DE LA RECHERCHE À L'INDUSTRIE

BOF Session: using Trinity Fuzzer for "monkey testing"

Philippe DENIEL (philippe.deniel@cea.fr)

www.cea.fr

Random is beautiful

The “Fly in a Bottle” method

- When trapped in a bottle, a fly chooses a random direction to go...
- ... and retries until it finds the bottleneck and exits

So does Trinity Fuzzer

- Basically, a “fuzz testing” engine
  - Use sycalls in a random way, with random arguments
  - Randomly, you can find such a dumb situation that something crashes
  - Do post-mortem analysis to find the bug

Trinity in smarter than the fly

Most of fuzz tests will exit with EINVAL

- Really dumb arguments are used
- Syscall wants a valid Fd and the provided argument is no file descriptor
- EINVAL exists slow down the fuzz test, increasing to time to find some interesting

Smart Fuzzing

- Trinity Fuzzer knows how to do “really evil call to syscalls”
  - Arguments are not purely random
  - on startup, Trinity creates a list of Fds (pipes, sysfs, procfs, /dev, sockets, ...)
  - when a syscall needs an fd, it gets passed one of these at random
- Every syscall has its arguments annotated, and where possible it tries to provide something at least semi-sensible.
  - "Length" arguments get passed one of a whole bunch of potentially interesting values
  - Fds are shared between multiple threads, which causes havoc sometimes.
  - Mmap buffers are fed to subsequent syscalls, sometimes with funny returns
  - And many other really nasty features...

Trinity for file system developers

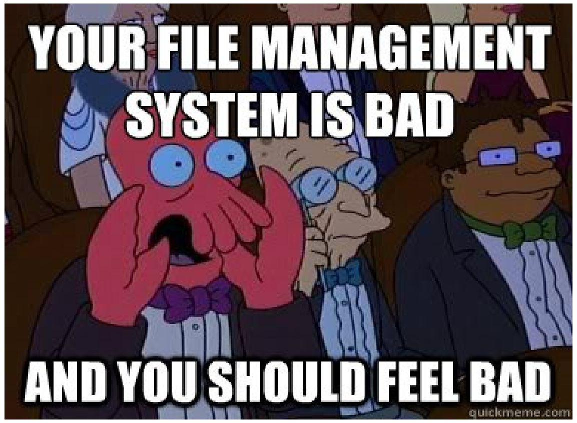
Trinity Fuzzer is really good at locating bugs in FS

- We run it against different versions of Lustre and found a bug in 5 minutes
- We run it to make the implementation of 9p in NFS-Ganesha “bullet proof”
- We have plans to run it against NFSv4.1

Difficulties

- When used to test a distributed file system, things become trickier
  - A client+server pair is involved, both can produce the bug
  - You can't test a server or a client in a standalone way, you test their association
  - You need to have a close watch at the generated report to locate the bug
- Trinity does test every available syscalls
  - When testing file system, you have to retrict it to File System related operations

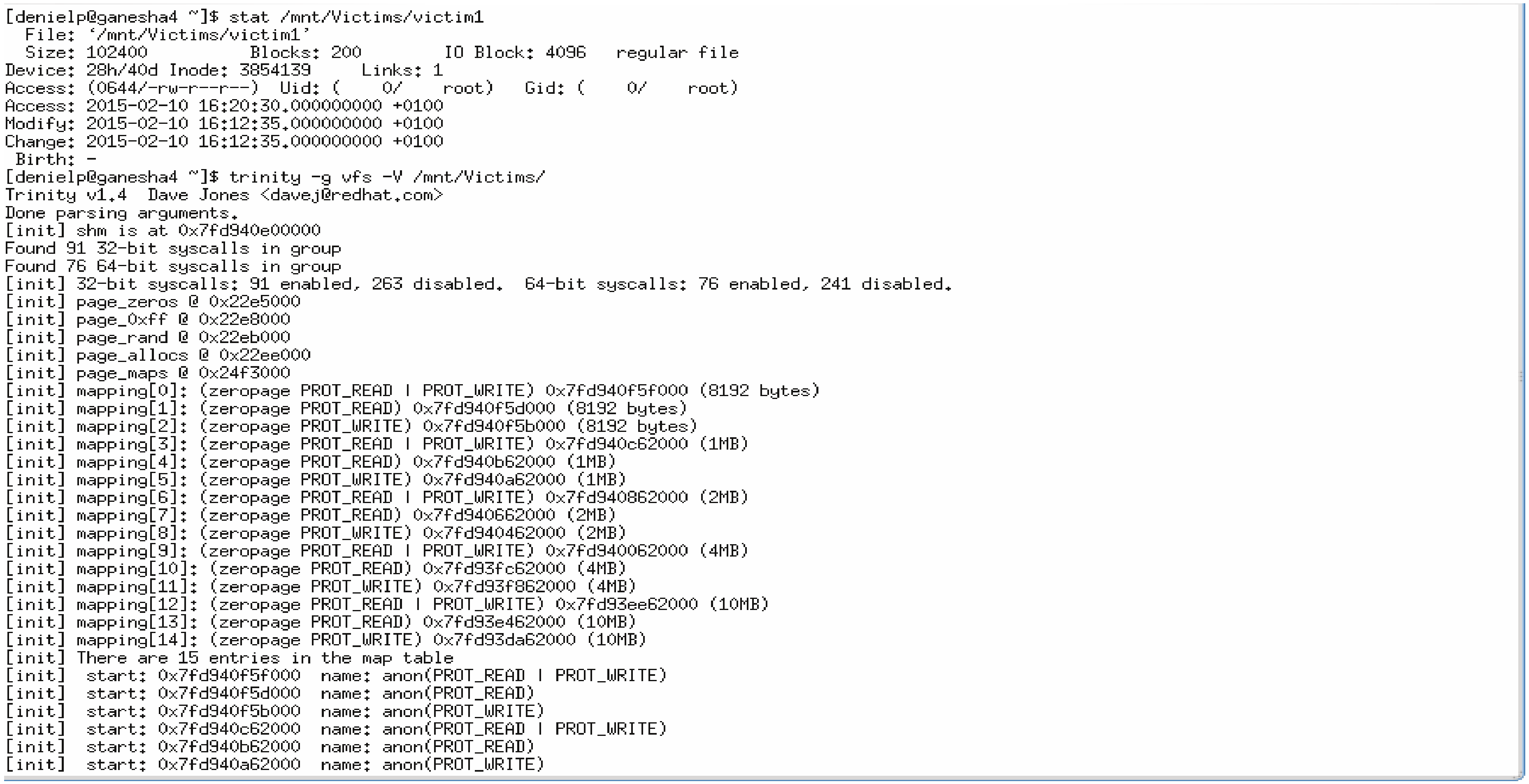
Finding bugs becomes easier... and it can be depressing..



Trinity in action (1/2)

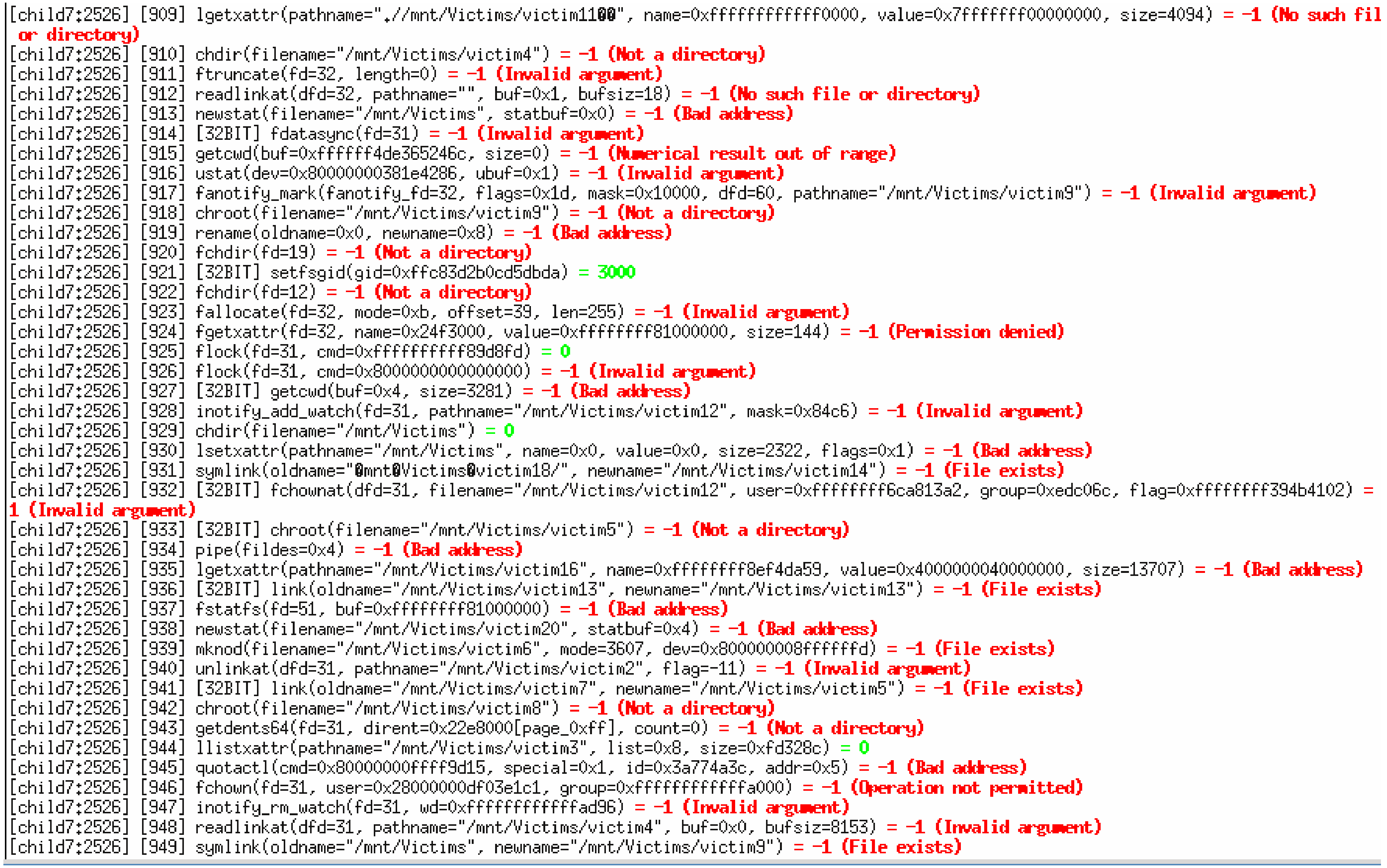
Prepare the environment to run trinity

- Create a dedicated directory in the FS to be tested
- Create a few files (/dev/urandom can be a good seed for test files)
- Run Trinity with on this “victim files”
  - Trinity has a “syscalls group” dedicated to VFS syscalls



```
[denielp@ganesha4 ~]$ stat /mnt/Victims/victim1
  File: '/mnt/Victims/victim1'
  Size: 102400          Blocks: 200        IO Block: 4096   regular file
Device: 28h/40d Inode: 3854139     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2015-02-10 16:20:30.000000000 +0100
Modify: 2015-02-10 16:12:35.000000000 +0100
Change: 2015-02-10 16:12:35.000000000 +0100
 Birth: -
[denielp@ganesha4 ~]$ trinity -g vfs -V /mnt/Victims/
Trinity v1.4  Dave Jones <davej@redhat.com>
Done parsing arguments.
[init] shm is at 0x7fd940e00000
Found 91 32-bit syscalls in group
Found 76 64-bit syscalls in group
[init] 32-bit syscalls: 91 enabled, 263 disabled.  64-bit syscalls: 76 enabled, 241 disabled.
[init] page_zeros @ 0x22e5000
[init] page_0xff @ 0x22e8000
[init] page_rand @ 0x22eb000
[init] page_allocs @ 0x22ee000
[init] page_maps @ 0x24f3000
[init] mapping[0]: (zeropage PROT_READ | PROT_WRITE) 0x7fd940f5f000 (8192 bytes)
[init] mapping[1]: (zeropage PROT_READ) 0x7fd940f5d000 (8192 bytes)
[init] mapping[2]: (zeropage PROT_WRITE) 0x7fd940f5b000 (8192 bytes)
[init] mapping[3]: (zeropage PROT_READ | PROT_WRITE) 0x7fd940c62000 (1MB)
[init] mapping[4]: (zeropage PROT_READ) 0x7fd940b62000 (1MB)
[init] mapping[5]: (zeropage PROT_WRITE) 0x7fd940a62000 (1MB)
[init] mapping[6]: (zeropage PROT_READ | PROT_WRITE) 0x7fd940862000 (2MB)
[init] mapping[7]: (zeropage PROT_READ) 0x7fd940662000 (2MB)
[init] mapping[8]: (zeropage PROT_WRITE) 0x7fd940462000 (2MB)
[init] mapping[9]: (zeropage PROT_READ | PROT_WRITE) 0x7fd940062000 (4MB)
[init] mapping[10]: (zeropage PROT_READ) 0x7fd93fc62000 (4MB)
[init] mapping[11]: (zeropage PROT_WRITE) 0x7fd93f862000 (4MB)
[init] mapping[12]: (zeropage PROT_READ | PROT_WRITE) 0x7fd93ee62000 (10MB)
[init] mapping[13]: (zeropage PROT_READ) 0x7fd93e462000 (10MB)
[init] mapping[14]: (zeropage PROT_WRITE) 0x7fd93da62000 (10MB)
[init] There are 15 entries in the map table
[init]  start: 0x7fd940f5f000  name: anon(PROT_READ | PROT_WRITE)
[init]  start: 0x7fd940f5d000  name: anon(PROT_READ)
[init]  start: 0x7fd940f5b000  name: anon(PROT_WRITE)
[init]  start: 0x7fd940c62000  name: anon(PROT_READ | PROT_WRITE)
[init]  start: 0x7fd940b62000  name: anon(PROT_READ)
[init]  start: 0x7fd940a62000  name: anon(PROT_WRITE)
```

Trinity in action (2/2)



```
[child7:2526] [909] lgetxattr(pathname=".//mnt/Victims/victim1100", name=0xffffffffffff0000, value=0x7fffffff00000000, size=4094) = -1 (No such fil
 or directory)
[child7:2526] [910] chdir(filename="/mnt/Victims/victim4") = -1 (Not a directory)
[child7:2526] [911] ftruncate(fd=32, length=0) = -1 (Invalid argument)
[child7:2526] [912] readlinkat(dfd=32, pathname="", buf=0x1, bufsiz=18) = -1 (No such file or directory)
[child7:2526] [913] newstat(filename="/mnt/Victims", statbuf=0x0) = -1 (Bad address)
[child7:2526] [914] [32BIT] fdatasync(fd=31) = -1 (Invalid argument)
[child7:2526] [915] getcwd(buf=0xffffff4de365246c, size=0) = -1 (Numerical result out of range)
[child7:2526] [916] ustat(dev=0x80000000381e4286, ubuf=0x1) = -1 (Invalid argument)
[child7:2526] [917] fanotify_mark(fanotify_fd=32, flags=0x1d, mask=0x10000, dfd=60, pathname="/mnt/Victims/victim9") = -1 (Invalid argument)
[child7:2526] [918] chroot(filename="/mnt/Victims/victim9") = -1 (Not a directory)
[child7:2526] [919] rename(oldname=0x0, newname=0x8) = -1 (Bad address)
[child7:2526] [920] fchdir(fd=19) = -1 (Not a directory)
[child7:2526] [921] [32BIT] setfsgid(gid=0xffc83d2b0cd5dbda) = 3000
[child7:2526] [922] fchdir(fd=12) = -1 (Not a directory)
[child7:2526] [923] fallocate(fd=32, mode=0xb, offset=39, len=255) = -1 (Invalid argument)
[child7:2526] [924] fgetxattr(fd=32, name=0x24f3000, value=0xffffffff81000000, size=144) = -1 (Permission denied)
[child7:2526] [925] flock(fd=31, cmd=0xffffffffff89d8fd) = 0
[child7:2526] [926] flock(fd=31, cmd=0x8000000000000000) = -1 (Invalid argument)
[child7:2526] [927] [32BIT] getcwd(buf=0x4, size=3281) = -1 (Bad address)
[child7:2526] [928] inotify_add_watch(fd=31, pathname="/mnt/Victims/victim12", mask=0x84c6) = -1 (Invalid argument)
[child7:2526] [929] chdir(filename="/mnt/Victims") = 0
[child7:2526] [930] lsetxattr(pathname="/mnt/Victims", name=0x0, value=0x0, size=2322, flags=0x1) = -1 (Bad address)
[child7:2526] [931] symlink(oldname="0mnt0Victims0victim18/", newname="/mnt/Victims/victim14") = -1 (File exists)
[child7:2526] [932] [32BIT] fchownat(dfd=31, filename="/mnt/Victims/victim12", user=0xffffffff6ca813a2, group=0xedc06c, flag=0xffffffff394b4102) =
1 (Invalid argument)
[child7:2526] [933] [32BIT] chroot(filename="/mnt/Victims/victim5") = -1 (Not a directory)
[child7:2526] [934] pipe(fildes=0x4) = -1 (Bad address)
[child7:2526] [935] lgetxattr(pathname="/mnt/Victims/victim16", name=0xffffffff8ef4da59, value=0x4000000040000000, size=13707) = -1 (Bad address)
[child7:2526] [936] [32BIT] link(oldname="/mnt/Victims/victim13", newname="/mnt/Victims/victim13") = -1 (File exists)
[child7:2526] [937] fstatfs(fd=51, buf=0xffffffff81000000) = -1 (Bad address)
[child7:2526] [938] newstat(filename="/mnt/Victims/victim20", statbuf=0x4) = -1 (Bad address)
[child7:2526] [939] mknod(filename="/mnt/Victims/victim6", mode=3607, dev=0x800000008ffffffd) = -1 (File exists)
[child7:2526] [940] unlinkat(dfd=31, pathname="/mnt/Victims/victim2", flag=-11) = -1 (Invalid argument)
[child7:2526] [941] [32BIT] link(oldname="/mnt/Victims/victim7", newname="/mnt/Victims/victim5") = -1 (File exists)
[child7:2526] [942] chroot(filename="/mnt/Victims/victim8") = -1 (Not a directory)
[child7:2526] [943] getdents64(fd=31, dirent=0x22e8000[page_0xff], count=0) = -1 (Not a directory)
[child7:2526] [944] llistxattr(pathname="/mnt/Victims/victim3", list=0x8, size=0xfd328c) = 0
[child7:2526] [945] quotactl(cmd=0x80000000ffff9d15, special=0x1, id=0x3a774a3c, addr=0x5) = -1 (Bad address)
[child7:2526] [946] fchown(fd=31, user=0x28000000df03e1c1, group=0xffffffffffffa000) = -1 (Operation not permitted)
[child7:2526] [947] inotify_rm_watch(fd=31, wd=0xffffffffffffad96) = -1 (Invalid argument)
[child7:2526] [948] readlinkat(dfd=31, pathname="/mnt/Victims/victim4", buf=0x0, bufsiz=8153) = -1 (Invalid argument)
[child7:2526] [949] symlink(oldname="/mnt/Victims", newname="/mnt/Victims/victim9") = -1 (File exists)
```

]

Examples of bugs found via Trinity Fuzzer

LU-3732 in Lustre Jira

- https://jira.hpdd.intel.com/browse/LU-3732
- Trinity put the light on the Bug
- A 3-line reproducer could be written to ease debugging

Linux Weekly News speaks about Trinity Fuzzer

- https://lwn.net/Articles/536173/

Getting Trinity Fuzzer

Project URL

- http://codemonkey.org.uk/projects/trinity/
- http://github.com/kernelslacker/trinity

Downloadable tarball

- http://codemonkey.org/uk/trinity/trinity-git-snapshot.tar.xz

We have no share in Trinity and did not took part in its development but we really like it. We hope you'll enjoy it.

enjoy!

Questions ?