The Challenge

$ ldd /usr/lib/postgresql/8.3/bin/postgres

    linux-gate.so.1 =>  (0xb7c6a000)
    libxml2.so.2 => /usr/lib/libxml2.so.2 (0xb7b05000)
    libpam.so.0 => /lib/libpam.so.0 (0xb7af9000)
    libssl.so.0.9.8 => /usr/lib/i686/cmov/libssl.so.0.9.8 (0xb7ab2000)
    libcrypto.so.0.9.8 => /usr/lib/i686/cmov/libcrypto.so.0.9.8 (0xb7967000)
    libkrb5.so.3 => /usr/lib/libkrb5.so.3 (0xb78d5000)
    libcom_err.so.2 => /lib/libcom_err.so.2 (0xb78d1000)
    libgssapi_krb5.so.2 => /usr/lib/libgssapi_krb5.so.2 (0xb78a7000)
    libcrypt.so.1 => /lib/tls/i686/cmov/libcrypt.so.1 (0xb7740000)
    libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7740000)
    libm.so.6 => /lib/tls/i686/cmov/libm.so.6 (0xb774b000)
    libldap_r-2.4.so.2 => /usr/lib/libldap_r-2.4.so.2 (0xb7808000)
    libncrypt.so.1 => /lib/tls/i686/cmov/libncrypt.so.1 (0xb7848000)
    libz.so.1 => /usr/lib/libz.so.1 (0xb7754000)
    libkeyutils.so.1 => /lib/libkeyutils.so.1 (0xb775a000)
    libresolv.so.2 => /lib/tls/i686/cmov/libresolv.so.2 (0xb775e000)
    libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0xb7763000)
    liblber-2.4.so.2 => /usr/lib/liblber-2.4.so.2 (0xb7762000)
    libbsas12.so.2 => /usr/lib/libbsas12.so.2 (0xb776e000)
    libgnutls.so.26 => /usr/lib/libgnutls.so.26 (0xb7771000)
    libtasn1.so.3 => /usr/lib/libtasn1.so.3 (0xb77f5000)
    libgcrypt.so.11 => /lib/libgcrypt.so.11 (0xb77f6000)
    libgpg-error.so.0 => /lib/libgpg-error.so.0 (0xb77f1000)

$ ldd /usr/lib/libgnutls.so.26

    linux-gate.so.1 =>  (0xb7f6a000)
    libtasn1.so.3 => /usr/lib/libtasn1.so.3 (0xb7f3c000)
    libz.so.1 => /usr/lib/libz.so.1 (0xb7f26000)
    libgcrypt.so.11 => /lib/libgcrypt.so.11 (0xb77ebc000)
    ...
The Problem

Documentation / man pages are incomplete

- `modify_ldt` claims returns only EFAULT, EINVAL, and ENOSYS
  - on Ubuntu, it can also return ENOMEM !

- `htmlParseDocument` (libxml2) claims to only return 0 or -1
  - on Ubuntu, it can also return 1 in some failure cases !

Porting PostgreSQL to other platforms …

- NetBSD : `close()` claims to only return errno codes EBADF or EINTR
- Solaris: ENOLINK is also possible !
- FreeBSD: ECONNRESET is also possible !
- Linux: EIO is also possible !
- HP/UX: ENOSPC is also possible !

Nobody is perfect (even Postgres hackers)

Do you trust PostgreSQL’s error recovery code ?
LFI = Library-level Fault Injector

- http://lfi.sourceforge.net/
- Tool came out of research lab at EPFL (Swiss Federal Inst. of Tech.)
  Test programs by injecting faults at the library interface level
- out-of-memory, conn errors, interrupted syscalls, bad hw...

Based on fault injection scenarios (described in XML files)
LFI profiler can automatically...

- use static analysis of library binaries to discover all errors that could be encountered
- generates injection scenarios
Appeal for Input

Which parts of PostgreSQL are most crucial?

- commit code? buffer management? ...

“Interesting” fault scenarios to start with?

We are looking for interested developers

- Help make LFI a solid tool
- Help solidify PostgreSQL through FI testing

http://lfi.sourceforge.net/