Elivepatch
Flexible distributed Linux Kernel live patching

Alice Ferrazzi

kernel :~ $ whoami

Alice Ferrazzi

- Gentoo
  - Gentoo Kernel Project Leader
  - Gentoo Kernel Security
  - Gentoo Foundation board member
  - Gentoo Google Summer of Code administrator and mentor for rust Gentoo project
- Cybertrust Japan
  - OSS Embedded Software Engineer

Summary

- Live patch explanation
- Current live patch services
  - Motivation for elivepatch
- Elivepatch solution
  - Implementation
  - Challenge
  - Status
  - Future
- Conclusion

At first this project was part of Google Summer of Code 2017 for the Gentoo organization.

Live patch explanation

Live patch

Modify the kernel without the need to reboot.

Why

- Downtime is expensive (containers, supercomputers)
- Security (vulnerability time shorter)

Where

- Embedded
- Desktops
- HPC (complex scientific computations)
- Cloud
- Any computer under heavy load

What

Kgraft

Suse Open Source live patching system that is routing the old function gradually.

Kpatch

Red Hat Open Source live patching system and use ftrace and stop_machine() for route functions toward the new function version.

Livepatch

Livepatch is a hybrid of kpatch and kgraft.

Livepatch has been merged into the kernel upstream.

Kpatch-build can work with both kpatch and livepatch for creating the live patch.

Livepatch is just a module

...

Livepatch module problem

A module that takes just about 1+ hour
to compile in a modern server

At Gentoo, we know what means to compile something for more than 1 hour…

Gentoo solution to compile for 1+ hour compilation problem

- Gentoo “binary host”
- Pre-compiled binary

What options do we have for compiling livepatch modules?

Current existing livepatch services

Current vendor solutions

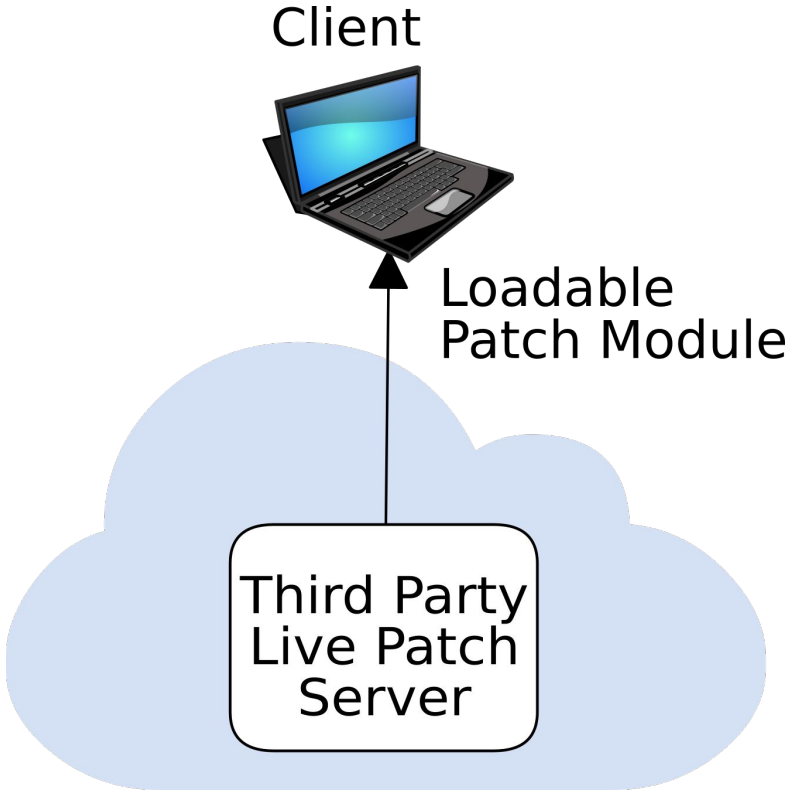
- Oracle, Ksplice (support only Oracle Linux kernels)
- Suse Linux Enterprise Live Patching (support only Suse Kernels for one year)
- Canonical Live Patch (support only Ubuntu 16.04 LTS and Ubuntu 14.04 LTS)
- Red Hat live patch (Support only Red Hat kernel)

Motivation for elivepatch

Problems of vendor solutions

- **trusting on third-party** vendors
- Lacking support for **custom kernel configurations**
- Lacking support for **request-driven** customization
- Lacking **long term** support
- **Closed source**

Vendor solutions representation



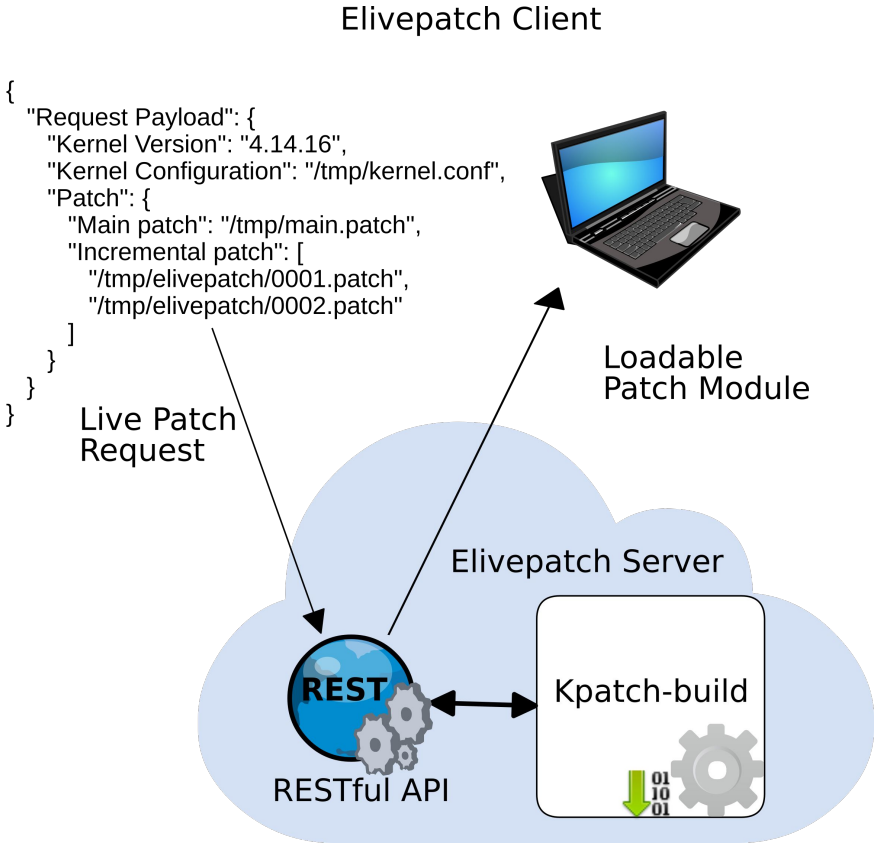
elivepatch solution

elivepatch

A web service framework to deliver Linux kernel live patches

- Supports **custom kernel configurations**
- User participation via **request-driven** customization
- **Open source**

Elivepatch solution



Implementation

Elivepatch-server (Main language: Python)

Flask + Flask-Restful + Werkzeug

Elivepatch-client (Main language: Python)

Requests + GitPython

Challenges

Challenges with elivepatch

- Some patches require manual modification to be converted to live patches
- Reproducing the build environment can be difficult:
  - Differences in compiler versions
  - Variations in the compiler and optimization flags
- Incompatible machine architectures (solaris, hpc)

Incompatibility with GCC

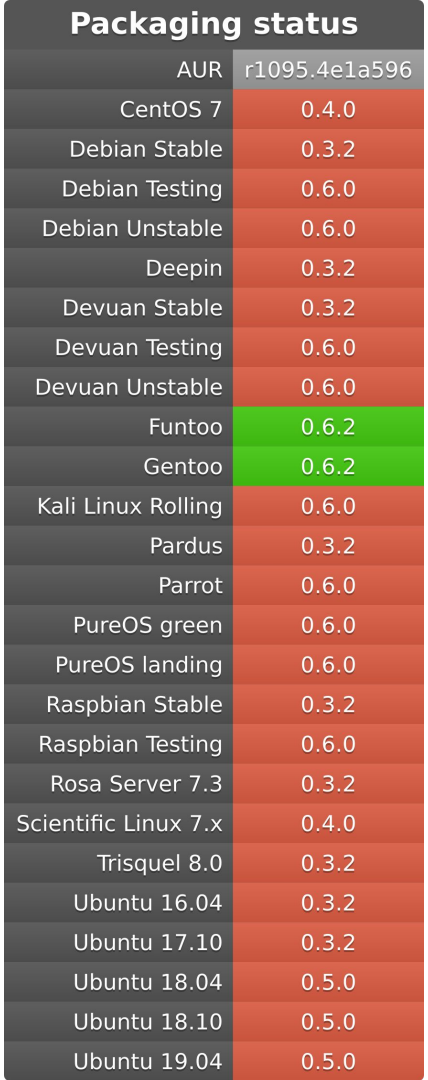
CCFLAGS and non vanilla gcc,

can sometime break elivepatch.

Current status

Elivepatch status

- First open source release 0.1 on 2017/9/06
- Packaged for Gentoo
- Kpatch version 0.6.2 in Gentoo
- Presented as poster at SOSP 2017
- Close collaboration with kpatch mainteiners


Packaging status
AUR r1095.4e1a596
CentOS 7 0.4.0
Debian Stable 0.3.2
Debian Testing 0.6.0
Debian Unstable 0.6.0
Deepin 0.3.2
Devuan Stable 0.3.2
Devuan Testing 0.6.0
Devuan Unstable 0.6.0
Funtoo 0.6.2
Gentoo 0.6.2
Kali Linux Rolling 0.6.0
Pardus 0.3.2
Parrot 0.6.0
PureOS green 0.6.0
PureOS landing 0.6.0
Raspbian Stable 0.3.2
Raspbian Testing 0.6.0
Rosa Server 7.3 0.3.2
Scientific Linux 7.x 0.4.0
Trisquel 8.0 0.3.2
Ubuntu 16.04 0.3.2
Ubuntu 17.10 0.3.2
Ubuntu 18.04 0.5.0
Ubuntu 18.10 0.5.0
Ubuntu 19.04 0.5.0

Future
What elivepatch needs

Future

- livepatch automatization
- Multi distribution
- Livepatch signing
- Kernel CI\CD check
- Elivepatch overlay

livepatch automatization

- Automatize the livepatch creation when there are no semantic changes.
- Tool for creating the extra relocations entries.

Multi distribution

Solve distributions compatibility issues

Current target:

- Debian
- Fedora
- Gentoo
- Android

Elivepatch client on Debian

```
root@debian-amd64:~/elivepatch-client# clear
root@debian-amd64:~/elivepatch-client# PYTHONPATH=/root/elivepatch-client python3 bin/elivepatch --kernel 4.9.0 --url http://192.168.122.2:5000 --debug --version --config /boot/config-4.9.0-6-amd64 --patch ~/main
patch --distro debian
Namespace(clear=False, conf_file=None, config='/boot/config-4.9.0-6-amd64', cve=False, debug=True, distro='debian', kernel_version='4.9.0', patch='/root/main.patch', url='http://192.168.122.2:5000', version=True)
List of current patches:
[]
This session uuid: 5cadb73d-cd16-4a08-9648-5398b4b56e3d
debug: kernel version = 4.9.0
incremental_patches: []
[('main_patch', ('main.patch', <_io.BufferedReader name='/root/main.patch'>, 'multipart/form-data', {'Expires': '0'})), ('config', ('config', <_io.BufferedReader name='/tmp/tmpl3_cv_v2'>, 'multipart/form-data', {
Expires': '0'}))]
send file: {'Response': 'debian is not yet supported'}
livepatch not received
root@debian-amd64:~/elivepatch-client#
```

Work in progress…

https://asciinema.org/a/187738

p.s. Gentoo kernel is still needed

Livepatch signing

- Implementing livepatch module signing in the server
- Implementing signing verification for the client

Kernel CI/CD checking

- Implement a buildbot plugin for testing elivepatch
- Implementing elivepatch-server on docker,
  for a ready to use livepatch building instance

[You can test your livepatch with the same settings and hardware as where you want to deploy it]

elivepatch overlay

Collaborative livepatch creation

Similar to Gentoo overlay for livepatch

example:

https://github.com/aliceinwire/elivepatch-overlay

Conclusion

Epilogue

- Livepatch is a module that takes time compiling
- Livepatch vendor service solutions solve the compilation problem in a propietary way
- Elivepatch offers a wider solution

With the diffusion of embedded systems and robotics,

Livepatch services will become always more important

https://github.com/gentoo/elivepatch-client

Please send every issues you found

We opened the first elivepatch server node:

http://elivepatch.amd64.dev.gentoo.org:5000

If you are interested in contributing,
Elivepatch is welcoming every form of contribution.