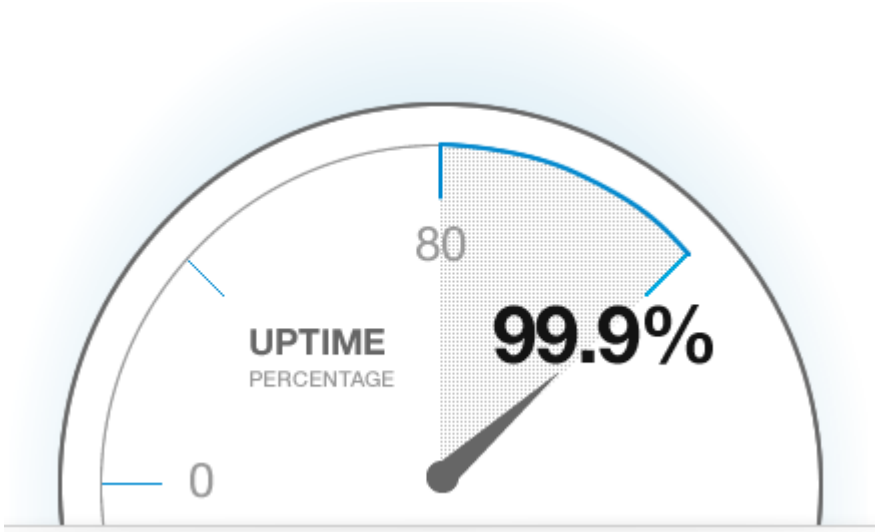
Virtuozzo

ReadyKernel — инструментарий и сервис обновления ядра без перезагрузки на основе kpatch

Денис Силаков, Евгений Шатохин

Проблематика

- Обновление ядра требует перезагрузки
  - Или хотя бы “рестарта” через kexec
- … А перезагружаться не хочется
  - GoDaddy - uptime at 99.95% (https://goo.gl/ek6EeY)



Изменения ядра “на лету”

Подходит для точечных изменений

- Исправление ошибок / проблем безопасности

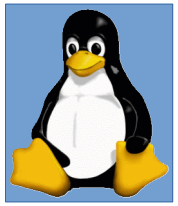
Применимость

RHEL 7.x, ядро 3.10.x, 2015 год:

- **800+** коммитов, из них:
  - ~70 CVE
    - ~10 критических
    - ~40 средней опасности

Инструменты

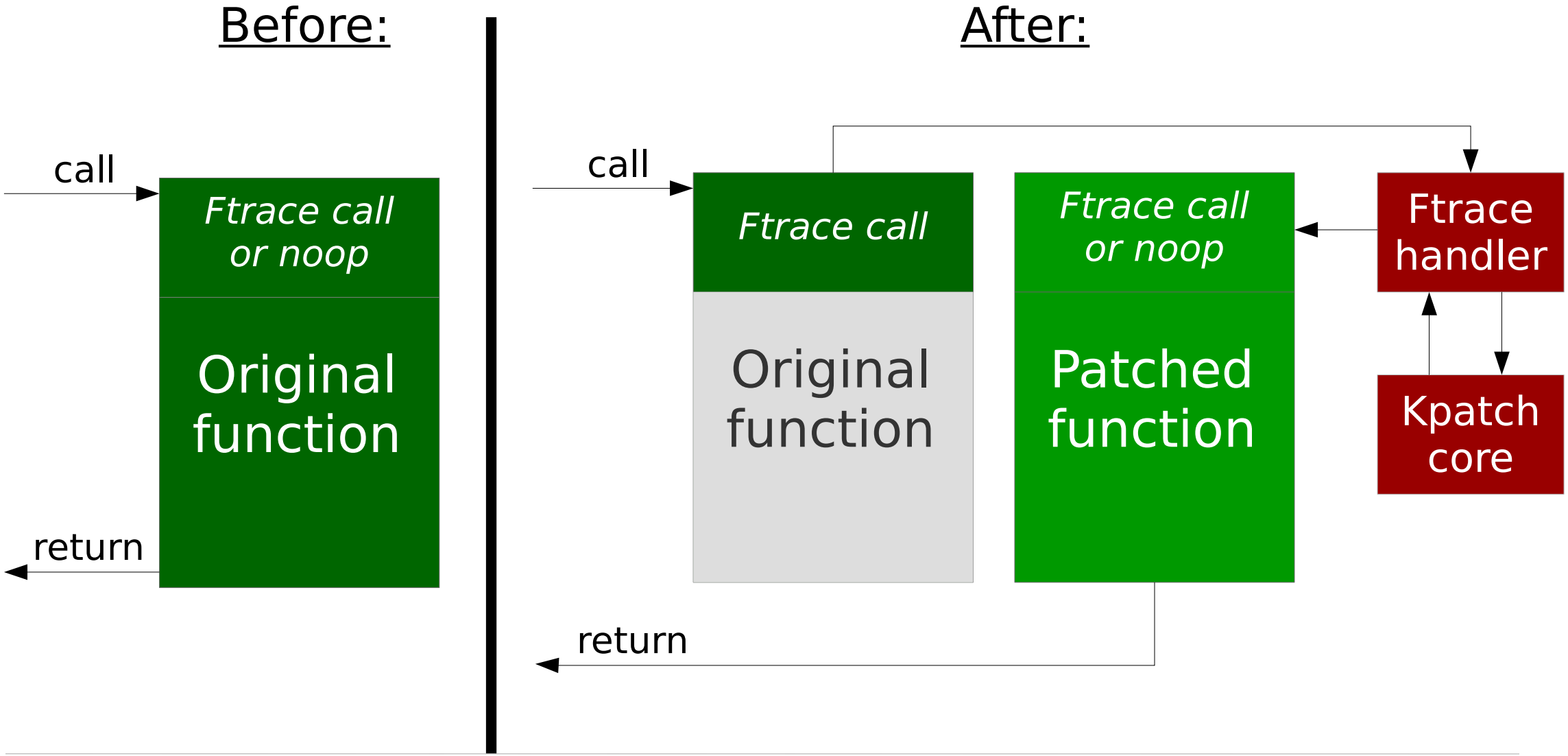
- Ksplice
- KernelCare
- Kpatch
- kGraft
- Livepatch



Технология

- Загружаем в память новую реализацию функции
- Перенаправляем обращения к старому коду на новый
  - Jmp в начале старой функции (KernelCare, kGraft)
  - Перехватывая вызовы через ftrace (Kpatch)

ReadyKernel – сервис на основе Kpatch

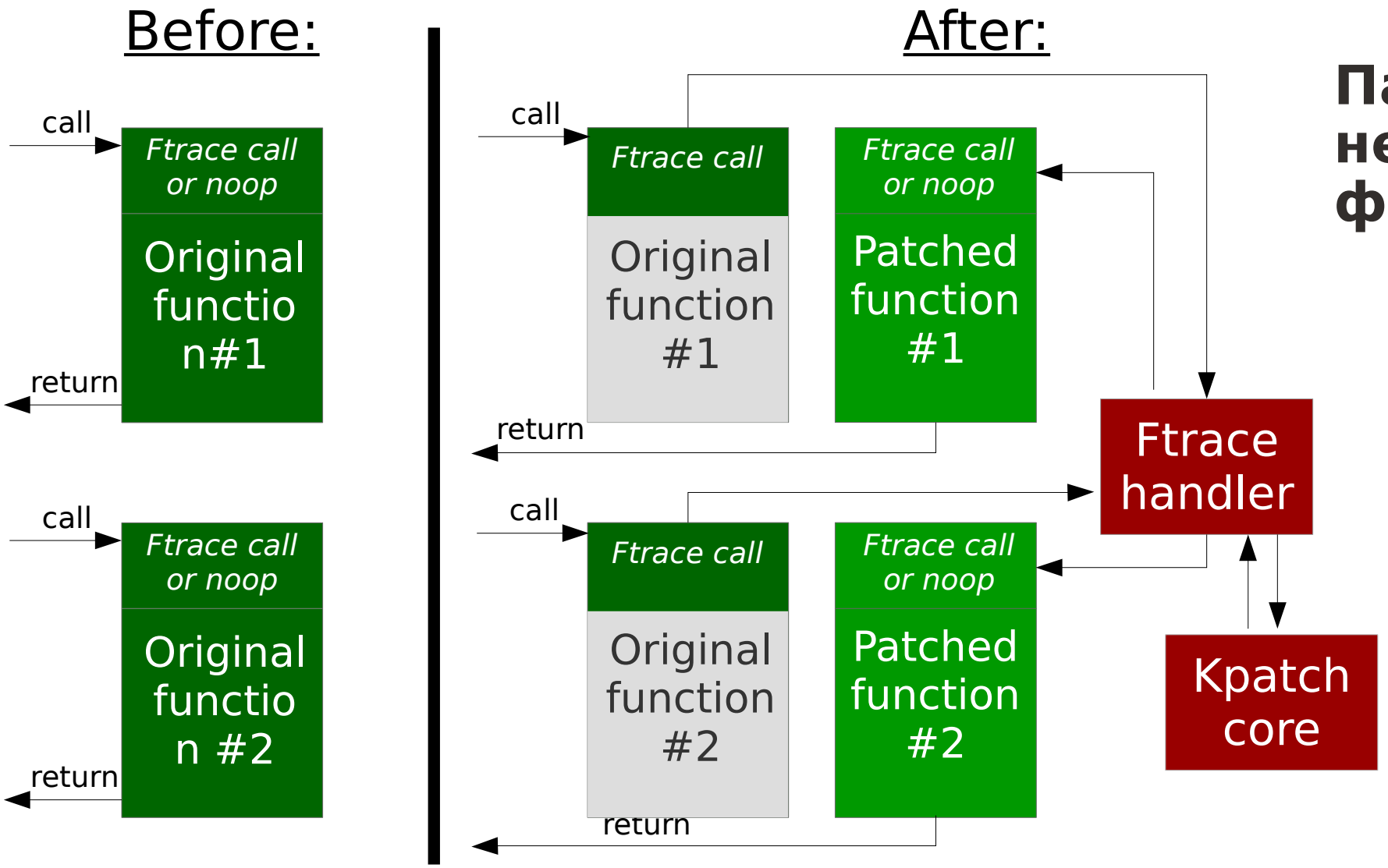


Почему Kpatch?

Virtuozzo 7: based on RHEL 7

- **Kpatch** – introduced by RHEL
- **Gcc** 4.8.5 (gcc >= 4.8 required for -mentry)
- **Kernel** 3.10.x (kernel >= 3.9 required for kpatch ftrace handlers)

Модели целостности



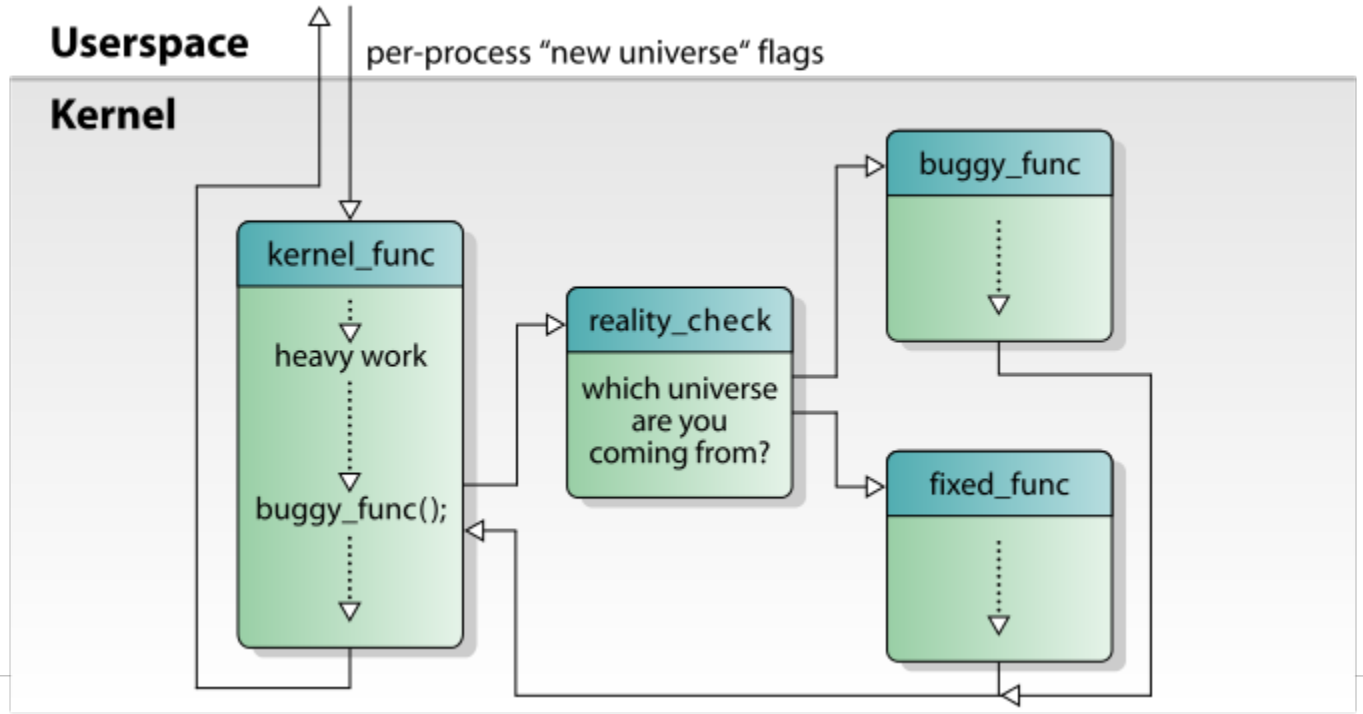
Патч на несколько функций ?

Варианты моделей

- **“Никакая”** – *Livepatch* в ядре 4.0
- **“Clean cut”** – *Kpatch* – проверяем стеки вызовов активных процессов (stop_machine)
  - для взаимосвязанных функций

Варианты моделей - 2

- **“Per-task”** – *kGraft, Livepatch* в ядре 4.12
  - “RCU” для каждого потока
  - Рекурсия, зависимость от истории – но one-function only



Варианты моделей - идеал

- **“Hybrid”** – объединение идей “Clean Cut” и “Per-task”

  Work-in-progress в Livepatch

Создание патчей (kpatch)

Патч == модуль ядра

- Берем патч для исходного кода
- Собираем ядро с патчем и без
- Анализируем разницу
- ... и превращаем ее в модуль ядра

Автоматизировано, но требует проверки

Проблемы (kpatch)

- Изменение структур данных
- Изменение семантики данных
- ...

  “The long answer is a book which unfortunately we haven't written yet.”

Реальная жизнь

99% времени – подготовка патчей

=> открытый инструментарий, закрытые патчи

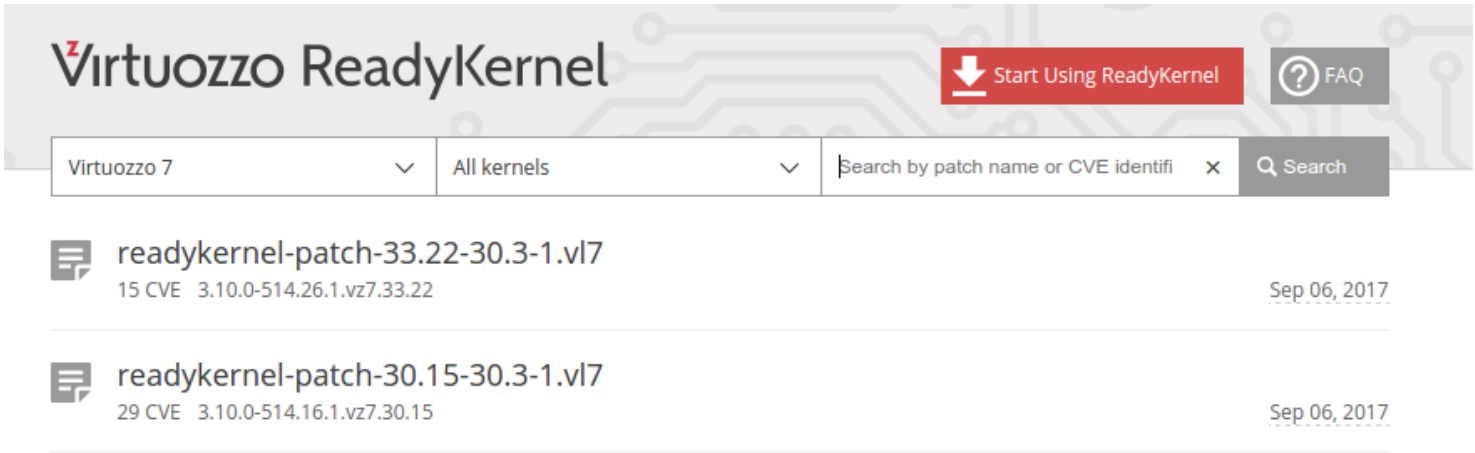
- **SUSE** – kGraft по подписке
- **Ubuntu** – Livepatch по подписке
- **RHEL** – kpatch в состоянии Technical Preview
- **Virtuozzo** – kpatch по подписке

Virtuozzo ReadyKernel

Патчи по подписке

Состав:

- “Подпатченный” Kpatch
- Обертка readykernel + портал http://readykernel.com
- Инструмент работы с лицензией



ReadyKernel на практике

- **Virtuozzo 7** - Обновления ядра между крупными обновлениями (~1 раз в квартал)
- PoC для **CentOS**, **RHEL**, **Ubuntu**, **Debian**

  (инфраструктура сборки и доставки патчей)
- **OpenVZ 7** - ? (не в этом году)

“ReadyKernel” for UserSpace?

- **Обновление гипервизора** без перезапуска ВМ
  - Через живую миграцию Qemu
- **NSB** https://github.com/virtuozzo/nsb
  - Остановка процесса – **ptrace**
  - Модель целостности – call stack через **libunwind**
  - Загрузка патча – **mmap + libcompel** from CRIU

  Устранение Heartbleed: https://youtu.be/lxP1zLvlczA
- **LibCare** https://github.com/cloudlinux/libcare

Спасибо!

Денис Силаков <dsilakov@virtuozzo.com>
Евгений Шатохин <eshatokhin@virtuozzo.com>

https://readykernel.com
https://src.openvz.org/projects/UP/repos/kpatch