Oracle Linux

Ksplice User's Guide

E39380-64
January 2024

Oracle Linux Ksplice User's Guide,

E39380-64

Contents

Preface

1 About Oracle Ksplice

2 Preparing to Use Oracle Ksplice

3 Using the Ksplice Enhanced Client

Preface

Oracle Linux: Ksplice User's Guide provides information about how to install, configure, and use Oracle Ksplice to update kernel, user space, and Xen hypervisor packages on a running system and how to use the Ksplice Uptrack API.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at https://www.oracle.com/corporate/accessibility/templates/t2-11535.html.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Oracle Ksplice

Oracle Ksplice updates select, critical components of your Linux installation with all important security patches without needing to reboot.

Ksplice is freely available for Oracle customers who subscribe to Oracle Linux Premier Support and Oracle Cloud Infrastructure services. If you are an Oracle Linux Basic, Basic Limited, or Network Support subscriber, contact your sales representatives to discuss a potential upgrade of your subscription to a Premier Support plan.

> **Important:**
>
> **About instructions in this guide**
>
> - Some examples use the `yum` command. For Oracle Linux 8 and Oracle Linux 9, use the `dnf` command, as appropriate.
> - Most of this guide *only* applies to Oracle Linux. To use Ksplice to patch the Xen hypervisor on Oracle VM Server 3.4.5 and later, refer to the corresponding Oracle VM documentation. For example, for Oracle VM 3.4.5, see *Updating Oracle VM Server With Oracle Ksplice* in the Oracle VM Administration Guide for Release 3.4.

Why Use Ksplice?

Ksplice can apply critical updates without rebooting. Traditionally, applying security updates to core operating system components requires you to manually install updated RPMs, schedule downtime, and reboot the server. Ksplice allows you to keep your systems secure and highly available by updating a running system with the latest kernel and key user space updates, as well as Xen hypervisor updates on Oracle VM Server 3.4.5, and later (minimum `xen-4.4.4-196.el6.x86_64.rpm`).

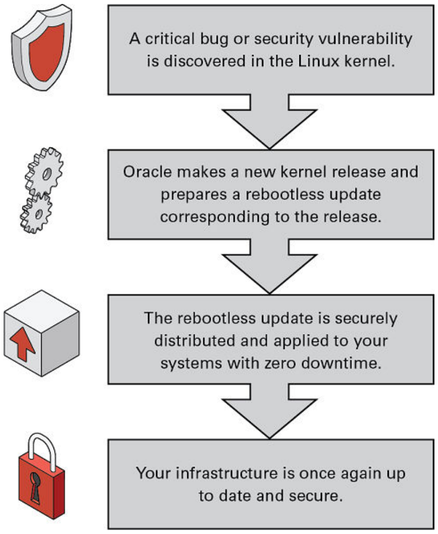
Ksplice rebootless updates:

- Save time and hassle by updating in seconds, while your system is running.

- Avoid downtime.
- Prevent disastrous security incidents by making it easy to stay up to date.

Life Cycle of a Ksplice Update

When a critical bug or security vulnerability is discovered in the Linux kernel, Oracle produces a new kernel release and prepares a rebootless update corresponding to that release. The rebootless update is securely distributed by using the Ksplice Uptrack server and ULN. The Ksplice Enhanced or Ksplice Uptrack Client then applies this update to your system, with zero downtime. Your infrastructure is again up to date and secure.


A critical bug or security vulnerability is discovered in the Linux kernel.
Oracle makes a new kernel release and prepares a rebootless update corresponding to the release.
The rebootless update is securely distributed and applied to your systems with zero downtime.
Your infrastructure is once again up to date and secure.

In-Memory vs. On-Disk Updates

A Ksplice update occurs in memory and takes effect immediately upon application, which is different than an on-disk change that requires a reboot. However, you must continue to apply on-disk updates, even when using Ksplice, to ensure that updated package binaries can be used if the system or processes restart. On-disk updates are handled by subscribing to Unbreakable Linux Network (ULN) or by using a local ULN mirror.

Ksplice patches keep a system up to date while it is running, but you must continue to install the regular kernel packages for released errata from ULN or the Oracle Linux Yum server so that the kernel is also updated on disk. Your system is then ready for the next maintenance window or reboot. When you restart the system, you can boot it from the newer kernel version. Ksplice then uses the new kernel as a baseline for applying patches when they become available.

Available Architectures

Ksplice is available for the following platforms:

- Intel 64-bit (x86_64)

- AMD 64-bit (x86_64)
- 64-bit Arm (aarch64)

> **Note:**
>
> Ksplice on the 64-bit Arm (aarch64) platform is only available with maintained Unbreakable Enterprise Kernel (UEK) releases. For more information, see the UEK release notes in the Unbreakable Enterprise Kernel documentation

Maintained Kernels

Only specific kernel versions are actively maintained by Ksplice.

- Kernels Actively Maintained With Ksplice
- Kernels No Longer Actively Maintained With Ksplice

> **Note:**
>
> Ksplice on Oracle Cloud Infrastructure supports specific Linux distributions. For more information, see Oracle Ksplice on Oracle Cloud Infrastructure.

For questions about supported kernels, send an email to ksplice-support_ww@oracle.com.

Kernels Actively Maintained With Ksplice

With Oracle Linux Premier Support or Premier Limited subscriptions, you can use Ksplice to bring various Linux kernels up-to-date with the latest important security and bug fix patches. The following table shows the distributions and kernel versions that are automatically maintained with Ksplice.

> **Note:**
>
> If the system is running RHEL and you recently migrated to Oracle Linux Premier Support, you must switch to RHCK to use Ksplice kernel patches. Oracle no longer maintains Ksplice patches for RHEL kernels.

Actively Maintained Kernel Type	More Information
UEK R7 (aarch64) starting with `5.15.0-0.30.19` (released Jun 30, 2022).	
UEK R7 (x86_64) starting with `5.15.0-0.30.19` (released Jun 30, 2022).	
UEK R6 (aarch64) starting with `5.4.17-2011.0.7` (released Mar 17, 2020).	

Actively Maintained Kernel Type	More Information
UEK R6 (x86_64) starting with `5.4.17-2011.1.2` (released Apr 27, 2020).	
UEK R5 (aarch64) starting with `4.14.35-1902.300.11` (released Mar 18, 2020).	
UEK R5 (x86_64) starting with `4.14.35-1818.0.9` (released Jun 20, 2018).	
UEK R4 starting with `4.1.12-32` (released Jan 25, 2016).	Must be version `v4.1.12-124.45.6` or later to be actively maintained with Ksplice on Oracle Linux 6. See Kernels No Longer Actively Maintained With Ksplice for more information.
Oracle Linux 9 Red Hat Compatible Kernels (RHCK) starting with the official release.	
Oracle Linux 8 Red Hat Compatible Kernels (RHCK) starting with the official release.	
Oracle Linux 7 Red Hat Compatible Kernels (RHCK) starting with the official release.	
Oracle Linux 6 Red Hat Compatible Kernels (RHCK) starting with the official release.	Must be version `2.6.32-754.35.1` or later to be actively maintained with Ksplice on Oracle Linux 6. See Kernels No Longer Actively Maintained With Ksplice for more information.
Ubuntu 22.04 Jammy kernels, starting with the official release.	
Ubuntu 20.04 Focal kernels starting with `5.4.0-37.41` (released Jun 3, 2020).	

Important:

If you have booted the most recent available kernel and no Ksplice updates are available for that kernel, some Ksplice commands might fail or might return an error message notifying you that the kernel version isn't yet supported by Ksplice Uptrack. These commands only succeed when Ksplice updates are available for the kernel that's running on the system. As soon as an update becomes available, the command succeeds, and the update is applied.

Kernels No Longer Actively Maintained With Ksplice

The following kernels don't receive Ksplice updates, but any Ksplice updates previously issued are still available if you have a support contract.

To maintain any of the following kernels on a listed Linux distribution, you need to manually upgrade them by using the `yum update` or `dnf update` command, or in the case of Ubuntu, by using the `apt` command. Kernel updates that don't use Ksplice require system reboots to be effective.

If you're an Extended Support customer who is running any of these kernel types on either Oracle Linux 6 or Oracle Linux 7, update to the minimum version of UEK R4.

Kernel Type	Kernel Version	Releases No Longer Actively Maintained
UEK R4	Versions earlier than `v4.1.12-124.45.6`	Oracle Linux 6
UEK R3	All Versions	Oracle Linux 6 Oracle Linux 7
UEK R2	All versions	Oracle Linux 6
RHCK	Versions earlier than `2.6.32-754.35.1`	Oracle Linux 6
Kernels shipped with RHEL 9.	All versions	RHEL 9
CentOS and RHEL 8 kernels.	All versions	RHEL or CentOS Linux 8
CentOS and RHEL 7 kernels.	All versions	RHEL or CentOS Linux 7
Kernels shipped in RHEL/CentOS Linux 6	All versions	RHEL or CentOS Linux 6
Kernels shipped in Ubuntu 18.04 LTS.	All versions	Ubuntu 18.04 LTS (Bionic Beaver)
Kernels shipped in Ubuntu 16.04 LTS	All versions	Ubuntu 16.04 LTS (Xenial Xerus)

About the Ksplice Inspector Tool

Use Ksplice Inspector, which is a free, online tool that lists available Ksplice updates for Maintained Kernels.

Ksplice Inspector helps you determine what updates are available for your currently running kernel and what updates can be automatically applied in-memory by using either the Ksplice Enhanced client or the Ksplice Uptrack client. The tool enables you to proactively identify security vulnerabilities, which is a critical step in assessing potential cybersecurity issues. The tool is publicly available and does not require a support subscription.

To get started using the tool, open a terminal on the Linux system that you want to check, and then run the following command:

```
echo "`uname -s`//`uname -m`//`uname -r`//`uname -v`"
```

Copy the output of the previous command into the Ksplice Inspector check box, and then click **Find Updates**.

The tool indicates what security patches are already available to Ksplice customers.

Oracle Cloud Infrastructure Ksplice Support

You can monitor and manage automatic updates for Oracle Linux systems that are running within Oracle Cloud infrastructure by using Ksplice.

Note the following key points about receiving automatic Ksplice updates on systems that are running within Oracle Cloud Infrastructure:

- By default, Ksplice configuration is shipped with the Oracle Cloud Infrastructure platform images by preconfiguring the Ksplice yum repositories and Ksplice online server URL.

- Bring Your Own Image (BYOI) configurations can use the same yum repository configuration file as the platform images (`/etc/yum.repos.d/ksplice-ol*.repo`), if copied there manually.

  > **Note:**
  >
  > The `/etc/yum.repos.d/ksplice-ol`*N*`.repo` file comes from the `ksplice-release-el`*N* RPM, which is in the yum repository that is configured by `oci_included_ol`*N*`.repo` and is part of the `oci-included-release-el`*N* package (`/etc/yum.repos.d/oci-included-ol`*N*`.repo`).

- Systems running within Oracle Cloud Infrastructure that have the Ksplice client configured in *online* mode do not need to be registered with ULN to access the Ksplice servers and receive automatic updates.
- Systems running within Oracle Cloud Infrastructure that have the Ksplice client configured in *offline* mode do not need to be registered with ULN, nor do they require a local ULN mirror configuration to receive automatic updates.

For further information, see Oracle Ksplice on Oracle Cloud Infrastructure.

Oracle Enterprise Manager Ksplice Support

All Oracle Linux systems on which Enterprise Manager Agent is installed and the Ksplice software is configured can be monitored and managed through Oracle Enterprise Manager, within the Oracle Linux Home Ksplice region of the Enterprise Manager user interface (UI).

To learn more about using Oracle Enterprise Manager to monitor and use Ksplice patching on Oracle Linux hosts, see the Oracle Enterprise Manager Life Cycle Management Administrator's Guide.

2

Preparing to Use Oracle Ksplice

The prerequisites for using Oracle Ksplice depend on which client you select and whether you will be using online or offline mode.

Before using Ksplice:

- Choose a Ksplice client
  - Select either the *Ksplice Enhanced Client* or the *Ksplice Uptrack Client*.
  - Determine if you're using the client in *online* or *offline* mode.
- Register your system with ULN (if using online mode)
- Configure a local Ksplice mirror (if using offline mode)

Choosing a Ksplice Client

You have the option to choose between the Ksplice Enhanced Client and the Ksplice Uptrack Client.

Table 2-1 Features Supported by Each Ksplice Client

Ksplice Client	User Space Support	x86_64 Support	Arm (aarch64) Support	Xen Hypervisor Patching Support	Known Exploit Detection Support	Legacy Compatibility (Pre-acquisition customers)
Ksplice Enhanced Client	Supported	Supported	Supported	Supported on x86_64 platform *only*	Supported on x86_64 platform *only*	Not supported
Ksplice Uptrack Client	Not supported	Supported	Supported	Not Supported	Not supported	Supported

For legacy compatibility, Oracle continues to support kernels for various Linux distributions for pre-acquisition customers. For more information, see https://ksplice.oracle.com/legacy#supported-kernels.

About the Ksplice Enhanced Client

The Ksplice Enhanced Client provides additional functionality over the Ksplice Uptrack Client.

In addition to the kernel updates that are applied by the Uptrack Client, the Enhanced Client can patch in-memory pages for the Ksplice-aware `glibc` and `openssl` shared libraries for user space processes. User space patching enables you to install bug fixes and protect your system against security vulnerabilities, without having to restart processes and services.

Key features of the Enhanced Client include:

- Kernel and user space updates (the Uptrack Client only supports kernel updates)
- Patching of Xen hypervisor on Oracle VM Server Release 3.4.5, and later (requires minimum `xen-4.4.4-196.el6.x86_64.rpm`)
- Known exploit detection
- Online and offline mode
- Use of the `ksplice` command

> **Note:**
>
> The Enhanced Client shares the same configuration file as the Uptrack Client, which is the `/etc/uptrack/uptrack.conf` file. For more information about this file, see Configuring the Ksplice Uptrack Client.

About the Ksplice Uptrack Client

Ksplice Uptrack enables you to apply the latest kernel security errata for Common Vulnerabilities and Exposures (CVEs) without halting the system or restarting any applications. Ksplice Uptrack applies the updated patches in the background with negligible impact, and usually only requires a pause of a few milliseconds.

Key features of the Uptrack Client include:

- Kernel updates (to also apply user space updates, consider the Ksplice Enhanced Client instead)
- Online and offline mode
- Use of the `uptrack` command

About Ksplice Offline Mode

You can use either the Ksplice Enhanced Client or Ksplice Uptrack Client in offline mode. The offline version does not require a direct connection to the Oracle Uptrack server or to ULN. For example, you could use the `yum` command to install an update package directly from a memory stick. However, a more typical method would be to configure a local ULN mirror that acts as a mirror for the Ksplice-aware ULN channels. Then, you can configure your systems to receive `yum` and Ksplice updates.

Oracle bundles all available Ksplice updates for each supported kernel version or user space package into an RPM that is specific to that version. Oracle updates this package every time a new Ksplice patch becomes available for the kernel. At regular intervals, you can download the latest Ksplice update packages to the local ULN server. Then, the Ksplice server can connect to the local server to receive updates without requiring direct access to the Oracle Uptrack server.

Offline mode does not support:

- Ksplice web interface
- Ksplice Uptrack API
- Patching the Xen hypervisor on Oracle VM Server

> **Important:**
>
> If you have booted the most recent available kernel and no Ksplice updates are available, an offline update RPM for that kernel might not yet exist. Offline update RPMs are made available shortly after the kernel releases. However, these RPMs might require additional time to synchronize with the local repository that you have set up.

For more information, see:

- Configuring a Local Ksplice Mirror for the Offline Client
- Configuring the Ksplice Enhanced Client for Offline Mode
- Configuring Ksplice Uptrack Clients for Offline Mode
- Switching Between Online and Offline Ksplice Uptrack Installation Modes

Registering With ULN

To use Ksplice to apply automatic updates, your system must have access to the Internet and it must be registered with ULN.

Systems that are configured to use the Ksplice *offline* client must have access to a local ULN mirror to receive automatic updates. For instructions, see Configuring a Local Ksplice Mirror for the Offline Client.

The requirements for systems running within Oracle Cloud Infrastructure are as follows:

- For instances that are configured to use the Ksplice *online* client, your system does not need to be registered with ULN, as these instances are preconfigured for automatic access to the Ksplice servers and all of the Ksplice updates (applies to both the Ksplice online and offline clients).
- For instances that are configured to use the Ksplice *offline* client, you do not need to configure a local ULN mirror to receive automatic updates.

For more information about receiving automatic Ksplice updates on systems that are running within Oracle Cloud Infrastructure, see Oracle Cloud Infrastructure Ksplice Support.

For more information about registering a system with ULN if you are running Oracle Linux 6 or Oracle Linux 7, see Oracle Linux: Unbreakable Linux Network User's Guide for Oracle Linux 6 and Oracle Linux 7.

If you are running Oracle Linux 8 or Oracle Linux 9, see Oracle Linux: Managing Software on Oracle Linux.

After registering, you can install either the Ksplice Enhanced client software or the Ksplice Uptrack client software from Ksplice for Oracle Linux channel on ULN by using the `yum` command. After installation, Oracle allocates the Ksplice client an identification key that associates it with the Customer Support Identifier (CSI) for your account. You can configure your system to automatically receive updates from the Ksplice Uptrack server.

Automatic Registration

Your account is automatically registered to use the Ksplice Uptrack server if you have one of the following:

- Oracle Linux Premier support subscription
- Premier Limited support subscription
- Oracle Premier Support for Systems and Operating Systems subscription
- Systems running within Oracle Cloud Infrastructure

Accessing the Ksplice Uptrack Web Interface

If your account has a valid CSI, you can log in to the Ksplice Uptrack server web interface at https://status-ksplice.oracle.com/status/settings by using your Oracle Account credentials. After logging in to the server, you can view the status of your registered systems, the patches that have been applied, and the patches that are available. You can also create access control groups for your registered systems.

Available Ksplice Channels

The following table describes the channels that are available for Ksplice in Oracle Linux.

Channel Name	Channel Label	Description
Ksplice for Oracle Linux 6 (i386)	`ol6_i386_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 6 on i386 systems.
Ksplice for Oracle Linux 6 (x86_64)	`ol6_x86_64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 6 on x86_64 systems.
Ksplice for Oracle Linux 7 (x86_64)	`ol7_x86_64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 7 on x86_64 systems.
Ksplice for Oracle Linux 7 (aarch64)	`ol7_aarch64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 7 on aarch64 systems.
Ksplice for Oracle Linux 8 (x86_64)	`ol8_x86_64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 8 on x86_64 systems.
Ksplice for Oracle Linux 8 (aarch64)	`ol8_aarch64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 8 on aarch64 systems.
Ksplice for Oracle Linux 9 (x86_64)	`ol9_x86_64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 9 on x86_64 systems.
Ksplice for Oracle Linux 9 (aarch64)	`ol9_aarch64_ksplice`	Ksplice clients, updates, and dependencies for Oracle Linux 9 on aarch64 systems.
Ksplice-aware user space packages for Oracle Linux 6 (x86_64)	`ol6_x86_64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 6 (x86_64). This channel should only be used with the Ksplice Enhanced client.

Channel Name	Channel Label	Description
Ksplice-aware user space packages for Oracle Linux 7 (x86_64)	`ol7_x86_64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 7 (x86_64). This channel should only be used with the Ksplice Enhanced client.
Ksplice-aware user space packages for Oracle Linux 7 (aarch64)	`ol7_aarch64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 7 (aarch64). This channel should only be used with the Ksplice Enhanced client.
Ksplice-aware user space packages for Oracle Linux 8 (x86_64)	`ol8_x86_64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 8 (x86_64). This channel should only be used with the Ksplice Enhanced client.
Ksplice-aware user space packages for Oracle Linux 8 (aarch64)	`ol8_aarch64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 8 (aarch64). This channel should only be used with the Ksplice Enhanced client.
Ksplice-aware user space packages for Oracle Linux 9 (x86_64)	`ol9_x86_64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 9 (x86_64). This channel should only be used with the Ksplice Enhanced client.
Ksplice-aware user space packages for Oracle Linux 9 (aarch64)	`ol9_aarch64_userspace_ksplice`	Latest packages for Ksplice-aware user space packages for Oracle Linux 9 (aarch64). This channel should only be used with the Ksplice Enhanced client.

Configuring a Local Ksplice Mirror for the Offline Client

To use the Ksplice Offline client, you must configure a local Ksplice mirror using either a local ULN mirror or an Oracle Linux Manager server. You can then download the latest Ksplice update packages to this server at regular intervals and configure your other systems to receive both yum and Ksplice updates.

For more information about offline clients, see:

- About Ksplice Offline Mode
- Configuring the Ksplice Enhanced Client for Offline Mode
- Configuring Ksplice Uptrack Clients for Offline Mode
- Switching Between Online and Offline Ksplice Uptrack Installation Modes

Local ULN Mirror

Configure an Oracle Linux host as a local ULN mirror to act as a Ksplice mirror.

Oracle Linux 8 and Oracle Linux 9

The instructions for Oracle Linux 8 and Oracle Linux 9 might vary slightly from the Oracle Linux 7 instructions that follow. For more information, see Oracle Linux: Managing Software on Oracle Linux .

Oracle Linux 7

For more information about setting up a local ULN mirror, see Oracle Linux: Unbreakable Linux Network User's Guide for Oracle Linux 6 and Oracle Linux 7.

1. Log in to https://linux.oracle.com, and provide the ULN user name and password that you used to register your system.
2. On the **Systems** tab, from the list of registered machines, select the link for your system's name.
3. On the System Details page, select **Edit**.
4. On the Edit System Properties page, select the **Yum Server** check box, then apply your changes.
5. On the System Details page, select **Manage Subscriptions**.
6. On the System Summary page, select the appropriate channels from the list of available or subscribed channels, then move the channels between the two lists by using the arrows.
7. Modify the list of subscribed channels to include the Ksplice for Oracle Linux channels that you want to make available to local, offline clients. See Available Ksplice Channels.
8. When you are finished with the channel selection process, save the subscription and log out of ULN.

Oracle Linux Manager Server

To set up an Oracle Linux Manager server to act as a Ksplice mirror, you must configure the repositories and associated software channels for the Oracle Linux releases and architectures of the systems on which you want to run the Ksplice Offline client. Each Ksplice channel should be a child of the appropriate, base software channel. For information about available channels, see Available Ksplice Channels.

You then need to specify the URL for the appropriate Ksplice channel. For the Oracle Linux 7 (x86_64) channel on ULN, you would specify the URL as follows:

```
uln:///ol7_x86_64_ksplice
```

For more information, see the chapter that describes how to use Ksplice with Oracle Linux Manager in Oracle Linux Manager: Client Life Cycle Management Guide.

3
Using the Ksplice Enhanced Client

> **Note:**
>
> Some examples use the yum command. For Oracle Linux 8 or Oracle Linux 9, use the dnf command, as appropriate.

Limitations of the Ksplice Enhanced Client

Be aware of the following important Oracle Ksplice limitations:

- Ksplice reports an error similar to the following if it cannot apply updates to processes that do not have access to the /var/cache/ksplice directory:

  ```
  Ksplice was unable to load the update as the target process is in a
  different mount namespace or has changed root.  The service must be
  restarted to apply on-disk updates.
  Extra information: the process has changed root or mount namespace.
    └─ rtkit-daemon (3680)
  ```

  This error might typically occur with processes that use chroot or those that run in an LXC or Docker container. In such cases, you must restart the process to apply any available updates. For example, to restart the rtkit-daemon service, you would use the systemctl restart rtkit-daemon command.

  To avoid having to restart a chrooted application that you maintain and compile, ensure that the /var/cache/ksplice directory is bind-mounted in the chrooted environment.

- Ksplice cannot patch applications that use either setcontext or swapcontext from glibc to perform user space context switching between process threads.
- Because of certain kernel limitations, Ksplice does not patch the init process (PID 1).

On Oracle Linux 7, the `init` process, which is actually `systemd`, is automatically executed again on system updates, so it does not require patching with Ksplice.

On Oracle Linux 6, Upstart is not capable of executing itself again, so any updates to `glibc` that can affect Upstart might require a reboot.

Installing the Ksplice Enhanced Client From ULN

> **Note:**
>
> If using Oracle Cloud Infrastructure, Ksplice is already installed by default (on all Oracle Linux instances launched after August 25, 2017). For more information, see Oracle Ksplice on Oracle Cloud Infrastructure.

> **Note:**
>
> The following procedure applies *only* to Oracle Linux releases. To use Ksplice to patch the Xen hypervisor on Oracle VM 3.4.5 and later releases, refer to the Oracle VM documentation that corresponds to the release that you are running. For example, if you are running Oracle VM 3.4.5, see *Updating Oracle VM Server With Oracle Ksplice* in the Oracle VM Administration Guide for Release 3.4.

1. Before installing the Enhanced Client:
   - Verify that the system is running Oracle Linux 6, Oracle Linux 7, Oracle Linux 8, or Oracle Linux 9 with a supported version of either the Unbreakable Enterprise Kernel (UEK) or the Red Hat Compatible Kernel (RHCK) installed. Use the `uname -a` command to verify the kernel version. See Maintained Kernels. Ksplice applies updates to the currently running kernel *only*, so ensure that the running kernel is the one you want to update.
   - For an online client, register the system with ULN and verify it has a connection to the Oracle Uptrack server.
   - For an offline client, configure a local ULN mirror.
2. Log in to ULN at https://linux.oracle.com. Provide the ULN user name and password that you used to register the system.
3. Subscribe to the necessary channels:
   a. On the Systems tab, click the link named for your system in the list of registered machines.
   b. On the System Details page, click **Manage Subscriptions**.

      The Ksplice Enhanced client and Ksplice-aware user space packages are available in the following channels on ULN:
      - Ksplice for Oracle Linux 6 (x86_64) (`ol6_x86_64_ksplice`)
      - Ksplice for Oracle Linux 7 (x86_64) (`ol7_x86_64_ksplice`)

- Ksplice for Oracle Linux 7 (aarch64) (`ol7_aarch64_ksplice`)
- Ksplice for Oracle Linux 8 (x86_64) (`ol8_x86_64_ksplice`)
- Ksplice for Oracle Linux 8 (aarch64) (`ol8_aarch64_ksplice`)
- Ksplice for Oracle Linux 9 (x86_64) (`ol9_x86_64_ksplice`)
- Ksplice for Oracle Linux 9 (aarch64) (`ol9_aarch64_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 6 (x86_64) (`ol6_x86_64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 7 (x86_64) (`ol7_x86_64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 7 (aarch64) (`ol7_aarch64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 8 (x86_64) (`ol8_x86_64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 8 (aarach64) (`ol8_aarch64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 9 (x86_64) (`ol9_x86_64_userspace_ksplice`)
- Ksplice-aware user space packages for Oracle Linux 9 (aarach64) (`ol9_aarch64_userspace_ksplice`)

c. On the System Summary page, select both the Ksplice user space and Ksplice channels from the list of available channels, then click the right arrow (**>**) to move them to the list of subscribed channels.

d. Accept the licensing terms for the Ksplice Enhanced client packages.

e. Save the subscription and log out of ULN.

4. If you use an Internet proxy, configure the HTTP and HTTPS settings for the proxy in the shell as follows:

   - For the `sh`, `ksh`, or `bash` shells, use commands such as the following:

     ```
     sudo http_proxy=http://proxy_URL:http_port
     sudo https_proxy=http://proxy_URL:https_port
     sudo export http_proxy https_proxy
     ```

     For the `csh` shell, use commands such as the following:

     ```
     sudo setenv http_proxy=http://proxy_URL:http_port
     sudo setenv https_proxy=http://proxy_URL:https_port
     ```

5. Log in to the system as the `root` user.

6. If `prelink` is installed, revert all of the prelinked binaries and any dependent libraries to their original state, then remove the `prelink` package:

   ```
   sudo prelink -au
   sudo yum remove prelink
   ```

> **Note:**
>
> `prelink` is installed and enabled by default on Oracle Linux 6, but not Oracle Linux 7, Oracle Linux 8 or Oracle Linux 9.

7. Install the `ksplice` package:

   - For the Ksplice online client, use the following command:

     ```
     sudo yum install -y ksplice uptrack
     ```

   - For the Ksplice offline client, use the following command:

     ```
     sudo yum install -y ksplice ksplice-offline uptrack-offline
     ```

   The access key for Ksplice Uptrack is retrieved from ULN and added to the `/etc/uptrack/uptrack.conf` file, as shown in the following example:

   ```
   [Auth]
   accesskey = 0e1859ad8aea14b0b4306349142ce9160353297daee30240dab4d61f4ea4e59b
   ```

   The following packages are installed on the system:

   `ksplice-core`
   Contains the shared user space libraries, such as `glibc` and `openssl`, that support Ksplice patching.

   `ksplice-helper`
   Contains a helper library that enables user space executables to be patched by Ksplice.

   `ksplice-helper-devel`
   Contains the development environment for creating user space libraries that support Ksplice patching.

   `ksplice-tools`
   Contains the `ksplice` executable and `ksplice(8)` manual page.

8. Update the system to install the Ksplice-aware versions of the user space libraries:

   ```
   sudo yum update
   ```

   To install just the libraries and not update any other packages, limit the update to the following channels, as appropriate:

   - `ol6_x86_64_userspace_ksplice`
   - `ol7_x86_64_userspace_ksplice`
   - `ol7_aarch64_userspace_ksplice`
   - `ol8_x86_64_userspace_ksplice`
   - `ol8_aarch64_userspace_ksplice`
   - `ol9_x86_64_userspace_ksplice`
   - `ol9_aarch64_userspace_ksplice`

   For example, you would update the packages for the Oracle Linux 7 Ksplice user-aware x86_64 channels as follows:

```
sudo yum --disablerepo=* --enablerepo=ol7_x86_64_userspace_ksplice update
```

You can also use the `glibc*` and `openssl*` syntax with the `install` command for your package manager. To use this client to perform kernel updates, install it in the same way that you are able to use the standard Uptrack client, for example:

```
sudo yum install uptrack-updates-`uname -r`
```

9. To enable the automatic installation of updates, change the entry in the `/etc/uptrack/uptrack.conf` file from `no` to `yes`:

   ```
   autoinstall = yes
   ```

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

    ```
    sudo systemctl reboot
    ```

    For Oracle Linux 6, use the following command:

    ```
    sudo reboot
    ```

The Kpslice Enhanced client uses the same configuration file (`/etc/uptrack/uptrack.conf`) as the Ksplice Uptrack client. See Configuring the Ksplice Uptrack Client.

To manage the Ksplice Enhanced client, use the `ksplice` command. See Using the ksplice Command to Manage the Ksplice Enhanced Client.

Using the ksplice Command to Manage the Ksplice Enhanced Client

You manage the Ksplice Enhanced client by using the `ksplice` command. Use this command instead of the `uptrack` commands that are used with the traditional Ksplice Uptrack client. The `ksplice` command enables you to perform user space patching, in addition to kernel patching.

List Targets

To display all of the running user space processes that the client can patch, use the `ksplice all list-targets` command, for example:

```
sudo ksplice all list-targets
User-space targets:

glibc-ISO8859-1-2.17.78.0.1.1.ksplice25.el7
  └─ gnome-shell (3783)

glibc-libutil-2.17.78.0.1.1.ksplice25.el7
  ├─ firewalld (680)
  ├─ tuned (695)
  ├─ libvirtd (1492)
  ├─ sshd (1497)
  ├─ httpd (1503)
  ├─ httpd (1706)
  ├─ httpd (1707)
  ├─ httpd (1708)
  ├─ httpd (1709)
  ├─ httpd (1710)
  ├─ colord (1942)
  ├─ gdm-session-wor (3418)
  ├─ gnome-session (3460)
```

```
  ├─ gvfsd (3534)
  ├─ gvfsd-fuse (3555)
  ├─ ssh-agent (3617)
  ├─ gnome-settings- (3658)
  ├─ gvfs-udisks2-vo (3727)
  ├─ gvfs-afc-volume (3754)
  ├─ gvfs-mtp-volume (3761)
  ├─ gvfs-gphoto2-vo (3765)
  ├─ gvfs-goa-volume (3769)
  ├─ goa-daemon (3772)
  ├─ gnome-shell (3783)
  ├─ ibus-daemon (3817)
  ├─ ibus-dconf (3821)
  ├─ ibus-x11 (3823)
  ├─ evolution-sourc (3853)
  ├─ nautilus (3882)
  ├─ ibus-engine-sim (3884)
  ├─ tracker-store (3943)
  ├─ abrt-applet (3980)
  ├─ tracker-miner-f (4040)
  ├─ gvfsd-trash (4062)
  ├─ sshd (29328)
  ├─ packagekitd (29465)
  └─ python (29679)
...
Kernel version: Linux/x86_64/3.10.0-229.el7.x86_64/#1 SMP Fri Mar 6 04:05:24 PST 
2015
Xen version: xen/x86_64/#2 SMP Tue Aug 15 13:47:00 PDT 2017/Tue Aug  1 20:27:56 
PDT 2017
```

To display just the Xen hypervisor targets that the client can patch, use the `ksplice xen list-targets` command:

```
sudo ksplice xen list-targets
```

For each Ksplice-aware library, the command reports the running processes that would be affected by an update. The command also reports the effective version of the loaded kernel.

Show Updates

To display the updates that have been applied to the system, use the `ksplice all show` command:

```
sudo ksplice all show
httpd (1706)
httpd (1708)
httpd (1707)
httpd (1709)
httpd (1710)
rsyslogd (689)
chronyd (705)
httpd (1503)
  ├─ [h73qvumn]: CVE-2014-7817: Command execution in wordexp().
  └─ [ml55ngz4]: CVE-2015-1781: Privilege escalation in gethostbyname_r().

Ksplice kernel updates installed:

Installed updates:
[rfywob9d] Clear garbage data on the kernel stack when handling signals.
[6w5ho5e2] Provide an interface to freeze tasks.
```

```
[ftjj21d0] CVE-2015-1421: Privilege escalation in SCTP INIT collisions.
[kw5m66w8] CVE-2015-8159: Privilege escalation in Infiniband userspace access.
[2w6jgsn7] CVE-2015-3331: Privilege escalation in Intel AES RFC4106 decryption.
[p0gek4ir] CVE-2014-9420: Infinite loop in isofs when parsing continuation entries.
[sjqkwypd] CVE-2014-9529: Use-after-free when garbage collecting keys.
[tfn81scy] CVE-2015-1593: Stack layout randomization entropy reduction.
[jga5l35w] CVE-2015-1573: Use-after-free when flushing netfilter rules.
[gdzmj5lc] CVE-2014-9584: Out-of-bounds memory access in ISO filesystem when printing
ER records.
[01560qvg] CVE-2015-2830: mis-handling of int80 fork from 64bits application.
[7ylonu77] CVE-2015-1805: Memory corruption in handling of userspace pipe I/O vector.
[7yehlpm8] Kernel hang on UDP flood with wrong checksums.
[xp1v1o7h] CVE-2014-9715: Remote code execution in the netfilter connection tracking
subsystem.
[89yjgn50] CVE-2015-3636: Memory corruption when unhashing IPv4 ping sockets.
[g327jyvw] CVE-2015-2922: Denial-of-service of IPv6 networks when handling router
advertisements.

Ksplice xen updates installed

  [87x4i9rd]: XSA-230: Information leak when using grant tables.
  [25aiflvq]: XSA-228: Race condition when allocating grant pages.
  [frevokn8]: XSA-227: User controlled memory corruption when mapping a grant
reference.
```

The command reports the updates that have been applied to running processes, as well as the updates to the kernel. In the previous example, Ksplice applied updates for `CVE-2014-7817` and `CVE-2015-1781` to all of the listed processes.

To restrict the scope of the `ksplice` command to user space updates or kernel updates, specify `user` or `kernel` instead of `all` with the command.

To restrict the `ksplice` command to just the Xen hypervisor, specify `xen` instead of `all` with the command.

To display the updates that have been applied to a process specified by its PID, use the `--pid=$PID` option with the `ksplice user show` command:

```
sudo ksplice user show --pid=705
```

Output similar to the following is displayed:

```
chronyd (705)
  ├─ [h73qvumn]: CVE-2014-7817: Command execution in wordexp().
  └─ [ml55ngz4]: CVE-2015-1781: Privilege escalation in gethostbyname_r().
```

Remove Updates

Use the `remove` subcommand to remove all of the updates from a process, for example:

```
sudo ksplice user remove --all --pid=705
```

To remove a specific update that Ksplice has applied to a process, use the `undo` subcommand:

```
sudo ksplice user undo --pid=705 h73qvumn
```

Note:

If necessary, you can prevent Ksplice from patching specified executables and libraries. See Preventing the Ksplice Enhanced Client From Patching User Space Processes and Libraries.

Ksplice patches are stored in the `/var/cache/uptrack` directory. Following a reboot, Ksplice automatically reapplies these patches early in the boot process before the network is configured so that the system is hardened before any remote connections can be established.

List Available Updates

To list all of the available Ksplice updates, use the `upgrade` subcommand:

```
sudo ksplice -n kernel upgrade
```

To install all of the available Ksplice updates, use the `upgrade` subcommand as follows:

```
sudo ksplice -y user upgrade
```

To list all of the available Ksplice updates for the Xen hypervisor, use the `upgrade` subcommand:

```
sudo ksplice -n xen upgrade
```

Show Kernel Version

After Ksplice applies updates to a running kernel, the kernel has an effective version that is different than the original boot version displayed by the `uname -a` command.

Use the `ksplice kernel uname -r` command to display the effective version of the kernel:

```
sudo ksplice kernel uname -r
3.8.13-55.1.1.el6uek.x86_64
```

The `ksplice kernel uname` command supports the commonly used `uname` flags, including `-a` and `-r`, and also provides a way for applications to detect that the kernel has been patched. The effective version is based on the version number of the latest patch that Ksplice Uptrack has applied to the kernel.

Examples

The following examples show ways in which you can view information about Ksplice updates and administer Ksplice updates on a system.

View the updates that Ksplice Uptrack has made to the running kernel:

```
sudo ksplice kernel show
```

View the updates that Ksplice Uptrack has made to the Xen hypervisor:

```
sudo ksplice xen show
```

View the updates that are available to be installed:

```
sudo ksplice kernel show --available
```

Remove all updates from the kernel:

```
sudo ksplice kernel remove --all
```

Remove all updates from the Xen hypervisor:

```
sudo ksplice xen remove --all
```

Prevent Ksplice from reapplying the updates at the next system reboot, create the empty file `/etc/uptrack/disable`:

```
touch /etc/uptrack/disable
```

Alternatively, you can specify `nouptrack` as a parameter on the boot command line when you next restart the system.

Manual Page

For more information and examples, see the `ksplice(8)` manual page.

Preventing the Ksplice Enhanced Client From Patching User Space Processes and Libraries

If you do not want Ksplice to patch the user space processes for certain executables or libraries, you can specify the information in a `/etc/ksplice/blacklist.d` configuration file. The following is an example of a `localblacklist.conf` file. The example shows how you would prevent Ksplice from patching any process that corresponds to any executable in the `/opt/app/bin` or `/usr/local/bin` directory, or from patching any shared library with a name matching `liblocal-*`.

The following example shows the format of the rules, which are Python regular expressions:

```
[executables]
^/opt/apt/bin/.*$
^/usr/local/bin/.*$

[targets]
^liblocal-.*$
```

Configuring the Ksplice Enhanced Client for Offline Mode

The offline version of the Ksplice Enhanced Client removes the requirement that a server on your intranet has a direct connection to the Oracle Uptrack server or ULN. Prior to configuring an offline client, you must set up a local ULN mirror that can act as a Ksplice mirror.

For more information about running Ksplice offline, see About Ksplice Offline Mode.

1. Before proceeding, ensure you have configured a local ULN mirror.
2. Import the GPG key.

   ```
   sudo rpm --import /usr/share/rhn/RPM-GPG-KEY
   ```

3. Disable any existing yum repositories that are configured in the `/etc/yum.repos.d` directory.

You can either edit any existing repository files and disable all of the entries by setting `enabled=0`; or, you can use `yum-config-manager`, for example:

```
sudo yum-config-manager --disable \*
```

Alternatively, you can rename any of the files in this directory so that they do not use the `.repo` suffix. This change causes the `yum` command to ignore these entries, as shown in the following example:

```
cd /etc/yum.repos.d
for i in *.repo; do mv $i $i.disabled; done
```

4. In the `/etc/yum.repos.d` directory, create the `local-yum.repo` file, which contains entries such as the following for an Oracle Linux 7 yum client:

```
[local_ol7_x86_64_ksplice]
name=Ksplice for Oracle Linux $releasever - $basearch
baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol7_x86_64_ksplice_userspace]
name=Ksplice aware userspace packages for Oracle Linux $releasever
- $basearch
baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/userspace/
ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol7_latest]
name=Oracle Linux $releasever - $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol7_UEKR5_latest]
name=Unbreakable Enterprise Kernel Release 5 for Oracle Linux $releasever
- $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/UEKR5/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol7_addons]
name=Oracle Linux $releasever - $basearch - addons
baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/addons/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1
```

- Replace *local_uln_mirror* with the IP address or resolvable host name of the local ULN mirror.
- To distinguish the local repositories from the ULN repositories, optionally prefix the labels for each entry with a string such as `local_`. Note that you must also edit the uptrack configuration, as described in step 7.

- The previous example configuration enables the `local_ol7_x86_64_ksplice`, `local_ol7_x86_64_ksplice_userspace`, `local_ol7_latest`, `local_ol7_UEKR5_latest`, and `local_ol7_addons` channels.

5. Test the configuration:

   a. Clear the yum metadata cache.

      ```
      sudo yum clean metadata
      ```

   b. Verify the configuration.

      ```
      sudo yum repolist
      ```

      If the `yum` commands cannot connect to the local ULN mirror, check that the firewall settings on the local ULN mirror server allow incoming TCP connections to the HTTP port (usually, port 80).

6. If `prelink` is installed, revert all of the prelinked binaries and dependent libraries to the original states and then remove the `prelink` package as follows:

   ```
   prelink -au
   sudo yum remove prelink
   ```

   The `prelink` package is installed and enabled by default on Oracle Linux 6, but not on Oracle Linux 7, Oracle Linux 8 or Oracle Linux 9.

7. Install the offline version of the enhanced client package.

   ```
   sudo yum install ksplice-offline
   ```

8. Add a configuration directive to the `/etc/uptrack/uptrack.conf` file to provide the enhanced client with the label of the local, user space channel in your local yum repository configuration.

   > **Note:**
   >
   > You can skip this step if you did not use the `local_` prefix for the channel label, and this label is an exact match of the label that is used on ULN. If you used the `local_` prefix or labeled this channel differently, add the following lines, but instead of *local_ol7_x86_64_ksplice_userspace*, specify the same label that you used for the Ksplice user space channel, for example:
   >
   > ```
   > [User]
   > yum_userspace_ksplice_repo_name =
   > local_ol7_x86_64_ksplice_userspace
   > ```

9. To install offline update packages, install the relevant packages, for example:

   ```
   sudo yum install ksplice-updates-glibc ksplice-updates-openssl
   ```

   If you are installing the offline updates package for the Xen hypervisor, specify the release in the command, for example:

   ```
   sudo yum install ksplice-updates-xen-$RELEASE
   ```

   For the previous command, *$RELEASE* is the update package that corresponds to the version of the hypervisor that is currently running, as shown in this example:

   ```
   sudo yum install ksplice-updates-xen-4.4.4-153.el6
   ```

After you have installed these packages, the offline version of the enhanced client behaves exactly the same way as the online version.

10. Update the system to install the Ksplice-aware versions of the user space libraries:

    ```
    sudo yum update
    ```

    To install just the libraries and not any other packages, limit the update to the Ksplice user space channel, for example, `ol7_x86_64_userspace_ksplice` channel:

    ```
    sudo yum --disablerepo=* --enablerepo=ol7_x86_64_userspace_ksplice update
    ```

    Alternatively, you can use the following command:

    ```
    sudo yum update *glibc *openssl*
    ```

    You might also use this client to perform kernel updates in the same way that you are able to use the standard uptrack client:

    ```
    sudo yum install uptrack-updates-`uname -r`
    ```

11. To enable the automatic installation of updates, change the entry in `/etc/uptrack/uptrack.conf` from `no` to `yes`, as shown in the following example:

    ```
    autoinstall = yes
    ```

12. Reboot the system so that the system uses the new libraries.

    > **Note:**
    >
    > If you installed updates for the Xen hypervisor, no special configuration is required, and you do not need to reboot the system for the updates to be applied.

Using the Known Exploit Detection Feature on the Ksplice Enhanced Client

> **Note:**
>
> Known exploit detection support is available for the Ksplice Enhanced client only and is currently not supported on the 64-bit Arm (aarch64) platform.

Oracle provides the known exploit detection feature for supported systems that have the Ksplice Enhanced client installed. This feature reports attempted exploitation by known attack vectors. When new Common Vulnerabilities and Exposures (CVEs) are discovered and patched with Ksplice, Oracle may add tripwires to the code that fire when an erroneous condition is triggered, thus enabling you to monitor your systems for suspicious activity.

> **Note:**
>
> Because not all security issues have tripwires added, and also because it is possible to trigger tripwires under normal operations, additional analysis of erroneous conditions might be required.

Running Known Exploit Detection on the Ksplice Enhanced Client

You can run the Ksplice known exploit detection on supported Oracle Linux systems that have the Ksplice Enhanced client installed. This feature works for both the online and offline Ksplice Enhanced client.

To run known exploit detection with the default configuration:

1. Install the `ksplice-known-exploit-detection` package:

   ```
   sudo yum install ksplice-known-exploit-detection
   ```

2. Add the following lines to the `/etc/uptrack/uptrack.conf` file:

   ```
   [Known-Exploit-Detection]
   enabled = yes
   ```

3. Enable the feature by running the `kernel upgrade` command:

   ```
   sudo ksplice kernel upgrade
   ```

4. Verify that the feature has been enabled for the current kernel:

   ```
   cat /proc/sys/kernel/known_exploit_detection
   ```

   If the value is `0` or the file is missing, then the kernel has not enabled kernel exploit detection. If the value is `1`, known exploit detection is enabled on the system.

The helper file, `/usr/sbin/log-known-exploit`, is invoked directly by the kernel. To invoke the help manually to check your configuration or perform dry-run tests, use the following command:

```
/usr/sbin/log-known-exploit --help
```

You can specify the following additional options and arguments with this command:

-h, --help
Display the help message and exit.

-c, --config *\/etc/example.conf*
Specify a compatible configuration file. Defaults to `/etc/log-known-exploit.conf`.

-f, --force
Run the command without checking for root permissions.

-n, --dry-run
Simulate the output and expected actions that would be performed by the helper file.

-d, --dummy
Use dummy data to verify that report logging is configured correctly.

Setting Up Email Alerts for Exploit Attempts

The default configuration for the Ksplice known exploit detection feature only logs exploit attempts to `syslog` by using the normal `syslog` facilities. To set up email alerts, edit the `/etc/log-known-exploit.conf` file as follows:

```
[email]
enabled: 1
recipients: admin@example.com
```

You can use the same configuration file to specify which tripwire reports should be logged or ignored:

```
[actions]
CVE-2019-12345: report
CVE-2019-12346: ignore
```

To define the logging behavior for tripwires that are not specified, add a value for `default` to the list. For example, to avoid logging any tripwire reports unless they are specified, do the following:

```
[actions]
default: ignore
```

Temporarily Disabling and Enabling Tripwires

For troubleshooting purposes, you can disable or enable a specific tripwire manually.

To disable a specific tripwire until the next reboot, remove the CVE reference from the `/proc/sys/kernel/known_exploit_detection_tripwires` file as follows:

```
echo -n '-CVE-2019-12345' |sudo tee /proc/sys/kernel/
known_exploit_detection_tripwires
```

To enable a specific tripwire, append the CVE reference to the same configuration file again:

```
echo -n '+CVE-2019-12345' |sudo tee /proc/sys/kernel/
known_exploit_detection_tripwires
```

Removing the Ksplice Enhanced Client Software

To remove the Ksplice Enhanced client software:

```
sudo yum -y remove ksplice
```

To remove the offline version of the Ksplice Enhanced client software:

```
sudo yum -y remove ksplice-offline
```

To remove the Ksplice-aware versions of the `glibc+openssl` packages from the system:

1. Unsubscribe all of the currently subscribed Ksplice-aware user space channels from the yum repository.
2. Manually downgrade the Ksplice-aware packages using the yum shell and enter the following lines separately:

   ```
   yum shell
   > erase ksplice-helper
   > downgrade glibc* openssl*
   > run
   ```

   > **Note:**
   >
   > The following single command performs the same downgrade action without needing manual entry and can be used for automation purposes:
   >
   > ```
   > printf 'erase ksplice-helper\n downgrade glibc* openssl*\n run' |
   > yum -y shell
   > ```

4

Using the Ksplice Uptrack Client

> **Note:**
>
> Some examples use the `yum` command. For Oracle Linux 8 or Oracle Linux 9, use the `dnf` command, as appropriate.

Installing Ksplice Uptrack from ULN

> **Note:**
>
> If using Oracle Cloud Infrastructure, Ksplice is already installed by default (on all Oracle Linux instances launched after August 25, 2017). For more information, see Oracle Ksplice on Oracle Cloud Infrastructure.

1. Verify the system meets requirements:
   - Must have access to the internet.
   - Must be registered with ULN.
   - Must be running a supported Oracle Linux release, with a supported version of either UEK or RHCK installed. You can verify the kernel version by using the `uname -a` command. For more details, see Maintained Kernels.
   - Ensure that the currently running is also the kernel you want to update, as Ksplice Uptrack applies updates to the running kernel *only*.
2. Log in as the `root` user on the system.
3. If you use an Internet proxy, configure the HTTP and HTTPS settings for the proxy in the shell.
   - For the `sh`, `ksh`, or `bash` shells, use commands such as the following:

```
sudo http_proxy=http://proxy_URL:http_port
sudo https_proxy=http://proxy_URL:https_port
sudo export http_proxy https_proxy
```

For the `csh` shell, use commands such as the following:

```
sudo setenv http_proxy=http://proxy_URL:http_port
sudo setenv https_proxy=http://proxy_URL:https_port
```

4. Using a browser, log in at https://linux.oracle.com with your ULN user name and password, then follow these steps:
   a. On the Systems tab, click the link that is named for your system in the list of registered machines.
   b. On the System Details page, click **Manage Subscriptions**.
   c. On the System Summary page, from the list of available channels, select the appropriate Ksplice for Oracle Linux channel your Oracle Linux release system's architecture (i386 or x86_64).
   d. Click the right arrow (**>**) to move your selection to the list of subscribed channels.
   e. Save the subscription and log out of ULN.
5. On your system, use the `yum` command to install the `uptrack` package.

   ```
   sudo yum install -y uptrack
   ```

   The access key for Ksplice Uptrack is retrieved from ULN and added to `/etc/uptrack/uptrack.conf`, for example:

   ```
   [Auth]
   accesskey = 0e1859ad8aea14b0b4306349142ce9160353297daee30240dab4d61f4ea4e59b
   ```
6. To enable automatic installation of updates, change the value of the `autoinstall` entry in the `/etc/uptrack/uptrack.conf` file from `no` to `yes`:

   ```
   autoinstall = yes
   ```

For information about configuring Ksplice Uptrack, see Configuring the Ksplice Uptrack Client.

For information about managing Ksplice updates, see Using the uptrack-upgrade Command to Manage Ksplice Updates.

Configuring the Ksplice Uptrack Client

The configuration file for both the Ksplice Uptrack client and the Ksplice Enhanced client is `/etc/uptrack/uptrack.conf`. You can modify this file to configure a proxy server, install updates automatically at boot time, and check for and apply new updates automatically.

If your system is registered with the Ksplice Uptrack repository, the client communicates with the Uptrack server by connecting to `https://updates.ksplice.com:443`. You can either configure your firewall to allow the connection through port 443, or you can configure the client to use a proxy server. To configure the client to use a proxy server, set the following entry in the `/etc/uptrack/uptrack.conf` file:

```
https_proxy = https://proxy_URL:https_port
```

You receive an email notification when Ksplice updates are available for your system.

To instruct the client to install all updates automatically, as they become available, set the following entry in the `/etc/uptrack/uptrack.conf` file:

```
autoinstall = yes
```

> **Note:**
>
> Enabling the automatic installation of updates does not automatically update the Ksplice client itself. Oracle notifies you by email when you can upgrade the Ksplice software by using the `yum` command.

Setting the `autoinstall` entry value to `yes` also installs updates automatically at boot time. When you boot the system, the `/etc/init.d/uptrack` script reapplies the installed Ksplice updates.

To install all available updates at boot time, uncomment the following entry in the `/etc/uptrack/uptrack.conf` file:

```
upgrade_on_reboot = yes
```

> **Note:**
>
> The `upgrade_on_reboot` setting is not implemented for user space updates.

Using the uptrack-upgrade Command to Manage Ksplice Updates

Use the `uptrack` command to manage the Ksplice Uptrack Client. For the Enhanced Client, see Using the ksplice Command to Manage the Ksplice Enhanced Client.

List all available updates

```
sudo uptrack-upgrade -n
```

Install all available Ksplice updates

```
sudo uptrack-upgrade -y
```

Display the effective version of the kernel

```
sudo uptrack-uname -r
```

You can compare this to the original boot version displayed by the `uname -a` command.

The `uptrack-uname` command supports commonly used `uname` flags, including `-a` and `-r`, and also provides a way for applications to detect that the kernel has been patched. The effective version is based on the version number of the latest patch that Ksplice has applied to the kernel.

View updates made to running kernel

```
uptrack-show
```

View the updates that are available for installation

```
uptrack-show --available
```

Remove all of the updates from the kernel

```
uptrack-remove --all
```

Prevent Ksplice from reapplying the updates at the next system reboot and create the empty file `/etc/uptrack/disable`

```
touch /etc/uptrack/disable
```

Alternatively, you can specify the `nouptrack` argument as a parameter on the boot command line when you next reboot the system.

Updating the Ksplice Uptrack Client to a Specific Effective Kernel Version

You might want to limit the set of updates that `uptrack-upgrade` installs. For example, the security policy at your site might require a senior administrator to approve Ksplice updates before you can install these updates on production systems. In such cases, you can direct `uptrack-upgrade` to upgrade to a specific effective kernel version instead of the latest available version.

> **Note:**
>
> You can only select a specific effective version when using the offline Ksplice client and offline update RPM packages. This ability enables production systems to remain at a tested update level temporarily, while the latest updates are tested in an integration or UAT environment.

1. Install the `uptrack-updates` package for the current kernel.

   ```
   sudo yum -y install uptrack-updates-`uname -r`
   ```

   > **Important:**
   >
   > If you have booted the most recent available kernel and no Ksplice updates are available, this command may fail or may return an error message notifying you that your kernel version is not yet supported by Ksplice Uptrack. This command only succeeds when Ksplice updates are available for the kernel that you are running.

2. Use the `uptrack-uname -r` command to display the current effective kernel version:

   ```
   sudo uptrack-uname -r
   ```

3. To list all of the effective kernel versions that are available, specify the `--list-effective` option to the `uptrack-upgrade` command, for example:

   ```
   sudo uptrack-upgrade --list-effective
   ```

   Output similar to the following is displayed:

   ```
   Available effective kernel versions:

   3.8.13-44.1.1.el6uek.x86_64/#2 SMP Wed Sep 10 06:10:25 PDT 2014
   3.8.13-44.1.3.el6uek.x86_64/#2 SMP Wed Oct 15 19:53:10 PDT 2014
   3.8.13-44.1.4.el6uek.x86_64/#2 SMP Wed Oct 29 23:58:06 PDT 2014
   3.8.13-44.1.5.el6uek.x86_64/#2 SMP Wed Nov 12 14:23:31 PST 2014
   3.8.13-55.el6uek.x86_64/#2 SMP Mon Dec 1 11:32:40 PST 2014
   3.8.13-55.1.1.el6uek.x86_64/#2 SMP Thu Dec 11 00:20:49 PST 2014
   ```

4. Remove the installed updates to revert the effective kernel version to the earliest that is available, which is 44.1.1 in the following example:

   ```
   sudo uptrack-remove --all
   sudo uptrack-uname -r
   ```

   The current effective kernel version is displayed:

   ```
   3.8.13-44.1.1.el6uek.x86_64
   ```

5. You can set the effective kernel version that you want the system to use by using either of the following methods:

   - Specify the `--effective` option to the `uptrack-upgrade` command.

     For example, if you want to update from 44.1.1 to 44.1.5 instead of updating to the latest 55.1.1, use the `--effective` option to specify 44.1.5:

     ```
     sudo uptrack-upgrade --effective="3.8.13-44.1.5.el6uek.x86_64/#2 SMP Wed Nov
     12 14:23:31 PST 2014"
     ```

     The effective kernel version is displayed after the upgrade:

     ```
     ...
     Effective kernel version is 3.8.13-44.1.5.el6uek
     ```

     You can check that the effective kernel version matches:

     ```
     sudo uptrack-uname -r
     ```

     Output similar to the following is displayed:

     ```
     3.8.13-44.1.5.el6uek.x86_64
     ```

     This method is suitable for setting the effective kernel version on individual systems.

   - Use the `effective_version` option in the `/etc/uptrack/uptrack.conf` file to set an effective package version for the `uptrack-upgrade` command. This method works the same as specifying `--effective` on the command line.

     Because `uptrack-upgrade` runs automatically whenever you update the `uptrack-updates` package on a system, the following entry would limit the effective kernel version to 44.1.5:

```
effective_version = 3.8.13-44.1.5.el6uek.x86_64/#2 SMP Wed Nov 12
14:23:31 PST 2014
```

This method is convenient for setting the effective version for a package on multiple production systems, where the content of the /etc/uptrack/uptrack.conf file can be obtained from a centrally maintained primary copy.

Switching Between Online and Offline Ksplice Uptrack Installation Modes

To switch from one Ksplice client software version (or mode) to another Ksplice software version, for example, switch from a Ksplice online installation to a Ksplice offline installation, you must first remove the existing Ksplice client software from the system. You can then install the new version of the Ksplice client software.

> **Caution:**
>
> Failure to remove an existing Ksplice client software version prior to installing a new Ksplice client software version results in transaction check errors during the package installation process.

For example, if you have the Ksplice Uptrack client software installed on the system and you want to install the offline version of the Ksplice Enhanced client software, you would need to first remove the Ksplice Uptrack client software, and then install the Ksplice offline Enhanced client software, for example:

```
sudo yum remove uptrack ksplice-tools
sudo yum install ksplice-offline
```

To switch from an offline installation to an online installation, for example, to switch from the offline Ksplice Uptrack client software to the Ksplice Uptrack (online) client software, use the following commands:

```
sudo yum remove ksplice-offline ksplice-tools
sudo yum install uptrack
```

Configuring Ksplice Uptrack Clients for Offline Mode

The offline Ksplice client eliminates the need for having a server on your intranet that has a direct connection to the Oracle Uptrack server. Also, a Ksplice offline client does not require a network connection to be able to apply the update package to the kernel. For example, you could use the yum command to install the update package directly from a memory stick.

For more information about running Ksplice offline, see About Ksplice Offline Mode.

1. Before proceeding, you must configure a local ULN mirror.
2. Import the GPG key:

   ```
   sudo rpm --import /usr/share/rhn/RPM-GPG-KEY
   ```

3. Set up a local ULN mirror:

- Disable any existing yum repositories configured in the `/etc/yum.repos.d` directory. You can either edit any existing repository files and disable all entries by setting `enabled=0` or you can use `yum-config-manager`:

  ```
  sudo yum-config-manager --disable \*
  ```

  Alternately, you can rename any of the files in this directory so that they do not use the `.repo` suffix. This causes yum to ignore these entries. For example:

  ```
  sudo cd /etc/yum.repos.d
  sudo for i in *.repo; do mv $i $i.disabled; done
  ```

- In the `/etc/yum.repos.d` directory, create the file `local-yum.repo`, which contains entries such as the following for an Oracle Linux 7 yum client:

  ```
  [local_ol7_x86_64_ksplice]
  name=Ksplice for Oracle Linux $releasever - $basearch
  baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/ksplice/$basearch/
  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
  gpgcheck=1
  enabled=1

  [local_ol7_latest]
  name=Oracle Linux $releasever - $basearch - latest
  baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/latest/$basearch/
  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
  gpgcheck=1
  enabled=1

  [local_ol7_UEKR5_latest]
  name=Unbreakable Enterprise Kernel Release 5 for Oracle Linux $releasever
  - $basearch - latest
  baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/UEKR5/latest/$basearch/
  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
  gpgcheck=1
  enabled=1

  [local_ol7_addons]
  name=Oracle Linux $releasever - $basearch - addons
  baseurl=http://local_uln_mirror/yum/OracleLinux/OL7/addons/$basearch/
  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
  gpgcheck=1
  enabled=1
  ```

  - To distinguish the local repositories from the ULN repositories, prefix the labels of their entries with a string such as `local_`.
  - Replace *local_uln_mirror* with the IP address or resolvable host name of the local ULN mirror.
  - The example configuration enables the `local_ol7_x86_64_ksplice`, `local_ol7_latest`, `local_ol7_UEKR5_latest`, and `local_ol7_addons` channels.

> **Note:**
>
> The Ksplice offline client package is unable to install user space updates, so you should not enable any `*_userspace_ksplice` channels unless you intend to use the offline version of the Ksplice Enhanced client.

4. Install the Ksplice offline client package:

   ```
   sudo yum -y install uptrack-offline
   ```

5. Test the configuration:

   a. Clear the yum metadata cache.

      ```
      sudo yum clean metadata
      ```

   b. Verify the configuration.

      ```
      sudo yum repolist
      ```

   > **Note:**
   >
   > If the `yum` command cannot connect to the local ULN mirror, check that the firewall settings on the local ULN mirror server allow incoming TCP connections to the HTTP port (usually, port 80).

6. Install the Ksplice updates that are available for the kernel.

   ```
   sudo yum -y install uptrack-updates-`uname -r`
   ```

   As new Ksplice updates are made available, use the same command to pick up and apply these updates. You should set up an `anacron` script to perform this task. For example, the following script named `uptrack-updates` in `/etc/cron.daily` would run one time daily:

   ```
   #!/bin/sh
   yum -y install uptrack-updates-`uname -r`
   exit 0
   ```

   > **Important:**
   >
   > The script must be executable and be owned by `root`. Also, you *must* include the `-y` option with the `yum` command when using a script; otherwise, the command hangs and waits for user input.

To display information about Ksplice updates, use the `rpm -qa | grep uptrack-updates` and `uptrack-show` commands.

Using the SNMP Plugin for Ksplice Uptrack

The SNMP plugin for Ksplice enables you to use Oracle Enterprise Manager to monitor the status of Ksplice on your systems. It also works with any monitoring solution that is compatible with SNMP.

Installing and Configuring the SNMP Plugin

Install the SNMP plugin on the system that you want to monitor.

1. Verify the system meets all prerequisites:
   - The `net-snmp` package must be installed.
   - The `net-snmp-utils` package must be installed if you want to be able to test the configuration using the `snmpwalk` command.
   - The `snmpd` service must be configured to start automatically.
   - SELinux must either be disabled or set to permissive mode on the system.
2. Subscribe the system to the appropriate Ksplice channel for the installed Oracle Linux distribution and system architecture, for example, `ol6_x86_64_ksplice` for Oracle Linux 6 on x86_64.
3. As the `root` use, install the `ksplice-snmp-plugin` package on the system:

   ```
   sudo yum -y install ksplice-snmp-plugin
   ```

4. (Optional) If you plan to test the configuration by using the `snmpwalk` command, install the `net-snmp-utils` package as follows:

   ```
   sudo yum -y install net-snmp-utils
   ```

5. Configure the system to use the SNMP plugin by editing the `/etc/snmp/snmpd.conf` file.

   The following example shows how the entries in this file might look on an Oracle Linux 6 system:

   ```
   # Setting up permissions
   # ======================
   com2sec local localhost public
   com2sec mynet source public

   group local v1 local
   group local v2c local
   group local usm local
   group mynet v1  mynet
   group mynet v2c mynet
   group mynet usm mynet

   view all included .1 80

   access mynet "" any noauth exact all none none
   access local "" any noauth exact all all none

   syslocation Oracle Linux 6
   syscontact sysadmin <root@localhost>

   # Load the plugin
   ```

```
# ===============
dlmod kspliceUptrack /usr/lib/ksplice-snmp/kspliceUptrack.so
```

a. In the `com2sec mynet` community entry, replace *source* with the IP address or resolvable host name of the server that hosts the SNMP monitoring software, or with a subnet address represented as *IP_address* / *netmask*, for example, `com2sec mynet 192.168.10.0/24 private`.

   For IPv6 configuration, specify an IPv6 address and netmask to a `com2sec6 mynet` community entry, for example, `com2sec6 mynet fec0::/64 private`.

b. In the `syslocation` entry, replace the argument for the identifier of the system being monitored.

c. In the `dlmod` entry that loads the `kspliceUptrack.so` plugin, replace the *lib* path element with `lib` on a 32-bit system and `lib64` on a 64-bit system.

This sample configuration file is suitable for the purposes of testing.

6. Restart the SNMP service:

   ```
   sudo systemctl restart snmpd
   ```

   For an Oracle Linux 6 client, use the following command:

   ```
   sudo service snmpd restart
   ```

For information about configuring SNMP, refer to the documentation at https://www.net-snmp.org/docs/readmefiles.html. See also the `snmpd(8)` and `snmpd.conf(5)` manual pages.

Testing the SNMP Plugin

You can use the `snmpwalk` command to check information and test the SNMP plugin.

1. Display the installed version of Ksplice.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceVersion
   ```

   Sample output:

   ```
   KSPLICE-UPTRACK-MIB::kspliceVersion.0 = STRING: 1.2.12
   ```

2. Check if available updates for a kernel have been installed.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceStatus
   ```

   Sample output (which shows the kernel is out of date):

   ```
   KSPLICE-UPTRACK-MIB::kspliceStatus.0 = STRING: outofdate
   ```

3. Compare the installed kernel with the Ksplice effective version.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceBaseKernel
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-
   MIB::kspliceEffectiveKernel
   ```

   Sample output (which shows the base kernel and effective kernel are the same, implying no updates have been applied):

   ```
   KSPLICE-UPTRACK-MIB::kspliceBaseKernel.0 = STRING: 2.6.18-274.3.1.el5
   KSPLICE-UPTRACK-MIB::kspliceEffectiveKernel.0 = STRING: 2.6.18-274.3.1.el5
   ```

4. Display a list of all of the updates that have been applied to the kernel.

```
snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceP‌atchTable
```

In this example, we receive no output, meaning no updates have been applied. This confirms why the base and effective kernel versions are the identical and why the kernel is out of date.

5. Display a list of updates that can be installed.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceAvailTable
   ```

   Sample output:

   ```
   KSPLICE-UPTRACK-MIB::ksplicea vailIndex.0 = INTEGER: 0
   KSPLICE-UPTRACK-MIB::kspliceavailIndex.1 = INTEGER: 1
   KSPLICE-UPTRACK-MIB::kspliceavailIndex.2 = INTEGER: 2
   ...
   KSPLICE-UPTRACK-MIB::kspliceavailDesc.23 = STRING: CVE-2011-4325: Denial of
   service in NFS direct-io.
   KSPLICE-UPTRACK-MIB::kspliceavailDesc.24 = STRING: CVE-2011-4348: Socking locking
   race in SCTP.
   KSPLICE-UPTRACK-MIB::kspliceavailDesc.25 = STRING: CVE-2011-1020, CVE-2011-3637:
   Information leak, DoS in /proc.
   ```

6. After fully upgrading your kernel by using Ksplice Uptrack, you can run the following `snmpwalk` commands to verify that the kernel is up to date.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceStatus
   ```

   Sample output:

   ```
   KSPLICE-UPTRACK-MIB::kspliceStatus.0 = STRING: uptodate
   ```

7. Check that there are no updates available for installation, and also that the patches that have been applied.

   ```
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceAvailTable
   snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::ksplicePatchTable
   ```

   Output similar to the following is displayed:

   ```
   KSPLICE-UPTRACK-MIB::ksplicepatchIndex.0 = INTEGER: 0
   KSPLICE-UPTRACK-MIB::ksplicepatchIndex.1 = INTEGER: 1
   KSPLICE-UPTRACK-MIB::ksplicepatchIndex.2 = INTEGER: 2
   ...
   ```

Removing the Ksplice Uptrack Client Software

To remove the online Ksplice Uptrack software from a system:

```
sudo yum -y remove uptrack
```

To remove the offline Ksplice Uptrack software from a system:

```
sudo yum -y remove uptrack-offline
```

5 Using the Ksplice Uptrack API

The Ksplice Uptrack API is a RESTful web API that enables you to query the status of machines that are running Oracle Ksplice Uptrack. The API provides information about the updates that your machines have, as well as status of any out-of-date, inactive, or unsupported machines.

You can use the command-line tools that are included with the Python bindings; or, you can write your own custom scripts by using the bindings. You can also write your own interface by using HTTP requests. The Python bindings include the `check_uptrack` and `check_uptrack_local` plugins for Nagios. These plugins enable you to monitor the status of your machines.

> **Note:**
>
> The Ksplice Uptrack API does not currently support user space or Xen updates. However, the online version of the Ksplice Enhanced client can patch shared libraries for user space processes that are running on an Oracle Linux 6, Oracle Linux 7, Oracle Linux 8 or Oracle Linux 9 system.

> **Note:**
>
> You cannot use the Ksplice Uptrack API to monitor machines that are running Ksplice Offline client because these systems are not registered with https://status.ksplice.oracle.com.

For more information about the Ksplice Uptrack API, visit http://www.ksplice.com/.

- Installing the API Command-Line Tools
- Using the Ksplice Uptrack API Commands
- About the API Implementation
- Using the Nagios Plugin

Installing the API Command-Line Tools

The API command-line tools are included with the Python bindings for the API in the `python-ksplice-uptrack` package. This package is available in the Ksplice for Oracle repositories on ULN at `linux.oracle.com` or the Ksplice Uptrack for Oracle Linux repositories at `www.ksplice.com`.

1. Ensure that you have a valid Oracle Linux Premier subscription, a Premier Limited subscription, or an Oracle Premier Support for Systems and Operating Systems subscription.

These subscriptions automatically register your system to use Ksplice. See Registering With ULN for more details.

2. Install the `python-ksplice-uptrack` package.

```
sudo yum install -y python-ksplice-uptrack
```

   The Python bindings are installed in the Python site-packages directory, which is typically `/usr/lib/python2.6/site-packages/ksplice`. The API tools are installed in the `/usr/bin` directory.

   The Nagios plugins are installed in `/usr/lib/nagios/plugins`.

Using the Ksplice Uptrack API Commands

The Python bindings include the following commands, which cover common uses of the Ksplice Uptrack API.

- uptrack-api-authorize
- uptrack-api-describe
- uptrack-api-list

The commands use the API user name and key for authentication.

- Viewing Your API User Name and API Key
- Generating a New API Key
- Specifying the username and api_key Variables
- Specifying a Proxy

uptrack-api-authorize

The `uptrack-api-authorize` command uses the `authorize` API call to change the authorization for a single machine, as shown in the following example:

```
                    uptrack-api-authorize -u api_username -k api_key
                       uuid deny
Successfully denied access for uuid.
uptrack-api-authorize -u api_username -k api_key
                       uuid allow
Successfully allowed access for uuid .
```

> **Note:**
>
> To view your API user name and API key, log in to https://status.ksplice.oracle.com and then select the **Settings** tab.
>
> The UUID of a registered machine is stored in `/var/lib/uptrack/uuid` on the system. An example of a UUID is `e82ba0ae-ad0a-4b92-a776-62b502bfd29d`.

uptrack-api-describe

The `uptrack-api-describe` command uses the `describe` API call to get detailed information about a single machine, which is specified by its UUID, for example:

```
                    uptrack-api-describe -u api_username -k api_key
                       uuid

    prod1.example.com (192.168.1.100)
    Effective kernel: 2.6.18-194.11.1.el5
    This machine is no longer active
    Last seen on 2010-09-12T10:19:35Z
    OS status: Up to date
```

Alternatively, you can specify the `--this-machine` option if you are running the script on the machine you want to check:

```
                       uptrack-api-describe -u api_username -k api_key --this-machine
    qa.example.com (192.168.1.200)
    Effective kernel: 2.6.18-194.8.1.el5
    This machine is active
    Last seen on 2010-09-15T12:43:07Z
    OS status: Out of date:
      * Install v8gacfip CVE-2010-2521: Remote buffer overflow in NFSv4 server.
      * Install 3c4sopia CVE-2010-2226: Read access to write-only files in XFS
filesystem.
      * Install oiqwvltu CVE-2010-2240: Privilege escalation vulnerability in memory
management.
```

uptrack-api-list

The `uptrack-api-list` command uses the `machines` API call to return a list of all of your machines and their statuses, for example:

```
                       uptrack-api-list -u api_username -k api_key

    - dev1.example.com (192.168.1.102): outofdate
    - qa1.example.com (192.168.1.103): outofdate (inactive)
    - prod1.example.com (192.168.1.100): uptodate
    - prod2.example.com (192.168.1.101): uptodate
```

Viewing Your API User Name and API Key

1. Log in to https://status.ksplice.oracle.com.
2. Select the **Settings** tab.

Generating a New API Key

> **Note:**
>
> Generating a new key invalidates your existing key.

1. Log in to https://status.ksplice.oracle.com.
2. Select the **Settings** tab.
3. On the Settings page, select the **Generate a new API key?** check box and click **Save Changes**.

Specifying the username and api_key Variables

If you set the username and api_key variables in the /etc/uptrack-api.conf file, you do not need to supply these variables as command-line arguments to the scripts.

Place the variables under an [uptrack] section heading, for example:

```
[uptrack]
username = jo.admin@example.com
api_key  = 3af3c2c1ec407feb0fdc9fc1d8c4460c
```

You can also set the username and api_key variables in the UPTRACK_API_USERNAME and UPTRACK_API_KEY environment variables, for example:

```
export UPTRACK_API_USERNAME=jo.admin@example.com
export UPTRACK_API_KEY=3af3c2c1ec407feb0fdc9fc1d8c4460c
uptrack-api-describe --this-machine
```

Specifying a Proxy

If you access the internet by using a proxy, specify the connection information in the [uptrack] section of the /etc/uptrack-api.conf file, as shown in the following example:

```
https_proxy = [protocol://][username:password@]proxy[:port]
```

In the previous example, *protocol* is either specified as http or https, *username* and *password* authenticate you with the proxy (if required), and *proxy* and *port* are the host name/IP address and port number that you use to connect to the proxy server, respectively.

The following example shows how you might specify this connection information:

```
https_proxy = http://proxy.example.com:3128/
```

Note that the proxy *must* support HTTPS connections.

About the API Implementation

Learn about the API version, authentication, request format, supported requests, and a sample interaction.

- API Version
- API Authentication
- API Request Format
- Supported API Requests
- Interaction Sample

API Version

This document describes version 1 of the API. All requests go to paths that begin with `/api/1/`.

API Authentication

Authentication to the Uptrack API server uses a user name and an API key that are specified in custom HTTP headers. Specifically, all requests must include `X-Uptrack-User` and `X-Uptrack-Key` HTTP headers that include the API user name and API key of the user who is making the request.

API Request Format

API requests or responses include JSON-encoded data in the request body. Requests should set a `Content-Type` header of `application/json`. Similarly, any requests that expect a response containing content should include an `Accept:` header that contains the value `application/json`.

These headers are not required currently, as the API supports only JSON-encoded data, but future versions of the API might support additional data-encoding formats.

Supported API Requests

The following are descriptions of the API requests that are currently supported.

GET /api/1/machines

The `GET /api/1/machines` API request returns a list of all of the registered machines. This list includes inactive machines that have uninstalled Uptrack or any machines that have not reported to the Uptrack server recently. The list does not include machines that you have hidden by using the web interface. The response shows a list of machines, which are represented as dictionaries, as shown in the following example:

```
{
        hostname: uptrack.example.com,
        ip: 184.73.248.238,
        last_seen: '2010-04-26T18:03:43Z',
        uuid: e82ba0ae-ad0a-4b92-a776-62b502bfd29d,
        active: true,
        status: uptodate,
        authorization: allowed,
        autoinstall: true,
        mmap_min_addr: 4096,
        uptrack_client_version: 1.2.1
    }
```

The following fields are provided in the response:

`status`
Contains one of the following values:

- `outofdate` - Additional updates are available for installation on the machine.
- `unsupported` - The machine's kernel is not supported by Ksplice Uptrack.
- `uptodate` - All available updates have been installed on the machine.

`authorization`
Contains one of the following values:

- `allowed` - The machine is allowed to communicate with the Uptrack servers and to receive updates.
- `denied` - The machine has been denied access to the Uptrack servers via the web interface, `uptrack-api-authorize`, or the `authorize` API call.
- `pending` - This account has the default deny policy set for new machines, and the machine has not yet been authorized.

`autoinstall`
Indicates whether `autoinstall` is set on the machine.

`mmap_min_addr`
Is the value of `/proc/sys/vm/mmap_min_addr` or `None` for clients prior to version 1.0.3.

`uptrack_client_version`
Is the version of the Uptrack client that the machine is running.

GET /api/1/machine/$UUID/describe

The `GET /api/1/machine/$UUID/describe` API request returns information about the machine with the specified UUID. The UUID of a machine is stored in `/var/lib/uptrack/uuid` and can be retrieved by using the `machines` query. The response is a dictionary of the same form that `GET /api/1/machines` returns, except that it includes the following additional fields:

`effective_kernel`
Ksplice has applied all of the important security and reliability updates that are needed to bring the machine into line with this kernel version.

`group`
The group to which the machine is assigned. You can also use the web interface to manage machine groups.

installed_updates
A list of 2-element dictionaries of the form `{'ID': update_id, 'Name': update_name}` that represent the updates currently installed on the machine. *update_id* is the ID code of an update (for example, `diptbg4f`) and *update_name* is a short descriptive name for the update (for example, `CVE-2010-0415: Information Leak in sys_move_pages`).

original_kernel
The kernel version of the machine before any Ksplice updates were applied.

steps
A list of two-element lists of the form `[action, {'ID': update_id, 'Name': update_name}]`,which represent the updates that need to be installed or removed to bring the machine up to date. For the *action* argument, you can specify `Install` or `Remove`. Note that an existing update is removed if it superseded by a more recent version.

POST /api/1/machine/$UUID/authorize

The `POST /api/1/machine/$UUID/authorize` API request authorizes the machine with the specified UUID to access the Uptrack service if you have configured your account to deny access to new machines.

The content is a dictionary of the following form:

```
{authorized: boolean}
```

Specify the *boolean* argument as `true` to authorize the machine or `false` to revoke authorization.

POST /api/1/machine/$UUID/group

The `POST /api/1/machine/$UUID/group` API request changes the group of the machine with the specified UUID.

The content is a dictionary that uses the following form:

```
{group_name: string}
```

In the previous example, *string* is the name of the new group. The group is created if it does not already exist. Note that if the account does not have a machine with the specified UUID, the request results in an `HTTP 404` error.

To remove a machine from a group, you can set the group to a different name, or you can specify an empty string for no group.

Interaction Sample

The following example, which is provided as a reference *only*, shows an interaction that might take place when using the Uptrack API.

This conversation takes place with the server `uptrack.api.ksplice.com` over port 443 by using the Secure Sockets Layer (SSL) protocol.

The following is a request for a list of registered machines that is made to the server:

```
GET /api/1/machines HTTP/1.1
Host: uptrack.api.ksplice.com
Accept: application/json
```

```
X-Uptrack-User: jo.admin@example.com
X-Uptrack-Key: 3af3c2c1ec407feb0fdc9fc1d8c4460c
```

The server authenticates the request and responds with a list of the machines, for example:

```
HTTP/1.0 200 OK
Date: Mon, 03 May 2010 21:09:48 GMT
Content-Type: application/json

[{"status": "uptodate", "uuid": "e82ba0ae-ad0a-4b92-a776-62b502bfd29d",
  "active": true, "ip": "192.168.248.238", "hostname": "utclient.example.com",
  "authorization": "allowed", "autoinstall": true,
  "last_seen": "2010-04-26T18:03:43Z", "mmap_min_addr": 4096,
  "uptrack_client_version": "1.2.1"}]
```

Configuring the check_uptrack Nagios Plugin

> **Note:**
>
> The Nagios software does not include the `python-ksplice-uptrack` package. For information about obtaining and using Nagios, visit the official Nagios website at http://www.nagios.org.

Configure the `check_uptrack` Nagios plugin as follows:

1. Set the `username` and `api_key` variables in the configuration file `/etc/uptrack-api.conf` under an `[uptrack]` section heading, for example:

   ```
   [uptrack]
   username = jo.admin@example.com
   api_key  = 3af3c2c1ec407feb0fdc9fc1d8c4460c
   ```

2. If you access the Internet by using a proxy, specify the connection information in the `[uptrack]` section of `/etc/uptrack-api.conf`:

   ```
   https_proxy = [protocol://][username:password@]proxy[:port]
   ```

   In the previous example, *protocol* is `http` or `https`, *username* and *password* authenticate you with the proxy (if required), and *proxy* and *port* are host name/IP address and port that you use to connect o the proxy server, respectively. The connection information you specify might be similar to the following:

   ```
   https_proxy = http://proxy.example.com:3128/
   ```

   The proxy *must* support HTTPS connections.

3. Configure the `check_uptrack` plugin in the Nagios configuration file, which is usually `/usr/local/nagios/etc/nagios.cfg`.

   The following minimal configuration enables you to run the plugin:

   ```
   # Dummy host with which to associate the Uptrack service
   define host {
          host_name                        uptrack-service
          notifications_enabled            0
          max_check_attempts               1
   ```

```
           notification_interval           0
           check_period                    never
           contacts                        server-admins
    }

    define service {
           host_name                       uptrack-service
           service_description             Ksplice Uptrack Update Status
           check_command                   check_uptrack
           notifications_enabled           1
           normal_check_interval           60
           retry_check_interval            15
           max_check_attempts              4
           notification_options            w,c,r
           contacts                        server-admins
    }

    define command {
           command_name     check_uptrack
           command_line     /usr/lib/nagios/plugins/check_uptrack
    }

    define command {
           command_name     check_uptrack_opts
           command_line     /usr/lib/nagios/plugins/check_uptrack -w $ARG1$ -c $ARG2$
    }
```

Using the Nagios Plugin

To monitor all of your machines by using the Nagios plugin, run the following command:

```
sudo /usr/lib/nagios/plugins/check_uptrack
```

The previous command produces a summary of your machines in the standard Nagios plug-in format, as shown in the following example:

```
2 machines are OUTOFDATE!|uptodate=1280;outofdate=1;unsupported=0;inactive=3
  prod1.example.com (192.168.1.1) is OUTOFDATE
  prod2.example.com (192.168.1.2) is OUTOFDATE
```

If you specify the `-c` or `-w` options with a comma-separated list of the arguments that also specify the `i`, `o`, or `u` options for inactive, out-of-date, or unsupported machines, the `check_uptrack` command displays critical or warning notices for machines that match the criteria.

For example, the following command returns warning notices for any machines that are inactive or unsupported, as well as critical notices for any machines that are out of date:

```
sudo /usr/lib/nagios/plugins/check_uptrack -w u,i -c o
```

To monitor the local machine, use the `check_uptrack_local` plugin:

```
sudo /usr/lib/nagios/plugins/check_uptrack_local
```

The output from the `check_uptrack_local` command is similar to the output from the `check_uptrack` command. However, for out-of-date machines, the command also lists the updates that are required to bring the machine up to date.

Note:

The `check_uptrack_local` command reads the local uptrack update cache; however, it does not use the settings from the `/etc/uptrack-api.conf` file.