Oracle® Linux

Ksplice User's Guide

E39380-33
April 2019

Oracle Legal Notices

Abstract

This guide provides information about using Ksplice to update a running system without the need to reboot the system.

Document generated on: 2019-04-12 (revision: 7373)

Table of Contents

Preface

The *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.

Audience

This document is intended for administrators who need to configure Oracle Ksplice on Oracle Linux systems. It is assumed that readers are familiar with and have a general understanding of Linux system administration.

Document Organization

The document is organized as follows:

- Chapter 1, *About Oracle Ksplice* provides an overview of Oracle Ksplice.
- Chapter 2, *Working With the Ksplice Enhanced Client* provides information about installing and configuring the Ksplice Enhanced client and applying updates to a running system.
- Chapter 3, *Working With Ksplice Uptrack* provides information about installing and configuring the Ksplice Uptrack client and applying updates to a running system.
- Chapter 4, *Working With the Ksplice Uptrack API* describes how to use the Ksplice Uptrack API.

Related Documents

The documentation for this product is available at:

https://www.oracle.com/technetwork/server-storage/linux/documentation/index.html.

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
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Chapter 1 About Oracle Ksplice

Table of Contents

This chapter provides a high-level overview of Oracle Ksplice.

1.1 Overview of Oracle Ksplice

Caution

The majority of the installation and configuration instructions in this guide apply *only* to Oracle Linux releases . If you plan to use Ksplice to patch the Xen hypervisor on Oracle VM Server 3.4.5 and later releases, refer to the documentation for the Oracle VM release that you are running for step-by-step instructions. 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*.

Linux systems receive regular security updates to core operating system components that necessitate patching and rebooting. Traditionally, applying such updates would require you to obtain and install the updated RPMs, schedule downtime, and reboot the server to the new package version, with any critical updates. However, as system setups become more complex, with many interdependencies, access to services and applications must remain as undisrupted as possible, as scheduling such reboots becomes more difficult and costly.

Oracle Ksplice provides a way for you to keep your systems secure and highly available by enabling you to update them with the latest kernel and key user-space security and bug fix updates, and Xen hypervisor updates on Oracle VM Server 3.4.5 and later.

Note

When using Ksplice to patch the Xen hypervisor on Oracle VM Server 3.4.5 and later, the minimum version that is required is `xen-4.4.4-196.el6.x86_64.rpm`.

Oracle Ksplice updates the running operating system without requiring a reboot. Your systems remains up to date with OS vulnerability patches and downtime is minimized. A Ksplice update takes effect immediately upon application. Note that a Ksplice update is not the same as an on-disk change that requires a subsequent reboot to take effect. However, note that on-disk updates are still required when

using Ksplice to ensure that package binaries are updated to the most recent version and can be used in the event that the system or processes are restarted. On-disk updates are handled by subscribing to the Unbreakable Linux Network (ULN) or by using a local ULN mirror.

Oracle creates each Ksplice update from a package update that originates either from Oracle or the open source community.

1.1.1 Supported Kernels

You can use Ksplice to bring the following Oracle Linux kernels up to date with the latest important security and bug fix patches:

- All Oracle Unbreakable Enterprise Kernel versions for Oracle Linux 5, Oracle Linux 6, or Oracle Linux 7, starting with 2.6.32-100.28.9 (released March 16, 2011).
- All Oracle Linux 6 and Oracle Linux 7 kernels, starting with the official release.
- All Oracle Linux 5 Red Hat Compatible Kernels, starting with Oracle Linux 5.4 (2.6.18-164.el5, released September 9, 2009).
- All Oracle Linux 5 Red Hat Compatible Kernels with bug fixes added by Oracle, starting with Oracle Linux 5.6 (2.6.18-238.0.0.0.1.el5, released January 22, 2011).

To confirm whether a particular kernel is supported, install the Ksplice Uptrack client or Ksplice Enhanced Client on a system that is running the kernel.

Note

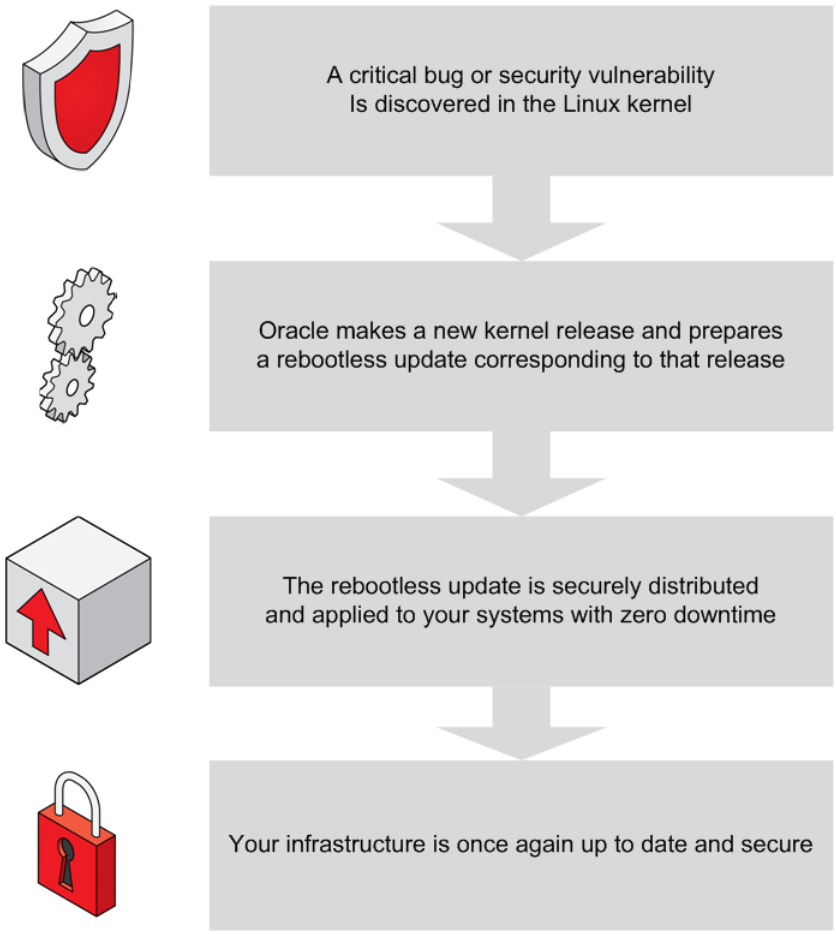
If your system is currently running Red Hat Enterprise Linux and you have recently migrated to Oracle Linux Premier Support, you can use Ksplice to update the existing Red Hat Enterprise Linux kernel. You do not need to switch to the Red Hat Compatible Kernel to use Ksplice kernel patches. These patches are available on ULN as `uptrack-updates-`*`kernel_version`* packages in the Ksplice for Oracle Linux channels.

For questions about supported kernels, send e-mail to ksplice-support_ww@oracle.com.

1.1.2 About Ksplice Updates

The following figure illustrates the life cycle of a Ksplice update for the Linux kernel.

Figure 1.1 Life Cycle of a Ksplice Update



Per the previous diagram, 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 using the Oracle Ksplice Uptrack server and the Unbreakable Linux Network (ULN) and is applied to your systems by the Ksplice Uptrack client or Ksplice Enhanced client with zero downtime. Your infrastructure is again up to date and secure.

Note

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

1.1.3 Patching and Updating Your System

Ksplice patches enable you to keep a system up to date while it is running. You should also use the `yum` command to install the regular kernel RPM packages for released errata that are available from the Unbreakable Linux Network (ULN) or the Oracle Linux yum server. Your system will then be ready for the next maintenance window or reboot. When you restart the system, you can boot it from a newer kernel version. Ksplice Uptrack uses the new kernel as a baseline for applying patches as they become available.

1.1.4 Using Ksplice With Oracle Enterprise Manager

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 UI.

For more information about using Oracle Enterprise Manager to monitor and manage Ksplice patching for Oracle Linux hosts, see the *Oracle Enterprise Manager Lifecycle Management Administrator's Guide*, which is available at:

https://docs.oracle.com/cd/cloud-control-13.3/EMLCM/GUID-DA483950-9009-4293-BEF2-2F3C9DAACF33.htm#EMLCM-GUID-DA483950-9009-4293-BEF2-2F3C9DAACF33

1.2 About the Ksplice Client Software

This section describes the different Ksplice client software types that are available in Oracle Linux. A description of each Ksplice client type, as well as information about when you might use each client, is provided. For a high-level overview of the support that each Ksplice client provides, see Section 1.3.1, “Choosing a Ksplice Client”.

1.2.1 About the Ksplice Enhanced Client

The Ksplice Enhanced client is available for Oracle Linux 6 and Oracle Linux 7, but not for Oracle Linux 5. The enhanced version of the Ksplice online client supports kernel and user-space updates and can also be used to patch the Xen hypervisor on Oracle VM Server Release 3.4.5 and later.

Note

To use Ksplice to patch the Xen hypervisor on Oracle VM 3.4.5 and later, the minimum Xen hypervisor version is `xen-4.4.4-196.el6.x86_64.rpm`.

For information about when to use the Ksplice Enhanced client, see Section 1.3.1, “Choosing a Ksplice Client”.

The Ksplice Enhanced client can patch in-memory pages of Ksplice aware shared libraries such as `glibc` and `openssl` for user-space processes, in addition to the kernel updates applied by the traditional Ksplice Uptrack client. User-space patching enables you to install bug fixes and protect your system against security vulnerabilities without having to restart processes and services. Both an online and an offline version of the enhanced client are available.

You manage the Ksplice Enhanced client by using the `ksplice` command rather than `uptrack` commands. Note that the enhanced client shares the same configuration file as the Uptrack client, which is located at `/etc/uptrack/uptrack.conf`. For more information about this file, see Section 3.3, “Configuring a Ksplice Uptrack Client”.

Note the following important information about Ksplice limitations:

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

```
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 `/var/cache/ksplice` 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.
- Due to 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 re-executed on system updates, so it does not require patching with Ksplice.

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

The offline version of the Ksplice Enhanced client removes the requirement that a server on your intranet have a direct connection to the Oracle Uptrack server or to ULN. All available Ksplice updates for each supported kernel version or user-space package are bundled into an RPM that is specific to that version. This package is updated every time a new Ksplice patch becomes available for the kernel. In this way, you can create a local ULN mirror that acts as a mirror for the Ksplice aware channels for Oracle Linux on ULN. See Section 2.4, “Configuring the Ksplice Enhanced Client for Offline Mode”.

At regular intervals, you can download the latest Ksplice update packages to this server. After installing the offline Ksplice Enhanced client on your local systems, they can then connect to the local ULN mirror to receive updates. See Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror” for more information about configuring a local ULN mirror.

When you have set up a local ULN mirror to act as a Ksplice mirror, you can then configure your other systems to receive `yum` updates, as well as Ksplice updates. For task-related information, see Chapter 2, *Working With the Ksplice Enhanced Client*.

1.2.2 About the Ksplice Uptrack Client

The Ksplice Uptrack client 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. You can use Ksplice Uptrack as well as continue to upgrade your kernel through the usual mechanism, such as running the `yum` command.

For information about when to use the Ksplice Uptrack client, see Section 1.3.1, “Choosing a Ksplice Client”.

Ksplice Uptrack is freely available for Oracle customers who subscribe to Oracle Linux Premier Support, and to 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.

The Ksplice Offline client removes the requirement that a server on your intranet have a direct connection to the Oracle Uptrack server. All available Ksplice updates for each supported kernel version are bundled into an RPM that is specific to that version. This package is updated every time a new Ksplice patch becomes available for the kernel.

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. However, a more typical method would be to create a local ULN mirror that acts as a mirror of the Ksplice for Oracle Linux channels on ULN. At regular intervals, you download the latest Ksplice update packages to this server. After installing the Ksplice Offline client on your local systems, the systems can connect to the local ULN mirror to receive updates without requiring access to the Oracle Uptrack server. See Section 3.7, “Working With the Ksplice Uptrack Client in Offline Mode”.

For information about when you might choose to use the Ksplice Offline client, see Section 1.3.1, “Choosing a Ksplice Client”.

Note

You cannot use the web interface or the Ksplice Uptrack API to monitor systems that are running Ksplice Offline client, as such systems are not registered with https://uptrack.ksplice.com.

1.3 Preparing to Use Oracle Ksplice

The following are tasks that you might need to perform prior to installing and configuring Ksplice, depending on the Ksplice client that you plan to use:

- Determine which Ksplice client will best suit your needs, as the additional tasks described in this section are dictated by the Ksplice client that you choose to install. See Section 1.3.1, “Choosing a Ksplice Client” for more details.
- Register your system with the Unbreakable Linux Network (ULN). See *Unbreakable Linux Network User's Guide*.
- Ensure that you have a valid Oracle Linux Premier, Premier Limited, or Oracle Premier Support for Systems and Operating Systems subscription, as any of these subscriptions automatically register you to use the Ksplice Uptrack server at https://uptrack.ksplice.com See Section 1.3.2, “About Oracle Ksplice and ULN Registration” for more details.
- If you plan to use either the Ksplice Enhanced client or the Ksplice Uptrack client as offline clients, you must set up a local ULN mirror first, as described in Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror”.
- If you are using Ksplice with Spacewalk, you must also set up a local ULN mirror, as described in Section 1.3.4, “Configuring a Spacewalk Server to Act as a Ksplice Mirror”.

For further details on setting up the Ksplice Enhanced client in offline mode, see Section 2.4, “Configuring the Ksplice Enhanced Client for Offline Mode”. For further details on setting up the Ksplice Uptrack client in offline mode, see Section 3.7.1, “Configuring Ksplice Uptrack Clients for Offline Mode”

1.3.1 Choosing a Ksplice Client

The following table describes feature support, requirements, and limitations for each Ksplice client. Use this information to decide which Ksplice client will best suit your needs.

Ksplice Client	User Space Support	Xen Hypervisor Patching Support	Legacy Compatibility
Ksplice Enhanced Client	Supported	Supported	Not supported
Ksplice Uptrack Client	Not supported	Not supported	Supported

1.3.2 About Oracle Ksplice and ULN Registration

To use Oracle Ksplice, your system must have access to the Internet, and you must register your system with the Unbreakable Linux Network (ULN) first, unless the system is configured to use the Oracle Ksplice client as an offline client. If your client is configured to function as an offline client, you must configure a local ULN mirror that the client can access to receive updates. For more information, see Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror”.

If you have an Oracle Linux Premier support subscription, a Premier Limited support subscription, or an Oracle Premier Support for Systems and Operating Systems subscription and a Customer Support Identifier (CSI), your account is automatically registered to use the Ksplice Uptrack server. Systems that are registered with ULN can install either the Ksplice Enhanced client software or the Ksplice Uptrack client software from ULN to automatically receive updates from the Ksplice Uptrack server. When the Ksplice client is installed, it is allocated an identification key that associates it with the CSI for your account.

If your account has a valid CSI, you can log in to the Ksplice Uptrack server web interface at https://uptrack.ksplice.com by using your Oracle Single Sign-on (SSO) credentials. After logging into 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.

1.3.3 Configuring a Local ULN Mirror to Act as a Ksplice Mirror

The following procedure describes how to configure a local ULN mirror to act as a Ksplice mirror. Use this procedure if you are planning to install and configure the Ksplice client as an offline client.

For more information about setting up a local ULN mirror, see Creating and Using a Local ULN Mirror in the *Oracle Linux Unbreakable Linux Network User's Guide*.

1. Using a browser, log in to https://linux.oracle.com by providing the ULN user name and password that you used to register your system.
2. On the Systems tab, click the link that is named for your system in the list of registered machines.
3. On the System Details page, click **Edit**.
4. On the Edit System Properties page, select the **Yum Server** check box and then click **Apply Changes**.
5. On the System Details page, click **Manage Subscriptions**.
6. On the System Summary page, select channels from the list of available or subscribed channels and click the arrows to move the channels between the lists.

   Modify the list of subscribed channels to include those Ksplice for Oracle Linux channels you want to make available to local Offline Clients.

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

Channel Name	Channel Label	Description
Ksplice for Oracle Linux 5 (i386)	`ol5_i386_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on i386 systems.
Ksplice for Oracle Linux 5 (x86_64)	`ol5_x86_64_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on x86-64 systems.
Ksplice for Oracle Linux 6 (i386)	`ol6_i386_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on i386 systems.

Channel Name	Channel Label	Description
Ksplice for Oracle Linux 6 (x86_64)	ol6_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on x86-64 systems.
Ksplice for Oracle Linux 7 (x86_64)	ol7_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 7 on x86_64 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.
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 6 (x86_64). This channel should only be used with the Ksplice Enhanced client.

7. When you are finished selecting channels, click **Save Subscriptions** and log out of ULN.

1.3.4 Configuring a Spacewalk Server to Act as a Ksplice Mirror

To configure a Spacewalk server to act as a Ksplice mirror, you configure repositories and associated software channels for the Oracle Linux releases and architectures of the clients on which you want to run the Offline client. Each Ksplice channel should be a child of the appropriate base software channel.

For more information, see "Working with Repositories" and "Working With Software Channels" in Chapter 2 of the *Spacewalk 2.7 for Oracle Linux Client Life Cycle Management Guide* for the Oracle Linux release that you are running.

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

Channel Name	Channel Label	Description
Ksplice for Oracle Linux 5 (i386)	ol5_i386_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on i386 systems.
Ksplice for Oracle Linux 5 (x86_64)	ol5_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on x86-64 systems.
Ksplice for Oracle Linux 6 (i386)	ol6_i386_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on i386 systems.
Ksplice for Oracle Linux 6 (x86_64)	ol6_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on x86-64 systems.
Ksplice for Oracle Linux 7 (x86_64)	ol7_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 7 on x86_64 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.
Latest packages for 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.

Using the information from the previous table, you would specify the URL of the Ksplice for Oracle Linux 6 (x86_64) channel on ULN as follows:

```
uln:///ol6_x86_64_ksplice
```

Chapter 2 Working With the Ksplice Enhanced Client

Table of Contents

This chapter describes how to install and configure the Ksplice Enhanced client to update packages on a running system. For more information about Ksplice Uptrack, go to http://www.ksplice.com/.

For an overview of Ksplice, see Chapter 1, *About Oracle Ksplice*.

2.1 Installing the Ksplice Enhanced Client From ULN

The Ksplice Enhanced client is available as either an online client, which requires the server to have a direct connection to the Oracle Uptrack server. Or, alternatively, you can use the Ksplice Enhanced client as an offline client, which requires access to a local ULN mirror. See Section 1.3, “Preparing to Use Oracle Ksplice”.

Caution

The following procedure applies to Oracle Linux releases *only*. If you plan to use Ksplice to patch the Xen hypervisor on Oracle VM 3.4.5 and later releases, refer to the documentation for the Oracle VM release that you are running for step-by-step instructions. 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*.

The system on which you install the enhanced client, must meet the following additional requirements:

- Must be registered with ULN or have access to the ULN channels on a mirror.
- Must have access to the Internet or a host that is running a local ULN mirror. See Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror”.
- Must be running either the Oracle Linux 6 or Oracle Linux 7 operating system, with a supported version of either the Unbreakable Enterprise Kernel or the Red Hat Compatible Kernel installed.

  Note

  The Ksplice Enhanced client is not available on Oracle Linux 5.

  You can verify the kernel version by using the `uname -a` command. See Section 1.1.1, “Supported Kernels”.

- Must be running the same kernel that you want to update. It is assumed that the currently running kernel is the one that you want to update, as Ksplice applies updates only to the running kernel.

The following procedure describes how to install the enhanced client and Ksplice-aware libraries from ULN:

1. Using a browser, log in at https://linux.oracle.com with the ULN user name and password that you used to register the system, then perform the following steps:

   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 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`)

   c. On the System Summary page, select both the Userspace Ksplice channel and the Ksplice channel 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. Click **Save Subscriptions** and log out of ULN.

2. 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:

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

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

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

3. Log in as `root` on the system.

4. If `prelink` is installed, revert all prelinked binaries and dependent libraries to their original state and use the `yum` command to remove the `prelink` package.

   ```
   # prelink -au
   # yum remove prelink
   ```

   Note

   `prelink` is installed and enabled by default on Oracle Linux 6 but not on Oracle Linux 7.

5. Install the `ksplice` package.

   ```
   # yum install -y ksplice
   ```

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

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

The packages that are installed on the system include the following:

`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)` man page.

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

```
# yum update
```

To install only the libraries and not update any other packages, limit the update to the `ol6_x86_64_userspace_ksplice` or `ol7_x86_64_userspace_ksplice` channel, as appropriate:

```
# yum --disablerepo=* --enablerepo=ol6_x86_64_userspace_ksplice update
```

You can also use the following command:

```
# yum update glibc* openssl*
```

You can also use this client to perform kernel updates, in the same way that you are able to use the standard Uptrack client:

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

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

```
autoinstall = yes
```

8. Reboot the system so that it uses the new libraries.

On Oracle Linux 6:

```
# reboot
```

On Oracle Linux 7:

```
# systemctl reboot
```

The enhanced client uses the same configuration file (`/etc/uptrack/uptrack.conf`) as Ksplice Uptrack. See Section 3.3, “Configuring a Ksplice Uptrack Client”.

To manage the enhanced client, use the `ksplice` command, see Section 2.2, “Managing the Ksplice Enhanced Client With the ksplice Command”.

2.2 Managing the Ksplice Enhanced Client With the ksplice Command

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

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

```
# 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:

```
# ksplice xen list-targets
xen/x86_64/4.4.4OVM/Tue Aug  1 20:27:56 PDT 2017
```

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.

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

```
# 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 both the updates that have been applied to running processes and to the kernel. In this example, Ksplice has 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:

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

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

Use the `remove` subcommand to remove all of the updates from a process, as shown in the following example:

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

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

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

Note

If necessary, you can prevent Ksplice from patching specified executables and libraries. See Section 2.3, “Preventing the Ksplice Enhanced Client From Patching User-Space Processes and Libraries”.

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

To list the available Ksplice updates, use the `upgrade` subcommand as follows:

```
# ksplice -n kernel upgrade
```

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

```
# ksplice -y user upgrade
```

To list the available Ksplice updates for the Xen hypervisor, use the `upgrade` subcommand as follows:

```
# ksplice -n xen upgrade
```

After Ksplice applies updates to a running kernel, the kernel has an effective version that is different from 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:

```
# 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 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.

To view the updates that Ksplice Uptrack has made to the running kernel:

```
# ksplice kernel show
```

To view the updates that Ksplice Uptrack has made to the Xen hypervisor:

```
# ksplice xen show
```

To view the updates that are available to be installed:

```
# ksplice kernel show --available
```

To remove all updates from the kernel:

```
# ksplice kernel remove --all
```

To remove all updates from the Xen hypervisor:

```
# ksplice xen remove --all
```

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

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

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

For more information about using the `ksplice` command, see the `ksplice(8)` man page.

2.3 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 them in a `/etc/ksplice/blacklist.d` configuration file. The following example of a `localblacklist.conf` file 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-*`. As shown in the following example, the format of the rules is Python regular expressions:

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

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

2.4 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 have a direct connection to the Oracle Uptrack server or to ULN.

At regular intervals, you can download the latest Ksplice update packages to this server. After installing the offline Ksplice Enhanced client on your local systems, they can then connect to the local ULN mirror to receive updates. After you have set up a local ULN mirror to act as a Ksplice mirror, you can then configure your other systems to receive `yum` updates, as well as Ksplice updates. See Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror” for more information about configuring a local ULN mirror.

Configure a system as an offline Ksplice Enhanced client as follows:

1. Import the GPG key:

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

2. 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`:

   ```
   # 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:

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

3. In the `/etc/yum.repos.d` directory, create the file `local-yum.repo`, which contains entries such as the following for an Oracle Linux 6 yum client:

```
[local_ol6_x86_64_ksplice]
name=Ksplice for Oracle Linux $releasever - $basearch
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_x86_64_ksplice_userspace]
name=Ksplice aware userspace packages for Oracle Linux $releasever - $basearch
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/userspace/ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_latest]
name=Oracle Linux $releasever - $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_UEKR3_latest]
name=Unbreakable Enterprise Kernel Release 3 for Oracle Linux $releasever - $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/UEKR3/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_addons]
name=Oracle Linux $releasever - $basearch - addons
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/addons/$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 if you do this, you must edit the uptrack configuration as described in step 7.

The example configuration enables the `local_ol6_x86_64_ksplice`, `local_ol6_x86_64_ksplice_userspace`, `local_ol6_latest`, `local_ol6_UEKR3_latest`, and `local_ol6_addons` channels.

4. Test the configuration as follows:

   a. Clear the yum metadata cache:

   ```
   # yum clean metadata
   ```

   b. Use the `yum repolist` command to verify the configuration:

   ```
   # yum repolist
   ```

```
 Loaded plugins: rhnplugin, security
 This system is receiving updates from ULN.
0 packages excluded due to repository protections
repo id                          repo name                                      status
local_ol6_addons                 Oracle Linux 6 - x86_64 - latest               112
local_ol6_x86_64_ksplice         Ksplice for Oracle Linux 6 - x86_64            961
ol6_x86_64_userspace_ksplice     Ksplice aware userspace packages for
                                 Oracle Linux 6 - x86_64                        42
local_ol6_x86_64_latest          Oracle Linux 6 - x86_64 - latest               17,976
local_ol6_x86_64_UEKR3_latest    Unbreakable Enterprise Kernel Release 3
                                 for Oracle Linux 6 - x86_64 - latest           41
```

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).

5. If `prelink` is installed, revert all prelinked binaries and dependent libraries to their original state and use the `yum` command to remove the `prelink` package.

```
# prelink -au
# yum remove prelink
```

Note

`prelink` is installed and enabled by default on Oracle Linux 6 but not on Oracle Linux 7.

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

```
# yum install ksplice-offline
```

7. Insert a configuration directive into `/etc/uptrack/uptrack.conf` 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 and replace *`local_ol6_x86_64_ksplice_userspace`* with the same label you used for the Ksplice Userspace channel:

```
[User]
yum_userspace_ksplice_repo_name = local_ol6_x86_64_ksplice_userspace
```

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

```
# 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:

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

where `$RELEASE` is the update package that corresponds to the version of the hypervisor that is currently running, as shown in this example:

```
# 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 as the online version.

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

```
# yum update
```

   To install just the libraries and not any other packages, limit the update to the `ol6_x86_64_userspace_ksplice` channel or the `ol7_x86_64_userspace_ksplice` channel, for example:

```
# yum --disablerepo=* --enablerepo=ol6_x86_64_userspace_ksplice update
```

   Alternatively, you can use the following command:

```
# 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:

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

10. 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
```

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

    On Oracle Linux 6:

```
# reboot
```

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.

    On Oracle Linux 7:

```
# systemctl reboot
```

2.5 Removing the Ksplice Enhanced Client Software

The following procedure describes how to remove the Ksplice Enhanced client software as. For information about switching between online and offline Ksplice Installations, see Section 3.6, “Switching Between Online and Offline Ksplice Uptrack Installation Modes”.

To remove the Ksplice Enhanced client software:

```
# yum -y remove ksplice
```

To remove the offline version of the Ksplice Enhanced client software from a system, type the following command:

```
# yum -y remove ksplice-offline
```

To remove the Ksplice aware versions of the `glibc+openssl` packages from the system, follow these steps:

1. Unsubscribe the `ol7_x86_64_userspace_ksplice` channel from the Oracle Linux 7 yum repository and the `ol6_x86_64_userspace_ksplice` channel from the Oracle Linux 6 yum repository.

2. Downgrade the Ksplice aware channels.

```
# yum downgrade glibc{,-devel,-headers,-common} openssl{,-libs}
```

3. You can then remove all other Ksplice packages.

2.6 Using Known Exploit Detection on the Ksplice Enhanced Client

Oracle provides known exploit detection support for systems with the Ksplice Enhanced client installed. The 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 because it is also possible that tripwires can be triggered under normal operations, additional analysis of erroneous conditions might be required.

2.6.1 Running Known Exploit Detection on the Ksplice Enhanced Client

You can run the Oracle Ksplice known exploit detection feature on Oracle Linux 6 and Oracle Linux 7 systems that have the Ksplice Enhanced client installed. Note that the feature works on both online and offline clients.

To run known exploit detection with the default configuration:

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

```
# 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:

```
# 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`, then 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.

2.6.2 Setting Up Email Alerts for Exploit Attempts

The default configuration for the Oracle Ksplice known exploit detection feaure 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
```

2.6.3 Temporarily Disabling and Re-Enabling Tripwires

For troubleshooting purposes, you can disable or re-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' > /proc/sys/kernel/known_exploit_detection_tripwires
```

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

```
# echo -n '+CVE-2019-12345' > /proc/sys/kernel/known_exploit_detection_tripwires
```

Chapter 3 Working With Ksplice Uptrack

Table of Contents

This chapter describes how to configure and use Ksplice Uptrack on a running system. For more information about Ksplice Uptrack, go to http://www.ksplice.com/.

For overview and prerequisite task information, see Chapter 1, *About Oracle Ksplice*

3.1 Installing Ksplice Uptrack From ULN

If you have an Oracle Linux Premier support subscription, a Premier Limited subscription, or an Oracle Premier Support for Systems and Operating Systems support subscription, you are automatically registered to use Oracle Ksplice. You can configure your registered systems to use Ksplice Uptrack through the Ksplice for Oracle Linux channel on ULN by using the `yum` command. See Section 1.3.2, "About Oracle Ksplice and ULN Registration".

The system on which you want to install Ksplice Uptrack must also meet the following requirements:

- Must have access to the Internet.
- Must be registered with ULN.
- Must be running Oracle Linux 5, Oracle Linux 6, or Oracle Linux 7 with a supported version of either the Unbreakable Enterprise Kernel or the Red Hat Compatible Kernel installed. You can verify the kernel version by using the `uname -a` command. See Section 1.1.1, "Supported Kernels".
- The kernel that is currently running is also the kernel you want to update, as Ksplice Uptrack applies updates only to the running kernel.

To install Ksplice Uptrack from ULN, follow these steps:

1. Log in as `root` on the system.
2. 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:

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

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

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

3. Using a browser, log in at https://linux.oracle.com with your ULN user name and password.

   Perform the following 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, select the Ksplice for Oracle Linux channel for the correct release and your system's architecture (`i386` or `x86_64`) from the list of available channels, then click the right arrow (**>**) to move it to the list of subscribed channels.

   d. Click **Save Subscriptions** and log out of ULN.

4. On your system, use the `yum` command to install the `uptrack` package.

   ```
   # 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
   ```

5. To enable the 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 Section 3.3, “Configuring a Ksplice Uptrack Client”.

For information about managing Ksplice updates, see Section 3.4, “Managing Ksplice Updates With the uptrack-upgrade Command”.

3.2 Installing Ksplice Uptrack Within the Oracle Cloud Infrastructure

If you are an Oracle Cloud Infrastructure customer, you can use Ksplice on any of the Oracle Linux systems that are hosted in your cloud environment. You do not need to register through ULN to use Ksplice. Any system that runs inside of the Oracle Cloud Infrastructure has automatic access to the Ksplice servers and all of the Ksplice updates.

To install Ksplice Uptrack on a system running on Oracle Cloud Infrastructure:

1. Log in as the `root` user on the system.

2. 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:

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

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

     ```
     # setenv http_proxy=http://proxy_URL:http_port
     ```

```
# setenv https_proxy=http://proxy_URL:https_port
```

3. Download the Ksplice installer for Oracle Cloud Infrastructure:

```
# wget -N https://www.ksplice.com/uptrack/install-uptrack-oc
```

4. Run the installer script.

   To enable the automatic installation of updates:

```
# sh install-uptrack-oc --autoinstall
```

   If you do not want Ksplice to automatically install updates, run the script without the command-line switch:

```
# sh install-uptrack-oc
```

For information about configuring Ksplice Uptrack, see Section 3.3, “Configuring a Ksplice Uptrack Client”.

For information about managing Ksplice updates, see Section 3.4, “Managing Ksplice Updates With the uptrack-upgrade Command”.

3.3 Configuring a 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, to install updates automatically at boot time, and to 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 e-mail 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 e-mail 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.

3.4 Managing Ksplice Updates With the uptrack-upgrade Command

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

To list the available Ksplice updates, use the `uptrack-upgrade` command:

```
# uptrack-upgrade -n
```

Install all available Ksplice updates as follows:

```
# uptrack-upgrade -y
```

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

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

```
# uptrack-uname -r
3.8.13-55.1.1.el6uek.x86_64
```

The `uptrack-uname` command supports the commonly used `uname` flags, including `-a` and `-r`, and 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 the updates that Ksplice has made to the running kernel as follows:

```
# uptrack-show
```

View the updates that are available for installation as follows:

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

Remove all updates from the kernel as follows:

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

To 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.

3.5 Removing the Ksplice Uptrack Client Software

You can remove the Ksplice Uptrack software from a system as follows:

```
# yum -y remove uptrack
```

To remove the offline Ksplice Uptrack software from a system, use the following command:

```
# yum -y remove uptrack-offline
```

3.6 Switching Between Online and Offline Ksplice Uptrack Installation Modes

If you want to switch from one Ksplice client software version 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, and then install the new version of the Ksplice client software.

Note

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 Ksplice Offline Enhanced client software, you would need to first remove the Ksplice Uptrack client software, and then install the Ksplice Offline Enhanced client software as follows:

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

To switch from an offline installation to an online installation, for example, switch from the Ksplice Uptrack Offline client software to the Ksplice Uptrack client software, you would run the following commands:

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

3.7 Working With the Ksplice Uptrack Client in Offline Mode

The Ksplice Offline client eliminates the need having a server on your intranet with 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. The following tasks describe how to configure systems to use the Ksplice Offline client.

Note

You cannot use the web interface or the Ksplice Uptrack API to monitor systems that are running Ksplice Offline client, as such systems are not registered with https://uptrack.ksplice.com.

3.7.1 Configuring Ksplice Uptrack Clients for Offline Mode

Prior to configuring a Ksplice Offline client, you must set up a local ULN mirror that can act as a Ksplice mirror. See Section 1.3.3, “Configuring a Local ULN Mirror to Act as a Ksplice Mirror”. After you set up a local ULN mirror that can act as a Ksplice mirror, you can configure your other systems to receive `yum` and Ksplice updates.

You can also configure Ksplice Offline Clients by creating software channels in Spacewalk that can act as a Ksplice mirror. For instructions, see "Installing and Configuring Existing Client Systems as Ksplice Offline Clients" in Chapter 12 of the *Spacewalk 2.7 for Oracle Linux Client Life Cycle Management Guide*.

To configure a system as a Ksplice Offline client by setting up a local ULN mirror:

1. Import the GPG key:

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

2. 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`:

```
# 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:

```
# cd /etc/yum.repos.d
# 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 6 yum client:

```
[local_ol6_x86_64_ksplice]
name=Ksplice for Oracle Linux $releasever - $basearch
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_latest]
name=Oracle Linux $releasever - $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_UEKR3_latest]
name=Unbreakable Enterprise Kernel Release 3 for Oracle Linux $releasever - $basearch - latest
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/UEKR3/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1

[local_ol6_addons]
name=Oracle Linux $releasever - $basearch - addons
baseurl=http://local_uln_mirror/yum/OracleLinux/OL6/addons/$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_ol6_x86_64_ksplice`, `local_ol6_latest`, `local_ol6_UEKR3_latest`, and `local_ol6_addons` channels. Note that 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.

3. Install the Ksplice Offline client package:

```
# yum -y install uptrack-offline
```

4. To test the configuration:

   a. Clear the `yum` metadata cache:

   ```
   # yum clean metadata
   ```

   b. Use the `yum repolist` command to verify the configuration, for example:

```
# yum repolist
 Loaded plugins: rhnplugin, security
 This system is receiving updates from ULN.
0 packages excluded due to repository protections
repo id                         repo name                                       status
local_ol6_addons                Oracle Linux 6 - x86_64 - latest                112
local_ol6_x86_64_ksplice        Ksplice for Oracle Linux 6 - x86_64             961
local_ol6_x86_64_latest         Oracle Linux 6 - x86_64 - latest                17,976
local_ol6_x86_64_UEKR3_latest   Unbreakable Enterprise Kernel Release 3
                                for Oracle Linux 6 - x86_64 - latest            41
```

If `yum` 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).

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

   For an Oracle Linux 5 client, use this command:

   ```
   # yum -y install uptrack-updates-`uname -r`.`uname -m`
   ```

   For an Oracle Linux 6 or Oracle Linux 7 client, use this command:

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

   As new Ksplice updates are made available, you can use this command to pick up these updates and apply them. It is recommended that you set up an `anacron` script to perform this task. For example, the following script named `uptrack-updates` in `/etc/cron.daily` on an Oracle Linux 6 system would run once every day:

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

Note

The script must be executable and be owned by `root`. It is important to include the `-y` option for the `yum` command if you intend to script this, as the command hangs and waits for user input if this option is not used.

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

3.8 Updating the Ksplice Uptrack Client to a Specific Effective Kernel Version

Under some circumstances, you might want to limit the set of updates that `uptrack-upgrade` installations. For example, the security policy at your site might require a senior administrator to approve Ksplice updates before you can install them 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.

The options for selecting a specific effective version are only available in the Ksplice Offline client for use with the offline update RPM packages.

Note

Ksplice is intended to provide the latest security and stability fixes, and the goal is to get the effective kernel up-to-date as soon as possible. Choosing a specific

> effective kernel version is only intended to allow the offline update RPM package to be updated without immediately applying the latest available patches bundled in that package. This enables production systems to remain temporarily at a tested update level, while the latest updates are tested in an integration or UAT environment.

To update a system to a specific effective kernel version, follow these steps:

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

   For an Oracle Linux 5 client, use this command:

   ```
   # yum -y install uptrack-updates-`uname -r`.`uname -m`
   ```

   For an Oracle Linux 6 or Oracle Linux 7 client, use this command:

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

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

   ```
   # uptrack-uname -r
   3.8.13-55.1.1.el6uek.x86_64
   ```

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

   ```
   # uptrack-upgrade --list-effective
   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 this example:

   ```
   # uptrack-remove --all
   ...
   # uptrack-uname -r
   3.8.13-44.1.1.el6uek.x86_64
   ```

5. You can set the effective kernel version that you want the system to use in either of the following ways:

   - 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:

     ```
     # uptrack-upgrade --effective="3.8.13-44.1.5.el6uek.x86_64/#2 SMP Wed Nov 12 14:23:31 PST 2014"
     ...
     Effective kernel version is 3.8.13-44.1.5.el6uek
     # uptrack-uname -r
     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 master copy.

3.9 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.

3.9.1 Installing and Configuring the SNMP Plugin

The following prerequisites apply to the system that you want to monitor:

- 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.

To install and configure the SNMP plugin on a system that you want to monitor using SNMP, follow these steps:

1. 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.
2. As `root`, use the `yum` command to install the `ksplice-snmp-plugin` package on the system:

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

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

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

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

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

   ```
   # 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.

5. Restart the SNMP service:

```
# service snmpd restart
```

For information about configuring SNMP, see the documentation at http://www.net-snmp.org/docs/readmefiles.html and the `snmpd(8)` and `snmpd.conf(5)` man pages.

3.9.2 Testing the SNMP Plugin

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

Display the installed version of Ksplice as follows:

```
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceVersion
KSPLICE-UPTRACK-MIB::kspliceVersion.0 = STRING: 1.2.12
```

Check if all of the available updates for a kernel have been installed as follows:

```
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceStatus
KSPLICE-UPTRACK-MIB::kspliceStatus.0 = STRING: outofdate
```

In the previous example, the kernel is shown as being out of date with regards to updates.

Display and compare the kernel that is installed on disk with the Ksplice effective version as follows:

```
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceBaseKernel
KSPLICE-UPTRACK-MIB::kspliceBaseKernel.0 = STRING: 2.6.18-274.3.1.el5
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceEffectiveKernel
KSPLICE-UPTRACK-MIB::kspliceEffectiveKernel.0 = STRING: 2.6.18-274.3.1.el5
```

The base kernel version and effective kernel version are shown as being the same, which implies that no updates have been applied.

Display a list of all updates that have been applied to the kernel as follows:

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

In the previous example, no updates have been applied, which confirms why the base kernel version and effective kernel version are the same and why the kernel is out of date.

Display a list of updates that can be installed as follows:

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

After fully upgrading your kernel by using Ksplice Uptrack, you can run the following `snmpwalk` commands to show that the kernel is up to date, that there are no updates available for installation, and that the patches that have been applied:

```
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceStatus
KSPLICE-UPTRACK-MIB::kspliceStatus.0 = STRING: uptodate
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::kspliceAvailTable
$ snmpwalk -v 1 -c public -O e localhost KSPLICE-UPTRACK-MIB::ksplicePatchTable
KSPLICE-UPTRACK-MIB::ksplicepatchIndex.0 = INTEGER: 0
KSPLICE-UPTRACK-MIB::ksplicepatchIndex.1 = INTEGER: 1
KSPLICE-UPTRACK-MIB::ksplicepatchIndex.2 = INTEGER: 2
...
```

Chapter 4 Working With the Ksplice Uptrack API

Table of Contents

This chapter describes the programming interfaces for Ksplice Uptrack.

4.1 About the Ksplice Uptrack API

The Ksplice Uptrack API is a RESTful web API that enables you to query the status of your machines that are running Ksplice Uptrack. You can use the command-line tools that come with the Python bindings, write your own custom scripts using the bindings, or write your own interface using HTTP requests.

The API provides information about the updates that machines have, out-of-date, inactive, or unsupported machines, and more. The Python bindings come with the `check_uptrack` and `check_uptrack_local` plugins for Nagios, which enables you to monitor the status of your machines.

You cannot use the Ksplice Uptrack API to monitor systems that are running Ksplice Offline client, as such systems are not registered with https://uptrack.ksplice.com.

4.2 Viewing Your API User Name and API Key

To view your API user name and API key, log in to https://uptrack.ksplice.com and select the **Settings** tab.

4.3 Generating a New API Key

To generate a new API key, follow these steps:

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

Note

This action invalidates your existing key.

4.4 Installing the API Command-Line Tools

The command-line API 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`.

Install the command-line API tools as follows:

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.

   The previously listed subscriptions automatically register your system to use Oracle Ksplice. See Section 1.3.2, “About Oracle Ksplice and ULN Registration” for more details.

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

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

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

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

4.5 Ksplice Uptrack API Commands

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

4.5.1 About the uptrack-api-authorize Command

The `uptrack-api-authorize` command uses the `authorize` API call to change the authorization for a single machine, for 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://uptrack.ksplice.com and select the **Settings** tab.

The UUID of a registered machine is stored in `/var/lib/uptrack/uuid` on the machine. An example of a UUID is `e82ba0ae-ad0a-4b92-a776-62b502bfd29d`.

4.5.2 About the uptrack-api-describe Command

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

```
$ uptrack-api-describe -u api_username -k api_key uuid
    prod1.mydom.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.mydom.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.
```

4.5.3 About the uptrack-api-list Command

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

4.5.4 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@mydom.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@mydom.com
$ export UPTRACK_API_KEY=3af3c2c1ec407feb0fdc9fc1d8c4460c
$ uptrack-api-describe --this-machine
```

4.5.5 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]
```

where *`protocol`* is either specifed 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.

4.6 About the API Implementation

The following information pertains to the implementation of the Ksplice Uptrack API.

4.6.1 API Version

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

4.6.2 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.

4.6.3 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.

4.6.4 Supported API Requests

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

4.6.4.1 GET /api/1/machines

`GET /api/1/machines` 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.

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

`GET /api/1/machine/`*`$UUID`*`/describe` 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 that the system had 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.

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

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

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

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

The content is a dictionary of the following form:

```
{group_name: string}
```

where `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.

4.6.5 Interaction Sample

The following is a sample of an interaction that might take place when using the Uptrack API. This example is provided as a reference *only*.

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

The following request for a list of registered machines 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@mydom.com
X-Uptrack-Key: 3af3c2c1ec407feb0fdc9fc1d8c4460c
```

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

```
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.mydom.com",
  "authorization": "allowed", "autoinstall": true,
  "last_seen": "2010-04-26T18:03:43Z", "mmap_min_addr": 4096,
  "uptrack_client_version": "1.2.1"}]
```

4.7 Configuring the check_uptrack Nagios Plugin

Note

The Nagios software is not included with the python-ksplice-uptrack package. For information about obtaining and using Nagios, go to 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@mydom.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]
```

where *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 $ARG2$
}
```

4.7.1 Using the Nagios Plugins

To monitor all of your machines, run the following command:

```
# /usr/lib/nagios/plugins/check_uptrack
```

This command produces a summary about your machines in the standard Nagios plug-in format, for example:

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

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

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

To monitor the local machine, you can use the `check_uptrack_local` plugin.

```
# /usr/lib/nagios/plugins/check_uptrack_local
```

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

Note

`check_uptrack_local` reads the local Uptrack update cache. It does not use the settings in `/etc/uptrack-api.conf`.

4.8 For More Information About the Ksplice Uptrack API

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