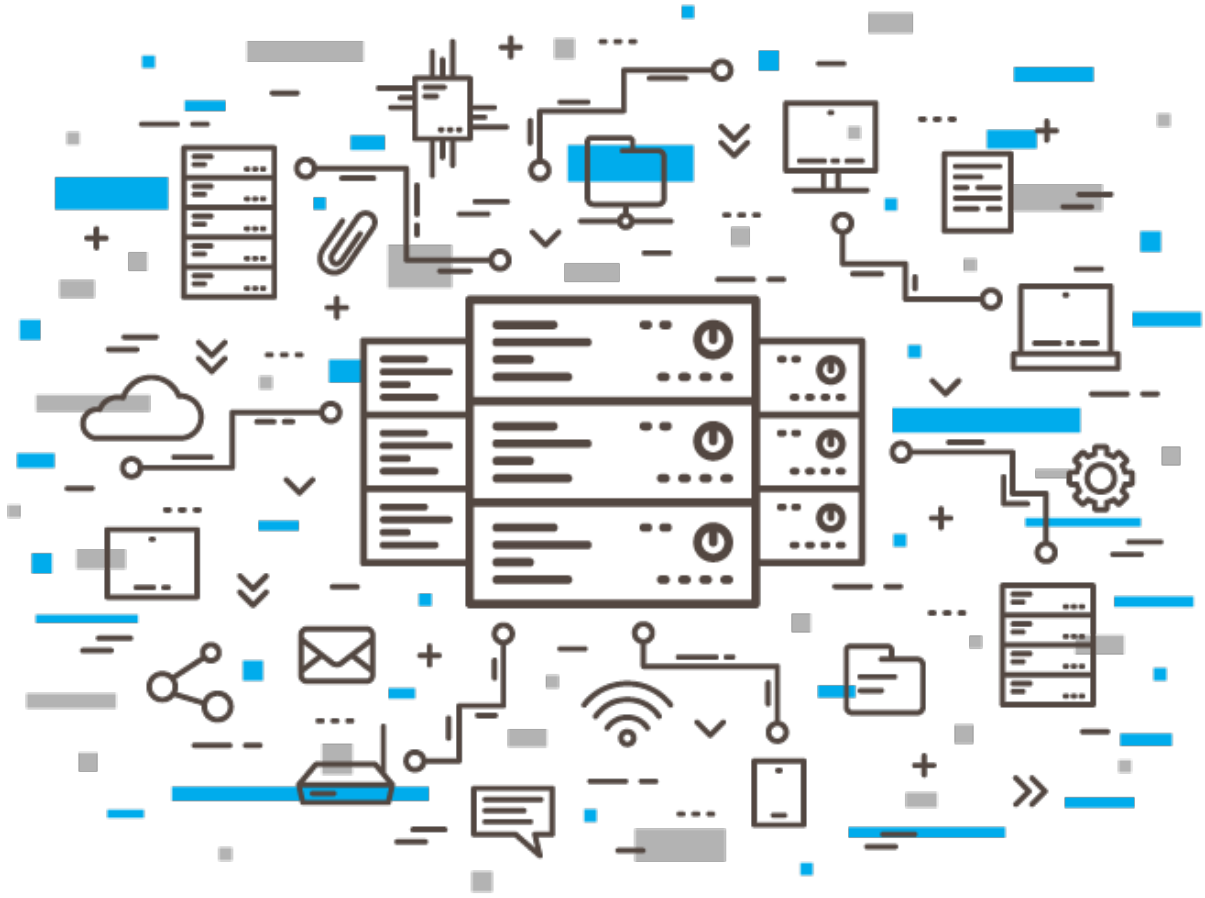
KernelCare We are CloudLinux

KernelCare: Live Kernel Patching for Linux

TECHNICAL WHITE PAPER



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, which in turn puts your customers and your revenue stream at risk.

EXECUTIVE SUMMARY

What is KernelCare?

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 year[1], 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

HOW IT WORKS

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 SCENARIOS

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.

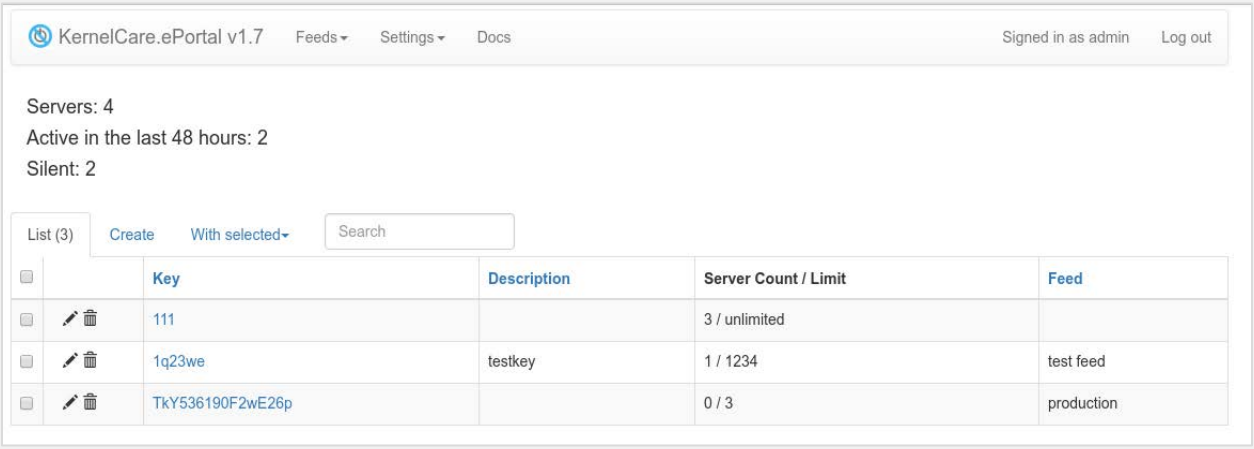
SPECIALIZED PATCH CONFIGURATIONS

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.



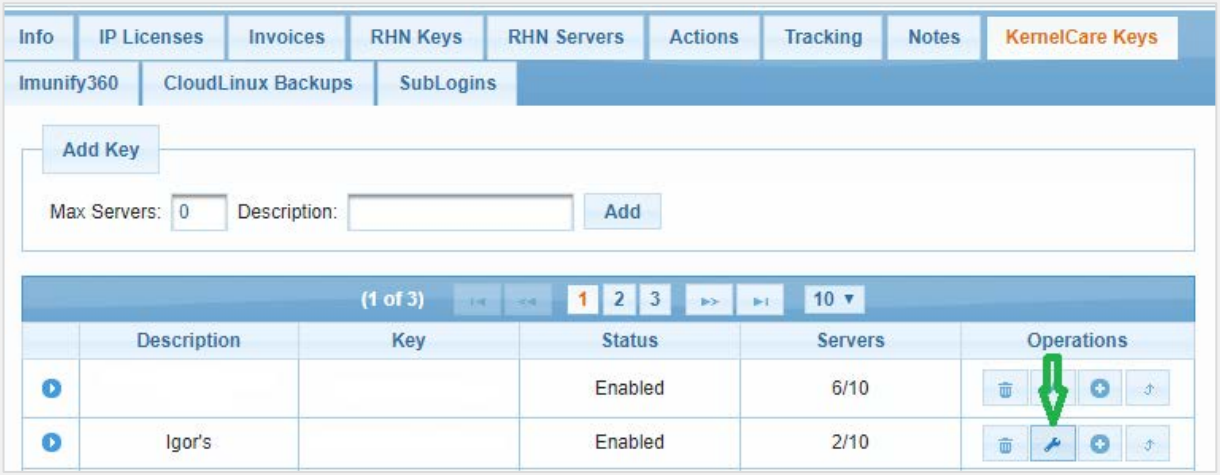
Examples of Servers in the ePortal GUI

Patch Servers and the CloudLinux Network

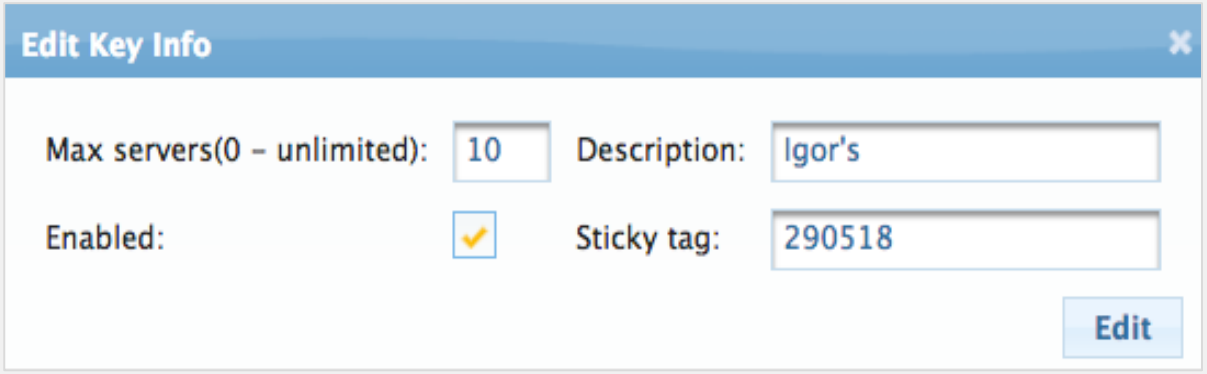
The CLN (CloudLinux Network) is where CloudLinux Inc. product licenses (including KernelCare) are managed. Each license can be given a *sticky tag*. This tag is the date at which licensed environments must be patched, given in `DDMMYY` format. Tagged servers will receive all patches released on or before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

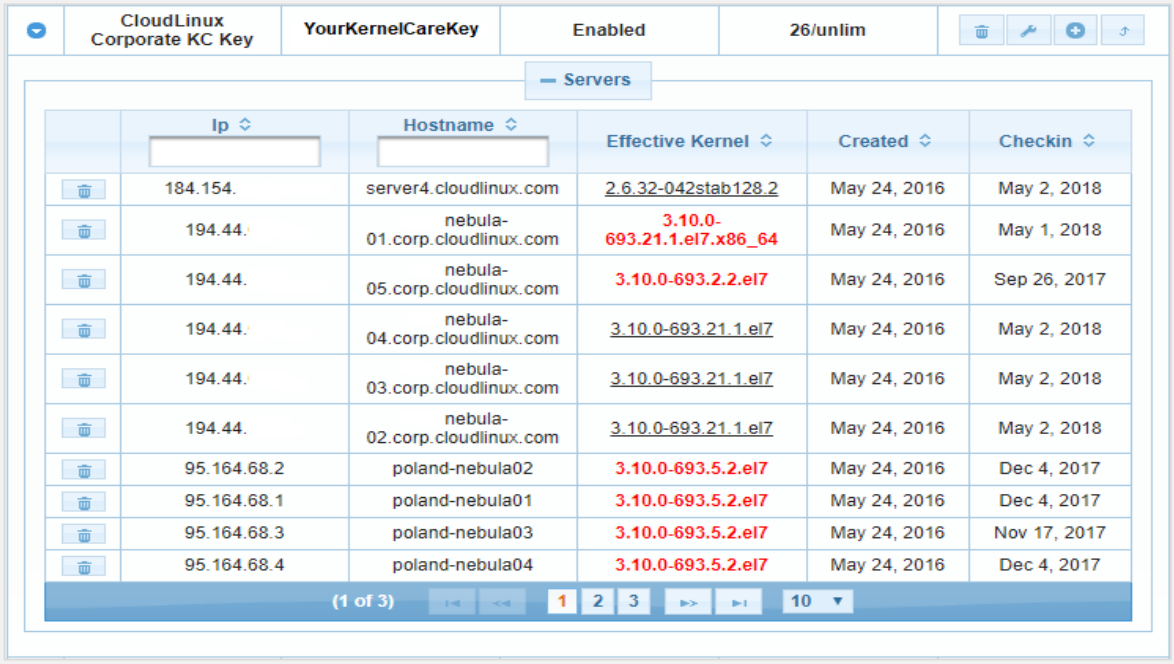
- `test` (for the test feed)
- `12h` (for the 12 hour delayed feed)
- `24h` (for the 24 hour delayed feed)
- `48h` (for the 48 hour delayed feed)

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.



CloudLinux Corporate KC Key | YourKernelCareKey | Enabled | 26/unlim

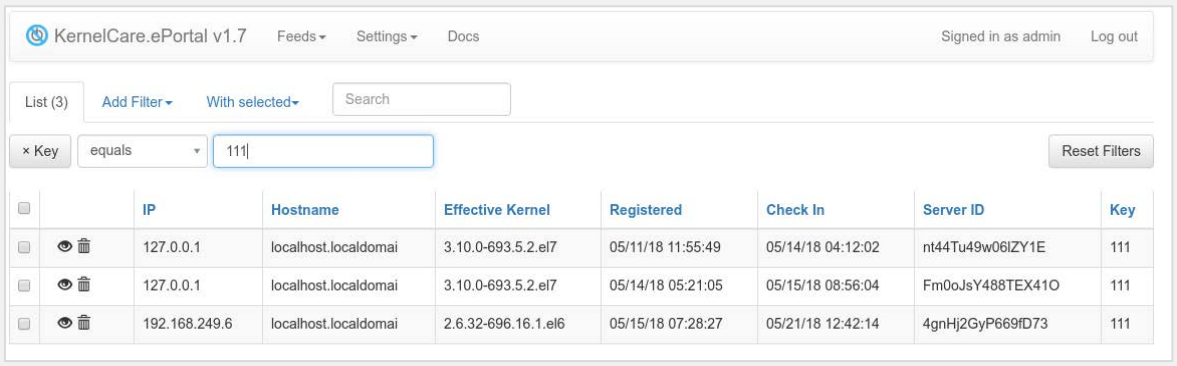
— Servers

Ip	Hostname	Effective Kernel	Created	Checkin
184.154.	server4.cloudlinux.com	2.6.32-042stab128.2	May 24, 2016	May 2, 2018
194.44.	nebula-01.corp.cloudlinux.com	3.10.0-693.21.1.el7.x86_64	May 24, 2016	May 1, 2018
194.44.	nebula-05.corp.cloudlinux.com	3.10.0-693.2.2.el7	May 24, 2016	Sep 26, 2017
194.44.	nebula-04.corp.cloudlinux.com	3.10.0-693.21.1.el7	May 24, 2016	May 2, 2018
194.44.	nebula-03.corp.cloudlinux.com	3.10.0-693.21.1.el7	May 24, 2016	May 2, 2018
194.44.	nebula-02.corp.cloudlinux.com	3.10.0-693.21.1.el7	May 24, 2016	May 2, 2018
95.164.68.2	poland-nebula02	3.10.0-693.5.2.el7	May 24, 2016	Dec 4, 2017
95.164.68.1	poland-nebula01	3.10.0-693.5.2.el7	May 24, 2016	Dec 4, 2017
95.164.68.3	poland-nebula03	3.10.0-693.5.2.el7	May 24, 2016	Nov 17, 2017
95.164.68.4	poland-nebula04	3.10.0-693.5.2.el7	May 24, 2016	Dec 4, 2017

(1 of 3) 1 2 3 10

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.



KernelCare.ePortal v1.7 Feeds Settings Docs Signed in as admin Log out

List (3) Add Filter With selected Search

× Key equals 111 Reset Filters

IP	Hostname	Effective Kernel	Registered	Check In	Server ID	Key
127.0.0.1	localhost.localdomai	3.10.0-693.5.2.el7	05/11/18 11:55:49	05/14/18 04:12:02	nt44Tu49w06lZY1E	111
127.0.0.1	localhost.localdomai	3.10.0-693.5.2.el7	05/14/18 05:21:05	05/15/18 08:56:04	Fm0oJsY488TEX41O	111
192.168.249.6	localhost.localdomai	2.6.32-696.16.1.el6	05/15/18 07:28:27	05/21/18 12:42:14	4gnHj2GyP669fD73	111

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`

A description of the CloudLinux REST API is at https://cln.cloudlinux.com/clweb/downloads/cloudlinux-rest-api.pdf.

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.

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
kcare-service	KernelCare Server Status Checker By Key	OK	03-26-2018 18:32:16	0d 1h 18m 10s	1/4	All 1 servers are up to date
localhost	Current Load	OK	03-26-2018 18:32:51	0d 5h 16m 3s	1/4	OK - load average: 0.01, 0.06, 0.06
	Current Users	OK	03-26-2018 18:34:02	0d 5h 15m 25s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	03-26-2018 18:30:38	0d 5h 14m 48s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.002 second response time
	PING	OK	03-26-2018 18:32:28	0d 5h 14m 8s	1/4	PING OK - Packet loss = 0%, RTA = 0.07 ms
	Root Partition	OK	03-26-2018 18:30:35	0d 5h 13m 33s	1/4	DISK OK - free space: / 148512 MB (96.69% inode=100%):
	SSH	OK	03-26-2018 18:31:33	0d 5h 12m 55s	1/4	SSH OK - OpenSSH_7.4 (protocol 2.0)
	Swap Usage	CRITICAL	03-26-2018 18:34:30	4d 2h 12m 18s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	03-26-2018 18:33:14	0d 5h 11m 39s	1/4	PROCS OK: 44 processes with STATE = RSZDT

To use the `check_kcare` script:

1. Download it from http://patches.kernelcare.com/downloads/nagios/check_kcare
2. Copy it to:
   - `/usr/lib64/nagios/plugins/` (for Nagios)
   - `/usr/lib/zabbix/externalscripts/` (for Zabbix)
3. Make it executable.

NOTE: *A template for Zabbix is at* http://patches.kernelcare.com/downloads/nagios/kcare_zabbix_template.xml*.*

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.

`kcare-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
```

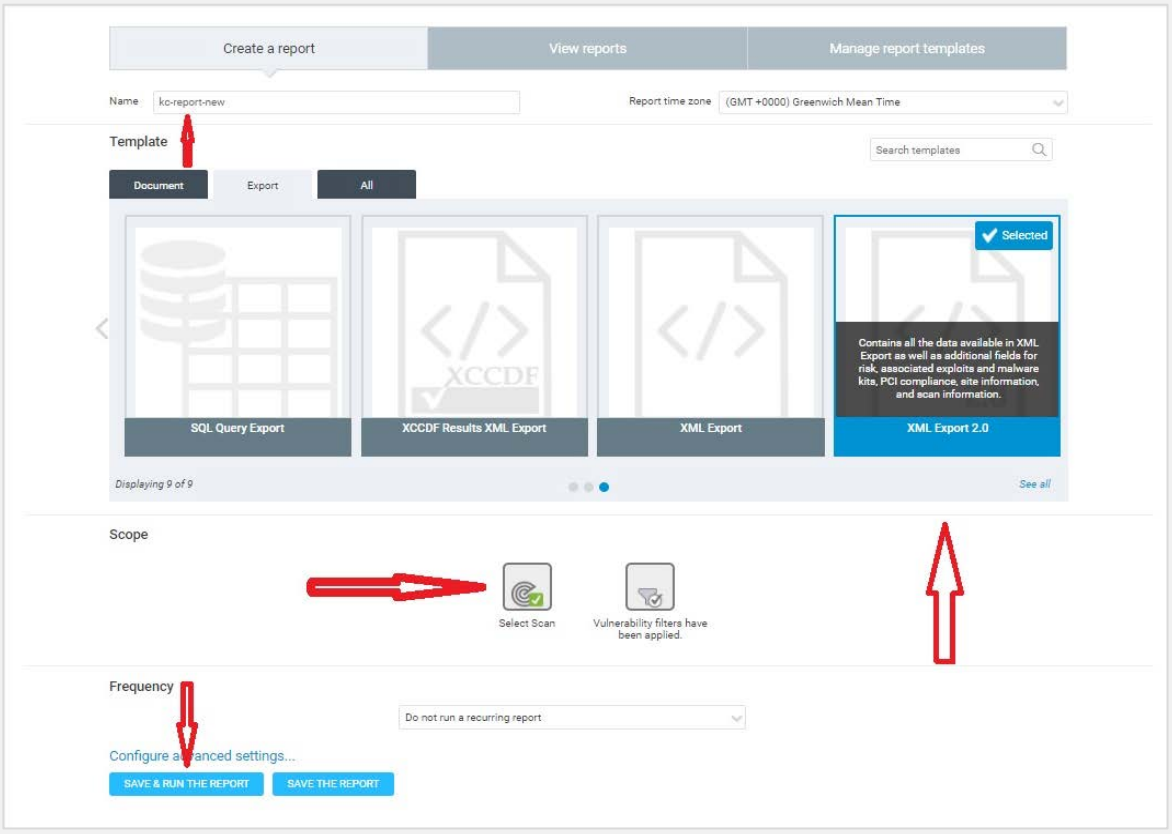
2. Install.

```
yum install kcare-nexpose
```

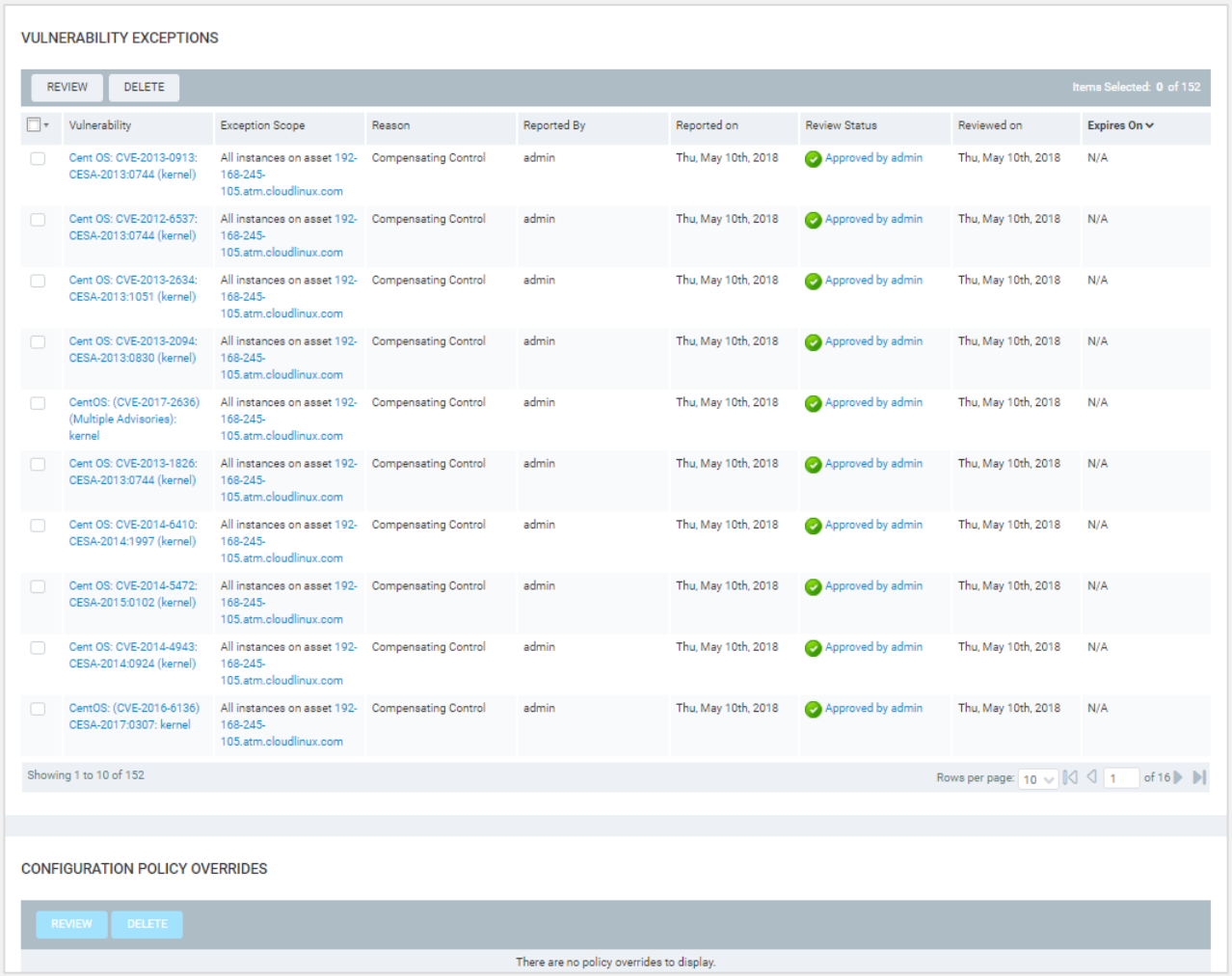
3. Create a configuration file in `/usr/local/etc/kcare-nexpose.yml`.

   An annotated sample of one is at http://docs.kernelcare.com/yaml_config_file_description.htm.

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.



VULNERABILITY EXCEPTIONS

REVIEW DELETE — Items Selected: 0 of 152

Vulnerability	Exception Scope	Reason	Reported By	Reported on	Review Status	Reviewed on	Expires On
Cent OS: CVE-2013-0913: CESA-2013:0744 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2012-6537: CESA-2013:0744 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2013-2634: CESA-2013:1051 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2013-2094: CESA-2013:0830 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
CentOS: (CVE-2017-2636) (Multiple Advisories): kernel	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2013-1826: CESA-2013:0744 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2014-6410: CESA-2014:1997 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2014-5472: CESA-2015:0102 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
Cent OS: CVE-2014-4943: CESA-2014:0924 (kernel)	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A
CentOS: (CVE-2016-6136) CESA-2017:0307: kernel	All instances on asset 192-168-245-105.atm.cloudlinux.com	Compensating Control	admin	Thu, May 10th, 2018	Approved by admin	Thu, May 10th, 2018	N/A

Showing 1 to 10 of 152 — Rows per page: 10 — 1 of 16

CONFIGURATION POLICY OVERRIDES

REVIEW DELETE

There are no policy overrides to display.

CONCLUSION

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

	With KernelCare.ePortal	Without KernalCare.ePortal
Update Server Location	On premises	http://patches.kernelcare.com
License Server Location	On premises	https://cln.cloudlinux.com
Installation Instructions	ePortal Server	KC agent
Costs	Per license, ePortal included at no additional cost	Per license
Patch Rollout flexibility	ePortal Feeds	Sticky patches
Multiple Environments	Yes (via ePortal feeds)	Yes (via Sticky patches)
Monitoring	ePortal, REST API (Nagios, Zabbix)	CLN, REST API

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