pledge(2): mitigating security bugs

Giovanni Bechis
<giovanni@openbsd.org>

pkgsrcCon 2017, London

About Me

- sys admin and developer @SNB
- OpenBSD developer
- Open Source developer in several other projects

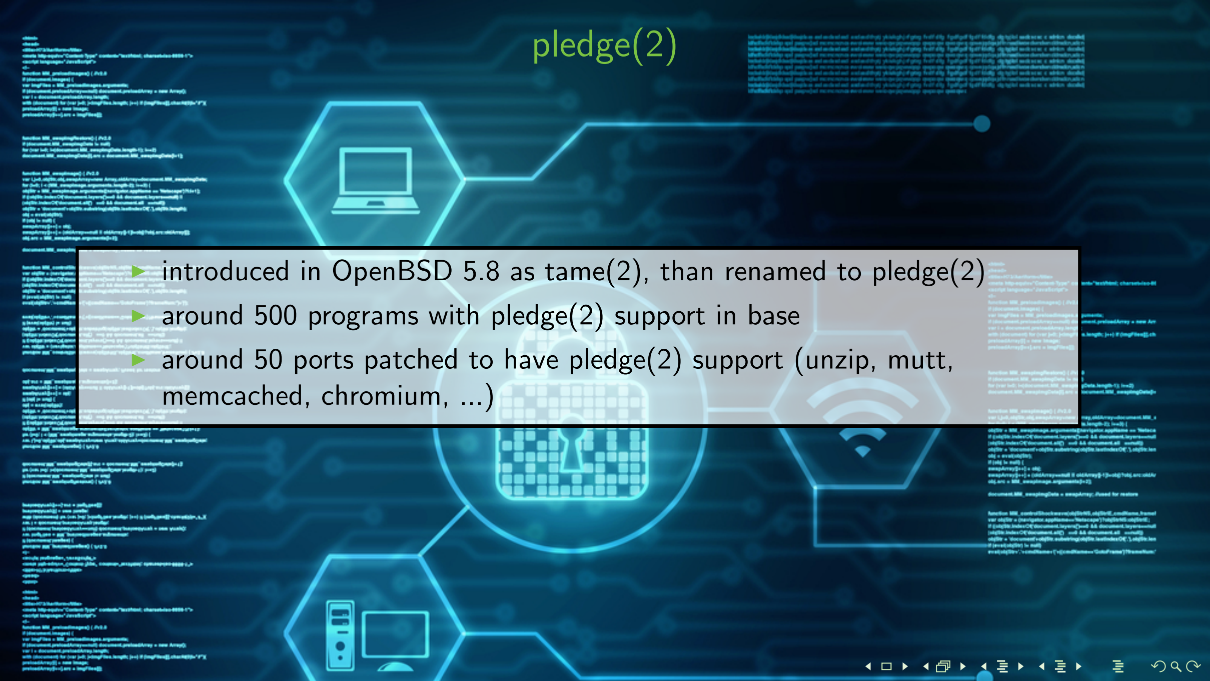
Mitigations

- stack protector
- ASLR
- W^X
- priv-sep and priv-drop

dangerous system calls

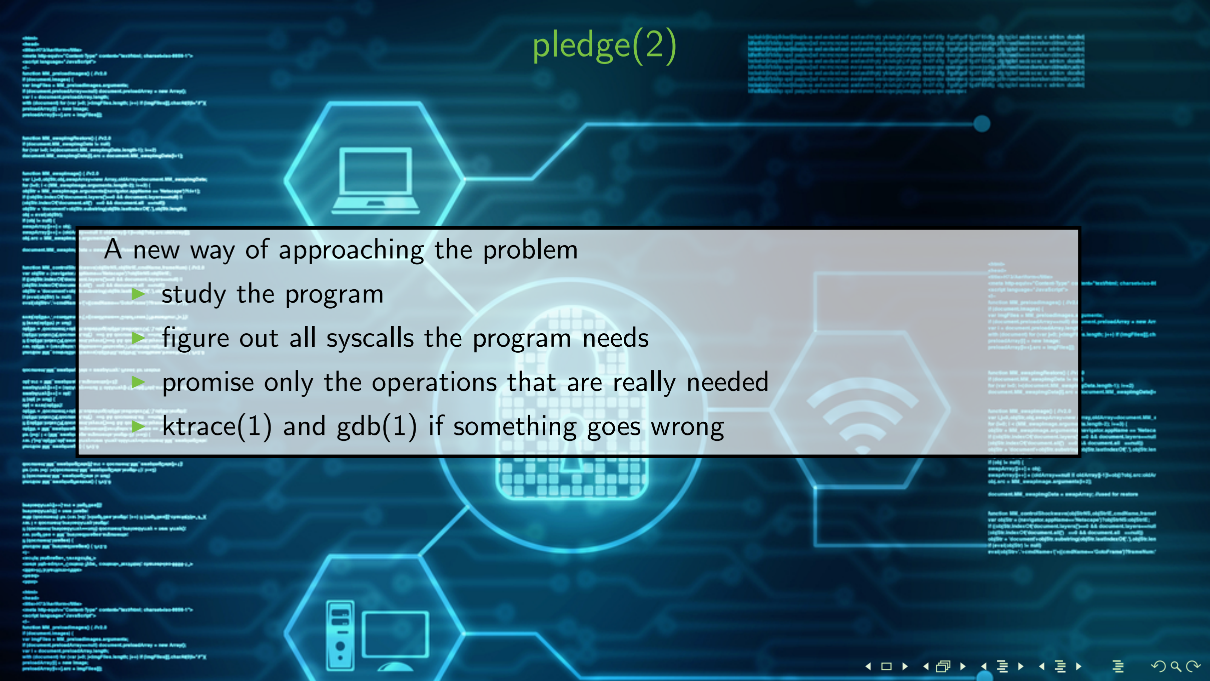
Disable system calls a process should not call

- SE Linux
- seccomp
- Capsicum
- systrace
- pledge(2)



pledge(2)

- introduced in OpenBSD 5.8 as tame(2), than renamed to pledge(2)
- around 500 programs with pledge(2) support in base
- around 50 ports patched to have pledge(2) support (unzip, mutt, memcached, chromium, ...)

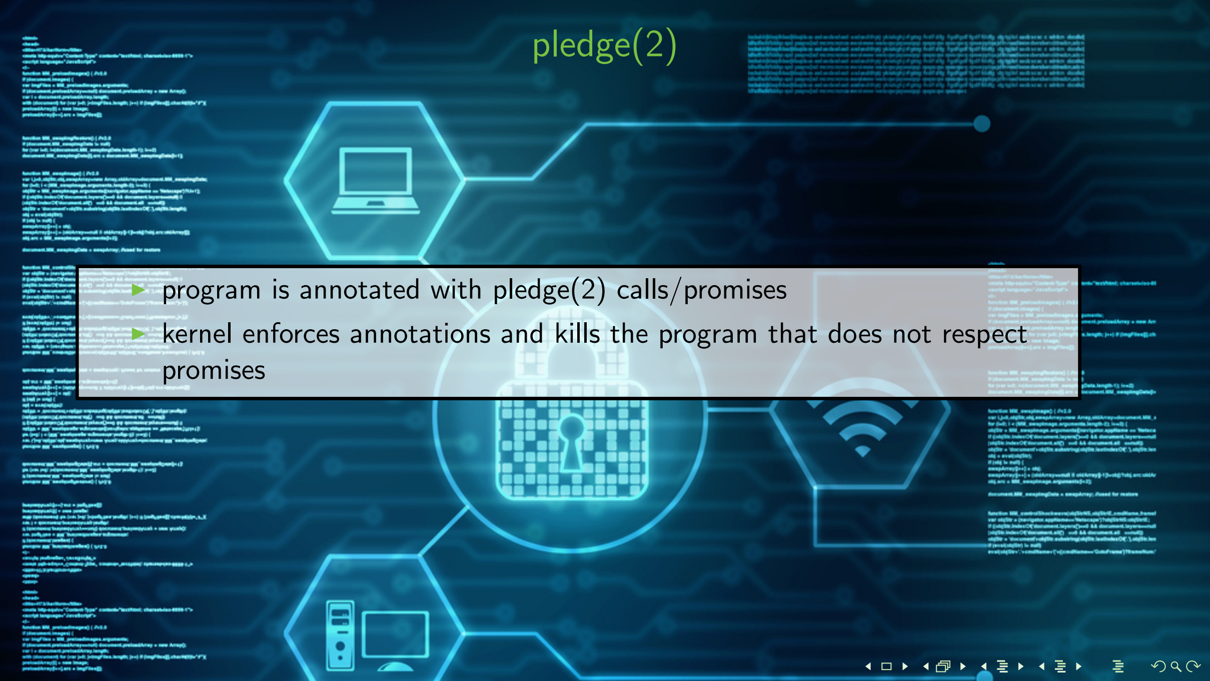


pledge(2)

A new way of approaching the problem

- study the program
- figure out all syscalls the program needs
- promise only the operations that are really needed
- ktrace(1) and gdb(1) if something goes wrong

pledge(2)



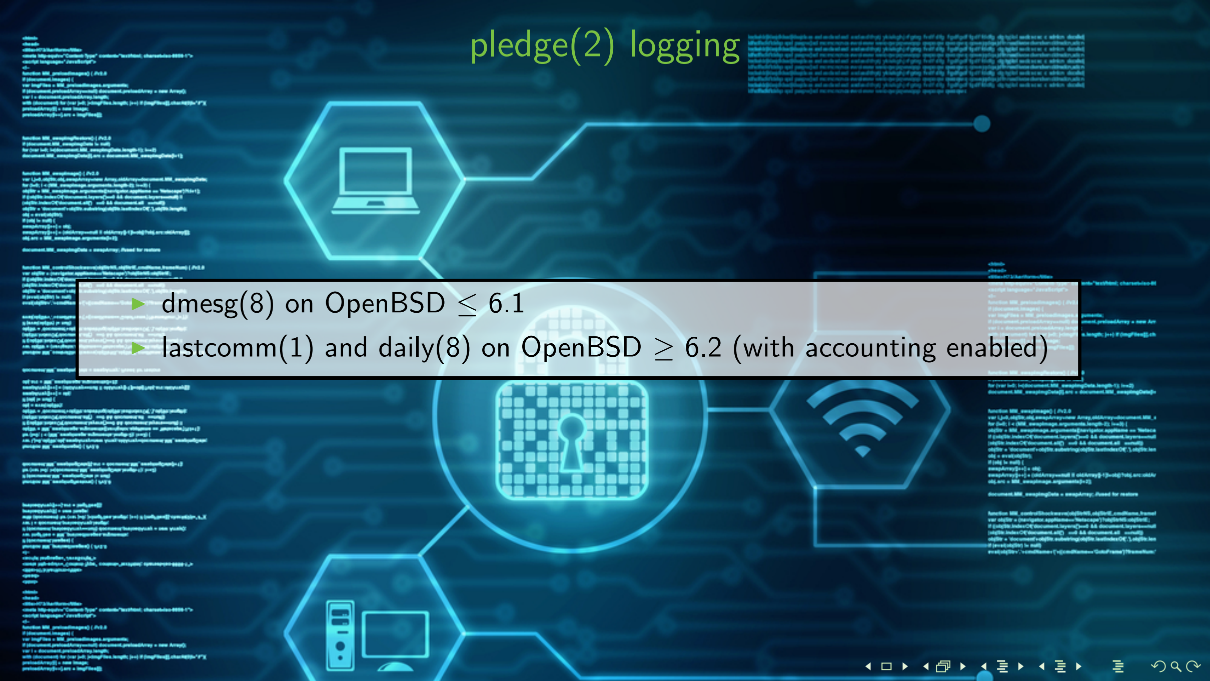
- program is annotated with pledge(2) calls/promises
- kernel enforces annotations and kills the program that does not respect promises

pledge(2) promises

```
#include <unistd.h>

int pledge(const char *promises, const char *paths[]);
```

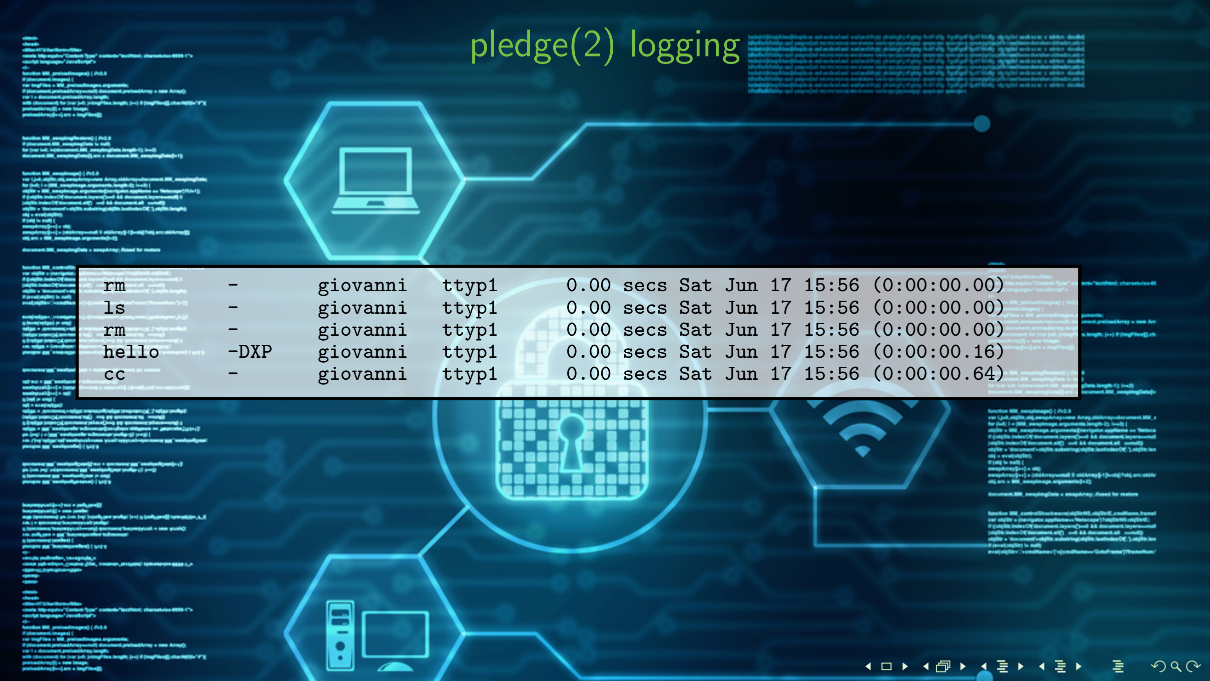
- "": _exit(2)
- stdio: malloc + rw stdio
- rpath, wpath, cpath, tmppath: open files
- fattr: explicit changes to "fd" (chmod & friends)
- unix, inet: open sockets
- dns: dns requests
- route: routing operations
- sendfd: sends file descriptors via sendmsg(2)
- recvfd: receive file descriptors via recvmsg(2)
- getpw: passwd/group file access
- ioctl: small subset of ioctls is permitted
- tty: subset of ioctl for tty operations
- proc: fork(2), vfork(2), kill(2) and other processes related operations
- exec: execve(2) is allowed to create another process which will be unpledged
- settime: allows to set the system time
- pf: allows a subset of ioctl(2) operations on pf(4) device

pledge(2) logging



- dmesg(8) on OpenBSD $\leq$ 6.1
- lastcomm(1) and daily(8) on OpenBSD $\geq$ 6.2 (with accounting enabled)

pledge(2) logging



```
rm        -      giovanni  ttyp1     0.00 secs Sat Jun 17 15:56 (0:00:00.00)
ls        -      giovanni  ttyp1     0.00 secs Sat Jun 17 15:56 (0:00:00.00)
rm        -      giovanni  ttyp1     0.00 secs Sat Jun 17 15:56 (0:00:00.00)
hello     -DXP   giovanni  ttyp1     0.00 secs Sat Jun 17 15:56 (0:00:00.16)
cc        -      giovanni  ttyp1     0.00 secs Sat Jun 17 15:56 (0:00:00.64)
```

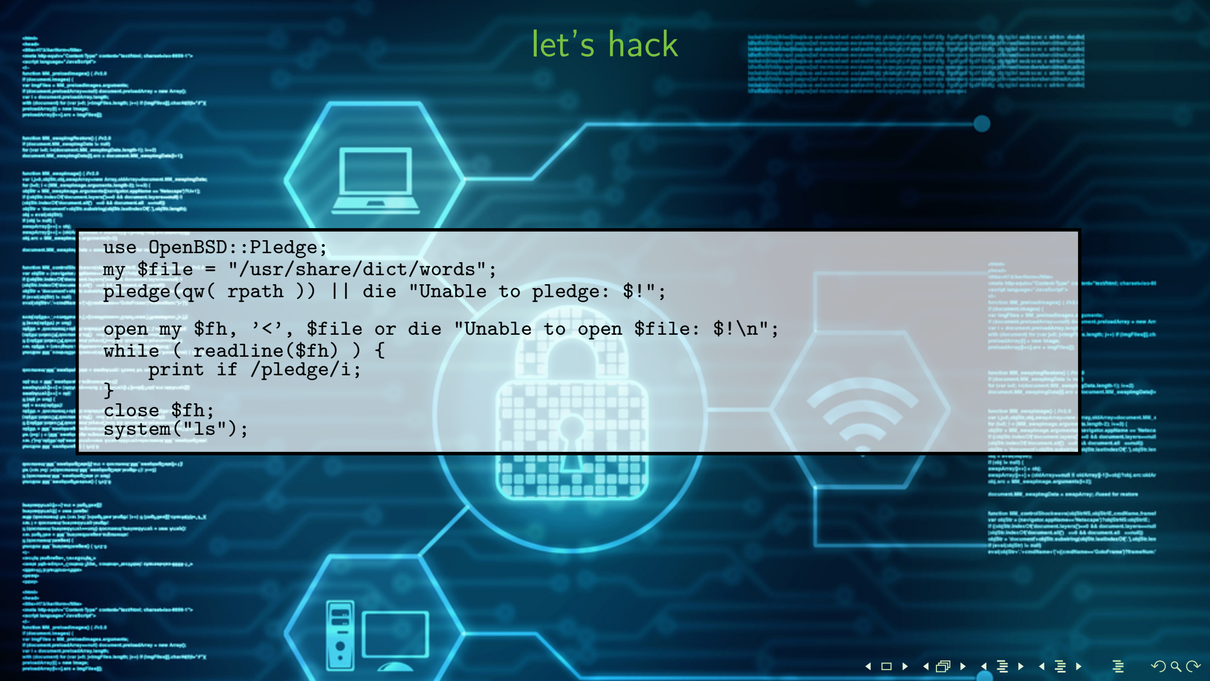
pledge(2) logging

```
From root@bigio.paclan.it Sat Jun 17 16:08:46 2017
Delivered-To: root@bigio.paclan.it
From: Charlie Root <root@bigio.paclan.it>
To: root@bigio.paclan.it
Subject: bigio.paclan.it daily output

OpenBSD 6.1-current (GENERIC) #1: Fri Jun 16 22:37:23 CEST 2017
    giovanni@bigio.paclan.it:/usr/src/sys/arch/amd64/compile/GENERIC

 4:08PM  up 35 mins, 3 users, load averages: 0.26, 0.13, 0.10

Purging accounting records:
hello      -DXP    giovanni   ttyp1      0.00 secs Sat Jun 17 15:56 (0:00:00.16)
```

let's hack

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv) {
FILE *fd;
    if(pledge("stdio", NULL) != -1) {
        printf("Hello pkgsrcCon !\n");
        if((fd = fopen("/etc/passwd", "r"))) {
            printf("Passwd file opened\n");
            fclose(fd);
        }
        return 1;
    } else {
        return 0;
    }
}
```

let’s hack



```
use OpenBSD::Pledge;
my $file = "/usr/share/dict/words";
pledge(qw( rpath )) || die "Unable to pledge: $!";

open my $fh, ’<’, $file or die "Unable to open $file: $!\n";
while ( readline($fh) ) {
    print if /pledge/i;
}
close $fh;
system("ls");
```

let's hack

```
Index: worms.c
===================================================================
RCS file: /var/cvs/src/games/worms/worms.c,v
retrieving revision 1.22
retrieving revision 1.23
diff -u -p -r1.22 -r1.23
--- worms.c 18 Feb 2015 23:16:08 -0000 1.22
+++ worms.c 21 Nov 2015 05:29:42 -0000 1.23
@@ -1,4 +1,4 @@
-/* $OpenBSD: worms.c,v 1.22 2015/02/18 23:16:08 tedu Exp $ */
+/* $OpenBSD: worms.c,v 1.23 2015/11/21 05:29:42 deraadt Exp $ */

 /*
  * Copyright (c) 1980, 1993
@@ -182,6 +182,9 @@ main(int argc, char *argv[])
  struct termios term;
  speed_t speed;
  time_t delay = 0;
+
+ if (pledge("stdio rpath tty", NULL) == -1)
+ err(1, "pledge");

  /* set default delay based on terminal baud rate */
  if (tcgetattr(STDOUT_FILENO, &term) == 0 &&
```

let’s hack, ports(7)

```
$OpenBSD: patch-memcached_c,v 1.11 2016/09/02 14:20:31 giovanni Exp $
--- memcached.c.orig Fri Jun 24 19:41:24 2016
+++ memcached.c Thu Jun 30 00:02:09 2016
@@ -23,6 +23,7 @@
 #include <sys/uio.h>
 #include <ctype.h>
 #include <stdarg.h>
+#include <unistd.h>

 /* some POSIX systems need the following definition
  * to get mlockall flags out of sys/mman.h.  */
@@ -6100,6 +6101,32 @@ int main (int argc, char **argv) {

     if (pid_file != NULL) {
         save_pid(pid_file);
+    }
+
+    if (settings.socketpath != NULL) {
+        if (pid_file != NULL) {
+            if (pledge("stdio cpath unix", NULL) == -1) {
+                fprintf(stderr, "%s: pledge: %s\n", argv[0], strerror(errno));
+                exit(1);
+            }
+        } else {
+            if (pledge("stdio unix", NULL) == -1) {
+                fprintf(stderr, "%s: pledge: %s\n", argv[0], strerror(errno));
+                exit(1);
+            }
+        }
+    } else {
+        if (pid_file != NULL) {
+            if (pledge("stdio cpath inet", NULL) == -1) {
+                fprintf(stderr, "%s: pledge: %s\n", argv[0], strerror(errno));
+                exit(1);
+            }
+        } else {
+            if (pledge("stdio inet", NULL) == -1) {
+                fprintf(stderr, "%s: pledge: %s\n", argv[0], strerror(errno));
+                exit(1);
+            }
+        }
     }

     /* Drop privileges no longer needed */
```

what to do if something goes wrong ?

```
94140 hello    CALL  write(1,0xb56246ae000,0x8)
94140 hello    GIO   fd 1 wrote 8 bytes
      "Hello pkgsrcCon !
      "
94140 hello    RET   write 8
94140 hello    CALL  kbind(0x7f7ffffcbee8,24,0x73b422cd44dee9e4)
94140 hello    RET   kbind 0
94140 hello    CALL  open(0xb53f8a00b20,0<O_RDONLY>)
94140 hello    NAMI  "/etc/passwd"
94140 hello    PLDG  open, "rpath", errno 1 Operation not permitted
94140 hello    PSIG  SIGABRT SIG_DFL
94140 hello    NAMI  "hello.core"
```

what to do if something goes wrong ?

```
$ gdb hello hello.core
GNU gdb 6.3
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "amd64-unknown-openbsd6.1"...
Core was generated by ‘hello’.
Program terminated with signal 6, Aborted.
Loaded symbols for /home/data/server/dati/Documenti/convegni/pkgsrcCon 2017/src/hello
Reading symbols from /usr/lib/libc.so.89.5...done.
Loaded symbols for /usr/lib/libc.so.89.5
Reading symbols from /usr/libexec/ld.so...done.
Loaded symbols for /usr/libexec/ld.so
#0  0x000009f584a507aa in _thread_sys_open () at {standard input}:5
5       {standard input}: No such file or directory.
        in {standard input}
(gdb) bt
#0  0x000009f584a507aa in _thread_sys_open () at {standard input}:5
#1  0x000009f584a3f559 in *_libc_open_cancel (path=Variable "path" is not available.
)
    at /usr/src/lib/libc/sys/w_open.c:36
#2  0x000009f584aaab82 in *_libc_fopen (file=0x9f2b8b00b20 "/etc/passwd", mode=Variable "mode" is not available.

) at /usr/src/lib/libc/stdio/fopen.c:54
#3  0x000009f2b8a005dc in main (argc=1, argv=0x7f7ffffc3c58) at hello.c:8
Current language:  auto; currently asm
(gdb)
```

The future ?