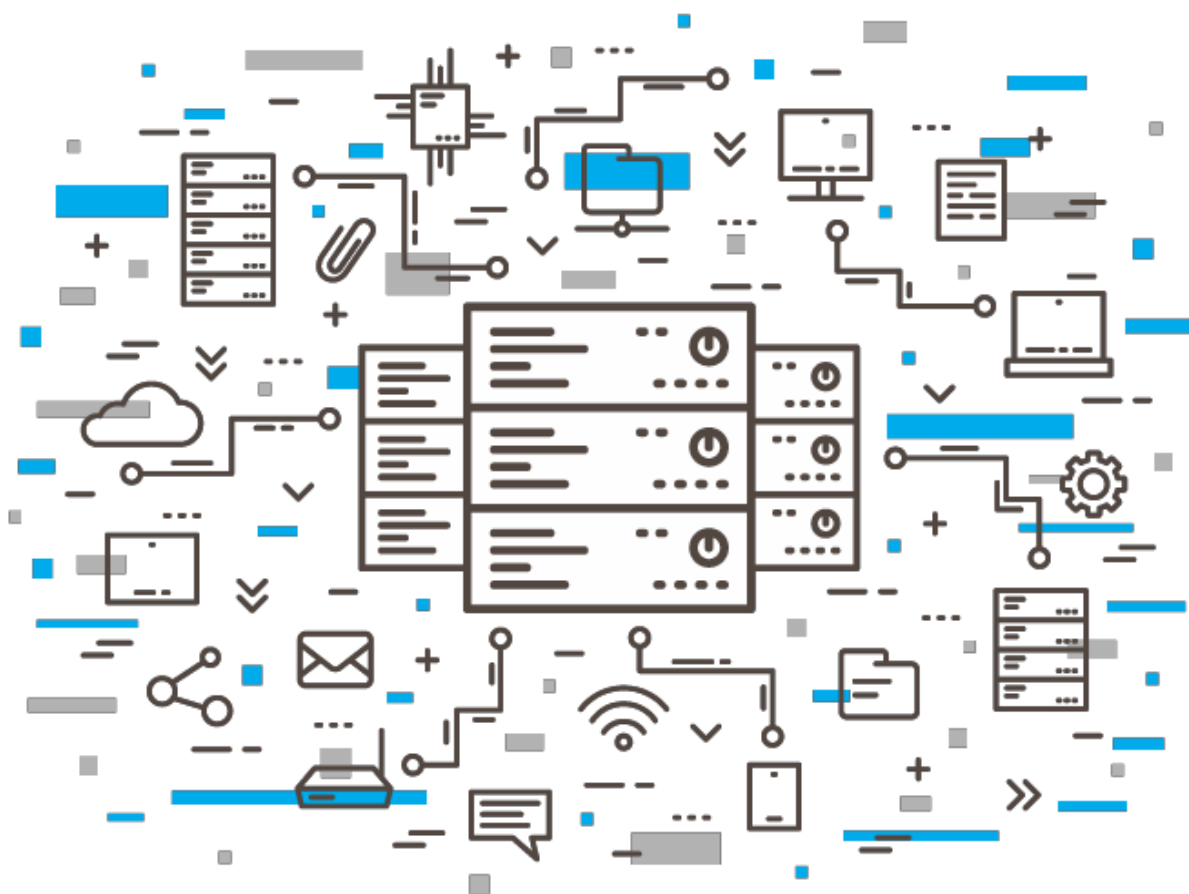


KernelCare: Linux用の ライブカーネルパッチ

テクニカルホワイトペーパー



KernelCareに関するお問合せ:

日本国内正規代理店

GDEPソリューションズ株式会社

TEL: 03-5802-7050 / E-mail: kcsales@gdep-sol.co.jp

URL: <http://www.gdep-sol.co.jp>

KernelCareとは？

カーネルはあらゆるLinuxシステムの最も重要な部分です。システム全体にとって不可欠な基本レベルの機能を提供しています。その中で検出されたセキュリティ上の問題点はサーバ全体を危険にさらします。KernelCareは最新のセキュリティパッチを適用し、Linuxカーネルを自動的に最新の状態に保つパッチマネジメントソフトウェアです。

サーバの再起動やシステムのダウンタイムは必要ありません。速く、簡単、そして展開するのが簡単となっており、パフォーマンスに影響を与えることなく複雑なパッチ構成またはカスタマイズされたカーネルを提供することができます。また、すべての主要なLinuxディストリビューションにおいて使用可能となっています。

CloudLinux Inc.社がKernelCareを作りました。Linuxサーバ用のターゲットを絞った、低オーバーヘッドのセキュリティパッチメンテナンスにおいて必要な条件を満たしています。

なぜそれが必要なのか？

Linuxには確かな信頼性を裏付ける長い歴史がありますが、最近のほとんどのオペレーティングシステムと同様に頻繁にアップデートが必要となっている大規模な複雑なソフトウェアです。これらの**アップデートは多くの場合、認識されたセキュリティ上の弱点を対象**としています。解決できないとサーバやデータを弱体化させ、不正アクセスに利用される可能性があります。

例えば、昨年検出されたLinuxカーネルの脆弱性は450を超えており、そのうちのいくつかは個別対応のパッチで修正されています。Linuxシステムで毎月更新と再起動が必要になることは珍しくないのです。

脆弱性の検出とパッチ更新によるその解決方法の間にはタイムラグがあります。これにより悪意のある脅威エージェントがシステムを標的にして、脆弱性を悪用されることが避けられない可能性があります。

ただし、いったんパッチがリリースされれば、そのパッチがすぐに適用されない場合、攻撃を防ぐためのその効果は大幅に低下します。このような状況こそ完全に回避可能な状況を作り出すKernelCareの出番です。パッチを自動的にインストールし、コアサービスを中断することなく、パッチの問題とそれに対するパッチの適用の間のギャップを実質的に排除できるのです。

KernelCareについて

私たちのチームはカーネルの脆弱性を監視し、それらのためにパッチを準備することを主要な役割とするカーネル開発者から構成されています。これらパッチは可能な限り早くリリースされます。脆弱性の種類により、NVDやLinuxベンダーから私たちに、早いリリースの禁止要請がある場合を除き、多くの場合においては、ほとんどのEnterprise Linuxベンダーのリリースよりもはるかに早くリリースされています。これが可能なのは、私たちは**カーネルセキュリティのみに焦点を当てており**、カーネルABI（アプリケーション・バイナリー・インタフェース）など他の機能に対して、その焦点を当てていないためです。

カーネルにパッチを適用する従来の方法は、カーネルにとって望ましくないまたは検出されていない機能に変更を引き起こしてしまう可能性があります。それは新しいまたは未知のセキュリティ脆弱性を導入してしまうかもしれません。またカーネルのバージョンを変更したり、セキュリティ警告を表示したり、ホストアプリケーションの前のバージョンになかった不具合の完全なテストを必要とする可能性もあります。

すべてのパッチアップデートは完全に監査可能です。配布とインストールのためにすべてが選択的にプレテストされ承認されるか、もしくは放棄されロールバックされることができます。これは**影響なしにいつでも行うことができます**。

エグゼクティブ・サマリー

¹ https://www.cvedetails.com/product/47/Linux-Linux-Kernel.html?vendor_id=33

どのように機能するか、その仕組み

KernelCareは、稼働中のLinuxカーネルにライブパッチを適用するサービスとして機能しています。サーバにインストールされた小さなエージェントがバイナリカーネルパッチを適用します。これは <http://patches.kernelcare.com> にあるメインのKernelCare Patch Serverであるリポジトリから直接ダウンロードされます。このサーバには直接アクセスすることも、ファイアウォール経由で（プロキシサーバ経由で）アクセスすることも、そしてローカルのパッチ更新サーバを、パッチを配信するためにセルフホストすることもできます。

パッチは累積バイナリパッケージとして配布され、サポートされている各カーネルバージョン用にカスタムビルドされ、そしてそれぞれがセキュリティのために署名されたGPGキーとなっています。

KernelCareでパッチが適用される場合、**システムの再起動は必要ありません。これは従来のアップデートツール（例: yum, apt-get）にはなかった特徴です。**その代わりにLinuxカーネルはメモリ内でバイナリパッチされています。その他に何も変更や追加をしていないので、カーネルの変更に合わせてシステムライブラリやパッケージを更新する必要はありません。実際、公式のパッチレベルは変わらないです（セキュリティコンプライアンスを参照してください）。

サーバへのパッチ適用

事例1: インターネットへ直接アクセス

NAT経由であっても、サーバがインターネットにアクセスできる場合、KernelCare Patch Serverを使用することができます。

事例シナリオ



キーベースのライセンスを使用すると、これら2つのコマンドを使用して、サーバにKernelCareをすばやく展開することができます。

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

注記: KEY という単語は、そのままのアルファベットで使ってください。ライセンスキー文字列に置き換えないでください。

事例2: プロキシ経由のアクセス

サーバが直接インターネットにアクセスできない場合、プロキシサーバを使用できます。KernelCareはこれらの標準的な環境変数を使ってプロキシを設定します。

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



KernelCareはこれらの変数を使ってプロキシ経由でインターネットに接続します。実行するコマンドは既に説明したものと同じです。

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

事例3: インターネットにアクセスできない(ローカルePortal)

インターネットに接続していないサーバでもKernelCareの自動パッチサービスの利点を活用できます。



KernelCare.ePortalはファイアウォールの外側において、内部で動作するパッチサーバです。これは内部のパッチサーバとメインのKernelCareパッチサーバの間のブリッジとして機能します。

このアプローチは外部ネットワークから厳密に分離する必要があるステージング環境や本番環境において理想的であり、また適用するパッチを厳密にコントロールする必要がある際に最適です。

何らかの自動展開ツールを使用し、KernelCareエージェントをサーバに配布することができます。

自動展開

Ansible、Puppet、Chefなどのツールを使用して、KernelCareの展開を自動化することができます。これにより、以下のことが可能になります。

- KernelCareエージェントパッケージの配布（インターネットにアクセスできないサーバにのみ必要）
- KernelCareエージェント設定ファイルの配布（/etc/sysconfig/kcare/kcare.conf）
- 環境変数の設定
- KernelCareエージェントのインストール（ローカルまたはリモートのダウンロードサーバから）
- キーベースまたはIPベースのライセンスでKernelCareの登録

KernelCareの自動化の詳細については <http://docs.kernelcare.com/index.html?automation.htm> を参照ください。

特殊なパッチ構成

カスタムパッチフィード

KernelCare.ePortalを使用すると、様々なサーバを、様々なパッチレベルに更新することができます。

それを使用し、それぞれ独自のパッチの組み合わせと構成を持ち、それぞれ独自のライセンスキーを持つカスタムパッチフィードを作成できます。

例えばサーバのグループ、特殊な環境（例えばテスト、ステージング、QA、等）、または運用リリースの監査用のパッチフィードを作成できます。

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

Servers: 4
Active in the last 48 hours: 2
Silent: 2

List (3)
Create
With selected
Search

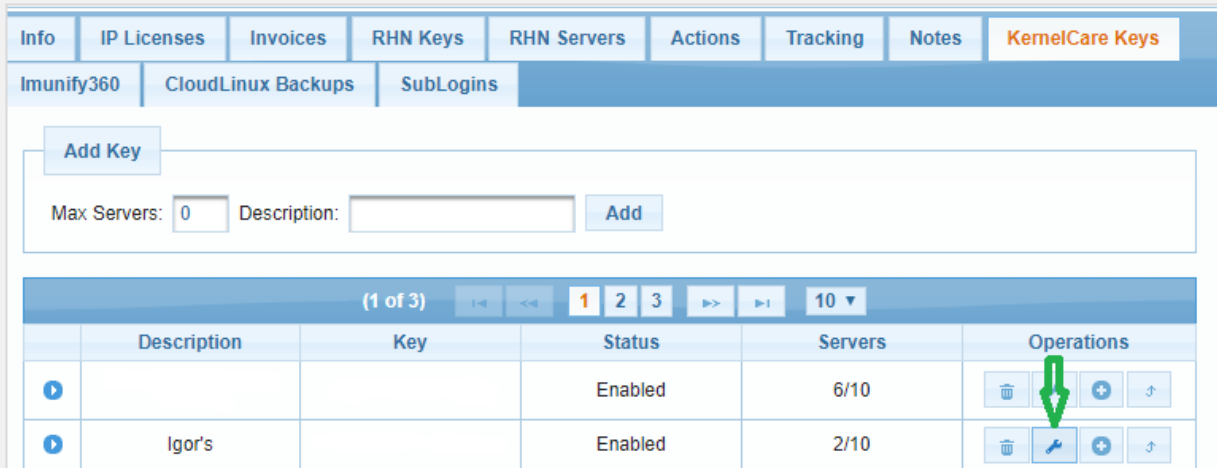
	Key	Description	Server Count / Limit	Feed
<input type="checkbox"/>	111		3 / unlimited	
<input type="checkbox"/>	1q23we	testkey	1 / 1234	test feed
<input type="checkbox"/>	TkY536190F2wE26p		0 / 3	production

パッチサーバとCloudLinux Network(CLN)

CLN(CloudLinux Network)はCloudLinux Inc.社の製品ライセンス(KernelCareを含む)が管理されている場所です。各ライセンスには sticky tag(スティッキータグ) を付けることができます。このタグはライセンスされた環境にパッチを適用しなければならない日付であり、DDMMYY形式で示されます。タグのついたサーバは指定された日付以前にリリースされたすべてのパッチを受け取ります。

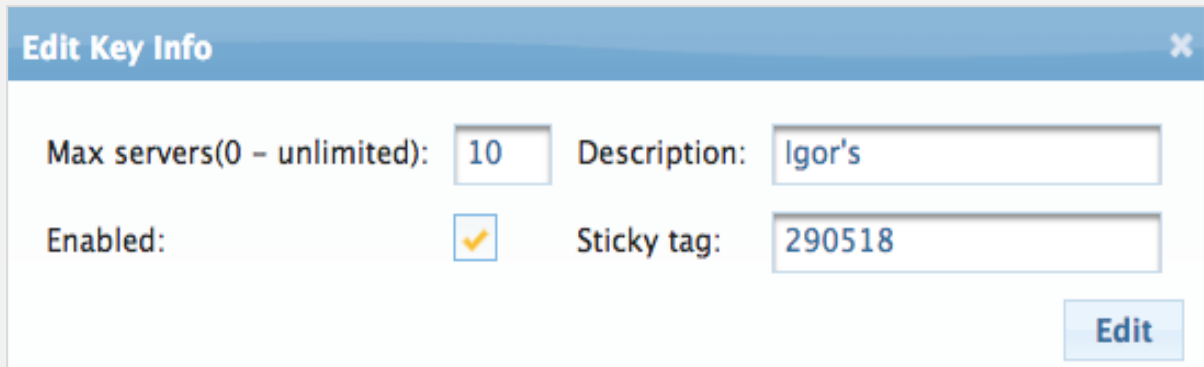
Sticky tagを設定するには以下を行います。

1. CLNポータルへログイン
2. KernelCare Keys → Edit Key Infoへ進み、Edit Keyダイアログ情報を開きます。



KernelCare Keys																							
Info	IP Licenses	Invoices	RHN Keys	RHN Servers	Actions																		
Imunify360	CloudLinux Backups	SubLogins																					
<div> Add Key </div> <div> Max Servers: <input type="text" value="0"/> Description: <input type="text"/> <input type="button" value="Add"/> </div>																							
<div> (1 of 3) 1 2 3 10 </div> <table> <thead> <tr> <th></th> <th>Description</th> <th>Key</th> <th>Status</th> <th>Servers</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td></td> <td></td> <td>Enabled</td> <td>6/10</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/> </td> </tr> <tr> <td><input type="radio"/></td> <td>Igor's</td> <td></td> <td>Enabled</td> <td>2/10</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/> </td> </tr> </tbody> </table>							Description	Key	Status	Servers	Operations	<input type="radio"/>			Enabled	6/10	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/>	<input type="radio"/>	Igor's		Enabled	2/10	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/>
	Description	Key	Status	Servers	Operations																		
<input type="radio"/>			Enabled	6/10	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/>																		
<input type="radio"/>	Igor's		Enabled	2/10	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Add"/> <input type="button" value="Refresh"/>																		

3. Sticky tagフィールドに記入します。



Edit Key Info

Max servers(0 - unlimited):
Description:

Enabled: ☒
Sticky tag:

Edit

4. パッチを適用するサーバにおいて、次のコマンドを実行します。
`/usr/bin/kcarectl --set-sticky-patch=KEY`
あるいは代わりに、ファイル/etc/sysconfig/kcare/kcare.conf を編集し、以下を追加します。
STICKY_PATCH=KEY

注記: KEY という単語はそのままのアルファベットで使ってください。ライセンスキー文字列に置き換えないでください

自動更新を無効にする

ファイルを編集して環境の自動更新を無効にすることができます。
/etc/sysconfig/kcare/kcare.conf を編集し、以下のように変数を設定します。

AUTO_UPDATE=False

そうするとサーバは自動でパッチ更新を行わなくなります。手動または自動化ツールを介して、以下のコマンドでアップデートを起動する必要があります。

/usr/bin/kcarectl --update

テストと遅延パッチフィード

標準の(つまり本番用の)パッチフィードと同様にKernelCareパッチサーバは以下のものを提供します。

- テストフィード：すべてのテストを完了していない最新のパッチ
- 遅延フィード：過去12、24、48時間以内にリリースされたパッチ。これらはスキップすることができ、ロードされません。

このようなフィードはこれらの値の1つをPREFIX変数に割り当てることによってファイル/etc/sysconfig/kcare/kcare.conf に設定されます。

- テスト(テストフィード用)
- 12時間(12時間の遅延フィード用)
- 24時間(24時間の遅延フィード用)
- 48時間(48時間の遅延フィード用)

モニタリング

KernelCareによって保護されているシステムはビルトインメソッドによって、またNagiosやZabbixなどのサードパーティーツールとともにREST APIを使用することによって監視することもできます。

CLNによるモニタリング

以下の例では登録済みのKernelCareインストールはライセンスキーによってグループ化されています。赤字で表示されているものには最新のパッチがインストールされていません。

	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

KernelCare.ePortalアドミンページによるモニタリング

KernelCare.ePortalサーバを使用している場合、アドミンページ(<http://ePortal IP/admin>)を使用してキーIDをフィルタリングすることができます。

KernelCare.ePortal v1.7 Feeds Settings Docs Signed in as admin Log out								
List (3)	Add Filter	With selected	Search					
x Key	equals	111		Reset Filters				
		IP	Hostname	Effective Kernel	Registered	Check In	Server ID	Key
<input type="checkbox"/>		127.0.0.1	localhost.localdomain	3.10.0-693.5.2.el7	05/11/18 11:55:49	05/14/18 04:12:02	nt44Tu49w06lZY1E	111
<input type="checkbox"/>		127.0.0.1	localhost.localdomain	3.10.0-693.5.2.el7	05/14/18 05:21:05	05/15/18 08:56:04	Fm0oJsY488TEX410	111
<input type="checkbox"/>		192.168.249.6	localhost.localdomain	2.6.32-696.16.1.el6	05/15/18 07:28:27	05/21/18 12:42:14	4gnHj2GyP669fD73	111

コマンドラインでのモニタリング

このコマンドで最新のパッチが適用されているかどうかを確認することができます。

```
/usr/bin/kcarectl --check
```


KernelCare APIを使用したモニタリング

KernelCareには監視目的でステータス情報を抽出するために使用できるREST APIがあります。構文は以下の通りです。

- キーベースのライセンスの場合
`https://cln.cloudlinux.com/api/kcare/nagios/key_id`
- IPベースのライセンス(再販業者)の場合
`https://cln.cloudlinux.com/api/kcare/nagios-res/login/token`
- ePortal/パッチ配布の場合
`http://ePortal IP/admin/api/kcare/nagios/key_id`

CloudLinux REST APIの説明は <https://cln.cloudlinux.com/clweb/downloads/cloudlinux-rest-api.pdf> を参照ください。

Nagios/Zabbixの統合

NagiosまたはZabbixのエンタープライズユーザは http://patches.kernelcare.com/downloads/nagios/check_kcare でスクリプトを使用することができます。

このスクリプトは上記の2つのベンダーツールと互換性のある出力を生成するコマンドラインユーティリティです。パッチを次のいずれかに分類します。

- Up to date (最新のもの)
- Out of date (期限切れのもの)
- Unsupported (未サポートのもの)
- Inactive (非アクティブのもの)

このスクリプトはKernelCareキー(CLNまたはKernelCare.ePortalに登録)を持つサーバとパートナーアカウント内のすべてのサーバ(CLNに登録)にレポートします。

NagiosのKernelCareステータスチェックスクリプトを使用したService Statusビューの例を以下に示します。

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

Check_kcareスクリプトを使用するに以下を行います。

1. http://patches.kernelcare.com/downloads/nagios/check_kcare からダウンロードします。
2. それを以下にコピーしてください。
 - /usr/lib64/nagios/plugins/(Nagios用)
 - /usr/lib/zabbix/externalscripts/(Zabbix用)
3. 実行可能な状態にします。

注記: Zabbixのテンプレートは以下にあります。

http://patches.kernelcare.com/downloads/nagios/kcare_zabbix_template.xml.

セキュリティ コンプライアンス

KernelCareは、メモリ内のカーネルに直接パッチを適用するので、正式なパッチIDは変わりません。つまり `uname -r` の出力も `/proc/version` のファイルの内容もパッチを適用しても変わりません。

それは glibc やカーネルABI(アプリケーション・バイナリー・インタフェース)に依存している他のライブラリはカーネルの正しいバージョンを知らなければならないためです。

このアプローチはサーバに最高レベルの安定性と互換性を提供しますが、セキュリティスキャナーの中にはそのアクティブカーネルを期限切れとレポートする原因になることがあります。

このようなレポートを防ぐためにKernelCareには有効バージョンのカーネルを返すコマンドがあります。

```
kcare-uname -r
```

Rapid7™Nexposeからのレポートを調整するための他のスクリプトも利用できます。

Rapid7™NexposeでのKernelCareの使用

KernelCareはカーネルにライブパッチが適用されている(つまり、有効なカーネルバージョンと起動されているカーネルバージョンが異なる)ことをRapid7™Nexposeに通知することができます。これは `kcare-nexpose` スクリプトを使用して脆弱性の例外を追加することによって実施されます。

注記: XML V2形式でレポートされ、キーベースのライセンスを使用したKernelCareインストールのみをサポートします。

このスクリプトはRapid7™Nexposeインスタンスに接続し、KernelCareパッチが適用されたセキュリティレポートを見つけ、すべてのバイナリーパッチが適用されたCVE(共通脆弱性識別子)に例外を追加します。

CVEのリストはインストールのタイプに応じて、KernelCare Patch ServerまたはKernelCare.ePortalサーバから取得されます。

CVEリストの変更を反映するために、レポートは自動的に返却されます。CVEを自動的に承認し、以前のカーネルバージョンで追加された期限切れのものを削除することができます。

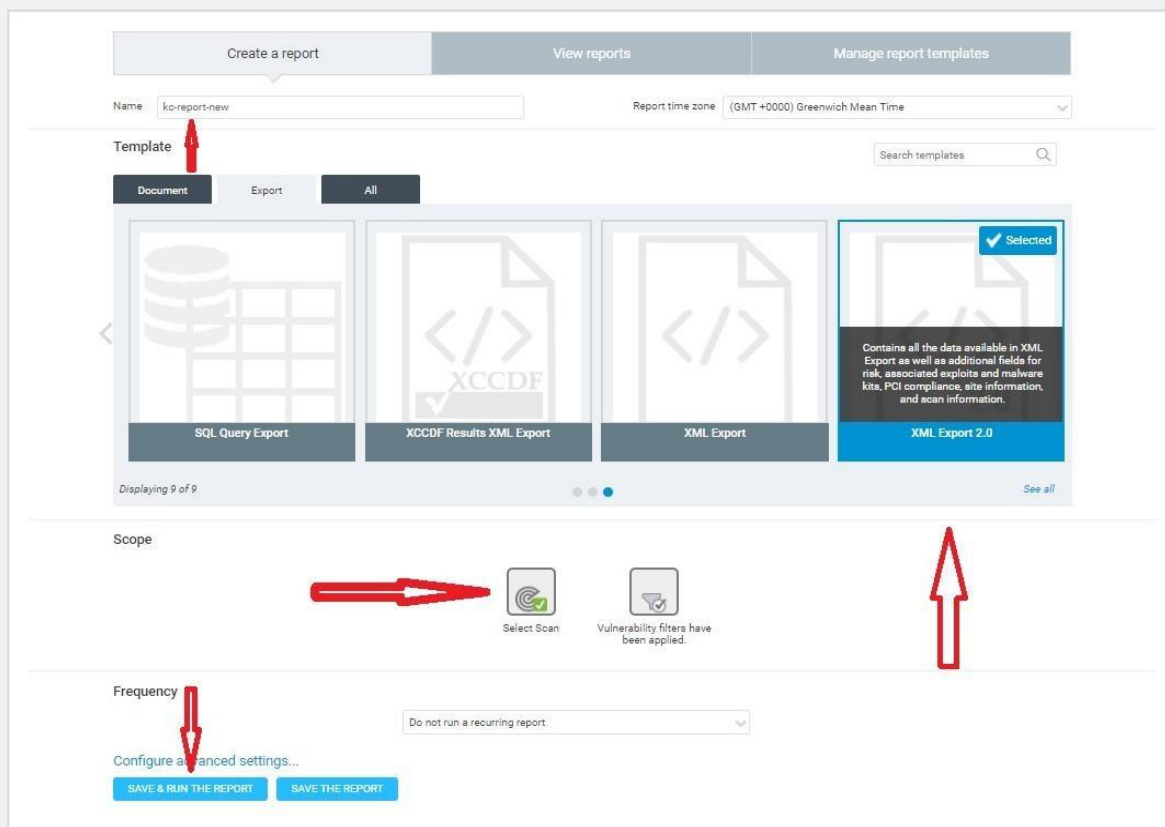
スクリプトのインストールと実行

注記: これらの説明はEL6ベースのシステム用です。

1. リポジトリの場所を設定します。

```
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. インストールします。

```
yum install kcare-nexpose
```
3. `/usr/local/etc/kcare-nexpose.yml` に構成ファイルを作成します。
 注釈付きのサンプルは http://docs.kernelcare.com/yaml_config_file_description.htm を参照ください。
4. 例外を追加するためにベースとして使用するレポート(下記の例)を生成します。



5. スクリプトを実行し、古いCVEを削除し、新しいCVEを追加します。
kcare-nexpose -c/usr/local/etc/kcare-nexpose.yml
6. 以下に示すように脆弱性の例外がアセット毎に個別に追加されます。

VULNERABILITY EXCEPTIONS

REVIEW		DELETE		Items Selected: 0 of 152				
<input type="checkbox"/>	Vulnerability	Exception Scope	Reason	Reported By	Reported on	Review Status	Reviewed on	Expires On
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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.

結論

このテクニカルホワイトペーパーではKernelCareのインストールと設定における重要なポイントについて説明しました。
またLinuxカーネルパッチマネジメントの主な要件についても説明しました。

- パッチの自動インストール
- カスタムパッチ構成とマニュアルでの上書き
- パッチリポジトリの選択
- 自動化および監視ユーティリティとの統合

	KernelCare.ePortalを使用	KernelCare.ePortalを使用しない
アップデートサーバの場所	オンプレミス	http://patches.kernelcare.com
ライセンスサーバの場所	オンプレミス	https://cln.cloudlinux.com
インストール指示	ePortal Server	KC agent
費用	ライセンス毎 追加費用なしにePortalも含まれる	ライセンス毎
パッチ展開の柔軟性	ePortal Feeds	Sticky patches (Stickyパッチ)
複数環境の構築	Yes (ePortal feeds を利用)	Yes (Stickyパッチを利用)
モニタリング	ePortal、REST API (Nagios, Zabbix)	CLN、REST API

詳細情報

- KernelCareのウェブサイト: <https://www.kernelcare.com/ja/>
- KernelCareのブログ: <https://www.kernelcare.com/blog/>
- KernelCare Patch Server: <https://patches.kernelcare.com/>
- KernelCareのドキュメンテーション: <https://docs.kernelcare.com/jp/>
- CloudLinux Network – CLN (ポータル): <https://cln.cloudlinux.com>
- CloudLinux 24時間365日オンラインサポートシステム: <https://cloudlinux.zendesk.com>

KernelCareに関するお問合せ: 日本国内正規代理店
GDEPソリューションズ株式会社
 TEL: 03-5802-7050 / E-mail: kcsales@gdep-sol.co.jp
 URL: <http://www.gdep-sol.co.jp>