 표시됩니다. **DVD**를 설치 소스로 설정하려면 **inst.repo=hmc**를 커널 매개 변수에 추가하십시오. 그러면 설치 프로그램을 사용하여 **SE** 및 **HMC** 파일에 액세스할 수 있고, **DVD**에서 **stage2** 이미지를 가져와서 소프트웨어 선택을 위해 **DVD**에서 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 **Anaconda**의 사용자 정의 파티션 설정 창 또는 **Kickstart**의 **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 **Anaconda**에서 서브스크립션 관리자에게 시스템 용도 정보를 제공하지 않았습니다. **Red Hat Enterprise Linux 8.0**에서는 **Anaconda**의 **System Purpose** 창 또는 **Kickstart**의 **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 관리자에게 시스템 용도 정보를 제공할 수 없었습니다. **Red Hat Enterprise Linux 8.0**에서 **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 관리자가 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 8**에서 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 **GUI**에 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 소스에서 이를 사용하지 않게 설정 해제할 수 있습니다. 이 기능은 다

른 기본 리포지토리를 사용하여 설치를 시작한 다음 통합 **ISO**로 변경하면 작동하지 않습니다. 이렇게 하면 기본 리포지토리가 교체됩니다. 그러나 **AppStream** 리포지토리는 교체되지 않으며 원래 파일을 가리킵니다.

(BZ#1610806)

Anaconda에서는 **Kickstart** 스크립트에 모듈식 패키지를 설치할 수 있습니다.

Anaconda 설치 프로그램은 애플리케이션 스트림과 관련된 모든 기능(모듈, 스트림 및 프로필)을 처리하도록 확장되었습니다. 이제 **Kickstart** 스크립트에서 모듈 및 스트림 조합을 활성화하고 모듈 프로필을 설치하며 모듈식 패키지를 설치할 수 있습니다. 자세한 내용은 [고급 RHEL 설치](#) 수행을 참조하십시오.

(JIRA:RHELPLAN-1943)

이제 **RHEL 8** 설치 옵션에서 **nosmt** 부팅 옵션을 사용할 수 있습니다.

새로 설치된 **RHEL 8** 시스템에 전달되는 설치 옵션에서 **nosmt** 부팅 옵션을 사용할 수 있습니다.

(BZ#1677411)

RHEL 8에서는 로컬 하드 드라이브의 리포지토리에서 설치를 지원

이전에는 하드 드라이브에서 **RHEL**을 설치하려면 설치 소스로 **ISO** 이미지가 필요했습니다. 그러나 일부 파일 시스템에는 **RHEL 8 ISO** 이미지가 너무 클 수 있습니다. 예를 들어 **FAT32** 파일 시스템은 **4GiB**보다 큰 파일을 저장할 수 없습니다.

RHEL 8에서는 로컬 하드 드라이브의 리포지토리에서 설치를 활성화할 수 있습니다. **ISO** 이미지 대신 디렉터리만 지정해야 합니다. 예: `'inst.repo=hd:<device>:<path to the repository>'`

(BZ#1502323)

RHEL 8에서 이미지 빌더를 사용하여 사용자 정의 시스템 이미지 생성 가능

Image Builder 도구를 사용하면 사용자가 사용자 지정 **RHEL** 이미지를 만들 수 있습니다. 이미지 빌더는 **lorax-composer** 패키지의 **AppStream**에서 사용할 수 있습니다.

Image Builder를 사용하면 추가 패키지를 포함하는 사용자 지정 시스템 이미지를 만들 수 있습니다. 이미지 빌더 기능은 다음을 통해 액세스할 수 있습니다.

- 웹 콘솔의 그래픽 사용자 인터페이스
- **composer-cli** 툴의 명령줄 인터페이스입니다.

Image Builder 출력 형식은 다음과 같습니다.

- 라이브 ISO 디스크 이미지
- **qcow2** 파일은 가상 머신 또는 **OpenStack**과 직접 사용할 수 있습니다.
- 파일 시스템 이미지 파일
- **Azure**, **VMWare** 및 **AWS**용 클라우드 이미지

Image Builder에 대한 자세한 내용은 [사용자 지정된 RHEL 시스템 이미지 비교](#) 문서 제목을 참조하십시오.

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

새로운 **Kickstart** 명령 추가: **authselect** 및 모듈

이번 릴리스에서는 다음과 같은 **kickstart** 명령이 추가되었습니다.

- **Authselect:** **authselect** 명령을 사용하여 설치 중에 시스템 인증 옵션을 설정합니다. **authselect**를 더 이상 사용되지 않는 **auth** 또는 **authconfig** **Kickstart** 명령 대신 사용할 수 있습니다. 자세한 내용은 [고급 설치 수행 가이드의 authselect 섹션](#)을 참조하십시오.
- **module:** **module** 명령을 사용하여 **kickstart** 스크립트 내에서 패키지 모듈 스트림을 활성화합니다. 자세한 내용은 [고급 설치 수행 가이드의 모듈 섹션](#)을 참조하십시오.

(BZ#1972210)

5.1.3. 커널

RHEL 8.0의 커널 버전

Red Hat Enterprise Linux 8.0은 커널 버전 4.18.0-80과 함께 배포됩니다.

(BZ#1797671)

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 버스에는 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 **128PiB**의 가상 주소 공간과 **4PB**의 실제 메모리 용량이 있는 **57/52** 비트의 가상/실제 메모리 주소 지정까지 확장되었습니다.

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

(BZ#1485525)

kernel-signing-ca.cer이 **RHEL 8**에서 **kernel-core**로 이동

Red Hat Enterprise Linux 7의 모든 버전에서 **kernel-signing-ca.cer** 공개 키는 **kernel-doc** 패키지에 있습니다. 그러나 **Red Hat Enterprise Linux 8**에서는 **kernel-signing-ca.cer**가 모든 아키텍처의 **kernel-core** 패키지로 재배포되었습니다.

(BZ#1638465)

Spectre V2 완화 기본값이 **IBRS**에서 **Retpolines**로 변경되었습니다.

Spectre V2 취약점에 대한 기본 완화 (**CVE-2017-5715**)는 **6th Generation Intel Core Processors** 및 그 가리치 [1]이 **IBRS (Indirect Branch Restricted Speculation)**에서 **Red Hat Enterprise Linux 8**의 **Retpolines**로 변경되었습니다. **Red Hat**은 **Intel**의 권장 사항에 따라 **Linux** 커뮤니티에서 사용되는 기본값과 손실된 성능을 복구하기 위해 이러한 변경 사항을 구현했습니다. 그러나 경우에 따라 **Retpolines**를 사용하는 경우 **Spectre V2**를 완전히 완화하지 못할 수 있습니다. **Intel**의 **Retpoline** 문서 [2] 이 문서에서는 또한 공격의 위험이 낮음을 나타냅니다.

완전한 **Spectre V2** 완화가 필요한 경우 사용자는 **spectre_v2=ibrs** 플래그를 추가하여 커널 부팅 라인을 통해 **IBRS**를 선택할 수 있습니다.

Retpoline 지원을 사용하여 하나 이상의 커널 모듈이 구축되지 않은 경우 **/sys/devices/system/cpu/vulnerabilities/spectre_v2** 파일은 취약점을 나타내며 **/var/log/ECDHE** 파일은 잘못된 모듈을 식별합니다. [How to determine which modules are responsible for spectre_v2 returning "Vulnerable: Retpoline with unsafe module\(s\)"](#) 을 참조하십시오.

[1] "6세대 Intel Core Processors 및 해당 근접 제품"은 **Intel**의 **Retpolines** 문서가 "Skylake-generation"이라고 하는 것입니다.

[2] 역설: 분기별 [Mitigation - White paper](#)

(BZ#1651806)

Intel® Omni-Path Architecture (OPA) 호스트 소프트웨어

Intel Omni-Path Architecture(OPA) 호스트 소프트웨어는 Red Hat Enterprise Linux 8에서 완전 지원됩니다.

Intel OPA는 고성능 데이터 전송(high bandwidth, high message rate, low latency)과 클러스터형 환경의 I/O 노드 간에 초기화 및 설정을 지원하는 HFI(Host Fabric Interface) 하드웨어를 제공합니다.

Intel Omni-Path Architecture 설명서 설치에 대한 자세한 내용은 https://www.intel.com/content/dam/support/us/en/documents/network-and-i-o/fabric-products/Intel_OP_Software_RHEL_8_RN_K51383.pdf을 참조하십시오.

(BZ#1683712)

RHEL 8에서 더 많은 노드 지원

이번 업데이트를 통해 NUMA(Non-Uniform Memory Access) 노드 수가 64비트 ARM 아키텍처를 사용하는 시스템에서 Red Hat Enterprise Linux 8의 NUMA 노드 4개에서 8개의 NUMA 노드로 증가했습니다.

(BZ#1550498)

RHEL 8에서 IOMMU 패스스루가 기본적으로 활성화되어 있습니다.

IOMMU(Input/Output Memory Management Unit) 패스스루는 기본적으로 활성화되어 있습니다. 이렇게 하면 호스트에 대해 Direct Memory Access(DMA) 재mapping이 비활성화되어 있기 때문에 AMD 시스템의 성능이 향상됩니다. 이번 업데이트에서는 OpenJDK 재mapping도 기본적으로 비활성화되어 있는 Intel 시스템과의 일관성을 제공합니다. 사용자는 하이퍼바이저를 포함하여 커널 명령줄에서 `iommu.passthrough=off` 또는 `iommu=nopt` 매개변수를 지정하여 이러한 동작을 비활성화(및 활성화)할 수 있습니다.

(BZ#1658391)

RHEL8 커널에서 5 레벨 페이지 테이블 지원

Red Hat Enterprise Linux 커널은 이제 최대 5 레벨의 페이지 테이블이 있는 향후 Intel 프로세서를 완벽하게 지원합니다. 이를 통해 프로세서는 최대 4PB의 물리적 메모리와 128PB의 가상 주소 공간을 지

원할 수 있습니다. 대용량 메모리를 활용하는 애플리케이션은 이제 4 레벨 페이지 테이블의 제약 조건 없이 시스템에서 제공하는 최대한 많은 메모리를 사용할 수 있습니다.

(BZ#1623590)

RHEL8 커널은 향후 Intel CPU에서 향상된 IBRS 지원

Red Hat Enterprise Linux 커널은 향상된 IBRS(Indirect Branch Restricted Speculation) 기능을 사용하여 Spectre V2 취약점을 완화할 수 있습니다. 이를 활성화하면 IBRS는 스펙터 V2를 완화하기 위해 Retpolines(기본값)보다 더 잘 수행하고 Intel Control-flow Enforcement 기술을 방해하지 않습니다. 결과적으로 향후 Intel CPU에서 Spectre V2 완화 기능을 활성화하면 성능이 저하됩니다.

(BZ#1614144)

eBPF 기반 프로그램 및 맵의 검사 및 조작을 위해 bpftool 추가

eBPF(extended Berkeley Packet Filtering)를 기반으로 프로그램과 맵의 검사 및 간단한 조작에 사용되는 bpftool 유틸리티가 Linux 커널에 추가되었습니다. bpftool 은 커널 소스 트리의 일부이며 커널 패키지의 하위 패키지로 포함된 bpftool 패키지에서 제공합니다.

(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 개선사항이 추가되었습니다.

(BZ#1592977)

5.1.4. 소프트웨어 관리

YUM 성능 개선 및 모듈식 콘텐츠 지원

Red Hat Enterprise Linux 8에서는 DNF 기술(YUM v4)을 기반으로 하는 YUM 툴의 새 버전으로 소프트웨어를 설치합니다.

YUM v4 는 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 v4** 툴과 **RHEL 7**의 이전 버전 **YUM v3**의 차이점에 대한 자세한 내용은 [YUM과 비교하여 DNF CLI의 변경 사항](#)을 참조하십시오.

YUM v4는 명령줄에서 사용하는 경우 **YUM v3**과 호환되며 구성 파일을 편집하거나 생성합니다.

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

선택한 **yum** 플러그인 및 유틸리티가 새로운 **DNF** 백엔드에 이식되었으며 **RHEL 7**에서와 동일한 이름으로 설치할 수 있습니다. 또한 호환성 심볼릭 링크를 제공하므로 바이너리, 구성 파일 및 디렉터리는 일반적인 위치에서 찾을 수 있습니다.

YUM v3에서 제공하는 레거시 **Python API**는 더 이상 사용할 수 없습니다. 사용자는 안정적이고 완전히 지원되는 **YUM v4 (DNF Python API)**에서 제공하는 새 **API**로 플러그인 및 스크립트를 마이그레이션하는 것이 좋습니다. **DNF Python API**는 [DNF API 참조](#)에서 사용할 수 있습니다.

Libdnf 및 **Hawkey API**(C 및 Python 모두)는 불안정하며 **Red Hat Enterprise Linux 8** 라이프 사이클 기간 동안 변경될 수 있습니다.

YUM 패키지 및 툴 가용성에 대한 자세한 내용은 [RHEL 8 도입 시 고려 사항](#)을 참조하십시오.

YUM v3 기능 중 일부는 **YUM v4**에서 다르게 작동할 수 있습니다. 이러한 변경이 워크플로우에 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 4.14**와 함께 배포됩니다. 이 버전에서는 **RHEL 7**에서 사용할 수 있는 **RPM 4.11**에 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)

RPM 이 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 손상된 경우 이전 버전의 일부 파일을 교체한 후 패키지 업그레이드가 중단되어 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 페이로드에서 새로운 **SHA-256** 해시를 사용합니다. 서명된 패키지에서 페이로드 해시는 서명으로 추가 보호되므로, 패키지 헤더에서 서명과 기타 해시를 손상시키지 않고 변경할 수 없습니다. 이전 패키지는 구성에 의해 비활성화되지 않는 한 헤더와 페이로드의 **MD5** 해시를 사용합니다.

%_pkgverify_level 매크로를 사용하여 설치 전에 추가로 서명 확인을 강제 시행하거나 페이로드 확인을 완전히 비활성화할 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-10596)

5.1.5. 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 **Tuned**가 실행 중이면 **atomic-host** 프로파일이 선택됩니다.
 - 가상 머신에서 **Tuned**가 실행 중이면 **atomic-guest** 프로파일이 선택됩니다.

- 가상 머신에서 **Tuned**가 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 Linux**에서 읽기 전용인 작업 디렉터리에 저장했습니다. 이번 업데이트에서는 선택한 파일의 경로가 쓰기가 허용된 하위 디렉터리로 변경되었습니다. 이제 기본 디렉터리 **Unix**와 **SELinux** 권한을 사용하면 디렉터리에 쓸 수 있습니다. 디렉터리에서 배포된 파일은 여전히 **named**에 대해 읽기 전용입니다.

(BZ#1588592)

Geolite Database는 **Geolite2 Database**로 대체

Red Hat Enterprise Linux 7에 있는 **Geolite Database**는 **Red Hat Enterprise Linux 8**의 **Geolite2 Database**로 대체되었습니다.

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

Geolite2 Database는 여러 패키지에서 제공합니다. **libmaxminddb** 패키지에는 주소를 수동으로 검색하는 데 사용할 수 있는 **mmdblookup** 명령행 툴과 라이브러리가 포함되어 있습니다. **libmaxminddb** 패키지에는 주소를 수동으로 검색할 수 있는 **library** 및 **mmdblookup** 명령줄 도구가 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 **journald**에서 처리

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

(JIRA:RHELPLAN-12764)

RHEL 8의 주요 BIND 기능

RHEL 8에는 9.11 버전의 **BIND**(Berkeley Internet Name Domain)가 포함되어 있습니다. 이 **DNS** 서버 버전에서는 버전 9.10에 비해 여러 가지 새로운 기능 및 기능 변경 사항이 도입되었습니다.

새로운 기능:

- **Catalog Zones** 라는 보조 서버를 프로비저닝하는 새로운 방법이 추가되었습니다.
- 이제 **Domain Name System** 쿠키는 **named** 서비스와 **dig** 유틸리티에 의해 전송됩니다.
- 이제 응답 속도 제한 기능이 **DNS** 수정 공격을 완화하는 데 도움이 될 수 있습니다.
- **response-policy** 영역(**RPZ**)의 성능이 향상되었습니다.
- **map** 이라는 새 영역 파일 형식이 추가되었습니다. 이 형식으로 저장된 영역 데이터는 메모리에 직접 매핑할 수 있으므로 영역이 훨씬 빠르게 로드될 수 있습니다.
- **delv** (도메인 엔티티 조회 및 유효성 검사)라는 새로운 도구가 추가되어 **DNS** 데이터를 찾고 내부 **DNSSEC**(**DNSSEC**) 검증을 수행하기 위한 **dig**와 유사한 의미가 추가되었습니다.
- 이제 새로운 **mdig** 명령을 사용할 수 있습니다. 이 명령은 여러 파이프라인 쿼리를 보낸 후 다음 쿼리를 보내기 전에 하나의 쿼리를 보내고 응답을 기다리는 대신 응답을 대기하는' 명령의 버전입니다.
- 재귀 확인자 성능을 향상시키는 새로운 **prefetch** 옵션이 추가되었습니다.

- 보기 간에 영역 데이터를 공유할 수 있는 새로운 뷰 영역 옵션이 추가되었습니다. 이 옵션을 사용하면 여러 뷰가 메모리에 여러 복사본을 저장하지 않고 권한 있는 동일한 영역을 제공할 수 있습니다.
- 영역의 최대 TTL을 적용하는 새로운 **max-zone-ttl** 옵션이 추가되었습니다. 높은 TTL을 포함하는 영역이 로드되면 로드가 실패합니다. TTL이 높은 **DDNS(Dynamic DNS)** 업데이트는 허용되지만 TTL은 잘립니다.
- 재귀 확인자가 보낸 쿼리를 제한하기 위해 새로운 할당량이 추가되어 서비스 거부 공격을 받을 수 있는 권한 있는 서버로 표시됩니다.
- 이제 **nslookup** 유틸리티는 기본적으로 **IPv6** 및 **IPv4** 주소를 모두 조회합니다.
- 이제 **named** 서비스에서 시작하기 전에 다른 이름 서버 프로세스가 실행 중인지 확인합니다.
- 서명된 영역을 로드할 때 **named** 는 이제 리소스 레코드 서명(**RSIG**)의 초기 시간이 나중에 있는지 확인하고 있으면 **RRSIG**를 즉시 다시 생성합니다.
- 이제 영역 전송에서 더 작은 메시지 크기를 사용하여 메시지 압축을 개선하여 네트워크 사용량을 줄입니다.

기능 변경 사항:

- 더 빠른 구문 분석을 위해 새로운 통계와 병합된 **XML** 트리를 포함한 통계 채널에 대한 버전 **3 XML** 스키마는 **HTTP** 인터페이스에서 제공합니다. 레거시 버전 **2 XML** 스키마는 더 이상 지원되지 않습니다.
- 이제 **named** 서비스는 기본적으로 **IPv6** 및 **IPv4** 인터페이스에서 수신 대기합니다.
- 명명된 서비스는 더 이상 **GeoIP**를 지원하지 않습니다. 쿼리 발신자의 미리 사용된 위치에 정의된 **ACL(액세스 제어 목록)**을 사용할 수 없습니다.

(JIRA:RHELPLAN-1820)

5.1.6. 셸 및 명령행 툴

nobody 사용자가 **nfsnobody**로 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.

둘 다 Red Hat Enterprise Linux 8에서 65534 ID를 사용하는 **nobody** 사용자 및 그룹 쌍으로 병합되었습니다. 새로운 설치에 더 이상 **nfsnobody** 쌍을 생성하지 않습니다.

이와 같이 변경하면 **nobody**에서 소유한 파일에 관한 혼동은 줄어들지만 **NFS**와는 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.

RHEL 7에서 사용 가능한 **CVS(Concurrent Versions System)** 및 버전 제어 시스템(**RCS**)은 RHEL 8에서는 배포되지 않습니다.

(BZ#1693775)

Subversion 1.10의 주요 변경 사항

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

(BZ#1571415)

dstat의 주요 변경 사항

RHEL 8은 새로운 버전의 **dstat** 툴과 함께 배포됩니다. 이 도구는 이제 **PCP(Performance Co-inspector)** 툴킷의 일부입니다. **/usr/bin/dstat** 파일과 **dstat** 패키지 이름은 이제 **pcp-system-tools** 패키지에서 제공합니다.

새로운 버전의 **dstat** 는 **RHEL 7**에서 사용 가능한 **dstat** 에 대해 다음과 같은 향상된 기능을 제공합니다.

- **python3** 지원
- 기록 분석

- 원격 호스트 분석
- 구성 파일 플러그인
- 새로운 성능 지표

([BZ#1684947](#))

5.1.7. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

Python 3은 RHEL 8의 기본 Python 구현

Red Hat Enterprise Linux 8은 Python 3.6과 함께 배포됩니다. 기본적으로 패키지가 설치되지 않을 수 있습니다. Python 3.6을 설치하려면 `yum install python3` 명령을 사용합니다.

Python 2.7은 `python2` 패키지에서 사용할 수 있습니다. Python 2는 라이프사이클이 더 짧기 때문에 고객이 더 원활하게 Python 3으로 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 [Python 소개](#)를 참조하십시오.

([BZ#1580387](#))

Python 스크립트는 RPM 빌드 시 인터프리터 지시문에 주요 버전을 지정해야 합니다.

RHEL 8에서는 실행 가능한 Python 스크립트는 주요 Python 버전을 명시적으로 지정하는 인터프리터 지시문(hashbangs)을 사용해야 합니다.

RPM 패키지를 빌드할 때 `/usr/lib/rpm/redhat/brp-mangle-shebangs` BRP(Buildroot 정책) 스크립트가 자동으로 실행됩니다. 이 스크립트는 모든 실행 파일에서 인터프리터 지시문을 수정하려고 시도합니다. 스크립트에서 Python의 주요 버전을 지정하지 않는 모호한 Python 인터프리터 지시문이 발생하면 오류가 생성되고 RPM 빌드가 실패합니다. 이러한 모호한 인터프리터 지시문의 예는 다음과 같습니다.

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

RPM 빌드 시 이러한 빌드 오류를 유발하는 **Python** 스크립트의 인터프리터 지시문을 수정하려면 **platform-python-devel** 패키지에서 **pathfix.py** 스크립트를 사용하십시오.

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

여러 개의 **PATH**를 지정할 수 있습니다. **PATH**가 디렉터리인 경우 **pathfix.py**는 모호한 **hashbang**을 사용하는 것이 아니라 `^[a-zA-Z0-9_]+\.[py]$` 패턴과 일치하는 **Python** 스크립트를 반복적으로 검사합니다. **pathfix.py** 실행을 위한 명령을 **%prep** 섹션 또는 **%install** 섹션의 끝에 추가합니다.

자세한 내용은 [Python 스크립트의 인터프리터 지시문](#) 처리를 참조하십시오.

(BZ#1583620)

PHP에서 주요 변경 사항

Red Hat Enterprise Linux 8은 **PHP 7.2**와 함께 배포됩니다. 이 버전에서는 **RHEL 7**에서 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 오류 및 경고는 **/var/log/httpd/error.log** 대신 **/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.
- **zip** 확장 기능이 **php-common** 패키지에서 별도의 패키지인 **php-pecl-zip**으로 이동

다음의 확장 기능이 제거되었습니다.

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

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

Ruby에서 주요 변경 사항

RHEL 8에서는 **RHEL 7**에서 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 7**에서 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 **.이 @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 pragma**의 기본 모드는 더 이상 지원되지 않습니다. **pragma**의 기본 모드 인코딩 은 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. **perl** 패키지는 코어 모듈도 설치하지만 **/usr/bin/perl** 인터프리터는 **perl-interpreter** 패키지에서 제공됩니다. 이전 릴리스에서는 **perl** 패키지에 최소한의 인터프리터만 포함되어 있었지만 **perl-core** 패키지에는 인터프리터와 코어 모듈이 모두 포함되어 있었습니다.

•

IO::Socket::SSL Perl 모듈에서는 더 이상 `./certs/my-ca.pem` 파일 또는 `./ca` 디렉터리에서 인증 기관 인증서를 로드하지 않습니다. `./certs/server-key.pem` 파일의 서버 개인 키, `./certs/server-cert.pem` 파일의 서버 인증서, `./certs/client-key.pem` 파일의 클라이언트 개인 키와 `./certs/client-cert.pem` 파일의 클라이언트 인증서입니다. 대신 파일에 대한 경로를 명시적으로 지정합니다.

(BZ#1511131)

RHEL의 Node.js 새로운 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** 표준에 대한 지원이 구현됩니다. 이제 **SWIG**는 **Go 1.6**, **PHP 7**, **Octave 4.2** 및 **Python 3.5**도 지원합니다.

(BZ#1660051)

Apache httpd에서 주요 변경 사항

RHEL 8은 **Apache HTTP Server 2.4.37**과 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 Server**에서는 **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 Server**와 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.

httpd 및 해당 사용법에 대한 자세한 내용은 [Setting up the Apache HTTP 웹 서버](#) 을 참조하십시오.

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

RHEL에서 새로 추가된 nginx 웹 서버

RHEL 8에서는 **HTTP** 및 기타 프로토콜을 지원하는 웹 및 프록시 서버인 **nginx 1.14** 를 도입하고 높은 동시성, 성능 및 낮은 메모리 사용에 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 Redis가 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.
- 재귀 및 비재귀 **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 8**을 통해 배포된 **MySQL**과 **MariaDB** 서버에 구축되는 애플리케이션 중 하나에 연결할 수 있습니다.

또한 **RHEL 8**의 클라이언트 툴과 라이브러리는 기본적으로 업스트림 **MySQL 8.0** 버전에서 사용되는 **caching_sha2_password** 방법과 호환되지 않으므로, **RHEL 8**을 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
```

[MySQL 사용](#)을 참조하십시오.

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

MariaDB 10.3에서 주요 변경 사항

MariaDB 10.3에서는 **RHEL 7**에서 배포된 버전 **5.5**에 비해 다음과 같은 새로운 기능을 제공합니다.

-

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 8**을 통해 배포된 **MySQL**과 **MariaDB** 서버에 구축되는 애플리케이션 중 하나에 연결할 수 있습니다.

기타 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
- **mariadb-bench** 하위 패키지가 제거되었습니다.
- 플러그인 완성의 기본 허용 수준이 서버 완성도보다 한 수준 미만으로 변경되었습니다. 결과적으로 이전에 작동하던 완성 수준이 낮은 플러그인이 더 이상 로드되지 않습니다.

MariaDB 사용을 참조하십시오.

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

PostgreSQL에서 주요 변경 사항

RHEL 8.0은 **postgresql** 모듈의 두 가지 스트림으로 배포된 **PostgreSQL** 데이터베이스 서버 두 가지 버전을 제공합니다. **PostgreSQL 10** (기본 스트림)과 **PostgreSQL 9.6**. **RHEL 7**에는 **PostgreSQL** 버전

9.2가 포함되어 있습니다.

PostgreSQL 9.6 의 주요 변경 사항은 다음과 같습니다.

- 순차적 작업의 병렬 실행: 검사,조인, 집계
- 동기 복제 기능 개선
- 사용자가 문구를 검색할 수 있도록 전체 텍스트 검색 개선
- **postgres_fdw** 데이터 페더레이션 드라이버에서 원격 조인,정렬,업데이트 및 **DELETE** 작업을 지원합니다.
- 멀티 CPU 소켓 서버의 확장성과 관련하여 상당한 성능 향상

PostgreSQL 10 의 주요 개선 사항은 다음과 같습니다.

- **publish** 및 **subscribe** 키워드를 사용한 논리 복제
- **SCRAM-SHA-256** 메커니즘을 기반으로 한 강력한 암호 인증
- 선언적 테이블 파티셔닝
- 쿼리 병렬 처리 개선
- 상당한 일반 성능 향상
- 향상된 모니터링 및 제어

[PostgreSQL 사용](#)을 참조하십시오.

(BZ#1660041)

Squid에서 주요 변경 사항

RHEL 8.0은 웹 클라이언트용 고성능 프록시 캐싱 서버, **FTP**, **Gopher** 및 **HTTP** 데이터 오브젝트를 지원하는 **Squid 4.4** 와 함께 배포됩니다. 이 릴리스에서는 **RHEL 7**에서 사용할 수 있는 버전 **3.5**에 대한 다양한 새로운 기능, 개선 사항 및 버그 수정을 제공합니다.

주요 변경 사항은 다음과 같습니다.

- 구성 가능한 도우미 대기열 크기
- 도우미 동시성 채널 변경
- 도우미 바이너리의 변경 사항
- **ICAP(Secure Internet Content Adaptation Protocol)**
- **Symmetric Multi Processing (SMP)**에 대한 지원 개선
- 프로세스 관리 개선
- **SSL** 지원 삭제
- **Edge Side Includes (ESI)** 사용자 정의 구문 분석
- 여러 구성 변경

(BZ#1656871)

RHEL의 Varnish 캐시 새로운 기능

RHEL에서 처음으로 고성능 HTTP 역방향 프록시인 **Varnish** 캐시 가 제공됩니다. 이전에는 소프트웨어 컬렉션으로만 제공되었습니다. **Varnish Cache** 는 향후 동등한 요청에 대한 응답 시간 및 네트워크 대역폭 소비를 줄이는 데 사용되는 파일 또는 파일 내용을 메모리에 저장합니다. **RHEL 8.0**은 **Varnish Cache 6.0** 과 함께 배포됩니다.

(BZ#1633338)

5.1.8. 데스크탑

RHEL 8의 GNOME 셸 (버전 3.28)

RHEL(Red Hat Enterprise Linux) 8에서는 **GNOME** 셸 버전 **3.28**을 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.

Wayland 는 **X.org** 에 비해 다양한 이점과 개선사항을 제공합니다. 특히:

- 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 Wayland에서 지원되지 않습니다.
- NVIDIA 바이너리 드라이버는 Wayland에서 작동하지 않습니다.
- xrandr 유틸리티는 해상도, 회전 및 레이아웃 처리 방법이 다르기 때문에 Wayland에서 작동하지 않습니다. 화면을 조작하는 기타 X.org 유틸리티도 Wayland에서 작동하지 않습니다.
- 화면 녹화, 원격 데스크탑 및 접근성은 Wayland에서 제대로 작동하지 않습니다.
- 클립보드 관리자는 사용할 수 없습니다.
- Wayland에서는 가상 머신 뷰어와 같이 X11 애플리케이션의 키보드 그랩(**grab**)을 무시합니다.
- 게스트 가상 머신(VM)의 Wayland에는 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.org를 사용합니다. 다음 그래픽 드라이버를 사용하는 경우 시스템에서 X.org로 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 대한 자세한 내용은 <https://wayland.freedesktop.org/> 을 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 8**에서는 애플리케이션과

gnome-shell 확장 기능을 설치 및 업데이트할 수 있는 **GNOME** 소프트웨어 유틸리티에서 유사한 기능을 제공합니다. **GNOME** 소프트웨어는 **gnome-software** 패키지로 배포됩니다.

(JIRA:RHELPLAN-3001)

Wayland에서 GNOME Shell에서 사용 가능한 부분 확장

Wayland 세션의 GNOME Shell에서 부분 확장 기능을 사용할 수 있습니다. 이 기능을 사용하면 GUI를 부분적으로 확장할 수 있으므로 특정 디스플레이에서 확장된 GUI의 모양을 개선할 수 있습니다.

이 기능은 현재 실험적인 것으로 간주되므로 기본적으로 비활성화되어 있습니다.

소수 부분 스케일링을 사용하려면 다음 명령을 실행합니다.

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

(BZ#1668883)

5.1.9. 하드웨어 지원

fwupd를 사용하여 펌웨어 업데이트 사용 가능

RHEL 8에서는 **fwupd** 데몬을 사용하여 **UEFI Capsule**, **Device Firmware Upgrade(DFU)** 등의 펌웨어 업데이트를 지원합니다. 데몬을 사용하면 세션 소프트웨어가 로컬 시스템에서 장치 펌웨어를 자동으로 업데이트할 수 있습니다.

업데이트를 보고 적용하려면 다음을 사용할 수 있습니다.

- GUI 소프트웨어 관리자(예: GNOME Software)
- **fwupdmgr** 명령줄 툴

메타데이터 파일은 **Linux Vendor Firmware Service(LVFS)** 보안 포털에서 자동으로 다운로드되고 **D-Bus**를 통해 **fwupd**에 제출됩니다. 적용해야 하는 업데이트를 다운로드하면 사용자 알림 및 업데이트

세부 정보가 표시됩니다. 업데이트를 수행하기 전에 사용자가 펌웨어 업데이트 작업에 명시적으로 동의해야 합니다.

LVFS에 대한 액세스는 기본적으로 비활성화되어 있습니다.

LVFS에 대한 액세스를 활성화하려면 **GNOME** 소프트웨어의 소스 대화 상자에서션을 클릭하거나 **fwupdmg** **enable-remote lvfs** 명령을 실행합니다. **fwupdmg** 을 사용하여 업데이트 목록을 가져오는 경우 **LVFS**를 활성화할지 묻는 메시지가 표시됩니다.

LVFS에 액세스하면 하드웨어 공급 업체에서 직접 펌웨어 업데이트를 받을 수 있습니다. 이러한 업데이트는 **Red Hat InfiniBand**에서 확인하지 않았습니다.

(BZ#1504934)

Optane DC 영구 메모리 기술의 메모리 모드가 완전히 지원됩니다.

Intel Optane DC 영구 메모리 스토리지 장치는 데이터 센터 클래스 영구 메모리 기술을 제공하므로 트랜잭션 처리량을 크게 높일 수 있습니다.

메모리 모드 기술을 사용하려면 시스템에 특별한 드라이버 또는 특정 인증이 필요하지 않습니다. 메모리 모드는 운영 체제에 투명합니다.

(BZ#1718422)

5.1.10. IdM (Identity Management)

Directory Server에서 새 암호 구문 검사

이번 개선된 기능에는 **Directory Server**에 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 Server**의 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 **ID**는 항상 **-1**로 설정되었습니다. 이번 개선된 기능을 통해 **Directory Server**는 실제 연결 및 작업 **ID**를 기록합니다. 이제 이 작업을 초래한 서버 또는 클라이언트 작업에 대한 내부 작업을 추적할 수 있습니다.

(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 **URL**을 사용하여 **OCSP** 검사를 설정할 수 있습니다.

(BZ#1636564)

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 **CSR**을 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 **CA**에서는 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 **CSR**을 생성할 수 있습니다.

(BZ#1656856)

SSSD에서 더 이상 **[nss]** 섹션의 **fallback_homedir** 값을 **AD** 도메인의 폴백으로 사용하지 않습니다.

RHEL 7.7 이전에는 **AD**(Active Directory) 공급자의 **SSSD fallback_homedir** 매개변수에 기본값이 없었습니다. **fallback_homedir** 을 설정하지 않은 경우 **SSSD**는 대신 **/etc/sss/sss.conf** 파일의 **[nss]** 섹션과 동일한 매개변수의 값을 사용합니다. 보안을 강화하기 위해 **RHEL 7.7**의 **SSSD**는 **fallback_homedir** 의 기본값을 도입했습니다. 결과적으로 **SSSD**는 더 이상 **[nss]** 섹션에 설정된 값으로 되돌아가지 않습니다. **AD** 도메인의 **fallback_homedir** 매개변수에 대한 기본값과 다른 값을 사용하려면 도메인의 섹션에서 수동으로 설정해야 합니다.

(BZ#1652719)

SSSD를 사용하여 여러 스마트 카드 인증 장치 중 하나를 선택 가능

기본적으로 **SSSD**(System Security Services Daemon)는 스마트 카드 인증 장치를 자동으로 탐지합니다. 연결된 장치가 여러 개 있는 경우 **SSSD**에서 감지한 첫 번째 장치를 선택합니다. 결과적으로 특정 장치를 선택할 수 없으므로 때때로 실패로 이어집니다.

이번 업데이트를 통해 **sss.conf** 설정 파일의 **[pam]** 섹션에 대해 새로운 **p11_uri** 옵션을 구성할 수 있습니다. 이 옵션을 사용하면 스마트 카드 인증에 사용되는 장치를 정의할 수 있습니다.

예를 들어 **OpenSC PKCS#11** 모듈에서 감지한 슬롯 **ID 2** 가 있는 리더를 선택하려면 다음을 추가합니다.

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

sss.conf의 **[pam]** 섹션에 추가합니다.

자세한 내용은 **man sss.conf** 페이지를 참조하십시오.

(BZ#1620123)

로컬 사용자는 **SSSD**를 통해 캐시하고 **nss_sss** 모듈을 통해 처리

RHEL 8에서 **SSSD(System Security Services Daemon)**는 기본적으로 **/etc/passwd** 및 **/etc/groups** 파일의 사용자 및 그룹을 제공합니다. **sss nsswitch** 모듈은 **/etc/nsswitch.conf** 파일의 파일 앞에 옵니다.

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 **SSSD**와 **nscd** 모두 자신의 고유한 독립 캐싱을 사용하므로 **SSSD**와 **nscd**를 동시에 사용하는 것이 쉽지 않을 수 있습니다. 결과적으로 **SSSD**에서 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 8**에서 더 빠르게 로컬 사용자와 그룹을 분석할 수 있습니다. **root** 사용자는 **SSSD**에서 처리하지 않으므로 **root** 분석은 **SSSD**에 있는 잠재적 버그의 영향을 받을 수 없습니다. 또한 **SSSD**가 실행되지 않으면 **nss_sss** 모듈에서 문제를 해결하기 위해 **nss_files**로 대체하여 상황을 단계적으로 적절하게 처리합니다. **SSSD**가 실행되고 있지 않은 경우에도 **nss_sss** 모듈은 문제를 방지하기 위해 **nss_files**로 대체하여 상황을 정상적으로 처리합니다. 어떠한 방식으로든 **SSSD**를 설정할 필요가 없으며 파일 도메인이 자동으로 추가됩니다.

(JIRA:RHELPLAN-10439)

KCM은 **KEYRING**을 기본 인증 정보 캐시 스토리지로 대체

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

이번 업데이트를 통해 **RHEL 8**에는 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 Management**를 관리 가능

이번 업데이트를 통해 **RHEL 8**에서는 **IdM(Identity Management)** 그룹의 구성원으로 **AD(Active Directory)** 사용자에 대한 사용자 **ID** 재정의의 추가할 수 있습니다. **ID** 재정의는 특정 **ID** 보기(이 경우 기

본 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 UI**의 셀프 서비스 기능을 사용하여 **SSH** 키를 업로드하거나 개인 데이터를 변경하는 등의 작업을 수행할 수 있습니다. **AD** 관리자가 두 개의 서로 다른 계정 및 암호 없이도 **IdM**을 완전히 관리할 수 있습니다. 현재 **AD** 사용자는 **IdM**에서 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 클라이언트를 설치하는 데 필요한 패키지가 모듈로 제공됩니다. 클라이언트 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 다음을 수행합니다.

1.
스트림을 활성화합니다.
2.
스트림을 통해 전달되는 **RPM**으로 전환합니다.

3.

yum module install idm:DL1/profile_name 명령을 실행합니다.

특정 스트림을 이미 활성화하여 패키지를 다운로드한 후 새 모듈 스트림으로 전환하려면 다음을 수행합니다.

1.

설치된 모든 콘텐츠를 제거하고 현재 모듈 스트림을 비활성화합니다.

2.

새 모듈 스트림을 활성화합니다.

(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. **Authselect**는 시스템 관리자가 **PAM** 구성을 더 쉽게 변경할 수 있도록 하는 **PAM** 스택 관리에 대한 보다 안전한 접근법을 제공합니다. **Authselect**는 암호, 인증서, 스마트 카드 및 지문과 같은 인증 방법을 구성하는 데 사용할 수 있습니다. **authselect**는 원격 도메인을 결합하는 데 필요한 서비스를 구성하지 않습니다. 이 작업은 **realmd** 또는 **ipa-client-install** 과 같은 특수 도구에서 수행됩니다.

(JIRA:RHELPLAN-10445)

SSSD가 이제 기본적으로 AD VDDK를 적용

SSSD 옵션 `ad_gpo_access_control` 의 기본 설정이 강제 적용되었습니다. RHEL 8에서는 SSSD가 기본적으로 **Active Directory** 그룹 정책 개체(GPO)에 따라 액세스 제어 규칙을 적용합니다.

RHEL 7에서 RHEL 8로 업그레이드하기 전에 **Active Directory**에서 VDDK를 올바르게 구성하는 것이 좋습니다. **NetNamespaces**를 적용하지 않으려면 `/etc/sss/sss.conf` 파일의 `ad_gpo_access_control` 옵션 값을 허용 으로 변경합니다.

(JIRA:RHELPLAN-51289)

5.1.11. 컴파일러 및 개발 도구

Boost가 1.66 버전으로 업데이트

Boost C++ 라이브러리가 업스트림 버전 1.66으로 업데이트되었습니다. Red Hat Enterprise Linux 7에 포함된 Boost는 1.53 버전입니다. 자세한 내용은 업스트림 변경 로그 <https://www.boost.org/users/history/>을 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 API**가 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 **11.0.0**을 지원하도록 업데이트되었습니다. 결과적으로 문자 세트 간 음역 및 변환을 포함한 모든 와이드 문자 및 멀티 바이트 문자 **API**는 이 표준을 준수하는 정확한 정보를 제공합니다.

(BZ#1512004)

boost 패키지는 **Python**에서 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 **GCC**에 **retpolines** 지원이 추가되었습니다. 재구성은 **CVE-2017-5715**에 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 v8.2**에서 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 개선사항은 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 128비트 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 **3.11**은 **Red Hat Enterprise Linux 8**에서 **cmake** 패키지로 사용 가능합니다.

(BZ#1590139, BZ#1502802)

make 버전 4.2.1

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

- 레시피에 실패하면 **makefile**의 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 makefile**과의 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 **makefile**의 이동성이 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.
- **makefile**의 **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 **makefile**에 있는 것처럼 **-p** 옵션의 출력에 표시됩니다.

(BZ#1641015)

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 SystemTap을 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 **IBM Z** 아키텍처 확장에 대한 지원이 개선되었습니다.

링크:

- 링크는 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 추가되었습

니다.

- 이전에는 링커에서 **Intel IBT(Indirect Branch Tracking)** 기술의 잘못된 실행 코드를 생성했기 때문에 생성된 실행 파일을 시작하지 못했습니다. 결과적으로 생성된 실행 파일을 시작할 수 없었습니다. 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.
- **ARMv8-R** 아키텍처 및 **Cortex-R52**, **Cortex-M23** 및 **Cortex-M33** 프로세서에 대한 지원이 어셈블러에 추가되었습니다.

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

Performance Co-inspector 버전 4.3.0

Red Hat Enterprise Linux 8은 PCP(Performance Co-inspector) 버전 4.3.0과 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 **ustat**로 향상되었습니다.
- **pmdammv** 툴에서는 메트릭 레이블을 내보내낼 수 있으며 형식 버전은 3으로 증가했습니다.
- **pmdagfs2** 툴에서는 추가 **glock**과 **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. **pkey_alloc()**, **pkey_free()** 및 **pkey_mprotect()** 함수에 대해 새로운 **glibc** 시스템 호출 래퍼가 추가되었습니다. 또한 스레드별 보호 플래그에 액세스할 수 있도록 **pkey_set()** 및 **pkey_get()** 함수가 추가되었습니다.

(BZ#1304448)

IBM Z의 GCC의 기본값은 z13입니다.

이번 업데이트를 통해 **IBM Z** 아키텍처에서 기본적으로 **GCC**는 **z13** 프로세서용 코드를 빌드하고 코드는 **z14** 프로세서에 맞게 조정됩니다. 이는 **-march=z13** 및 **-mtune=z14** 옵션을 사용하는 것과 동일합니다. 사용자는 대상 아키텍처 및 튜닝에 대한 옵션을 사용하여 명시적으로 이 기본값을 재정의할 수 있습니다.

(BZ#1571124)

elfutils가 버전 **0.174**로 업데이트

Red Hat Enterprise Linux 8에서 **elfutils** 패키지는 버전 **0.174**에서 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.
- **CU**가 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 버그가 수정되었습니다. 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 **3.14**와 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 로드되지 않은 코드에 대한 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 기록 스택 추적의 계산을 지정하기 위해 **--delta-stacktrace** 옵션을 사용하여 확장되었습니다. 특히 이 옵션을 **--history-level=full** 과 함께 사용하면 **Helgrind** 성능을 최대 **25%** 향상할 수 있습니다.
- **Intel** 및 **AMD 64비트** 아키텍처 및 **ARM 64비트** 아키텍처에서 최적화된 코드에 대한 **Memcheck** 툴의 잘못된 긍정 속도가 감소되었습니다. **--expensive-definedness-checks** 를 사용하여 정의성 검사 처리를 제어하고 성능 비용을 절감할 수 있습니다.
- **Valgrind**에서는 **IBM Power Systems**의 **little-endian** 변형의 명령어를 더 많이 인식할 수 있습니다.
- **Valgrind**에서는 **IBM Z** 아키텍처 **z13** 프로세서의 정수 및 문자열 벡터 명령을 대부분 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 **8.2**와 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 **C**와 **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 **unwinder**로 향상되었습니다. 또한 **GDB** 구성의 **.debug_gdb_scripts** 섹션에 있는 스크립트가 자동으로 로드됩니다.
- **GDB**에서는 **Python** 버전 **3**을 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.
- **GDB**는 이제 64비트 **ARM** 아키텍처에서 확장 가능한 벡터 확장(**SVE**)을 지원합니다.
- **Intel PKU** 레지스터 및 **Intel Processor Trace**에 대한 지원이 추가되었습니다.
- **Intel** 기반 시스템에서 **rdrand** 및 **rdseed** 명령을 포함하도록 레코드 및 재생 기능이 확장되었습니다.
- **IBM Z** 아키텍처에서 **GDB**의 기능은 추적점과 빠른 추적점, 벡터 레지스터와 **ABI**, **Catch** 시스템 호출을 지원하도록 확장되었습니다. 또한 **GDB**는 아키텍처에 대한 최신 명령을 지원합니다.
- **GDB**는 64bit **ARM** 아키텍처에서 **SystemTap** 정적 사용자 공간 프로브(**SDT**)를 사용할 수 있습니다.

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

RHEL의 **glibc** 지역화는 여러 패키지로 배포

RHEL 8에서는 **glibc** 로케일 및 번역은 단일 **glibc-common** 패키지에서 더 이상 제공되지 않습니다. 대신, 모든 로케일과 언어는 **glibc-langpack-CODE** 패키지에서 사용 가능합니다. 또한 대부분의 경우 모

든 로케일이 기본적으로 설치되지는 않으며 설치 프로그램에서 선택한 로케일만 설치됩니다. 사용자는 별도로 필요한 추가 로케일 패키지를 설치하거나, **glibc-all-langpacks** 를 설치하고자 하는 경우 이전과 마찬가지로 설치된 모든 **glibc** 로케일이 포함된 로케일 아카이브를 가져올 수 있습니다.

자세한 내용은 [langpacks 사용](#)을 참조하십시오.

(BZ#1512009)

GCC 버전 8.2

Red Hat Enterprise Linux 8에서 GCC 툴체인은 GCC 8.2 릴리스 시리즈를 기반으로 합니다. 주요 변경 사항은 다음과 같습니다.

- 별칭 분석, 벡터화 개선, 동일한 코드 접기, 절차 간 분석, 저장소 병합 최적화 통과 등과 같은 다양한 일반적인 최적화가 추가되었습니다.
- **Sanitizer** 주소가 개선되었습니다. **Leak Sanitizer** 및 **Undefined Behavior Sanitizer**가 추가되었습니다.
- 이제 디버그 정보를 **DWARF5** 형식으로 생성할 수 있습니다. 이 기능은 실험적입니다.
- 소스 코드 적용 분석 도구 **GCOV**는 다양한 개선 사항으로 확장되었습니다.
- 더 많은 프로그래밍 오류를 정적 탐지하기 위해 새로운 경고 및 향상된 진단이 추가되었습니다.
- **GCC**는 생성된 코드를 추가로 강화할 수 있는 도구를 제공하도록 확장되었습니다. 보안과 관련된 개선 사항에는 오버플로 검사를 위한 기본 기능, 스택 충돌에 대한 추가 보호, 컨트롤 흐름 명령의 대상 주소 확인, 바인딩된 문자열 조작 기능에 대한 경고, 바인딩되지 않은 배열 인덱스를 감지하기 위한 경고가 포함됩니다.

아키텍처 및 프로세서 지원 개선 사항은 다음과 같습니다.

- **Intel AVX-512** 아키텍처에 대한 여러 가지 새로운 아키텍처별 옵션, 여러 마이크로 아키텍처 및 **Intel SGX(Software Wide Extensions)**가 추가되었습니다.

- 코드 생성은 이제 64비트 ARM 아키텍처 LSE 확장, ARMv8.2-A 16비트 부동 소수점 확장 (FPE) 및 ARMv8.2-A, ARMv8.3-A 및 ARMv8.4-A 아키텍처 버전을 대상으로 지정할 수 있습니다.
- IBM Z 아키텍처의 z13 및 z14 프로세서 지원이 추가되었습니다.

언어 및 표준과 관련된 주요 변경 사항은 다음과 같습니다.

- C 언어에서 코드를 컴파일할 때 사용되는 기본 표준은 GNU 확장과 함께 C17으로 변경되었습니다.
- C++ 언어에서 코드를 컴파일할 때 사용되는 기본 표준은 GNU 확장과 함께 C++14로 변경되었습니다.
- C++ 런타임 라이브러리는 이제 C++ 11 및 C++14 표준을 지원합니다.
- C++ 컴파일러는 이제 C++14 표준을 구현합니다.
- C 언어 표준 C11 지원이 개선되었습니다.
- 새로운 `__auto_type` GNU C 확장자는 C 언어로 C ++11 자동 키워드 기능의 하위 집합을 제공합니다.
- ISO/IEC TS 18661-3:2015 표준에 의해 지정된 `_FloatN` 및 `_FloatNx` 유형 이름은 이제 C 프런트 엔드에서 인식됩니다.
- 이제 빈 클래스를 인수로 전달하면 플랫폼 ABI에서 필요에 따라 Intel 64 및 AMD64 아키텍처에서 공간을 차지하지 않습니다.
- C++ 11 `alignof` 연산자가 반환한 값은 C `_Alignof` 연산자 및 반환 최소 정렬과 일치하도록 수정되었습니다. 선호하는 정렬을 찾으려면 GNU 확장 `__alignof__` 를 사용하십시오.

- **Fortran** 언어 코드의 **libgfortran** 라이브러리의 기본 버전이 **5**로 변경되었습니다.
- **Ada (GNAT)**, **GCC Go** 및 **aim C/C++** 언어에 대한 지원이 제거되었습니다. **Go** 코드 개발을 위해 **Go** 도구 세트를 사용합니다.

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

Go 암호화 라이브러리 **FIPS** 모드에서 시스템 설정 준수

이전 버전에서는 라이브러리를 사용하여 애플리케이션 빌드 시 명시적으로 비활성화한 경우를 제외하고 **Go** 표준 암호화 라이브러리는 항상 **FIPS** 모드를 사용했습니다. 그 결과 **Go** 기반 애플리케이션 사용자는 **FIPS** 모드 사용 여부를 제어할 수 없었습니다. 이 변경으로 인해 시스템이 **FIPS** 모드에서 구성되지 않은 경우 라이브러리는 기본적으로 **FIPS** 모드로 설정되지 않습니다. 결과적으로 **RHEL** 시스템의 **Go** 기반 애플리케이션 사용자는 **Go** 암호화 라이브러리의 **FIPS** 모드 사용을 제어할 수 있습니다.

(BZ#1633351)

strace는 버전 **4.24**로 업데이트

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

- 시스템 호출 변조 기능이 **-e inject=** 옵션으로 추가되었습니다. 여기에는 오류, 반환 값, 지연 및 신호 삽입이 포함됩니다.
- 시스템 호출 자격 구문이 개선되었습니다.
 - 정규 표현식을 사용하여 시스템 호출을 필터링하기 위해 **-e trace=/regex** 옵션이 추가되었습니다.
 - 자격이 시스템 호출과 일치하지 않더라도 **-e trace=** 옵션에서 시스템 호출 자격 앞에 물음표를 추가하면 **strace**가 계속됩니다.
 - **-e trace** 옵션의 시스템 호출 자격에 성격 지정이 추가되었습니다.
-

kvm vcpu 종료 이유의 디코딩이 추가되었습니다. 이를 위해서는 **-e kvm=vcpu** 옵션을 사용합니다.

- 이제 **-k** 옵션을 사용할 때 **elfutils**의 **libdw** 라이브러리가 스택 폴링에 사용됩니다. 또한 기호 분리는 **libiberty** 라이브러리를 사용하여 수행됩니다.
- 이전에는 **-r** 옵션을 사용하면 **strace**가 **-t** 옵션을 무시했습니다. 이 문제가 해결되었으며 두 옵션은 이제 독립적입니다.
- **append** 모드에서 출력 파일을 열기 위해 **-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.
- 이제 **IBM Z** 아키텍처에서 단일 **strace** 톨을 사용하여 64비트 및 32비트 바이너리를 추적할 수 있습니다. 결과적으로 **RHEL 8**에는 별도의 **strace32** 패키지가 더 이상 존재하지 않습니다.

또한 다음 항목에 대한 디코딩이 추가, 개선, 업데이트되었습니다.

- **netlink** 프로토콜, 메시지 및 속성
- **arch_prctl, bpf, getsockopt, io_pgetevent, keyctl, prctl, pkey_alloc, pkey_free, pkey_mprotect, ptrace, rseq, setsockopt, 소켓, statx** 및 기타 시스템 호출
- **ioctl** 시스템 호출을 위한 여러 명령
- 다양한 유형의 상수
- 간접 인수가 있는 **execveat, inotify_add_watch, inotify_init, select, symlink, symlinkat** 시스템 호출 및 **mmap** 시스템 호출 경로 추적

- 신호 코드 목록

(BZ#1641014)

RHEL 8의 컴파일러 툴 세트

RHEL 8.0은 다음과 같은 컴파일러 툴셋을 **Application Streams**로 제공합니다.

- **LLVM** 컴파일러 인프라 프레임워크, **C** 및 **C++** 언어용 **Clang** 컴파일러, **LLDB** 디버거 및 코드 분석을 위한 관련 툴을 제공하는 **Clang** 및 **LLVM Toolset 7.0.1** [Using Clang and LLVM Toolset](#) 문서를 참조하십시오.
- **rust Toolset 1.31**은 **Rust** 프로그래밍 언어 컴파일러 **rustc**, **loading** 빌드 도구 및 종속성 관리자, **stuff -vendor** 플러그인 및 필수 라이브러리를 제공합니다. [Rust Toolset 사용](#) 문서를 참조하십시오.
- **Go Toolset 1.11.5**: **Go** 프로그래밍 언어 도구 및 라이브러리를 제공합니다. **Go**는 **golang**라고도 합니다. [Go Toolset 사용](#) 문서를 참조하십시오.

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

RHEL 8의 Java 구현 및 Java 툴

RHEL 8 AppStream 리포지토리에는 다음이 포함됩니다.

- **OpenJDK 11 Java** 런타임 환경 및 **OpenJDK 11 Java** 소프트웨어 개발 키트를 제공하는 **java-11-openjdk** 패키지.
- **OpenJDK 8 Java** 런타임 환경 및 **OpenJDK 8 Java** 소프트웨어 개발 키트를 제공하는 **java-1.8.0-openjdk** 패키지.
- **Java Web Start**를 구현하는 **Iced tea-web** 패키지입니다.
-

Java 애플리케이션을 컴파일, 조립, 테스트 및 실행하는 **Java** 라이브러리 및 명령줄 툴을 제공하는 **ant** 모듈입니다. **ant**가 버전 **1.10**으로 업데이트되었습니다.

- **maven** 모듈은 소프트웨어 프로젝트 관리 및 이해 툴을 제공하는 **maven** 모듈입니다. **Maven**은 이전에는 소프트웨어 컬렉션 또는 지원되지 않는 선택적 채널에서만 사용할 수 있었습니다.
- **scala** 모듈은 **Java** 플랫폼용 일반 용도의 프로그래밍 언어를 제공합니다. **Scala**는 이전에는 소프트웨어 컬렉션으로만 사용할 수 있었습니다.

또한 **java-1.8.0-ibm** 패키지는 추가 리포지토리를 통해 배포됩니다. 이 리포지토리의 패키지는 **Red Hat**에서 지원되지 않습니다.

(BZ#1699535)

C ++ ABI가 **mac ::string** 및 **-4.8 ::list**에서 변경

libstdc++ 라이브러리의 **ABI(Application Binary Interface)** 및 **cryptsetup ::list** 클래스가 **RHEL 7 (GCC 4.8)**과 **C++11** 표준을 준수하도록 **RHEL 8(GCC 8)** 간에 변경되었습니다. **libstdc++** 라이브러리는 이전 **ABI**와 새로운 **ABI**를 모두 지원하지만 다른 **C++** 시스템 라이브러리는 지원하지 않습니다. 따라서 이러한 라이브러리에 대해 동적으로 연결되는 애플리케이션을 다시 빌드해야 합니다. 이는 **C++98**을 포함한 모든 **C++** 표준 모드에 영향을 미칩니다. **RHEL 7용 Red Hat Developer Toolset** 컴파일러로 빌드된 애플리케이션에도 영향을 미치므로 시스템 라이브러리와 호환성을 유지하기 위해 이전 **ABI**를 유지했습니다.

(BZ#1704867)

5.1.12. 파일 시스템 및 스토리지

DIF/DIX(데이터 무결성 필드/데이터 무결성 확장) 지원

DIF/DIX는 하드웨어 벤더가 자격을 갖춘 구성에서 지원되며, **RHEL**에서 특정 **HBA(호스트 버스 어댑터)** 및 스토리지 어레이 구성에 대한 완전한 지원을 제공합니다.

DIF/DIX는 다음 구성에서 지원되지 않습니다.

- 부팅 장치에서 사용하는 것은 지원되지 않습니다.

- 가상화된 게스트에서는 지원되지 않습니다.
- **DIF/DIX**가 활성화된 경우 **Red Hat**은 자동 스토리지 관리 라이브러리(**ASMLib**) 사용을 지원하지 않습니다.

DIF/DIX는 스토리지 장치에서 활성화 또는 비활성화되어 있으며 애플리케이션을 포함(및 포함)할 수 있습니다. 스토리지 장치의 **DIF**를 활성화하는 방법은 장치에 따라 다릅니다.

DIF/DIX 기능에 대한 자세한 내용은 [DIF/DIX](#)의 내용을 참조하십시오.

(BZ#1649493)

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 **XFS**는 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/O**를 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 **Extent**를 사용할 수 있습니다.

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

이 기능은 보다 더욱 효율적인 운영을 위해 **Overlayfs** 및 **NFS**와 같은 커널 하위 시스템에서도 사용됩니다.

공유 **COW(Copy-On-Write)** 데이터 **Extent**는 **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 **Extent**가 있는 **XFS**를 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 **Extent**로 **XFS** 파일 시스템을 마운트할 수 있습니다.

(BZ#1494028)

XFS 파일 시스템 최대 크기는 **1024TiB**

XFS 파일 시스템의 최대 지원 크기가 **500TiB**에서 **1024TiB**로 증가했습니다.

500TiB 이상의 파일 시스템에는 다음이 필요합니다.

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

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)

ext4 파일 시스템에서 메타데이터 체크섬 지원

이번 업데이트를 통해 **ext4** 메타데이터는 체크섬 으로 보호됩니다. 이를 통해 파일 시스템이 손상된 메타데이터를 인식하여 손상을 방지하고 파일 시스템의 탄력성을 높일 수 있습니다.

(BZ#1695584)

모든 아키텍처를 지원하는 **VDO**

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

지원되는 아키텍처 목록은 [???](#) 에서 참조하십시오.

(BZ#1534087)

BOOM 부팅 관리자가 부팅 항목을 생성하는 과정을 단순화

BOOM은 부팅 항목 설정을 위해 **BootLoader Specification**을 지원하는 부팅 로더를 사용하는 **Linux** 시스템용 부팅 관리자입니다. 유연한 부팅 설정이 가능하며 **LVM**을 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 **API**가 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 내부 유연한 레이아웃으로 인해 향후 기능의 활성화이기도 합니다. **libcryptsetup**에 구축된 일반 커널 키링 토큰을 통해 자동 잠금을 지원하므로 사용자가 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 및 Marvell Qlogic Fibre Channel 어댑터에서 NVMe/FC 완전 지원

Broadcom Emulex 및 Marvell Qlogic Fibre Channel 32Gbit 어댑터와 함께 사용할 때 **Initiator** 모드에서 **NVMe over Fibre Channel (NVMe/FC)** 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.

NVMe/FC 활성화:

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

```
lpfc_enable_fc4_type=3
```

- **qla2xxx** 드라이버에서 **NVMe/FC**를 활성화하려면 **/etc/modprobe.d/qla2xxx.conf** 파일을 편집하고 다음 옵션을 추가합니다.

```
qla2xxx.ql2xnvmeenable=1
```

추가 제한 사항:

- **NVMe/FC**에서 멀티패스가 지원되지 않습니다.
- **NVMe/FC**에서 **NVMe** 클러스터링이 지원되지 않습니다.
- **NVMe/FC**에서 **kdump**가 지원되지 않습니다.
- **SAN(Storage Area Network)**의 **NVMe/FC**에서 시스템 부팅이 지원되지 않습니다.

(BZ#1649497)

새로운 **scan_lvs** 구성 설정

기본적으로 새로운 **lvm.conf** 구성 파일 설정인 **scan_lvs**가 추가되어 **0**으로 설정됩니다. 새 기본 동작은 **LVM**에서 **LV** 상단에 있을 수 있는 **PV**를 검색하지 않도록 중지합니다. 즉, 추가 **PV**에 대해 활성 **LV**를 검사하지 않습니다. 기본 설정은 **LVM**이 **LV**에서 **PV**를 생성하지 못하도록 합니다.

LV 상단에 있는 **PV**를 계층화하면 **LV**에 배치된 **VM** 이미지가 발생할 수 있습니다. 이 경우 호스트가 **PV**에 액세스하는 것이 안전하지 않습니다. 이 안전하지 않은 액세스를 방지하는 것이 새로운 기본 동작의 주요 이유입니다. 또한 활성 **LV**가 많은 환경에서 **LVM**에서 수행하는 장치 스캔의 양을 크게 줄일 수 있습니다.

이 설정을 **1**로 변경하여 이전 동작을 복원할 수 있습니다.

(BZ#1676598)

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 **GUI**를 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/O**를 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 Multipath**는 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)

블록 장치의 다중 대기열 스케줄링

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.

RHEL 7 및 이전 버전에서 사용 가능한 기존 스케줄러가 제거되었습니다. RHEL 8에서는 다중 대기열 스케줄러만 지원합니다.

(BZ#1647612)

5.1.13. 고가용성 및 클러스터

새로운 pcs 명령으로 사용 가능한 위치독 장치 나열 및 테스트

Pacemaker로 SBD를 구성하려면 위치독 장치가 작동해야 합니다. 이 릴리스에서는 pcs stonith sbd watchdog list 명령을 지원하여 로컬 노드에서 사용 가능한 위치독 장치를 나열하고 pcs stonith sbd watchdog test 명령을 지원하여 위치독 장치를 테스트합니다. sbd 명령행 툴에 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. pcs resource failcount show 명령을 사용하면 리소스, 노드, 작업, 간격별로 오류를 필터링할 수 있습니다. 리소스 및 노드별로 집계된 오류를 표시하거나 리소스, 노드, 작업 및 간격별로 세부 정보를 표시하는 옵션을 제공합니다. 또한 pcs resource cleanup 명령을 사용하면 리소스, 노드, 작업 및 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 pcs는 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)

새로운 pcs 명령

Red Hat Enterprise Linux 8에는 다음과 같은 새로운 명령이 도입되었습니다.

- RHEL 8에서는 새로운 명령 **pcs cluster node add-guest | remove-guest** 를 도입했습니다. 이 명령은 RHEL 7에서 **pcs cluster remote-node add | remove** 명령을 대체합니다.
- RHEL 8에는 RHEL 7에서 **pcs cluster** 쿼럼 차단 해제 명령을 대체하는 새로운 명령 **pcs quorum** 차단 해제 명령이 도입되었습니다.
- **pcs resource failcount reset** 명령이 **pcs resource cleanup** 명령의 기능을 복제하므로 제거되었습니다.
- RHEL 8에서는 RHEL 7에서 **pcs resource [show]** 명령을 바꾸는 새로운 명령이 도입되었습니다.
 - RHEL 8의 **pcs resource [status]** 명령은 RHEL 7의 **pcs resource [show]** 명령을 대체합니다.
 - RHEL 8의 **pcs resource config** 명령은 RHEL 7에서 **pcs resource [show] --full** 명령을 대체합니다.
 - RHEL 8의 **pcs resource config resource id** 명령은 RHEL 7의 **pcs resource show**

resource id 명령을 대체합니다.

- RHEL 8에서는 RHEL 7에서 **pcs stonith [show]** 명령을 교체하는 새로운 명령이 도입되었습니다.
 - RHEL 8의 **pcs stonith [status]** 명령은 RHEL 7의 **pcs stonith [show]** 명령을 대체합니다.
 - RHEL 8의 **pcs stonith config** 명령은 RHEL 7의 **pcs stonith [show] --full** 명령을 대체합니다.
 - RHEL 8의 **pcs stonith config resource id** 명령은 RHEL 7의 **pcs stonith show resource id** 명령을 대체합니다.

(BZ#1654280)

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 기본 값으로 설정해야 합니다. 더 이상 사용하지 않는 구문이 있는 기존 구성(새롭게 생성된 구문이 아님)은 지원되는 구문을 사용하도록 자동 업데이트됩니다.
- 자세한 변경 사항 목록은 [Red Hat Enterprise Linux 8의 Pacemaker 2.0 업그레이드를 참조](#) 하십시오.

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. 그런 다음 **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)

5.1.14. 네트워킹

nftables가 기본 네트워크 패킷 필터링 프레임워크로 **iptables** 대체

nftables 프레임워크는 패킷 분류 기능을 제공하며 **iptables**, **ip6tables**, **arptables**, **ebtables** 툴을 대

체합니다. 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

iptables과 마찬가지로 **nftables**는 체인을 저장하기 위해 테이블을 사용합니다. 체인에는 작업을 수행하기 위한 개별 규칙이 포함되어 있습니다. **nft** 툴은 이전 패킷 필터링 프레임워크의 모든 툴을 대체합니다. **libnftables** 라이브러리는 **libmnl** 라이브러리를 통해 **nftables** Netlink API와 낮은 수준의 상호 작용에 사용할 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 8**에서는 **nftables** 기반의 **iptables** 툴로 다음 버전 문자열이 출력됩니다.

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

기존 **iptables** 틀이 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 더 높은 성능, 향상된 확장성 및 안정성을 제공하는 **TCP** 네트워킹 스택 버전 **4.18**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/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.

firewalld가 설정된 연결을 일반적으로 수락하기 전에 직접 규칙 실행이 이루어짐

(BZ#1509026)

RHEL 8 wpa_supplicant에서 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-IOV**와 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 **4.18**로 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 예를 들어 **pfifo_fast**는 이제 잠금이 없습니다.
- 가비지 컬렉션 커널 스레드를 제거하여 **IP** 리셀러 장치의 확장성 개선과 **ip fragments**는 시간 초과 시에만 만료됩니다. 그 결과 **DoS**의 **CPU** 사용량이 훨씬 낮고 지속 가능한 최대 조각율은 **IP** 리싱 단위에 대해 구성된 메모리 양으로 제한됩니다.

(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 누락되

어 있는 경우도 있습니다. 이러한 해당 기능이 누락된 확장 기능이 있는 경우 둘은 # 기호가 앞에 있는 미 변환 규칙을 출력합니다. 예를 들어 다음과 같습니다.

```
| % 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. 예를 들어 다음과 같습니다.

```
| % 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-DATA**가 **SCTP**에 추가

이번 업데이트에는 **SCTP(Stream Control Transmission Protocol)**에 새로운 데이터 체크 유형, **I-**

DATA 및 스트림 스케줄러를 추가되었습니다. 이전에는 **SCTP**가 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** 청크를 사용하면 **TSN(Transmission Sequence Number)** 필드가 오버로드되지 않으므로 **SCTP**는 스트림을 다른 방식으로 스케줄링할 수 있고, **I-DATA**는 사용자의 메시지 인터리브를 허용합니다(**RFC 8260**). 결과적으로 **SCTP**는 이제 다양한 방식으로 스트림을 예약할 수 있으며 **I-DATA**는 사용자 메시지를 인터리빙(**RFC 8260**)할 수 있습니다. 두 피어 모두 **I-DATA** 청크 유형을 지원해야 합니다.

(BZ#1273139)

NetworkManager 가 ethtool 오프로드 기능 구성을 지원

이 향상된 기능을 통해 **NetworkManager**는 **ethtool** 오프로드 기능 구성을 지원하며 사용자는 더 이상 **init** 스크립트 또는 **NetworkManager** 디스패처 스크립트를 사용할 필요가 없습니다. 결과적으로 사용자는 다음 방법 중 하나를 사용하여 오프로드 기능을 연결 프로필의 일부로 구성할 수 있습니다.

- **nmcli** 유틸리티 사용
- **/etc/NetworkManager/system-connections/** 디렉터리에서 키 파일을 편집
- **/etc/sysconfig/network-scripts/ifcfg-*** 파일을 편집하여

이 기능은 현재 그래픽 인터페이스 및 **nmtui** 유틸리티에서 지원되지 않습니다.

(BZ#1335409)

RHEL 8의 TCP BBR 지원

새로운 **TCP** 정체 제어 알고리즘, **Bottleneck Bandwidth** 및 **BBR(BBR)**이 이제 **Red Hat Enterprise Linux (RHEL) 8**에서 지원됩니다. **BBR**은 성능 장애 링크의 대역폭과 **round-trip** 시간 (**RTT**)을 확인하려고 시도합니다. 대부분의 혼잡 알고리즘은 패킷 손실(기본 **Linux TCP** 혼잡 제어 알고리즘 포함)을 기반으로 하는데, 이는 높은 처리량 링크에 문제가 있습니다. **BBR**은 손실 이벤트에 직접 반응하지 않으며, 가능한 대역폭과 일치하도록 **TCP pacing** 속도를 조정합니다. **TCP BBR** 사용자는 모든 관련 인터페이스에서 **fq** 큐링 설정으로 전환해야 합니다.

사용자는 **fq_codel** 대신 **fq_codel** 을 명시적으로 사용해야 합니다.

자세한 내용은 **tc-fq** 매뉴얼 페이지를 참조하십시오.

(BZ#1515987)

RHEL 8의 lksctp-tools, 버전 1.0.18

lksctp-tools 패키지인 3.28 버전은 RHEL(Red Hat Enterprise Linux) 8에서 사용할 수 있습니다. 주요 개선 사항 및 버그 수정 사항은 다음과 같습니다.

- Travis CI 및 Coverity Scan과의 통합
- sctp_peeloff_flags 함수에 대한 지원
- 사용 가능한 커널 기능 표시
- Coverity Scan 문제 해결

(BZ#1568622)

RHEL 8에서 기본적으로 SCTP 모듈 블랙리스트 지정

보안을 강화하기 위해 커널 모듈 세트가 **kernel-modules-extra** 패키지로 변경되었습니다. 이는 기본적으로 설치되지 않습니다. 따라서 루트가 아닌 사용자는 기본적으로 블랙리스트에 올 때 이러한 구성 요소를 로드할 수 없습니다. 이러한 커널 모듈 중 하나를 사용하려면 시스템 관리자가 **kernel-modules-extra**를 설치하고 모듈 블랙리스트를 명시적으로 제거해야 합니다. 따라서 루트가 아닌 사용자는 소프트웨어 구성 요소를 자동으로 로드할 수 있습니다.

(BZ#1642795)

driverctl 0.101에서 주요 변경 사항

Red Hat Enterprise Linux 8.0은 **driverctl 0.101**과 함께 배포됩니다. 이 버전에는 다음과 같은 버그 수정이 포함되어 있습니다.

- **shellcheck** 경고가 수정되었습니다.
- **bash-completion**은 **driverctl - bash-completion.sh** 대신 **driverctl** 으로 설치됩니다.
- **PCI** 버스에 대한 **load_override** 기능이 수정되었습니다.
- **driverctl** 서비스는 **basic.target systemd** 대상에 도달하기 전에 모든 재정의의를 로드합니다.

(BZ#1648411)

firewalld에 리치 규칙 우선순위 추가

우선 순위 옵션이 리치 규칙에 추가되었습니다. 이를 통해 사용자는 규칙 실행 중에 바람직한 우선 순위 순서를 정의할 수 있으며 리치 규칙에 대한 고급 제어를 제공합니다.

(BZ#1648497)

RHEL 8에서 RDMA를 통한 NVMe 지원

RHEL(Red Hat Enterprise Linux) 8에서는 RDMA(Remote Direct Memory Access)를 통한 NVMe(Nonvolatile Memory Express)는 Infiniband, RoCEv2 및 iWARP만 이니시에이터 모드에서 지원합니다.

Multipath는 장애 조치 모드에서만 지원됩니다.

추가 제한 사항:

- **NVMe/RDMA**에서는 **kdump**가 지원되지 않습니다.
- **RDMA**를 통해 **NVMe** 장치에서 부팅하는 것은 지원되지 않습니다.

(BZ#1680177)

nf_tables 백엔드는 **dmesg**를 사용한 디버깅을 지원하지 않습니다.

Red Hat Enterprise Linux 8.0은 **dmesg** 유틸리티의 출력을 사용하여 방화벽 디버깅을 지원하지 않는 방화벽에 **nf_tables** 백엔드를 사용합니다. 방화벽 규칙을 디버깅하려면 **xtables-monitor -t** 또는 **nft monitor** 추적 명령을 사용하여 규칙 평가 이벤트를 디코딩합니다.

(BZ#1645744)

Red Hat Enterprise Linux에서 VRF 지원

RHEL 8.0의 커널은 가상 라우팅 및 전달(VRF)을 지원합니다. **ip** 유틸리티를 사용하여 규칙 세트와 결합된 VRF 장치를 사용하면 관리자가 Linux 네트워크 스택에서 VRF 도메인을 생성할 수 있습니다. 이러한 도메인은 트래픽을 계층 3에서 분리하므로 관리자는 다른 라우팅 테이블을 생성하고 한 호스트에서 다른 VRF 도메인 내에서 동일한 IP 주소를 재사용할 수 있습니다.

(BZ#1440031)

iproute, RHEL 8의 4.18 버전

iproute 패키지는 RHEL(Red Hat Enterprise Linux) 8의 버전 4.18과 함께 배포됩니다. 가장 주목할 만한 변경 사항은 **ethX:Y**로 표시된 인터페이스 별칭(예: **eth0:1**)이 더 이상 지원되지 않는다는 것입니다. 이 문제를 해결하려면 **ip link show**를 입력하기 전에 콜론과 다음 번호인 별칭 접미사를 제거해야 합니다.

(BZ#1589317)

5.1.15. 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](#) 페이지에서 가져올 수도 있습니다.

(BZ#1636338)

전체 시스템의 암호화 정책이 기본적으로 적용

암호화 정책은 TLS, IPsec, DNSSEC, Kerberos 및 SSH 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 매개 변수는 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)

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 **2048bit**로 변경됨
- **ExposeAuthInfo** 설정 옵션의 의미가 변경됨
- **UsePrivilegeSeparation=sandbox** 옵션은 필수 사항이며 비활성화할 수 없음
- 허용되는 최소의 **RSA** 키 크기를 **1024bit**로 설정

(BZ#1622511)

자동 **OpenSSH** 서버 키 생성을 **sshd-keygen@.service**에서 처리

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

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

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

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

(BZ#1228088)

SSH 인증에 대해 **ECDSA** 키 지원

이번 **OpenSSH** 제품군에서는 **PKCS #11** 스마트 카드에 저장된 **ECDSA** 키를 지원합니다. 따라서 사용자는 이제 **SSH** 인증에 **RSA** 및 **ECDSA** 키를 모두 사용할 수 있습니다.

(BZ#1645038)

libssh는 코어 암호화 구성 요소로서 **SSH**를 구현

이러한 변경으로 인해 **Red Hat Enterprise Linux 8**의 코어 암호화 구성 요소인 **libssh**가 도입되었습니다. **libssh** 라이브러리는 **SSH(Secure Shell)** 프로토콜을 구현합니다.

libssh 클라이언트 측은 시스템 전체 암호화 정책을 통해 **OpenSSH**에 설정된 구성을 따르지만, 시스템 차원의 암호화 정책을 통해 서버 측은 구성은 변경할 수 없습니다.

(BZ#1485241)

암호화 라이브러리에서 **TLS 1.3** 지원

이번 업데이트에는 모든 주요 백엔드 암호화 라이브러리에서 기본적으로 **TLS(Transport Layer Security) 1.3**을 사용하도록 설정되어 있습니다. 이를 통해 운영 체제 통신 계층 전반에서 낮은 대기 시간을 단축하고 **RSA-PSS** 또는 **X25519**와 같은 새로운 알고리즘을 사용하여 애플리케이션의 개인정보 보호 및 보안을 강화합니다.

(BZ#1516728)

NSS는 기본적으로 **SQL** 사용

이제 **NSS(Network Security Services)** 라이브러리는 기본적으로 신뢰 데이터베이스에 **SQL** 파일 형식을 사용합니다. 이전 릴리스에서 기본 데이터베이스 형식으로 사용된 **DBM** 파일 형식은 여러 프로세스에서 동일한 데이터베이스에 대한 동시 액세스를 지원하지 않으며 업스트림에서 더 이상 사용되지 않습니다. 결과적으로 **NSS** 신뢰 데이터베이스를 사용하여 키, 인증서 및 취소 정보를 저장하는 애플리케이션에서는 기본적으로 **SQL** 형식으로 데이터베이스를 생성합니다. 레거시 **DBM** 형식으로 데이터베이스를 생성하려고 하면 실패합니다. 기존 **DBM** 데이터베이스는 읽기 전용 모드로 열리고 자동으로 **SQL** 형식으로 변환됩니다. **NSS**는 **Red Hat Enterprise Linux 6** 이후의 **SQL** 파일 형식을 지원합니다.

(BZ#1489094)

스마트 카드 및 **HSM**에 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 **HSM**을 완벽하게 지원

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

(BZ#1516741)

Firefox는 이제 시스템 전체에서 등록된 **PKCS #11** 드라이버와 함께 작동합니다.

Firefox 웹 브라우저는 **p11-kit-proxy** 모듈을 자동으로 로드하고 **p11-kit**에 시스템 전체에서 등록된 모든 스마트 카드는 **pkcs11.conf** 파일을 통해 자동으로 감지됩니다. **TLS** 클라이언트 인증을 사용하려면 추가 설정이 필요하지 않으며 서버가 요청할 때 스마트 카드의 키가 자동으로 사용됩니다.

(BZ#1595638)

RSA-PSS가 **OpenSC**에서 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 Libreswan에서 주요 변경 사항

libreswan 패키지가 업스트림 버전 **3.27**로 업그레이드되어 이전 버전에 비해 많은 버그 수정 및 개선 사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- **authby=rsa-sha2, ECDSA (RFC 7427)**를 통한 **RSA-PSS(RFC 7427)** 지원, **authby=ecdsa-**

sha2 를 통해 RFC 7427(RFC 7427)에 대한 지원은 dh31 키워드를 사용하는 CURVE25519 및 IKE20-POLY 1305 를 통해 IKE20-POLY1305를 통해 암호화를 추가했습니다.

- 업스트림에서 KLIPS를 완전히 사용하지 않으므로 KLIPS 대체 커널 모듈에 대한 지원이 Libreswan 에서 제거되었습니다.
- Diffie-Hellman 그룹은 DH22, DH23 및 DH24는 더 이상 지원되지 않습니다(RFC 8247 참조).

authby=rsasig 는 항상 RSA v1.5 방법을 사용하도록 변경되었으며 authby=rsa-sha2 옵션은 RSASSA-PSS 방법을 사용합니다. authby=rsa-sha1 옵션은 RFC 8247에 따라 유효하지 않습니다. 이는 Libreswan 이 더 이상 디지털 서명으로 SHA-1을 지원하지 않는 이유입니다.

(BZ#1566574)

시스템 전체 암호화 정책은 Libreswan의 기본 IKE 버전을 IKEv2로 변경합니다.

Libreswan IPsec 구현의 기본 IKE 버전이 IKEv1(RFC tunnel9)에서 IKEv2(RFC 7296)로 변경되었습니다. IPsec과 함께 사용할 기본 IKE 및 ESP/AH 알고리즘이 시스템 전체 암호화 정책, RFC 8221 및 RFC 8247을 준수하도록 업데이트되었습니다. 이제 256비트의 암호화 키 크기가 128비트의 키 크기보다 우선합니다.

기본 IKE 및 ESP/AH 암호에는 이제 AES-GCM, CHACHA20POLY1305 및 AES-CBC가 포함되어 있습니다. 무결성 검사를 위해 AEAD 및 SHA-2를 제공합니다. 이제 Diffie-Hellman 그룹에 DH19, DH20, DH21, DH14, DH15, DH16, DH18이 포함됩니다.

다음 알고리즘은 기본 IKE 및 ESP/AH 정책(AES_CTR, 3DES, SHA1, DH5, DH2, DH23 및 DH24)에서 제거되었습니다. DH22, DH23 및 DH24의 예외로, IPsec 구성 파일의 ike= 또는 phase2alg=/esp=/ah= 옵션으로 이러한 알고리즘을 활성화할 수 있습니다.

IKEv1 프로토콜이 필요한 IPsec VPN 연결을 구성하려면 연결 구성 파일에 ikev2=no 옵션을 추가합니다. 자세한 내용은 ipsec.conf(5) 도움말 페이지를 참조하십시오.

(BZ#1645606)

Libreswan의 버전 관련 변경 사항

이번 개선된 기능을 통해 **Libreswan**은 인터넷 키 교환(IKE) 설정을 다르게 처리합니다.

- 기본 인터넷 키 교환(IKE) 버전이 1에서 2로 변경되었습니다.
- 이제 연결은 **IKEv1** 또는 **IKEv2** 프로토콜을 사용할 수 있지만 둘 다 사용할 수는 없습니다.
- **ikev2** 옵션의 해석이 변경되었습니다.
 - **sist**의 값은 **IKEv2-only**로 해석됩니다.
 - **no** 값은 **IKEv1-only**로 해석되지 않습니다.
 - 값이 **yes**, **allow**는 더 이상 유효하지 않으며 오류가 발생합니다. **IKE** 버전이 이러한 값으로 인해 발생한 버전이 명확하지 않기 때문에 오류가 발생합니다.

(BZ#1648776)

RHEL 8 OpenSCAP의 새로운 기능

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

- **API** 및 **ABI**가 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)

SCAP Security Guide 가 시스템 전체의 암호화 정책 지원

scap-security-guide 패키지는 코어 암호화 하위 시스템을 구성하기 위해 사전 정의된 시스템 전체 암호화 정책을 사용하도록 업데이트되었습니다. 시스템 전체 암호화 정책과 충돌하거나 과도하게 충돌하는 보안 콘텐츠가 제거되었습니다.

이 변경 사항은 **scap-security-guide** 의 보안 콘텐츠에만 적용되며 **OpenSCAP** 스캐너 또는 기타 **SCAP** 구성 요소를 업데이트할 필요가 없습니다.

(BZ#1618505)

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 schematron**에 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)

SCAP Security Guide PCI-DSS 프로파일이 버전 3.2.1과 일치

scap-security-guide 패키지는 Red Hat Enterprise Linux 8의 **PCI-DSS(Payment Card Industry Data Security Standard)** 프로필을 제공하며 최신 **PCI-DSS** 버전 - **3.2.1**과 일치하도록 이 프로필이 업데이트되었습니다.

(BZ#1618528)

SCAP Security Guide가 OSPP 4.2 지원

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

(BZ#1618518)

RHEL 8 rsyslog에서 주요 변경 사항

rsyslog 패키지가 업스트림 버전 **8.37.0**으로 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 **스팸 출처**가 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

- 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 디버깅하는 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
- **journald**가 파일을 순환할 때 나타날 수 있는 중복 레코드를 피하기 위해 **imjournal** 옵션이 추가되었습니다. 이 옵션을 사용하면 성능에 영향을 줄 수 있습니다.

rsyslog를 사용한 시스템은 **journald** 없이 시스템 로깅 구성 또는 최소 **journald** 사용 지식 베이스 문서에 설명된 대로 더 나은 성능을 제공하도록 구성할 수 있습니다.

(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)

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 순환 시킬 수 있습니다.

(BZ#1614179)

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)

Audit 3.0에서 **audispd**를 **auditd**로 교체

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

(BZ#1616428)

tangd_port_t를 사용하면 **Tang**의 기본 포트를 변경할 수 있습니다.

이번 업데이트에서는 **tangd_port_t** SELinux 유형이 있어 **tangd** 서비스가 SELinux 강제 모드로 제한됨으로 실행될 수 있습니다. 이러한 변경으로 인해 사용자 정의 포트에서 수신 대기하도록 **Tang** 서버 구성이 간소화되고 SELinux가 강제 모드에서 제공하는 보안 수준도 유지하는 데 도움이 됩니다.

자세한 내용은 [정책 기반 암호 해독](#) 섹션을 사용하여 암호화된 볼륨의 자동 잠금 해제 구성 섹션을 참조하십시오.

(BZ#1664345)

새로운 SELinux 부울값

이번 SELinux 시스템 정책 업데이트로 다음과 같은 부울값이 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_pcscd`

의미를 포함한 부울 목록을 가져오고 활성화되어 있는지 확인하려면 `selinux-policy-devel` 패키지를 설치하고 다음을 사용합니다.

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

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 };
```

예를 들어 다음과 같습니다.

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

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

(BZ#1594111)

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 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. **SELinux**는 프로세스에서 **prlimit()**를 통해 자체 리소스 제한을 처리하는 작업을 제한하지 않습니다. 자세한 내용은 **prlimit(2)** 및 **getrlimit(2)** 매뉴얼 페이지를 참조하십시오.

(BZ#1549772)

SELinux-policy 에서 VxFS 라벨 지원

이번 업데이트에서는 **Veritas File System (VxFS)** 보안 확장 속성(**xattrs**)을 지원합니다. 이를 통해 일반 **vxfs_t** 유형 대신 파일 시스템에 오브젝트를 사용하여 적절한 **SELinux** 레이블을 저장할 수 있습니다. 따라서 **SELinux**를 완전히 지원하는 **VxFS** 시스템을 더 안전하게 사용할 수 있습니다.

(BZ#1483904)

컴파일 타임 보안 강화 플래그가 더 일관되게 적용됨

컴파일 타임 보안 강화 플래그는 **RHEL 8** 배포판의 **RPM** 패키지에 더 일관되게 적용되며 **redhat-rpm-config** 패키지에서 보안 강화 플래그를 자동으로 제공합니다. 적용된 컴파일 타임 플래그도 **CCC(Common Criteria)** 요구 사항을 충족하는 데 도움이 됩니다. 다음과 같은 보안 강화 플래그가 적용됩니다.

- 버퍼-overflow 오류 탐지: **D_FORTIFY_SOURCE=2**
- C++ 배열, 벡터 및 문자열을 확인하는 표준 라이브러리 강화: **D_GLIBCXX_ASSERTIONS**
- Stack Smashing Protector (SSP): **fstack-protector-strong**
- 예외 강화의 경우: **fexceptions**
- CFI(Control-Flow Integrity): **fcf-protection=full** (AMD 및 Intel 64비트 아키텍처에서만)
- 주소 공간 레이아웃 복원(ASLR)의 경우: **fPIE** (플레이북용) 또는 **fPIC** (라이브러리용)
- Stack Clash 취약점을 보호하기 위해: **fstack-clash-protection** (ARM 제외)
- 시작 시 모든 기호를 해결하기 위한 링크 플래그: **-Wl,-z,now**

자세한 내용은 **gcc(1)** 매뉴얼 페이지를 참조하십시오.

(JIRA:RHELPLAN-2306)

5.1.16. 가상화

RHEL 8의 qemu-kvm 2.12

Red Hat Enterprise Linux 8은 **qemu-kvm 2.12**와 함께 배포됩니다. 이 버전은 Red Hat Enterprise Linux 7에서 사용할 수 있는 버전 **1.5.3**에 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 스텔딩**

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

(BZ#1559240)

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 8**에서 생성된 가상 머신은 기본적으로 **Q35**를 사용하도록 설정되어 있습니다.

이전의 기본 **PC** 시스템 유형은 더 이상 사용되지 않으며 **Q35**를 지원하지 않는 이전 운영 체제를 가상화할 때만 사용해야 합니다.

(BZ#1599777)

Post-copy 가상 머신 마이그레이션

RHEL 8에서는 **KVM** 가상 머신(**VM**)의 사후 복사 마이그레이션을 수행할 수 있습니다. 사용하는 경우 **post-copy** 마이그레이션은 소스 호스트에서 **VM**의 **vCPU**를 일시 중지하고, 최소 메모리 페이지만 전송하

고, 대상 호스트에서 VM의 vCPU를 활성화한 다음, VM이 대상에서 실행되는 동안 나머지 메모리 페이지를 전송합니다.

이렇게 하면 마이그레이션된 VM의 다운타임이 크게 감소하고 소스 VM의 메모리 페이지가 얼마나 빠르게 변경되는지에 관계없이 마이그레이션이 완료됩니다. 따라서 표준 사전 복사 마이그레이션으로 마이그레이션할 수 없는 지속적인 사용으로 VM을 마이그레이션하는 것이 좋습니다.

(JIRA:RHELPLAN-14323)

KVM 가상화에서 virtio-gpu 지원

KVM(VM)에 virtio-gpu 디스플레이 장치가 도입되었습니다. virtio-gpu 는 VM 그래픽 성능을 개선하고 가상 GPU 장치가 향후 구현될 수 있도록 다양한 개선사항을 지원합니다.

(JIRA:RHELPLAN-14329)

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)

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 Hat**에서 지원하는 모든 **CPU** 아키텍처에서 **Ceph** 스토리지를 지원합니다.

(BZ#1578855)

IBM Z의 **KVM** 가상 머신용 대화형 부트 로더

IBM Z 호스트에서 **KVM** 가상 머신을 부팅할 때 **QEMU** 부팅 로더 펌웨어가 게스트 **OS**의 대화형 콘솔 인터페이스를 제공할 수 있습니다. 이렇게 하면 호스트 환경에 액세스하지 않고도 게스트 **OS** 부팅 문제를 해결할 수 있습니다.

(BZ#1508137)

가상 머신에서 **IBM z14 ZR1** 지원

이제 **KVM** 하이퍼바이저가 **IBM z14 ZR1** 서버의 **CPU** 모델을 지원합니다. 이를 통해 **IBM Z** 시스템에서 실행되는 **KVM** 가상 머신에서 이 **CPU**의 기능을 사용할 수 있습니다.

(BZ#1592337)

KVM이 **IBM Z**에서 **Telnet 3270** 지원

RHEL 8을 **IBM Z** 시스템에서 호스트로 사용하는 경우 **Telnet 3270** 클라이언트를 사용하여 호스트의 가상 머신에 연결할 수 있습니다.

(BZ#1570029)

QEMU 샌드박스 추가

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

(JIRA:RHELPLAN-10628)

PV TLB 플러시 **Hyper-V** 시행

RHEL 8에서는 **PV TLB flush Hyper-V Enlightenment** 기능이 추가되었습니다. 이를 통해 **KVM** 하이퍼바이저에서 오버 커밋된 환경에서 실행되는 **Windows VM**(가상 머신)의 성능이 향상됩니다.

(JIRA:RHELPLAN-14330)

IBM POWER의 KVM 가상 머신 새 머신 유형

IBM POWER 8 및 **IBM POWER 9** 시스템에서 실행되는 **KVM** 하이퍼바이저에 대해 새로운 **rhel-p** 시리즈 머신 유형이 여러 개 활성화되어 있습니다. 이를 통해 **IBM POWER** 시스템의 **RHEL 8**에서 호스팅되는 가상 머신(VM)이 이러한 머신 유형의 **CPU** 기능을 올바르게 사용할 수 있습니다. 또한 **IBM POWER**의 VM을 **KVM** 하이퍼바이저의 최신 버전으로 마이그레이션할 수 있습니다.

(BZ#1585651, BZ#1595501)

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)

OVMF에 대해 IPv6 활성화

이제 **IPv6** 프로토콜이 **OVMF**(Open Virtual Machine Firmware)에서 활성화됩니다. 이를 통해 **OVMF**를 사용하는 가상 머신은 **IPv6**에서 제공하는 다양한 네트워크 부팅 개선 사항을 활용할 수 있습니다.

(BZ#1536627)

NVMe 장치용 VFIO 기반 블록 드라이버 추가

QEMU 에뮬레이터에서는 **NVMe**(Non-volatile Memory Express) 장치의 **VFIO**(가상 기능 I/O) 기반의 드라이버를 도입합니다. 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)

Hyper-V 일반 UIO 드라이버에 대한 다중 채널 지원

RHEL 8에서는 이제 **Hyper-V** 일반 사용자 공간 I/O(UIO) 드라이버의 다중 채널 기능을 지원합니다. 따라서 **Hyper-V** 하이퍼바이저에서 실행되는 **RHEL 8 VM**이 **DPDK(Data Plane Development Kit)** **Netvsc Poll Mode** 드라이버(PMD)를 사용하여 이러한 VM의 네트워킹 기능을 개선할 수 있습니다.

그러나 **Netvsc** 인터페이스 상태는 실행 중이고 사용 가능한 경우에도 현재 **Down**으로 표시됩니다.

(BZ#1650149)

대규모 페이지 지원 개선

RHEL 8을 가상화 호스트로 사용하는 경우 사용자는 가상 머신(VM) 메모리를 지원하는 페이지 크기를 **CPU**에서 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)

POWER 9 호스트의 VM은 **THP**를 사용할 수 있습니다.

IBM POWER 9 아키텍처에서 실행되는 **RHEL 8** 호스트에서 VM(가상 머신)은 **THP**(투명한 대규모 페이지) 기능을 활용할 수 있습니다. **THP**를 사용하면 호스트 커널이 프로세스에 대규모 메모리 페이지를 동적으로 할당할 수 있으므로 메모리 양이 많은 VM의 성능이 향상됩니다.

(JIRA:RHELPLAN-13440)

5.1.17. 지원 관련 기능

sosreport가 **eBPF** 기반 프로그램 및 맵 보고

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

(BZ#1559836)

5.2. 버그 수정

이 부분에서는 사용자에게 큰 영향을 미치는 Red Hat Enterprise Linux 8.0에서 수정된 버그에 대해 설명합니다.

5.2.1. 데스크탑

PackageKit에서 **rpm** 패키지에서 작동할 수 있음

이번 업데이트를 통해 **rpm** 패키지에서 작동하는 지원이 **PackageKit**에 추가되었습니다.

(BZ#1559414)

5.2.2. 그래픽 인프라

QEMU에서 8바이트 **ggtt** 항목을 올바르게 처리하지 않음

QEMU는 때때로 8바이트 **ggtt** 항목을 연속된 4바이트 쓰기에 씁니다. 이러한 각 부분 쓰기는 별도의 호스트 **ggtt** 쓰기를 트리거할 수 있습니다. 두 개의 **ggtt** 쓰기가 잘못 결합되는 경우가 있습니다. 결과적으로 머신 주소로 변환이 실패하고 오류 로그가 발생했습니다.

(BZ#1598776)

5.2.3. IdM (Identity Management)

Enterprise Security Client는 토큰 탐지를 위해 **opensc** 라이브러리를 사용합니다.

Red Hat Enterprise Linux 8.0은 스마트 카드용 **opensc** 라이브러리만 지원합니다. 이번 업데이트를 통해 **ESC(Enterprise Security Client)**는 제거된 **cold key** 라이브러리 대신 **opensc**를 토큰 탐지에 사용합니다. 결과적으로 애플리케이션에서 지원되는 토큰을 올바르게 탐지합니다.

(BZ#1538645)

인증서 시스템에서 순환 디버그 로그 지원

이전에는 인증서 시스템이 로그 회전을 지원하지 않는 사용자 지정 로깅 프레임워크를 사용했습니다. 그 결과 **/var/log/pki/instance_name/ca/debug grew**와 같은 디버그 로그가 무기한으로 제한됩니다. 이번 업데이트를 통해 인증서 시스템은 로그 회전을 지원하는 **java.logging.util** 프레임워크를 사용합니다.

결과적으로 `/var/lib/pki/instance_name/conf/logging.properties` 파일에서 로그 회전을 구성할 수 있습니다.

로그 순환에 대한 자세한 내용은 `java.util.logging` 패키지에 대한 설명서를 참조하십시오.

(BZ#1565073)

인증서 시스템은 서비스가 시작될 때 `SetAllPropertiesRule` 작업 경고를 더 이상 기록하지 않습니다.

이전에는 서비스가 시작될 때 `/var/log/ipv` 로그 파일의 `SetAllPropertiesRule` 작업에 대한 인증서 시스템이 기록되어 있었습니다. 문제가 해결되었으며 언급된 경고는 더 이상 기록되지 않습니다.

(BZ#1424966)

Certificate System KRA 클라이언트는 키 요청 응답을 올바르게 구문 분석합니다.

이전에는 인증서 시스템이 새 **JSON** 라이브러리로 전환했습니다. 그 결과 특정 개체의 직렬화가 다르게 **Python** 키 복구 기관(**KRA**) 클라이언트가 키 요청 응답을 구문 분석하지 못했습니다. 클라이언트는 이전 **JSON** 라이브러리와 새 **JSON** 라이브러리를 모두 사용하여 응답을 지원하도록 수정되었습니다. 결과적으로 **Python KRA** 클라이언트는 키 요청 응답을 올바르게 구문 분석합니다.

(BZ#1623444)

5.2.4. 컴파일러 및 개발 도구

GCC는 아웃 바운드 액세스에 대해 더 이상 **false positive** 경고를 생성하지 않습니다.

이전 버전에서는 **-O3** 최적화 수준 옵션으로 컴파일할 때 **GNU 컴파일러 컬렉션 (GCC)**은 컴파일된 코드에 포함되지 않은 경우에도 아웃 바운드 액세스에 대한 **false positive** 경고를 반환했습니다. 최적화가 수정되어 **GCC**는 더 이상 **false positive** 경고를 표시하지 않습니다.

(BZ#1246444)

ltrace 가 대규모 구조를 올바르게 표시합니다.

이전에는 **ltrace** 툴에서 함수에서 반환된 대규모 구조를 올바르게 출력할 수 없었습니다. **ltrace** 의 대규모 구조 처리가 개선되어 이제 올바르게 출력됩니다.

(BZ#1584322)

GCC 내장 함수 `__builtin_clz` 는 **IBM Z**에서 올바른 값을 반환합니다.

이전에는 **IBM Z** 아키텍처의 **FLOGR** 명령이 **GCC** 컴파일러에 의해 잘못 접어졌습니다. 결과적으로 이 명령을 사용하는 `__builtin_clz` 함수는 **-funroll-loops** **GCC** 옵션으로 코드를 컴파일할 때 잘못된 결과를 반환할 수 있습니다. 이 버그가 수정되어 이제 기능이 올바른 결과를 제공합니다.

(BZ#1652016)

GDB는 배치 모드에서 마지막 명령이 실패하면 종료 상태가 0이 아닌 상태 제공

이전 버전에서는 **GDB**는 명령의 오류에 관계없이 항상 일괄 모드에서 실행할 때 **status 0** 으로 종료됩니다. 그 결과 명령이 성공했는지 여부를 확인할 수 없었습니다. 이 동작이 변경되었으며 **GDB**는 이제 마지막 명령에서 오류가 발생하면 **status 1** 로 종료됩니다. 이렇게 하면 모든 명령이 실행되는 이전 동작과 호환성이 유지됩니다. 그 결과 **GDB** 일괄 처리 모드 실행이 성공했는지 확인할 수 있습니다.

(BZ#1491128)

5.2.5. 파일 시스템 및 스토리지

출력 수준이 높을수록 더 이상 **iscsiadm** 이 예기치 않게 종료되지 않습니다.

이전 버전에서는 사용자가 **--print** 또는 **-P** 옵션으로 0보다 높은 출력 수준을 지정하면 **iscsiadm** 유틸리티가 예기치 않게 종료되었습니다. 이 문제가 해결되어 이제 모든 인쇄 수준이 예상대로 작동합니다.

(BZ#1582099)

경로의 **WWID**를 가져오지 못하면 **multipathd** 가 더 이상 경로를 비활성화하지 않습니다.

이전에는 **multipathd** 서비스에서 경로가 비어 있는 **WWID**를 가져오는 것처럼 실패한 시도를 처리했습니다. **multipathd** 가 경로의 **WWID**를 가져오지 못하는 경우 해당 경로가 비활성화된 경우가 있습니다.

이번 업데이트를 통해 경로가 변경되었는지 확인할 때 **WWID**를 가져오지 못하는 경우 이전 **WWID**를 계속 사용합니다.

결과적으로 **WWID**가 변경되었는지 확인할 때 **multipathd** 가 **WWID**를 가져오지 못할 때 경로를 더 이상 비활성화하지 않습니다.

(BZ#1673167)

5.2.6. 고가용성 및 클러스터

클라이언트 시작 **SSL/TLS** 재협상을 거부하는 새로운 `/etc/sysconfig/pcsd` 옵션

서버에서 **TLS** 재협상이 활성화되면 클라이언트에서 재협상 요청을 보내 새 핸드셰이크를 시작할 수 있습니다. 핸드셰이크의 컴퓨터 요구 사항은 클라이언트보다 서버에서 높습니다. 이렇게 하면 서버가 **DoS** 공격에 취약합니다. 이번 수정으로 `/etc/sysconfig/pcsd` 설정 파일의 **PCSD_SSL_OPTIONS** 설정은 재협상을 거부하도록 **OP_NO_RENEGOTIATION** 옵션을 허용합니다. 클라이언트는 모든 환경에서 수행되는 핸드셰이크를 사용하여 서버에 대한 여러 연결을 계속 열 수 있습니다.

(BZ#1566430)

삭제된 클러스터 노드가 더 이상 클러스터 상태에 표시되지 않습니다.

이전 버전에서는 **pcs cluster node remove** 명령을 사용하여 노드를 제거할 때 제거된 노드는 **pcs status** 디스플레이 출력에 표시되었습니다. 이번 수정으로 삭제된 노드가 더 이상 클러스터 상태에 표시되지 않습니다.

(BZ#1595829)

이제 최신, 기본 매개변수 이름 또는 더 이상 사용되지 않는 매개 변수 이름을 사용하여 펜싱 에이전트를 구성할 수 있습니다.

이전 매개변수 이름은 더 이상 사용되지 않는 동안 많은 수의 펜싱 에이전트 매개변수의 이름이 변경되었습니다. 이전에는 **--force** 옵션과 함께 사용하지 않는 한 **pcs** 가 새 매개변수를 설정할 수 없었습니다. 이번 수정으로 **pcs** 는 이제 더 이상 사용되지 않는 매개변수에 대한 지원을 유지하면서 이름이 변경된 펜싱 에이전트 매개 변수를 지원합니다.

(BZ#1436217)

pcs 명령으로 표시할 클러스터의 **XML** 상태를 올바르게 읽습니다.

pcs 명령은 **crm_mon** 유틸리티를 실행하여 클러스터의 상태를 **XML** 형식으로 가져옵니다. **crm_mon** 유틸리티는 표준 출력 및 경고에 **XML**을 표준 오류 출력에 출력합니다. 이전에는 **pcs mixed XML**과 **warnings**를 하나의 스트림에 포함시키고 **XML**로 구문 분석할 수 없었습니다. 이번 수정을 통해 표준 및 오류 출력이 **pcs** 로 분리되어 클러스터의 **XML** 상태를 읽을 수 있습니다.

(BZ#1578955)

기존 클러스터에서 노드를 사용하여 새 클러스터를 생성할 때 사용자가 더 이상 클러스터를 제거하는 것을 권장하지 않습니다.

이전 버전에서는 **pcs cluster setup** 명령을 실행하거나 **pcsd Web UI**를 사용하여 클러스터를 생성할 때 사용자가 기존 클러스터에서 노드를 지정한 경우 **pcsd Web UI**를 사용하여 오류를 보고하여 사용자가 노드에서 클러스터를 삭제하도록 제안했습니다. 결과적으로 사용자는 노드에서 클러스터를 삭제하고, 클러스터를 손상시키고 나머지 노드에서는 삭제된 노드가 클러스터의 일부로 간주되므로 노드가 손상되었습니다. 이번 수정으로 사용자는 대신 클러스터에서 노드를 제거하여 클러스터를 손상시키지 않고 문제를 해결하는 방법을 더 잘 알려주는 것이 좋습니다.

(BZ#1596050)

pcs 명령에서 더 이상 대화식으로 인증 정보를 요청하지 않습니다.

루트가 아닌 사용자가 **root** 권한이 필요한 **pcs** 명령을 실행하는 경우 **pcs** 는 로컬로 실행되는 **pcsd** 데몬에 연결하고 명령을 전달합니다. **pcsd** 데몬은 **root** 권한으로 실행되고 명령을 실행할 수 있기 때문입니다. 이전에는 사용자가 로컬 **pcsd** 데몬에 인증되지 않은 경우 **pcs** 는 사용자 이름과 암호를 대화식으로 요청했습니다. 이는 사용자에게 혼란스럽고 **pcs** 를 실행하는 스크립트에서 특수 처리가 필요했습니다. 이번 수정을 통해 사용자가 인증되지 않은 경우 **pcs** 는 수행할 작업을 설명하는 오류로 종료됩니다. 즉, **pcs** 를 **root**로 실행하거나 새로운 **pcs client local-auth** 명령을 사용하여 인증합니다. 결과적으로 **pcs** 명령은 대화식으로 인증 정보를 요청하지 않고 사용자 환경을 개선합니다.

(BZ#1554310)

이제 **crypto-policies** 가 **FUTURE** 로 설정된 경우 **pcsd** 데몬이 기본 자체 생성된 **SSL** 인증서로 시작됩니다.

FUTURE 의 **crypto-policies** 설정은 **SSL** 인증서의 **RSA** 키가 **3072b** 이상이어야 합니다. 이전 버전에서는 **2048b** 키가 있는 **SSL** 인증서를 생성하기 때문에 이 정책이 설정된 경우 **pcsd** 데몬이 시작되지 않았습니다. 이번 업데이트를 통해 **pcsd** 자체 생성 **SSL** 인증서의 키 크기가 **3072b**로 증가했으며, **pcsd** 가 기본 자체 생성된 **SSL** 인증서로 시작됩니다.

(BZ#1638852)

네트워크가 준비되면 **pcsd** 서비스가 시작됩니다.

이전 버전에서는 사용자가 특정 **IP** 주소에 바인딩하도록 **pcsd** 를 구성했는데 **pcsd** 를 시작하려고 할 때 부팅 중에 주소가 준비되지 않은 경우 **pcsd** 를 시작할 수 없어 **pcsd** 를 시작하는 데 수동 개입이 필요했습니다. 이번 수정으로 **pcsd.service** 는 **network-online.target** 에 따라 다릅니다. 결과적으로 **pcsd** 는 네트워크가 준비될 때 시작되고 **IP** 주소에 바인딩할 수 있습니다.

(BZ#1640477)

5.2.7. 네트워킹

약한 TLS 알고리즘은 더 이상 **glib-networking**에서 허용되지 않습니다.

이전 버전에서는 **glib-networking** 패키지가 RHEL 8 시스템 전체 **Crypto** 정책과 호환되지 않았습니다. 그 결과, **glib** 라이브러리를 네트워킹에 사용하는 애플리케이션에서는 관리자가 의도한 것보다 약한 알고리즘을 사용하여 TLS(Transport Layer Security) 연결을 허용할 수 있습니다. 이번 업데이트를 통해 시스템 전체 암호화 정책이 적용되고 이제 **glib**를 네트워킹에 사용하는 애플리케이션은 정책에 따라 허용되는 TLS 연결만 허용합니다.

(BZ#1640534)

5.2.8. 보안

SELinux 정책에서 **iscsiuio** 프로세스가 검색 포털에 연결 가능

이전에는 **iscsiuio** 프로세스에 대해 SELinux 정책이 너무 제한적이며 이러한 프로세스는 **mmap** 시스템 호출을 사용하여 **/dev/uio*** 장치에 액세스할 수 없었습니다. 그 결과 검색 포털에 대한 연결이 실패했습니다. 이번 업데이트에서는 SELinux 정책 및 **iscsiuio** 프로세스에 누락된 규칙이 설명된 시나리오에서 예상대로 작동합니다.

(BZ#1626446)

5.2.9. 서브스크립션 관리

이제 **DNF** 및 **yum**이 **subscription-manager** 값에 관계없이 저장소에 액세스할 수 있습니다.

이전에는 **dnf** 또는 **yum** 명령에서 **subscription-manager** 서비스에서 추가한 URL의 **https://** 접두사를 무시했습니다. 업데이트된 **dnf** 또는 **yum** 명령은 유효하지 않은 **https://** URL을 무시하지 않습니다. 그 결과 **dnf** 및 **yum**이 리포지토리에 액세스하지 못했습니다. 문제를 해결하기 위해 새 구성 변수 **proxy_scheme**가 **/etc/rhsm/rhsm.conf** 파일에 추가되어 값을 **http** 또는 **https**로 설정할 수 있습니다. 값을 지정하지 않으면 **subscription-manager**가 기본적으로 **http**를 더 일반적으로 사용합니다.

프록시가 **http**를 사용하는 경우 대부분의 사용자는 **/etc/rhsm/rhsm.conf**의 구성에서 아무것도 변경하지 않아야 합니다. 프록시에서 **https**를 사용하는 경우 사용자는 **proxy_scheme** 값을 **https**로 업데이트해야 합니다. 두 경우 모두 사용자는 **subscription-manager repos --list** 명령을 실행하거나 **rhsmcertd** 데몬 프로세스가 **/etc/yum.repos.d/redhat.repo**를 올바르게 재생성할 때까지 기다립니다.

(BZ#1654531)

5.2.10. 가상화

Azure에서 임시 디스크 마운트가 보다 안정적으로 작동합니다.

이전 버전에서는 VM이 "Stopped(deallocated)"된 다음 VM이 시작된 경우 Microsoft Azure 플랫폼에서 실행 중인 VM(가상 머신)에 임시 디스크를 마운트하지 못했습니다. 이번 업데이트를 통해 설명된 상황에서 디스크를 다시 연결하여 문제가 발생하지 않도록 합니다.

(BZ#1615599)

5.3. 기술 프리뷰

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

기술 프리뷰 기능에 대한 Red Hat 지원 범위에 대한 자세한 내용은 [기술 프리뷰 기능 지원](#) 범위를 참조하십시오.

5.3.1. 커널

eBPF를 기술 프리뷰로 이용 가능

eBPF(extended Berkeley Packet Filtering) 기능은 네트워킹 및 추적을 위해 기술 프리뷰로 사용할 수 있습니다. eBPF는 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. 이 기능에는 다양한 유형의 맵 생성을 지원하는 새로운 시스템 호출 `bpf()`가 포함되어 있으며 커널에 다양한 유형의 프로그램을 삽입할 수도 있습니다. `bpf()` syscall은 root 사용자와 같은 CAP_SYS_ADMIN 기능이 있는 사용자만 성공적으로 사용할 수 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. BCC는 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 **inode**와 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 링크를 기반으로 했습니다. <https://www.kernel.org/doc/Documentation/cgroup-v2.txt>. 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 기능을 사용하면 크래시 커널 및 **initramfs**가 조기 충돌하더라도 **vmcore** 정보를 캡처할 수 있을 만큼 충분히 일찍 로드할 수 있습니다. **early kdump**에 대한 자세한 내용은 `/usr/share/doc/kexec-tools/early-kdump-howto.txt` 파일을 참조하십시오.

(BZ#1520209)

기술 프리뷰로 제공되는 **IBMvnic** 장치 드라이버

Red Hat Enterprise Linux 8.0을 사용하면 **IBM POWER** 아키텍처용 **IBM Virtual Network Interface Controller(vNIC)** 드라이버는 기술 프리뷰로 사용할 수 있습니다. **vNIC**는 엔터프라이즈 기능을 제공하고 네트워크 관리를 단순화하는 **PowerVM** 가상 네트워킹 기술입니다. **SR-IOV NIC**와 결합할 때 가상 **NIC** 수준에서 대역폭 제어 품질(**QoS**) 기능을 제공하는 고성능의 효율적인 기술로, **vNIC**는 가상화 오버헤드를 크게 줄여 네트워크 가상화에 필요한 **CPU** 및 메모리를 포함한 대기 시간을 줄이고 서버 리소스를 줄일 수 있습니다.

(BZ#1524683)

soft-RoCE를 기술 프리뷰로 이용 가능

RoCE(Remote Direct Memory Access) over Converged Ethernet (RoCE)는 이더넷을 통해 **RDMA**를 구현하는 네트워크 프로토콜입니다. **soft-RoCE**는 **RoCE v1** 및 **RoCE v2**의 두 가지 프로토콜 버전을 지원하는 **RoCE**의 소프트웨어 구현입니다. **soft-RoCE** 드라이버인 **rdma_rxe**는 **RHEL 8**에서 지원되지 않는 기술 프리뷰로 사용할 수 있습니다.

(BZ#1605216)

5.3.2. 그래픽 인프라

64비트 ARM 아키텍처의 기술 프리뷰로 VNC 원격 콘솔 사용 가능

64비트 ARM 아키텍처에서는 VNC(Virtual Network Computing) 원격 콘솔을 기술 프리뷰로 사용할 수 있습니다. 나머지 그래픽 스택은 현재 64비트 ARM 아키텍처용으로 검증되지 않았습니다.

(BZ#1698565)

5.3.3. 하드웨어 지원

클러스터 인식 MD RAID1은 기술 프리뷰로 사용할 수 있습니다.

RAID1 클러스터는 커널 필드에서 기본적으로 활성화되어 있지 않습니다. RAID1 클러스터를 사용해 보려는 경우 먼저 RAID1 클러스터를 모듈로 커널을 빌드해야 합니다.

1. **make menuconfig** 명령을 입력합니다.
2. **make && make modules && make modules_install && make install** 명령을 입력합니다.
3. **reboot** 명령을 입력합니다.

(BZ#1654482)

5.3.4. IdM (Identity Management)

DNSSEC를 IdM에서 기술 프리뷰로 이용 가능

통합된 DNS가 있는 IdM(Identity Management) 서버는 이제 DNS 프로토콜의 보안을 향상시키는 DNS에 대한 확장 기능 세트인 DNSSEC(DNS Security Extensions)를 지원합니다. IdM 서버에서 호스팅되는 DNS 영역은 DNSSEC를 사용하여 자동으로 서명할 수 있습니다. 암호화 키는 자동으로 생성되고 순환됩니다.

DNSSEC로 DNS 영역을 보호하기로 결정한 사용자는 다음 문서를 읽고 따르는 것이 좋습니다.

- DNSSEC controlPlane Practices, 버전 2 : <http://tools.ietf.org/html/rfc6781#section-2>

- **DNS(Secure Domain Name System) 배포 가이드:**
<http://dx.doi.org/10.6028/NIST.SP.800-81-2>

- **DNSSEC 주요 롤오버 시간 고려 사항:** <http://tools.ietf.org/html/rfc7583>

통합 **DNS**가 있는 **IdM** 서버는 **DNSSEC**를 사용하여 다른 **DNS** 서버에서 얻은 **DNS** 응답을 검증합니다. 이는 권장 이름 지정 방식에 따라 구성되지 않은 **DNS** 영역의 가용성에 영향을 미칠 수 있습니다.

(BZ#1664718)

Identity Management JSON-RPC API를 기술 프리뷰로 사용 가능

IdM(Identity Management)에서 **API**를 사용할 수 있습니다. **API**를 보기 위해 **IdM**은 **API** 브라우저를 기술 프리뷰로 제공합니다.

Red Hat Enterprise Linux 7.3에서는 여러 버전의 **API** 명령을 사용하도록 **IdM API**가 개선되었습니다. 이전에는 개선 사항이 호환되지 않는 방식으로 명령의 동작을 변경할 수 있었습니다. 이제 **IdM API**가 변경되는 경우에도 기존 툴과 스크립트를 계속 사용할 수 있습니다. 이렇게 하면 다음을 활성화합니다.

- 관리자는 관리 클라이언트보다 서버에서 이전 버전의 **IdM**을 사용합니다.
- 서버에서 **IdM** 버전이 변경되더라도 특정 버전의 **IdM** 호출을 사용하는 개발자.

모든 경우에 서버와의 통신은 한 쪽이 사용하는 경우(예: 기능에 대한 새로운 옵션을 도입하는 최신 버전)와 관계없이 가능합니다.

API 사용에 대한 자세한 내용은 [Using the Identity Management API to Communicate with the IdM Server \(TECHNOLOGY PREVIEW\)](#) 를 참조하십시오.

(BZ#1664719)

5.3.5. 파일 시스템 및 스토리지

Aero 어댑터를 기술 프리뷰로 사용 가능

다음 **Aero** 어댑터는 기술 프리뷰로 사용할 수 있습니다.

- **MP t3sas** 드라이버로 제어되는 **PCI ID 0x1000:0x00e2** 및 **0x1000:0x00e6**
- **PCI ID 0x1000:0x10e5** 및 **0x1000:0x10e6**은 **NodeNetwork raid_sas** 드라이버로 제어

(BZ#1663281)

Stratis 사용 가능

Stratis는 새로운 로컬 스토리지 관리자입니다. 이는 스토리지 풀 위에 관리형 파일 시스템을 사용자에게 추가 기능을 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.

Stratis는 기술 프리뷰로 제공됩니다.

자세한 내용은 **Stratis** 설명서: [Stratis 파일 시스템](#) 설정을 참조하십시오.

(JIRA:RHELPLAN-1212)

OverlayFS

OverlayFS는 공용 파일 시스템의 유형입니다. 이를 통해 한 파일 시스템을 다른 파일 시스템에 오버레이할 수 있습니다. 변경 사항은 대문자 파일 시스템에 기록되지만 하위 파일 시스템은 수정되지 않은 상태로 유지됩니다. 이를 통해 여러 사용자가 컨테이너 또는 **DVD-ROM**과 같은 파일 시스템 이미지를 공유할 수 있습니다. 여기서 기본 이미지는 읽기 전용 미디어에 있습니다. 자세한 내용은 **Linux** 커널 설명서를 참조하십시오. <https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt>.

OverlayFS는 대부분의 상황에서 기술 프리뷰로 남아 있습니다. 따라서 이 기술이 활성화되면 커널 로그가 경고합니다.

지원되는 컨테이너 엔진(**podman** 또는 **buildah**)과 함께 사용할 때 **OverlayFS**에 대해 완전 지원을 사용할 수 있습니다.

- **OverlayFS**는 컨테이너 엔진 그래프 드라이버로만 사용할 수 있도록 지원됩니다. 영구 스토리지에서는 컨테이너 **COW** 콘텐츠에만 사용이 지원됩니다. 모든 영구 스토리지를 **OverlayFS**가 아닌 볼륨에 배치해야 합니다. 기본 컨테이너 엔진 구성만 사용할 수 있습니다. 즉, 하나의 오버레이 수준, 하위 디렉터리 1개 및 상위 수준은 모두 동일한 파일 시스템에 있습니다.

- 현재 **XFS**만 하위 계층 파일 시스템으로 사용할 수 있습니다.

또한 **OverlayFS** 사용에 다음 규칙 및 제한이 적용됩니다.

- **OverlayFS** 커널 **ABI** 및 사용자 공간 동작은 안정적인 것으로 간주되지 않으며 향후 업데이트의 변경 사항을 확인할 수 있습니다.
- **OverlayFS**는 제한된 **POSIX** 표준 세트를 제공합니다. **OverlayFS**를 사용하여 배포하기 전에 철저히 애플리케이션을 테스트합니다. 다음 사례는 **POSIX**와 호환되지 않습니다.
 - **O_RDONLY**로 열린 하위 파일은 파일을 읽을 때 **st_atime** 업데이트가 수신되지 않습니다.
 - **O_RDONLY**로 열린 하위 파일은 **MAP_SHARED**로 매핑된 후 후속 수정과 일치하지 않습니다.
 - 완전히 호환되는 **st_ino** 또는 **d_ino** 값은 **RHEL 8**에서 기본적으로 활성화되어 있지 않지만 모듈 옵션 또는 마운트 옵션으로 전체 **POSIX** 준수를 활성화할 수 있습니다.

일관된 **inode** 번호 지정을 얻으려면 **xino=on** 마운트 옵션을 사용합니다.

redirect_dir=on 및 **index=on** 옵션을 사용하여 **POSIX** 규정 준수를 향상시킬 수도 있습니다. 이 두 옵션을 사용하면 상위 계층의 포맷이 이러한 옵션 없이 오버레이와 호환되지 않습니다. 즉, **redirect_dir=on** 또는 **index=on** 으로 오버레이를 생성하고 오버레이를 마운트 해제한 다음 이러한 옵션 없이 오버레이를 마운트하면 예기치 않은 결과 또는 오류가 발생할 수 있습니다.

- **XFS에 사용되는 명령:**
 - 오버레이로 사용하려면 **XFS** 파일 시스템을 **-n ftype=1** 옵션을 사용하여 생성해야 합니다.
 - **rootfs** 및 시스템 설치 중에 생성된 파일 시스템을 사용하여 **Anaconda kickstart**에서 **-mkfsoptions=-n ftype=1** 매개변수를 설정합니다.
 - 설치 후 새 파일 시스템을 생성할 때 **# mkfs -t xfs -n ftype=1 /PATH/TO/DEVICE** 명령을 실행합니다.
 - 기존 파일 시스템이 오버레이로 사용할 수 있는지 확인하려면 **# xfs_info /PATH/TO/DEVICE | grep ftype** 명령을 실행하여 **ftype=1** 옵션이 활성화되어 있는지 확인합니다.
- **SELinux** 보안 레이블은 **OverlayFS**가 있는 지원되는 모든 컨테이너 엔진에서 기본적으로 활성화됩니다.
- 이 릴리스에서는 **OverlayFS**와 관련된 몇 가지 알려진 문제가 있습니다. 자세한 내용은 **Linux** 커널 설명서의 *비표준 동작*을 참조하십시오.
<https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt>.

(BZ#1690207)

이제 파일 시스템 **DAX**를 기술 프리뷰로 **ext4** 및 **XFS**에 사용할 수 있습니다.

Red Hat Enterprise Linux 8.0에서 파일 시스템 **DAX**는 기술 프리뷰로 제공됩니다. **DAX**는 애플리케이션이 영구 메모리를 주소 공간에 직접 매핑하는 수단을 제공합니다. **DAX**를 사용하려면 일반적으로 하

나 이상의 비접근 다각형 메모리 모듈(NVDIMM) 및 DAX를 지원하는 파일 시스템을 NVDIMM(NVDIMM)에서 시스템에서 사용할 수 있는 일종의 영구 메모리가 있어야 합니다. 또한 **dax** 마운트 옵션을 사용하여 파일 시스템을 마운트해야 합니다. 그런 다음 **dax-mounted** 파일 시스템에 있는 파일의 **mmap** 을 사용하면 애플리케이션의 주소 공간에 스토리지를 직접 매핑합니다.

(BZ#1627455)

5.3.6. 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 Hat**은 **Red Hat Openstack**에 대한 **Pacemaker** 번들의 사용을 완전하게 지원합니다.

(BZ#1619620)

5.3.7. 네트워킹

XDP를 기술 프리뷰로 이용 가능

기술 프리뷰로 사용할 수 있는 **eXpress Data Path(XDP)** 기능은 커널 수신 데이터 경로의 초기 시점에서 고성능 패킷 처리를 위해 **eBPF(extended Berkeley Packet Filter)** 프로그램을 첨부할 수 있는 수단을 제공하여, 효율적으로 프로그래밍 가능한 패킷 분석, 필터링 및 조작을 가능하게 합니다.

(BZ#1503672)

Tc용 **eBPF**를 기술 프리뷰로 이용 가능

기술 프리뷰로, **Tc**(트래픽 제어) 커널 하위 시스템과 **tc** 툴은 **eBPF(extended Berkeley Packet Filtering)** 프로그램을 패킷 **classified**으로 연결하고 수신 및 송신 대기열 작업 모두에 대한 작업을 연결할 수 있습니다. 이를 통해 커널 네트워크 데이터 경로 내에서 프로그래밍 가능한 패킷 처리를 가능하게 합니다.

(BZ#1699825)

AF_XDP 를 기술 프리뷰로 이용 가능

주소 제품군 **eXpress Data Path (AF_XDP)** 소켓은 고성능 패킷 처리를 위해 설계되었습니다. **XDP**를 사용하고 추가 처리를 위해 프로그래밍 방식으로 선택된 패킷을 사용자 공간 애플리케이션으로 효율적

으로 리디렉션합니다.

(BZ#1633143)

KTLS를 기술 프리뷰로 이용 가능

Red Hat Enterprise Linux 8에서 **KSM(Kernel Transport Layer Security)**은 기술 프리뷰로 제공됩니다. **KTLS**는 **AES-GCM** 암호화에 대해 커널의 대칭 암호화 또는 암호 해독 알고리즘을 사용하여 **TLS** 레코드를 처리합니다. **KTLS**는 이 기능을 지원하는 **NIC(Network Interface Controller)**로 **TLS** 레코드 암호화를 오프로드하는 인터페이스도 제공합니다.

(BZ#1570255)

idrac PC를 기술 프리뷰로 이용 가능

EgressPC(Transparent Inter Process Communication)는 느슨하게 연결된 노드의 클러스터 내에서 효율적인 통신을 위해 특별히 설계된 프로토콜입니다. 커널 모듈로 작동하며 **iproute2** 패키지의 **tips c** 툴을 제공하여 설계자가 클러스터 내의 위치에 관계없이 다른 애플리케이션과 빠르고 안정적으로 통신할 수 있는 애플리케이션을 만들 수 있습니다. 이 기능은 기술 프리뷰로 제공됩니다.

(BZ#1581898)

systemd-resolved 서비스가 기술 프리뷰로 사용 가능

systemd-resolved 서비스는 로컬 애플리케이션에 이름 확인을 제공합니다. 이 서비스는 캐싱 및 **DNS** 스텝 확인자, **LLMNR(Link-Local Multicast Name Resolution)**, **Multicast DNS resolver** 및 **responder**를 구현합니다.

systemd 패키지에서 **systemd** 확인 기능을 제공하는 경우에도 이 서비스는 지원되지 않는 기술 프리뷰입니다.

(BZ#1906489)

5.3.8. Red Hat Enterprise Linux System Roles

RHEL System Roles의 **postfix** 역할을 기술 프리뷰로 사용 가능

Red Hat Enterprise Linux System Roles는 **Red Hat Enterprise Linux** 하위 시스템에 대한 구성 인터페이스를 제공하므로 **Ansible** 역할이 포함되어 시스템을 보다 쉽게 구성할 수 있습니다. 이 인터페이스

를 통해 여러 버전의 **Red Hat Enterprise Linux**에서 시스템 구성을 관리하고 새로운 주요 릴리스를 채택할 수 있습니다.

rhel-system-roles 패키지는 **AppStream** 리포지토리를 통해 배포됩니다.

postfix 역할은 기술 프리뷰로 사용할 수 있습니다.

다음 역할이 완전히 지원됩니다.

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

자세한 내용은 [RHEL 시스템 역할에 대한 지식베이스 문서](#)를 참조하십시오.

(BZ#1812552)

5.3.9. 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 **SEV**는 **VM** 메모리를 암호화하여 호스트가 **VM**의 데이터에 액세스할 수 없도록 합니다. 이제 호스트가 악성 코드에 의해 감염된 경우 **VM**의 보안이 향상됩니다.

단일 호스트에서 이 기능을 한 번에 사용할 수 있는 **VM**의 개수는 호스트 하드웨어에 의해 결정됩니다. 현재 **AMD EPYC** 프로세서는 **SEV**를 사용하여 실행 중인 최대 **15개**의 **VM**을 지원합니다.

또한 **SEV**가 부팅되도록 구성된 **VM**의 경우에도 하드 메모리 제한을 사용하여 **VM**을 구성해야 합니다. 이렇게 하려면 **VM**의 **XML** 구성에 다음을 추가합니다.

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

N에 권장되는 값은 게스트 **RAM + 256MiB**입니다. 예를 들어 게스트에 **2GiB RAM**이 할당되면 **N**은 **2359296** 이상이어야 합니다.

(BZ#1501618, BZ#1501607)

Intel vGPU

기술 프리뷰로 물리적 **Intel GPU** 장치를 중재 장치라고 하는 여러 가상 장치로 나눌 수 있습니다. 그런 다음 이러한 미디어 장치를 가상 **GPU**로 여러 가상 머신(**VM**)에 할당할 수 있습니다. 결과적으로 이러한 **VM**은 단일 물리적 **Intel GPU**의 성능을 공유합니다.

선택한 **Intel GPU**만 **vGPU** 기능과 호환됩니다. 또한 **VM**에 물리적 **GPU**를 할당하면 호스트가 **GPU**를 사용할 수 없으며 호스트의 그래픽 디스플레이 출력이 작동하지 않을 수 있습니다.

(BZ#1528684)

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 **VM**을 실행할 수 있습니다.

중첩된 가상화는 **AMD64** 및 **Intel 64** 시스템에서도 기술 프리뷰로 남아 있습니다.

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

- **RHEL 8**

POWER 9용 RHEL 7

(BZ#1505999, BZ#1518937)

RHEL 8 Hyper-V 가상 머신에서 KVM 가상화 사용 가능

기술 프리뷰로 중첩된 KVM 가상화는 이제 Microsoft Hyper-V 하이퍼바이저에서 사용할 수 있습니다. 따라서 Hyper-V 호스트에서 실행되는 RHEL 8 게스트 시스템에서 가상 머신을 만들 수 있습니다.

현재 이 기능은 Intel 시스템에서만 작동합니다. 또한 중첩된 가상화는 경우에 따라 Hyper-V에서 기본적으로 활성화되어 있지 않습니다. 사용하도록 설정하려면 다음 Microsoft 문서를 참조하십시오.

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

(BZ#1519039)

5.4. 사용되지 않는 기능

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

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.

더 이상 사용되지 않는 기능의 지원 상태는 Red Hat Enterprise Linux 8에서는 변경되지 않습니다. 지원 기간에 대한 자세한 내용은 [Red Hat Enterprise Linux 라이프 사이클](#) 및 [Red Hat Enterprise Linux Application Streams 라이프 사이클](#)을 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.

RHEL 7에 존재하지만 **RHEL 8**에서 *제거*된 기능에 대한 자세한 내용은 **RHEL 8 채택 시 고려 사항을** 참조하십시오.

5.4.1. 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)

몇 가지 **Kickstart** 명령 및 옵션이 더 이상 사용되지 않음

RHEL 8 Kickstart 파일에서 다음 명령과 옵션을 사용하면 로그에 경고가 출력됩니다.

- **auth** 또는 **authconfig**
- **device**
- **deviceprobe**
- **dmraid**
- **install**
- **lilo**
- **lilocheck**

- **mouse**
- **multipath**
- **bootloader --upgrade**
- **ignoredisk --interactive**
- **partition --active**
- **reboot --kexec**

특정 옵션만 나열된 경우에도 기본 명령 및 기타 옵션은 계속 사용할 수 있으며 더 이상 사용되지 않습니다.

Kickstart의 자세한 내용 및 관련 변경 사항은 ***RHEL 8 도입 시 고려 사항***의 **Kickstart 변경** 섹션을 참조하십시오.

(BZ#1642765)

5.4.2. 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 **UDP**를 통한 **NFS**가 더 이상 지원되지 않습니다.

(BZ#1592011)

멀티플레이어 커널 명령행 매개변수가 더 이상 사용되지 않음

이전 **RHEL** 릴리스에서 모든 장치에 대한 디스크 스케줄러를 설정하는 데 멀티플레이어 커널 명령줄 매개 변수가 사용되었습니다. **RHEL 8**에서는 매개변수가 더 이상 사용되지 않습니다.

업스트림 **Linux** 커널은 멀티플레이어 매개 변수에 대한 지원을 제거했지만 호환성을 위해 **RHEL 8**에서 계속 사용할 수 있습니다.

커널은 장치 유형에 따라 기본 디스크 스케줄러를 선택합니다. 일반적으로 최적의 설정입니다. 다른 스케줄러가 필요한 경우 **Red Hat**은 **udev** 규칙 또는 **Tuned** 서비스를 사용하여 구성하는 것이 좋습니다. 선택한 장치와 일치하고 해당 장치에 대해서만 스케줄러를 전환합니다.

자세한 내용은 다음 문서를 참조하십시오. [RHEL8에서 'elevator=' 매개변수가 더 이상 작동하지 않는 이유](#)를 참조하십시오.

(BZ#1665295)

VDO 패키지의 VDO Ansible 모듈

VDO Ansible 모듈은 현재 **controlPlane RPM** 패키지에서 제공합니다. 향후 릴리스에서는 **VDO Ansible** 모듈이 **Ansible RPM** 패키지로 이동합니다.

(BZ#1669537)

5.4.3. 네트워킹

RHEL 8에서 네트워크 스크립트가 더 이상 사용되지 않음

네트워크 스크립트가 **Red Hat Enterprise Linux 8**에서 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 8**에서 **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)

5.4.4. 커널

rdma_rxe soft-RoCE 드라이버가 더 이상 사용되지 않음

RXE(Remote Direct Memory Access over Converged Ethernet)는 **RXE(Remote Direct Memory Access)**를 에뮬레이션하는 기능입니다. **RHEL 8**에서는 **soft-RoCE** 기능을 지원되지 않는 기술 프리뷰로 사용할 수 있습니다. 그러나 안정성 문제로 인해 이 기능은 더 이상 사용되지 않으며 **RHEL 9**에서 제거됩니다.

(BZ#1878207)

5.4.5. 보안

RHEL 8에서는 **DSA**가 더 이상 사용되지 않음

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** 시스템 전체 암호화 정책 수준에서도 **DSA** 호스트 키를 허용하지 않습니다.

(BZ#1646541)

NSS에서 **SSL2 Client Hello** 가 더 이상 사용되지 않음

TLS(Transport Layer Security) 프로토콜 버전 1.2 이하에서는 **SSL(Secure Sockets Layer)** 프로토콜 버전 2와 역호환되는 방식으로 포맷된 **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 8**의 이후 릴리스

에서 완전히 삭제될 수 있습니다.

(BZ#1645153)

TLS 1.0 및 TLS 1.1이 더 이상 사용되지 않음

TLS 1.0 및 TLS 1.1 프로토콜은 **DEFAULT** 시스템 전체의 암호화 정책 수준에서 비활성화됩니다. 예를 들어 **Firefox** 웹 브라우저의 비디오 회의 애플리케이션에 대한 시나리오가 필요한 경우 더 이상 사용되지 않는 프로토콜을 사용해야 하는 경우 시스템 전체 암호화 정책을 **LEGACY** 수준으로 전환합니다.

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

자세한 내용은 [Red Hat 고객 포털](#) 및 [update-crypto-policies\(8\) 매뉴얼 페이지](#)에서 더 약한 암호화 알고리즘 지식 베이스 문서 사용 중단 및 [RHEL 8의 tekton 암호화 기본값](#) 을 참조하십시오.

(BZ#1660839)

5.4.6. 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 8**에서 **VM** 스냅샷을 사용하지 않는 것이 좋습니다.

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

(BZ#1686057)

Cirrus VGA 가상 GPU 유형이 더 이상 사용되지 않음

향후 **Red Hat Enterprise Linux**에 대한 주요 업데이트가 있을 경우 **KVM** 가상 머신에서 **Cirrus VGA GPU** 장치가 더 이상 지원되지 않습니다. 따라서 **Red Hat**은 **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. **Cockpit**라고도 하는 **RHEL 8** 웹 콘솔은 후속 릴리스에서 교체될 예정입니다. 따라서 **GUI**에서 가상화를 관리하기 위해 웹 콘솔을 사용하는 것이 좋습니다. 그러나 **Red Hat Enterprise Linux 8.0**의 일부 기능은 **virt-manager** 또는 명령줄에서만 액세스할 수 있습니다.

(JIRA:RHELPLAN-10304)

5.4.7. 더 이상 사용되지 않는 패키지

다음 패키지는 더 이상 사용되지 않으며 향후 **Red Hat Enterprise Linux** 주요 릴리스에 포함되어 있지 않을 것입니다.

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

- **yp-tools**
- **ypbind**
- **ypserv**

5.5. 확인된 문제

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

5.5.1. 웹 콘솔

session_historying shell을 사용하여 **RHEL** 웹 콘솔에 로그인할 수 없습니다.

현재 **tlog** 기록 지원 사용자에게 대해 **RHEL** 웹 콘솔 로그인이 실패합니다. **RHEL** 웹 콘솔에는 로그인 성공할 수 있도록 **/etc/shells** 디렉토리에 사용자 셸이 있어야 합니다. 그러나 **tlog-rec-session** 이 **/etc/shells**에 추가되면 기록된 사용자는 셸을 **tlog-rec-session**에서 **/etc/shells**에서 다른 셸로 변경하여 레코딩을 비활성화할 수 있습니다. **Red Hat**은 이러한 이유로 **tlog-rec-session**을 **/etc/shell**에 추가하지 않는 것이 좋습니다.

(BZ#1631905)

5.5.2. 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)

xorg-x11-drv-fbdev, **xorg-x11-drv-vesa** 및 **xorg-x11-drv-vmware** 비디오 드라이버는 기본적으로 설치되지 않습니다.

특정 **AMD** 가속 처리 장치가 있는 특정 **NVIDIA** 그래픽 카드 및 워크스테이션이 있는 워크스테이션은 **RHEL 8.0 Server** 설치 후 그래픽 로그인 창을 표시하지 않습니다.

이 문제를 해결하려면 워크스테이션 시스템에서 **RHEL 8.0 Workstation** 을 설치합니다. 워크스테이션에 **RHEL 8.0 Server** 설치가 필요한 경우 **yum -y groupinstall base-x** 명령을 실행하여 설치 후 **base-x** 패키지 그룹을 수동으로 설치합니다.

또한 **Hyper-V**와 같은 그래픽 지원을 위한 **EFI**에 의존하는 가상 머신도 영향을 받습니다. **Hyper-V**에서 **GUI** 기본 환경을 사용하여 서버를 선택한 경우 재부팅 시 표시되는 검정 화면으로 인해 로그인할 수 없게 될 수 있습니다. **Hyper-v**에서 이 문제를 해결하려면 다음 단계를 사용하여 다중 사용자 모드를 활성화합니다.

1.
가상 머신을 재부팅합니다.
2.
부팅 프로세스 중에 키보드의 위쪽 및 아래쪽 화살표 키를 사용하여 필요한 커널을 선택합니다.
3.
키보드에서 **e** 키를 눌러 커널 명령줄을 편집합니다.
4.
systemd.unit=multi-user.target 을 **GRUB**의 커널 명령줄에 추가합니다.
5.
Ctrl-X 를 눌러 가상 머신을 시작합니다.
6.
로그인한 후 **yum -y groupinstall base-x** 명령을 실행합니다.
7.
가상 머신을 재부팅하여 그래픽 모드에 액세스합니다.

(BZ#1687489)

reboot --kexec 명령을 사용하면 설치에 실패합니다.

reboot --kexec 명령이 포함된 **Kickstart** 파일을 사용할 때 **RHEL 8** 설치에 실패합니다. 문제가 발생하지 않도록 하려면 **Kickstart** 파일에서 **--kexec**를 재부팅하지 않고 **reboot** 명령을 사용하십시오.

(BZ#1672405)

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

로컬 설치 중에 **RHEL 8 Binary DVD.iso** 이미지 파일을 파티션에 복사하는 동안 **cp <path>/\ * <mounted partition>/dir** 명령의 *는 **.treeinfo** 및 **.discinfo** 파일을 복사하지 못합니다. 이러한 파일은 성공적으로 설치하려면 필요합니다. 결과적으로 **BaseOS** 및 **AppStream** 리포지토리가 로드되지 않으며 **anaconda.log** 파일에 있는 디버그 관련 로그 메시지가 문제의 유일한 레코드입니다.

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

(BZ#1692746)

Anaconda 설치에는 최소한의 리소스 설정 요구 사항이 적은 제한이 포함되어 있습니다.

Anaconda는 최소한의 리소스 설정을 사용하여 시스템에 설치를 시작하고 설치를 성공적으로 수행하는 데 필요한 리소스에 대한 이전 메시지 경고를 제공하지 않습니다. 결과적으로 설치에 실패할 수 있으며 출력 오류로 인해 가능한 디버그 및 복구에 대한 명확한 메시지가 표시되지 않습니다. 이 문제를 해결하려면 시스템에 설치에 필요한 최소한의 리소스 설정이 있는지, **PPC64(LE)**의 **2GB** 메모리 및 **x86_64**의 **1GB**가 있는지 확인하십시오. 따라서 성공적으로 설치를 수행할 수 있어야 합니다.

(BZ#1696609)

reboot --kexec 및 **inst.kexec** 명령은 예측 가능한 시스템 상태를 제공하지 않습니다.

reboot --kexec Kickstart 명령 또는 **inst.kexec** 커널 부팅 매개변수를 사용하여 **RHEL** 설치를 수행해도 전체 재부팅과 동일한 예측 가능한 시스템 상태가 제공되지 않습니다. 결과적으로 재부팅하지 않고 설치된 시스템으로 전환하면 예기치 않은 결과가 발생할 수 있습니다.

kexec 기능은 더 이상 사용되지 않으며 향후 **Red Hat Enterprise Linux** 릴리스에서 제거될 예정입니다.

(BZ#1697896)**5.5.3. 커널**

부팅 시 **i40iw** 모듈이 자동으로 로드되지 않음

많은 **i40e NIC**에서 **iWarp**를 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 **LUN**에 대한 다중 경로가 있는 경우에 이러한 상황이 발생합니다. 커널이 수행하고 있는 다른 작업 외에도 콘솔 메시지가 많아 커널이 중지된 것처럼 보이기 때문에 커널 **watchdog**가 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 lockup**으로 인해 **vmcore** 이미지가 캡처됩니다.

이 문제를 해결하려면 **watchdog** 잠금 타이머를 늘려야 합니다. 이렇게 하려면 커널 명령줄에 **watchdog_thresh=N** 옵션을 추가합니다. **N**을 초 단위로 바꿉니다.

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

스토리지의 경우 장치 개수는 모든 **LUN**에 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()** 함수에 의해 설정된 메모리 정책을 무시하고 일부 메모리 영역의 페이지를 정책과 일치하지 않는 NUMA(Non-Uniform Memory Access) 노드에 병합할 수 있습니다.

이 문제를 해결하려면 QEMU에서 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)**qede 드라이버는 NIC를 중지하고 사용할 수 없게 만듭니다.**

버그로 인해 41000 및 45000 QLogic 시리즈 NIC의 qede 드라이버로 인해 펌웨어 업그레이드 및 디버깅 데이터 수집 작업이 실패하고 호스트의 재부팅 (PCI reset)이 재부팅될 때까지 NIC를 사용할 수 없거나 **ung** 상태가 다시 작동할 수 있습니다.

이 문제는 다음 모든 시나리오에서 탐지되었습니다.

- **inbox** 드라이버를 사용하여 NIC의 펌웨어를 업그레이드할 때
- **ethtool -d ethx** 명령을 실행하는 디버깅 데이터를 수집할 때
- **ethtool -d ethx**가 포함되어 있으므로 **sosreport** 명령을 실행합니다.
- **inbox** 드라이버가 IO 시간 초과, **mail Box** 명령 시간 제한, 하드웨어 **Attention**과 같은 자동 디버깅 데이터 컬렉션을 시작할 때.

이 문제를 해결하기 위해 Red Hat의 향후 에라타가 Red Hat 버그 권고 (RHBA)를 통해 릴리스될 예정입니다. 이 문제를 해결하려면 RHBA가 릴리스될 때까지 해당 문제에 대해 지원되는 수정 사항을 요청하려면 <https://access.redhat.com/support>에서 케이스를 생성합니다.

(BZ#1697310)

Rax 트리 기호가 **kernel-abi-whitelists**에 추가되었습니다.

Red Hat Enterprise Linux 8에서 **kernel-abi-whitelists** 패키지에 다음과 같은 방사 트리 기호가 추가되었습니다.

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

위의 기호는 존재하지 않으며 **RHEL8** 허용 목록에서 제거됩니다.

(BZ#1695142)

Podman 이 RHEL 8에서 컨테이너를 체크포인트하지 못했습니다.

Red Hat Enterprise Linux 8에서는 체크포인트 및 **CRIU**(사용자 공간 복원) 패키지의 버전이 오래되었습니다. 결과적으로 **CRIU**는 컨테이너 검사점 및 복원 기능을 지원하지 않으며 **podman** 유틸리티는 컨

테이너를 검사하지 못합니다. **podman container checkpoint** 명령을 실행하면 다음 오류 메시지가 표시 됩니다. '컨테이너 확인에는 **CRIU 31100** 이상이 필요합니다.

(BZ#1689746)

dracut.conf에서 **add_dracutmodules+=early kdump** 옵션을 사용하면 **early-kdump** 및 표준 **kdump** 가 실패합니다.

현재 **early-kdump**에 설치되어 있는 커널 버전 간에 불일치가 발생하고 커널 버전 **initramfs**가 에 대해 생성됩니다. 그 결과 **early-kdump**를 사용하여 부팅하면 **early-kdump**가 실패합니다. 또한 **early-kdump**가 표준 **kdump initramfs** 이미지에 포함되어 있음을 탐지하면 종료를 강제 적용합니다. 따라서 **early-kdump**가 기본 **dracut** 모듈로 추가된 경우 **kdump initramfs**를 다시 빌드하려고 할 때 표준 **kdump** 서비스도 실패합니다. 그 결과 **early-kdump** 및 표준 **kdump**가 모두 실패합니다. 이 문제를 해결하려면 **add_dracutmodules+=earlykdump** 또는 **dracut.conf** 파일에 동등한 구성을 추가하지 마십시오. 결과적으로 **early-kdump**는 기본적으로 **dracut**에 포함되지 않으므로 문제가 발생하지 않습니다. 그러나 **early-kdump** 이미지가 필요한 경우 수동으로 생성해야 합니다.

(BZ#1662911)

디버그 커널은 **RHEL 8**의 크래시 캡처 환경에서 부팅되지 않음

디버그 커널의 메모리 요구 특성으로 인해 디버그 커널이 사용 중이며 커널 패닉이 트리거되면 문제가 발생합니다. 결과적으로 **debug** 커널은 캡처 커널로 부팅할 수 없으며, 스택 추적이 대신 생성됩니다. 이 문제를 해결하려면 그에 따라 크래시 커널 메모리를 늘리십시오. 결과적으로 디버그 커널이 크래시 캡처 환경에서 성공적으로 부팅됩니다.

(BZ#1659609)

fadump를 사용할 때 네트워크 인터페이스 이름이 **kdump-<interface-name>**으로 변경됩니다.

펌웨어 지원 덤프(**fadump**)를 사용하여 **vmcore**를 캡처하고 **SSH** 또는 **NFS** 프로토콜을 사용하여 원격 머신에 저장할 때 **<interface-name>**이 일반적인 경우 네트워크 인터페이스 이름이 **kdump- <interface-name>**으로 변경됩니다(예: ***eth#** 또는 **net#**). 이 문제는 초기 **RAM 디스크(initrd)**의 **vmcore** 캡처 스크립트가 **kdump-** 접두사를 네트워크 인터페이스 이름에 추가하기 때문에 발생합니다. 일반 부팅에도 동일한 **initrd**가 사용되므로 프로텍션 커널에서도 인터페이스 이름이 변경됩니다.

(BZ#1745507)

5.5.4. 소프트웨어 관리

root가 아닌 사용자에서 **yum list**를 실행하면 **YUM** 충돌 발생

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

(BZ#1642458)

YUM v4 는 기본적으로 사용할 수 없는 리포지토리를 건너뜁니다.

YUM v4 는 모든 리포지토리의 기본적으로 "**skip_if_unavailable=True**" 설정으로 설정됩니다. 결과적으로 필요한 리포지토리를 사용할 수 없는 경우 리포지토리의 패키지는 설치, 검색 또는 업데이트 작업에서 고려되지 않습니다. 결과적으로 일부 **yum** 명령 및 **yum** 기반 스크립트는 사용할 수 없는 리포지토리가 있는 경우에도 종료 코드 **0**으로 성공합니다.

현재는 **libdnf** 패키지를 업데이트하는 것보다 다른 해결방법은 없습니다.

(BZ#1679509)

5.5.5. 인프라 서비스

nslookup 및 호스트 유틸리티는 재귀를 사용할 수 없는 이름 서버의 응답을 무시합니다.

이름 서버에서 더 많은 이름 서버를 구성하고 재귀를 사용할 수 없는 경우 **nslookup** 및 호스트 유틸리티는 마지막으로 구성된 경우가 아니면 이름 서버의 응답을 무시합니다. 마지막으로 구성된 이름 서버의 경우 사용 가능한 재귀 플래그를 사용하지 않고도 응답이 허용됩니다. 그러나 마지막으로 구성된 이름 서버가 응답하지 않거나 연결할 수 없는 경우 이름 확인이 실패합니다.

이 문제를 해결하려면 다음을 수행하십시오.

- 구성된 이름 서버가 항상 사용 가능한 재귀 플래그 세트로 응답하는지 확인합니다.
- 모든 내부 클라이언트에 대한 재귀를 허용합니다.

이 문제를 해결하려면 **dig** 유틸리티를 사용하여 재귀를 사용할 수 있는지 여부를 탐지할 수도 있습니다.

(BZ#1599459)

5.5.6. 셸 및 명령행 툴

net-snmp 패키지의 **Python** 바인딩을 사용할 수 없음

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

(BZ#1584510)

디버그 모드에서 **systemd** 는 불필요한 로그 메시지 생성

디버그 모드에서 **systemd** 시스템 및 서비스 관리자는 다음으로 시작하는 불필요한 로그 메시지를 생성합니다.

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

다음은 실행하여 메시지를 나열합니다.

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

이러한 디버그 메시지는 무해하며 무시해도 됩니다.

현재는 사용할 수 있는 해결방법이 없습니다.

(BZ#1658691)

KEYBD 트랩을 사용한 **KSH**는 멀티바이트 문자를 잘못 처리

KEYBD 트랩이 활성화된 경우 **Korn Shell(KSH)**은 멀티바이트 문자를 올바르게 처리할 수 없습니다. 그 결과, 사용자가 (예: 일본어 문자)를 입력하면 **ksh** 에 잘못된 문자열이 표시됩니다. 이 문제를 해결하려면 다음 줄을 주석으로 처리하여 **/etc/kshrc** 파일에서 **KEYBD** 트랩을 비활성화합니다.

```
trap keybd_trap KEYBD
```

자세한 내용은 관련 [지식베이스 솔루션](#)을 참조하십시오.

(BZ#1503922)

5.5.7. 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 7 용 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 VirtualHost에 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 설정

정하여 **TLS 1.3**의 사용을 비활성화합니다.

(BZ#1632600)

생성된 **Scala** 문서가 읽을 수 없음

scaladoc 명령을 사용하여 문서를 생성할 때 **JavaScript** 리소스가 누락되어 결과 **HTML** 페이지를 사용할 수 없습니다.

(BZ#1641744)

5.5.8. 데스크탑

Q XL은 **Wayland** 기반 **VM**에서 작동하지 않습니다.

qxl 드라이버는 특정 하이퍼바이저에서 커널 모드 설정 기능을 제공할 수 없습니다. 그 결과, **Wayland** 프로토콜을 기반으로 하는 그래픽은 **qxl**를 사용하는 가상 머신(**VM**)에서 사용할 수 없으며 **Wayland** 기반 로그인 화면이 시작되지 않습니다.

이 문제를 해결하려면 다음 중 하나를 사용하십시오.

- **QuarkXpress Element Library (QXL)** 그래픽을 기반으로 하는 **VM**에서 **Wayland**의 **GNOME Shell** 대신 **Xorg** 디스플레이 서버.

또는

- **VM**의 **qxl** 드라이버 대신 **virtio** 드라이버입니다.

(BZ#1641763)

systemctl isolate multi-user.target을 실행할 때 콘솔 프롬프트가 표시되지 않습니다.

systemctl isolate multi-user.target 명령을 **GNOME** 데스크탑 세션의 **GNOME Terminal**에서 실행하면 콘솔 프롬프트가 아닌 커서만 표시됩니다. 이 문제를 해결하려면 **Ctrl+Alt+F2** 키를 누릅니다. 결과적으로 콘솔 프롬프트가 표시됩니다.

이 동작은 **Wayland** 및 **X.Org** 디스플레이 서버의 **GNOME Shell**에 모두 적용됩니다.

(BZ#1678627)

5.5.9. 그래픽 인프라

X.Org 에서 실행되는 데스크탑은 낮은 화면 해상도로 변경될 때 중지됨

X.Org 디스플레이 서버와 함께 **GNOME** 데스크탑을 사용할 때 화면 해상도를 낮은 값으로 변경하려고 하면 데스크탑이 응답하지 않습니다. 이 문제를 해결하려면 화면 해상도를 **800 × 600**픽셀보다 낮은 값으로 설정하지 마십시오.

(BZ#1655413)

Radeon 이 하드웨어를 올바르게 재설정하지 못했습니다.

현재 **radeon** 커널 드라이버는 **kexec** 컨텍스트의 하드웨어를 올바르게 재설정하지 않습니다. 대신 **radeon** 이 대체되어 **kdump** 서비스의 나머지 부분이 실패합니다.

이 문제를 해결하려면 **/etc/ kdump.conf** 파일에 다음 행을 추가하여 **kdump**에서 **radeon** 을 블랙리스트로 설정합니다.

```
dracut_args --omit-drivers "radeon"
force_rebuild 1
```

머신을 다시 시작하고 **kdump** 를 다시 시작합니다. **kdump** 를 시작한 후 **force_rebuild 1** 행이 구성 파일에서 제거될 수 있습니다.

이 시나리오에서는 **kdump** 중에 그래픽을 사용할 수 없지만 **kdump** 가 성공적으로 작동합니다.

(BZ#1694705)

5.5.10. 하드웨어 지원

ARP 링크 모니터를 사용할 때 백업 슬레이브 **MII** 상태가 작동하지 않음

기본적으로 **i40e** 드라이버에서 관리하는 장치는 수신 필터 중 하나와 일치하는 소스 **Media Access**

Control(MAC) 주소가 있는 패킷을 삭제하는 소스 정리를 수행합니다. 그 결과 채널 본딩에서 **Address Resolution Protocol(ARP)** 모니터링을 사용할 때 백업 슬레이브 **Media Independent Interface (MII)** 상태가 작동하지 않습니다. 이 문제를 해결하려면 다음 명령을 사용하여 소스 정리를 비활성화합니다.

```
# ethtool --set-priv-flags <ethX> disable-source-pruning on
```

결과적으로 백업 슬레이브 **MII** 상태가 예상대로 작동합니다.

(BZ#1645433)

경우에 따라 **HP NMI watchdog**가 크래시 덤프를 생성하지 않음

HP NMI 위치독의 **hpwdt** 드라이버는 **NMI**가 대신 **perfmon** 드라이버에서 소비했기 때문에 **HPE** 위치독 타이머에 의해 생성된 **NMI(Non-maskable interrupt)**를 요청할 수 없는 경우가 있습니다.

결과적으로 경우에 따라 **hpwdt**는 패닉을 호출하여 크래시 덤프를 생성할 수 없습니다.

(BZ#1602962)

5.5.11. IdM (Identity Management)

KCM 인증 정보 캐시가 단일 인증 정보 캐시의 많은 인증 정보에 적합하지 않음

Kerberos 자격 증명 관리자(**KCM**)는 최대 **64KB**의 **ccache** 크기를 처리할 수 있습니다. 인증 정보가 너무 많으면 **kinit**와 같은 **Kerberos** 작업은 **sssd-kcm** 구성 요소와 기본 데이터베이스 간에 데이터를 전송하는 데 사용되는 버퍼의 하드 코딩된 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.

(BZ#1448094)

/etc/nsswitch.conf를 변경하려면 수동 시스템 재부팅이 필요합니다.

authselect select profile_id 명령을 실행하는 경우와 같이 **/etc/nsswitch.conf** 파일에 대한 변경 사항을 적용하려면 모든 관련 프로세스에서 업데이트된 버전의 **/etc/nsswitch.conf** 파일을 사용하도록 시

시스템 재부팅이 필요합니다. 시스템 재부팅이 불가능한 경우 시스템을 **SSSD(System Security Services Daemon)** 또는 **winbind** 에 해당하는 **Active Directory**에 결합하는 서비스를 다시 시작합니다.

(BZ#1657295)

시간 초과 값의 충돌로 인해 **SSSD**가 서버에 연결되지 않음

SSSD(System Security Services Daemon)에서 사용하는 장애 조치 작업과 관련된 기본 시간 초과 값 일부가 충돌합니다. 결과적으로 **SSSD**가 단일 서버와 통신하도록 예약된 시간 초과 값은 전체 시간 초과로 **SSSD**가 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)

SSSD에서 ID 덮어쓰기에서 고유한 인증서만 조회할 수 있습니다.

여러 ID 덮어쓰기에 동일한 인증서가 포함된 경우 **SSSD(System Security Services Daemon)**에서 인증서와 일치하는 사용자에 대한 쿼리를 확인할 수 없습니다. 이러한 사용자를 검색하려고 하면 사용자가 반환되지 않습니다. 사용자 이름 또는 **UID**를 사용하여 사용자를 찾는 작업이 예상대로 작동합니다.

(BZ#1446101)

SSSD는 동일한 우선 순위로 여러 인증서 일치 규칙을 올바르게 처리하지 않습니다.

지정된 인증서가 동일한 우선 순위의 여러 인증서 일치 규칙과 일치하면 **SSSD(System Security Services Daemon)**는 규칙 중 하나만 사용합니다. 해결 방법으로 **LDAP** 필터가 | (또는) 연산자와 연결된 개별 규칙 필터로 구성된 단일 인증서 일치 규칙을 사용하십시오. 일치하는 인증서의 예를 보려면 **sss-certamp(5)** 매뉴얼 페이지를 참조하십시오.

(BZ#1447945)

SSSD에서 로컬 사용자에 대해 잘못된 **LDAP** 그룹 멤버십 반환

SSSD(System Security Services Daemon)가 로컬 파일의 사용자에게 서비스를 제공하는 경우 파일 공급자에 다른 도메인의 그룹 멤버십이 포함되지 않습니다. 결과적으로 로컬 사용자가 **LDAP** 그룹의 멤버인 경우 **id local_user** 명령은 사용자의 **LDAP** 그룹 멤버십을 반환하지 않습니다. 이 문제를 해결하려면 시스템이 **/etc/nsswitch.conf** 파일의 사용자 그룹 멤버십을 찾고 있는 데이터베이스의 순서를 되돌리거나 **sss** 파일을 파일로 교체하거나 **sss**을 추가하여 암시적 파일 도메인을 비활성화하십시오.

enable_files_domain=False

■

`/etc/sss/sss.conf` 파일의 `[sss]` 섹션에 추가합니다.

결과적으로 `id local_user` 는 로컬 사용자에게 대해 올바른 LDAP 그룹 멤버십을 반환합니다.

(BZ#1652562)

`sudo` 규칙에서 그룹 이름을 참조하는 경우 `sudo` 규칙이 `id_provider=ad` 에서 작동하지 않을 수 있습니다.

SSSD(System Security Services Daemon)는 캐시를 사용하여 AD와 SSSD 간의 통신을 최적화하여 `initgroups` 작업 중에 Active Directory 그룹 이름을 확인하지 않습니다. 캐시 항목에는 이름 또는 ID로 그룹을 요청할 때까지 그룹 이름이 아니라 보안 식별자(SID)만 포함됩니다. 따라서 `sudo` 규칙이 `sudo` 를 실행하기 전에 그룹이 완전히 해결되지 않는 한 AD 그룹과 일치하지 않습니다.

이 문제를 해결하려면 최적화를 비활성화해야 합니다. `/etc/sss/sss.conf` 파일을 열고 `[domain/example.com]` 섹션에 `ldap_use_tokengroups = false` 매개 변수를 추가해야 합니다.

(BZ#1659457)

RHEL 8에서는 `systemd-user` 의 기본 PAM 설정이 SSSD 동작에 영향을 미칠 수 있음

Red Hat Enterprise Linux 8에서는 PAM(Pluggable Authentication Module) 스택이 변경되었습니다. 예를 들어 `systemd` 사용자 세션은 이제 `systemd-user` PAM 서비스를 사용하여 PAM 대화식을 시작합니다. 이제 이 서비스에는 `pam_sss.so` 인터페이스가 포함될 수 있는 `system-auth` PAM 서비스가 반복적으로 포함됩니다. 즉, SSSD 액세스 제어를 항상 호출합니다.

RHEL 8 시스템에 대한 액세스 제어 규칙을 설계할 때 변경 사항에 유의하십시오. 예를 들어 `systemd-user` 서비스를 허용된 서비스 목록에 추가할 수 있습니다.

IPA HBAC 또는 ADtektons와 같은 일부 액세스 제어 메커니즘의 경우 `systemd-user` 서비스가 기본적으로 허용되는 서비스 목록에 추가되어 작업을 수행할 필요가 없습니다.

(BZ#1669407)

IdM 서버가 FIPS에서 작동하지 않음

Tomcat용 SSL 커넥터 구현이 불완전하기 때문에 인증서 서버가 설치된 IdM(Identity Management) 서버가 FIPS 모드가 활성화된 시스템에서 작동하지 않습니다.

(BZ#1673296)

sss ID 매핑 플러그인을 사용할 때 Samba에서 액세스를 거부

AD(Active Directory) 도메인에 연결된 RHEL 호스트의 파일 서버로 Samba를 사용하려면 AD에서 사용자 및 그룹을 관리하는 데 SSSD를 사용하는 경우에도 Samba Winbind 서비스를 실행해야 합니다. `realm join --client-software=sss` 명령을 사용하여 도메인에 가입하거나 이 명령에서 `--client-software` 매개변수를 지정하지 않으면 `realm`은 `/etc/sss/sss.conf` 파일만 생성합니다. 이 구성을 사용하여 도메인 멤버에서 Samba를 실행하고 sss ID 매핑 백엔드를 `/etc/requests/year.conf` 파일에 사용하여 구성을 추가하면 ID 매핑 백엔드의 변경으로 인해 오류가 발생할 수 있습니다. 결과적으로 Samba는 사용자 또는 그룹이 존재하고 SSSD에 의해 알려진 특정 경우의 파일에 대한 액세스를 거부합니다.

이전 RHEL 버전에서 업그레이드하려는 경우 `/etc/sss/sss.conf` 파일의 `ldap_id_mapping` 매개변수가 기본값인 `True`로 설정되어 있으면 사용 가능한 해결방법이 없습니다. 이 경우 문제가 해결될 때까지 호스트를 RHEL 8로 업그레이드하지 마십시오.

다른 시나리오에서 가능한 해결 방법:

- 새로 설치하는 경우 영역 `join --client-software=winbind` 명령을 사용하여 도메인에 결합합니다. 그러면 모든 사용자 및 그룹 조회에 SSSD 대신 Winbind를 사용하도록 시스템을 구성합니다. 이 경우 Samba는 `--automatic-id-mapping` 옵션을 `yes` (기본값)로 설정했는지 여부에 따라 `/etc/tekton/tekton.conf`에서 `remove` 또는 `ad` ID 매핑 플러그인을 사용합니다. 향후 또는 다른 시스템에서 SSSD를 사용하려는 경우 `--automatic-id-mapping=no`를 사용하면 더 쉽게 마이그레이션할 수 있지만 모든 사용자 및 그룹에 대해 AD에 POSIX UID 및 GID를 저장해야 합니다.

- 이전 RHEL 버전에서 업그레이드하는 경우 및 `/etc/sss/sss.conf` 파일의 `ldap_id_mapping` 매개변수가 `False`로 설정되어 있고 시스템은 ID 매핑을 위해 AD에서 `uidNumber` 및 `gidNumber` 속성을 사용합니다.

1. `idmap config <domain> : backend = ssss` 항목을 `idmap config <domain> : backend = ad`으로 `idmap config <domain> : backend = ad`로 변경
2. `systemctl status winbind` 명령을 사용하여 Winbind를 다시 시작합니다.

(BZ#1657665)

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

nuxwdog watchdog 서비스가 인증서 시스템에 통합되었습니다. 결과적으로 **nuxwdog** 은 더 이상 별도의 패키지로 제공되지 않습니다. **watchdog** 서비스를 사용하려면 **pki-server** 패키지를 설치합니다.

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 8.0**은 **keyutils**를 패키지 종속성으로 자동 설치하지 않습니다. 패키지를 수동으로 설치하려면 **dnf install keyutils** 명령을 사용하십시오.

([BZ#1652269](#))

AD 사용자의 **ID** 덮어쓰기 추가는 **IdM CLI**에서만 작동합니다.

현재는 **IdM** 웹 UI에서 관리 역할에 대한 액세스 권한을 부여하기 위해 **IdM(Identity Management)** 그룹에 **AD(Active Directory)** 사용자의 **ID** 덮어쓰기를 추가합니다. 문제를 해결하려면 **IdM CLI**(명령줄 인터페이스)를 대신 사용하십시오.

ipa 유틸리티를 사용하여 특정 작업을 수행한 후 **IdM** 서버에 **ipa-idoverride-memberof-plugin** 패키지를 설치한 경우 **ipa** 유틸리티 캐시를 정리하여 **IdM** 서버 메타데이터에 대한 보기를 새로 고치는 것이 좋습니다.

이렇게 하려면 **ipa** 유틸리티가 실행되는 사용자의 **~/.cache/ipa** 디렉터리의 콘텐츠를 제거하십시오. 예를 들어 **root**의 경우:

```
# rm -r /root/.cache/ipa
```

([BZ#1651577](#))

IdM에서 **AD** 신뢰 지원을 활성화할 때 표시되는 필수 **DNS** 레코드에 대한 정보는 없습니다.

외부 **DNS** 관리를 사용하여 **Red Hat Enterprise Linux IdM(Identity Management)** 설치에 **AD(Active Directory)** 신뢰 지원을 활성화하면 필수 **DNS** 레코드에 대한 정보가 표시되지 않습니다. 필요한 **DNS** 레코드가 추가될 때까지 **AD**에 대한 **domain** 신뢰는 성공하지 못합니다. 이 문제를 해결하려면

'ipa dns-update-system-records --dry-run' 명령을 실행하여 IdM에 필요한 모든 DNS 레코드 목록을 가져옵니다. IdM 도메인의 외부 DNS에서 필요한 DNS 레코드를 정의하는 경우, AD로 forest 트러스트를 설정할 수 있습니다.

(BZ#1665051)

5.5.12. 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** 틀은 **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)

5.5.13. 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)

VDO 볼륨은 다른 엔드안 플랫폼으로 전환 한 후 중복 제거 조언을 잃게 됩니다.

VDO(가상 데이터 최적화 도구)는 **UDS(Universal Deduplication Service)** 인덱스 헤더를 플랫폼 네이티브 **endian** 형식으로 씁니다. **VDO**는 **UDS** 인덱스가 손상된 것을 고려하고 **VDO** 볼륨을 다른 **endian**을 사용하는 플랫폼으로 이동하면 새로운 빈 인덱스로 덮어씁니다.

결과적으로 덮어쓰기 전에 **UDS** 인덱스에 저장된 중복 제거 조언을 잃게 됩니다. 그런 다음 **VDO**는 볼륨을 이동하기 전에 저장된 데이터에 대해 새로 작성된 데이터를 중복시킬 수 없으므로 공간 절약을 줄일 수 없습니다.

(BZ#1696492)

XFS DAX 마운트 옵션이 공유 **COW(Copy-On-Write)** 데이터 **Extent**와 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.

자세한 내용은 [NVDIMM에서 파일 시스템 DAX 네임스페이스 생성](#)을 참조하십시오.

(BZ#1620330)

특정 SCSI 드라이버는 때때로 과도한 양의 메모리를 사용할 수 있습니다.

특정 SCSI 드라이버는 RHEL 7에서보다 많은 양의 메모리를 사용합니다. HBA(Fibre Channel host bus adapter)에서 vPort 생성과 같은 특정 경우에는 시스템 구성에 따라 메모리 사용량이 과도할 수 있습니다.

증가된 메모리 사용량은 블록 계층의 메모리 사전 할당으로 인해 발생합니다. RHEL 8의 각 I/O 요청에 대해 다중 대기열 블록 장치 스케줄링(BLK-MQ) 및 다중 대기열 SCSI 스택(SCSI-MQ)을 미리 할당하면 메모리 사용량이 증가합니다.

(BZ#1733278)

5.5.14. 네트워킹

nftables이 다차원 IP 집합 유형을 지원하지 않음

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

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

(BZ#1593711)

iptables-extensions(8) 도움말 페이지의 TRACE 대상은 nf_tables 변형을 참조하지 않습니다.

iptables-extensions(8) 도움말 페이지의 **TRACE** 대상에 대한 설명은 **compat** 변형만 참조하지만 **RHEL(Red Hat Enterprise Linux) 8.0**은 **nf_tables** 변형을 사용합니다. **RHEL**의 **nftables** 기반 **iptables** 유틸리티는 내부적으로 **meta nfttrace** 표현식을 사용합니다. 따라서 커널은 커널 로그에 **TRACE** 이벤트를 출력하지 않지만 사용자 공간으로 보냅니다. 그러나 도움말 페이지는 이러한 이벤트를 표시하는 **xtables-monitor** 명령줄 유틸리티를 참조하지 않습니다.

([BZ#1658734](#))

RHEL 8은 스위치를 장기간 사용할 수 없는 후 **802.3ad** 본딩의 상태를 **"Churned"**로 표시합니다.

현재 **802.3ad** 네트워크 본딩을 구성하고 장기간 전환이 중단된 경우 **Red Hat Enterprise Linux**는 연결이 작동 상태로 복귀한 후에도 **"Churned"**로 본딩 상태를 올바르게 표시합니다. 그러나 **"Churned"** 상태는 상당한 링크 중단이 발생했음을 관리자에게 알리는 것을 목표로 하므로 이는 의도한 동작입니다. 이 상태를 지우려면 네트워크 본딩을 다시 시작하거나 호스트를 재부팅합니다.

([BZ#1708807](#))

ebtables 명령은 **broute** 테이블을 지원하지 않습니다.

Red Hat Enterprise Linux 8.0의 **nftables** 기반 **ebtables** 명령은 **broute** 테이블을 지원하지 않습니다. 따라서 사용자는 이 기능을 사용할 수 없습니다.

([BZ#1649790](#))

GRO가 비활성화되면 **IPsec** 오프로드 중에 **IPsec** 네트워크 트래픽이 실패합니다.

IPsec 오프로드는 장치에서 **Generic Receive Offload (GRO)**가 비활성화되면 작동하지 않을 수 있습니다. **IPsec** 오프로드가 네트워크 인터페이스에 구성되어 해당 장치에서 **GRO**가 비활성화되어 있는 경우 **IPsec** 네트워크 트래픽이 실패합니다.

이 문제를 해결하려면 장치에서 **GRO**를 사용하도록 설정해야 합니다.

([BZ#1649647](#))

NetworkManager 는 기본적으로 내부 **DHCP** 플러그인 사용

NetworkManager 는 내부 및 **dhclient** **DHCP** 플러그인을 지원합니다. 기본적으로 **RHEL(Red Hat Enterprise Linux) 7**의 **NetworkManager** 는 **dhclient** 및 **RHEL 8** 내부 플러그인을 사용합니다. 특정 상

항에서는 플러그인이 다르게 작동합니다. 예를 들어 **dhclient** 는 **/etc/dhcp/** 디렉터리에 지정된 추가 설정을 사용할 수 있습니다.

RHEL 7에서 **RHEL 8**으로 업그레이드하고 **NetworkManager** 가 다른 경우 **/etc/NetworkManager/NetworkManager.conf** 파일의 **[main]** 섹션에 다음 설정을 추가하여 **dhclient** 플러그인을 사용합니다.

```
[main]
dhcp=dhclient
```

(BZ#1571655)

IPsec 기반 **VPN**의 고급 옵션은 **gnome-control-center**를 사용하여 변경할 수 없습니다.

gnome-control-center 애플리케이션을 사용하여 **IPsec** 기반 **VPN** 연결을 구성할 때 고급 대화 상자에는 설정만 표시되지만 변경할 수는 없습니다. 결과적으로 사용자는 고급 **IPsec** 옵션을 변경할 수 없습니다. 이 문제를 해결하려면 **nm-connection-editor** 또는 **nmcli** 툴을 사용하여 고급 속성 구성을 수행합니다.

(BZ#1697326)

/etc/hosts.allow 및 **/etc/hosts.deny** 파일에는 부정확한 정보가 포함되어 있습니다.

tcp_wrappers 패키지는 **RHEL (Red Hat Enterprise Linux) 8**에서 제거되지만 파일 **/etc/hosts.allow** 및 **/etc/hosts.deny**는 제거되지 않습니다. 결과적으로 이러한 파일에는 **RHEL 8**에는 적용되지 않는 오래된 정보가 포함되어 있습니다.

이 문제를 해결하려면 서비스에 대한 액세스를 필터링하는 방화벽 규칙을 사용합니다. 사용자 이름 및 호스트 이름을 기반으로 필터링하려면 애플리케이션별 구성을 사용합니다.

(BZ#1663556)

IP 조각 모음은 네트워크 트래픽 과부하 하에서 지속 가능한 일 수 없습니다.

Red Hat Enterprise Linux 8에서 가비지 컬렉션 커널 스레드가 제거되었으며 **IP** 조각은 시간 초과 시에만 만료됩니다. 그 결과 **DoS(Denial of Service)** 아래의 **CPU** 사용량이 훨씬 낮으며, 가능한 최대 조각율은 **IP** 리셀링 단위에 대해 구성된 메모리 양에 따라 제한됩니다. 패킷 드롭이 있을 때 조각화된 트래픽이 필요한 기본 설정 워크로드에서는 패킷 재주문 또는 많은 동시 분할 흐름이 관련 성능 회귀가 발생할 수 있습니다.

이 경우 사용자는 `/proc/sys/net/ipv4` 디렉터리에서 **IP** 조각화 캐시의 적절한 튜닝을 사용하여 `ipfrag_high_thresh` 변수를 설정하여 메모리 양을 제한하고 `ipfrag_time` 변수를 사용하여 메모리에 **IP** 조각을 유지할 수 있습니다. 예를 들면 다음과 같습니다.

```
echo 419430400 > /proc/sys/net/ipv4/ipfrag_high_thresh echo 1 >
/proc/sys/net/ipv4/ipfrag_time
```

위의 내용은 **IPv4** 트래픽에 적용됩니다. **IPv6**의 경우 관련 튜닝 가능 항목은 `/proc/sys/net/ipv6/` 디렉터리에 있는 `ip6frag_thresh` 및 `ip6frag_time`입니다.

고속 조각된 트래픽에 의존하는 모든 워크로드는 특히 패킷 드롭으로 안정성 및 성능 문제를 일으킬 수 있으며 이러한 종류의 배포는 프로덕션에서 좋지 않습니다.

(BZ#1597671)

RHEL 8에서 네트워크 인터페이스 이름이 변경됨

Red Hat Enterprise Linux 8에서는 기본적으로 **RHEL 7**과 동일한 일관된 네트워크 장치 이름 지정 스키마를 사용합니다. 그러나 `e1000e`, `nfp`, `qede`, `sfc`, `tg3` 및 `bnxt_en` 과 같은 일부 커널 드라이버는 **RHEL 8**의 새로운 설치에서 일관된 이름을 변경했습니다. 그러나 이름은 **RHEL 7**에서 업그레이드하는 데 유지됩니다.

(BZ#1701968)

5.5.15. 보안

libselinux-python 은 해당 모듈을 통해서만 사용 가능

libselinux-python 패키지에는 **SELinux** 애플리케이션 개발을 위한 **Python 2** 바인딩만 포함되어 있으며 이전 버전과의 호환성을 위해 사용됩니다. 이러한 이유로 `dnf install libselinux-python` 명령을 통해 기본 **RHEL 8** 리포지토리에서 더 이상 **libselinux-python**을 사용할 수 없습니다.

이 문제를 해결하려면 **libselinux-python** 및 **python27** 모듈을 모두 활성화하고 다음 명령을 사용하여 **libselinux-python** 패키지 및 해당 종속 항목을 설치합니다.

```
# dnf module enable libselinux-python
# dnf install libselinux-python
```

또는 단일 명령으로 **install** 프로필을 사용하여 **libselinux-python** 을 설치합니다.

```
# dnf module install libselinux-python:2.8/common
```

결과적으로 해당 모듈을 사용하여 **libselinux-python** 을 설치할 수 있습니다.

(BZ#1666328)

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)

특정 **rsyslog** 우선순위 문자열이 올바르게 작동하지 않음

암호화를 세부적으로 제어할 수 있는 **imtcp**의 **GnuTLS** 우선순위 문자열 지원은 완료되지 않습니다. 결과적으로 다음 우선순위 문자열이 **rsyslog**에서 제대로 작동하지 않습니다.

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+DHE-RSA:+AES-256-GCM:+SIGN-RSA-SHA384:+COMP-ALL:+GROUP-ALL
```

이 문제를 해결하려면 올바른 작동 우선 순위 문자열만 사용하십시오.

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+ECDHE-RSA:+AES-128-CBC:+SIGN-RSA-SHA1:+COMP-ALL:+GROUP-ALL
```

따라서 현재 구성이 올바르게 작동하는 문자열로 제한되어야 합니다.

(BZ#1679512)

성능에 대한 기본 로깅 설정의 부정적인 영향

기본 로깅 환경 설정에서는 **4GB**의 메모리를 사용하거나 **rsyslog** 를 사용하여 **systemd-journald** 를 실행할 때 **rate-limit** 값의 조정이 복잡할 수 있습니다.

자세한 내용은 [RHEL 기본 로깅 설정의 성능 및 완화](#) 기술 자료 문서를 참조하십시오.

(JIRA:RHELPLAN-10431)

OpenSCAP rpmverifypackage가 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)

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)

Kickstart는 RHEL 8에서 **com_redhat_oscap** 대신 **org_fedora_oscap** 을 사용합니다.

Kickstart는 **OSCAP**(Open Security Content Automation Protocol) **Anaconda** 애드온을 **com_redhat_oscap** 대신 **org_fedora_oscap** 으로 참조하여 혼동을 일으킬 수 있습니다. 이는 **Red Hat Enterprise Linux 7**과의 이전 버전과의 호환성을 유지하기 위해 수행됩니다.

(BZ#1665082)

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_test가 사용되면 실패합니다. 따라서 설명된 시나리오에서 `oscap-chroot`이 중단됩니다.

(BZ#1636431)

OpenSCAP 에서 가상 머신 및 컨테이너의 오프라인 스캔을 제공하지 않음

OpenSCAP 코드베이스를 리팩토링하면 특정 RPM 프로브에서 VM 및 컨테이너 파일 시스템을 오프라인 모드로 스캔하지 못했습니다. 따라서 다음 도구가 openscap-utils 패키지에서 제거되었습니다. `oscap-vm` 및 `oscap-chroot`. 또한 `openscap-containers` 패키지가 완전히 제거되었습니다.

(BZ#1618489)

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)

OpenSSL TLS 라이브러리는 PKCS#11 토큰이 원시 RSA 또는 RSA -PSS 서명 생성을 지원하는지 감지하지 않습니다.

TLS-1.3 프로토콜에서는 RSA-PSS 서명을 지원해야 합니다. PKCS#11 토큰이 raw RSA 또는 RSA-PSS 서명을 지원하지 않는 경우, OpenSSL TLS 라이브러리를 사용하는 서버 애플리케이션은 PKCS#11 토큰에 의해 보유되면 RSA 키로 작동하지 않습니다. 결과적으로 TLS 통신이 실패합니다.

이 문제를 해결하려면 TLS-1.2 버전을 사용 가능한 최고 TLS 프로토콜 버전으로 사용하도록 서버 또는 클라이언트를 구성합니다.

(BZ#1681178)

PKCS#11 장치와 **RSA-PSS** 인증서에 저장된 **RSA** 개인 키를 사용하는 경우 **Apache httpd** 가 시작되지 않습니다.

PKCS#11 표준은 **RSA**와 **RSA-PSS** 키 오브젝트를 구분하지 않으며 두 가지 모두에 대해 **CKK_RSA** 유형을 사용합니다. 그러나 **OpenSSL**은 **RSA** 및 **RSA-PSS** 키에 대해 다른 유형을 사용합니다. 결과적으로 **openssl-pkcs11** 엔진은 **PKCS#11 RSA** 키 오브젝트의 **OpenSSL**에 제공해야 하는 유형을 결정할 수 없습니다. 현재 엔진은 모든 **PKCS#11 CKK_RSA** 오브젝트에 대한 키 유형을 **RSA** 키로 설정합니다. **OpenSSL**이 인증서에서 얻은 **RSA-PSS** 공개 키 유형을 엔진에서 제공하는 **RSA** 개인 키 오브젝트에 포함된 유형과 비교하면 유형이 다른 것으로 간주합니다. 따라서 인증서와 개인 키가 일치하지 않습니다. **X509_check_private_key()** 함수에서 수행한 검사에서 이 시나리오에서 오류를 반환합니다. **httpd** 웹 서버는 시작 프로세스에서 이 함수를 호출하여 제공된 인증서와 키가 일치하는지 확인합니다. 이 검사는 **RSA-PSS** 공개 키와 **PKCS#11** 모듈에 저장된 **RSA** 개인 키가 포함된 인증서에 항상 실패하므로 **httpd** 는 이 설정 사용을 시작하지 못합니다. 이 문제에 대한 해결방법은 없습니다.

(BZ#1664802)

PKCS#11 장치에 저장된 해당 공개 키가 없는 **ECDSA** 개인 키를 사용하는 경우 **httpd** 가 시작되지 않습니다.

RSA 키와 달리 **ECDSA** 개인 키에는 공개 키 정보가 반드시 포함되어 있지 않습니다. 이 경우 **ECDSA** 개인 키에서 공개 키를 가져올 수 없습니다. 이러한 이유로 **PKCS#11** 장치는 공개 키 오브젝트인지 인증서 오브젝트와 관계없이 별도의 오브젝트에 공개 키 정보를 저장합니다. **OpenSSL**은 개인 키의 공개 키 정보를 포함할 수 있도록 엔진에서 제공하는 **rfcP_PKEY** 구조를 예상합니다. **openssl-pkcs11** 패키지의 엔진에서 공개 키 오브젝트만 가져오고 현재 인증서 오브젝트를 무시합니다.

OpenSSL이 엔진에서 **ECDSA** 개인 키를 요청하는 경우, 공개 키가 포함된 일치하는 인증서를 사용할 수 있더라도 공용 키가 **PKCS#11** 장치에 없는 경우 제공된 **steps P_PKEY** 구조에 공개 키 정보가 포함되지 않습니다. 그 결과 **Apache httpd** 웹 서버는 공용 키가 필요한 **X509_check_private_key()** 함수를 호출하므로 시작 프로세스에서 **httpd** 가 이 시나리오에서 시작되지 않습니다. 이 문제를 해결하려면 **ECDSA** 키를 사용할 때 개인 키와 공개 키를 **PKCS#11** 장치에 저장하십시오. 결과적으로 **ECDSA** 키가 **PKCS#11** 장치에 저장되면 **httpd** 가 올바르게 시작됩니다.

(BZ#1664807)

OpenSSH는 라벨이 일치하지 않는 키의 **PKCS #11 URI**를 올바르게 처리하지 않습니다.

OpenSSH 제품군은 라벨을 통해 키 쌍을 식별할 수 있습니다. 라벨은 스마트 카드에 저장된 개인 키와 공개 키에 따라 다를 수 있습니다. 그 결과 오브젝트 부분(키 레이블)을 사용하여 **PKCS #11 URI**를 지정하면 **OpenSSH**가 **PKCS #11**에서 적절한 오브젝트를 찾을 수 없습니다.

이 문제를 해결하려면 오브젝트 부분 없이 **PKCS #11 URI**를 지정합니다. 결과적으로 **OpenSSH**는 **PKCS #11 URI**를 사용하여 참조되는 스마트 카드에서 키를 사용할 수 있습니다.

(BZ#1671262)

iptables-ebtables의 출력은 **ebtables**와 100% 호환되지 않습니다.

RHEL 8에서는 **ebtables** 명령은 툴의 **nftables** 기반 재구축이 포함된 **iptables-ebtables** 패키지에서 제공됩니다. 이 툴에는 다른 코드 베이스가 있으며 출력은 부정하거나 고의적 설계 옵션 중 하나 인 측면에서 분리됩니다.

결과적으로 일부 **ebtables** 출력을 구문 분석하는 스크립트를 마이그레이션할 때 다음이 반영되도록 스크립트를 조정합니다.

- **MAC** 주소 형식이 길이가 수정되었습니다. 필요한 경우 개별 바이트 값의 앞에 **0**을 포함시켜서 이 두 문자의 형식을 **character 1**개씩 유지 관리합니다.
- **RFC 4291**을 준수하도록 **IPv6** 접두사 형식이 변경되었습니다. 슬래시 문자 뒤에 있는 후행 부분에는 더 이상 **IPv6** 주소 형식의 넷마스크는 포함되지 않고 접두사 길이가 포함됩니다. 이 변경은 유효한(**left-contiguous**) 마스크에만 적용되지만 다른 항목은 이전 형식으로 인쇄됩니다.

(BZ#1674536)

curve25519-sha256은 **OpenSSH**에서 기본적으로 지원되지 않습니다.

curve25519-sha256 SSH 키 교환 알고리즘은 기본 정책 수준을 준수하더라도 **OpenSSH** 클라이언트 및 서버의 시스템 전체 암호화 정책 구성에서 누락됩니다. 결과적으로 클라이언트 또는 서버에서 **curve25519-sha256**을 사용하고 호스트에서 이 알고리즘을 지원하지 않는 경우 연결이 실패할 수 있습니다.

이 문제를 해결하려면 **openssh.config** 및 **opensshserver.config** 파일의 **openssh.config** 및 **opensshserver.config** 파일을 **OpenSSH** 클라이언트 및 서버의 **/etc/crypto-policies/back-ends/** 디렉터리에서 수정하여 시스템 차원의 암호화 정책 구성을 수동으로 재정의할 수 있습니다. 이 구성은 시스템 전체 암호화 정책의 모든 변경에 대해 덮어씁니다. 자세한 내용은 **update-crypto-policies(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1678661)

OpenSSL 은 원시 RSA 또는 RSA-PSS 서명을 지원하지 않는 PKCS #11 토큰을 잘못 처리합니다.

OpenSSL 라이브러리는 PKCS #11 토큰의 키 관련 기능을 감지하지 않습니다. 그 결과 원시 RSA 또는 RSA-PSS 서명을 지원하지 않는 토큰을 사용하여 서명이 생성될 때 TLS 연결을 설정하는 데 실패합니다.

이 문제를 해결하려면 `/etc/pki/tls/openssl.cnf` 파일의 `crypto_policy` 섹션 끝에 `.include` 행 뒤에 다음 행을 추가합니다.

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

결과적으로 설명된 시나리오에서 TLS 연결을 설정할 수 있습니다.

([BZ#1685470](#))

VMware 호스트 시스템과의 SSH 연결이 작동하지 않음

OpenSSH 제품군의 현재 버전에는 SSH 패킷의 기본 IP 품질(IPQoS) 플래그가 변경되어 VMware 가상화 플랫폼에서 올바르게 처리되지 않습니다. 따라서 VMware의 시스템과의 SSH 연결을 설정할 수 없습니다.

이 문제를 해결하려면 `ssh_config` 파일에 `IPQoS=throughput` 를 포함합니다. 이로 인해 VMware 호스팅 시스템의 SSH 연결이 올바르게 작동합니다.

자세한 내용은 [SSH를 통해 SSH를 통해 다른 호스트 기술 솔루션 문서에 연결할 수 없는 VMWare Workstation에서 실행 중인 RHEL 8](#) 을 참조하십시오.

([BZ#1651763](#))

5.5.16. 서브스크립션 관리

서비스 수준의 성공적인 설정 및 설정 해제에 대한 메시지가 인쇄되지 않음

Curve pin 서비스에 'syspurpose' 기능이 없는 경우 서브스크립션 관리자는 다른 코드 경로를 사용하여 서비스 수준 인수를 설정합니다. 이 코드 경로는 작업 결과를 출력하지 않습니다. 결과적으로 서브스크

립션 관리자가 서비스 수준을 설정할 때 메시지가 표시되지 않습니다. 이는 서비스 수준 세트에 오타가 있거나 실제로 사용할 수 없는 경우 특히 문제가 됩니다.

(BZ#1661414)

syspurpose addons 는 **subscription-manager attach --auto** 출력에 영향을 미치지 않습니다.

Red Hat Enterprise Linux 8에서 **syspurpose** 명령줄 톨의 네 가지 속성(역할,사용법,**service_level_agreement** 및 **addons**)이 추가되었습니다. 현재는 **role,usage** 및 **service_level_agreement** 만 **subscription-manager attach --auto** 명령의 출력에 영향을 미칩니다. **addons** 인수로 값을 설정하려는 사용자는 자동으로 연결된 서브스크립션에 영향을 미치지 않습니다.

(BZ#1687900)

5.5.17. 가상화

cloud-init 및 복제된 부팅이 매우 느리게 설정된 **ESXi** 가상 머신

현재 **cloud-init** 서비스를 사용하여 **VMware ESXi** 하이퍼바이저에서 실행되는 **VM**(가상 시스템)을 수정하여 고정 **IP**를 사용하고 **VM**이 복제되는 경우 일부 경우에 새로 복제된 **VM**이 재부팅되는 데 시간이 매우 오래 걸립니다. 이로 인해 **cloud-init** 에서 **VM**의 고정 **IP**를 **DHCP**에 다시 작성한 다음 사용 가능한 데이터 소스를 검색합니다.

이 문제를 해결하려면 **VM**이 처음 부팅된 후 **cloud-init** 를 제거할 수 있습니다. 결과적으로 후속 재부팅이 느려지지 않습니다.

(BZ#1666961, BZ#1706482)

중첩된 가상화 블록 활성화 실시간 마이그레이션

현재 중첩된 가상화 기능은 실시간 마이그레이션과 호환되지 않습니다. 따라서 **RHEL 8** 호스트에서 중첩 가상화를 활성화하면 **VM**(가상 머신)을 호스트에서 마이그레이션하고 **VM** 상태 스냅샷을 디스크에 절약할 수 있습니다.

중첩된 가상화는 현재 **RHEL 8**에서 기술 프리뷰로 제공되므로 지원되지 않습니다. 또한 중첩된 가상화는 기본적으로 비활성화되어 있습니다. 해당 모듈을 활성화하려면 **kvm_intel.nested** 또는 **kvm_amd.nested** 모듈 매개 변수를 사용합니다.

(BZ#1689216)**cloud-init** 를 사용하여 **Microsoft Azure**에서 가상 머신 프로비저닝 실패

현재 **cloud-init** 유틸리티를 사용하여 **Microsoft Azure** 플랫폼에서 **RHEL 8** 가상 머신(VM)을 프로비저닝할 수 없습니다. 이 문제를 해결하려면 다음 방법 중 하나를 사용하십시오.

- **cloud-init** 대신 **WALinuxAgent** 패키지를 사용하여 **Microsoft Azure**에 VM을 프로비저닝합니다.
- **/etc/NetworkManager/NetworkManager.conf** 파일의 **[main]** 섹션에 다음 설정을 추가합니다.

```
[main]
dhcp=dhclient
```

(BZ#1641190)

2세대 **RHEL 8** 가상 머신은 **Hyper-V Server 2016** 호스트에서 부팅할 수 없는 경우가 있습니다.

Microsoft Hyper-V Server 2016 호스트에서 실행 중인 가상 머신(VM)에서 **RHEL 8**을 게스트 운영 체제로 사용하는 경우 경우에 따라 VM을 부팅하지 못하고 **GRUB** 부팅 메뉴로 반환되지 않습니다. 또한 다음 오류는 **Hyper-V** 이벤트 로그에 기록됩니다. In addition, the following error is logged in the **Hyper-V** event log:

The guest operating system reported that it failed with the following error code: 0x1E

이 오류는 **Hyper-V** 호스트의 **UEFI** 펌웨어 버그로 인해 발생합니다. 이 문제를 해결하려면 **Hyper-V Server 2019**를 호스트로 사용하십시오.

(BZ#1583445)**virsh iface-*** 명령이 일관되게 작동하지 않음

현재는 **virsh iface-start** 및 **virsh iface-destroy** 와 같은 **virsh iface-*** 명령이 구성 종속성으로 인해 자주 실패합니다. 따라서 호스트 네트워크 연결을 구성하고 관리하기 위해 **virsh iface-*** 명령을 사용하지 않는 것이 좋습니다. 대신 **NetworkManager** 프로그램과 관련 관리 애플리케이션을 사용합니다.

(BZ#1664592)

Azure용 Linux 가상 머신 확장이 제대로 작동하지 않는 경우

RHEL 8에는 기본적으로 **python2** 패키지가 포함되어 있지 않습니다. 따라서 경우에 따라 RHEL 8 VM에서 **azure-linux-extensions** 라고도 하는 Azure용 Linux 가상 머신 확장을 실행하면 실패합니다.

azure-linux-extensions 가 예상대로 작동할 확률을 늘리려면 RHEL 8 VM에 수동으로 **python2** 를 설치합니다.

```
# yum install python2
```

(BZ#1561132)

5.5.18. 지원 관련 기능

redhat-support-tool 은 공개 사례에서 **sosreport** 를 자동으로 수집하지 않습니다.

redhat-support-tool 명령은 **sosreport** 아카이브를 생성할 수 없습니다. 이 문제를 해결하려면 **sosreport** 명령을 별도로 실행한 다음 **redhat-support-tool addattachment -c** 명령을 입력하여 아카이브를 업로드하거나 고객 포털에서 웹 UI를 사용합니다. 결과적으로 케이스가 생성되고 **sosreport** 가 업로드됩니다.

findkerneldebugs,btextract,analyze diagnose 명령이 예상대로 작동하지 않으며 향후 릴리스에서 수정될 예정입니다.

(BZ#1688274)

6장. 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 툴과 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. **Pod**라고 하는 컨테이너 및 컨테이너 그룹을 관리할 수 있는 **libpod** 라이브러리를 기반으로 합니다.

podman 을 사용하는 방법에 대한 자세한 내용은 [Building, running, and managing containers](#) 를 참조하십시오.
- RHEL 8 GA에서는 **Red Hat Universal Base Images (UBI)**가 새롭게 제공됩니다. **UBI**는 표준 및 최소 RHEL 기반 이미지와 같이 **Red Hat**에서 이전에 제공한 이미지의 일부를 교체합니다.

이전 **Red Hat** 이미지와 달리 **UBI**는 자유롭게 재배포가 가능합니다. 즉, 모든 환경에서 사용하고 어디에서나 공유할 수 있습니다. **Red Hat** 고객이 아니어도 사용할 수 있습니다.

UBI 문서의 경우 [컨테이너 빌드, 실행 및 관리](#)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. **sos** 및 **redhat-support-tool** 툴은 **support-tools** 컨테

이너에 제공됩니다. 시스템 관리자는 이 이미지를 시스템 도구 컨테이너 이미지를 빌드하기 위한 기반으로 사용할 수도 있습니다.

- 루트리스(**rootless**) 컨테이너에 대한 지원은 **RHEL 8**에서 기술 프리뷰로 사용할 수 있습니다.

루트리스(**rootless**) 컨테이너는 관리자 권한 없이 일반 시스템 사용자가 생성 및 관리하는 컨테이너입니다.

7장. 국제화

7.1. RED HAT ENTERPRISE LINUX 8 INTERNATIONAL LANGUAGES

Red Hat Enterprise Linux 8은 여러 언어를 설치하고 요구 사항에 따라 언어 변경을 지원합니다.

- 동아시아 언어 - 일본어, 한국어, 중국어 간체, 중국어 번체 중국어.
- 유럽 언어 - 영어, 독일어, 스페인어, 프랑스어, 이탈리아어, 포르투갈어 및 러시아어.

다음 표에는 다양한 주요 언어에 대해 제공되는 글꼴 및 입력 방법이 나열되어 있습니다.

언어	기본 글꼴 (Font Package)	입력 방법
영어	dejavu-sans-fonts	
프랑스어	dejavu-sans-fonts	
독일어	dejavu-sans-fonts	
이탈리아어	dejavu-sans-fonts	
러시아어	dejavu-sans-fonts	
스페인어	dejavu-sans-fonts	
포르투갈어	dejavu-sans-fonts	
중국어 단순화	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libpinyin, libpinyin
기존 중국어	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libzhuyin, libzhuyin
일본어	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-kkc, libkkc
한국어	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	IBus-hangul, libhangu

7.2. RHEL 8 국제화에 대한 주요 변경 사항

RHEL 8에서는 RHEL 7에 비해 국제화에 다음과 같은 변경 사항이 추가되었습니다.

- 유니코드 11 컴퓨팅 업계 표준에 대한 지원이 추가되었습니다.
- 국제화는 여러 패키지로 배포되므로 공간 절약 설치를 가능하게 합니다.

자세한 내용은 [glibc Localization for RHEL is distributed in multiple packages](#) 를 참조하십시오.

- 여러 로케일에 대한 **glibc** 패키지 업데이트가 이제 **CLDR(Common Locale Data Repository)**와 동기화됩니다.

부록 A. 구성 요소별 티켓 목록

구성 요소	티켓
389-ds-base	BZ#1334254, BZ#1358706
NetworkManager	BZ#1555013, BZ#1555012, BZ#1557035, BZ#1335409, BZ#1571655
PackageKit	BZ#1559414
WALinuxAgent	BZ#1561132
anaconda	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904, BZ#1672405 , JIRA:RHELPLAN-1943, BZ#1677411, BZ#1696609
audit	BZ#1616428
authselect	BZ#1657295
bcc	BZ#1548302
bind	BZ#1588592
boom-boot	BZ#1649582
boost	BZ#1494495, BZ#1616244
cloud-init	BZ#1615599, BZ#1641190
cmake	BZ#1590139
cockpit	BZ#1619993, BZ#1631905
criu	BZ#1689746
crypto-policies	BZ#1591620, BZ#1645606, BZ#1678661 , BZ#1660839
cryptsetup	BZ#1564540
device-mapper-multipath	BZ#1643550, BZ#1673167
distribution	BZ#1516728, BZ#1516741, BZ#1566048
dnf	BZ#1622580, BZ#1647760, BZ#1581191
driverctl	BZ#1648411

구성 요소	티켓
edk2	BZ#1536627
esc	BZ#1538645
firewalld	BZ#1509026, BZ#1648497
gcc	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1571124, BZ#1246444, JIRA:RHELPLAN-7437, BZ#1652016
gdb	BZ#1491128
gdm	BZ#1589678, BZ#1641763, BZ#1678627
glib-networking	BZ#1640534
glibc	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
gnome-control-center	BZ#1697326
go-toolset-1.10-golang	BZ#1633351
grub2	BZ#1583445
httpd	BZ#1633224, BZ#1632754
ipa-idoverride-memberof	BZ#1651577
ipa	BZ#1664718 , BZ#1664719 , BZ#1665051
iproute	BZ#1640991, BZ#1589317
iptables	BZ#1644030, BZ#1564596, BZ#1646159, BZ#1658734 , BZ#1649790, BZ#1674536
iscsi-initiator-utils	BZ#1626629, BZ#1582099
kernel-rt	BZ#1592977

구성 요소	티켓
kernel	BZ#1598448, BZ#1559607, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1562987, BZ#1273139, BZ#1273139, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1633143, BZ#1596240, BZ#1534870, BZ#1153521, BZ#1515987, BZ#1642795, BZ#1570255, BZ#1645744, BZ#1440031, BZ#1649647 BZ#1494705, BZ#1650149, BZ#1655413, BZ#1651806, BZ#1620330, BZ#1665295, BZ#1505999, BZ#1645433, BZ#1663281, BZ#1627455, BZ#1627455, BZ#1581898, BZ#1597671, BZ#1557671, BZ#15596 BZ#1658391, BZ#1623590, BZ#1614144, BZ#1519039, BZ#1524683, BZ#1694705
kexec-tools	BZ#1520209, BZ#1662911
kmod-kvdo	BZ#1534087, BZ#1639512, BZ#1696492
ksh	BZ#1503922
libdnf	BZ#1642458, BZ#1679509
libreswan	BZ#1566574, BZ#1648776, BZ#1657854
libssh	BZ#1485241
libvirt	BZ#1528684
lksctp-tools	BZ#1568622
ltrace	BZ#1618748, BZ#1584322
lvm2	BZ#1676598 , BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
mariadb	BZ#1637034
mdadm	BZ#1654482
controler	BZ#1668883
net-snmp	BZ#1584510
nfs-utils	BZ#1592011, BZ#1639432
nftables	BZ#1593711
nginx	BZ#1545526

구성 요소	티켓
nodejs-10-module	BZ#1622118
nss	BZ#1489094, BZ#1645153
nuxwdog	BZ#1652269
openldap	BZ#1570056
opensc	BZ#1595638, BZ#1595626
openscap	BZ#1614273, BZ#1618484, BZ#1646197, BZ#1636431, BZ#1618489, BZ#1642373, BZ#1618464
openssh	BZ#1622511, BZ#1228088, BZ#1645038, BZ#1671262, BZ#1651763
openssl-pkcs11	BZ#1664802 , BZ#1664807
OpenSSL	BZ#1685470
oscap-anaconda-addon	BZ#1665082
pacemaker	BZ#1543494
pcs	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
perl	BZ#1511131
pki-core	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, BZ#1669257 , BZ#1656856, BZ#1673296
postgresql-9.6-module	BZ#1660041
pykickstart	BZ#1637872, BZ#1612061
python-rtslib	BZ#1666377
qemu-kvm	BZ#1559240, BZ#1508139, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1592337, BZ#1570029, BZ#1689216 , BZ#1585651, BZ#1519004
redhat-release	BZ#1636338

구성 요소	티켓
redhat-support-tool	BZ#1688274
rsyslog	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645, BZ#1679512 , JIRA:RHELPLAN-10431
Scala-2.10-module	BZ#1641744
scap-security-guide	BZ#1618505, BZ#1618528, BZ#1618518
scap-workbench	BZ#1640715
selinux-policy	BZ#1664345 , BZ#1594111, BZ#1592244, BZ#1549772, BZ#1483904, BZ#1626446
setup	BZ#1591969, BZ#1663556
sos	BZ#1559836
squid	BZ#1656871
sssd	BZ#1448094, BZ#1382750, BZ#1446101, BZ#1447945, BZ#1620123, BZ#1652562 , BZ#1659457 , BZ#1669407 , BZ#1657665
subscription-manager	BZ#1654531 , BZ#1661414
subversion	BZ#1571415
swig-3.0-module	BZ#1660051
systemd	BZ#1658691
tomcatjss	BZ#1424966, BZ#1636564
tuned	BZ#1565598
valgrind	BZ#1500481, BZ#1538009
varnish	BZ#1633338
vdo	BZ#1669537
virt-manager	BZ#1599777, BZ#1643609
wpa_supplicant	BZ#1582538, BZ#1537143

구성 요소	티켓
xorg-x11-server	BZ#1687489, BZ#1698565
기타	JIRA:RHELPLAN-10347, BZ#1646563, JIRA:RHELPLAN-2306, BZ#1640697, BZ#1623712, BZ#1649404, BZ#16581198, BZ#1581997, BZ#1695584, BZ#1654280, BZ#1643294, BZ#1647612, BZ#1647612 BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1641007, BZ#1641029, BZ#1641029, BZ#1641022, JIRA:RHELPLAN-1212, BZ#1649493, BZ#1559616, BZ#1699825, BZ#1646541, BZ#1647725, BZ#1686057, BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, BZ#1650701, JIRA:RHELPLAN-10439, JIRA:RHELPLAN-104402, JIRA:RHELPLAN-10442, JIRA:RHELPLAN-10443, JIRA:RHELPLAN-10438, JIRA:RHELPLAN-2878, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA JIRA:RHELPLAN-6746, JIRA:RHELPLAN-10354, JIRA:RHELPLAN-10304, JIRA:RHELPLAN-10628, JIRA:RHELPLAN-10441, JIRA:RHELPLAN-10444, JIRA:RHELPLAN-1842, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-7291, JIRA:RHELPLAN-12764, BZ#1680177, JIRA:RHELPLAN-14607, JIRA:RHELPLAN-1820, BZ#1684947, BZ#1683712, BZ#1659609, BZ#1504934, BZ#1642765, BZ#1642765 BZ#1641014, BZ#1692746, BZ#1692746, BZ#1690207, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#16956981, BZ#1695698, BZ#1695698, BZ#1695698, BZ#1697896, BZ#1698613, BZ#1699535, BZ#1701968, BZ#1704867

감사 인사

RHEL 8 준비 챌린지의 일부로 피드백을 주신 모든 사람에게 감사드립니다. **The top 3 Wins are:**

- 알렉시스바르에스케일
- 존 Pittman
- Jake Hunsaker

부록 B. 개정 내역

0.1-3

4월 29일, LRKKKKKKKK (lspackova@redhat.com)

- 더 이상 사용되지 않는 기능 소개가 업데이트되었습니다.
- [BZ#1605216](#) 의 고정 오타.
- 손상된 링크가 수정되었습니다.

0.1-2

Thu Mar 17 2022, Lucie Makovtekton(Imanasko@redhat.com)

[JIRA:RHELPLAN-14323](#), [JIRA:RHELPLAN-14329](#), [JIRA:RHELPLAN-14330](#) 을 새로운 기능 섹션 (Virtualization)에 추가되었습니다.

0.1-1

2021년 12월 23일 2017년 12월 23일

- **soft-RoCE** 드라이버 `rdma_rxe` 에 기술 프리뷰가 [BZ#1605216](#) 및 더 이상 사용되지 않는 기능 [BZ#1878207](#) (Kernel)에 대한 정보가 추가되었습니다.

0.1-0

2021년 9월 23일, Lucie Matektonskovtekton(Imanasko@redhat.com)

- 잘못된 새 기능 설명(가상 설정)이 제거되었습니다.

0.0-9

Thu Aug 19 2021, Lucie Makovtekton(Imanasko@redhat.com)

- [YUM/DNF를 사용하여 패키지 관리를 배포 장에 추가했습니다.](#)

0.0-8

2021년 6월 23일, Lucie Matektonskovtekton(Imanasko@redhat.com)

- 새로운 기능 섹션(설치 관리자)이 업데이트되었습니다.

0.0-7

2021년 4월 6일 <mailto:lspackova@redhat.com>

- 지원되는 아키텍처 목록 개선

0.0-6

2021년 1월 28일, Lucie Matektonskovtekton(Imanasko@redhat.com)

- 기술 프리뷰 장을 업데이트했습니다.

0.0-5

2020년 12월 10일 <mailto:lspackova@redhat.com>

- SSSD에서 새로운 기능(Identity Management)에 AD VDDKs를 처리하는 방법에 대한 추가 정보를 제공합니다.

0.0-4

2020년 8월 28일 <mailto:lspackova@redhat.com>

- 개요의 내부 업그레이드에 대한 업데이트된 정보입니다.

0.0-3

2020년 3월 12일 <mailto:lspackova@redhat.com>

- 누락된 postfix RHEL 시스템 역할을 기술 프리뷰에 추가했습니다.

0.0-2

2020년 2월 12일, Jaroslav Klech (<mailto:jklech@redhat.com>)

- 아키텍처 및 새 기능 장에 완전한 커널 버전이 제공됩니다.

0.0-1

2019년 7월 30일 <mailto:imanasko@redhat.com>

- Red Hat Enterprise Linux 8.0.1 릴리스 노트

0.0-0

2019년 5월 7일 Ioanna Gkioka (igkioka@redhat.com)

- Red Hat Enterprise Linux 8.0 릴리스 정보.