Red Hat Enterprise Linux 6
6.8 Technical Notes

Technical Notes for Red Hat Enterprise Linux 6.8
Edition 8

Red Hat Customer Content Services

Technical Notes for Red Hat Enterprise Linux 6.8
Edition 8

Red Hat Customer Content Services
rhel-notes@redhat.com

Legal Notice

Abstract

The Technical Notes provide information about notable bug fixes, Technology Previews, deprecated functionality, and other details in Red Hat Enterprise Linux 6.8. For high-level coverage of the improvements implemented in Red Hat Enterprise Linux 6.8 and a list of known problems in this release, refer to the Release Notes.

Table of Contents

Preface

Red Hat Enterprise Linux minor releases are an aggregation of individual enhancement, security, and bug fix errata. The *Red Hat Enterprise Linux 6.8 Technical Notes* document provides a list of notable bug fixes, all currently available Technology Previews, deprecated functionality, and other information. The Release Notes document describes the major changes made to the Red Hat Enterprise Linux 6 operating system and its accompanying applications for this minor release, as well as known problems.

Capabilities and limits of Red Hat Enterprise Linux 6 as compared to other versions of the system are available in the Red Hat Knowledgebase article available at https://access.redhat.com/articles/rhel-limits.

For information regarding the Red Hat Enterprise Linux life cycle, refer to https://access.redhat.com/support/policy/updates/errata/.

Chapter 1. Red Hat Enterprise Linux 6.8 International Languages

Red Hat Enterprise Linux 6.8 supports installation of multiple languages and changing of languages based on your requirements.

The following languages are supported in Red Hat Enterprise Linux 6.8:

- East Asian Languages - Japanese, Korean, Simplified Chinese, and Traditional Chinese
- European Languages - English, German, Spanish, French, Portuguese Brazilian, and Russian,

The table below summarizes the currently supported languages, their locales, default fonts installed and packages required for some of the supported languages

Table 1.1. Red Hat Enterprise Linux 6 International Languages

Territory	Language	Locale	Fonts	Package Names
China	Simplified Chinese	zh_CN.UTF-8	AR PL (ShanHeiSun and Zenkai) Uni	fonts-chinese, scim-pinyin, scim-tables
Japan	Japanese	ja_JP.UTF-8	Sazanami (Gothic and Mincho)	fonts-japanese, scim-anthy
Korea	Hangul	ko_KR.UTF-8	Baekmuk (Batang, Dotum, Gulim, Headline)	fonts-korean, scim-hangul
Taiwan	Traditional Chinese	zh_TW.UTF-8	AR PL (ShanHeiSun and Zenkai) Uni	fonts-chinese, scim-chewing, scim-tables
Brazil	Portuguese	pt_BR.UTF-8	standard latin fonts	
France	French	ft_FR.UTF-8	standard latin fonts	
Germany	German	de_DE.UTF-8	standard latin fonts	
Italy	Italy	it_IT.UTF-8	standard latin fonts	
Russia	Russian	ru_RU.UTF-8	Cyrillic	dejavu-lgc-sans-fonts, dejavu-lgc-sans-mono-fonts, dejavu-lgc-serif-fonts, xorg-x11-fonts-cyrillic
Spain	Spanish	es_ES.UTF-8	standard latin fonts	

Chapter 2. Important Changes to External Kernel Parameters

This chapter provides system administrators with a summary of significant changes in the kernel shipped with Red Hat Enterprise Linux 6.8. These changes include added or updated **proc** entries, **sysctl**, and **sysfs** default values, boot parameters, kernel configuration options, or any noticeable behavior changes.

force_hrtimer_reprogram [KNL]

Force the reprogramming of expired timers in the **hrtimer_reprogram()** function.

softirq_2ms_loop [KNL]

Set **softirq** handling to 2 ms maximum. The default time is the existing Red Hat Enterprise Linux 6 behaviour.

tpm_suspend_pcr=[HW,TPM]

Specify that, at suspend time, the **tpm** driver should extend the specified principal components regression (PCR) with zeros as a workaround for some chips which fail to flush the last written PCR on a **TPM_SaveState** operation. This guarantees that all the other PCRs are saved.

Format: integer pcr id

/proc/fs/fscache/stats

Table 2.1. class Ops:

new:	ini=N	Number of async ops initialised
changed:	rel=N	will be equal to ini=N when idle

Table 2.2. new class CacheEv

nsp=N	Number of object lookups or creations rejected due to a lack of space
stl=N	Number of stale objects deleted
rtr=N	Number of objects retired when relinquished
cul=N	Number of objects culled

/proc/sys/net/core/default_qdisc

The default queuing discipline to use for network devices. This allows overriding the default queue discipline of **pfifo_fast** with an alternative. Since the default queuing discipline is created with no additional parameters, it is best suited to queuing disciplines that work well without configuration, for example, a stochastic fair queue (**sfq**). Do not use queuing disciplines like Hierarchical Token Bucket or Deficit Round Robin, which require setting up classes and bandwidths.

Default: pfifo_fast

/sys/kernel/mm/ksm/max_page_sharing

Maximum sharing allowed for each KSM page. This enforces a deduplication limit to avoid the virtual memory **rmap** lists to grow too large. The minimum value is 2 as a newly created KSM page will have at least two sharers. The **rmap** walk has O(N) complexity where **N** is the number of **rmap_items**, that is virtual mappings that are sharing the page, which is in turn capped by

max_page_sharing. So this effectively spreads the linear O(N) computational complexity from **rmap** walk context over different KSM pages. The **ksmd** walk over the **stable_node chains** is also O(N), but N is the number of **stable_node dups**, not the number of **rmap_items**, so it has not a significant impact on **ksmd** performance. In practice the best **stable_node dups** candidate is kept and found at the head of the **dups** list. The higher this value the faster KSM merges the memory, because there will be fewer **stable_node** dups queued into the **stable_node** chain->hlist to check for pruning. And the higher the deduplication factor is, but the slowest the worst case **rmap** walk could be for any given KSM page. Slowing down the **rmap** walk means there will be higher latency for certain virtual memory operations happening during swapping, compaction, NUMA balancing, and page migration, in turn decreasing responsiveness for the caller of those virtual memory operations. The scheduler latency of other tasks not involved with the VM operations doing the **rmap** walk is not affected by this parameter as the **rmap** walks are always scheduled friendly themselves.

/proc/sys/net/core/default_qdisc

The default queuing discipline to use for network devices. This allows overriding the default queue discipline of **pfifo_fast** with an alternative. Since the default queuing discipline is created with no additional parameters so is best suited to queuing disciplines that work well without configuration, for example, a stochastic fair queue (**sfq**). Do not use queuing disciplines like Hierarchical Token Bucket or Deficit Round Robin which require setting up classes and bandwidths.

Default: pfifo_fast

/sys/kernel/mm/ksm/stable_node_chains_prune_millisecs

How frequently to walk the whole list of **stable_node** "dups" linked in the **stable_node chains** in order to prune stale **stable_node**. Smaller millisecs values will free up the KSM metadata with lower latency, but they will make **ksmd** use more CPU during the scan. This only applies to the **stable_node** chains so it is a noop unless a single KSM page hits **max_page_sharing**. In such a case there are no **stable_node** chains.

/sys/kernel/mm/ksm/stable_node_chains

Number of stable node chains allocated. this is effectively the number of KSM pages that hit the **max_page_sharing** limit.

/sys/kernel/mm/ksm/stable_node_dups

Number of stable node dups queued into the **stable_node** chains.

Part I. Notable Bug Fixes

This part describes bugs fixed in Red Hat Enterprise Linux 6.8 that have a significant impact on users.

Chapter 3. General Updates

abrt no longer missing a dependency on *python-argparse*

A previously missing dependency of the *abrt* packages on the *python-argparse* package resulting in errors like **`ImportError: No module named argparse`** has been fixed. This problem usually occurred if customers upgraded from an earlier version of Red Hat Enterprise Linux, or during a fresh installation if customers removed the *nfs-utils* or *ipa-client* packages. (BZ#1246539)

`rds-stress` can now correctly send messages of varying size

The **`rds-stress`** command previously could not send Reliable Datagram Sockets (RDS) messages of varying sizes if RDMA was enabled due to bugs in both the kernel and in the *rds-tools* package. These bugs have been fixed and you can now send RDS messages of any size as expected. (BZ#746716)

Chapter 4. Authentication and Interoperability

The ca.subsystem.certreq parameter is no longer reported missing

Previously, Identity Management (IdM) expected the **`ca.subsystem.certreq`** parameter to be defined in the **`CS.cfg`** public key infrastructure (PKI) configuration file. When starting the IdM server, an error occurred if **`ca.subsystem.certreq`** was missing. The error was not necessary because neither PKI nor IdM services use the parameter. To fix this problem, PKI code has been updated to ensure the parameter is only retrieved if it exists. (BZ#1313207)

The ipa-server-install utility no longer terminates unexpectedly due to unexpected comment lines in CS.cfg

An attempt to install an Identity Management server previously sometimes failed due to a problem with the *pki-common* package. The fail occurred because the **`CS.cfg`** certificate authority (CA) configuration file which was being parsed contained unexpected comment lines before configuration. This problem has been fixed by making the parsing code ignore comment and blank lines. (BZ#1306989)

Installing an IdM server no longer fails if Java 1.8 is installed

The Public Key Infrastructure (PKI) server, included in Identity Management (IdM), supports Java version 1.7 on Red Hat Enterprise Linux 6. The **`ipa-server-install`** installation script failed on systems where the *java-1.8* package was installed and selected as the current system **`java`** using the **`alternatives`** utility. To fix this problem, the *pki-core* code has been updated to bypass **`alternatives`** on Red Hat Enterprise Linux 6 by forcing PKI servers to always run under OpenJDK version 1.7 regardless of the version of **`java`** selected using **`alternatives`**. (BZ#1290535)

Samba no longer denies access when sharing the root directory of the system

Previously, due to a missing path check, Samba denied access when sharing the root directory of the system by using the **`path = /`** setting in the **`/etc/samba/smb.conf`** file. With this update, Samba no longer incorrectly treats the `/` path as a symbolic link and does not incorrectly deny access in the described situation. (BZ#1305870)

Acquiring keytabs takes longer with SELinux after memory leaks have been fixed

Previously, SELinux support in the *krb5* packages caused *krb5* to leak memory. This bug has been fixed. Note that acquiring keytabs now takes longer than before when SELinux is in **`enforcing`** or **`permissive`** mode. (BZ#1311287)

`sudo` smart refresh updates no longer fail due to USN parsing errors

System Security Services Daemon (SSSD) did not correctly handle the format of the **`modifyTimestamp`** attribute of the OpenLDAP server. Consequently, smart refresh updates for the **`sudo`** utility did not work. After the user changed a **`sudo`** rule with SSSD running, the logs showed an error stating that SSSD was unable to parse the Update Sequence Number (USN) scheme. This update fixes the problem, and smart refresh updates now work in the described situation. (BZ#1312062)

SSSD stores sudo rules correctly when `id_provider = ipa` is set

Identity Management version 3.0 and previous use different format for the **ipasudocmd** distinguished name (DN). Consequently, the System Security Services Daemon (SSSD) service was unable to store **sudo** rules correctly when the **id_provider** option was set to **ipa** in the **/etc/sssd/sssd.conf** file. This update fixes the problem, and **sudo** rules now work as expected in the described situation. (BZ#1313940)

The user is prompted for smart card PIN as expected

Due to insufficient SELinux policy rules, the **ppl_child** process, running in the **sssd_t** SELinux domain, was unable to manage the authentication cache and connect to Apache ports. Consequently, the system did not prompt the user for smart card PIN. The SELinux policy rules, provided by the *selinux-policy* package, have been updated to allow this functionality. As a result, the user is prompted for smart card PIN as expected in the described situation. (BZ#1299066)

Cloning a PKI server with an externally-signed CA certificate to Red Hat Enterprise Linux 7 no longer fails

Previously, when a Red Hat Enterprise Linux 6 public key infrastructure (PKI) server was installed with an externally-signed certificate authority (CA) certificate, the subsystem user was not created properly. Consequently, cloning to Red Hat Enterprise Linux 7 failed.

For new Red Hat Enterprise Linux 6 installations, the code has been fixed to create the subsystem user, add it to the subsystem group, and map the subsystem certificate to the user properly. For existing Red Hat Enterprise Linux 6 installations, the code has been modified to automatically restore the subsystem user to the correct configuration on restart.

As a result, cloning to Red Hat Enterprise Linux 7 now succeeds in the described situation. (BZ#1256039)

`ypserv` no longer fails if the `domainname` parameter is unset

Previously, the **ypserv** service failed to start when the **domainname** parameter was not set in the **/etc/init.d/ypserv** file. This update moves the check for **domainname** to the **yppasswdd** service, and in the described circumstances, **ypserv** now starts as expected. (BZ#456249)

`yppasswd` now correctly reports a failure of a user password change

Prior to this update, when the **yppasswd** service failed to change the password of a **yppasswdd** user, it still reported a success. A test has been added to **yppasswdd** that verifies whether the write operation was successful. As a result, if **yppasswdd** fails to change a user password, an error message is now logged about it. (BZ#747334)

`ypserv` now correctly reports a non-existent map

The **ypserv** service previously incorrectly returned an **Internal NIS error** error message when a NIS client asked for a non-existent map using the **yp_first** or **yp_next** system calls. Now, **ypserv** correctly returns the **No such map in server's domain** error message in this scenario. (BZ#988203)

`mknetid` no longer crashes when the `passwd` file contains empty lines or an unexpected format

Previously, using the **mknetid** utility on the **passwd** file with empty lines or an unexpected format in some cases caused **mknetid** to terminate unexpectedly. With this update, **mknetid** ignores the redundant elements in the **passwd** file, and no longer crashes in the situation described. (BZ#1071962)

`ypbind` no longer restarts on every renewal of DHCP

Prior to this update, the **`ypbind`** service restarted on every renewal of the dynamic host configuration protocol (DHCP), which caused NIS lookups to be slower, and in some cases to time out. Now, **`ypbind`** restarts on a DHCP renewal only if any changes occurred on the NIS domain or the NIS domain or the NIS server. As a result, NIS lookups are faster and experience less timeouts. (BZ#1238771)

Chapter 5. Clustering

Pacemaker does not update the fail count when `on-fail=ignore` is used

When a resource in a Pacemaker cluster failed to start, Pacemaker updated the resource's last failure time and fail count, even if the **`on-fail=ignore`** option was used. This could cause unwanted resource migrations. Now, Pacemaker does not update the fail count when **`on-fail=ignore`** is used. As a result, the failure is displayed in the cluster status output, but is properly ignored and thus does not cause resource migration. (BZ#1200853)

pacemaker and other Corosync clients again connect successfully

Previously, the libqb library had a limited buffer size when building names for IPC sockets. If the process IDs on the system exceeded 5 digits, they were truncated and the IPC socket names could become non-unique. As a consequence, clients of the Corosync cluster manager could fail to connect and could exit, assuming the cluster services were unavailable. This could include pacemaker which could fail, leaving no cluster services running. This update increases the buffer size used for building IPC socket names to cover the maximum possible process ID number. As a result, pacemaker and other Corosync clients start consistently and continue running regardless of the process ID size. (BZ#1276345)

Security features added to the luci interface to prevent clickjacking

Previously, **`luci`** was not defended against clickjacking, a technique to attack a web site in which a user is tricked into performing unintended or malicious actions through purposefully injected elements on top of the genuine web page. To guard against this type of attack, **`luci`** is now served with **`X-Frame-Options: DENY`** and **`Content-Security-Policy: frame-ancestors 'none'`** headers that are intended to prevent **`luci`** pages from being contained within external, possibly malicious, web pages. Additionally, when a user configures **`luci`** to use a custom certificate and is properly anchored with a recognized CA certificate, a **`Strict-Transport-Security`** mechanism with a validity period of 7 days is enforced in newer web browsers, also by means of a dedicated HTTP header. These new static HTTP headers can be deactivated, should it be necessary to overcome incompatibilites, and a user can add custom static HTTP headers in the **`/etc/sysconfig/luci`** file, which provides examples. (BZ#1270958)

`glusterfs` can now properly recover from failed synchronization of cached writes to backend

Previously, if synchronization of cached writes to a Gluster backend failed due to a lack of space, write-behind marked the file descriptor (**`fd`**) as bad. This meant virtual machines could not recover and could not be restarted after synchronization to backend failed for any reason.

With this update, **`glusterfs`** retries synchronization to backend on error until synchronization succeeds until a flush. Additionally, file descriptors are not marked as bad in this scenario, and only operations overlapping with regions with failed synchronizations fail until the synchronization is successful. Virtual machines can therefore be resumed normally once the underlying error condition is fixed and synchronization to backend succeeds. (BZ#1171261)

Fixed an AVC denial error when setting up `Gluster` storage on NFS Ganesha clusters

Attempting to set up Gluster storage on an NFS-Ganesha cluster previously failed due to an Access Vector Cache (AVC) denial error. The responsible SELinux policy has been adjusted to allow handling of volumes mounted by NFS-Ganesha, and the described failure no longer occurs. (BZ#1241386)

Installing *glusterfs* no longer affects default `logrotate` settings

When installing the *glusterfs* packages on Red Hat Enterprise Linux 6, the **`glusterfs-logrotate`** and **`glusterfs-georep-logrotate`** files were previously installed with several global **`logrotate`** options. Consequently, the global options affected the default settings in the **`/etc/logrotate.conf`** file. The glusterfs RPMs have been rebuilt to prevent the default settings from being overridden. As a result, global settings in **`/etc/logrotate.conf`** continue to function as configured without being overridden by settings from **`glusterfs logrotate`** files. (BZ#1171865)

Fence agent for DM Multipath no longer loses SCSI keys on non-cluster reboot

Previously, the fence agent for DM Multipath lost SCSI keys when the node was not rebooted using cluster methods. This resulted in an error when the cluster tried to fence the node. With this update, keys are properly regenerated after each reboot in this situation. (BZ#1254183)

Fence agent for HP Integrated Lights-Out (iLo) now uses TLS1.0 automatically when connection over SSL v3 fails

Previously, the fence agent for HP Integrated Lights-Out (iLO) required the tls1.0 argument in order to use TLS1.0 instead of SSL v3. With this update, TLS1.0 is used automatically when the connection over SSL v3 fails. (BZ#1256902)

Chapter 6. Compiler and Tools

`tcsh` no longer in an inconsistent state after a command interruption

Interrupting the **`eval sleep 10`** command left the **`tcsh`** shell in an incosistent state. Consequently, it was necessary to press the Ctrl+D key combination twice to successfully exit the shell. With this update, **`tcsh`** correctly exits after pressing Ctrl+D once. (BZ#1219923)

Correct parsing of the `if` statement in `tcsh`

The **`tcsh`** shell failed to correctly parse the **`if`** statements if there was no space before the **`then`** keyword. Consequently, incorrect branches of the **`if`** statement were processed, which led to execution of incorrect sections of the code and misbehavior. The **`tcsh`** shell has been fixed to correctly process the source code when spaces before the **`then`** keyword are missing. (BZ#1231097)

RELRO protection now properly applied when requested

Previously, binary files started by the system loader would, in some cases, lack the Relocation Read-Only (RELRO) protection even though this had been explicitly requested when the application was built. This was due to a miscommunication between the static linker and the system loader. The underlying source code of the linker has been adjusted to ensure that it makes it possible for the loader to apply the RELRO protection, thus restoring the security feature for applications. Applications and all dependent object files, archives, and libraries built with a previous version of *binutils* should be rebuilt to correct this defect. (BZ#1227839)

Escape sequences correctly interpreted by `tcsh`

Previously, the **`tcsh`** command-language interpreter incorrectly consumed certain escape sequences, which start with the backslash `\` character. This update fixes the interpretation of the escape sequences, and **`tcsh`** now returns the same results as in Red Hat Enterprise Linux 5. (BZ#1301857)

OpenMP heuristics adjusted for higher performance on multi-CPU systems

Heuristics used by the GNU OpenMP runtime (**`libgomp`**) to determine latency-reduction measures have been adjusted to take into account CPU affinity on multiprocessor systems and to use a lower spin count. As a result, programs that appeared deadlocked or ran very slowly due to resource starvation now run considerably faster. (BZ#1229852)

Reduced lock contention and increased performance for threaded applications calling localtime_r()

Lock contention in the localtime_r() function previously decreased performance for threaded applications that needed to call this function frequently. The lock acquisition in the **`glibc`** library's internal routines has been reorganized to decrease the possibility of lock contention. (BZ#1244585)

The `make` utility no longer terminates unexpectedly when processing `$$eval` commands or the `$(shell)` construct.

Due to bugs in the *make* package, the **`make`** utility sometimes terminated unexpectedly when processing files which contained **`$$eval`** commands or **`$(shell)`** processes. These bugs have been fixed, and **`make`** no longer crashes in these situations. (BZ#835483)

Parallel builds no longer terminate unexpectedly

Previously, a bug caused the **make** utility to display an error message and terminate unexpectedly when executing parallel builds. This has now been fixed, and parallel builds no longer crash due to this bug. (BZ#861189)

Parallel builds no longer terminate unexpectedly due to `$(eval)` constructs

Previously, a bug caused the **make** utility to terminate unexpectedly when executing parallel builds containing **$(eval)** constructs. This has now been fixed, and parallel builds no longer crash due to this bug. (BZ#1093149)

Multithreaded applications no longer crash when calling dprintf() and fork() concurrently

Multithreaded applications that use the dprintf() and fork() **glibc** functions concurrently could previously terminate unexpectedly with a segmentation fault. With this update, the fork() implementation has been fixed to ignore temporary streams created by dprintf(), and the described problem no longer occurs. (BZ#1275384)

Core C library (glibc) enhanced to increase malloc() scalability

A defect in the implementation of the **malloc()** function could result in the unnecessary serialization of memory allocation requests across threads. This update fixes the problem and substantially increases the concurrent throughput of allocation requests for applications that frequently create and destroy threads. (BZ#1264189)

Improved documentation in `/etc/mcelog/mcelog.conf`

The default **mcelog** configuration file found at **/etc/mcelog/mcelog.conf** now contains better descriptions for several available options, such as **mem-ce-error-log**. (BZ#1170580)

The dynamic linker (ld.so) does not fail anymore when an audit module provides a DSO path

Previously, when an audit module provided an alternate DSO (dynamic shared object) path, the **ld.so** dynamic linker terminated unexpectedly with a segmentation fault. This update addresses the bug, and now the dynamic linker keeps track of the original DSO path for future reference and does not fail anymore. (BZ#1211098)

glibc and *tzdata* updates no longer replace `/etc/localtime` symbolic links

Previously, on systems where the **/etc/localtime** file was a symbolic link, updates of the *glibc* and *tzdata* packages replaced the link with the time-zone file defined in the **/etc/sysconfig/clock** configuration file. A patch has been applied to address the problem and **/etc/localtime** as a symbolic link is no longer replaced. For optimum compatibility, it is recommended not to modify **/etc/localtime** directly. Instead, edit the **/etc/sysconfig/clock** configuration file and execute the **tzdata-update** command afterwards. (BZ#1200555)

`glibc` POSIX real-time support no longer fails on large TLS data

A defect in the POSIX real-time support in the **glibc** library caused asynchronous I/O or certain timer API calls to fail in the presence of large thread-local storage (TLS) data. The **librt** library has been fixed, and

the impacted APIs no longer return error messages when large TLS data is present in applications. (BZ#1299319)

`iconv` no longer adds a redundant shift sequence

Previously, the **`iconv`** utility sometimes mishandled character conversion for the IBM930, IBM933, IBM935, IBM937, and IBM939 character sets. Consequently, a redundant shift sequence was included in the output of **`iconv`**. The generated non-conforming output could result in an inability to read the output data. The character conversion routines have been corrected and no longer output a redundant shift sequence. (BZ#1293914)

`/etc/gai.conf` no longer listed as belonging to the *glibc-common* package

Previously, the **`/etc/gai.conf`** configuration file could be flagged as modified when the **`rpm -V`** command was used to check the *glibc-common* package, even though that package did not own that file. This update removes the erroneous file entry from *glibc-common*. As a result **`rpm -V glibc-common`** no longer lists the **`/etc/gai.conf`** configuration file as modified. (BZ#1223818)

Naming convention for *java-1.8.0-openjdk* installation made consistent

Previously, the automatically generated name for the installation directory for the *java-1.8.0-openjdk* package was not consistent with installation directory names for the *java-1.6.0-openjdk* and *java-1.7.0-openjdk* packages. As a consequence, applications and automated tools that expected a consistent Java naming pattern failed when interacting with *java-1.8.0-openjdk*. With this update, *java-1.8.0-openjdk* installation directory follows the same pattern as the previous versions, which prevents the described problem. Note that due to the changes in the path structure, it is now impossible to revert to a version of *java-1.8.0-openjdk* prior to this update. (BZ#1217177)

LDAP bind passwords are properly obfuscated

In some cases, it was previously possible for the **`sosreport`** utility to capture LDAP bind credentials in plain text. This problem has been fixed, and LDAP bind passwords are now obfuscated in **`sosreport`** as expected. (BZ#1227462)

`sosreport` no longer times out when `ipa-replica-image` is used

Previously, using the **`ipa-replica-image`** command caused the **`sosreport`** utility to take longer to execute and in some cases to time out. In addition, not all the information needed to troubleshoot certification-related problems was captured in **`sosreport`**. This update ensures that sufficient certification-related information is collected by **`sosreport`** and removes **`ipa-replica-image`**, which prevents the described timeouts from occurring. (BZ#1203947)

`tar` now correctly sets ACL when `--acls` is used

Previously, when a tar file was extracted using the **`--acls`** option, the extracted files inherited the default Access Control Lists (ACL) from the parent directory. With this update, when **`--acls`** is used, **`tar`** extracts all files and directories with the same ACLs that are set in the archive. (BZ#1220891)

`tar` correctly handles archives created by a user with a big UID

When a user with a UID or GID greater than 2097151 created a pax archive, the archive was created correctly but **tar** returned a misleading warning message and a non-zero exit status. This bug has been fixed, **tar** now exits with the correct exit status and no false warning messages. (BZ#1247788)

OProfile fixed to prevent buffer overflow caused by long arguments

Prior to this update, the argument-check code in OProfile suffered from a problem that could cause a buffer overflow when passed a long path name or event unit mask. This update addresses the problems, and long arguments no longer cause a buffer overflow. (BZ#1206242)

OProfile now works correctly on IBM POWER7+ systems

Prior to this update, **OProfile** for IBM POWER Systems was built using an old version of the **libpfm** libraries, which resulted in support for the POWER7+ processors not being available and the **operf** and **ocount** tools not being able to run successfully on new IBM POWER Systems. This bug has been fixed, and the **operf** and **ocount** commands now work as expected on POWER7+ systems as well. (BZ#1303970)

Applications no longer fail with 'dlopen: cannot load any more object with static TLS'

The **glibc** dynamic loader was unable to load more than 16 shared libraries that make use of static thread-local storage (TLS). Consequently, applications could fail with an error message, **dlopen: cannot load any more object with static TLS**. This bug has been fixed, and applications now start correctly in this scenario, provided sufficient static storage space is available. (BZ#1198802)

GDB now generates smaller core files and respects core-dump filtering

The **gcore** command, which provides **GDB** with its own core-dumping functionality, has been updated to more closely simulate the function of the Linux kernel core-dumping code, thus generating smaller core-dump files. **GDB** now also respects the **/proc/PID/coredump_filter** file, which controls what memory segments are written to core-dump files. (BZ#1085906)

GDB no longer kills running processes with deleted executables

Prior to this update, **GDB** attempting to attach to a running process with a deleted executable would accidentally kill the process. This bug has been fixed, and **GDB** no longer erroneously kills processes with deleted executables. (BZ#1219747)

Fixed signal handling in Bash

Due to the signal handler function calling certain signal-unsafe functions such as **malloc()**, the **Bash** shell in some cases became unresponsive after it received a signal. This update ensures that the signal handler no longer calls signal-unsafe functions, which prevents the described bug from occurring. (BZ#868846)

Command substitution now works inside arithmetic expressions in bash

Previously, **bash** did not save input line state across recursive calls to the shell parser, causing the shell to fail to parse command substitution inside arithmetic expressions. For example, the following command failed:

```
for point in "/boot"; do disk_dir["$( df -P $point | awk '{print $1}' )"]=1;
done
```

This bug has been fixed, **bash** now saves input line state before making recursive calls to the parser, and command substitution now works inside arithmetic expressions. (BZ#1207803)

KornShell now resets and modifies signal traps as expected and no longer crashes

Previously, KornShell (ksh) terminated unexpectedly with a segmentation fault when attempting to reset or modify certain signal traps. With this update, ksh does not attempt to free memory used for a string literal. As a result, ksh no longer crashes in the described situation. (BZ#1247383)

Printing the `ls` output no longer takes a long time when extended attributes and ACLs are not supported

Previously, listing files and directories on file systems without support for extended attributes and access control lists (ACLs) involved unnecessary and high-cost lstat() and lgetxattr() system calls for every file. As a consequence, it could take several seconds to print the output of the **ls** command. Now, when the result of the system call is **not supported**, **ls** does not try the call again on this mount point. As a result, the printing speed of the **ls** output has improved significantly in the described situation. (BZ#1248141)

KornShell no longer omits a background process in a command list when forking

Due to a bug in KornShell (ksh), forking a background process in a command list could previously omit that process completely. The source code that handles forking sub-processes has been modified, and a background process in a command list is now executed as expected. (BZ#1217236)

Multibyte characters are no longer corrupted in KornShell

Previously, the multibyte unicode parser received shifted input after a command-line option variable in KornShell (ksh). Consequently, the multibyte character after a command-line variable became corrupted. A patch has been applied to fix multibyte input parsing. As a result, multibyte characters are handled correctly in the described situation. (BZ#1256495)

The Oracle ACFS is now included among known file systems

Previously, the Oracle ASM Cluster file system (ACFS) was not listed among known file systems for the **stat** and **tail** utilities. As a consequence, the **tail** utility printed an error message stating that the file system was not recognized. ACFS has been added to the list of known file systems, and the error message no longer appears in the described situation.

In addition, other file systems recognized by upstream have been added to the list of known file systems as well, namely **acfs**, **bpf_fs**, **btrfs_test**, **configfs**, **efivarfs**, **exofs**, **f2fs**, **hfs+**, **hfsx**, **hostfs**, **ibrix**, **logfs**, **nsfs**, **overlayfs**, **smackfs**, **snfs**, **tracefs**, and **ubifs**. (BZ#1280333)

The `netfs stop` command now only unmounts relevant loop devices

Previously, all loop devices were umnounted when stopping the **netfs** service. With this update, when running the **netfs stop** command, only relevant loop devices are unmounted; that is, the ones that are mounted on top of a network mount. (BZ#1156231)

Chapter 7. Desktop

pdftops now generates valid PostScript files

Previously, the pdftops utility of Poppler converted PDF files with embedded PostScript Type 1 fonts in the Printer Font Binary (PFB) format together with the PFB headers. As a consequence, incorrect PostScript files were produced from such conversions. Poppler has been fixed to recognize the PFB headers and exclude them from the conversion, and the described problem no longer occurs. (BZ#1232210)

Creating thumbnails with Nautilus works as expected

Prior to this update, Nautilus was restricting the stack size for the thread that was creating thumbnails, and the libraries creating those thumbnails were at some point out of memory. As a consequence, Nautilus terminated unexpectedly when creating a thumbnail of big images, specially images of the jp2 type. This bug has been fixed, and Nautilus no longer crashes when creating thumbnails. (BZ#1268970)

`xfreerdp` no longer crashes when connecting to a CA-signed RDP server

Previously, the **`xfreerdp`** client terminated unexpectedly when connecting to a Certification Authority (CA)-signed Remote Desktop Protocol (RDP) server. With this update, the pointer dereference in CA verification code that caused the crash has been fixed, and connecting to a CA-signed RDP server works as expected. (BZ#1186916)

Responsiveness fixes for *gnome-vfs2* user interfaces

Previously, the *gnome-vfs2* service called the **`stat()`** function for every file on the Multiversion File System (MVFS), used for example by the IBM Rational ClearCase utilities. This behavior significantly slowed down file operations. With this update, the unnecessary **`stat()`** operations have been limited. As a result, *gnome-vfs2* user interfaces, such as Nautilus, are more responsive on MVFS. (BZ#917810)

`gvfs` correctly checks if the metadata database has been loaded

Prior to this update, the **`gvfs`** metadata daemon or client applications such as Nautilus terminated unexpectedly when uninitialized internal structures were used for corrupted or unreadable metadata databases. This update adds a missing check that verifies that the metadata database has been loaded properly. As a result, in the described circumstances, an error is returned instead of a crash occurring. (BZ#1110451)

The `gvfs` archive backend no longer crashes due to certain archive file paths

Mounting the **`gvfs`** archive backend previously terminated unexpectedly when used with archives that had a single dot (".") character as a component of their file path. With this update, these file path components are skipped, and in the mentioned situation, the archive backend mounts correctly. (BZ#713179)

Chapter 8. Directory Server in Red Hat Enterprise Linux

About Directory Server for Red Hat Enterprise Linux

This section describes changes in the main server component for Red Hat Directory Server - the *389-ds-base* package, which includes the LDAP server itself and command line utilities and scripts for its administration. This package is part of the Red Hat Enterprise Linux base subscription channel and therefore available on all Red Hat Enterprise Linux Server systems due to Red Hat Identity Management components which depend on it.

Additional Red Hat Directory Server components, such as the **`Directory Server Console`**, are available in the **`rhel-x86_64-server-6-rhdirserv-9`** additional subscription channel. A subscription to this channel is also required to obtain support for Red Hat Directory Server. Changes to the additional components in this channel are not described in this document.

Red Hat Directory Server version 9 is available for Red Hat Enterprise Linux 6. See https://access.redhat.com/products/red-hat-directory-server/get-started-v9 for information about getting started with Directory Server 9, and https://access.redhat.com/documentation/en/red-hat-directory-server/?version=9 for full documentation. (BZ#1333801)

Large amounts of skipped updates in fractional replication no longer cause performance loss

During fractional replication, if a large number of skipped updates was present, the supplier could previously acquire a replica for a long time and fail to update the Replica Update Vector (RUV) at the end of the session. This then caused the next session to evaluate the same skipped updates, resulting in poor performance. This bug has been fixed by adding a system subentry which is occasionally updated even if there are no applicable changes to be replicated, and the problem no longer occurs. (BZ#1259383)

Fixed a crash while trimming the retro changelog

When trimming the retro changelog (**`retroCL`**), entries are first deleted from the changelog itself and then also from the cache. The **`389-ds-base`** server was, however, missing a check to verify that the entries are actually present in the cache, which could lead to the server attempting to delete nonexistent entries and subsequently crash on systems where not all changelog entries could fit in the cache due to its small size. A check has been added to make sure only entries actually present in the cache are being deleted, and the server no longer crashes when trimming the retro changelog. (BZ#1244970)

Fixed a crash in the backend add function

When a callback at **`BE_TXN`** in the backend add function failed on a cached entry, the function was attempting to free the entry twice instead of removing it from the cache and then freeing it. This update adds remove and free code to the backend add function and the function no longer attempts to free cached entries twice. (BZ#1265851)

`389-ds-base` server no longer crashes when attempting to replace a nonexistent attribute

When a replace operation for a nonexistent attribute was performed without providing new values, the entry was stored with incorrect metadata: an empty deleted value without an attribute deletion change state number (CSN). This entry could then result in memory corruption and cause the server to terminate unexpectedly. To fix this bug, additional space to store metadata is now allocated and the server no longer crashes in this scenario. (BZ#1298496)

`389-ds-base` no longer hangs due to modified entry remaining locked

During a modify operation, the modified entry is inserted into entry cache and locked until the modified entry is returned. In cases where the entry is removed from the entry cache after it is committed but before the return operation, the modified entry previously remained locked, and any subsequent write operations on the same entry then caused the server to hang. This bug has been fixed by adding a flag so that the entry can be unlocked whether it is present in the entry cache or not, and the server no longer hangs in this situation. (BZ#1273552)

Fixed a deadlock during backend deletion in Directory Server

Previously, transaction information was not passed to one of the database helper functions during backend deletion. This could result in a deadlock if a plug-in attempted to access data in the area locked by the transaction. With this update, transaction info is passed to all necessary database helper functions, and a deadlock no longer occurs in the described situation. (BZ#1278585)

`ns-slapd` no longer crashes on multiple asynchronous searches if a request is abandoned

When multiple simple paged results searches were requested asynchronously in a persistent connection and one of the requests was abandoned, contention among the asynchronous requests could occur and cause the **`ns-slapd`** service to crash. This bug has been fixed and **`ns-slapd`** no longer crashes due to abandoned requests. (BZ#1247792)

Simple paged results slots are now being correctly released after search failure

Previously, if a simple paged results search failed in the Directory Server backend, its slot was not released, which caused the connection object to accumulate unreleased slots over time. This problem has been fixed, and slots are now correctly released in the event of a search failure. (BZ#1290243)

`ns-slapd` no longer crashes when freeing a search results object

Previously, when Directory Server freed a search results object, there was a brief period of time before the freed information was set to the **`pagedresults`** handle. If the **`paged-results`** handle was released due to a timeout in during this period, a double free event occured, causing **`ns-slapd`** to crash. This problem has been eliminated and double free no longer occurs when freeing search results objects. (BZ#1267296)

Fixed a deadlock in asynchronous simple paged results requests

A previous fix to deadlock in the asynchronous simple paged results requests caused another self deadlock due to a regression. To address this problem, a simple **`PR_Lock`** on a connection object has been replaced with a re-entrant **`PR_Monitor`**. As a result, the deadlock no longer occurs. (BZ#1296694)

Deletion of attributes without a value on the master server now replicates correctly

Previously, when an attribute which does not have a value on the master server was deleted, the deletion was not replicated to other servers. The regression that caused this bug has been fixed and the change now replicates as expected. (BZ#1251288)

Directory Server no longer logs false `attrlist_replace` errors

Previously, Directory Server could in some circumstances repeatedly log **attrlist_replace** error messages in error. This problem was caused by memory corruption due to a wrong memory copy function being used. The memory copy function has been replaced with **memmove**, which prevents this case memory corruption, and the server no longer logs these error messages. (BZ#1267405)

cleanAllRUV now clears the changelog completely

Previously, after the **cleanAllRUV** task finished, the changelog still contained entries from the cleaned **rid**. As a consequence, the RUV could contain undesirable data, and the RUV element could be missing the replica URL. Now, **cleanAllRUV** cleans changelog completely as expected. (BZ# 1270002)

Replication failures no longer result in missing changes after additional updates

Previously, if a replicated update failed on the consumer side, it was never retried due to a bug in the replication asynchronous result thread which caused it to miss the failure before another update was replicated successfully. The second update also updated the consumer Replica Update Vector (RUV), and the first (failed) update was lost. In this release, replication failures cause the connection to close, stopping the replication session and preventing any subsequent updates from updating the consumer RUV, which allows the supplier to retry the operation in the next replication session. No updates are therefore lost. (BZ#1294770)

Unnecessary keep alive entries no longer cause missing replication

Previously, a **keep alive** entry was being created at too many opportunities during replication, potentially causing a race condition when adding the entry to the replica changelog and resulting in operations being dropped from the replication. With this update, unnecessary **keep alive** entry creation has been eliminated, and missing replication no longer occurs. (BZ#1307152)

nsMatchingRule is now correctly applied to attribute information

Previously, when **nsMatchingRule** was dynamically updated in an index entry, the value was not applied to the attribute information. This caused the **dbverify** utility to report database corruption in error. In this release, **nsMatchingRule** changes are correctly applied to attribute information, and **dbverify** no longer falsely reports database corruption. (BZ#1236656)

Tombstone entries no longer create unnecessary index entries

When an entry is deleted, its indexed attribute values are also removed from each index file. However, if the entry is turned into a tombstone entry, reindexing previously added the removed attribute value back into the index. This bug has been fixed, and index files no longer contain unnecessary key-value pairs generated by tombstone entries. (BZ#1255290)

Index is now updated properly when several values of the same attribute are deleted

Previously, when several values of the same attribute were deleted using the **ldapmodify** command, and at least one of them was added again during the same operation, the equality index was not updated. As a consequence, an exact search for the re-added attribute value did not return the entry. The logic of the index code has been modified to update the index if at least one of the values in the entry changes, and the exact search for the re-added attribute value now returns the correct entry. (BZ#1282457)

COS cache now correctly adds all definitions

A previous bug fix related to the Class of Service (COS) object cache introduced a regression which caused it to stop adding definitions after the first one, instead of adding all definitions. This problem has been fixed and the COS cache now correctly adds all definitions as designed. (BZ#1259546)

Improved ACL performance

Previously, unnecessarily complicated regular expressions were used in the Access Control List (ACL) implementation in Directory Server. These regular expressions have been removed and the ACL implementation reworked, resulting in improved performance. (BZ#1236156)

`ntUserlastLogon` and `ntUserlastLogoff` attributes are now synchronized between Directory Server and Active Directory

Previously, **`WinSync`** account synchronization could not update the **`ntUserlastLogon`** and **`ntUserlastLogoff`** attributes in Directory Server when synchronizing with Active Directory. This bug has been fixed and these attributes are now being updated correctly based on the **`lastLogonTimestamp`** and **`lastLogoffTimestamp`** attributes in Active Directory. (BZ#1245237)

Chapter 9. Installation and Booting

Add-on repositories are now handled correctly when generating and reading kickstart files.

Previously, installation would stop and display an error when performing an installation from a kickstart file generated by a previous installation which used optical media, and enabled one or more add-on repositories. With this update, generated kickstart files will include commands to automatically enable add-on repositories when necessary. (BZ#1099178)

The `zerombr` command is now correctly added to `anaconda-ks.cfg` when installing using `kickstart`

Previously, when an installation was performed with the **kickstart** utility using the **zerombr** option, this option was not added to the generated **/root/anaconda-ks.cfg** kickstart file. This bug has been fixed, and **zerombr** is now correctly added to **anaconda-ks.cfg**. (BZ#1246663)

When using the `network` service, default routes are now correctly created on an installed system.

Previously, device-specific **GATEWAY** values were being included in the **/etc/sysconfig/network** configuration file, which applies to all devices. As a consequence, for some network configurations using the **network** service, default routes were not created. With this update, the **GATEWAY** parameter is no longer created in **/etc/sysconfig/network**, and default routes are now created correctly. (BZ#1181290)

The `DEFROUTE` option is now handled correctly when the installer generates a kickstart file.

Previously, if the **DEFROUTE** option was set in an **ifcfg** configuration file during installation, this was not reflected in the kickstart file subsequently generated by the installer. This bug has been fixed, and now the installer generates kickstart files which reflect **DEFROUTE** settings used during installation by setting the **--nodefroute** network command option accordingly. (BZ#1274686)

The `kdump` kernel is no longer added to `/etc/zipl.conf` when *kernel-kdump* is marked for installation

Previously, when installing *kernel-kdump*, an entry for the **kdump** kernel was added to the list of kernels in the **/etc/zipl.conf** configuration file. This bug is now fixed, and the **kdump** kernel is no longer added to the list. (BZ#1256211)

Chapter 10. Kernel

`/dev/disk/by-path/` now accounts for NPIV paths

Previously, if two or more virtual host bus adapters (HBAs) were created on a single physical HBA, only a single link to the device was created in the **`/dev/disk/by-path/`** directory instead of one link for each path. As a consequence, creating a **`virsh`** pool with virtual HBAs by using Fibre Channel N_Port ID Virtualization (NPIV) did not work correctly. With this update, symbolic links in **`/dev/disk/by-path/`** are created correctly and are unique. Symbolic links in **`/dev/disk/by-path/`** created by **`udev`** for logical unit numbers (LUNs) connected through a physical Fibre Channel N_Port stay the same. (BZ#1032218)

Removed unintended kernel warning message

A recent change in Red Hat Enterprise Linux 6.8 caused an unintended warning message to be displayed in certain situations where a file size is increased, such as by using **`fallocate`** operations:

```
WARNING: at mm/truncate.c:614 pagecache_isize_extended+0x10d/0x120()
```

This bug has been fixed, and operations which increase file size no longer cause this warning message to be displayed or logged. (BZ#1205014)

`librdmacm` no longer outputs warnings and errors if no RDMA hardware is present

Previously, if **`librdmacm`** was installed on a system with no RDMA hardware present, it could, in some circumstances, output superfluous warning and error messages to the standard error stream (stderr). With this update, **`librdmacm`** no longer outputs warning and error messages to stderr in such cases. (BZ#1231766)

Fixed kernel booting issues with the `mlx5` driver

When the **`mlx5`** driver was enabled on a system with non-fatal PCIe errors, the kernel previously failed to boot, crashing in the **`mlx5`** probe routine shortly after it enabled PCIe error handling. The patch causing this bug has been removed, and kernel now boots successfully when this driver is enabled. (BZ#1324599)

Changing snapshot read-only status no longer causes a kernel crash

Previously, the **`dm-snapshot`** target had improper handover of the exception store when the target was reloaded. As a consequence, when changing read-only status of the snapshot volume with **`lvchange -p r`** or **`lvchange -p rw`** commands and there was I/O to the origin volume in progress, the kernel crashed with the **`BUG()`** macro. With this update, the origin logical volume is suspended during exception store handover, so that there is no I/O in progress during the handover. As a result, changing snapshot read-only status no longer causes the aforementioned kernel crash. (BZ#1177389)

`qla2xxx` updated to version 8.07.00.26.06.8-k

The **`qla2xxx`** driver has been updated to version 8.07.00.26.06.8-k. This update backports initiator side upstream fixes and minor enhancements through 8.07.00.26. (BZ#1252111)

Memory leak in `devpts_kill_sb()` fixed

The **devpts** pseudo-file system allocates IDR resources during use. However, prior to this update, **devpts** did not free them when it was unmounted. Consequently, the resources use by the IDR system were leaked which could cause problems with frequent starting and stopping of containers, particularly with a high number of containers used. This update applies an upstream patch which releases these resources at unmount, and the IDR resources used by the **devpts** file system are no longer leaked at unmount. (BZ#1283557)

Setting a `sysctl` parameter now executes successfully

While executing the **sysctl -w vm.compact_memory=1** command to set a **sysctl** parameter, the system previously returned the following error messages:

```
error: "Success" setting key "vm.compact_memory"
```

The provided patch fixes this bug, and the aforementioned command now executes successfully. (BZ#1278842)

`netconsole` no longer causes kernel crash

Resetting an **ixgbe** or **vmxnet3** adapter while sending a message over **netconsole** or **netpoll** at the same time could previously cause a kernel crash. This update adds mutual exclusion between the core adapter reset path and netpoll transmit path, preventing kernel crashes in this situation. (BZ#1252212)

Loop checks added to VFS to prevent kernel crashes

The NFS client was previously failing to detect a directory loop for some NFS server directory structures. This failure could cause NFS inodes to remain referenced after attempting to unmount the file system, leading to a kernel crash. This update adds loop checks to VFS, which effectively prevents this problem from occurring. (BZ#1254020)

Playing audio from a USB sound card works as expected

Due to incorrect **URB_ISO_ASAP** semantics, playing an audio file using a USB sound card could previously fail for some hardware configurations. This update fixes the bug, and playing audio from a USB sound card now works as expected. (BZ#1255071)

Page fault and subsequent kernel oops in the HID driver fixed

Previously, when the Human Interface Device (HID) driver ran a report on an unaligned buffer, it could cause a page fault interrupt and a kernel oops when the end of the report was read. This update fixes this bug by padding the end of the report with extra bytes, so the reading of the report never crosses a page boundary. As a result, the page fault and subsequent kernel oops no longer occur. (BZ#1256568)

Fixed a deadlock when syncing a frozen file system

Due to broken **s_umount** lock ordering, a race condition occurred when an unlinked file was closed and the **sync** (or **syncfs**) utility was run at the same time. As a consequence, a deadlock occurred on a frozen file system between **sync** and a process trying to unfreeze the file system. With this update, **sync** (or **syncfs**) is skipped on frozen file systems, and deadlock no longer occurs in the aforementioned situation. (BZ#1241791)

`dracut` dependencies updated to prevent boot failures

The Deterministic Random Bit Generator (DRBG) module must be loaded during boot before cryptographic ciphers can be used. However, older versions of **dracut** did not include DRBG in the **initramfs** image which could use cryptographic ciphers for disk encryption. As a consequence, if disk encryption was in use on the root file system, the boot process failed. This update adds the DRBG module into the dependency list of **dracut**, ensuring that the module is present in the **initramfs**, and systems with encrypted root file systems can now boot successfully. (BZ#1241338)

Packets are now counted correctly

Due to a regression, packets counter detected only the number of normally processed completions (packets), but failed to detect erroneous ones. As these packets were thus never acknowledged, the firmware kept returning interrupt requests (IRQs). A patch has been provided to fix this bug, and all packets are now counted as expected. (BZ#1241287)

Fixed a deadlock when removing directories

When removing a directory while a reference was held to that directory by a reference to a negative child dentry, the directory dentry was previously not killed. In addition, once the negative child dentry was killed, an unlinked and unused dentry was still present in the cache. This could cause a deadlock by forcing dentry eviction while the file system in question was frozen. With this update, all unused dentries are unhashed and evicted immediately after a successful directory removal, which avoids the deadlock, and the system no longer hangs in the aforementioned scenario. (BZ#1241030)

Mapping `hugetlb` areas no longer causes data corruption

Inside **hugetlb**, region data structures were protected by a combination of a memory map semaphore and a single **hugetlb** instance mutex. However, a page-fault scalability improvement backported to the kernel in a previous release removed the single mutex and introduced a new mutex table, making the locking combination insufficient and leading to possible race windows that could cause corruption and undefined behavior. The problem could be observed for example when software mapping or remapping **hugetlb** areas with concurrent threads reading or writing to same areas, which caused page faults. This update fixes the problem by introducing a required **spinlock** to the region tracking functions for proper serialization. (BZ#1260755)

multipath request queue no longer causes stalls

Previously, running the multipath request queue caused regressions in cases where paths failed regularly under I/O load. This regression manifested as I/O stalls that exceeded 300 seconds. This update reverts the changes aimed to reduce running the multipath request queue, resulting in I/O completing in a timely manner. (BZ#1240767)

inodes are now freed as intended

Previously, when opening a file by its file handle (**fhandle**) with its **dentry** not present in the **dcache** ('cold dcache'), and then making use of the **unlink()** and **close()** functions, the inode was not freed upon the **close()** system call. As a consequence, the **iput()** final was delayed indefinitely. A patch has been provided to fix this bug, and the inode is now freed as expected. (BZ#1236736)

The `vmxnet3` driver is now compatible with the `vmxnet3` adapter version 2

Due to a bug, the **vmxnet3** driver demonstrated incorrect behavior such as memory leaks or 'screaming interrupts' when in use with **vmxnet3** adapter version 2. Several upstream patches have been applied to fix the behavior of the **vmxnet3** driver - namely, this update fixes memory leaks in the **rx** path, implements a handler for PCI shutdown, and makes **vmxnet3** compatible with adapter version 2. (BZ#1236564)

IP fragments are discarded in time

The memory used by the defragmentation engine is accounted for per CPU. However, on systems with numerous CPUs, the per-CPU caches could deviate from reality, thus causing the defragmentation engine to discard old fragments too early. This update adds a fix to minimize this discrepancy, and old IP fragments are now discarded at the correct time. (BZ#1235465)

GFS2 now references correct value

The GFS2 file system previously had a rare timing window that sometimes caused it to reference an uninitialized variable. Consequently, a kernel panic occurred. The code has been changed to reference the correct value during this timing window, and the kernel no longer panics. (BZ#1267995)

Software using IPC SysV semaphores works with kernel correctly

At a process or thread exit, when the Linux kernel undoes any SysV semaphore operations done previously (ones done using **semop** with the **SEM_UNDO** flag), there was a possible race condition with another process or thread removing the same semaphore set where the operations occurred, leading to a possible use of in-kernel-freed memory and then to possible unpredictable behaviour. This bug could be noticed with software which uses IPC SysV semaphores, such as **IBM DB2**, which could in certain cases have some of its processes or utilities get incorrectly stalled in an IPC semaphore operation or system call after the race condition happened. A patch has been provided to fix this bug, and the kernel now behaves as expected in the aforementioned scenario. (BZ#1233300)

Fixed a race condition in `perf buildid-cache`

Prior to this update, multiple instances trying to copy the same file triggered a race condition in **perf buildid-cache** that could truncate system libraries and other files. With this update, unique temporary files are used when copying to the **buildid** directory to prevent the aforementioned race condition from occurring. (BZ#1229673)

Cache serialization has been added to prevent kernel crashes

Due to a race condition whereby a cache operation could be submitted after a cache object was killed, the kernel occasionally crashed on systems running the **cachefilesd** service. The provided patch prevents the race condition by adding serialization in the code that makes the object unavailable. As a result, all subsequent operations on the object are rejected and the kernel no longer crashes in this scenario. (BZ#1096893)

Reloading or removing `edac` modules now works as expected

Previously, reloading or removing **edac** modules on a system using the **i7core_edac** module could lead to a number of warning messages to be returned and a subsequent kernel crash. The underlying source code has been patched, and the kernel no longer crashes when operating with **edac** modules. (BZ#1227845)

Custom MAC addresses can be specified again for bond interfaces

On a system with a bonded interface, the user could not specify their own custom MAC address for the bond. A patch has been provided to fix this bug, and custom MAC addresses can be specified again in the aforementioned situation. (BZ#1225359)

The `st` and `sg` drivers now work correctly

Due to the incorrect length for the **`FCP_RSP_INFO`** field, parts of the field could be copied, and the **`st`** and **`sg`** drivers thus did not work correctly. With this update, the code related to the FCP protocol has been updated, and **`st`** and **`sg`** now work as expected. (BZ#1223105)

Slave interfaces turn into promiscuous mode automatically

If a bonding VLAN interface turned into promiscuous mode while it was inactive, the slave interfaces previously did not turn into promiscuous mode automatically even after the bonding VLAN interface became active again. With this update, flag changes are always propagated to interfaces, and slave interfaces thus enter promiscuous mode as expected. (BZ#1222823)

`force_hrtimer_reprogram` parameter added to kernel

Due to a timer expiry issue, the scheduler tick previously stopped for too long when the **`ksoftirqd`** daemon for **`hrtimer`** was blocked by a running process. This update adds the **`force_hrtimer_reprogram`** kernel parameter. If **`force_hrtimer_reprogram=1`** is used on the kernel command line, the reprogramming of all expired timers is forced, which prevents this bug from occuring. (BZ#1285142)

`ipr` memory buffer indexing updated

A bug in the **`ipr`** driver on 64-bit IBM Power Systems (ppc64) could result in backwards memory buffer indexing and cause a kernel crash when running the Hardware Test Exerciser (HTX) test suite. With this update, **`ipr`** memory buffer indexing uses a bit mask operation instead of modulo, causing low bits to be masked off so that no backwards indexing is possible, and preventing the crash. (BZ#1209543)

`cgroup_threadgroup_rwsem` variable added to kernel

Previously, the **`attach_task_by_pid()`** function in some cases raced with an exiting thread and tried to lock or unlock the already freed **`group_rwsem`** member of the **`signal_struct`** list. As a consequence, a kernel crash could occur. This update adds the **`cgroup_threadgroup_rwsem`** variable, which fixes this bug and prevents the kernel crash from occurring in this scenario. (BZ#1198732)

Adding keys into a revoked keyring no longer causes a memory leak

Attempting to use the **`request_key()`** function to add a key into a revoked keyring was previously causing a resource leak in the kernel error path. Keys which were allocated and then failed became stuck in kernel memory and were impossible for the garbage collector to remove. With this update, the reference count on failed keys will now correctly reach 0 in this situation, allowing the garbage collector to remove them so that failed keys will no longer stay in memory indefinitely. (BZ#1188442)

Kernel panic caused by repeated fork() no longer occurs

Previously, an unusual forking pattern could cause the **`anon_vma_chain`** and **`anon_vma slab`** memory to grow infinitely even though the number of processes involved stayed low. As a consequence, a kernel panic occurred. The provided patch adds a heuristic which reuses existing **`anon_vma`** instead of forking a new one and adds the **`anon_vma->degree`** counter which makes sure the count of **`anon_vma`** members is not bigger

than twice the count of virtual memory areas. As a result, the kernel panic no longer occurs in this situation. (BZ#1151823)

Fixed job scheduling now ensures balanced CPU load

Due to prematurely decremented **calc_load_task**, the calculated load average was off by up to the number of CPUs in the machine. As a consequence, job scheduling worked improperly causing a drop in the system performance. This update keeps the delta of the CPU going into **NO_HZ** idle separately, and folds the pending idle delta into the global active count while correctly aging the averages for the idle-duration when leaving **NO_HZ** mode. Now, job scheduling works correctly, ensuring balanced CPU load. (BZ#1167755)

Only single processe can free specific memory page

A race condition was found in hash table invalidation code between inode invalidation and inode clearing code in the GFS2 file system. In some circumstances, two processes could attempt to free the same memory, resulting in a kernel panic. This update adds a **spin_lock** to the hash table invalidation code allowing only a single process to attempt to free a specific memory page, which prevents the race condition from occurring. (BZ#1250663)

macvtap transfers VLAN packets over be2net successfully

Previously, VLAN stacked on the **macvlan** or **macvtap** device did not work for devices that implement and use VLAN filters. As a consequence, **macvtap** passthrough mode failed to transfer VLAN packets over the **be2net** driver. This update implements VLAN **ndo** calls to the **macvlan** driver to pass appropriate VLAN tag IDs to lower devices. As a result, **macvtap** transfers VLAN packets over **be2net** successfully. (BZ#1213846)

primary_reselect=failure now works properly

A bug caused the **primary_reselect=failure** bond parameter to work incorrectly. The primary interface was always taking over even if others did not fail. With this update, the parameter works as expected, and the primary bond interface only takes over if the current non-primary active interface fails. (BZ#1290672)

Log messages from logshifter are now processed correctly

Under significant load, some applications such as logshifter could generate bursts of log messages too large for the system logger to spool. Due to a race condition, log messages from that application could then be lost even after the log volume dropped to manageable levels. This update fixes the kernel mechanism used to notify the transmitter end of the socket used by the system logger that more space is available on the receiver side, removing a race condition which previously caused the sender to stop transmitting new messages and allowing all log messages to be processed correctly. (BZ#1284900)

KVM virtual guests now connect via a bridged interface successfully

Previously, a bridge interface could exist on top of a bonded interface which was above a physical interface with the large receive offload (LRO) flag still on. Bridge interfaces are incompatible with LRO enabled on any underlying devices, which caused network communications on the bridge, such as that from a Virtual Machine (VM) to fail to function properly. This update makes sure devices underneath a bridge all get LRO disabled, and a VM now connects via a bridged interface successfully. (BZ#1258446)

SwapFree size is now correct

A previous change in the get_swap_page() locking removed the use of the **swap_lock** spinlock. This could cause **nr_swap_pages** corruption and invalid **SwapFree** information in the **/proc/meminfo** file, where the

size of **SwapFree** could exceed the size of **SwapTotal**. This update uses an atomic variable for **nr_swap_pages**, and the size of **SwapFree** in **/proc/meminfo** is now correct. (BZ#1252362)

SCSI error handling no longer causes deadlocks

Previously, when a SCSI command timed out on a removable media device, the error handling code always attempted to re-lock the door of the device. This could cause a deadlock because the request to issue a command to re-lock the door could not be allocated if all requests were in use. With this update, SCSI error handling only attempts to re-lock if the device was reset as part of the error handling procedure, and the deadlock no longer occurs. (BZ#995234)

LRO flags now propagate correctly

Large Receive Offload (LRO) flag disabling was not being propagated downwards from above devices in the VLAN and bond hierarchy, breaking the flow of traffic. This bug has been fixed and LRO flags now propagate correctly. (BZ#1259008)

multicast group assignments fixed

The kernel was incorrectly assigning multicast groups for the **nl80211** protocol, causing problems with **nl80211** wireless drivers, for example, preventing **hostapd** from starting and initializing wireless devices in Access Point mode. This update fixes multicast group assignments for **nl80211** and allows wireless devices to be managed correctly. (BZ#1259870)

Sending a UDP datagram over IPv6 works as expected

Due to a race condition, an **ipv6_txoptions** corruption previously appeared when sending a UDP datagram over the IPv6 protocol. An upstream patch has been applied to prevent data corruption that led to the kernel panic. (BZ#1312740)

nvme hard-lockup panic no longer occurs

When the the **nvme** driver held the queue lock for too long, for example during DMA mapping, a lockup occurred leading to the **nvme** hard-lockup panic. This update fixes the underlying source code, and **nvme** now works as expected. (BZ#1227342)

BUG_ON() in fs_clear_inode() no longer occurs

Previously, the BUG_ON() signal appeared in the fs_clear_inode() function where the nfs_have_writebacks() function reported a positive value for nfs_inode->npages. As a consequence, a kernel panic occurred. The provided patch performs a serialization by holding the inode i_lock over the check of PagePrivate and locking the request, which fixes this bug. (BZ#1135601)

UID and GID are assigned correct values

Due to a regression, the UID and GID environment variables were not assigned correct values during autofs mount requests. This update provides a patch that fixes the UID and GID assignment so that UID and GID now take on the value of the user that has triggered the mount. (BZ#1248820)

Using LUKS and IPSEC simultaneously no longer leads to data corruption

When using IPSEC and a LUKS-encrypted volume simultaneously, data corruption on a LUKS volume could occur. The provided patch fixes this bug, and data corruption no longer occurs when using LUKS and IPSEC simultaneously. (BZ#1259023)

VLAN_GROUP_ARRAY_LEN has been revived

In a previous update, the VLAN_GROUP_ARRAY_LEN kernel macro was renamed to VLAN_N_VID. Due to this rename, when compiling a kernel module requiring VLAN_GROUP_ARRAY_LEN, for example the vmxnet3 external driver, the compilation failed. With this update, the old macro has been revived so that the third party modules succeed to compile. (BZ#1242145)

Corrupted ELF header has been fixed

Previously, the corrupted ELF header of the **`/proc/vmcore`** ELF file caused that the ELF file could not be read correctly. As a consequence, the **`kdump`** service terminated unexpectedly, resulting in a kernel panic. The provided patch fixes the ELF header, and **`kdump`** now succeeds as expected. (BZ#1236437)

Quota warning deadlocks on tty mutex have been fixed

Previously, the quota code could call into the tty layer to print a warning, which could cause a lock inversion between tty->atomic_write_lock and dqptr_sem. The provided patch prevents the quota utility code from calling the tty layer with dqptr_sem semaphore held, and processes no longer end up in a deadlock. (BZ#1232387)

anon_vma degree is always decremented when the VMA list is empty

In the **`anon_vma`** data structure, the degree counts the number of child **`anon_vma`** members and of virtual memory areas that point to this **`anon_vma`**. In the unlink_anon_vma() function, when its list is empty, **`anon_vma`** is going to be freed whether the external reference count is zero or not, so the parent's degree should be decremented. However, failure to decrement the degree triggered a BUG_ON() signal in unlink_anon_vma(). The provided patch fixes this bug, and the degree is now decremented as expected. (BZ#1309898)

Repeated sysrq events proceed as expected

Previously, repeated **`sysrq`** events in an NMI context could cause a deadlock, leading to a system crash. The provided patchset adds minimal support for the **`seq_buf`** buffer and a per_cpu printk() function, which prevents the aforementioned deadlock from occurring. (BZ#1104266)

Unix domain datagram socket no longer experiences deadlock

Due to a regression, a Unix domain datagram socket could come to a deadlock when sending a datagram to itself. The provided patch adds another sk check to the unix_dgram_sendmsg() function, and the aforementioned deadlock no longer occurs. (BZ#1309241)

Exiting process decrements a counter as expected

Previously, when Kernel Shared Memory (KSM) or page migration were in use, an exiting process could fail to decrement a counter related to anonymous virtual memory areas. As a consequence, the counter unbalance triggered a kernel panic. The provided patch fixes this bug, and the kernel panic no longer occurs in the aforementioned scenario. (BZ#1126228)

VGA output speed in UEFI boot mode improved

Previously, the VGA console was very slow in UEFI boot mode, which resulted in a large difference in boot time for servers with many CPUs or I/O devices. As a consequence, printing large amount of debug output during the boot phase was extremely slow, making it difficult to analyze issues that occur during boot time. In addition, the VGA output slowdown continued during OS runtime, which could lead to a system hang. The provided fix improves the VGA output speed in UEFI boot mode, preventing the aforementioned problems. (BZ#1290686)

ndo_set_multicast_list field is again present in network drivers

When creating a VLAN interface on top of a **`netxen_nic`** physical interface after changing its MAC address, **`ping`** over VLAN to a remote VLAN previously failed. The provided patch adds back the use of the **`ndo_set_multicast_list`** field in network drivers, and the ping now succeeds as expected. (BZ#1213207)

fio no longer corrupts XFS

After adjusting the extent size with the **`xfs_fio`** utility and running the **`fio`** tool with the configuration file provided, the XFS file system previously became corrupted. The provided patch extends the size hints, and **`fio`** no longer corrupts XFS. (BZ#1211110)

NFS mount now reports correctly

When configuring the firewall on the NFS server to reject all the packets of 2049 and mounting the share on the NFS client, the following error was returned:

```
connection timed out
```

The provided fix corrects the error message, which now reads:

```
connection refused
```

(BZ#1206555)

Automatic signing is now enabled

When setting a security type with the **`sec=`** mount option and no signing had been specified with the trailing **`i`**, automatic signing was not previously enabled. For example, in DFS mounts where the DFS node requires signing but the client had disabled it using **`sec=`**, the user could not mount the DFS node if the node required signing to be enabled. The provided fix sets **`MAY_SIGN`** flags for all security types, thus fixing this bug. (BZ#1197875)

Writing a large file using direct I/O now proceeds successfully

Previously, writing a large file using direct I/O in 16 MB chunks sometimes caused a pathological allocation pattern where 16 MB chunks of large free extent were allocated to a file in a reversed order. The provided patch avoids the backward allocation, and writing a large file using direct I/O now proceeds successfully. (BZ#1302777)

Fix for shrinker return value prevents system hang

The **shrink_dcache_memory** shrinker is prone to overflow, reporting the following line in the log:

```
negative objects to delete
```

As a consequence, the system previously hung. The provided patch tests for this overflow sign extension from any shrinker return value, and refuses to set the **max_pass** variable larger than the **INT_MAX** preprocessor macro. As a result, the aforementioned hang no longer occurs. (BZ#1159675)

perf has been updated

To support a greater range of hardware and incorporate numerous bug fixes, **perf** has been updated. Notable enhancements include:

- Added support for additional model numbers of 5th Generation Intel Core i7 processors.
- Added support for Intel Xeon v5 mobile and desktop processors.
- Enabled support for the uncore subsystem for Intel Xeon v3 and v4 processors.
- Enabled support for the uncore subsystem for Intel Xeon Processor D-1500. (BZ#1189317)

Configuring settings for multiple WWPNs is now easier

This enhancement update adds support for **tag** and **untag** commands in targetcli. Instead of configuring LUN mapping using the numeric WWPN, for example **20:00:00:1b:21:59:12:36**, it is now possible to give one or more WWPNs a descriptive name with the **tag** command, and then use the tag to configure LUN mappings. See **help tag** and **help untag** commands within the **acls** configuration node for more information. (BZ#882092)

Systems with `iscsi_firmware` are able to boot

A previous regression in dracut caused systems with iSCSI offloading or iSCSI Boot Firmware Table (iBFT) to stop booting in some cases. Consequently, freshly installed Red Hat Enterprise Linux 6.8 systems with **iscsi_firmware** on the kernel command line could be unable to boot. This update fixes the bug, and systems in the described scenario are able to boot as expected. (BZ#1322209)

Chapter 11. Networking

`logrotate` now correctly works with `wpa_supplicant`

Previously, **wpa_supplicant** did not correctly truncate the log file when the **logrotate** script attempted to rotate it. This bug has been fixed and **logrotate** now correctly coordinates log rotation with **wpa_supplicant**. (BZ#908306)

Bug fixes in *system-config-network*

This release brings multiple bug fixes to the Network Configuration tool (*system-config-network*). Notable fixes include:

- Previously, when **system-config-network** was used to change the system host name, the new host name was appended to the **/etc/hosts** file every time, even if the same host name was previously used. This could cause the **/etc/hosts** file to be unnecessarily cluttered. With this update, new host names are only appended if they have not been used previously.
- A bug preventing suppression of DNS settings has been fixed and you can now suppress DNS settings by leaving the DNS field empty.
- In some circumstances, **system-config-network** could display text messages in the text-based interface before the text framework was properly cleaned, resulting in the message being distorted. This bug has been fixed and text messages from this tool now display correctly. (BZ#1086282)

`NetworkManager` no longer brings down connections when saving a configuration file in `vim`

Previously, editing network connection configuration files in editors which save files by deleting and recreating them (such as **vim**) caused **NetworkManager** to bring down the edited connection if it was active at the time. This bug has been fixed and active connections can now be safely edited in any text editor. (BZ#1272617)

Bond devices not created by `NetworkManager` now work correctly

Previously, bond devices named **bond0**, which created when the **bonding** module was loaded and not by **NetworkManager**, were incorrectly configured if the **network** service was disabled. This bug has been fixed and bond devices now work correctly with **NetworkManager**. (BZ#1292502)

`NetworkManager` no longer ignores the DHCP-provided list of search domains

Previously, **NetworkManager** used the host's DNS domain suffix to configure the DNS resolver (**/etc/resolv.conf**), and ignored the list of search domain supplied by DHCP. This bug has been fixed and **NetworkManager** now correctly configures the DNS resolver using DHCP. (BZ#1202539)

`NetworkManager` can now distinguish between software and hardware devices with the same hadware address

Previously, **NetworkManager** ignored connections for software devices such as bonds and bridges if the underlying hardware devices used the same hardware address (the **HWADDR** key) and used the **NM_CONTROLLED=no** setting. This bug has been fixed and **NetworkManager** now works with such devices correctly. (BZ#902907)

Chapter 12. Security

Fixed ordering in the output of `semanage fcontext -l`

Previously, the **`semanage fcontext -l`** command did not print SELinux rules in the order in which the user added them and the **`restorecon`** utility used them. This could pose problems when managing SELinux rules, because the order in which the rules were displayed to the user did not match the order in which restorecon executed them. A patch has been applied to fix this problem, and **`semanage fcontext -l`** now displays the rules in the correct and expected order. (BZ#1206767)

Chapter 13. Servers and Services

Tomcat 6 starts as expected when the fr_FR language is configured

Previously, there was an incorrect entry in the Tomcat 6 **`LocalStrings_fr.properties`** file. As a consequence, Tomcat 6 showed and exception during the startup when the **`LANG`** variable in the **`/etc/tomcat6/tomcat6.conf`** file was set to **`fr_FR`**. This update fixes the entry and now Tomcat 6 starts without the exception. (BZ#1072484)

tomcat6 now provides noarch packages

Previous releases provided the *tomcat6* packages as architecture-dependent. However, the Tomcat 6 servlet container is a Java application without any native components. Therefore, this release provides the packages as architecture-independent. (BZ#1155509)

The Tomcat 6 NIO connector does not leak memory anymore

Previously, a memory leak sometimes occurred when using the Tomcat 6 Non-blocking I/O (NIO) connector. This update ensures that Tomcat 6 removes processors from the RequestGroupInfo list and returns them to the recycledProcessors queue. As a result, the NIO connector no longer leaks memory. (BZ#1268352)

mod_nss now supports changing the SSL renegotiation buffer size

This update adds the **`NSSRenegBufferSize`** parameter to the *mod_nss* package. The parameter allows users to configure the amount of memory to be used for buffering a POST request when a per-location SSL renegotiation is required. Previously, *mod_nss* did not support this functionality, which caused such requests to fail with the following message recorded in Apache logging:

```
request body exceeds maximum size for SSL buffer, could not buffer message
body to allow SSL renegotiation to proceed.
```

`NSSRenegBufferSize` accepts buffer size in bytes. The default value is 128K. Setting **`NSSRenegBufferSize`** to **`0`** disables the buffering. (BZ#1214366)

Documentation for `tcp_wrappers` no longer refers to unavailable binaries

The **`hosts_access(5)`** man page, which is a part of the *tcp_wrappers* package, previously referred to **`tcpdchk`** and **`tcpdmatch`** binaries which were not included in this package, causing confusion. References to **`tcpdchk`** have been removed from the man page, and a modified version of **`tcpdmatch`** has been added to the package, allowing you to test your configurations by following the provided instructions. (BZ#1084458)

`openssh-clients` no longer keeps exited sessions open

Previously, the implementation of openssh-clients did not adhere to RFC 4253, The Secure Shell (SSH) Transport Layer Protocol, as in some cases, a language tag was not sent for the **`SSH_MSG_DISCONNECT`** message. As a consequence, when connected to the server from a Red Hat Enterprise Linux 6 **`ssh-client`** and disconnected by closing the session, the server kept the session (TCP socket) open until it timed out. This bug has been fixed by adding correct parameters for the **`SSH_MSG_DISCONNECT`** message, which makes the server close the session as expected. (BZ#1222500)

Pegasus CIM server now disables SSLv3 and uses TLS1.0 or later by default

The Pegasus CIM server previously had no option to disable the SSLv3 protocol, which is now considered insecure. This update contains a backported upstream fix which changes the default behavior so that SSLv3 is disabled, TLS1.0 or later is used, and SSLv3 can be reenabled using the **sslBackwardCompatibility** option if necessary. (BZ#1238329)

vsftpd can now use wildcards in commands correctly

A regression in the **vsftpd** daemon previously caused commands which used wildcards such as ***** or **?** to fail. This bug has been fixed and you can now use wildcards in commands such as **ls** with **vsftpd** again. (BZ#1315957)

Print jobs no longer disappear from cups queue for non-responsive printers

Previously, when a print job was submitted to a print queue which was trying to send jobs to a non-responsive printer, and then the queue was disabled and reenabled, the print job disappeared due to a bug in the **cups** service. An upstream fix was backported into **cups**, and jobs no longer disappear from queues when they are disabled and reenabled. (BZ#1293498)

The Dovecot IMAP server now returns the CP932 character in IMAP search results

A bug in the charset conversion algorithm caused IMAP searches not to return messages that contained the CP932 character. An upstream fix has been backported to fix this bug, and the IMAP search command na finds messages containing the CP932 character as expected. (BZ#1275233)

Applications no longer access database files on a NFS share ineffectively

Prior to this update, some applications performed poorly when performing operations on database files hosted on a NFS share. This was caused by the frequent invalidations of cache on the NFS client. This update introduces a new environment variable **NDBM_LOCK**, which prevents cache invalidation. As a result, the relevant applications no longer perform poorly in the described scenario. (BZ#668702)

Chapter 14. Storage

`rescan-scsi-bus.sh` now correctly interprets multiple word device descriptions

The **`rescan-scsi-bus.sh`** script, found in the *sg3_utils* package, previously misinterpreted SCSI device types that were described using more than one word, such as **`Medium Changer`** or **`Optical Device`**. Consequently, when the script was run on systems that had such device types attached, the script printed multiple misleading error messages. With this update, device types described with multiple words are handled correctly, and the proper device type description is returned to the user without any errors. (BZ#1210438)

`rescan-scsi-bus.sh` no longer removes `/dev/null`

When running the **`rescan-scsi-bus.sh`** script, due to incorrect syntax in redirecting output to the **`/dev/null`** device file while executing the **`/bin/rm`** utility, the redirection did not happen but **`/dev/null`** was instead interpreted as a file to be removed. As a consequence, running **`rescan-scsi-bus.sh`** with the **`--update`** option removed **`/dev/null`** during execution. This bug has been fixed, and /dev/null is no longer removed by **`rescan-scsi-bus.sh`**. (BZ#1245302)

Additional result codes are now recognized by `sg_persist`

Previously, some SCSI hosts could return result codes which were not recognized by **`sg_persist`**, causing it to output an error message claiming the result code is invalid. This update adds additional return codes, such as **`DID_NEXUS_FAILURE`**, and the problem no longer occurs. (BZ#886611)

iSCSI boot works correctly in Multi Function mode

Due to incorrect handling of Multi Function mode when dealing with the bnx2x driver, booting iSCSI from Storage Area Network (SAN) did not work correctly for some Host Bus Adapters (HBAs). The underlying source code has been fixed, and iSCSI boot now works correctly in Multi Function mode. (BZ#1276545)

Chapter 15. System and Subscription Management

`iostat` can now print device names longer than 72 characters

Previously, device names longer than 72 characters were being truncated in **`iostat`** output because the device name field was too short. The allocated space for device names has been increased, and **`iostat`** can now print significantly longer device names in the output. (BZ#1308862)

Corrupted data files no longer crash `sar`

Previously, the **`sar`** command could crash when loading a corrupted system activity data due to **`localtime()`** function calls not being properly checked in **`sysstat`**. This bug has been fixed and corrupted system activity data files no longer crash **`sar`**. (BZ#887231)

`pidstat` no longer outputs values above 100% for certain fields

Previously, **`pidstat`** could potentially run out of preallocated space for PIDs on systems with many short-lived processes. This could cause **`pidstat`** to output nonsensical values (values larger than 100%) in the **`%CPU`**, **`%user`**, and **`%sys`** fields. With this update, **`pidstat`** now automatically reallocates space for PIDs, and outputs correct values for all fields. (BZ#1224878)

`curl` no longer requires both private and public SSH keys

Previously, the **`curl`** tool required a full pair of a private and a public SSH keys for user authentication. If you only provided a private SSH key, which is common when using certain tools such as **`scp`**, user authentication failed. An upstream patch has been applied on **`curl`** source code to improve SSH user authentication so that the public key does not need to be specified, and **`curl`** can now authenticate using only a private SSH key. (BZ#1260742)

`NSS` no longer reuses TLS sessions for servers with different host names

Previously, Network Security Services (NSS) could incorrectly reuse an existing TLS session to connect to a server with a different host name. This caused some HTTPS servers to refuse requests made within that session and to respond with HTTP code 400 (**`Bad Request`**). A patch which prevents reusing TLS sessions for different servers has been applied to **`libcurl`** source code, allowing NSS to successfully communicate with servers which require the HTTP host name to match the TLS session host name. (BZ#1269660)

Fixed a memory leak in `libcurl`

DNS cache implementation in **`libcurl`** could previously fail to remove cache entries which were no longer used. This resulted in a memory leak in applications using this library while resolving host names. This bug has been fixed, and libcurl-based applications no longer leak memory while resolving host names. (BZ#1302893)

Enhancements to `abrt` reporting workflow

The problem reporting workflow in **`abrt`** has been enhanced to improve the overall crash reporting experience and customer case creation. The enhancements include:

- The **Provide additional information** screen now allows you to select whether the problem happens repeatedly, and also contains an additional input field for providing steps to reproduce the problem.
- A new reporting workflow **Submit anonymous report**, which should be used when the reported problem is not critical and no Red Hat support team assistance is required.
- New tests have been added to the internal logic to should ensure that users open cases only for critical problems and software released by Red Hat.

Additionally, the client identifier has been updated to **abrt_version: 2.0.8.1**. (BZ#1258474)

pmap no longer reports incorrect totals

With the introduction of **VmFlags** in the kernel **smaps** interface, the **pmap** tool could no longer reliably process the content due to format differences of the **VmFlags** entry. As a consequence, **pmap** reported incorrect totals. The underlying source code has been patched, and **pmap** now works as expected. (BZ#1262870)

Fixes in free output

With the introduction of the human readable ("-h") switch in the **free** tool, the layout generator had to be modified to support the new feature. This, however, affected printing of values longer than the column width. The values were truncated to prevent the layout from breaking when the values became longer than the reserved space in the columns. At the same time, the change caused **free** to insert an unwanted space character at the end of each line. Due to these two changes, the output could not be used in custom scripts. With this update, values longer than the column width are no longer truncated, no extra spaces are inserted at line ends, and the output of the **free** tool can now be processed without problems. (BZ#1246379)

Fixed a race condition when processing of detected problems in abrtd

This update fixes a race condition in the **abrtd** service which was causing a loss of detected problem data, filling system logs with repeated error messages, and causing **abrt** core dumper processes to hang, which in turn prevented dumped programs from being restarted. (BZ#1245893)

Chapter 16. Virtualization

Hyper-V guests work properly with VHDX files

Previously, when running Red Hat Enterprise Linux as a guest on a Microsoft Hyper-V hypervisor with a large dynamic Hyper-V virtual hard disk (VHDX) attached and using the ext3 file system, a call trace in some cases appeared and made it impossible to shut down the guest. With this update, Red Hat Enterprise Linux guests on Windows Hyper-V handle VHDX files correctly, and the described problem no longer occurs. (BZ#982542)

The `hv_netvsc` module works correctly with Hyper-V

Due to a race condition, the **hv_netvsc** module previously in some cases terminated unexpectedly when it was unloading. This caused a kernel crash on Red Hat Enterprise Linux guests running on the Microsoft Hyper-V hypervisor. The race condition has been removed, which prevents the described kernel crashes from occurring. (BZ#1118163)

Guests shut down correctly when processing interrupts

Prior to this update, if processes that generate interrupts were active during the guest shut down sequence, the virtio driver in some cases did not correctly clear the interrupts. As a consequence, the guest kernel became unresponsive, which prevented the shut down from completing. With this update, the virtio driver processes interrupts more effectively, and guests now shut down reliably in the described scenario. (BZ#1199155)

Consistent save times for taking guest snapshots

Prior to this update, saving a KVM guest snapshot involved overwriting the state of the virtual machine using copy-on-write operations. As a consequence, taking every snapshot after the first one took an excessive amount of time. Now, the guest state written in the active layer is discarded after the snapshot is taken, which avoids the need for copy-on-write operations. As a result, saving subsequent snapshots is now as quick as saving the first one. (BZ#1219908)

The `at` program works correctly with `virt-sysprep`

When using the **virt-sysprep** utility to create a Red Hat Enterprise Linux guest template, the **at** program in the resulting guest could not be used. This update ensures that **virt-sysprep** does not delete **/var/spool/at/.SEQ** files in these guests, and **at** now works as expected. (BZ#1229305)

Failed logical volume creation no longer deletes existing volumes

Previously, when attempting to create a logical volume in a logical-volume pool that already contained a logical volume with the specified name, libvirt in some cases deleted the existing logical volume. This update adds more checks to determine the cause of failure when creating logical volumes, which prevents libvirt from incorrectly removing existing logical volumes in the described circumstances. (BZ#1232170)

Domain information from `LIBVIRT-MIB.txt` is loaded correctly

Previously, the **LIBVIRT-MIB.txt** file in the *libvirt-snmp* package did not fully comply with the formatting rules of the Simple Network Management Protocol (SNMP). As a consequence, SNMP software could not load the file and thus failed to read the domain information it provides, such as exposed variables, their ranges, or certain named values. This update ensures that **LIBVIRT-MIB.txt** is fully compliant with SNMP formatting rules, and the file is now loaded as expected. (BZ#1242320)

System log is no longer flooded with error messages about missing metadata

Prior to this update, the libvirt library was logging the **VIR_ERR_NO_DOMAIN_METADATA** error code with the **error** priority, rather than the 'debug' severity usual for this kind of message. As a consequence, if the metadata APIs were used heavily while metadata entries were missing, the system log was flooded with irrelevant messages. With this update, the severity of **VIR_ERR_NO_DOMAIN_METADATA** has been lowered to **debug**, thus fixing this problem. (BZ#1260864)

Guests with strict NUMA pinning boot more reliably

When starting a virtual machine configured with strict Non-Uniform Memory Access (NUMA) pinning, the KVM module could not allocate memory from the Direct Memory Access (DMA) zones if the NUMA nodes were not included in the configured limits set by the libvirt daemon. This led to a Quick Emulator (QEMU) process failure, which in turn prevented the guest from booting. With this update, the cgroup limits are applied after the KVM allocates the memory, and the QEMU process, as well as the guest, now starts as expected. (BZ#1263263)

Kernel panics caused by `struct kvm` handling are fixed

When creating a KVM guest, the **struct kvm** data structure corresponding to the virtual machine was in some cases not handled properly. This caused corruption in the kernel memory and triggered a kernel panic on the host. Error conditions during guest creation are now treated properly, which prevents the described kernel panic from occurring. (BZ#1270791)

Limited KSM deduplication factor

Previously, the kernel same-page merging (KSM) deduplication factor was not explicitly limited, which caused Red Hat Enterprise Linux hosts to have performance problems or become unresponsive in case of high workloads. This update limits the KSM deduplication factor, and thus eliminates the described problems with virtual memory operations related to KSM pages. (BZ#1262294)

Hyper-V daemon services are no longer unavailable on slowly-booting Red Hat Enterprise Linux 6 guests

Prior to this update, if a Red Hat Enterprise Linux 6 guest running on a Hyper-V hypervisor took a long time to boot, the **hypervkvpd**, **hypervvssd**, and **hypervfcopy** Hyper-V daemons in some cases failed to start due to a negotiation timeout. As a consequence, the guest could not use the services provided by these daemons, including online backup, file copy, and network settings. This update ensures that the Hyper-V daemons start properly in the described scenario, which makes the affected services available as expected. (BZ#1216950)

Starting guests when using macvtap and Cisco VM-FEX no longer fails

Prior to this update, on hosts using macvtap connections to Cisco Virtual Machine Fabric Extender (VM-FEX) network cards, starting a virtual machine failed with the following error message:

```
internal error missing IFLA_VF_INFO in netlink response
```

This bug has been fixed, and starting guests on the described hosts now works as expected. (BZ#1251532)

Faster startup for `virt-manager` on hosts with many network interfaces

On hosts with very large numbers of bridged, VLAN, or bond interfaces, starting the **`virt-manager`** utility previously took a very long time. This update optimizes the **`netcf`** query that caused this delay, which significantly improves the start-up speed of **`virt-manager`** on the described systems. (BZ#1235959)

Part II. Technology Previews

This chapter provides a list of all available Technology Previews in Red Hat Enterprise Linux 6.8.

Technology Preview features are currently not supported under Red Hat Enterprise Linux subscription services, may not be functionally complete, and are generally not suitable for production use. However, these features are included as a customer convenience and to provide the feature with wider exposure.

Customers may find these features useful in a non-production environment. Customers are also free to provide feedback and functionality suggestions for a Technology Preview feature before it becomes fully supported. Errata will be provided for high-severity security issues.

During the development of a Technology Preview feature, additional components may become available to the public for testing. It is the intention of Red Hat clustering to fully support Technology Preview features in a future release.

For information about the Technology Preview features support scope, see https://access.redhat.com/support/offerings/techpreview/.

Chapter 17. Authentication and Interoperability

Apache Modules for External Authentication

A set of Apache modules was added to Red Hat Enterprise Linux 6.6 as a Technology Preview. The **mod_authnz_pam**, **mod_intercept_form_submit**, and **mod_lookup_identity** Apache modules in the respective packages can be used by Web applications to achieve tighter interaction with external authentication and identity sources, such as Identity Management in Red Hat Enterprise Linux.

Simultaneous maintaining of TGTs for multiple KDCs

Kerberos version 1.10 added a new cache storage type, DIR:, which allows Kerberos to maintain Ticket Granting Tickets (TGTs) for multiple Key Distribution Centers (KDCs) simultaneously and auto-select between them when negotiating with Kerberized resources. Red Hat Enterprise Linux 6.4 and later includes SSSD enhanced to allow the users to select the DIR: cache for users that are logging in using SSSD. This feature is introduced as a Technology Preview.

Package: *sssd-1.13.3*

Cross-Forest Kerberos Trust Functionality in Identity Management

The Cross-Forest Kerberos Trust functionality provided by Identity Management (IdM) is included as a Technology Preview. This feature allows to create a trust relationship between an IdM and an Active Directory (AD) domain. This means that users from the AD domain can access resources and services from the IdM domain with their AD credentials. No data needs to be synchronized between the IdM and AD domain controllers; AD user are always authenticated against the AD domain controller and information about users is looked up without the need for synchronization.

This feature is provided by the optional *ipa-server-trust-ad* package. This package depends on features which are only available in **samba4**. Because *samba4-** packages conflicts with the corresponding *samba-** packages, all *samba-** packages must be removed before *ipa-server-trust-ad* can be installed.

When the *ipa-server-trust-ad* package is installed, the **ipa-adtrust-install** utility must be run on all IdM servers and replicas to enable IdM to handle trusts. When this is done, a trust can be established from the command line using the **ipa trust-add** command or the IdM web UI. For more information, see the *Identity Management Guide* for Red Hat Enterprise Linux .

Note that Red Hat recommends to connect Red Hat Enterprise Linux 6 IdM clients to a Red Hat Enterprise Linux 7 IdM server for cross-forest trust capability. Trusts are fully supported on servers running Red Hat Enterprise Linux 7. Configuration with Red Hat Enterprise Linux 6 clients connected to a Red Hat Enterprise Linux 7 server for cross-forest trust is fully supported as well. In such setups, it is recommended to use the latest version of Red Hat Enterprise Linux 6 on the client side and the latest version of Red Hat Enterprise Linux 7 on the server side.

Packages: *ipa-3.0.0* and *samba-3.6.23*

Chapter 18. Compiler and Tools

System Information Gatherer and Reporter (SIGAR)

The System Information Gatherer and Reporter (SIGAR) is a library and command-line tool for accessing operating system and hardware level information across multiple platforms and programming languages. In Red Hat Enterprise Linux 6.4 and later, SIGAR is considered a Technology Preview package.

Package: *sigar-1.6.5-0.4.git58097d9*

Chapter 19. Clustering

clufter

The *clufter* package, available as a Technology Preview in Red Hat Enterprise Linux 6, provides a tool for transforming and analyzing cluster configuration formats. It can be used to assist with migration from an older stack configuration to a newer configuration that leverages Pacemaker. For information on the capabilities of **clufter**, see the **`clufter(1)`** man page or the output of the **`clufter -h`** command.

Package: *clufter-0.56.2-1*

luci support for fence_sanlock

The **luci** tool now supports the sanlock fence agent as a Technology Preview. The agent is available in the luci's list of agents.

Package: *luci-0.26.0-78*

Recovering a node using a hardware watchdog device

New fence_sanlock agent and checkquorum.wdmd, included in Red Hat Enterprise Linux 6.4 as a Technology Preview, provide new mechanisms to trigger the recovery of a node using a hardware watchdog device. Tutorials on how to enable this Technology Preview will be available at https://fedorahosted.org/cluster/wiki/HomePage

Note that SELinux in enforcing mode is currently not supported.

Package: *cluster-3.0.12.1-78*

Chapter 20. File Systems

FS-Cache

FS-Cache in Red Hat Enterprise Linux 6 enables networked file systems (for example, NFS) to have a persistent cache of data on the client machine.

Package: *cachefilesd-0.10.2-3*

Chapter 21. Kernel

Kernel Media support

The following features are presented as Technology Previews:

- The latest upstream video4linux
- Digital video broadcasting
- Primarily infrared remote control device support
- Various webcam support fixes and improvements

Package: *kernel-2.6.32-642*

Linux (NameSpace) Container [LXC]

Linux containers provide a flexible approach to application runtime containment on bare-metal systems without the need to fully virtualize the workload. Red Hat Enterprise Linux 6 provides application level containers to separate and control the application resource usage policies through cgroups and namespaces. This release includes basic management of container life-cycle by allowing creation, editing and deletion of containers using the **libvirt** API and the **virt-manager** GUI. Linux Containers are a Technology Preview.

Packages: *libvirt-0.10.2-60*, *virt-manager-0.9.0-29*

Diagnostic pulse for the fence_ipmilan agent, BZ#655764

A diagnostic pulse can now be issued on the IPMI interface using the **`fence_ipmilan`** agent. This new Technology Preview is used to force a kernel dump of a host if the host is configured to do so. Note that this feature is not a substitute for the **`off`** operation in a production cluster.

Package: *fence-agents-4.0.15-12*

Chapter 22. Networking

Mellanox SR-IOV Support

Single Root I/O Virtualization (SR-IOV) is now supported as a Technology Preview in the Mellanox **libmlx4** library and the following drivers:

- **mlx_core**
- **mlx4_ib** (InfiniBand protocol)
- **mlx_en** (Ethernet protocol)

Package: *kernel-2.6.32-642*

QFQ queuing discipline

In Red Hat Enterprise Linux 6, the **tc** utility has been updated to work with the Quick Fair Scheduler (QFQ) kernel features. Users can now take advantage of the new QFQ traffic queuing discipline from userspace. This feature is considered a Technology Preview.

Package: *kernel-2.6.32-642*

vios-proxy, BZ#721119

vios-proxy is a stream-socket proxy for providing connectivity between a client on a virtual guest and a server on a Hypervisor host. Communication occurs over virtio-serial links.

Package: *vios-proxy-0.2-1*

Chapter 23. Security

TPM

TPM (Trusted Platform Module) hardware can create, store and use RSA keys securely (without ever being exposed in memory), verify a platform's software state using cryptographic hashes and more. The *trousers* and *tpm-tools* packages are considered a Technology Preview.

Packages: *trousers-0.3.13-2*, *tpm-tools-1.3.4-2*

Chapter 24. Storage

dm-era Device Mapper

The *device-mapper-persistent-data* package now provides tools to help use the new **dm-era** device mapper functionality released as a Technology Preview. The **dm-era** functionality keeps track of which blocks on a device were written within user-defined periods of time called an **era**. This functionality allows backup software to track changed blocks or restore the coherency of a cache after reverting changes.

DIF/DIX support

DIF/DIX, is a new addition to the SCSI Standard and a Technology Preview in Red Hat Enterprise Linux 6. DIF/DIX increases the size of the commonly used 512-byte disk block from 512 to 520 bytes, adding the Data Integrity Field (DIF). The DIF stores a checksum value for the data block that is calculated by the Host Bus Adapter (HBA) when a write occurs. The storage device then confirms the checksum on receive, and stores both the data and the checksum. Conversely, when a read occurs, the checksum can be checked by the storage device, and by the receiving HBA.

The DIF/DIX hardware checksum feature must only be used with applications that exclusively issue ***O_DIRECT*** I/O. These applications may use the raw block device, or the XFS file system in ***O_DIRECT*** mode. (XFS is the only file system that does not fall back to buffered I/O when doing certain allocation operations.) Only applications designed for use with ***O_DIRECT*** I/O and DIF/DIX hardware should enable this feature.

For more information, refer to section *Block Devices with DIF/DIX Enabled* in the Storage Administration Guide.

Package: *kernel-2.6.32-642*

LVM Application Programming Interface (API)

Red Hat Enterprise Linux 6 features the new LVM application programming interface (API) as a Technology Preview. This API is used to query and control certain aspects of LVM.

Package: *lvm2-2.02.143-7*

Chapter 25. Virtualization

Performance monitoring in KVM guests

As a Technology Preview, **KVM** can virtualize a performance monitoring unit (vPMU) to allow virtual machines to use performance monitoring. Additionally it supports Intel's “architectural PMU” which can be live-migrated across different host CPU versions, using the **-cpu** host option.

The virtual performance monitoring feature allows virtual machine users to identify sources of performance problems in their guests, using their preferred pre-existing profiling tools that work on the host as well as the guest. Note that this is an addition to the existing ability to profile a KVM guest from the host.

Package:*kernel-2.6.32-642*

System monitoring using SNMP

As a Technology Preview, Red Hat Enterprise Linux 6 allows Simple Network Management Protocol (SNMP) to be used for system monitoring. This allows KVM hosts to send SNMP traps on events so that hypervisor events can be communicated to the user via standard SNMP protocol. In addition, SNMP is capable of performing basic virtual networking operations, such as starting and stopping the virtual domain.

Package:*libvirt-snmp-0.0.2-5*

Zero-copy compatibility for macvtap-vhost

The *macvtap-vhost* zero-copy capability is available on Red Hat Enterprise Linux 6 as a Technology Preview. This feature allows running networking work loads in very high wire speeds but with low CPU resource consumption, and it does not limit other features such as memory overcommit and guest migration, which is not the case when using PCI device assignment to achieve the wire speed. Note that this feature is disabled by default.

Package:*qemu-kvm-0.12.1.2-2.491*

vCPU hot unplug

Although hot-plugging a virtual CPU (vCPU) is a supported operation, hot-unplugging a vCPU remains a Technology Preview in Red Hat Enterprise Linux 6, and is strongly recommended not to be used in high-value deployments.

Package:*libvirt-0.10.2-60*

Part III. Device Drivers

This part provides a comprehensive listing of all device drivers which were updated in Red Hat Enterprise Linux 6.8.

Chapter 26. Storage Driver Updates

- The **md** driver has been updated to the latest upstream version.
- The **nvme** driver has been updated to version 0.10.
- The **O2Micro** card reader driver, which specifically enables the SDHCI card reader to work on the O2Micro chips, has been updated to the latest upstream version.
- The **ipr** driver, used to enable new SAS VRAID adapters on POWER, has been updated to version 2.6.3.
- The **tcm_fc.ko** (FCoE fabric) driver has been updated to the latest upstream version.
- The **qla2xxx** driver has been updated to version 8.07.00.26.06.8-k.
- The **LPFC** (Avago Emulex Fibrechannel) driver has been updated to version 11.0.0.4.
- The **megaraid_sas** driver has been updated to version 06.810.09.00-rh1.
- The **mpt2sas** driver has been updated to version 20.102.00.00.
- The **mpt3sas** driver has been updated to version 09.102.00.00-rh.
- The **hpsa** (HP Smart Array SCSI driver) driver has been updated to version 3.4.10-0-RH1.

Chapter 27. Network Driver Updates

- The **ixgbevf** driver has been updated to version 2.12.1-k.
- The **ixgbe** driver has been updated to version 4.2.1-k.
- The **mlx5_core** driver has been updated to version 3.0-1.
- The **3c59x** network card driver has been updated to the latest upstream version.
- The **rds** (Reliable Datagram Sockets) driver has been updated to the latest upstream version.
- The **be2iscsi** driver has been updated to version 10.4.272.1r, which is the equivalent of upstream version 10.6.0.1.
- The **fnic** driver has been updated to version 1.6.0.17a.
- The **ocrdma** driver has been updated to the latest upstream version.
- The **ibmveth** (IBM Virtual Ethernet) driver has been updated to version 1.05.
- The **hfi** driver has been updated to the latest upstream version.
- The **ocrdma** network hardware driver has been rebased to upstream version 4.1.
- The **bna** driver has been updated to version 3.2.25.1r.
- The **i40e** driver has been updated to version 1.4.7-k. Note that only the Ethernet functionalities of this driver are supported in Red Hat Enterprise Linux 6.8.
- The **i40evf** driver has been updated to version 1.4.3.
- The **enic** driver has been updated to version 2.3.0.12.
- The **be2net** driver has been updated to version 10.6.0.3r.
- The **bnx2x** driver has been updated to version 1.712.30-0.
- The **bnx2** driver has been updated to version 2.2.6.
- The **tg3** driver has been updated to version 3.137.
- The **e100** driver has been updated to version 3.5.24-k2-NAPI.
- The **e1000** driver has been updated to version 7.3.21-k8-NAPI.
- The **e1000e** driver has been updated to version 3.2.6-k.
- The **bfa** driver has been updated to upstream version 3.2.25.
- The wireless drivers, **brcmfmac**, **rtlwifi**, **rt2x00**, **iwlwifi**, **b43**, **iwlegacy**, **carl9170**, **ath5k**, and **ath9k**, have been updated to the upstream kernel version 4.3.6.

Chapter 28. Graphics Driver and Miscellaneous Driver Updates

- The **hv_utils** driver, which implements guest/host integration for Hyper-V guests, has been updated to the latest upstream version.
- The **drm** subsystem drivers (ast, bochs, cirrus, gma500, i915, mga200, nouveau, qxl, radeon, udl, and vmwgfx) have been updated to version 4.4.
- The **xorg-x11-drv-intel** driver has been updated to the latest upstream version.

Chapter 29. Deprecated Functionality

This chapter provides an overview of functionality that has been deprecated, or in some cases removed, in all minor releases up to Red Hat Enterprise Linux 6.8.

Deprecated functionality continues to be supported until the end of life of Red Hat Enterprise Linux 6. Deprecated functionality will likely not be supported in future major releases of this product and is not recommended for new deployments. For the most recent list of deprecated functionality within a particular major release, refer to the latest version of release documentation.

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

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

Deprecated device drivers

- 3w-9xxx
- 3w-sas
- 3w-xxxx
- aic7xxx
- i2o
- ips
- megaraid_mbox
- mptbase
- mptctl
- mptfc
- mptlan
- mptsas
- mptscsih
- mptspi
- sym53c8xx
- qla3xxx

The following controllers from the **`megaraid_sas`** driver have been deprecated:

- Dell PERC5, PCI ID 0x15
- SAS1078R, PCI ID 0x60
- SAS1078DE, PCI ID 0x7C

- SAS1064R, PCI ID 0x411
- VERDE_ZCR, PCI ID 0x413
- SAS1078GEN2, PCI ID 0x78

The following controllers from the **be2iscsi** driver have been deprecated:

- BE_DEVICE_ID1, PCI ID 0x212
- OC_DEVICE_ID1, PCI ID 0x702
- OC_DEVICE_ID2, PCI ID 0x703

Note that other controllers from the mentioned drivers that are not listed here remain unchanged.

openswan component

The *openswan* packages have been deprecated, and *libreswan* packages have been introduced as a direct replacement for *openswan* to provide the VPN endpoint solution. *openswan* is replaced by *libreswan* during the system upgrade.

seabios component

Native KVM support for the S3 (suspend to RAM) and S4 (suspend to disk) power management states has been discontinued. This feature was previously available as a Technology Preview.

The zerombr yes Kickstart command is deprecated

In some earlier versions of Red Hat Enterprise Linux, the **zerombr yes** command was used to initialize any invalid partition tables during a Kickstart installation. This was inconsistent with the rest of the Kickstart commands due to requiring two words while all other commands require one. Starting with Red Hat Enterprise Linux 6.7, specifying only **zerombr** in your Kickstart file is sufficient, and the old two-word form is deprecated.

Btrfs file system

B-tree file system (Btrfs) is considered deprecated for Red Hat Enterprise Linux 6. Btrfs was previously provided as a Technology Preview, available on AMD64 and Intel 64 architectures.

eCryptfs file system

eCryptfs file system, which was previously available as a Technology Preview, is considered deprecated for Red Hat Enterprise Linux 6.

mingw component

Following the deprecation of Matahari packages in Red Hat Enterprise Linux 6.3, at which time the *mingw* packages were noted as deprecated, and the subsequent removal of Matahari packages from Red Hat Enterprise Linux 6.4, the *mingw* packages were removed from Red Hat Enterprise Linux 6.6 and later.

The *mingw* packages are no longer shipped in Red Hat Enterprise Linux 6 minor releases, nor will they receive security-related updates. Consequently, users are advised to uninstall any earlier releases of the *mingw* packages from their Red Hat Enterprise Linux 6 systems.

virtio-win component, BZ#1001981

The VirtIO SCSI driver is no longer supported on Microsoft Windows Server 2003 platform.

fence-agents component

Prior to Red Hat Enterprise Linux 6.5 release, the Red Hat Enterprise Linux High Availability Add-On was considered fully supported on certain VMware ESXi/vCenter versions in combination with the fence_scsi fence agent. Due to limitations in these VMware platforms in the area of SCSI-3 persistent reservations, the **`fence_scsi`** fencing agent is no longer supported on any version of the Red Hat Enterprise Linux High Availability Add-On in VMware virtual machines, except when using iSCSI-based storage. See the Virtualization Support Matrix for High Availability for full details on supported combinations:

https://access.redhat.com/site/articles/29440

Users using **`fence_scsi`** on an affected combination can contact Red Hat Global Support Services for assistance in evaluating alternative configurations or for additional information.

`systemtap` component

The *systemtap-grapher* package has been removed from Red Hat Enterprise Linux 6. For more information, see https://access.redhat.com/solutions/757983.

`matahari` component

The **Matahari** agent framework (*matahari-**) packages have been removed from Red Hat Enterprise Linux 6. Focus for remote systems management has shifted towards the use of the CIM infrastructure. This infrastructure relies on an already existing standard which provides a greater degree of interoperability for all users.

`distribution` component

The following packages have been deprecated and are subjected to removal in a future release of Red Hat Enterprise Linux 6. These packages will not be updated in the Red Hat Enterprise Linux 6 repositories and customers who do not use the MRG-Messaging product are advised to uninstall them from their system.

- *python-qmf*
- *python-qpid*
- *qpid-cpp*
- *qpid-qmf*
- *qpid-tests*
- *qpid-tools*
- *ruby-qpid*
- *saslwrapper*

Red Hat MRG-Messaging customers will continue to receive updated functionality as part of their regular updates to the product.

`fence-virt` component

The **libvirt-qpid** is no longer part of the fence-virt package.

`openscap` component

The *openscap-perl* subpackage has been removed from *openscap*.

Appendix A. Revision History

Revision 0.1-7 **Thu Apr 27 2017** **Lenka Špačková**

Added the deprecated **`zerombr yes`** Kickstart command to Deprecated Functionality.

Revision 0.1-6 **Tue Mar 17 2017** **Jiří Herrmann**

Updated a deprecated feature for virtualization.

Revision 0.1-5 **Fri Dec 16 2016** **Lenka Špačková**

Removed Indic languages from the International Languages chapter.

Revision 0.1-4 **Fri Sep 23 2016** **Lenka Špačková**

Added the **`qla3xxx`** driver to Deprecated Functionality.

Revision 0.1-3 **Thu Jul 14 2016** **Lenka Špačková**

Added details regarding support of the **`i40e`** driver.

Revision 0.1-2 **Fri Jun 03 2016** **Lenka Špačková**

Added Bugzilla numbers to individual descriptions. Updated Deprecated Functionality.

Revision 0.1-1 **Tue May 31 2016** **Lenka Špačková**

Added **ypserv** and **ypbind** bug fixes to Authentication and Interoperability.

Revision 0.1-0 **Mon May 16 2016** **Lenka Špačková**

Added new bug fixes to Clustering (fence agent) and Kernel (iscsi_firmware).

Revision 0.0-9 **Mon May 09 2016** **Lenka Špačková**

Release of the Red Hat Enterprise Linux 6.8 Technical Notes.

Revision 0.0-5 **Tue Mar 15 2016** **Lenka Špačková**

Release of the Red Hat Enterprise Linux 6.8 Beta Technical Notes.