Oracle Linux Ksplice Hands-on LAB

This hands-on lab takes you through several steps on how-to provide zero downtime kernel updates to your Oracle Linux server thanks to Oracle Ksplice, the service and utility capable of introducing hot-patch capabilities for Kernel, Hypervisor and User-Space components like glibc and openssl.

The entire hands-on lab runs on an Oracle VM VirtualBox virtual machine based on Oracle Linux 7.4, it receives the Ksplice updates from a local repository. In the lab we do the following steps:

- Inspect the kernel and search for vulnerabilities
- Perform Local Denial of Service attack based on found vulnarability (CVE-14489)
- Apply Ksplice kernel patches as rebootless updates

The Ksplice client is available in online or offline mode, in this hands-on lab we use the offline Ksplice client. The offline version of the Ksplice client removes the requirement that a server on your intranet has a direct connection to the Oracle Ksplice server or to Unbreakable Linux Network (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.

Preparation

First, import the Virtual Machine template in VirtualBox on your laptop, use the preconfigured OVA template from the instructor. There are two versions:

```
oraclelinux-7.4-kspliceoffline           (CLI version)
oraclelinux-7.4-gui_kspliceoffline       (GUI version)
```

Depending on your preference install one of the VMs and when imported start the VM with a normal start. When the server is ready, login with the following credentials:

```
Username:    demo
Password:    demo
```

Inspect the Oracle Linux server

We can use the Ksplice Inspector to review the security patches available for the installed kernel on the server. This can be done online via the Ksplice website or via a CLI command connecting to the Ksplice API server.

We will use one of the found vulnarabilities (CVE-14489 used as a local Denial of Service) as an example to show how easy it is to attack a system.

Use the ksplice inspector with CLI and search for CVE 14489

```
# more ksplice-inspector.sh
# ./ksplice-inspector.sh
# ./ksplice-inspector.sh | grep 14489
```

Use the ksplice inspector with GUI and search for CVE 14489. Run the following command in your terminal.

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

Launch a browser and goto http://www.ksplice.com/inspector
Copy the output of the echo command into the text box and click Find Updates.
```

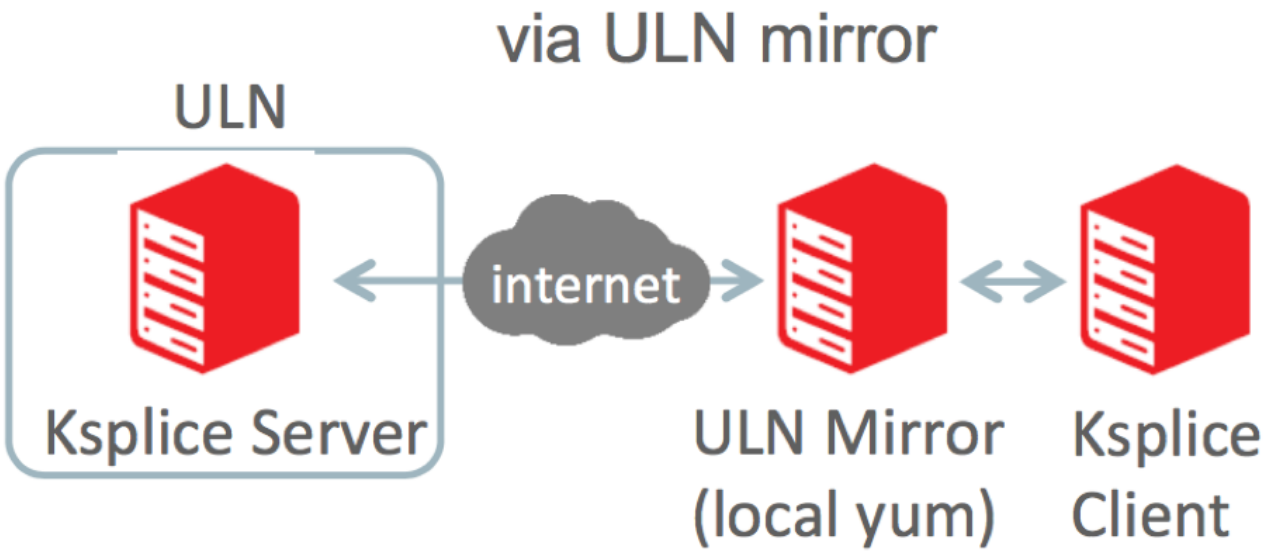
In the list with available Ksplice Updates you will find several CVEs including the one we like to use (CVE-14489). We found the code for this exploit, compiled it and made it available to you for this exercise.

```
# cd exploit
# ./cve14489
```

The Oracle Linux server will crash because of the local denial of service. The only thing we can do is a power reset and reboot the VM.

Install Ksplice in Offline Mode

The Ksplice Offline client eliminates the need having a server on your intranet with a direct connection to Oracle's online Ksplice service. In this lab we use a local yum server that has Oracle Linux packages, updates and Ksplice updates synchronized with Oracle ULN. This is a very common deployment model for Ksplice deployments.



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.

After the reboot of the Oracle Linux VM login as the root user.

```
Username:    root
Password:    demo
```

Verify you are able to connect to the local Ksplice repository with the Ksplice patches, make sure your server uses the correct IP-address for the yum server. Enter your IP-address (provided by the instructor) in the hosts file.

```
# vi /etc/hosts
xxx.xxx.xxx.xxx localrepo
# ping localrepo
```

Install the Ksplice Offline client package.

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

Clear the yum metadata cache.

```
# yum clean metadata
```

Verify the configuration and check available Oracle Linux repositories.

```
# yum repolist
should be “local_ol7_ksplice” and “ol7_latest/x86_64”
```

Verify the installed (running) kernel and the current effective kernel (they should be the same).

```
# uname –r
# uptrack–uname –r
```

Install the Ksplice updates that are available for the kernel in use.

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

Verify the current effective kernel again and compare with installed kernel version.

```
# uptrack–uname –r
# uname –r
```

Print the number of ksplice updates installed (also run without --count, it shows the installed updates).

```
# uptrack–show ––count
# uptrack–show
```

Verify if the Ksplice updates were effective, logon as demo/demo again (in a new terminal with `su – demo`) and run the CVE-14489 exploit and see if the VM crashes. It doesn't :-)

```
# cd exploit
# ./cve14489
Use Ctrl–C to exit the program.
```

Removing Ksplices and Ksplice Uptrack software

It's easy to remove a single Ksplice update or even all off the updates, again this happens without a reboot. It's also possible to bring the kernel to a specific effective update of the kernel, but that is left to an exercise for the attendee.

Remove a single ksplice update (notice it also removes depending updates) by specifying the ID or remove all ksplice updates.

```
# uptrack–remove
# uptrack–remove 51tkixls                (the ID is just an example)
# uptrack–remove –all
```

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

```
# yum –y remove uptrack–offline
```