Ksplice – Is Rebooting Your Oracle Linux Database Server Now Obsolete?

Robert Bialek
Trivadis GmbH
Munich, DE

Keywords:

Ksplice, Ksplice Uptrack, Uptrack API, Oracle Enterprise Linux, Zero Downtime Kernel Patches, High Availability, Dynamic Software Updating, ELSA, ELBA.

Introduction

Applying important Linux kernel patches and security updates might be a painful task. While many operating system updates are online and might be applied without disrupting the running services, the operating system kernel is always an exception. To activate the new version of the installed patched kernel a reboot is required, which leads to a service outage as well as loss of all software states on that server. Moreover, in case of a problem with the new version, a fallback to the old kernel would again lead to a server downtime.

Ksplice patches are mostly well-known as a way to address very efficiently and quickly a kernel security bug. But actually, they are more than that. A detailed look at the released Ksplice patches reveals that they also include many fixes for critical kernel bugs. Furthermore, Ksplice technology is not only limited to security or critical kernel bugs – it can also be used by Oracle Support to create online kernel diagnostic patches which help to turn, on demand, the running kernel into a debug mode.

For kernel bugs or CVEs (Common Vulnerabilities and Exposures) classified by Oracle as important, appropriate Ksplice patches will be created and made available via ULN (Unbreakable Linux Network).

This paper will describe how to use and configure Ksplice to apply or rollback Oracle Enterprise Linux kernel patches without rebooting. At the end, we will assess the use case of this technology in an Oracle database environment, considering the implications and patch management of the whole Oracle database software stack.

Ksplice History and Availability

Ksplice software called Ksplice Uptrack was created by four MIT students and released under the GNU General Public License version 2. The Ksplice, Inc. software company was founded back in 2008 and provided initially prebuilt kernel patches on a subscription basis for enterprise Linux distributions like Red Hat, and free of charge for Ubuntu Desktop and Fedora Linux.

On July 21, 2011 Oracle Corporation acquired Ksplice and made this technology a standard feature of the Oracle Linux Premier Support level subscription.

As of now, Ksplice Uptrack is available for the following Oracle Enterprise Linux kernel versions:

- Oracle Linux 6: all UEK (Unbreakable Enterprise Kernel) as well as Red Hat compatible kernel versions.
- Oracle Linux 5: UEK starting with the version 2.6.32-100.28.9; Red Hat compatible kernels starting with the version 2.6.18-164; Red Hat compatible kernels with bug fixes added by Oracle starting with the version 2.6.18-238.0.0.0.1

Ksplice Uptrack is not available for other Linux distributions supported for Oracle database, like SUSE Linux Enterprise Server or Red Hat Enterprise Linux (though, for RHEL customers there is a 30-day Ksplice trial).

Ksplice Uptrack is still available, free of charge for Ubuntu Desktop as well as Fedora Linux.

Ksplice – How it Works?

The core functionality of Ksplice rebootless patches is based on dynamically loadable kernel modules. This makes the technology very flexible and powerful. In case a traditional patch does not change the sematic of persistent kernel data structures, Ksplice can create an online update even without a developer writing any new code! Theoretically, this technology might be used in the future to patch almost every part of the kernel code, by applying a new patched version directly into memory, without a downtime or a disruption to application running in user mode. This approach allows for not only applying a new patch online, but also for rolling back patches without the need to reboot a system.

Ksplice patches always replace a whole kernel function, even though only some code lines require a change. Each patch consists in most cases amongst other files of three core kernel modules:

1. *ksplice-<kspliace_patch_id>.ko* – kernel module used by Ksplice Uptrack to perform safety checks, locating the function to be patched in memory and the actual patch apply or rollback procedure.
2. *ksplice-<kspliace_patch_id>_vmlinux-new.ko* – includes an object file which contains the new patched kernel function code.
3. *ksplice-<kspliace_patch_id>_vmlinux-old.ko* – includes the object file with the old version of the kernel function. It will be used to locate the function to be replaced in memory. Eventually, the module will be unloaded.

Example of kernel modules for the CVE-2013-2634 (*Kernel leak in data center bridging and netlink*) addressed in the Ksplice patch oviq2kch:

```
# tar -xvzf ksplice-oviq2kch.tar.gz | grep .ko

ksplice-oviq2kch/ksplice-oviq2kch.ko
ksplice-oviq2kch/ksplice-oviq2kch_vmlinux-new.ko
ksplice-oviq2kch/ksplice-oviq2kch_vmlinux-old.ko
```

Application of a new rebootless patch executed by the Ksplice Uptrack consists of many steps, a few of which are depicted in the illustration 1:

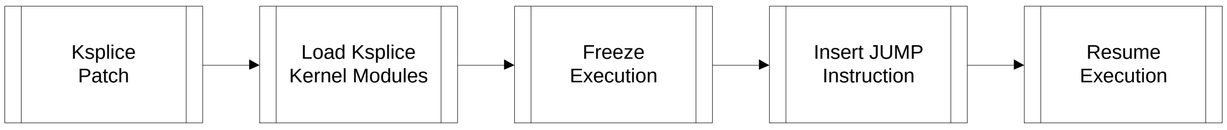


Illustration 1 – Application of a Ksplice patch.

Before touching the old version of the code to be replaced, Ksplice Uptrack needs to find in memory the function and perform a byte-by-byte comparison of the object file. The check is necessary, to make sure the code is exactly the same, and has not been modified. In case the safety check has succeeded, the normal execution of other programs on that system will be stopped for a short period of time, (according to the Ksplice developers for about 0.7ms). The function to be replaced cannot be executed by any processor – in other case Ksplice Uptrack will exit the apply procedure. The code replacement is actually implemented by injecting at the beginning of the old function in memory a JUMP instruction to the new code. The JUMP instruction will cause some more CPU cycles to be used, to execute the code, but the overhead is actually negligible. Additionally, for each applied Ksplice patch there is a small memory overhead – on average about 200kB.

As described above, Ksplice patches will be applied only in memory, not on disk. In the default configuration of the Ksplice Uptrack, a server reboot will not lead to a permanent loss of all Ksplice patches; they will be re-applied to the running kernel in a very early boot phase by the *uptrack* service, even before activating the network:

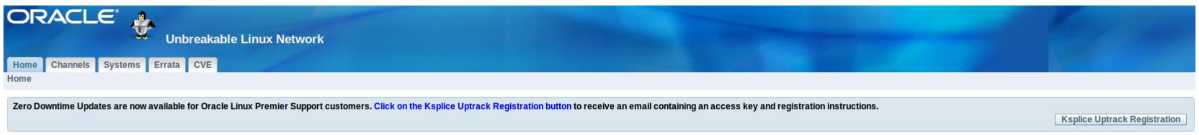
```
/etc/rc3.d/S09uptrack -> ../init.d/uptrack
/etc/rc3.d/S60uptrack-late -> ../init.d/uptrack-late
```

The second Uptrack INIT script, *uptrack-late* will prefetch new available updates at the boot time. All Ksplice patches will be stored in */var/cache/uptrack*.

The question is: do I still need to use YUM to update my system, in case I use Ksplice? The answer is: Yes. Ksplice Uptrack will not patch any files on a disk, nor will it provide all regular kernel RPM packages, updates to 3rd party kernel modules or updates to any important shared libraries (e.g. GLIBC, etc.). Ksplice is an add-on, which gives you the possibility to fix kernel security holes at once – no downtime or disruption. But to install the regular updates, you still need to use your regular package management system.

Installing and Configuring Ksplice Uptrack

Ksplice Uptrack can be installed online without reboot or activating a specially prepared kernel. The requirements to use Uptrack are an Oracle Linux Premium Support subscription as well as a Ksplice access key, which can be requested via *http://linux.oracle.com site*:

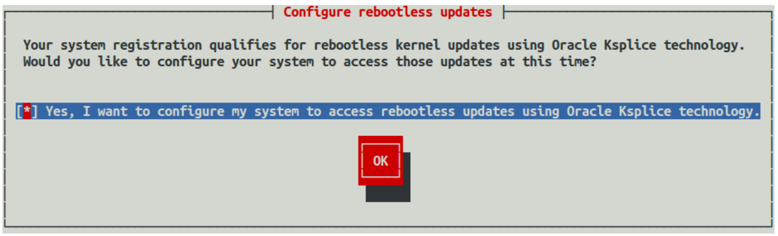


Zero Downtime Updates are now available for Oracle Linux Premier Support customers. Click on the Ksplice Uptrack Registration button to receive an email containing an access key and registration instructions.

Illustration 2 – Ksplice Uptrack registration at http://linux.oracle.com

There are two ways to subscribe a server to the *Ksplice for Oracle Linux* channel at the Oracle ULN:

- Channel subscription during the server registration (*uln_register*)



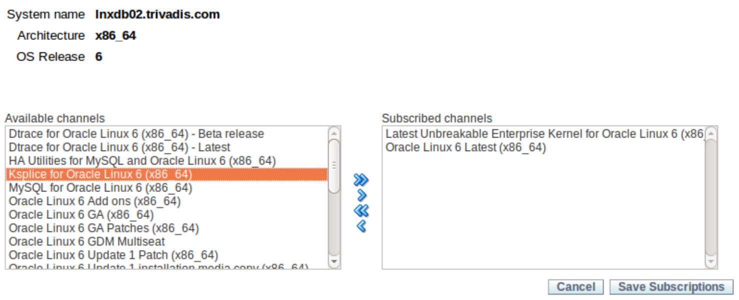
```
Configure rebootless updates

Your system registration qualifies for rebootless kernel updates using Oracle Ksplice technology.
Would you like to configure your system to access those updates at this time?

[*] Yes, I want to configure my system to access rebootless updates using Oracle Ksplice technology.
```

Illustration 3 – Subscription to the Ksplice channel during a server registration

- At a later time, you can change the server *Subscribed Channels* directly in the ULN



System name **lnxdb02.trivadis.com**
Architecture **x86_64**
OS Release **6**

Available channels
Dtrace for Oracle Linux 6 (x86_64) - Beta release
Dtrace for Oracle Linux 6 (x86_64) - Latest
HA Utilities for MySQL and Oracle Linux 6 (x86_64)
Ksplice for Oracle Linux 6 (x86_64)
MySQL for Oracle Linux 6 (x86_64)
Oracle Linux 6 Add ons (x86_64)
Oracle Linux 6 GA (x86_64)
Oracle Linux 6 GA Patches (x86_64)
Oracle Linux 6 GDM Multiseat
Oracle Linux 6 Update 1 Patch (x86_64)
Oracle Linux 6 Update 1 installation media copy (x86_64)

Subscribed channels
Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86
Oracle Linux 6 Latest (x86_64)

Illustration 4 – Changes to the subscribed channels at http://linux.oracle.com

- For servers not having internet access, there is a possibility to use the Ksplice Uptrack offline client with a local YUM repository, described later in this paper.

After subscribing a server to the correct channel and before starting to use the new framework, the last step is to install the Ksplice Uptrack package (for dependencies also the packages *uptrack-PyYAML* and *uptrack-libyaml* will be installed):

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

During the Uptrack installation process, the Ksplice Access Key will be retrieved from the ULN and added to the configuration file */etc/uptrack/uptrack.conf*. The most important Uptrack configuration parameters are listed in the table below.

accesskey =	Ksplice Access Key will be retrieved automatically from the ULN.
install_on_reboot = yes	During a server reboot, Uptrack will re-install the same set of updates that were present before. If you don't want to install Ksplice patches temporarily during reboot, create with *touch* the file */etc/uptrack/disable*.
upgrade_on_reboot = no	If set to yes, all available updates will be installed automatically at a boot time, even if the updates were not present before the reboot.
autoinstall = no	If set to yes, a default Uptrack cron job (*/etc/cron.d/uptrack*) which starts twice each hour to download any new Ksplice updates, will install the new patches automatically. If set to no (default), the updates need to be installed manually.
https_proxy =	If set, Uptrack will use the proxy to contact the Oracle Ksplice server.

To download updates, the Ksplice Uptrack client connects to *https://updates.ksplice.com:443*. Make sure that your firewall configuration is aware of it. To use internet proxy, either use the *https_proxy* parameter mentioned in the table above, or set the shell variables *https_proxy* or *http_proxy*.

Applying and Removing Ksplice Updates

The management of the Ksplice patches is not a difficult or a complex task. The Uptrack framework performs all the necessary steps internally, hiding the complexity from the administrator. The patch management on an Oracle Enterprise Linux can be performed with command line tools. On Fedora and Ubuntu there is also additionally a graphical interface for that purpose.

Ksplice patches applied by the Uptrack, will not reflect in the kernel version shown by the standard Linux tools, like *uname*:

```
# uname -r
2.6.32-100.34.1.el6uek.x86_64
```

To query the in memory effective patched kernel version, use *uptrack-uname*:

```
# uptrack-uname -r
2.6.32-400.29.3.el6uek.x86_64
```

To download and apply all Ksplice patches available for your system:

```
# uptrack-upgrade -y
...
Installing [seloxg7f] CVE-2012-6544: Information leak in Bluetooth
                                     L2CAP socket name.
Installing [hmlhk1ra] CVE-2013-2206: NULL pointer dereference in
                                     SCTP duplicate cookie handling.

Your kernel is fully up to date.
Effective kernel version is 2.6.32-400.29.3.el6uek
```

To install only one Ksplice patch, use *uptrack-install <ksplice_patch_id>*:

```
# uptrack-install xxqhbxgk -y
The following steps will be taken:
Install [xxqhbxgk] CVE-2013-1929: Buffer overflow in TG3 VPD
                                  firmware parsing.
Installing [xxqhbxgk] CVE-2013-1929: Buffer overflow in TG3 VPD
                                     firmware parsing.
Effective kernel version is 2.6.18-348.6.1.el5
```

Rolling back patches can also be performed online, by using *uprack-remove <ksplice_patch_id>*. In many cases though it will be not possible to remove only one patch, but rather because of dependencies a set of them. To remove all installed patches, use *uprack-remove --all.*

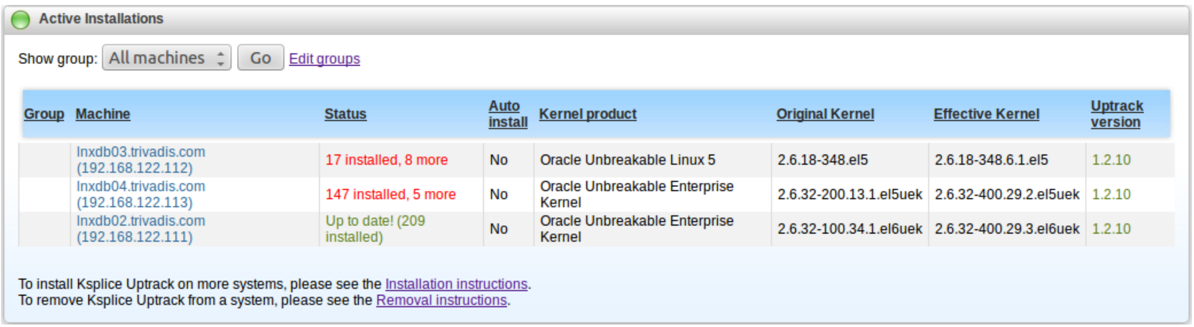
As already mentioned in the *Ksplice – How it Works* part, the code being patched cannot be executed by any processor or the performed safety checks will fail, exiting the apply procedure. This might happen especially with some kernel code on a heavily loaded system, as in the example below (though re-trying many times helps in most cases):

```
The following actions failed:
Install [t6yfkr0b] NULL pointer dereference in SCSI device removal.
Ksplice was unable to install the update because one or more
programs are constantly using the kernel functions patched by this
update.
  - kworker/0:2 (pid 32)
  - kworker/0:1 (pid 18)
```

With *uptrack-show* you can check all the installed Ksplice patches on the system (with additional *--available* option, you can check which patches have been already downloaded, but have not been installed yet):

```
# uptrack-show
Installed updates:
[hlhx6na6] Clear garbage data on the kernel stack when handling
           signals.
[6b96sdwt] CVE-2012-1568: A predictable base address with shared
                          libraries and ASLR.
[3mkg7zqm] CVE-2012-4444: Prohibit reassembling IPv6 fragments when
                          some data overlaps.
[b4d1gxxh] CVE-2012-3400: Buffer overflow in UDF parsing.
[v92kuk35] CVE-2013-0268: /dev/cpu/*/msr local privilege escalation.
[7q7ua9bf] CVE-2013-0871: Privilege escalation in PTRACE_SETREGS.
[4u8j0tof] CVE-2012-6537: Kernel information leaks in network
                          transformation subsystem.
[sufcr9jy] CVE-2013-1826: NULL pointer dereference in XFRM buffer
                          size mismatch.
...
Effective kernel version is 2.6.18-348.6.1.el5
```

Ksplice Uptrack also comes with a web interface (*https://uptrack.ksplice.com*) which centralized the patch information about all servers subscribed to the ULN Ksplice Channel.


Active Installations

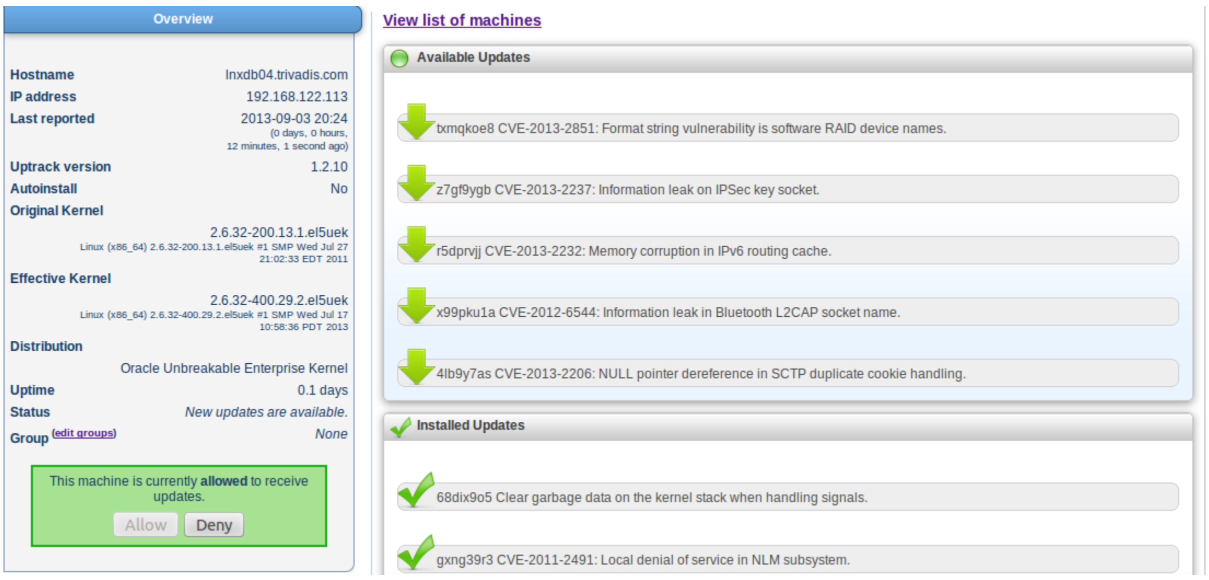
Show group: All machines Go Edit groups

Group	Machine	Status	Auto install	Kernel product	Original Kernel	Effective Kernel	Uptrack version
	lnxdb03.trivadis.com (192.168.122.112)	17 installed, 8 more	No	Oracle Unbreakable Linux 5	2.6.18-348.el5	2.6.18-348.6.1.el5	1.2.10
	lnxdb04.trivadis.com (192.168.122.113)	147 installed, 5 more	No	Oracle Unbreakable Enterprise Kernel	2.6.32-200.13.1.el5uek	2.6.32-400.29.2.el5uek	1.2.10
	lnxdb02.trivadis.com (192.168.122.111)	Up to date! (209 installed)	No	Oracle Unbreakable Enterprise Kernel	2.6.32-100.34.1.el6uek	2.6.32-400.29.3.el6uek	1.2.10

To install Ksplice Uptrack on more systems, please see the Installation instructions.
To remove Ksplice Uptrack from a system, please see the Removal instructions.

Illustration 5 – Server overview page at https://uptrack.ksplice.com

To get detailed information about the applied and available Ksplice patches, you can perform a server drill-down, as in the example Illustration 6 below.


Overview

Hostname	lnxdb04.trivadis.com
IP address	192.168.122.113
Last reported	2013-09-03 20:24 (0 days, 0 hours, 12 minutes, 1 second ago)
Uptrack version	1.2.10
Autoinstall	No
Original Kernel	2.6.32-200.13.1.el5uek Linux (x86_64) 2.6.32-200.13.1.el5uek #1 SMP Wed Jul 27 21:02:33 EDT 2011
Effective Kernel	2.6.32-400.29.2.el5uek Linux (x86_64) 2.6.32-400.29.2.el5uek #1 SMP Wed Jul 17 10:58:36 PDT 2013
Distribution	Oracle Unbreakable Enterprise Kernel
Uptime	0.1 days
Status	*New updates are available.*
Group (edit groups)	*None*

This machine is currently **allowed** to receive updates.

View list of machines

Available Updates

txmqkoe8 CVE-2013-2851: Format string vulnerability is software RAID device names.
z7gf9ygb CVE-2013-2237: Information leak on IPSec key socket.
r5dprvjj CVE-2013-2232: Memory corruption in IPv6 routing cache.
x99pku1a CVE-2012-6544: Information leak in Bluetooth L2CAP socket name.
4lb9y7as CVE-2013-2206: NULL pointer dereference in SCTP duplicate cookie handling.

Installed Updates

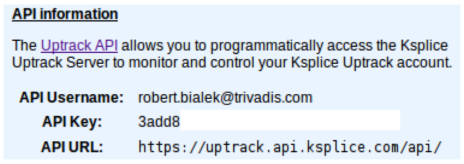
68dix9o5 Clear garbage data on the kernel stack when handling signals.
gxng39r3 CVE-2011-2491: Local denial of service in NLM subsystem.

Illustration 6 – Detailed server information at https://uptrack.ksplice.com

Most of the information available at *https://uptrack.ksplice.com* can be queried also centrally from one server, by using the Uptrack API. To use the Uptrack API install on a server the *python-ksplice-uptrack* RPM and insert the api_key as well as the username into the */etc/uptrack-api.conf* configuration file.

```
# cat /etc/uptrack-api.conf
[uptrack]
username = robert.bialek@trivadis.com
api_key = 3add8__________________________
```

The API Key can be queried or changed at *https://uptrack.ksplice.com*:


API information

The Uptrack API allows you to programmatically access the Ksplice Uptrack Server to monitor and control your Ksplice Uptrack account.

API Username:	robert.bialek@trivadis.com
API Key:	3add8
API URL:	`https://uptrack.api.ksplice.com/api/`

Illustration 7 – Uptrack API information at https://uptrack.ksplice.com

To list the up-to-date status of all your registered servers use the *uptrack-api-list* command:

```
# uptrack-api-list
- lnxdb03.trivadis.com (192.168.122.112): outofdate
- lnxdb04.trivadis.com (192.168.122.113): outofdate
- lnxdb02.trivadis.com (192.168.122.111): uptodate
```

Detailed information can be queried centrally by using *uptrack-api-describe UUID* (the UUID used as the first parameter is stored for each server locally in */var/lib/uptrack/uuid*):

```
# uptrack-api-describe 9160ca3c-db4d-4e4b-b3f0-c3c6ea259ca1
```

```
lnxdb04.trivadis.com (192.168.122.113)
Effective kernel: 2.6.32-400.29.2.el5uek
This machine is active
Last seen on 2013-09-03T20:24:10Z
OS status: Out of date:
  * Install txmqkoe8 CVE-2013-2851: Format string vulnerability is
                                    software RAID device names.
  * Install z7gf9ygb CVE-2013-2237: Information leak on IPSec key
                                    socket.
  * Install r5dprvjj CVE-2013-2232: Memory corruption in IPv6
                                    routing cache.
```

Ksplice Uptrack Offline Client

In many cases it is not practical, or even not possible to download the patches for each server directly from the Oracle Ksplice Update server. To support an offline Uptrack client, Oracle bundles all available Ksplice patches for each supported kernel version to one RPM package.

To use the offline Uptrack client, download locally all Ksplice patches by subscribing your local YUM server to the Ksplice channel. For details how to create a local YUM server and to use the Oracle script 167283.sh, see: *http://docs.oracle.com/cd/E37670_01/E37355/html/ol_createlocal_repo.html*

Create a YUM repo file for each Ksplice client, example:

```
# cat /etc/yum.repos.d/local_ksplice.repo
```

```
[ol5_ksplice]
name=Oracle Linux 5 Ksplice Updates
baseurl=http://lnxrepo.trivadis.com/OL5/ksplice/x86_64/
gpgcheck=1
enabled=1
```

Install the offline version of the Ksplice Uptrack framework:

```
# yum install uptrack-offline.noarch
```

For Oracle Enterprise Linux, install all available Ksplice patches by using the yum command as in the example below:

```
# yum install uptrack-updates-$(uname -r)
```

```
Installing:   uptrack-updates-2.6.39-400.17.1.el6uek.x86_64-20130808-
0.noarch
```

```
The following steps will be taken:
Install [ptj4l5wq] CVE-2013-0268: /dev/cpu/*/msr local privilege
                                  escalation.
…
Your kernel is fully up to date.
Effective kernel version is 2.6.39-400.109.5.el6uek
```

For Oracle Enterprise Linux 5 the yum command is slightly different:

```
# yum install uptrack-updates-$(uname –r).$(uname –m)
```

The Uptrack web interface, as well as the Uptrack API described in the previous chapter cannot be used with servers using the offline version of Ksplice Uptrack.

Ksplice Uptrack within an Oracle Database Environment

Applying Ksplice patches in an Oracle Database (cluster or a single server) environment does not have any impact on the running database instances or any Oracle processes. Ksplice does not touch any shared libraries being used by Oracle, therefore relinking the software or scheduling a downtime is not necessary. It is a big advantage; you can apply important critical kernel patches online, no downtime and no service disruption imposed.

Anyway, patch management in an Oracle database environment needs to consider the whole software stack; also non-kernel OS packages updated using traditional patches and the impact on the Oracle Grid Infrastructure and Oracle database processes.

Regarding operating system packages, most of them can be patched without any service disruption, a small fraction of them might however require a server reboot (e.g. some 3rd party kernel modules, hardware drivers).

Oracle database and ASM software have some strong dependencies to shared operating system libraries amongst others from the GLIBC and LIBAIO RPMs:

```
# ldd /u00/app/grid/product/11.2.0.3/bin/oracle
...
  /lib64/ld-linux-x86-64.so.2 (0x00007f4734c88000)
```

As an example, the GLIBC library ld-linux-x86-64.so.2 is mapped into the private memory of almost every Oracle process:

```
# fuser -uv /lib64/ld-linux-x86-64.so.2                          USER
PID             ACCESS            COMMAND/lib64/ld-linux-x86-64.so.2:
...                     oracle    27299 ....m (oracle)oracle
                     grid      27301 ....m (grid)oracle
                     grid      27624 ....m (grid)oraagent.bin
                     grid      27641 ....m (grid)evmd.bin
                     grid      27695 ....m (grid)evmlogger.bin
                     grid      27707 ....m (grid)cssdagent
                     grid      27733 ....m (grid)ocssd.bin
```

Although during our tests upgrading the operating system shared libraries used by Oracle, without relinking the Oracle software, did not lead to any problems in a single, as well as cluster database environment, Oracle recommendation is to relink the software (in a cluster environment it is not only suggested, but rather required).

To sum it up: Ksplice is a great and very powerful technology! But we should also set the right expectations. Ksplice itself cannot guarantee 100% service uptime itself, at least not in an Oracle database environment. You need to consider, that in some cases Oracle recommends relinking the software, and in some it is even required. We need to patch the Grid Infrastructure and the Database software itself – which will in most cases also lead to a service disruption on a particular server. In a cluster environment, with RAC, we can however patch in a rolling fashion, and thus not causing a real downtime for the clients. In a single instance database environment you will mostly have no choice; a scheduled downtime is a must.

If you need high database service availability, you have no choice – a cluster is a must. Ksplice might be used as an add-on complementary technology to apply important kernel patches at once – no planning, no immediate downtime.

And, 100 % uptime in a real IT world is anyway not possible!

References

Ksplice documentation: *http://docs.oracle.com/cd/E37670_01/E37355/html/ol_ksplice.html*
Ksplice: Automatic Rebootless Kernel Updates; Jeff Arnold and M. Frans Kaashoek; Massachusetts Institute of Technology: *http://www.ksplice.com/doc/ksplice.pdf*
Ksplice Website: *http://www.ksplice.com*
http://en.wikipedia.org/wiki/Ksplice

Contact Address:

Robert Bialek
Trivadis GmbH
Lehrer-Wirth-Strasse, 4
D-81829 Munich

Telefon:	+49 (0) 89-99 27 59 30
Fax:	+49 (0) 89-99 27 59 59
E-Mail	robert.bialek@trivadis.com
Internet:	www.trivadis.com