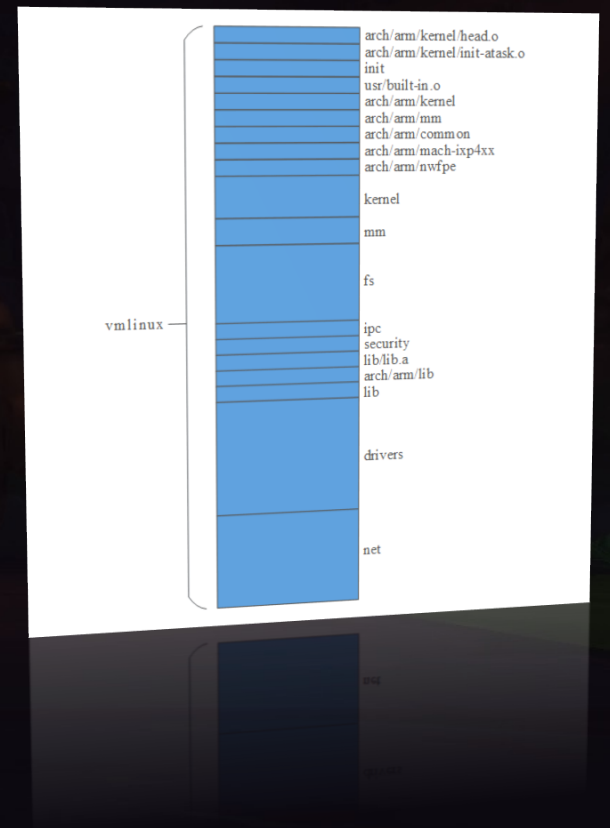
Patchwerk: Kernel Patching for Fun and Profit

Parker Wiksell - @pwiksell - wiksellp@battelle.org

Jewell Seay - seayj@battelle.org

ShmooCon 2019

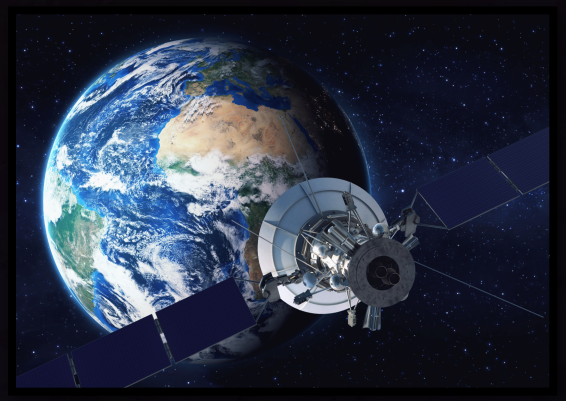


Who are we?

Parker Wiksell
@pwiksell
wiksellp@battelle.org

Jewell Seay
seayj@battelle.org

BATTELLE



A little background...

Problem Domain

Existing open source base

Customizations

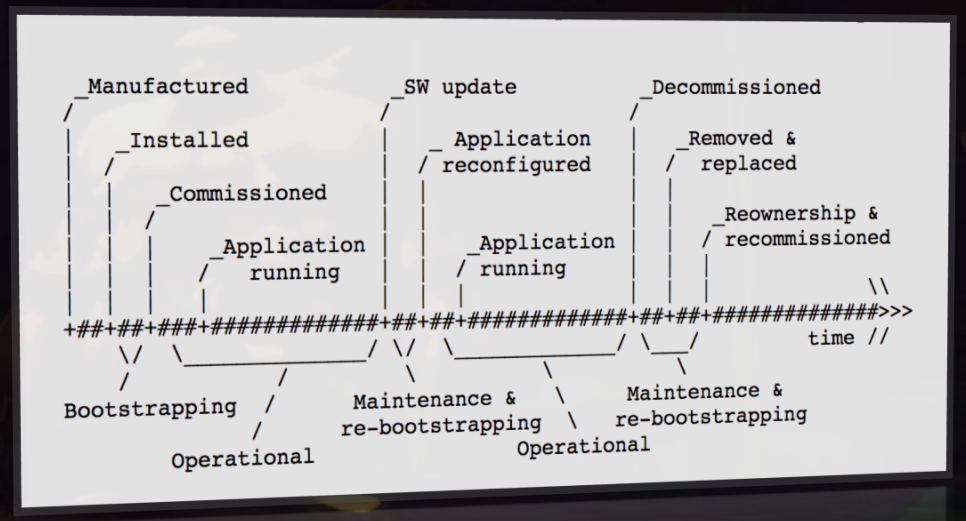
Manufactured

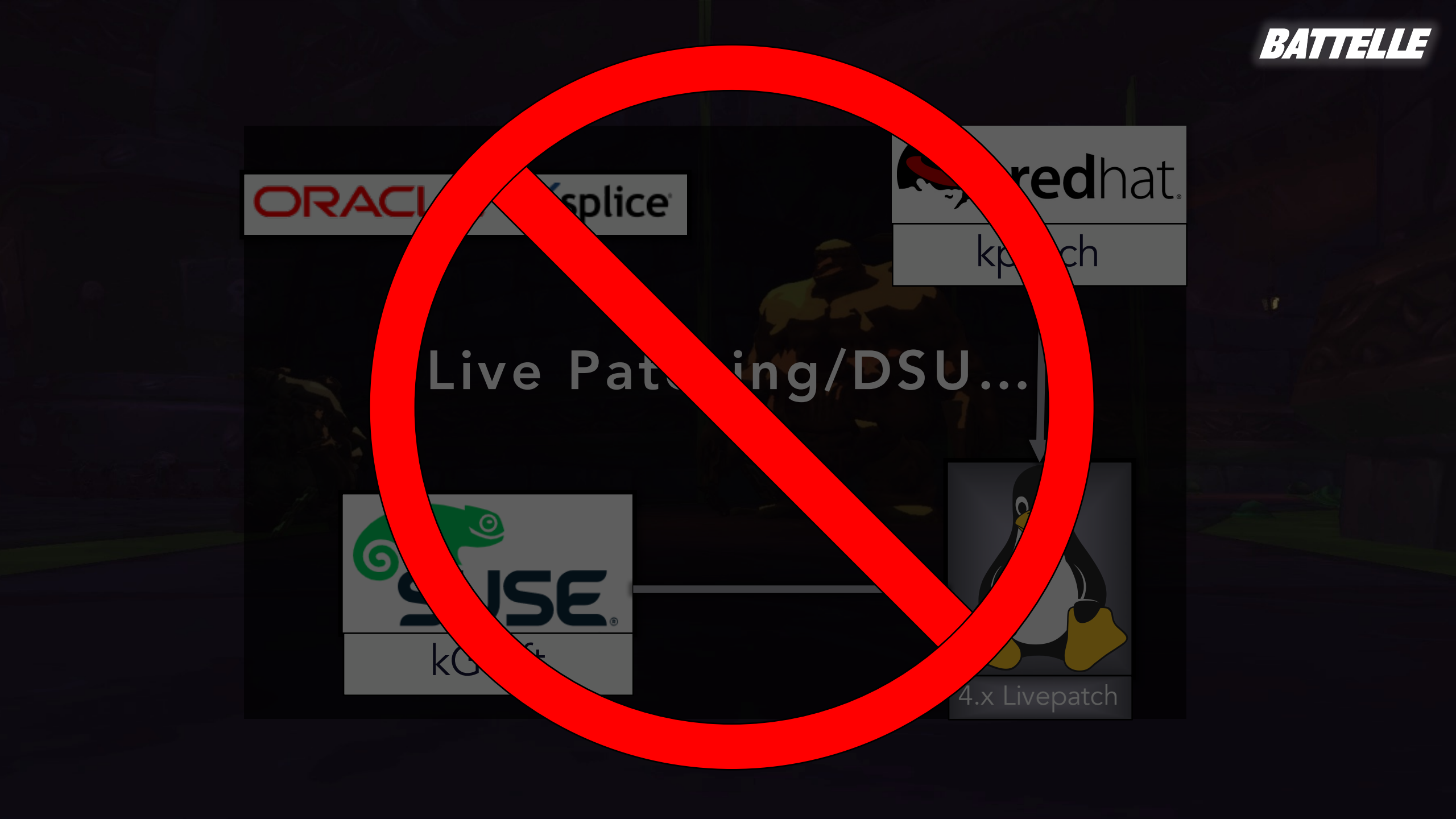
Re-branded

Installed and configured

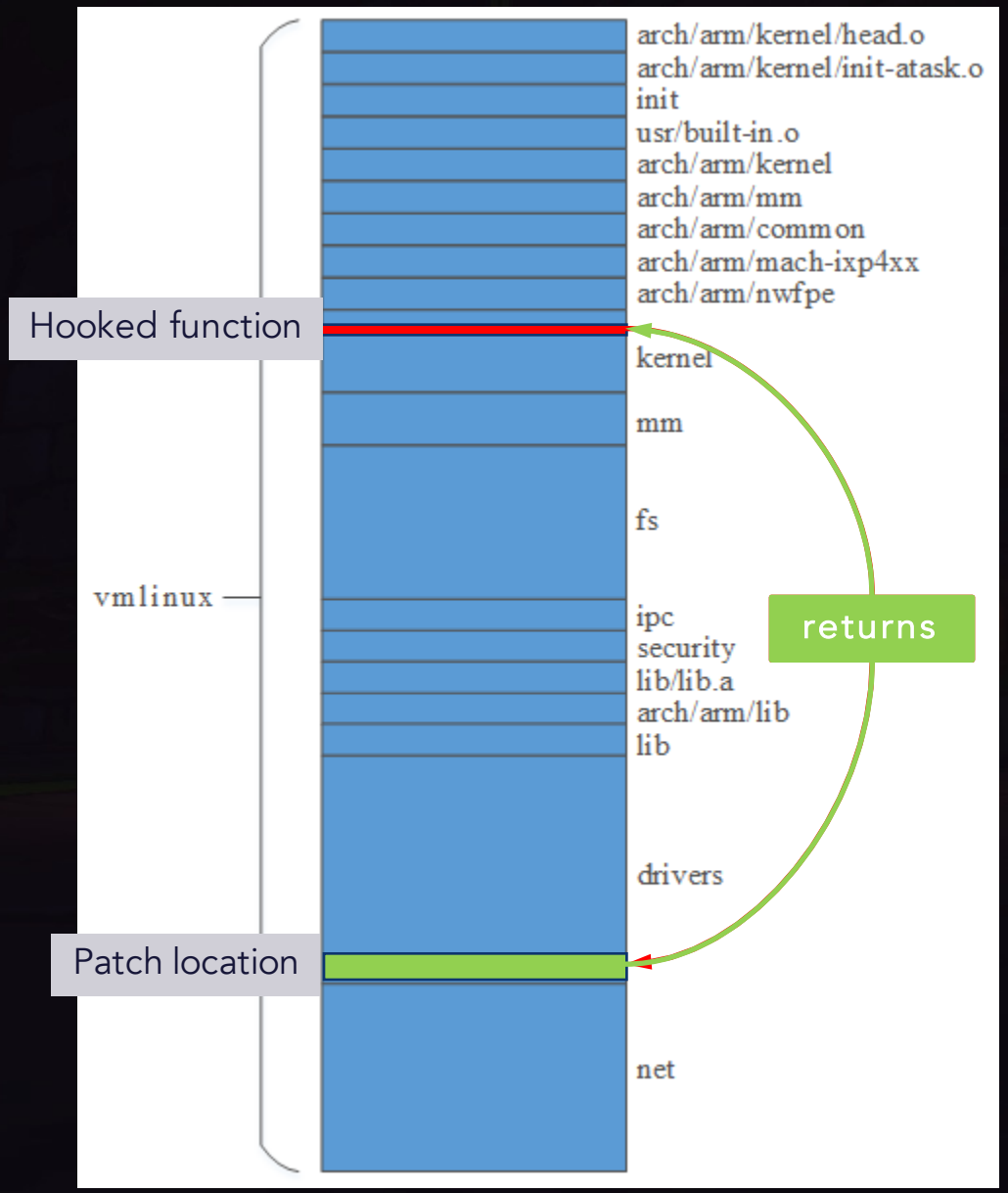
Maintenance and upgrades

End-of-life



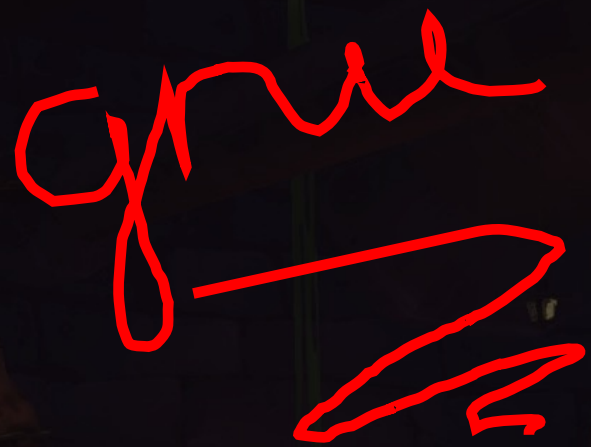

BATTELLE
ORACL
splice
redhat.
kp ch
Live Pat ing/DSU...
kG
4.x Livepatch

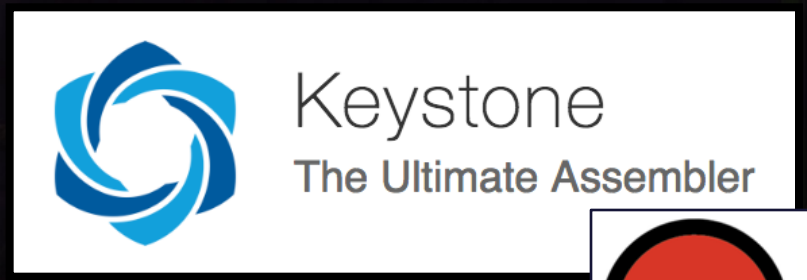

/dev/kmem
-Silvio Cesare


arch/arm/kernel/head.o
arch/arm/kernel/init-atask.o
init
usr/built-in.o
arch/arm/kernel
arch/arm/mm
arch/arm/common
arch/arm/mach-ixp4xx
arch/arm/nwfpe
Hooked function
kernel
mm
fs
vmlinux
ipc
security
lib/lib.a
arch/arm/lib
lib
returns
drivers
Patch location
net

BATTELLE

~~Roll our own?~~

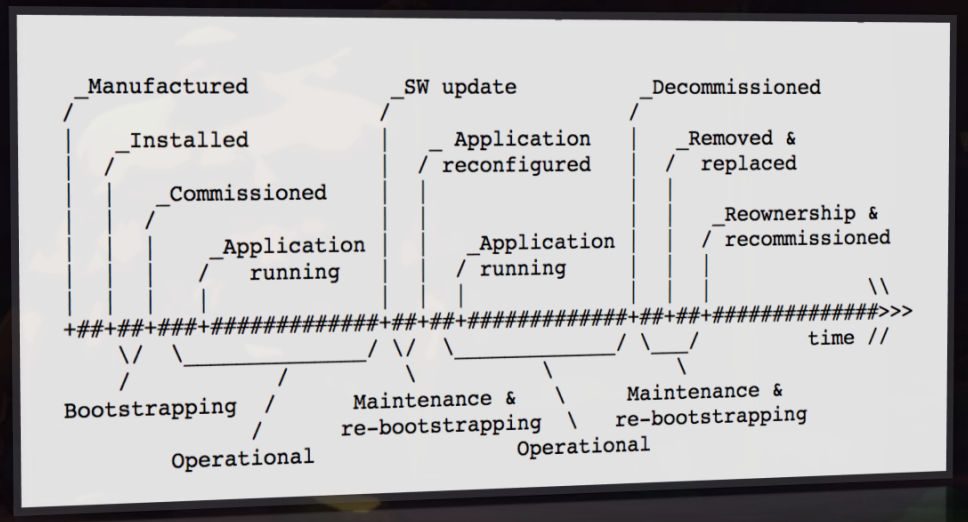




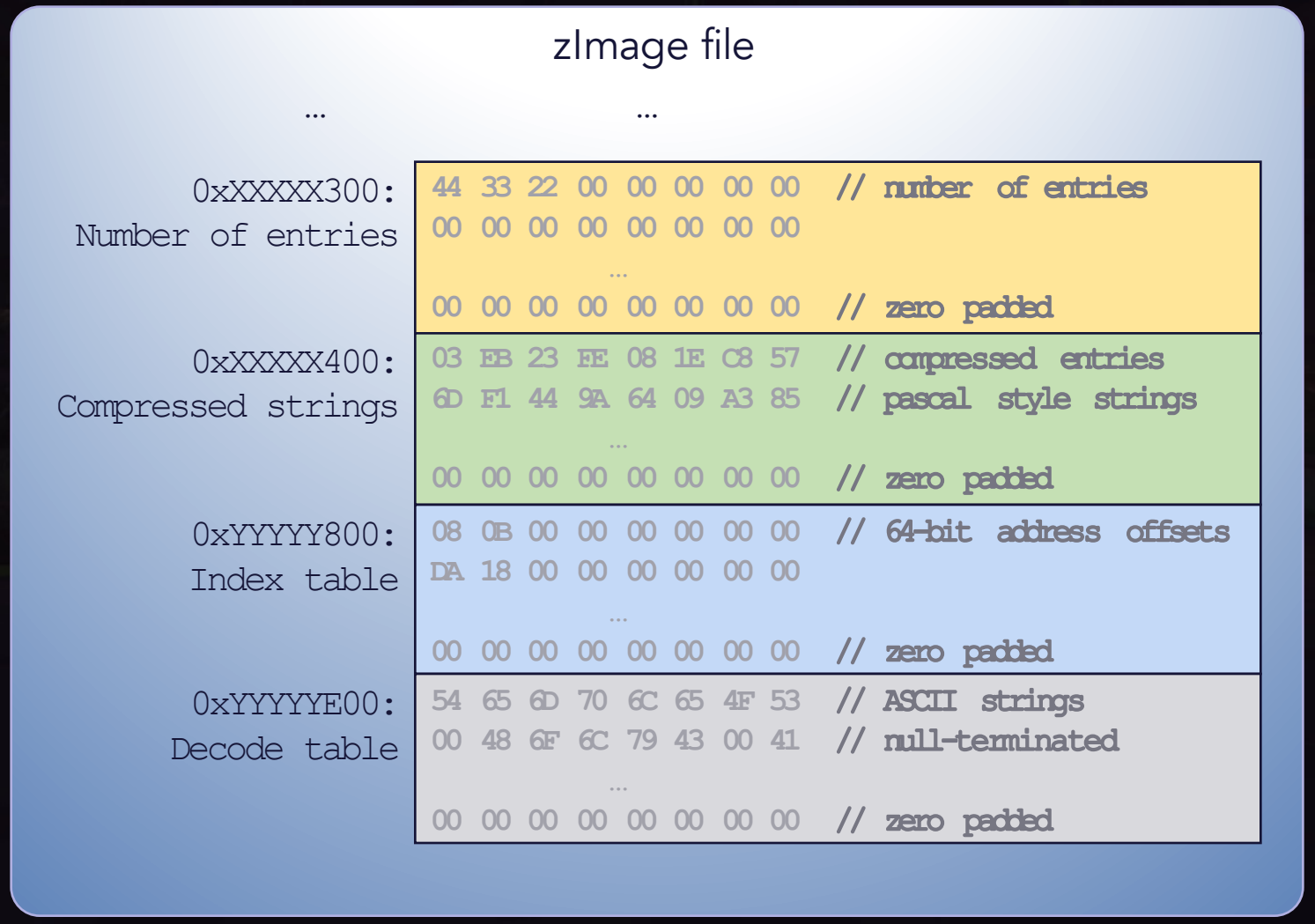
ADR/ADRP/B(+) instructions relative?

Patching Steps

1. Locate KALLSYMS
2. Compile patch
3. Create loader script
4. Link into binary output
5. Write patch to kernel



aarch64 KALLSYMS layout



my_loader.lds

brought to you by:

create-linker-script.py

…and the letter ‘ld’

```
change_memory_common = 0x17cf8;
printk = 0xfe20c;

        SECTIONS
        {
        . = 0x521624;
.text BLOCK(1) : { *(.text) }

            . = 0x521700;
            .data BLOCK(0x10) : {
                *(.data)
            }
            .got BLOCK(0x10) : {
                *(.got)
                gotcheck = .;
                QUAD(0)
            }
            .bss BLOCK (0x10) : {
                *(.bss)
            }
            .dataend : {}
            datastart = ADDR(.data);
            datasize = ADDR(.dataend) - ADDR(.data) + 0x0;
            gotstart = ADDR(.got);
            gotend = ADDR(.got) + SIZEOF(.got);


            . = 0x521730;
            .rodata BLOCK(0x10) : { *(.rodata) }
            . = 0x5217b4;
.text.after.urandom_read BLOCK(1) : { *(.text.after.urandom_read) }
}

```

This bug won't patch itself…

```
597  /* Data structure for the WLAN parts (802.11 cores) of the b43 chip. */
598  struct b43_wl {
599      /* Pointer to the active wireless device on this chip */
600      struct b43_wldev *current_dev;
601      /* Pointer to the ieee80211 hardware data structure */
602      struct ieee80211_hw *hw;
603
604      /* .... SNIP .... */
605
606      /* filter flags */
607      unsigned int filter_flags;
608      /* Stats about the wireless interface */
609      struct ieee80211_low_level_stats ieee_stats;
610
611  #ifdef CONFIG_B43_HWRNG
612      struct hwrng rng;
613      bool rng_initialized;
614      char rng_name[30 + 1];
615  #endif /* CONFIG_B43_HWRNG */
616
617      /* List of all wireless devices on this chip */
618      struct list_head devlist;
619      u8 nr_devs;
620
621      bool radiotap_enabled;
622      bool radio_enabled;
623
624      /* .... SNIP .... */
625
```

Offset??

BATTELLE

QEMU

Full-system emulator

ARM & ARM64 compatible

```
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
 vda: unknown partition table
mousedev: PS/2 mouse device common for all mice
TCP: cubic registered
NET: Registered protocol family 17
NET: Registered protocol family 15
registered taskstats version 1
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
EXT4-fs (vda): couldn't mount as ext3 due to feature incompatibilities
EXT4-fs (vda): mounting ext2 file system using the ext4 subsystem
EXT4-fs (vda): mounted filesystem without journal. Opts: (null)
VFS: Mounted root (ext2 filesystem) readonly on device 254:0.
devtmpfs: mounted
Freeing unused kernel memory: 148K (ffffffc0004b5000 - ffffffc0004da000)
EXT4-fs (vda): warning: mounting unchecked fs, running e2fsck is recommended
EXT4-fs (vda): re-mounted. Opts: (null)
devpts: called with bogus options
Starting logging: OK
Initializing random number generator... random: dd urandom read with 3 bits of entropy available
urandom_read (51): ffffffc005949c00 4bc010 200 ffffffc0059abec8
done.
Starting network: NET: Registered protocol family 10
OK

Welcome to Buildroot
buildroot login:
```


DEMO

What's Next?

Additional architectures

Current: ARM64

Future: ARM, X86_64, MIPS

Additional binary types

Port to inspect ELF file exports

Expand kernel .text sections for our patches

How Did This Happen?

"RUNTIME KERNEL KMEM PATCHING" by Silvio Cesare
http://www.ouah.org/runtime-kernel-kmem-patching.txt

Phrack Issue #50: "Abuse of the Linux Kernel for Fun and Profit" by halflife
http://phrack.org/issues/50/5.html

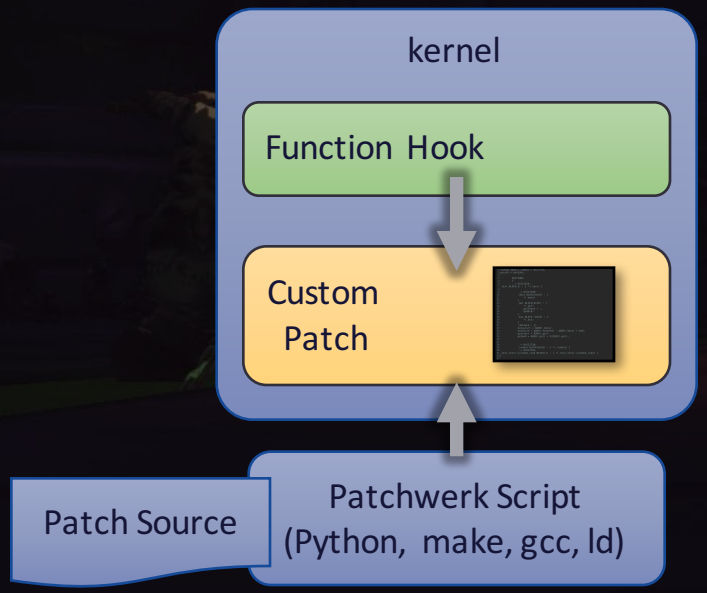
"(nearly) Complete Linux Loadable Kernel Modules -the definitive guide for hackers, virus coders and system administrators-" by pragmatic / THC
http://www.ouah.org/LKM_HACKING.html

Phrack Issue #58: "Linux on-the-fly kernel patching without LKM" by devik & sd
http://phrack.org/issues/58/7.html

"Effectively bypassing kptr_restrict on Android" by laginimaineb
http://bits-please.blogspot.com/2015/08/effectively-bypassing-kptrrestrict-on.html

Patchwerk: Kernel Patching for Fun and Profit

Modify kernel binaries without needing OEM source



Parker Wiksell
@pwiksell
wiksellp@battelle.org

Jewell Seay
seayj@battelle.org

Source code & Install:
https://github.com/battelle/patchwerk

Tutorials:
Patchwerk README +
sample patches

Questions?