Ksplice

Linux kernel patching without a reboot

Udo Seidel

Frühjahrsfachgespräch 2013

Agenda

- Who?
- Why?
- How?
- Show!
- And?

Me :-)

- Teacher of mathematics and physics
- PhD in experimental physics
- Started with Linux in 1996
- Linux/UNIX trainer
- Solution engineer in HPC and CAx environment
- Head of the Linux Strategy team @Amadeus

Why?

Why kernel updates?

- Business critical applications on Linux
  - Bug fixing
  - New functions or improvements
  - External requirements
- Importance of security, e.g. PCI-DSS

What is 'wrong' with reboots?

- Missing HA
- Procedures, Operations, ...
- External requirements

Question ....

Do we really need a reboot?

Looking back and around

- Not new
  - mainframes
  - hot updates for Unix
  - Early days of Linux
- Picked up 'recently'
  - Rootkits

How?

Hotfix preparation

Source code comparison

- One approach for generation of hot updates
- Looks simple ... but
  - High programming language skills needed
  - Analysis complex
  - Code replacement unclear

Object code comparison

- New approach for generation of hot updates
- Advantages
  - Reduced need for developing skills
  - Implicit patch analysis
  - Can be automated
- Challenges
  - Object code generation
  - Code replacement

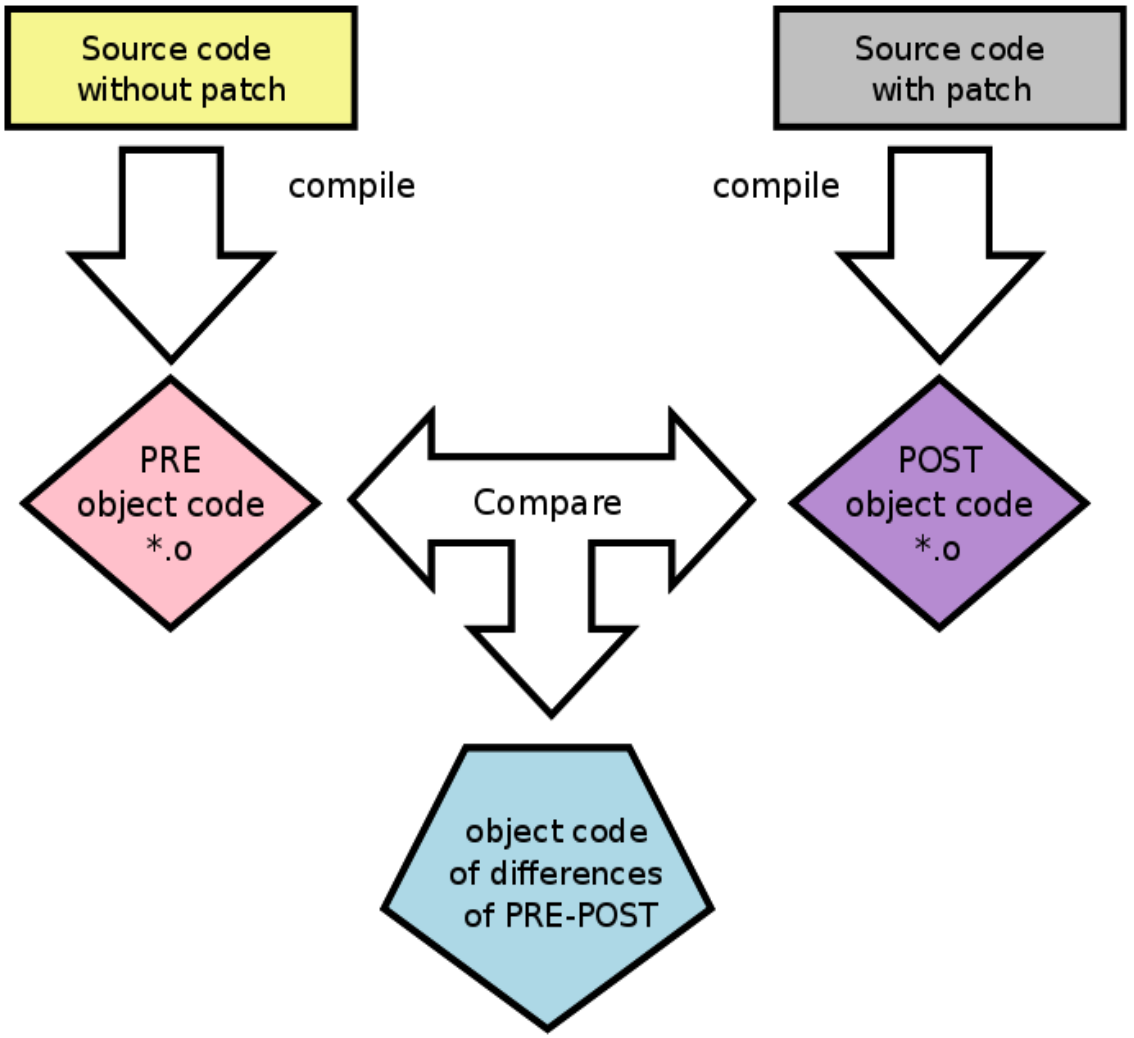
Ksplice arises ...

- 2008/2009
  - 4 students @ MIT
    - Thesis from Jeff Arnolds
  - Ksplice Inc. founded
    - GPLv2
    - Supported: Debian, Ubuntu, Fedora, CentOS, RHEL
- July 2011
  - Acquired by Oracle

Ksplice – high level

- Patching original source code
- Generation of new object code
- Comparison of 'old' and new object code
- Load of the delta code
- Address redirection to activate new object code

Ksplice – how it works part I

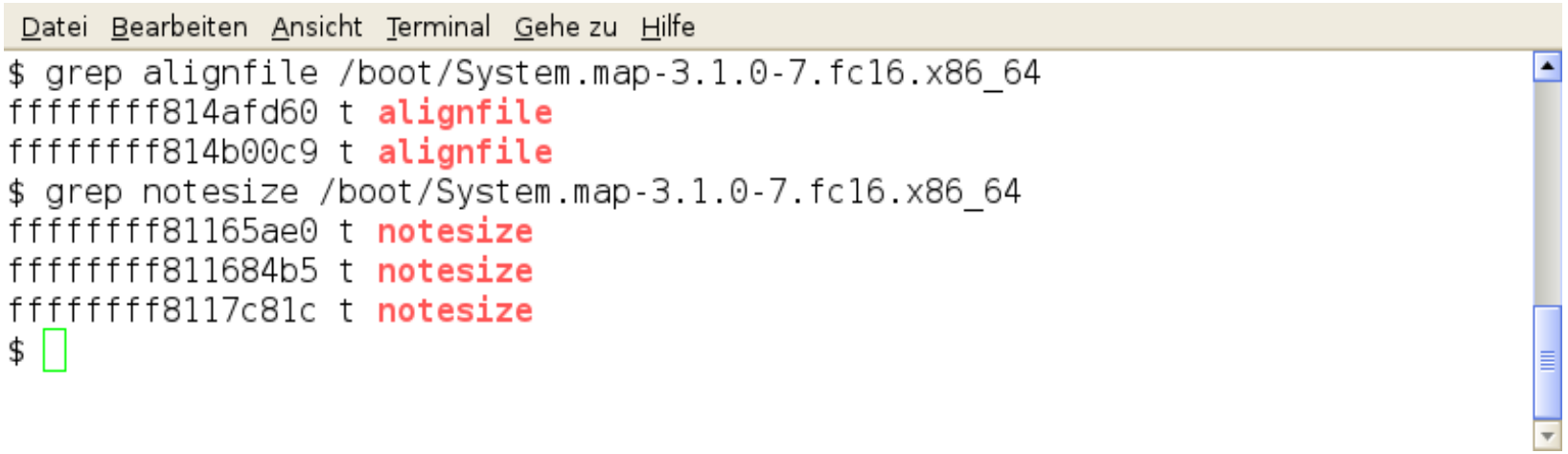


Ksplice – some first details

- Creation of dedicated sections for functions and structures of source code
- Identical compiler recommended
- Unnecessary replacement code possible
- Memory addresses?

Address determination – first trial

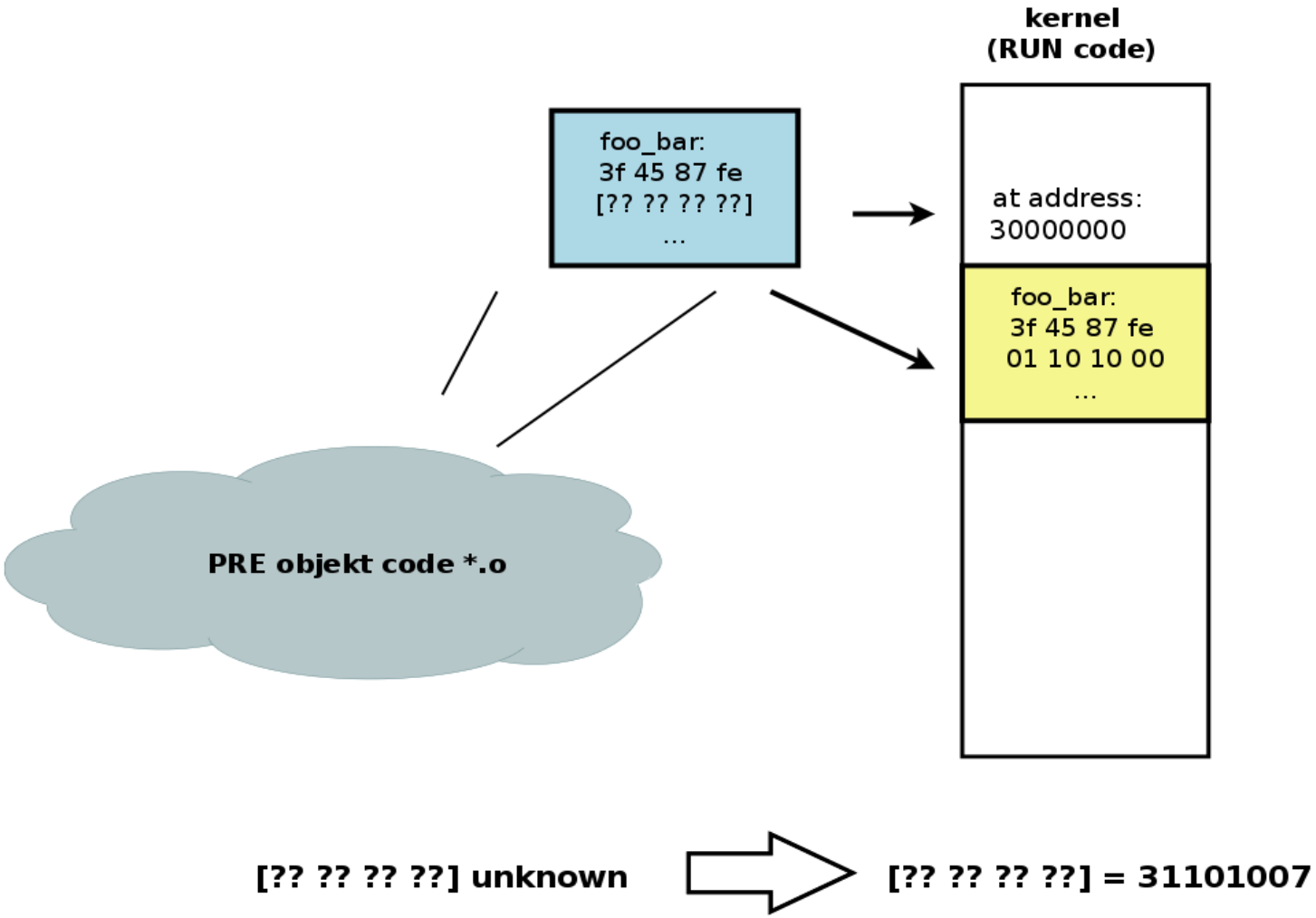
- Symbol to address translation
  - Known from core dump analysis
  - `/boot/System.map`
  - Does not work here :-(



Address determination by Ksplice

- Pair up object code on disk (PRE) with object code in memory (RUN)
- Filter of 'wrong' symbols
- Analysis of unknown address in PRE objects via RUN objects

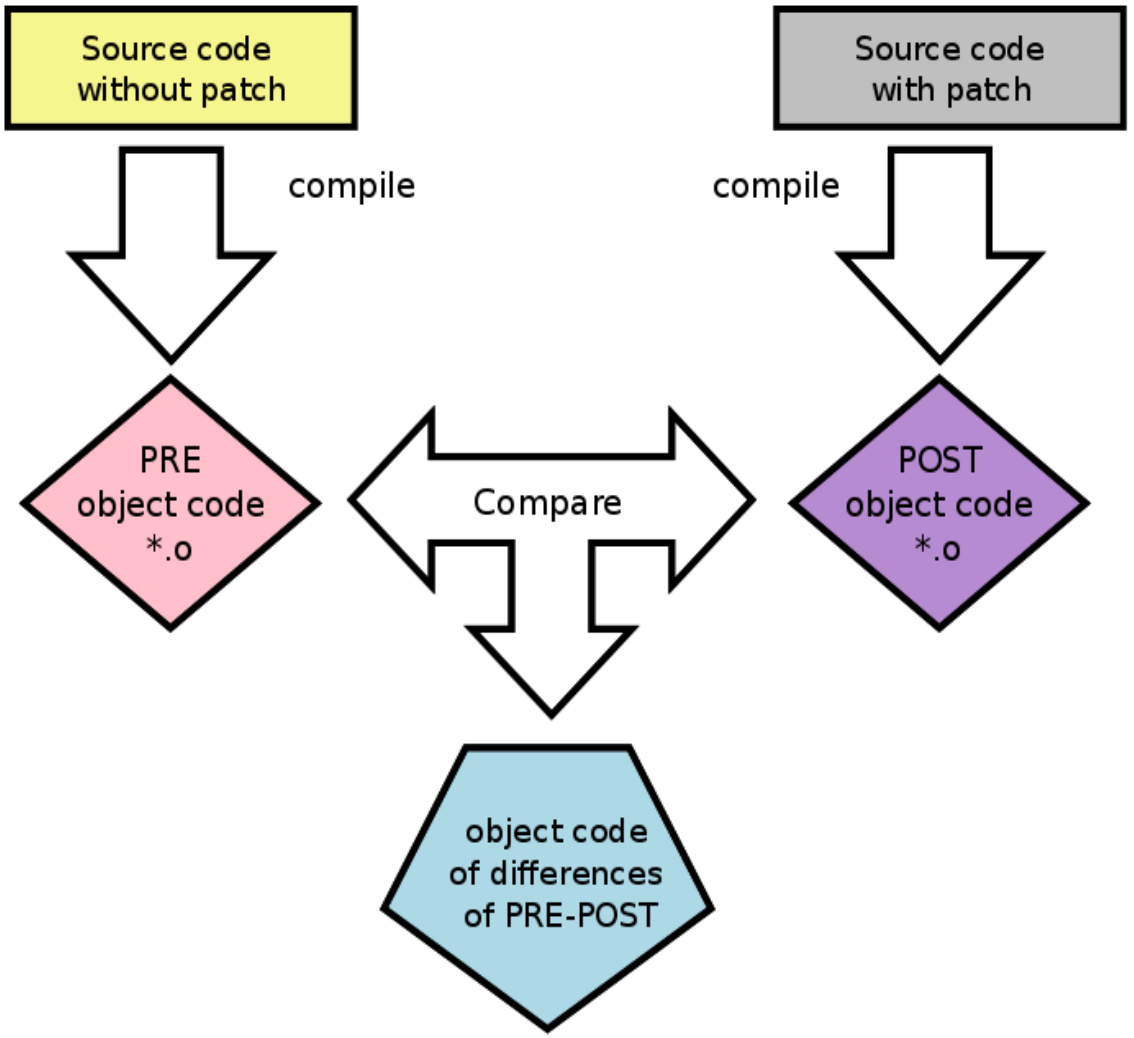
Address determination by Ksplice



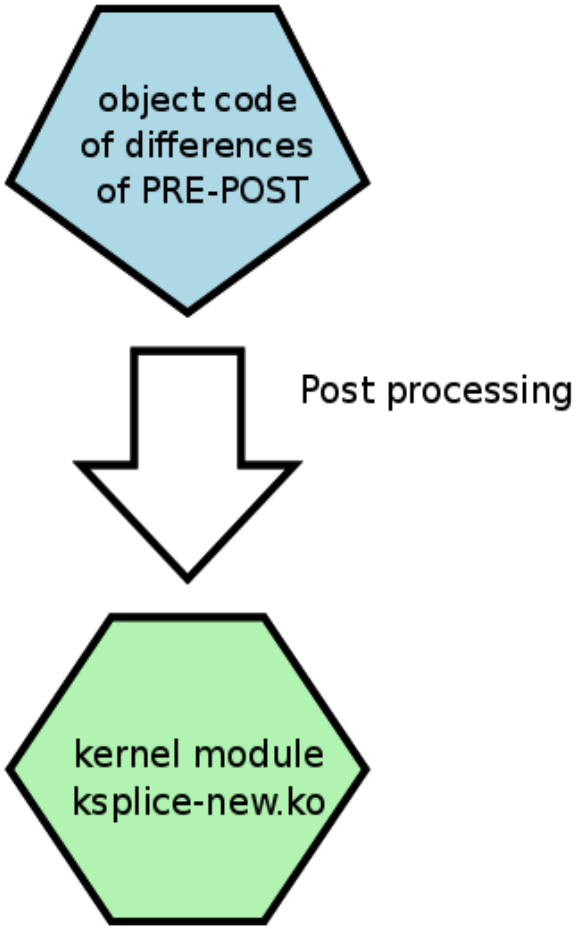
Ksplice - more details

- Cancel if unexpected event during PRE-RUN comparison
- PRE-RUN comparison => kernel module

Ksplice – how it works part I



Ksplice – how it works part II



Last step

- Tell the kernel to use the new object code
  - Kernel module `ksplice.ko`
- `Load the new object code`
  - Kernel module `ksplice-new.ko`

What else

- Limitations
  - Scalability
  - Patch context
  - Patch size
- Future
  - Offline client (available since 01/2013)
  - SUSE?!
  - Applications?

How?

Hotfix application

Client side

- Download via Ksplice client
- Modification of `depmod` and `modprobe`
- Update of initial ram disk
- Automation via SysV init scripts

Client commands

Command	description
`uptrack-install`	Install available patch(es)
`uptrack-remove`	Remove installed patch(es)
`uptrack-show`	Status of installed and/or available patches
`uptrack-uname`	Effective/patched kernel version
`uptrack-upgrade`	Install all availabe patches

Show!

Example

- CVE-2012-0207
  - Kernel 2.6.36
  - Bug in IGMP code
- Simple source code fix

```
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c ...
@@ -875,6 +875,8 @@ static void igmp_heard_query ...
...
        max_delay = IGMPV3_MRC(ih3>code)*...
+         if (!max_delay)
+         max_delay = 1;   /* can't mod w/ 0 */
    } else { /* v3 */
```

And?

Summary

- Promising technology
- Available for selected Linux distributions only
- Operational check needed

References

- http://www.ksplice.com
- http://kerneltrap.org/mail-archive/linux-kernel/2008/4/23/1570474
- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0207

Thank you!

Ksplice

Linux kernel patching without a reboot

Udo Seidel

Frühjahrsfachgespräch 2013