KernelCare: Live Kernel Patching for Linux

TECHNICAL WHITE PAPER
What is KernelCare?
The kernel is the most important part of any Linux system. It provides vital low-level functions to the entire system. Any security issues detected within it jeopardize the whole server. KernelCare is patch management software that automatically keeps your Linux kernel up to date with the latest security patches.

No server rebooting or system downtime is necessary. It is fast, simple and easy to deploy, and can deliver complex patch configurations or customized kernels without affecting performance. It is available for all major Linux distributions.

CloudLinux Inc. created KernelCare, fulfilling a need for targeted, low-overhead, security patch maintenance for Linux servers.

Why is it needed?
Linux has a long history of solid dependability, but like most modern operating systems, it is a large body of complex software that needs frequent updates. These updates often target perceived security weaknesses, which, if not resolved, can be exploited to compromise or debilitate your servers and data.

For example, there were over 450 Linux kernel vulnerabilities detected last year¹, some of which are fixed by individual patches. It is not uncommon for a Linux system to need monthly updates and reboots.

There is a time lag between the detection of a vulnerability and its resolution by a patch update. This offers an unavoidable window of opportunity for malicious threat agents within which to target systems and exploit vulnerabilities.

However, once a patch is released, its effectiveness in preventing attack is severely curtailed if the patch is not immediately applied. This entirely avoidable situation is where KernelCare comes in. It virtually eliminates the gap between patch issue and patch application, by installing patches automatically and without disruption to your core services.

About KernelCare
Our team consists of expert kernel developers whose primary role is to watch for kernel vulnerabilities and prepare patches for them. These are released as soon as possible, often much sooner than most Enterprise Linux vendor releases. We can do this, quickly, because our sole focus is on kernel security, and none of its other functionalities—we do not touch any kernel ABIs (Application Binary Interfaces).

The traditional way of patching kernels can cause unwanted or undetected functional changes to your kernel. It may even introduce new or unknown security vulnerabilities. It can also change your kernel version, triggering security alerts or necessitating full regression testing of hosted applications.

All patch updates are fully auditable - all can be selectively pre-tested and approved for distribution and installation or abandoned and rolled back. This can be done at any time with zero impact.

¹ https://www.cvedetails.com/product/47/Linux-Linux-Kernel.html?vendor_id=33
KernelCare runs as a service that live-patches a running Linux kernel. A small agent installed on a server applies binary kernel patches. These are downloaded directly from our repository, the main KernelCare Patch Server at http://patches.kernelcare.com. This server can be accessed directly or through a firewall (via a proxy server), or a local patch update server can be self-hosted to deliver patches.

Patches are distributed as cumulative binary packages, custom-built for each supported kernel version, and each is GPG-key signed for security.

When a patch is applied with KernelCare, a reboot of the system is not required. This is not the case when using traditional update tools (e.g. yum, apt-get). Instead, the Linux kernel is binary patched, in memory. Nothing else is touched, so there is no need to update system libraries or packages to keep in step with kernel changes. In fact, the official patch level does not change (see Security Compliance).

**Patching Servers**

**Example 1: Direct Internet Access**

If your servers have access to the internet, even if via NAT, you can use the KernelCare Patch Server.

Using key-based licensing, you can quickly deploy KernelCare on your servers with these two commands.

curl -s https://repo.cloudlinux.com/kernelcare/kernelcare_install.sh | bash
/usr/bin/kcarectl --register KEY

**NOTE:** The word **KEY** is literal. Do not replace it with a license key string.
Example 2: Access Via Proxy

If your server has no direct internet access, a proxy server can be used. KernelCare uses these standard environment variables to configure the proxy.

```
http_proxy=http://proxy.domain.com:port
https_proxy=http://proxy.domain.com:port
```

KernelCare will use these variables to connect to the internet via the proxy. The command to run it is the same as before.

```
curl -s https://repo.cloudlinux.com/kernelcare/kernelcare_install.sh | bash
/usr/bin/kcarectl --register KEY
```

Example 3: No Internet Access (local ePortal)

Servers without an internet connection can still take advantage of the automated patch service of KernelCare.

KernelCare.ePortal is a patch server that runs internally, but outside of your firewall. It acts as a bridge between internal patch servers and the main KernelCare patch server.

This approach is ideal for staging and production environments which need strict isolation from external networks, or which requires stricter control over the patches to be applied.

You can use automated deployment to distribute the KernelCare agent to your servers.
Automated Deployment

Tools such as Ansible, Puppet, Chef, and others, can be used to automate the deployment of KernelCare. With these, you can:

- Distribute the KernelCare agent package (only necessary for servers with no internet access).
- Distribute the KernelCare agent configuration file `/etc/sysconfig/kcare/kcare.conf`.
- Set environment variables.
- Install the KernelCare agent (from either local or remote download servers).
- Register KernelCare with key-based or IP-based licenses.

For more details on automating KernelCare, see [http://docs.kernelcare.com/index.html?automation.htm](http://docs.kernelcare.com/index.html?automation.htm).

Custom Patch Feeds

KernelCare.ePortal lets you update different servers to different patch levels. With it you create custom patch feeds, each with their own patch combinations and configurations, and each with their own license keys.

For example, you might create patch feeds for groups of servers, for specialized environments (e.g. testing, staging, QA), or for production release auditing.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>Server Count / Limit</th>
<th>Feed</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>testkey</td>
<td>3 / unlimited</td>
<td>test feed</td>
</tr>
<tr>
<td>key23we</td>
<td>testkey</td>
<td>1 / 1234</td>
<td>test feed</td>
</tr>
<tr>
<td>TLYS5978GF2wE26p</td>
<td>testkey</td>
<td>0 / 3</td>
<td>production</td>
</tr>
</tbody>
</table>

Examples of Servers in the ePortal GUI
Patch Servers and the CloudLinux Network

The CLN (CloudLinux Network) is where CloudLinux Inc. product licenses (including KernelCare) are managed. Each license can be given a *sticky tag*. This tag is the date at which licensed environments must be patched, given in **DDMMYY** format. Tagged servers will receive all patches released on or before the specified date.

To set a sticky tag:

1. Log into the CLN portal.
2. Open the *Edit Key info* dialogue by navigating to *KernelCare Keys → Edit Key Info*.

3. Fill out the *Sticky tag* field.

4. On the server to be patched, run:
   
   ```bash
   /usr/bin/kcarectl --set-sticky-patch=KEY
   ```

   Alternatively, edit the file `/etc/sysconfig/kcare/kcare.conf` and add:

   ```
   STICKY_PATCH=KEY
   ```

   **NOTE:** The word **KEY** is literal. Do not replace it with a license key string.
Disabling Auto-Update

You can disable the automatic update of environments by editing the file /etc/sysconfig/kcare/kcare.conf and setting the variable as shown below.

AUTO_UPDATE=False

The server will no longer get automatic patch updates. You must manually, or via automation tools, invoke the update with this command.

/usr/bin/kcarectl --update

Test and Delayed Patch Feeds

As well as the standard (i.e. production) patch feed, the KernelCare patch server provides:

- Test feed - the latest patches that have not completed all tests.
- Delayed feeds - patches released within the past 12, 24 or 48 hours. These can be skipped and will not be loaded.

Such feeds are configured in the file /etc/sysconfig/kcare/kcare.conf, by assigning one of these values to the PREFIX variable.

- test (for the test feed)
- 12h (for the 12 hour delayed feed)
- 24h (for the 24 hour delayed feed)
- 48h (for the 48 hour delayed feed)

MONITORING

Systems protected by KernelCare can be monitored with built-in methods, or by using the REST API together with third-party tools, such as Nagios or Zabbix.
Monitoring via the CLN

In the example below, registered KernelCare installations are grouped by license keys. Those in red do not have the latest patches installed.

![Image of KernelCare installation status]

Monitoring via the KernelCare.ePortal Admin Page

If you are using a KernelCare.ePortal server, the administration page (http://ePortal IP/admin) can be used to filter on key ID.

![Image of KernelCare.ePortal admin page]

Monitoring on the Command Line

You can check whether the latest patch has been applied with this command.

```
/usr/bin/kcarectl --check
```
Monitoring with the KernelCare API

KernelCare has a REST API that can be used to extract status information for monitoring purposes. The syntax is as follows.

- For key-based licenses:
  https://cln.cloudlinux.com/api/kcare/nagios/key_id
- For IP-based licenses (resellers):
  https://cln.cloudlinux.com/api/kcare/nagios-res/login/token
- For ePortal patch distribution:
  http://ePortal IP/admin/api/kcare/nagios/key_id


Nagios/Zabbix Integration

Enterprise users of Nagios or Zabbix can use the script at http://patches.kernelcare.com/downloads/nagios/check_kcare.

This script is a command-line utility that produces output compatible with the above two vendor tools. It classifies patches as one of:

- Up to date
- Out of date
- Unsupported
- Inactive

The script only reports servers with a KernelCare key (registered at CLN or KernelCare.ePortal) and all servers within partner accounts (registered at CLN).

An example of the Service Status view using the KernelCare status checker script in Nagios is shown below.
To use the `check_kcare` script:

2. Copy it to:
   - `/usr/lib64/nagios/plugins/` (for Nagios)
   - `/usr/lib/zabbix/externalscripts/` (for Zabbix)
3. Make it executable.


---

**SECURITY COMPLIANCE**

Because KernelCare patches the kernel directly in memory, the official patch identification does not change. In other words, neither the output of `uname -r` nor the contents of the file `/proc/version` change when patched.

We do this because `glibc` and other libraries relying on the kernel ABI (Application Binary Interface) must know the exact version of the kernel.

Although this approach provides the highest levels of stability and compatibility for servers, it can cause some security scanners to report the active kernel as 'out of date'.

To prevent such reports, KernelCare has a command that returns the effective version of the kernel.

```
kcure-uname -r
```

Other scripts are available to adjust reports from Rapid7™ Nexpose.

**Using KernelCare with Rapid7™ Nexpose**

KernelCare can inform Rapid7™ Nexpose that the kernel is live-patched (i.e. that the effective and booted kernel versions differ). This is done by adding vulnerability exceptions using the `kcare-nexpose` script.

**Note:** *It reports in XML V2 format and only supports KernelCare installations with key-based licenses.*

The script connects to the Rapid7™ Nexpose instance, finds KernelCare-patched security reports and adds exceptions for all binary patched CVEs (Common Vulnerabilities and Exposures).

The list of CVEs is retrieved from the KernelCare Patch Server or the KernelCare.ePortal server, depending on the type of installation.

The report is automatically rerun to reflect any changes in the CVE list. It can automatically approve CVEs and remove outdated ones added in previous kernel versions.
Installing and Running the Script

**NOTE:** These instructions are for an EL6-based system.

1. Set up the repository location.
   ```
   cat > /etc/yum.repos.d/kcare-eportal.repo <<EOL
   [kcare-eportal]
   name=KernelCare ePortal
   baseurl=http://repo.eportal.kernelcare.com/x86_64/
   gpgkey=http://repo.cloudlinux.com/kernelcare-debian/6/conf/kcaredsa_pub.gpg
   enabled=1
   gpgcheck=1
   EOL
   ```
2. Install.
   ```
   yum install kcare-nexpose
   ```
4. Generate a report (example below) to use as the basis for adding exceptions.
5. Run the script to remove old CVEs and add new ones.
   kcare-nexpose -c /usr/local/etc/kcare-nexpose.yml

6. Vulnerability exceptions are added independently for each asset, as shown below.
This Technical White Paper covered the key points in installing and configuring KernelCare. It also mentioned the key requirements for Linux kernel patch management:

- Automatic installation of patches
- Custom patch configurations and manual overrides
- Choice of patch repositories
- Integration with automation and monitoring utilities

<table>
<thead>
<tr>
<th></th>
<th>With KernelCare.ePortal</th>
<th>Without KernelCare.ePortal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Server Location</td>
<td>On premises</td>
<td><a href="http://patches.kernelcare.com">http://patches.kernelcare.com</a></td>
</tr>
<tr>
<td>License Server Location</td>
<td>On premises</td>
<td><a href="https://cln.cloudlinux.com">https://cln.cloudlinux.com</a></td>
</tr>
<tr>
<td>Installation Instructions</td>
<td>ePortal Server</td>
<td>KC agent</td>
</tr>
<tr>
<td>Costs</td>
<td>Per license, ePortal included at no additional cost</td>
<td>Per license</td>
</tr>
<tr>
<td>Patch Rollout flexibility</td>
<td>ePortal Feeds</td>
<td>Sticky patches</td>
</tr>
<tr>
<td>Multiple Environments</td>
<td>Yes (via ePortal feeds)</td>
<td>Yes (via Sticky patches)</td>
</tr>
<tr>
<td>Monitoring</td>
<td>ePortal, REST API (Nagios, Zabbix)</td>
<td>CLN, REST API</td>
</tr>
</tbody>
</table>

**More Information**

- KernelCare website: https://www.kernelcare.com
- KernelCare Blog: https://www.kernelcare.com/blog
- KernelCare Patch Server: http://patches.kernelcare.com
- KernelCare documentation: http://docs.kernelcare.com
- CloudLinux Network - CLN (Billing Portal): https://cln.cloudlinux.com
- CloudLinux 24/7 online support system: https://cloudlinux.zendesk.com