SINE NOMINE ASSOCIATES

Using virtualization to debug the OpenAFS Linux kernel module

Cheyenne Wills
(cwills@sinenomine.net)
Sine Nomine Associates
2022 AFS Technologies Workshop

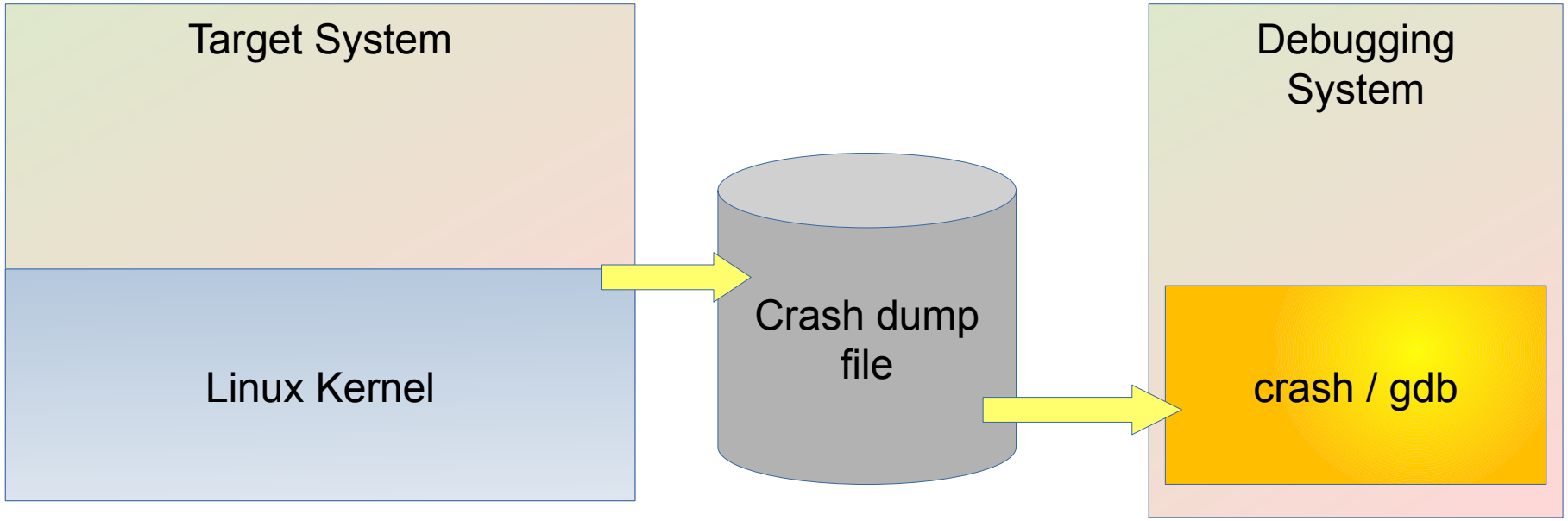
Overview

- Linux kernel debugging methods
- Virtualization
- Preparing a guest and the debugging environment
- Virtualization engines
- Debugging session
- Scripts
- Gotcha’s
- Questions?

SINE NOMINE ASSOCIATES

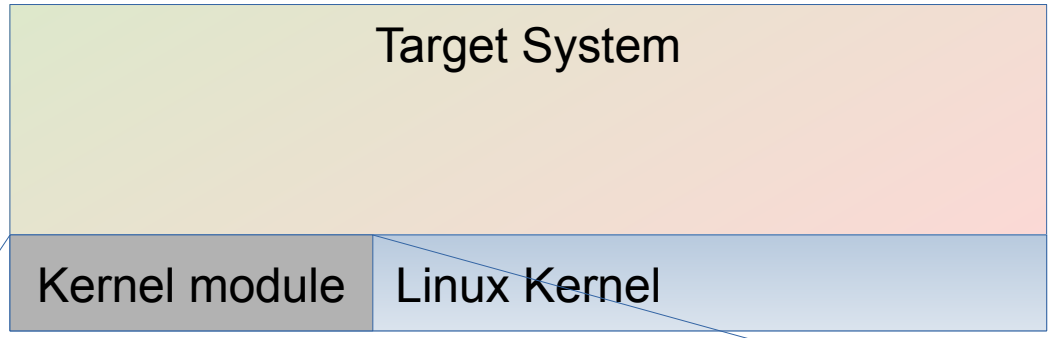
Different debugging methods

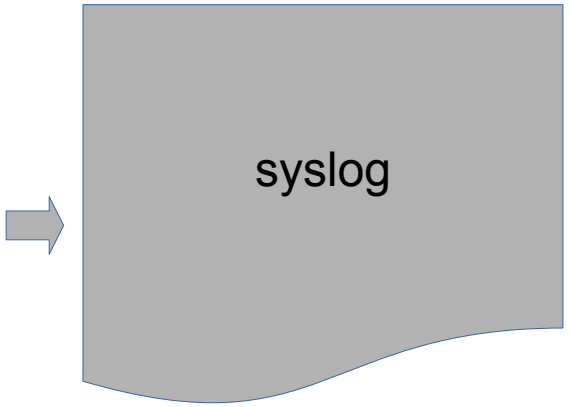
- Postmortem (crash)



Different debugging methods

Using printk...



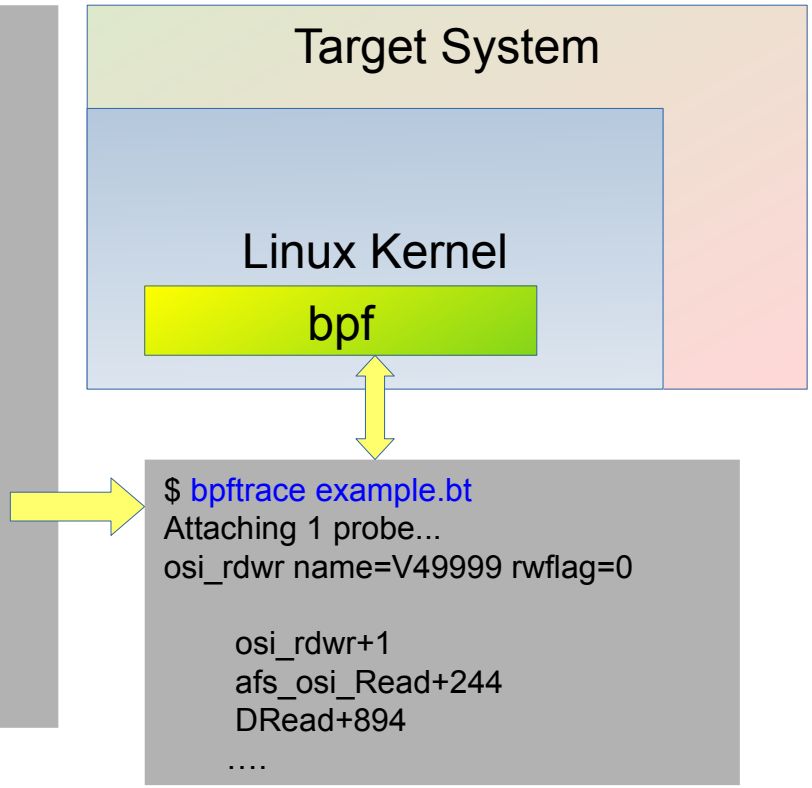



```
if (!code && !data.found) {
   printk("afs: get_name(%s, 0x%08x/%d/%d.%d): not found\n",
          parent->d_name.name ?
                (char *)parent->d_name.name : "?",
          data.fid.Cell,      data.fid.Fid.Volume,
          data.fid.Fid.Vnode, data.fid.Fid.Unique);
   code = ENOENT;
}
```

Different debugging methods

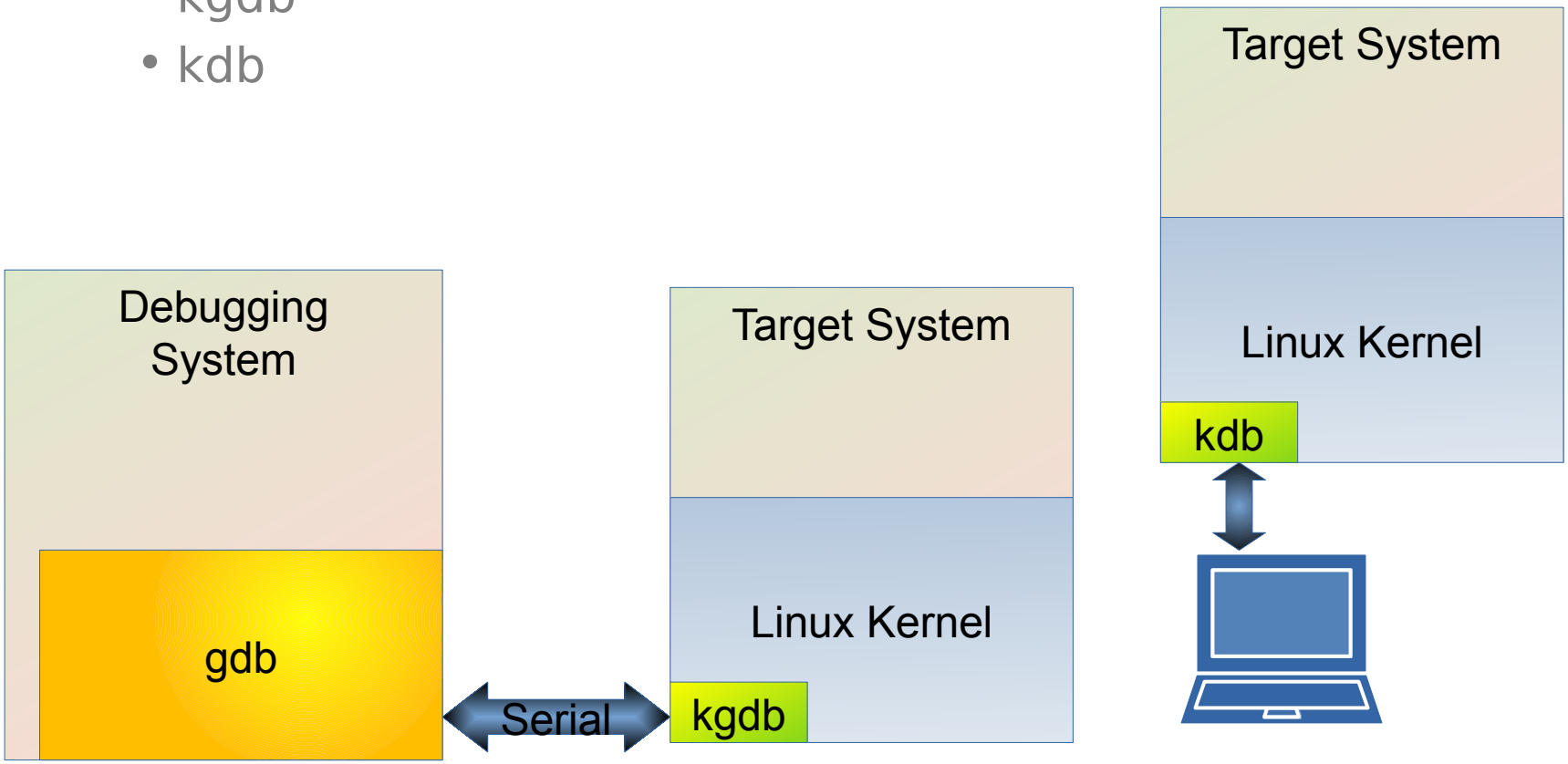
- Systemtap, bpftrace

```
#include <linux/path.h>
#include <linux/dcache.h>
#include <linux/fs.h>
struct osifile {
   int size;
   struct file *filp;
   int offset;
   int (*proc)(void);
   char *rock;
};
kprobe:osi_rdwr
{
    printf("osi_rdwr name=%s rwflag=%d\n",
    str(((struct osifile *)arg0)->filp->f_path.dentry-
>d_name.name), arg2);
    printf("%s\n",kstack);
}
```



Different debugging methods

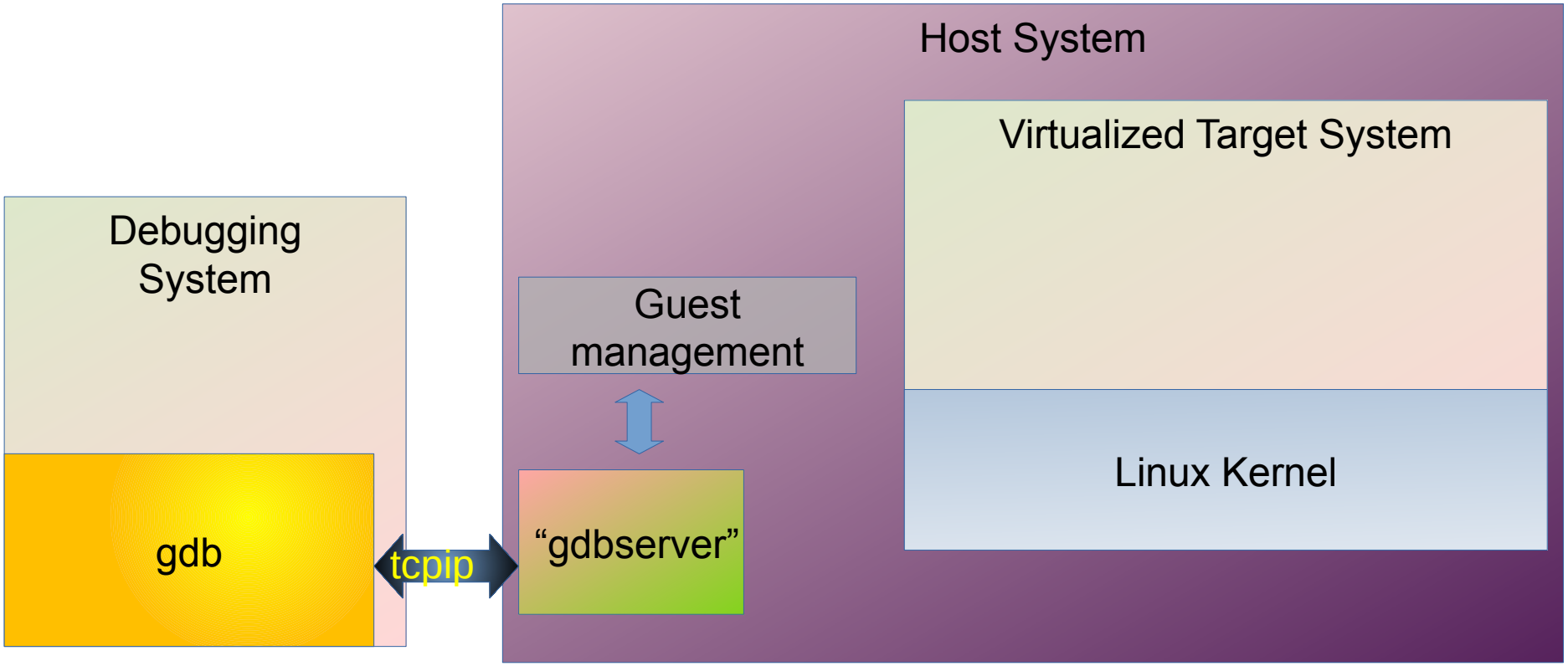
- kgdb
- kdb



SINE NOMINE ASSOCIATES

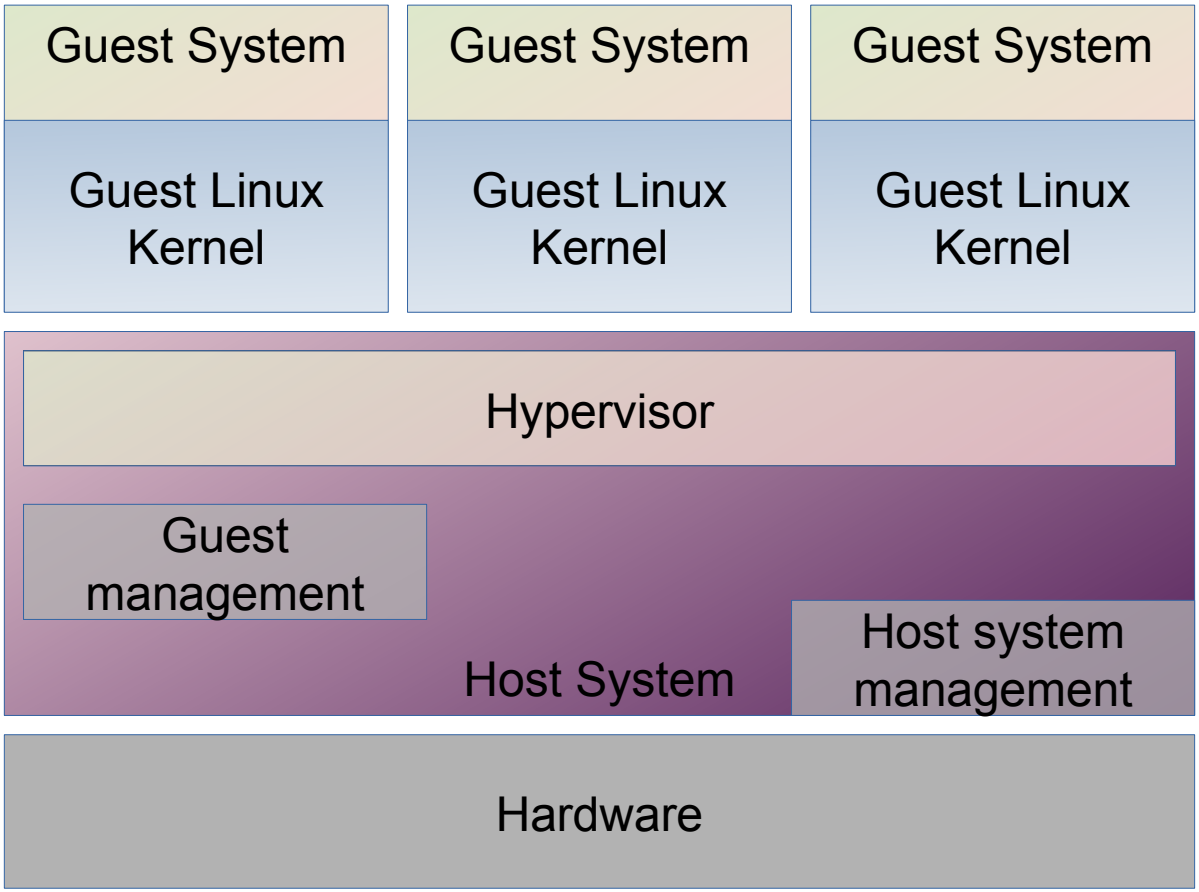
Different debugging methods

- Virtulizatized system



Virtualization

SINE NOMINE ASSOCIATES

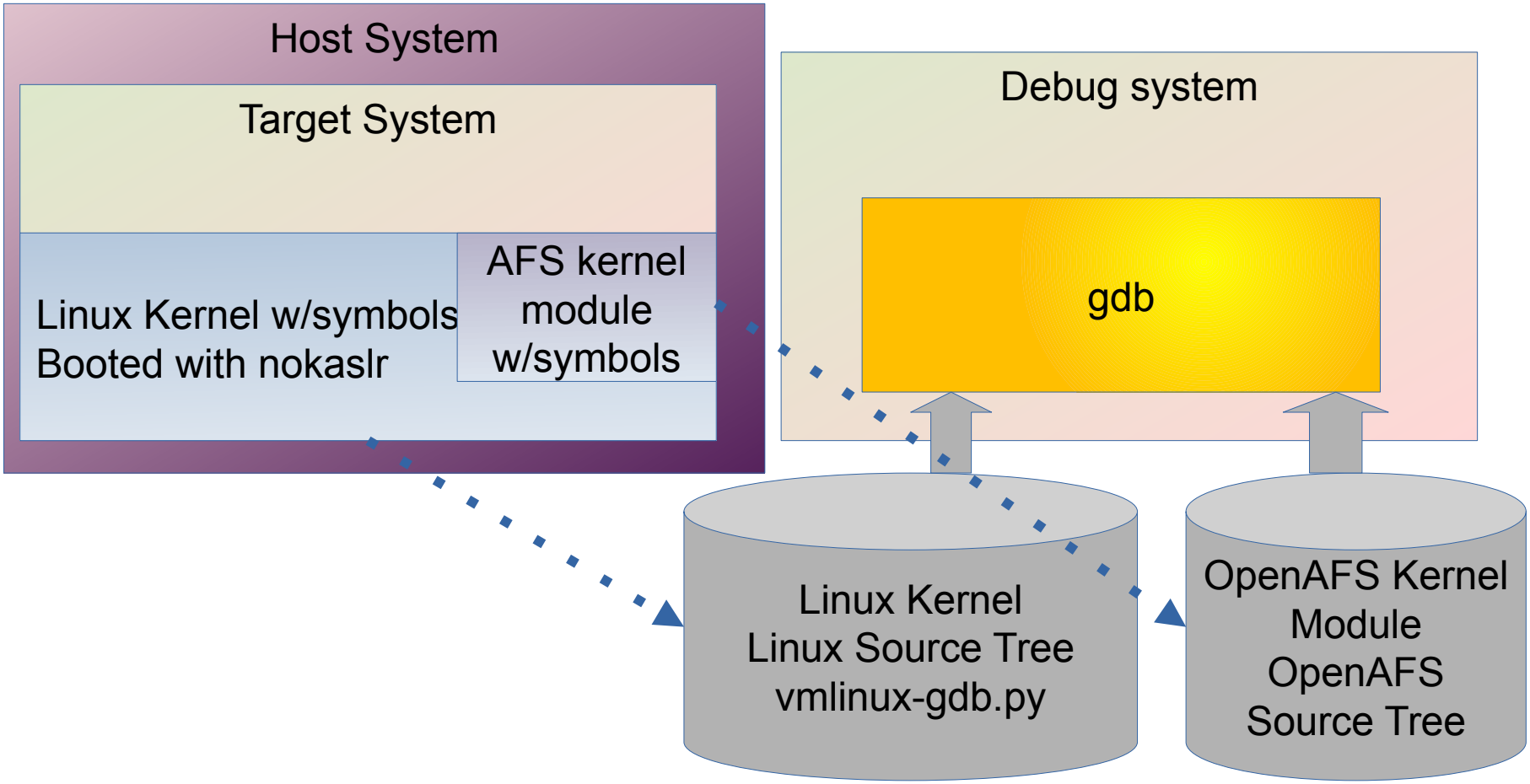


Preparation

SINE NOMINE ASSOCIATES

- Target system
  - Kernel with debugging symbols
  - openAFS kernel module with debugging symbols
  - Booted with nokaslr
- Debugging system
  - Copy of the linux kernel file
  - Linux source tree matching the kernel build
  - Copy of the openAFS kernel module
  - openAFS source tree matching the kernel module build
  - Access to vmlinux-gdb.py and its associated files

Preparation



Preparation

SINE NOMINE ASSOCIATES

- Host system
  - Set up gdb stub
  - Ensure that the debugging system can access the TCP/IP port of the gdb stub.

Preparation Qemu (KVM)

SINE NOMINE ASSOCIATES

- Parameter to the qemu module
  - qemu -gdb tcp::1234
  - Libvirt – virsh edit {domain}

```
<domain type='kvm' id='26'
  xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
...
<qemu:commandline>
    <qemu:arg value='-gdb'/>
    <qemu:arg value='tcp::1234'/>
  </qemu:commandline>
...
```

SINE NOMINE ASSOCIATES

Preparation VMware

Edit the guest's .vmx file

```
# 64 bit guest
debugStub.listen.guest64 = “TRUE”
debugStub.port.guest64 = “nnnn” (default 8864)
# 32 bit guest
debugStub.listen.guest32 = “TRUE”
debugStub.port.guest32 = “nnnn” (default 8832)
# Use int 3 break points
debugStub.hideBreakPoints = “FALSE”
# Use hardware breakpoints
debugStub.hideBreakPoints = “TRUE”
```

Preparation Xen

gdbsx Command line utility

```
$ xl list
Name                                        ID   Mem VCPUs      State
Time(s)
Domain-0                                     0  4091     2     r-----   19904.6
m-afs00                                     26  2032     2     -b----    2394.6
$ gdbsx -a 26 64 1234
```

Starting the debugging session

SINE NOMINE ASSOCIATES

- On the host system, start the guest system with appropriate configurations set.
- On the debugging system
  - start gdb pointing to the linux kernel
  - connect to the remote gdb stub

```
$ gdb vmlinux
(gdb) target remote {host-system}:{port#}
...
(gdb)
```

Fedora36 target

SINE NOMINE ASSOCIATES

```
$ cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt2)/vmlinuz-5.17.13-300.fc36.x86_64 root=UUID=a1642000-8f1b-4618-9a41-
1aceaad499cb ro  console=tty0 rd_NO_PLYMOUTH console=ttyS0,115200 selinux=0 nokaslr

$ rpm -qa|grep kernel
kernel-core-5.17.13-300.fc36.x86_64
kernel-modules-5.17.13-300.fc36.x86_64
kernel-5.17.13-300.fc36.x86_64
kernel-headers-5.17.11-300.fc36.x86_64
kernel-devel-5.17.13-300.fc36.x86_64
kernel-srpm-macros-1.0-14.fc36.noarch
kernel-tools-libs-5.17.11-300.fc36.x86_64
kernel-tools-5.17.11-300.fc36.x86_64

$ modinfo openafs
filename:       /lib/modules/5.17.13-300.fc36.x86_64/extra/openafs/openafs.ko
license:        http://www.openafs.org/dl/license10.html
rhelversion:    9.99
depends:
retpoline:      Y
name:           openafs
vermagic:       5.17.13-300.fc36.x86_64 SMP preempt mod_unload
```

Fedora36 debugging system

SINE NOMINE ASSOCIATES

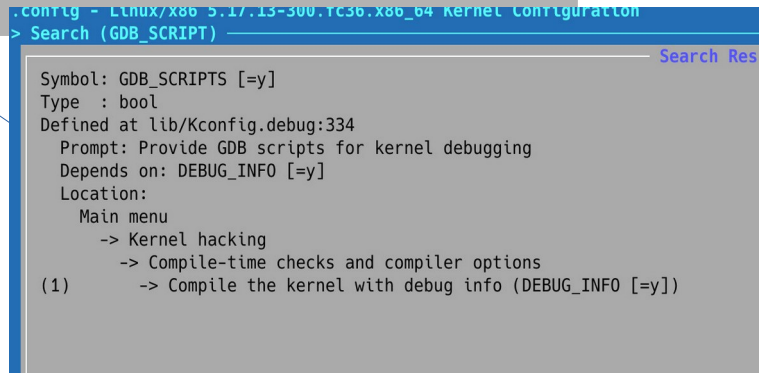
- Install the debuginfo that matches the kernel for the target system
- Install the source tree that matches the kernel for the target system

```
$ rpm -qa|grep kernel
kernel-core-5.17.13-300.fc36.x86_64
kernel-modules-5.17.13-300.fc36.x86_64
kernel-5.17.13-300.fc36.x86_64
kernel-headers-5.17.11-300.fc36.x86_64
kernel-debuginfo-common-x86_64-5.17.13-300.fc36.x86_64
kernel-debuginfo-5.17.13-300.fc36.x86_64
kernel-devel-5.17.13-300.fc36.x86_64
kernel-srpm-macros-1.0-14.fc36.noarch

$ cd /usr/lib/debug/usr/lib/modules/5.17.13-300.fc36.x86_64/

$ ls
internal  kernel  openafs.ko  scripts  vdso  vmlinux  vmlinux-gdb.py
myafscmds.gdb
```

Fedora36 debugging system

- Build vmlinux-gdb

```
$ cd /usr/src/kernels/5.17.13-300.fc36.x86_64
# Update kernel config to set CONFIG_GDB_SCRIPTS=y
$ make menuconfig
$ make scripts_gdb
# Make the scripts/gdb directory available to gdb
$ rsync -a --mkpath scripts/gdb/ /usr/lib/debug/usr/lib/modules/5.17.13-300.fc36.x86_64/scripts/gdb/
$ cd  /usr/lib/debug/usr/lib/modules/5.17.13-300.fc36.x86_64
$ ln -sf scripts/gdb/vmlinux-gdb.py .
```



Fedora36 debugging system

- Ensure the openaAFS source tree matches the source tree used to build the kernel module on the target system

```
$ cd /home/cwills/openafs
$ ls
acinclude.m4   build-tools  config.status  configure-libafs     doc      lib
LICENSE      Makefile-libafs.in  README          src
aclocal.m4     CODING       configure      configure-libafs.ac  include  libafsdep
Makefile     NEWS              README-WINDOWS  tests
amd64_linux26  config.log   configure.ac   CONTRIBUTING         INSTALL  libtool
Makefile.in  NTMakefile          regen.sh

$ ls -l src/libafs/MODLOAD-5.17.13-300.fc36.x86_64-SP/afs_init.c
lrwxrwxrwx 1 cwills cwills 39 Jun 13 12:09 src/libafs/MODLOAD-5.17.13-300.fc36.x86_64-
SP/afs_init.c -> /home/cwills/openafs/src/afs/afs_init.c
```

Fedora36 debugging system

SINE NOMINE ASSOCIATES

```
$ gdb vmlinux
...
(gdb) target remote 10.0.0.2:1234
...
0xffffffff81d1a6cb in native_safe_halt () at ./arch/x86/include/asm/irqflags.h:52
52       }
(gdb) lx-symbols
loading vmlinux
scanning for modules in /usr/lib/debug/usr/lib/modules/5.17.13-300.fc36.x86_64
...
loading @0xffffffffc078d000: /usr/lib/debug/usr/lib/modules/5.17.13-
300.fc36.x86_64/openafs.ko
...
# point to source directory for openafs used to build openafs kernel module
(gdb) dir /home/cwills/openafs
(gdb)
```

Fedora36 debugging system

SINE NOMINE ASSOCIATES

```
(gdb) list osi_rdwr
399      * seek, then read or write to an open inode. addrp points to data in
400      * kernel space.
401      */
402     int
403     osi_rdwr(struct osi_file *osifile, struct uio *uiop, int rw)
404     {
...
(gdb) list 440
435         continue;
436     }
437
438     pos = uiop->uio_offset;
439     if (rw == UIO_READ)
440         code = afs_file_read(filp, iov->iov_base, count, &pos);
441     else
442         code = afs_file_write(filp, iov->iov_base, count, &pos);
443
444     if (code < 0) {
(gdb) break 440
Breakpoint 1 at 0xffffffffc07f8a1c: file .../src/libafs/MODLOAD-5.17.13-300.fc36.x86_64-SP/osi_file.c, line 444.
(gdb) continue
Continuing.
```

Fedora36 debugging system

SINE NOMINE ASSOCIATES

```
Breakpoint 1, osi_rdwr (osifile=osifile@entry=0xffff888016fdf940,
uiop=uiop@entry=0xffffc900008939e8, rw=rw@entry=0) at .../src/libafs/MODLOAD-5.17.13-
300.fc36.x86_64-SP/osi_file.c:444
444         if (code < 0) {

(gdb) bt
#0  osi_rdwr (osifile=osifile@entry=0xffff888016fdf940,
uiop=uiop@entry=0xffffc900008939e8, rw=rw@entry=0) at .../src/libafs/MODLOAD-5.17.13-
300.fc36.x86_64-SP/osi_file.c:444
#1  0xffffffffc07f8ba4 in afs_osi_Read (afile=0xffff888016fdf940, offset=<optimized out>,
aptr=<optimized out>, asize=2048) at
.../src/libafs/MODLOAD-5.17.13-300.fc36.x86_64-SP/osi_file.c:287
...

(gdb) print *iov
$3 = {iov_base = 0xffffc90000833000, iov_len = 2048}
```

Fedora36 debugging system

SINE NOMINE ASSOCIATES

Using a gdb script

```
(gdb) source myafscmds.gdb
(gdb) walkdcache
00000 inode: 9272933 fid: (0 0 0 0)
...
01555 inode: 9474424 fid: (2 536871327 2 1623)
01556 inode: 9474425 fid: (2 536871327 1 1)
01557 inode: 9474426 fid: (2 536870916 278 143)
01558 inode: 9474427 fid: (2 536870916 1 1)
01559 inode: 9474428 fid: (1 536870915 1 1)
01560 inode: 9474429 fid: (1 536870912 1 1)
01561 inode: 9474430 fid: (0 1 1 1)
(gdb)
```

vmlinux-gdb

SINE NOMINE ASSOCIATES

- Part of the Linux source tree, located in {linux_src}/scripts/gdb
- Python extension to gdb
- May need to be configured and built (depending on the Linux distro)
- Provides new commands and functions that can be used within gdb
- Help available via gdb help command
- {linux_src}/Documentation/dev-tools/gdb-kernel-debugging.rst

vmlinux-gdb commands

SINE NOMINE ASSOCIATES

- Lx-clk-summary
- lx-cpus
- lx-device-list-tree
- lx-genpd-summary
- lx-list-check
- lx-ps
- lx-version
- lx-cmdline
- lx-device-list-bus
- lx-dmesg
- lx-iomem
- lx-lsmod
- lx-symbols
- lx-configdump
- lx-device-list-class
- lx-fdtdump
- lx-ioports
- lx-mounts
- lx-timerlist

vmlinux-gdb functions

SINE NOMINE ASSOCIATES

- container_of
- lx_clk_core_lookup
- lx_current_func
- lx_device_find_by_bus_name
- lx_device_find_by_class_name
- lx_module
- lx_rb_first
- lx_rb_last
- lx_rb_next
- lx_rb_prev
- lx_task_by_pid_func
- lx_thread_info_by_pid_func
- lx_thread_info_func
- per_cpu

```
(gdb) help function container_of
Return pointer to containing data structure.

$container_of(PTR, "TYPE", "ELEMENT"): Given PTR, return a pointer to the
data structure of the type TYPE in which PTR is the address of ELEMENT.
Note that TYPE and ELEMENT have to be quoted as strings.
(gdb)
```

Example scripts

SINE NOMINE ASSOCIATES

```
define walkdcache
  set $i = 0
  while ($i < afs_cacheFiles)
    set $tdc = afs_indexTable[$i]
    if ($tdc != 0)
       printf "%05d ", $i
       printf "inode: %d ", $tdc.f.inode.ufs.fh.i32.ino
       printf "fid: (%d %d %d %d) ", $tdc.f.fid.Cell, \
                                     $tdc.f.fid.Fid.Volume, \
                                     $tdc.f.fid.Fid.Vnode,  \
                                     $tdc.f.fid.Fid.Unique
       printf "\n"
    end
    set $i = $i + 1
  end
```

Example scripts

SINE NOMINE ASSOCIATES

```
define showvcaches
  set $next = VLRU.next
  while ($next != VLRU.prev)
    set $v = *$container_of($next, "struct vcache", "vlruq")
    set $mvstat = $v.mvstat
    printf "mvstat: %d ", $mvstat
    set $fid = $v.f.fid
    printf "cell: %4d ", $fid.Cell
    printf "vol : %10d ", $fid.Fid.Volume
    printf "vnode : %8d ", $fid.Fid.Vnode
    printf "states : %08x ", $v.f.states
    printf "\n"
    set $next = $next.next
  end
end
```

Example script

```
(gdb) showvcaches
mvstat: 0 cell:    2 vol :  536880536 vnode :      3 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    395 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    375 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    283 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    337 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    299 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :     53 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    275 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :    253 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :  13528 states : 00000405
mvstat: 0 cell:    2 vol :  536880536 vnode :     73 states : 00000405
...
(gdb)
```

Gotcha's

SINE NOMINE ASSOCIATES

- Running a kernel with kalsr, gdb will not be able to match symbol locations with the running kernel.
- Source trees for Linux and OpenAFS must match the linux kernel and the openafs kernel module respectively.
- Absolute symlinks are used in the openafs build process when building the kernel module. The source tree needs to be in the same absolute location to avoid broken symlinks.
- Possible Kernel watchdog timeouts in the guest if sitting at a (gdb) prompt for too long (depending on the kernel configuration).
- The virtualization engine may not support all of the remote debugging facilities (e.g. xen's gdbsx doesn't support hardware watchpoints).
- Debugging multiple CPU guest may not be supported by some virtualization engines.

Questions?