Unbreakable Enterprise Kernel

Unbreakable Enterprise Kernel Release 7 Update 3 - Release Notes (Version 5.15.0-300)

F99954-11
December 2025

ORACLE

Unbreakable Enterprise Kernel Unbreakable Enterprise Kernel Release 7 Update 3 - Release Notes (Version 5.15.0-300),

F99954-11

Contents

4 Installation and Availability

5 List of CVEs fixed in this release

Preface

Unbreakable Enterprise Kernel Release 7 Update 3: Release Notes (5.15.0-300) provides a summary of the new features, significant changes, and any known issues in Unbreakable Enterprise Kernel Release 7 Update 3 (UEK R7U3).

1

About Unbreakable Enterprise Kernel Release 7 Update 3

This chapter provides an overview of Unbreakable Enterprise Kernel Release 7 Update 3 (UEK R7U3) and contains important information about this major release.

> **Note**
>
> Upgrading from an Unbreakable Enterprise Kernel Developer Preview release to its later official version isn't supported. If you're running the Developer Preview version, you must reinstall the official UEK release upon its general availability.

UEK R7U3 is initially released with the 5.15.0-300.163.18 version of the kernel. The kernel's source code is available through a public git source code repository at https://github.com/oracle/linux-uek.

The following is a general description of the scope of support for UEK R7U3:

- The kernel is developed, built, and tested on the 64-bit Arm (aarch64), Intel® 64-bit x86_64, and AMD 64-bit x86_64 architectures and is based on the mainline Linux kernel version 5.15.0.
- UEK R7U3 is made available for installation on the latest Oracle Linux 8 and Oracle Linux 9 update releases.
- In UEK R7U3, more features are enabled to provide support for key functional requirements and patches are applied to improve performance and optimize the kernel for use on Oracle operating environments. Note that Oracle actively monitors upstream check-ins and applies critical bug and security fixes to UEK R7U3.
- Although UEK R7U3 uses the same versioning model as the mainline Linux kernel version, it's possible that some applications might not understand the 5.15.0 versioning scheme. Note, however, that regular Linux applications are usually neither aware of nor affected by Linux kernel version numbers.
- A version of UEK R7U3 that enables 64k pages is available for 64-bit Arm (aarch64) platforms for Oracle Linux 9. The `kernel-uek64k` package is available on Oracle Cloud Infrastructure Arm compute shapes only. Use of this kernel outside of Oracle Cloud Infrastructure is only available as a technical preview.

Certification of UEK R7 for Oracle Products

The following important information applies to the certification of Oracle products with UEK R7.

Note that certification of different Oracle products with UEK R7 might not be immediately available at the time of the UEK R7 release. Ensure that the product you're using is certified for use with UEK R7 before upgrading or installing the kernel. You can check for certification information at https://support.oracle.com/epmos/faces/CertifyHome.

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) certification for different kernel versions is described in Document ID 1369107.1, which is available at https://support.oracle.com/epmos/faces/DocumentDisplay?id=1369107.1.

Oracle Automatic Storage Management Filter Driver (Oracle ASMFD) certification for different kernel versions is described in Document ID 2034681.1, which is available at https://support.oracle.com/epmos/faces/DocumentDisplay?id=2034681.1.

Compatibility

Oracle Linux maintains full user space compatibility with Red Hat Enterprise Linux (RHEL), which is independent of the kernel version that's running underneath the OS. Note that existing applications in user space continue to run unmodified with UEK R7; no recertifications are required for RHEL certified applications.

To minimize any impact on interoperability during releases, the Oracle Linux team works with third-party vendors that have hardware and software with dependencies on kernel modules. The kernel ABI for UEK R7 will remain unchanged in all subsequent updates to the initial release. Customers migrating from UEK6 must be aware that kernel ABIs have changed in UEK7. If an application is using kernel modules, users must verify the support status with the application vendor.

Notable changes in kernel headers

Upstream changes to kernel headers might mean that third-party modules do not compile across different kernel versions without modification to source code. Notably, the `memcg_cache_params` structure has been moved from `include/linux/slab.h` to `mm/slab.h`, which means that code needs to be refactored to account for the change if you are compiling across kernel versions.

To solve this problem so that the code can compile for UEK R6 and UEK R7, change the header requirements in the source code. For example, change lines like those in the following example to what is shown in the second example:

```
#ifdef CONFIG_SLUB
#include <linux/slub_def.h>
#endif


#if ( LINUX_VERSION_CODE < KERNEL_VERSION(5,4,0) )

#ifdef CONFIG_SLUB
#include <linux/slub_def.h>
#endif

#endif
```

2
New Features and Changes

This chapter describes new features, enhancements, and other notable changes that are introduced in UEK R7U3.

(aarch64) 64k Base Page Size on Arm

In addition to the standard build of UEK for Arm (aarch64), which sets a base 4k page size, a `kernel-uek64k` package that sets a 64k base page size is available for Ampere Arm-based Compute shapes in Oracle Cloud Infrastructure only. For use cases other than OCI, the `kernel-uek64` package is available only as a technical preview.

The 64k page size kernel is a useful option for Ampere (Arm-based) platforms that process workloads with large, contiguous memory datasets, and can achieve better performance for some types of memory and CPU intensive operations.

The 4k page size kernel is useful for smaller environments, where minimizing physical system memory usage is a priority.

Note that the 4k page size kernel and 64k page size kernel don't differ in user experience as the user space is the same.

After a system is installed with `kernel-uek64k` switching to a 4k kernel page size is unsupported.

Installing `kernel-uek64k`

> **Note**
>
> Installation of `kernel-uek64k` on systems outside of Oracle Cloud Infrastructure (OCI) is only available as a technical preview. Don't install this kernel on production systems outside of OCI.

To install the `kernel-uek64k` on a system installed with the standard 4k page size `kernel-uek`:

1. Install the `kernel-uek64k` package.

   ```
   sudo dnf install -y kernel-uek64k
   ```

2. Set the 64k page size kernel as the default kernel.

   ```
   sudo grubby --set-default=$(echo /boot/vmlinuz*64k)
   ```

Note that if you have more than one 64k page kernel installed, you must explicitly declare the kernel that you intend to be the default. For example:

```
sudo grubby --set-default=/boot/
vmlinuz-5.15.0-306.177.4.1.el9uek.aarch64.64k
```

3. Reboot the system.

   ```
   sudo reboot
   ```

4. After the system is rebooted, verify that the page size is 64k.

   ```
   getconf PAGESIZE
   ```

   If the PAGESIZE returns 65536, the 64k kernel is loaded. If the PAGESIZE returns 4096, the 4k kernel is loaded and you must check that the default kernel is set correctly.

   You can also check that the running kernel contains the 64k string, for example:

   ```
   uname -a|grep 64k
   ```

5. If the system is running the 64k kernel, proceed to remove the 4k page size kernel packages to avoid future conflicts.

   ```
   sudo dnf erase kernel-uek-core
   ```

TLS Encrypted Connections for NFS

RPC-With-TLS is enabled in the Linux NFS server and client. This update provides a standards-based peer authentication mechanism over an encrypted connection using TLS. The TLS Record protocol is handled entirely by kTLS.

Note that both the server and client systems must run UEK R7U3 or later, or must be running a kernel and user space client that supports RFC 9289, to use this functionality. The user space package, `ktls-utils`, is also required and must be installed on both the client and server systems. Also ensure that you have installed the most recent version of the `nfs-utils` package or that you have done a full system update.

RPC-With-TLS is contributed upstream by Oracle and is described in RFC 9289.

TIOCSTI Hardening Option

TIOCSTI is an ioctl system call in the Linux kernel that lets a process simulate terminal input by pushing characters into the input queue for a controlling TTY. This legacy mechanism can be abused for malicious purposes. We recommend always disabling it on systems running Oracle Linux.

Harden a system by disabling TIOCSTI. Set the value of the sysfs parameter `dev.tty.legacy_tiocsti` to `0`. For example, run:

```
echo "dev.tty.legacy_tiocsti = 0" | sudo tee -a /etc/sysctl.d/50-tiocsti.conf
sudo sysctl -p /etc/sysctl.d/50-tiocsti.conf
```

> **Note**
>
> Processes that run with CAP_SYS_ADMIN, such as BRLTTY, can use TIOCSTI even when this functionality is disabled.

BPF-LSM Enabled at Boot

BPF-LSM, the ability to attach Berkeley Packet Filter (BPF) programs to Linux Security Module (LSM) hooks to implement some security enhancements, is enabled in all UEK R7 kernel configurations, however it previously required setting the `lsm=bpf` boot command line option to use the feature.

In this release, `bpf` is added to CONFIG_LSM so that it doesn't need to be manually enabled at boot.

You can check that BPF is added to LSM by running:

```
cat /sys/kernel/security/lsm
```

> **Note**
>
> This feature was enabled in a UEK R7U3 errata release and is available in kernel-uek-5.15.0-315.196.5 and later.

Updated Drivers

In close cooperation with hardware and storage vendors, Oracle has updated several device drivers from the versions in mainline Linux 5.15.0.

Many driver modules no longer track version information. Oracle works with vendors to align device drivers included in UEK R7U3 with the code available in upstream kernel versions.

Notable driver updates are presented in the following table:

Table 2-1 Driver Alignment

Driver Module	Driver Description	Aligned Kernel Version	Notable Updates
`mlx5`	NVIDIA 5th Generation Network Adapters (NVIDIA ConnectX series) Core Driver	6.7	N/A
`lpfc`	Broadcom Emulex Fibre Channel HBA Driver	6.9	N/A
`qla2xxx`	Marvell QLogic Fibre Channel HBA Driver	6.10	N/A
`mpt3sas`	Broadcom (formerly LSI) MPT Fusion SAS 3.0 Device Driver	6.9	N/A

Table 2-1 (Cont.) Driver Alignment

Driver Module	Driver Description	Aligned Kernel Version	Notable Updates
`megaraid_sas`	Broadcom MegaRAID SAS Driver	6.9	N/A
`mpi3mr`	Broadcom MPI3 Storage Controller Device Driver	6.10	N/A
`smartpqi`	Microchip Smart Family Controller Driver	6.9	N/A
`bnxt_en`	Broadcom BCM573xx Network Driver	6.8	The driver now includes patches to work with the latest BCM57608 chip.
`mana`	Microsoft Azure Network Adapter	6.10	N/A

Deprecated and Removed Features

The following features are deprecated or no longer available in: UEK R7U3:

- **Unrestricted access to the kernel ring buffer is deprecated.**

  Unprivileged access to the kernel ring buffer through the `dmesg` command output is deprecated and will be removed in a future release of UEK. Use the `sudo` command to escalate to administrator privileges when running the `dmesg` command. To restrict access to the kernel ring buffer, set the `kernel.dmesg_restrict` sysfs parameter to 1.

- **`CONFIG_SECURITY_SELINUX_DISABLE` and `CONFIG_SECURITY_WRITABLE_HOOKS` options for disabling SELinux at runtime**

  The SELinux file system (`selinuxfs`) `/sys/fs/selinux/disable` node lets you disable SELinux at runtime before a policy is loaded into the kernel. If disabled using this mechanism, SELinux remains disabled until the system is rebooted.

  The option to disable SELinux at runtime makes it difficult to secure the kernel's LSM hooks using the "__ro_after_init" feature. Therefore, these options are deprecated in this UEK release.

  The preferred method of disabling SELinux is by using the `selinux=0` boot parameter

- **`CONFIG_CRYPTO_OFB` and `CONFIG_CRYPTO_CFB` cryptographic modes**

  The CFB (Cipher Feedback) mode (NIST SP800-38A) used for TPM2 cryptography and the OFB (Output Feedback) mode (NIST SP800-38A) used to turn a block cipher into a synchronous stream cipher are deprecated in this UEK release, and might be removed from the kernel in a future UEK release.

- **`CONFIG_RPCSEC_GSS_KRB5_ENCTYPES_DES` option for 3DES/DES3 RPCSEC GSS encryption types**

  The RPCSEC GSS encryption types DES and Triple-DES (3DES/DES3) are deprecated in this UEK release, and might be removed from the kernel in a future UEK release.

  These encryption types were deprecated by RFCs 6649 and 8429 because they're known to be insecure.

- **`CONFIG_NFS_V2` and `CONFIG_NFSD_V2` options for NFSv2 client and server**
  Support for NFSv2 clients and NFSv2 servers is deprecated in this UEK release, and might be removed from the kernel in a future UEK release.

  NFSv2 has long been replaced by NFSv3 and NFSv4, which offer improved functionality, performance, and security.

- **`CONFIG_NFS_DISABLE_UDP_SUPPORT` option for NFSv3 over UDP**
  Support for NFS version 3 over the UDP network protocol is deprecated in this UEK release, and might be removed from the kernel in a future UEK release.

  Modern NFS/RPC over TCP and RDMA implementations provide better performance than UDP, and provide reliable ordered delivery of data combined with congestion control.

  Note that NFSv4 is already not supported over UDP, for the same reasons.

- **`CONFIG_STAGING` option**

  With the `CONFIG_STAGING` kernel configuration option, you can select drivers that don't necessarily meet the highest kernel quality level but are merely made available for test use. However, the kernel option `CONFIG_STAGING` is deprecated in this UEK release and might be removed in a future release.

- **`CONFIG_IXGB` option**
  The `CONFIG_IXGB` for Intel PRO/10GbE hardware is deprecated and might be removed from the kernel in a future UEK release.

- **`CONFIG_IP_NF_TARGET_CLUSTERIP` option**
  The `CONFIG_IP_NF_TARGET_CLUSTERIP` option that allowed you to build load-balancing clusters of network servers without a dedicated load-balancing router or switch is deprecated in favor of functionality already in Netfilter cluster match.

- **`CONFIG_EFI_VARS` option**
  The `CONFIG_EFI_VARS` option that provided the `efivars` sysfs interface to configure UEFI variables is removed from the upstream kernel and is deprecated in this release of UEK. Replacement functionality has been present in the kernel since 2012. For more information, see https://www.kernel.org/doc/html/latest/filesystems/efivarfs.html.

- **Firewire driver**

  The `CONFIG_FIREWIRE` option was disabled in Oracle Linux 9. Thus, the Firewire driver is deprecated and unusable in this UEK release.

- **`crashkernel=auto` option**

  The `crashkernel=auto` option is deprecated and no longer supported on Oracle Linux 9 and therefore unsupported for UEK R7 on Oracle Linux 9. Some platforms, such as the Raspberry Pi have maximum limits for `crashkernel` memory reservation and these must be specified explicitly. This option will be removed in a future UEK release.

- **Several network scheduler modules**

  The following network scheduler modules are deprecated:

  - `cls_tcindex`
  - `cls_rsvp`
  - `sch_dsmark`
  - `sch_atm`
  - `sch_cbq`

  These modules might be disabled or blocklisted and can be removed in a future release of UEK. The modules are already removed in the upstream Linux kernel.

- **`resilient_rdmaip` Module Deprecated**
  The `resilient_rdmaip` module is deprecated in UEK R7. This module will be removed in a future UEK release.

3
Known Issues

This chapter describes any known issues for Unbreakable Enterprise Kernel Release 7.

dracut-install: ERROR: installing 'virtio' might be displayed during UEK R7 installation

In UEK R7, `virtio` isn't built as a module, but is built directly into the kernel. As such, you don't have to specify `virtio` in the dracut configuration file to add it to initramfs. If you previously had dracut configuration that included this module, attempting to install UEK R7 displays the following dracut error:

```
dracut-install: ERROR: installing 'virtio'
dracut: FAILED:  /usr/lib/dracut/dracut-install -D
/var/tmp/dracut.FOKWjy/initramfs --kerneldir
/lib/modules/5.15.0-0.21.1.el8uek.x86_64/ -m xen_netfront xen_blkfront
virtio_blk virtio_net virtio virtio_pci virtio_balloon hyperv_keyboard
hv_netvsc hid_hyperv hv_utils hv_storvsc hyperv_fb ahci libahci
dracut-install: ERROR: installing 'virtio'
dracut: FAILED:  /usr/lib/dracut/dracut-install -D
/var/tmp/dracut.G2XSGh/initramfs --kerneldir
/lib/modules/5.15.0-0.21.1.el8uek.x86_64/ -m xen_netfront xen_blkfront
virtio_blk virtio_net virtio virtio_pci virtio_balloon hyperv_keyboard
hv_netvsc hid_hyperv hv_utils hv_storvsc hyperv_fb ahci libahci
```

This error is displayed, regardless of whether you use the `yum` or `rpm` command to install UEK R7.

To work around the issue, before installing UEK R7, remove the "virtio" text from the dracut configuration file. Make sure to remove *only* the "virtio" text, leaving all other "virtio_*" entries intact, for example:

```
cat /etc/dracut.conf.d/01-dracut-vm.conf


add_drivers+=" xen_netfront xen_blkfront "
add_drivers+=" virtio_blk virtio_net virtio virtio_pci virtio_balloon "
add_drivers+=" hyperv_keyboard hv_netvsc hid_hyperv hv_utils hv_storvsc
hyperv_fb "
add_drivers+=" ahci libahci "
```

Use the following command to verify that `virtio` is built into the kernel:

```
grep CONFIG_VIRTIO= /boot/config-5.15.0-0.30.4.el8uek.x86_64
```

If `virtio` is built into the kernel, the output should be as follows:

```
CONFIG_VIRTIO=y
```

(Bug ID 33834972)

Upgrading from UEK R6 to UEK R7 on Arm platform may fail if RAID 5 default page size differs from default stripe size

Starting with UEK R7, the default page size on the Arm platform has changed to 4 KB, from the previous 64 KB default. This change in page size might cause an upgrade from UEK R6 to UEK R7 to fail on systems that are configured for RAID 5 when the default page size differs from the default stripe size.

For this reason, before upgrading from UEK R6 to UEK R7, back up and reformat RAID 5 volumes. In cases where retaining the same RAID 5 configuration is preferred, we recommend that you continue to run UEK R6.

See Default Page Size on Arm Platform Changed to 4 KB for additional information.

(Bug ID 33858264)

Swap partitions created on Arm platform using an earlier UEK release don't work after upgrade to UEK R7

The UEK R7 release includes a significant change for the Arm platform regarding the default page size, which has changed to 4 KB, from the previous 64 KB default. Any swap partitions that were created on the Arm platform using an earlier UEK release, for example, UEK R6, don't work after upgrading to UEK R7.

> **Note**
>
> This issue applies to the Arm platform, irrespective of file system type.

Upon the first boot into UEK R7 after an upgrade, the following `systemd` service failure is indicated:

```
systemctl list-units --failed
UNIT LOAD ACTIVE SUB DESCRIPTION

dev-mapper-ol_myhost\x2dswap.swap loaded failed failed
/dev/mapper/ol_myhost-swap
```

To work around this issue, you must reinitialize the swap device with the new page size after upgrading to UEK R7. Use the `swapon` command as follows and specify the swap location:

```
sudo swapon --fixpgsz /dev/mapper/ol_myhost-swap
```

```
swapon: /dev/mapper/ol_myhost-swap: swap format pagesize does not match.
swapon: /dev/mapper/ol_myhost-swap: reinitializing the swap.
mkswap: /dev/mapper/ol_myhost-swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=d7ef0a33-403f-447b-863f-d52b7f66c803
```

In the previous command, `/dev/mapper/ol_myhost-swap` is an example of a typical swap location that you might specify.

For more information about the important change in default page size for the Arm platform in UEK R7, see Default Page Size on Arm Platform Changed to 4 KB.

(Bug ID 34322552)

Cloud-init and systemd-udevd fail to rename mlx5_core network interfaces during upgrade from UEK R6 to UEK R7

During an upgrade from UEK R6 to UEK R7 on an Oracle Infrastructure instance, `cloud-init` and `systemd-udevd` revert to using the older UEK R6 device naming scheme (`ifcfg-ens300f0`) for the `mlx5_core` network interface, rather than correctly renaming the device with the new UEK R7 device naming scheme (`ens300f0np0`).

To ensure that the `mlx5_core` network interface does not revert to using the former UEK R6 device naming scheme, do the following after the upgrade to UEK R7 has completed, prior to rebooting the system:

1. Remove the old network configuration file, for example:

   ```
   sudo rm /etc/sysconfig/network-scripts/ifcfg-ens300f0
   ```

2. Remove any cached data saved by `cloud-init`:

   ```
   sudo cloud-init clean
   ```

3. Reboot the instance for the changes to take effect.

(Bug ID 34146775)

Mellanox NIC interface name subject to change after upgrading from UEK R6 to UEK R7

During a kernel upgrade from UEK R6 to UEK R7, the `mlx5_core` device name is subject to change, from `ens2f0` (UEK R6) to `ens2f0np0` (UEK R7).

You might encounter this issue under the following circumstances:

- When upgrading an Oracle Linux 8 system that is running UEK R6 to UEK R7.

- When upgrading an Oracle Linux 8 system that is running UEK R6 to Oracle Linux 9 (which ships with UEK R7 by default).
- When upgrading an Oracle Linux 8 system that is already running UEK R7 to Oracle Linux 9.

> **Note**
>
> In the case where an Oracle Linux 8 system is already running UEK R7, if you previously configured the system to use backwards-compatible device names (`ens2f0`), you might need to apply the workaround that follows to your GRUB configuration after the upgrade to Oracle Linux 9 has completed.

Note that fresh installations of UEK R7 on Oracle Linux 8 and Oracle Linux 9 use the default naming convention for UEK R7 (`enp2s0f0np0`) by default.

To retain backwards-compatible (UEK R6) device names for the `mlx5_core` driver-based network interface card (NIC), perform the following workaround after upgrading to UEK R7, prior to rebooting your system. It is recommended that you back up your existing `grub.cfg` file before making this change.

1. Edit the `/etc/default/grub` file and append the end of the line in the `GRUB_CMDLINE_LINUX=` module as follows:

   ```
   GRUB_CMDLINE_LINUX="console=xxxx mlx5_core.expose_pf_phys_port_name=0"
   ```

2. After editing the file, locate the `grub.cfg` file on your system, then run the command to update GRUB configuration, as appropriate:
   - On BIOS-based systems, the `grub.cfg` output/target file is usually located at `/boot/grub2/grub.cfg` and you would run the following command:

     ```
     sudo grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

   - On UEFI-based systems, the `grub.cfg` output/target file could be located at `/etc/grub2-efi.cfg` or `/boot/efi/EFI/redhat/grub.cfg`. Depending on the location of the file, you would run one of the following commands:

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

     ```
     sudo grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
     ```

3. Reboot the system for the changes to take effect.

(Bug IDs 34103369, 34145887)

Random high CPU utilization issue encountered with database benchmark program

A random high CPU utilization issue has been encountered with the database benchmark program running on a 192-CPU virtual machine in Azure. This issue was initially discovered in Oracle Linux 8.4 and Ubuntu 20.04 (5.11.0-1022-azure); however, a complete fix for the issue isn't yet available in the upstream kernels.

This issue typically manifests itself with a >90% CPU utilization spike occurring every 1 to 2 minutes and lasting approximately 5 to 20 seconds, which degrades the system's performance significantly. When the CPU utilization spike is occurring, *each* of the 192 CPUs' %sys increases up to 60+%, and the %si increases up to 30%. In certain cases, the >90% CPU utilization spike has been observed 100% of the time.

To avoid encountering this issue, set the `dm_mod.dm_mq_queue_depth=256` kernel parameter.

(Bug ID 33665982)

(aarch64) Disk Encryption Password Prompt Not Being Displayed at System Boot

If you install Oracle Linux with GUI on an encrypted disk, for example, by choosing Server with GUI during the installation stage, and VGA is enabled, the password prompt doesn't appear on the VGA output at system boot. Consequently, the boot process can not be completed. The prompt appears only on a serial console, and therefore, you would need to switch to a serial console to provide the password there.

This issue is specific to systems on the Arm platform only and occurs regardless of whether you're using secure boot or not. Further, the issue applies to Oracle Linux 8 or Oracle Linux 9 systems that use UEKR6 or UEKR7.

To make the GUI password prompt for disk encryption appear at boot time on VGA output without using a serial console, add `plymouth.ignore-serial-consoles` to the kernel command line in the GRUB configuration. For instructions, see Oracle Linux 8: Managing Kernels and System Boot or Oracle Linux 9: Managing Kernels and System Boot.

(Bug ID 35034465)

XFS DAX Mount Option Is Incompatible With Oracle Linux 9 With Reflink Enabled

On Oracle Linux 9 with UEK R7, the file system DAX mount option `dax=always` is incompatible with reflink-enabled XFS file systems. For example, running the command `sudo mount -o dax=always /dev/pmem1 /mnt` displays the following error:

```
mount: /mnt: wrong fs type, bad option, bad superblock on /dev/pmem1, missing
codepage
    or helper program, or other error.
mount: (hint) your fstab has been modified, but systemd still uses the old
version;
    use 'systemctl daemon-reload' to reload.
```

(Bug ID 35991195)

xdp-tools on Oracle Linux 9 Is Incompatible With UEK R7

The Oracle Linux 9 xdp-tools package that contains the xdp-monitor and xdp-bench commands is incompatible with UEK R7. The following errors are displayed when these commands are run on an Oracle Linux 9 system that's running UEK R7:

```
– END PROG LOAD LOG –
libbpf: prog 'tp_xdp_cpumap_kthread': failed to load: -22
libbpf: failed to load object 'xdp_sample'
libbpf: failed to load BPF skeleton 'xdp_sample': -22
```

If you need this package, use Oracle Linux 8 with xdp-tools v1.2.10-1.el8 or earlier.

(Bug ID 36014171)

4

Installation and Availability

This chapter provides information about the availability of UEK R7 on Oracle Linux and includes installation and instructions on upgrading from a previous UEK release to UEK R7.

UEK R7 is supported on the Intel® 64-bit x86_64, AMD 64-bit x86_64, and 64-bit Arm (aarch64) platforms.

About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7

UEK R7 is made available for installation on Oracle Linux 8, starting with the Oracle Linux 8.5 release. By default, Oracle Linux 9 ships with UEK R7.

The suggested migration path for upgrading the system from an earlier UEK release to UEK R7 is as follows:

- If you're running Oracle Linux 7 with an earlier UEK release, upgrade the operating system to the latest Oracle Linux 8 release. For instructions on upgrading the Oracle Linux 7 system, see Oracle Linux 8: Upgrading Systems With Leapp.
- If you're running an Oracle Linux 8 release that's earlier than Oracle Linux 8.5 with UEK R6, first upgrade the system to the latest Oracle Linux 8 update release. From here, you can upgrade to UEK R7. If you're already running Oracle Linux 8.5 or later with UEK R6, you can directly upgrade the system to UEK R7.

  For instructions on upgrading an Oracle Linux 8 system to Oracle Linux 9, see Oracle Linux 9: Upgrading Systems With Leapp.

Important

In UEK R7, the default page size for the 64-bit Arm (aarch64) architecture has changed to 4 KB default, from the previous 64 KB default. The new 4 KB default page size might have significant implications on Arm-based systems that are running Oracle Linux 8 with an earlier UEK release, with either a Btrfs or an XFS file system.

- If an Arm-based system uses a Btrfs or an XFS file system, and you're running Oracle Linux 8 with an earlier UEK release, you might not be able to upgrade to UEK R7 without first migrating data to an alternative file system. The default on-disk file system block size is set to be the equivalent of the page size for these file systems, which means that the change in page size can render the file system inaccessible and can cause data corruption.

  Note, however, that Oracle has placed checks within the UEK R7 Arm RPM that prevent the installation of UEK R7 if a Btrfs file system is detected and the resulting change in block size could cause data to become inaccessible.

- For an XFS file system, the default block size is 4 KB. XFS enables you to manually set the block size at file system creation time. If you have XFS file systems with a block size greater than 4 KB, you're required to migrate data before upgrading to UEK R7.

  Typically, a data migration plan might involve adding another storage device, formatting it with an unaffected file system or using XFS with the block size specified as 4 KB, and then moving the data onto the newly formatted device.

- Users of the Oracle Linux 8 developer image installed on Raspberry Pi systems are necessarily affected because the image uses a Btrfs file system, by default. If you're using this image, and you intend to upgrade to UEK R7, you must migrate data to an alternative unaffected file system before trying to install UEK R7. For more information about using the Raspberry Pi hardware platform, see Install Oracle Linux on a Raspberry Pi.

- Any existing swap partitions that were created on the Arm platform using an earlier UEK release, such as UEK R6, don't work after upgrading to UEK R7. The change to a 4 KB default page size on the aarch64 platform requires that any existing swap partitions on the system *must* be reinitialized with the new page size after booting the system with UEK R7. For further details, see Swap partitions created on Arm platform using an earlier UEK release don't work after upgrade to UEK R7.

For general information about working with file systems in Oracle Linux 8, see Oracle Linux 8: Performing File System Administration.

Obtaining Packages for Installation

If you have a subscription to Oracle Unbreakable Linux support, you can obtain the packages for UEK R7 by registering the system with the Unbreakable Linux Network (ULN) and then subscribing it to any extra channels. See Subscribing to ULN Channels.

If the system isn't registered with ULN, you can obtain most of the required packages from the Oracle Linux yum server. See Enabling Access to Oracle Linux Yum Server Repositories.

When you have subscribed the system to the appropriate ULN channels or to the Oracle Linux yum server, you can proceed to upgrade the system to UEK R7. See Upgrading a System to UEK R7.

Enabling Access to Oracle Linux Yum Server Repositories

Packages for UEK R7 and any associated user space applications are available on the Oracle Linux yum server at https://yum.oracle.com/.

For Oracle Linux 8, the kernel images and all the associated user space packages for both the x86_64 and aarch64 platforms are made available by enabling the following repositories:

- `ol8_UEKR7`
- `ol8_baseos_latest`

For Oracle Linux 9, the kernel images and all the associated user space packages for both the x86_64 and aarch64 platforms are made available by enabling the following repositories:

- `ol9_UEKR7`
- `ol9_baseos_latest`

To enable access to repositories on the Oracle Linux yum server, use the `dnf config-manager` command and specify the appropriate repositories for the release that you're running.

For example, you would enable access to the Oracle Linux 8 repositories as follows:

```
sudo dnf config-manager --enable ol8_baseos_latest ol8_UEKR7
```

> **Note**
>
> You can only use the `dnf config-manager` to enable or disable repositories that already have a configuration file for the specified repository. Repository configurations are typically stored in the `/etc/yum.repos.d` file. The repository configurations that are required to install the UEK release on Oracle Linux 8 and Oracle Linux 9 are included in the `oraclelinux-release-el8` and `oraclelinux-release-el9` packages. Note that you might need to update the package to the latest version to obtain the correct yum repository configuration.

Subscribing to ULN Channels

For Oracle Linux 8, kernel image and user space packages are made available for the x86_64 platform in the following ULN channels:

- `ol8_x86_64_UEKR7`
- `ol8_x86_64_baseos_latest`

For Oracle Linux 8, kernel image and user space packages are made available for the aarch64 platform in the following ULN channels:

- `ol8_aarch64_UEKR7`
- `ol8_aarch64_baseos_latest`

For Oracle Linux 9, kernel image and user space packages are made available for the x86_64 platform in the following ULN channels:

- `ol9_x86_64_UEKR7`

- `ol9_x86_64_baseos_latest`

For Oracle Linux 9, kernel image and user space packages are made available for the aarch64 platform in the following ULN channels:

- `ol9_aarch64_UEKR7`
- `ol9_aarch64_baseos_latest`

The following instructions assume that you have already registered the system with ULN.

To subscribe a system to a ULN channel:

1. Sign in to https://linux.oracle.com with a ULN username and password.
2. On the Systems tab, in the list of registered machines, click the link that corresponds to the name of the system.
3. On the System Details page, click **Manage Subscriptions**.
4. On the System Summary page, from the list of available channels, select each of the required channels, then click the right arrow to move the selected channel to the list of subscribed channels.
5. Click **Save Subscriptions**.

For more information about using ULN, see Oracle Linux: Managing Software on Oracle Linux.

Upgrading a System to UEK R7

The following instructions describe how to upgrade a system to UEK R7. For more details about the suggested migration paths for upgrading to UEK R7, see About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7.

1. Enable access to the appropriate ULN channels or yum repositories, as described in Subscribing to ULN Channels and Enabling Access to Oracle Linux Yum Server Repositories.

   > **Tip**
   >
   > Disable any other UEK channels or repositories that you might have previously configured as good practice.

2. After enabling access to the appropriate channels or repositories, upgrade the system to UEK R7 by running the following commands:

   ```
   sudo dnf install -y kernel-uek
   sudo dnf update -y
   ```

3. After the upgrade has completed, reboot the system.

   Ensure to select the UEK R7 kernel (version 5.15.0) if it's not the default boot kernel.

For questions regarding installing software or updating a system, see Oracle Linux: Managing Software on Oracle Linux.

Installing and Upgrading Oracle-Supported RDMA Packages on Oracle Linux

The following instructions describe how to install and upgrade Oracle-supported RDMA packages on Oracle Linux 8 and Oracle Linux 9.

Installing Oracle-Supported RDMA Packages on Oracle Linux 8

> **Note**
>
> These instructions apply to the x86_64 platform.

The following instructions describe how to install Oracle-Supported RDMA on an Oracle Linux 8 system. These instructions include steps on how to remove other previously installed RDMA packages that could cause conflicts when installing the UEK R7U3 RDMA packages.

If the system is running Oracle Linux 9, see Installing Oracle-Supported RDMA Packages on Oracle Linux 9 for instructions.

1. Subscribe the system to the appropriate RDMA ULN channel or yum repository.
   - If you're using the Oracle Linux yum server, enable the `ol8_UEKR7_RDMA` repository for Oracle Linux 8, for example:

     ```
     sudo dnf config-manager --enable ol8_baseos_latest ol8_UEKR7
     ol8_UEKR7_RDMA
     ```

   - If you're using ULN, subscribe to `ol8_x86_64_UEKR7_RDMA` channel.

   For further instructions, see Subscribing to ULN Channels and Enabling Access to Oracle Linux Yum Server Repositories.
2. Remove any existing packages that are related to RDMA, for example:

   ```
   sudo dnf remove 'ibacm*'
   sudo dnf remove 'ibutils*'
   sudo dnf remove 'infiniband-diags*'
   sudo dnf remove 'libibacl*'
   sudo dnf remove 'libibcm*'
   sudo dnf remove 'libibmad*'
   sudo dnf remove 'libibumad*'
   sudo dnf remove 'libibverbs*'
   sudo dnf remove 'librdmacm*'
   sudo dnf remove 'mstflint*'
   sudo dnf remove 'opensm*'
   sudo dnf remove 'oracle-rdma-release'
   sudo dnf remove 'oracle-rdma-tools'
   sudo dnf remove 'perftest*'
   sudo dnf remove 'qperf*'
   sudo dnf remove 'rdma*'
   sudo dnf remove 'rds-tools*'
   ```

3. Clean the yum cached files from all the enabled repositories:

   ```
   sudo dnf clean all
   ```

4. Install the RDMA packages for UEK R7.

   - Use the following commands to install the core packages:

     ```
     sudo dnf install rdma-core
     sudo dnf install libibverbs-utils
     sudo dnf install librdmacm-utils
     sudo dnf install mstflint
     sudo dnf install oracle-rdma-tools
     sudo dnf install rds-tools
     ```

     – If installing on a bare-metal system, install the `infiniband-diags` package:

       ```
       sudo dnf install infiniband-diags
       ```

     – If installing on a guest VM, install the `infiniband-diags-guest` package:

       ```
       sudo dnf install infiniband-diags-guest
       ```

   - (Optional) If you require the `perftest` package, install the package by running:

     ```
     sudo dnf install perftest
     ```

   - (Optional) If you require the `qperf` package, install the package by running:

     ```
     sudo dnf install qperf
     ```

   - (Optional) If you require the `libpcap` package, install the package by running:

     ```
     sudo dnf install libpcap
     ```

   - (Optional) If you require the `ibacm` package, install the package by running:

     ```
     sudo dnf install ibacm
     ```

   - (Optional) If you require the `srp_daemon` package, install the package by running:

     ```
     sudo dnf install srp_daemon
     ```

Each UEK release requires a different set of RDMA packages. If you change the kernel on the system to a UEK release that's earlier than UEK R7, remove the RDMA packages as instructed earlier before installing the correct packages for the new kernel.

> ⚠ **Caution**
>
> Downgrading UEK versions isn't advised, except for testing purposes.

Installing Oracle-Supported RDMA Packages on Oracle Linux 9

> **Note**
>
> These instructions apply to the x86_64 platform.

The process of installing Oracle-supported RDMA packages on Oracle Linux 9 is simplified by using new user space packages, and a dedicated ULN channel and yum repository for RDMA-related packages.

If the system is running Oracle Linux 8, the process of installing Oracle-supported RDMA packages remains the same as it was in previous releases. For instructions, see Installing Oracle-Supported RDMA Packages on Oracle Linux 8.

The following instructions describe how to install RDMA release packages on an Oracle Linux 9 system:

1. Ensure that you have subscribed to the ULN channel or have enabled the yum repository that contains the RDMA-related user space packages for Oracle Linux 9.
   - If you're installing packages from ULN, subscribe to the `ol9_x86_64_RDMA` channel.
   - If you're installing packages from the Oracle Linux yum server, enable the `ol9_RDMA` yum repository.
2. Clean the yum cached files from all the enabled repositories by running the following command:

   ```
   sudo dnf clean all
   ```

3. Install the RDMA packages for UEK R7.
   - Use the following commands to install the core packages:

     ```
     sudo dnf install rdma-core
     sudo dnf install libibverbs-utils
     sudo dnf install librdmacm-utils
     sudo dnf install mstflint
     sudo dnf install oracle-rdma-tools
     sudo dnf install rds-tools
     ```

     – If installing on a bare-metal system, install the `infiniband-diags` package:

       ```
       sudo dnf install infiniband-diags
       ```

     – If installing on a guest VM, install the `infiniband-diags-guest` package:

       ```
       sudo dnf install infiniband-diags-guest
       ```

   - (Optional) If you require the `perftest` package, install the package by running:

     ```
     sudo dnf install perftest
     ```

- (Optional) If you require the `qperf` package, install the package by running:

  ```
  sudo dnf install qperf
  ```

- (Optional) If you require the `libpcap` package, install the package by running:

  ```
  sudo dnf install libpcap
  ```

- (Optional) If you require the `ibacm` package, install the package by running:

  ```
  sudo dnf install ibacm
  ```

- (Optional) If you require the `srp_daemon` package, install the package by running:

  ```
  sudo dnf install srp_daemon
  ```

Upgrading Oracle-Supported RDMA Packages on Oracle Linux 8 and Oracle Linux 9

You can upgrade the Oracle-supported RDMA packages on Oracle Linux 8 and Oracle Linux 9 by using the `dnf update` command.

If you're upgrading a system that has the `oracle-rdma-release` or `oracle-rdma-release-guest` package installed, if the package version is lower than version 0.18.1-1 and you intend to upgrade to version 0.18.1-1, or later, you must first manually remove the `rdma-core-devel` package. Remove this package by using the `rpm -e --nodeps` command, which removes the package outside of the standard yum or DNF package manager control and leaves any dependencies intact, for example:

```
sudo /bin/rpm -e --nodeps rdma-core-devel
sudo dnf update
```

If the system you have upgraded has the `oracle-rdma-release` or `oracle-rdma-release-guest` package installed and if the package version is version 0.31.0-1, then you can remove it because that package no longer serves any purpose:

```
sudo dnf remove oracle-rdma-release*
```

5 List of CVEs fixed in this release

The following list describes the CVEs that are fixed in UEK R7U3 (5.15.0-300.163.18) as compared to initial release of UEK R7U2 (5.15.0-200.131.27). The content provided here is automatically generated and includes the CVE identifier and a summary of the issue.

Note that CVEs are continually handled in patch updates that are made available as errata builds for the current release. For this reason, it's critical that keep systems updated with the latest package updates for this kernel release. Many of the issues listed here might have already been resolved in prior errata builds for the previous update level.

You can keep current with the latest CVE information at https://linux.oracle.com/cve .

- CVE-2019-25162
- CVE-2020-26555
- **CVE-2021-3923**
- CVE-2021-4204
- CVE-2021-33631
- CVE-2021-46934
- CVE-2021-47579
- CVE-2021-47596
- CVE-2021-47624
- CVE-2022-3566
- CVE-2022-3567
- CVE-2022-4095
- CVE-2022-4269
- CVE-2022-4744
- CVE-2022-36402
- CVE-2022-38096
- CVE-2022-48619
- CVE-2022-48627
- CVE-2022-48632
- **CVE-2022-48637**
- CVE-2022-48743
- **CVE-2022-48746**
- CVE-2022-48747
- CVE-2022-48757
- **CVE-2022-48827**
- **CVE-2022-48828**

- **CVE-2022-48829**
- **CVE-2023-0045**
- CVE-2023-0386
- CVE-2023-0458
- **CVE-2023-0459**
- CVE-2023-0590
- CVE-2023-1206
- CVE-2023-1249
- CVE-2023-1281
- CVE-2023-1380
- CVE-2023-1382
- CVE-2023-1513
- CVE-2023-1582
- **CVE-2023-1611**
- CVE-2023-1637
- CVE-2023-1652
- CVE-2023-1670
- CVE-2023-1838
- CVE-2023-1855
- **CVE-2023-1872**
- CVE-2023-1989
- CVE-2023-1998
- CVE-2023-2002
- **CVE-2023-2008**
- **CVE-2023-2019**
- CVE-2023-2124
- **CVE-2023-2156**
- CVE-2023-2162
- CVE-2023-2163
- CVE-2023-2166
- **CVE-2023-2177**
- CVE-2023-2194
- CVE-2023-2235
- CVE-2023-2248
- CVE-2023-2269
- CVE-2023-2513
- **CVE-2023-2860**
- **CVE-2023-3006**

- CVE-2023-3090
- **CVE-2023-3111**
- **CVE-2023-3117**
- CVE-2023-3141
- CVE-2023-3159
- CVE-2023-3161
- CVE-2023-3212
- CVE-2023-3268
- **CVE-2023-3355**
- **CVE-2023-3357**
- CVE-2023-3358
- **CVE-2023-3389**
- CVE-2023-3390
- CVE-2023-3567
- CVE-2023-3609
- CVE-2023-3610
- CVE-2023-3611
- CVE-2023-3772
- CVE-2023-3773
- CVE-2023-3776
- CVE-2023-3777
- CVE-2023-3812
- CVE-2023-4004
- CVE-2023-4015
- CVE-2023-4128
- CVE-2023-4132
- CVE-2023-4147
- CVE-2023-4206
- CVE-2023-4207
- CVE-2023-4208
- CVE-2023-4244
- CVE-2023-4273
- CVE-2023-4387
- **CVE-2023-4389**
- **CVE-2023-4394**
- CVE-2023-4459
- **CVE-2023-4569**
- CVE-2023-4622

- CVE-2023-4623
- CVE-2023-4881
- CVE-2023-4921
- CVE-2023-5090
- **CVE-2023-5158**
- CVE-2023-5178
- CVE-2023-5197
- CVE-2023-6040
- CVE-2023-6111
- CVE-2023-6121
- CVE-2023-6176
- CVE-2023-6356
- CVE-2023-6531
- CVE-2023-6535
- CVE-2023-6536
- CVE-2023-6546
- CVE-2023-6622
- CVE-2023-6817
- CVE-2023-6932
- CVE-2023-7192
- **CVE-2023-21400**
- **CVE-2023-25012**
- CVE-2023-25775
- **CVE-2023-26605**
- **CVE-2023-28327**
- CVE-2023-28328
- **CVE-2023-28410**
- CVE-2023-28466
- CVE-2023-31084
- CVE-2023-31248
- CVE-2023-31436
- CVE-2023-33203
- **CVE-2023-33288**
- CVE-2023-34256
- CVE-2023-35001
- CVE-2023-35788
- CVE-2023-35823
- CVE-2023-35824

- CVE-2023-35827
- CVE-2023-37453
- CVE-2023-39189
- CVE-2023-39192
- CVE-2023-39193
- CVE-2023-39194
- CVE-2023-39198
- CVE-2023-40283
- CVE-2023-42752
- CVE-2023-42754
- CVE-2023-42756
- **CVE-2023-44466**
- CVE-2023-45862
- CVE-2023-45863
- CVE-2023-45871
- CVE-2023-46813
- **CVE-2023-46838**
- **CVE-2023-47233**
- CVE-2023-51042
- CVE-2023-51043
- CVE-2023-51779
- CVE-2023-51780
- CVE-2023-52340
- **CVE-2023-52429**
- **CVE-2023-52433**
- CVE-2023-52434
- **CVE-2023-52436**
- **CVE-2023-52438**
- CVE-2023-52439
- CVE-2023-52445
- CVE-2023-52448
- CVE-2023-52451
- CVE-2023-52458
- CVE-2023-52463
- CVE-2023-52464
- CVE-2023-52469
- CVE-2023-52476
- CVE-2023-52477

- CVE-2023-52486
- CVE-2023-52489
- CVE-2023-52513
- CVE-2023-52520
- CVE-2023-52522
- CVE-2023-52528
- CVE-2023-52529
- CVE-2023-52574
- CVE-2023-52578
- CVE-2023-52580
- CVE-2023-52581
- **CVE-2023-52585**
- CVE-2023-52594
- CVE-2023-52595
- CVE-2023-52598
- CVE-2023-52606
- CVE-2023-52607
- CVE-2023-52610
- CVE-2023-52615
- CVE-2023-52619
- CVE-2023-52620
- CVE-2023-52622
- CVE-2023-52623
- CVE-2023-52628
- **CVE-2023-52635**
- CVE-2023-52638
- CVE-2023-52662
- CVE-2023-52667
- CVE-2023-52669
- CVE-2023-52675
- CVE-2023-52679
- CVE-2023-52686
- CVE-2023-52703
- CVE-2023-52707
- CVE-2023-52730
- CVE-2023-52762
- CVE-2023-52764
- CVE-2023-52775

- CVE-2023-52781
- CVE-2023-52784
- CVE-2023-52791
- CVE-2023-52796
- CVE-2023-52803
- CVE-2023-52809
- CVE-2023-52811
- CVE-2023-52813
- CVE-2023-52832
- CVE-2023-52834
- CVE-2023-52835
- CVE-2023-52845
- CVE-2023-52847
- CVE-2023-52864
- CVE-2023-52877
- CVE-2023-52880
- CVE-2023-52881
- **CVE-2023-52882**
- **CVE-2023-52885**
- CVE-2024-0193
- CVE-2024-0340
- **CVE-2024-0562**
- CVE-2024-0565
- CVE-2024-0607
- **CVE-2024-0639**
- **CVE-2024-0641**
- CVE-2024-0775
- CVE-2024-1085
- CVE-2024-1086
- CVE-2024-2201
- CVE-2024-21823
- **CVE-2024-23850**
- **CVE-2024-23851**
- CVE-2024-25739
- CVE-2024-26581
- CVE-2024-26583
- CVE-2024-26584
- CVE-2024-26586

- **CVE-2024-26589**
- **CVE-2024-26591**
- **CVE-2024-26592**
- CVE-2024-26593
- **CVE-2024-26594**
- **CVE-2024-26597**
- **CVE-2024-26598**
- CVE-2024-26600
- **CVE-2024-26601**
- CVE-2024-26602
- **CVE-2024-26606**
- **CVE-2024-26608**
- CVE-2024-26610
- CVE-2024-26614
- CVE-2024-26615
- **CVE-2024-26622**
- **CVE-2024-26625**
- **CVE-2024-26627**
- CVE-2024-26629
- **CVE-2024-26631**
- CVE-2024-26633
- CVE-2024-26635
- **CVE-2024-26636**
- CVE-2024-26640
- **CVE-2024-26641**
- CVE-2024-26642
- CVE-2024-26643
- **CVE-2024-26644**
- CVE-2024-26645
- CVE-2024-26651
- **CVE-2024-26654**
- CVE-2024-26659
- CVE-2024-26660
- **CVE-2024-26663**
- CVE-2024-26664
- CVE-2024-26665
- CVE-2024-26668
- CVE-2024-26671

- CVE-2024-26673
- CVE-2024-26675
- **CVE-2024-26676**
- CVE-2024-26679
- **CVE-2024-26684**
- **CVE-2024-26685**
- **CVE-2024-26687**
- **CVE-2024-26688**
- **CVE-2024-26689**
- **CVE-2024-26695**
- **CVE-2024-26696**
- **CVE-2024-26697**
- CVE-2024-26698
- **CVE-2024-26702**
- CVE-2024-26704
- **CVE-2024-26707**
- **CVE-2024-26712**
- **CVE-2024-26715**
- CVE-2024-26717
- **CVE-2024-26722**
- **CVE-2024-26727**
- CVE-2024-26735
- **CVE-2024-26736**
- CVE-2024-26737
- CVE-2024-26743
- CVE-2024-26744
- **CVE-2024-26747**
- **CVE-2024-26748**
- **CVE-2024-26749**
- **CVE-2024-26750**
- **CVE-2024-26751**
- **CVE-2024-26752**
- **CVE-2024-26754**
- **CVE-2024-26763**
- **CVE-2024-26764**
- **CVE-2024-26766**
- CVE-2024-26769
- **CVE-2024-26771**

- CVE-2024-26772
- CVE-2024-26773
- **CVE-2024-26774**
- **CVE-2024-26776**
- **CVE-2024-26777**
- CVE-2024-26778
- CVE-2024-26779
- **CVE-2024-26780**
- **CVE-2024-26781**
- **CVE-2024-26782**
- **CVE-2024-26787**
- **CVE-2024-26788**
- **CVE-2024-26790**
- **CVE-2024-26791**
- **CVE-2024-26793**
- **CVE-2024-26795**
- CVE-2024-26801
- CVE-2024-26802
- **CVE-2024-26803**
- CVE-2024-26804
- CVE-2024-26805
- CVE-2024-26808
- **CVE-2024-26809**
- CVE-2024-26810
- **CVE-2024-26811**
- **CVE-2024-26812**
- **CVE-2024-26813**
- **CVE-2024-26814**
- **CVE-2024-26817**
- **CVE-2024-26820**
- **CVE-2024-26825**
- CVE-2024-26826
- **CVE-2024-26829**
- **CVE-2024-26833**
- **CVE-2024-26834**
- **CVE-2024-26835**
- **CVE-2024-26838**
- **CVE-2024-26839**

- CVE-2024-26840
- CVE-2024-26843
- **CVE-2024-26845**
- CVE-2024-26846
- **CVE-2024-26848**
- **CVE-2024-26851**
- CVE-2024-26852
- CVE-2024-26855
- **CVE-2024-26856**
- **CVE-2024-26857**
- CVE-2024-26859
- **CVE-2024-26861**
- **CVE-2024-26862**
- **CVE-2024-26863**
- CVE-2024-26870
- CVE-2024-26872
- **CVE-2024-26874**
- **CVE-2024-26875**
- **CVE-2024-26877**
- CVE-2024-26878
- **CVE-2024-26879**
- CVE-2024-26880
- **CVE-2024-26881**
- **CVE-2024-26882**
- **CVE-2024-26883**
- **CVE-2024-26884**
- **CVE-2024-26885**
- **CVE-2024-26889**
- **CVE-2024-26891**
- CVE-2024-26894
- **CVE-2024-26895**
- CVE-2024-26897
- **CVE-2024-26898**
- **CVE-2024-26900**
- CVE-2024-26901
- CVE-2024-26903
- CVE-2024-26906
- CVE-2024-26907

- **CVE-2024-26910**
- **CVE-2024-26915**
- **CVE-2024-26922**
- CVE-2024-26923
- **CVE-2024-26924**
- CVE-2024-26925
- **CVE-2024-26926**
- CVE-2024-26929
- CVE-2024-26931
- CVE-2024-26934
- **CVE-2024-26935**
- **CVE-2024-26936**
- **CVE-2024-26937**
- **CVE-2024-26950**
- **CVE-2024-26951**
- **CVE-2024-26954**
- **CVE-2024-26955**
- **CVE-2024-26956**
- **CVE-2024-26957**
- CVE-2024-26958
- CVE-2024-26960
- CVE-2024-26961
- CVE-2024-26964
- **CVE-2024-26965**
- **CVE-2024-26966**
- **CVE-2024-26969**
- **CVE-2024-26970**
- **CVE-2024-26972**
- CVE-2024-26973
- CVE-2024-26974
- **CVE-2024-26976**
- **CVE-2024-26977**
- **CVE-2024-26979**
- **CVE-2024-26980**
- **CVE-2024-26981**
- **CVE-2024-26984**
- **CVE-2024-26988**
- **CVE-2024-26989**

- CVE-2024-26993
- **CVE-2024-26994**
- **CVE-2024-26996**
- **CVE-2024-26997**
- **CVE-2024-26999**
- **CVE-2024-27000**
- **CVE-2024-27001**
- **CVE-2024-27004**
- **CVE-2024-27008**
- **CVE-2024-27009**
- CVE-2024-27013
- **CVE-2024-27015**
- CVE-2024-27016
- **CVE-2024-27018**
- CVE-2024-27019
- CVE-2024-27020
- **CVE-2024-27028**
- CVE-2024-27030
- **CVE-2024-27034**
- **CVE-2024-27037**
- **CVE-2024-27038**
- **CVE-2024-27039**
- **CVE-2024-27043**
- **CVE-2024-27044**
- **CVE-2024-27045**
- CVE-2024-27046
- **CVE-2024-27047**
- CVE-2024-27052
- **CVE-2024-27053**
- **CVE-2024-27054**
- CVE-2024-27059
- CVE-2024-27065
- **CVE-2024-27073**
- **CVE-2024-27074**
- **CVE-2024-27075**
- **CVE-2024-27076**
- **CVE-2024-27077**
- **CVE-2024-27078**

- CVE-2024-27388
- **CVE-2024-27390**
- CVE-2024-27393
- CVE-2024-27395
- **CVE-2024-27396**
- **CVE-2024-27398**
- **CVE-2024-27399**
- **CVE-2024-27401**
- **CVE-2024-27403**
- **CVE-2024-27405**
- CVE-2024-27410
- **CVE-2024-27412**
- **CVE-2024-27413**
- **CVE-2024-27414**
- CVE-2024-27415
- **CVE-2024-27416**
- **CVE-2024-27419**
- **CVE-2024-27431**
- **CVE-2024-27432**
- **CVE-2024-27436**
- **CVE-2024-27437**
- CVE-2024-33621
- **CVE-2024-33847**
- **CVE-2024-34027**
- **CVE-2024-34777**
- **CVE-2024-35247**
- **CVE-2024-35785**
- CVE-2024-35789
- CVE-2024-35791
- **CVE-2024-35796**
- **CVE-2024-35804**
- **CVE-2024-35805**
- **CVE-2024-35806**
- CVE-2024-35807
- CVE-2024-35809
- **CVE-2024-35811**
- **CVE-2024-35813**
- **CVE-2024-35815**

- **CVE-2024-35817**
- **CVE-2024-35819**
- **CVE-2024-35822**
- CVE-2024-35823
- **CVE-2024-35825**
- **CVE-2024-35828**
- **CVE-2024-35829**
- **CVE-2024-35830**
- **CVE-2024-35833**
- CVE-2024-35835
- **CVE-2024-35837**
- **CVE-2024-35840**
- **CVE-2024-35844**
- CVE-2024-35845
- CVE-2024-35847
- CVE-2024-35848
- **CVE-2024-35849**
- **CVE-2024-35851**
- CVE-2024-35852
- CVE-2024-35853
- CVE-2024-35854
- CVE-2024-35855
- CVE-2024-35857
- **CVE-2024-35871**
- **CVE-2024-35872**
- CVE-2024-35877
- **CVE-2024-35879**
- CVE-2024-35884
- CVE-2024-35885
- CVE-2024-35886
- CVE-2024-35888
- CVE-2024-35890
- CVE-2024-35893
- CVE-2024-35896
- CVE-2024-35897
- CVE-2024-35898
- CVE-2024-35899
- CVE-2024-35900

- **CVE-2024-35905**
- CVE-2024-35907
- CVE-2024-35910
- CVE-2024-35912
- **CVE-2024-35915**
- **CVE-2024-35918**
- CVE-2024-35922
- CVE-2024-35925
- **CVE-2024-35927**
- CVE-2024-35930
- **CVE-2024-35933**
- **CVE-2024-35934**
- CVE-2024-35935
- **CVE-2024-35936**
- CVE-2024-35938
- **CVE-2024-35940**
- CVE-2024-35944
- CVE-2024-35947
- **CVE-2024-35950**
- **CVE-2024-35955**
- CVE-2024-35958
- CVE-2024-35960
- CVE-2024-35962
- CVE-2024-35969
- **CVE-2024-35970**
- **CVE-2024-35973**
- CVE-2024-35976
- CVE-2024-35978
- CVE-2024-35982
- **CVE-2024-35984**
- **CVE-2024-35988**
- CVE-2024-35989
- **CVE-2024-35990**
- **CVE-2024-35995**
- **CVE-2024-35996**
- **CVE-2024-35997**
- CVE-2024-36004
- CVE-2024-36005

- CVE-2024-36006
- CVE-2024-36007
- **CVE-2024-36008**
- CVE-2024-36014
- CVE-2024-36015
- CVE-2024-36016
- CVE-2024-36017
- CVE-2024-36020
- CVE-2024-36025
- **CVE-2024-36029**
- **CVE-2024-36031**
- CVE-2024-36032
- CVE-2024-36033
- CVE-2024-36270
- CVE-2024-36286
- CVE-2024-36288
- CVE-2024-36484
- CVE-2024-36489
- **CVE-2024-36880**
- CVE-2024-36883
- CVE-2024-36886
- CVE-2024-36889
- CVE-2024-36894
- **CVE-2024-36897**
- CVE-2024-36901
- CVE-2024-36902
- CVE-2024-36904
- CVE-2024-36905
- **CVE-2024-36906**
- **CVE-2024-36916**
- CVE-2024-36919
- **CVE-2024-36928**
- CVE-2024-36929
- **CVE-2024-36931**
- CVE-2024-36933
- CVE-2024-36934
- **CVE-2024-36937**
- CVE-2024-36939

- CVE-2024-36940
- CVE-2024-36941
- **CVE-2024-36942**
- CVE-2024-36946
- **CVE-2024-36947**
- CVE-2024-36950
- CVE-2024-36952
- CVE-2024-36953
- CVE-2024-36954
- **CVE-2024-36955**
- CVE-2024-36957
- **CVE-2024-36959**
- CVE-2024-36960
- **CVE-2024-36964**
- **CVE-2024-36965**
- **CVE-2024-36967**
- **CVE-2024-36969**
- CVE-2024-36971
- **CVE-2024-36972**
- CVE-2024-36974
- **CVE-2024-36975**
- CVE-2024-36978
- CVE-2024-37078
- CVE-2024-37353
- CVE-2024-37356
- **CVE-2024-38381**
- **CVE-2024-38546**
- **CVE-2024-38547**
- **CVE-2024-38548**
- CVE-2024-38549
- **CVE-2024-38550**
- CVE-2024-38552
- CVE-2024-38555
- CVE-2024-38558
- CVE-2024-38559
- CVE-2024-38560
- CVE-2024-38565
- CVE-2024-38567

- **CVE-2024-38571**
- CVE-2024-38573
- CVE-2024-38578
- CVE-2024-38579
- CVE-2024-38580
- CVE-2024-38582
- CVE-2024-38583
- CVE-2024-38586
- **CVE-2024-38587**
- CVE-2024-38588
- CVE-2024-38589
- **CVE-2024-38590**
- **CVE-2024-38591**
- CVE-2024-38596
- **CVE-2024-38597**
- CVE-2024-38598
- CVE-2024-38599
- **CVE-2024-38600**
- CVE-2024-38601
- **CVE-2024-38605**
- **CVE-2024-38610**
- CVE-2024-38612
- CVE-2024-38613
- CVE-2024-38615
- CVE-2024-38618
- CVE-2024-38619
- CVE-2024-38621
- **CVE-2024-38623**
- **CVE-2024-38624**
- CVE-2024-38627
- CVE-2024-38633
- CVE-2024-38634
- CVE-2024-38635
- CVE-2024-38637
- CVE-2024-38659
- CVE-2024-38661
- **CVE-2024-38662**
- CVE-2024-38780

- CVE-2024-39276
- **CVE-2024-39277**
- CVE-2024-39292
- CVE-2024-39301
- CVE-2024-39362
- **CVE-2024-39466**
- CVE-2024-39467
- CVE-2024-39468
- CVE-2024-39469
- CVE-2024-39471
- CVE-2024-39475
- CVE-2024-39476
- CVE-2024-39480
- CVE-2024-39482
- CVE-2024-39484
- CVE-2024-39487
- CVE-2024-39488
- CVE-2024-39489
- **CVE-2024-39490**
- **CVE-2024-39493**
- CVE-2024-39495
- CVE-2024-39499
- CVE-2024-39500
- CVE-2024-39501
- CVE-2024-39502
- CVE-2024-39503
- CVE-2024-39505
- CVE-2024-39506
- CVE-2024-39507
- CVE-2024-39509
- CVE-2024-40901
- CVE-2024-40902
- CVE-2024-40904
- CVE-2024-40905
- CVE-2024-40908
- CVE-2024-40911
- CVE-2024-40912
- CVE-2024-40914

- CVE-2024-40916
- CVE-2024-40927
- CVE-2024-40929
- CVE-2024-40931
- CVE-2024-40932
- CVE-2024-40934
- CVE-2024-40937
- CVE-2024-40941
- CVE-2024-40942
- CVE-2024-40943
- CVE-2024-40945
- CVE-2024-40947
- CVE-2024-40956
- CVE-2024-40957
- CVE-2024-40958
- CVE-2024-40959
- CVE-2024-40960
- CVE-2024-40961
- CVE-2024-40963
- CVE-2024-40967
- CVE-2024-40968
- CVE-2024-40970
- CVE-2024-40971
- CVE-2024-40974
- CVE-2024-40976
- CVE-2024-40978
- CVE-2024-40980
- CVE-2024-40981
- CVE-2024-40983
- CVE-2024-40987
- CVE-2024-40988
- CVE-2024-40990
- CVE-2024-40993
- CVE-2024-40994
- CVE-2024-40995
- CVE-2024-41000
- CVE-2024-41002
- CVE-2024-41005

- CVE-2024-41006
- CVE-2024-41007
- CVE-2024-41034
- CVE-2024-41035
- CVE-2024-41040
- CVE-2024-41041
- CVE-2024-41044
- CVE-2024-41046
- CVE-2024-41047
- CVE-2024-41048
- CVE-2024-41049
- CVE-2024-41055
- CVE-2024-41087
- CVE-2024-41089
- CVE-2024-41090
- CVE-2024-41091
- CVE-2024-41092
- CVE-2024-41093
- CVE-2024-41095
- CVE-2024-41097
- CVE-2024-42068
- CVE-2024-42069
- CVE-2024-42070
- CVE-2024-42076
- CVE-2024-42077
- CVE-2024-42080
- CVE-2024-42082
- CVE-2024-42084
- CVE-2024-42085
- CVE-2024-42086
- CVE-2024-42087
- CVE-2024-42089
- CVE-2024-42090
- CVE-2024-42092
- CVE-2024-42093
- CVE-2024-42094
- CVE-2024-42095
- CVE-2024-42096

- CVE-2024-42097
- CVE-2024-42098
- CVE-2024-42101
- CVE-2024-42103
- CVE-2024-42104
- CVE-2024-42105
- CVE-2024-42106
- CVE-2024-42109
- CVE-2024-42115
- CVE-2024-42116
- CVE-2024-42119
- CVE-2024-42120
- CVE-2024-42121
- CVE-2024-42124
- CVE-2024-42127
- CVE-2024-42130
- CVE-2024-42131
- CVE-2024-42137
- CVE-2024-42140
- CVE-2024-42143
- CVE-2024-42145
- CVE-2024-42148
- CVE-2024-42152
- CVE-2024-42153
- CVE-2024-42154
- CVE-2024-42157
- CVE-2024-42161
- CVE-2024-42223
- CVE-2024-42224
- CVE-2024-42225
- CVE-2024-42229
- CVE-2024-42232
- CVE-2024-42236
- CVE-2024-42244
- CVE-2024-42247