Best practices of sandboxing applications with Capsicum

Mariusz Zaborski
<m.zaborski@fudosecurity.com>
<oshogbo@FreeBSD.org>

Capsicum

Capsicum

kernel infrastructure that provides:

- tight sandboxing

```
int cap_enter(void);
```

Capsicum vs. namespace

- Process IDs
- File paths
- NFS file handle
- Filesystems IDs
- Sysctl MIB
- System V IPC
- POSIX IPC
- System clocks
- Jails
- CPU sets
- Protocol address
- Routing tables

Capabilities

- Should represent many things in OS
- Duplicate capability
- Send/Recv to other process
- Remove capability

Capabilities

- Should represent many things in OS
- Duplicate capability
- Send/Recv to other process
- Remove capability

Descriptors

- Handles to almost everything
- dup(2)
- Over the UNIX domain socket
- close(2)

Allowed syscalls...

sys/kern/capabilites.conf

```
##
## Operations relative to directory capabilities.
##
chflagsat
faccessat
fchmodat
fchownat
fstatat
futimesat
linkat
mkdirat
mkfifoat
mknodat
openat
readlinkat
renameat
symlinkat
unlinkat
utimensat
```

```
##
## Process descriptor-related system calls are allowed.
##
pdfork
pdgetpid
pdkill
#pdwait4         # not yet implemented
```

Capsicum

kernel infrastructure that provides:

- tight sandboxing

```
int cap_enter(void);
```

- capability rights

```
int cap_rights_limit(int fd, const cap_rights_t *rights);
```

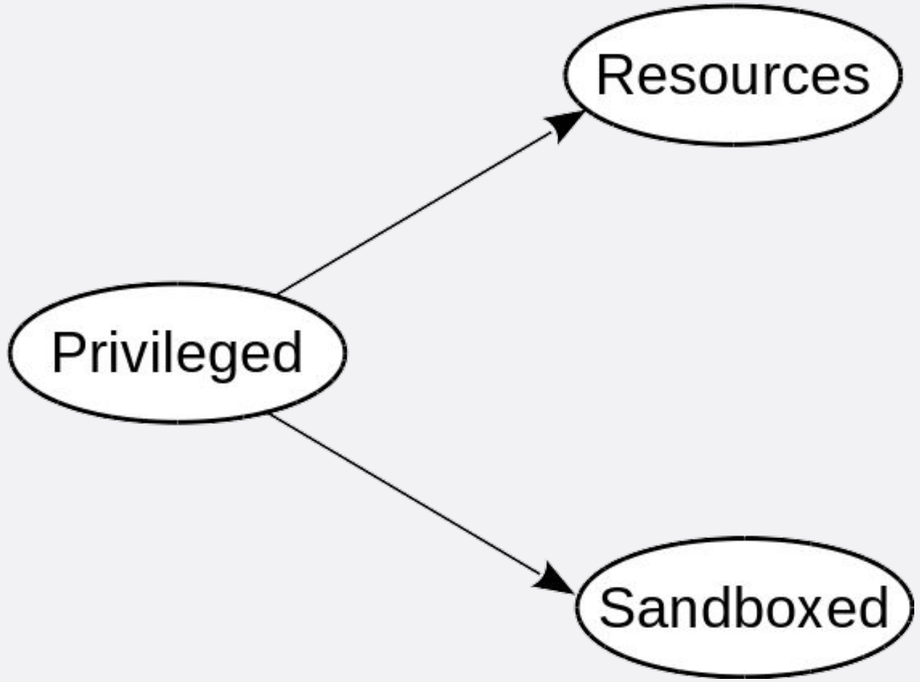
Capsicum rights

- CAP_READ
- CAP_WRITE
- CAP_APPEND
- CAP_ACCEPT
- CAP_FCHMOD
- CAP_CREATE
- CAP_UNLINKAT
- CAP_IOCTL
- CAP_RECV
- CAP_LISTEN
- ...

Capsicum

Two ways to obtain more capabilities:

- the initialization phase
- delegation




```c
int
main(void)
{
        int fd;

        fd = open(...);
        if (cap_enter() < 0 && errno != ENOSYS)
                exit(1);
```

Capsicum is enabled by default in 12.0!

First you need to understand the code!

Understand the code!

```
int
getrandom(void) {
        int fd;

        fd = open("/dev/radnom", O_RDONLY);
        if (fd < 0) {
                /* Fair dice roll. */
                return (4);
        }
```

Debugging infrastructure

Debugging - ktrace

- ktrace/kdump
- Getting only trace

- Very easy to miss something
- Hard to cover all code paths

Debugging - ktrace

```
802 random  CALL  cap_enter
802 random  RET   cap_enter 0
802 random  CALL  openat(AT_FDCWD,0x400877,0<O_RDONLY>)
802 random  CAP   restricted VFS lookup
802 random  RET   openat -1 errno 94 Not permitted in capability mode
802 random  CALL  sigprocmask(SIG_BLOCK,0x8008209c8,0x7ffffffe640)
802 random  RET   sigprocmask 0
802 random  CALL  sigprocmask(SIG_SETMASK,0x8008209dc,0)
802 random  RET   sigprocmask 0
802 random  CALL  sigprocmask(SIG_BLOCK,0x8008209c8,0x7ffffffe1b0)
```

Debugging - enotcap

- kern.trap_enotcap
- procctl(PROC_TRAPCAP_CTL)
- Getting core dump

- Hard to miss something
- Hard to cover all code paths

- kern.capmode_coredump

Debugging - enotcap

```
Program received signal SIGTRAP, Trace/breakpoint trap.
0x0000000080090b34a in _openat () from /lib/libc.so.7
Current language:  auto; currently minimal
Breakpoint 1 at 0x80090b34a
(gdb) bt
#0  0x0000000080090b34a in _openat () from /lib/libc.so.7
#1  0x0000000080086e457 in open (path=<value optimized out>,
flags=<value optimized out>)
    at /usr/src/lib/libc/sys/open.c:57
#2  0x0000000000400a18 in main () at a.c:24
```

Debugging - procstat(1)

```
PID COMM                FD  T FLAGS    CAPABILITIES PRO NAME
494 dhclient          text  v r------- -            -   /sbin/dhclient
494 dhclient          cwd   v r------- -            -   /
494 dhclient          root  v r------- -            -   /
494 dhclient            0   v rw------ rd,wr,se,mm  -   /dev/null
494 dhclient            1   v rw------ rd,wr,se,mm  -   /dev/null
494 dhclient            2   v rw------ rd,wr,se,mm  -   /dev/null
```

Deduplicate your code

Capsicum helpers

- capsicum_helpers.h
- Inline functions:
  - caph_enter()
  - caph_enter_casper()
  - caph_limit_stdio()
  - caph_limit_stdout()
  - caph_limit_stdin()
  - caph_limit_stderr()
  - caph_cache_catpages()
  - caph_cache_tzdata()
  - caph_ioctls_limit()
  - caph_rights_limit()

IPC - libnv

- nvlist_create
- nvlist_add_${type}
- nvlist_get_${type}
- nvlist_take_${type}
- nvlist_move_${type}
- nvlist_send
- nvlist_recv
- nvlist_destroy

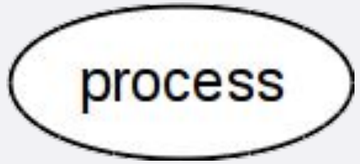
- Types:
  - string
  - number
  - bool
  - nvlist
  - descriptor
  - binary
  - array

Casper

- Provides functionality not available in capability mode through convenient APIs making Capsicum more practical

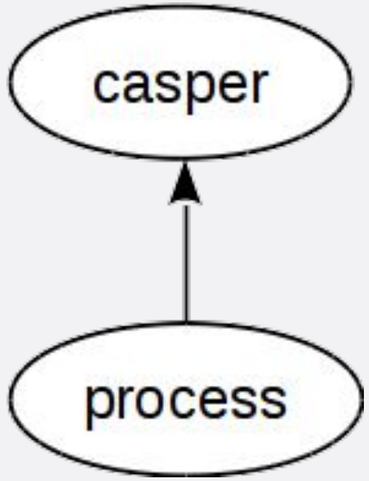
- Make easier to separate process
- Create before entering Capability mode
- Set of dynamic libraries

Casper - how its works?

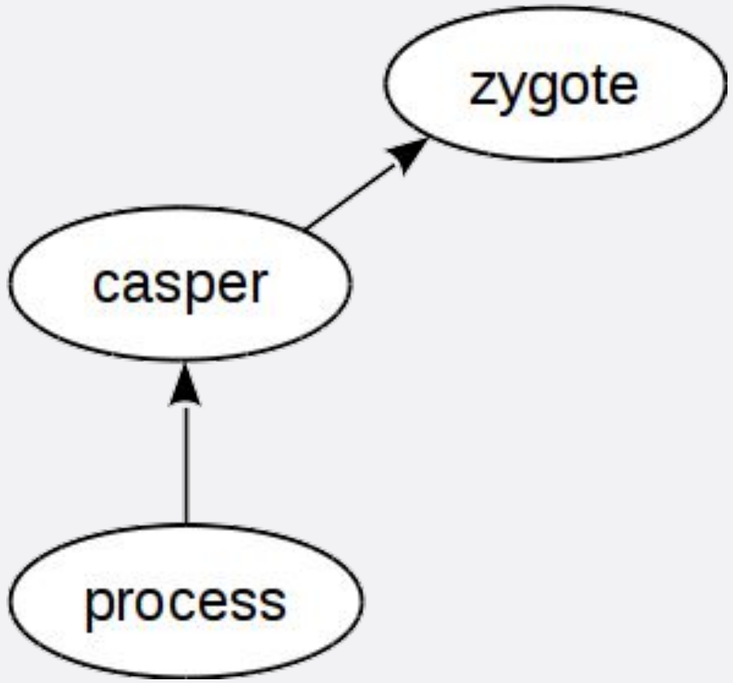


Casper - how its works?

- cap_init()
- cap_service_open()
- cap_close()

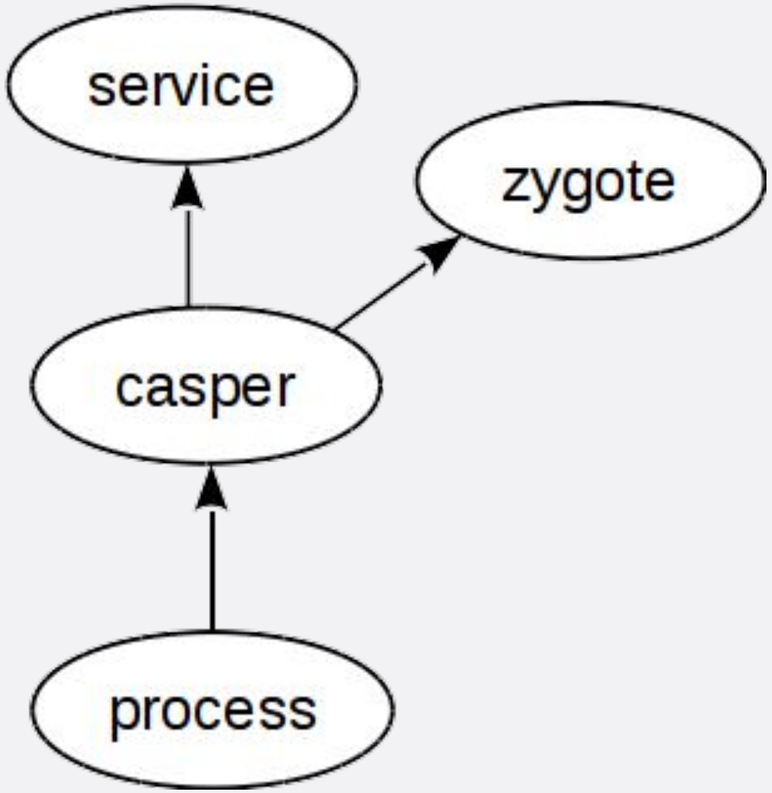


Casper - how its works?



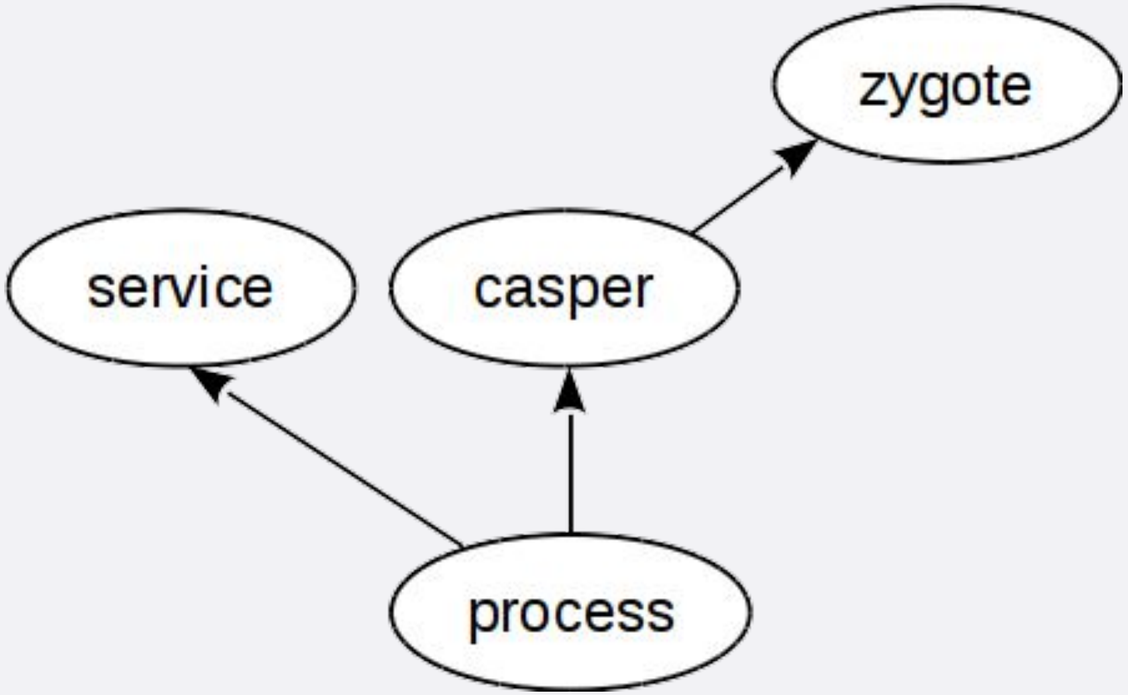
- cap_init()
- cap_service_open()
- cap_close()

Casper - how its works?



- cap_init()
- cap_service_open()
- cap_close()

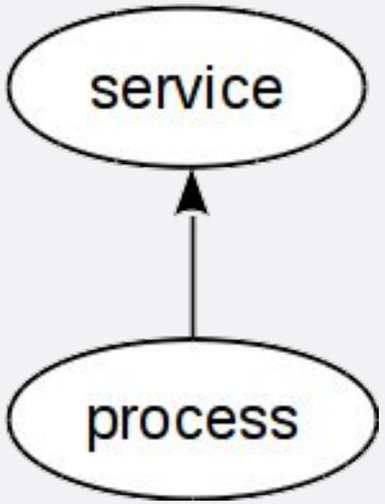
Casper - how its works?



- cap_init()
- cap_service_open()
- cap_close()

Casper - how its works?

- cap_init()
- cap_service_open()
- cap_close()



Casper services

- system.dns
- system.grp
- system.pwd
- system.random
- system.sysctl
- system.syslog

Casper services

- system.dns
- system.grp
- system.pwd
- system.random
- system.sysctl
- system.syslog

- system.tls
- system.socket
- system.configuration

Let’s sandbox something

bspatch(1) - Step 0: read the code

```
if ((f = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);

if (fread(header, 1, 32, f) < 32) {
        if (feof(f))
                errx(1, "Corrupt patch\n");
        err(1, "fread(%s)", argv[3]);
}

if (memcmp(header, "BSDIFF40", 8) != 0)
        errx(1, "Corrupt patch\n");

bzctrllen = offtin(header + 8);
bzdatalen = offtin(header + 16);
newsize = offtin(header + 24);
if ((bzctrllen < 0) || (bzdatalen < 0) || (newsize < 0))
        errx(1, "Corrupt patch\n");

if (fclose(f))
        err(1, "fclose(%s)", argv[3]);
if ((cpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(cpf, 32, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)32);
```

```
if ((cpfbz2 = BZ2_bzReadOpen(&cbz2err, cpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", cbz2err);
if ((dpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(dpf, 32 + bzctrllen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen));
if ((dpfbz2 = BZ2_bzReadOpen(&dbz2err, dpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", dbz2err);
if ((epf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(epf, 32 + bzctrllen + bzdatalen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen + bzdatalen));
if ((epfbz2 = BZ2_bzReadOpen(&ebz2err, epf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", ebz2err);
```

bspatch(1) - Step 0: read the code

```
if ((f = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);

if (fread(header, 1, 32, f) < 32) {
        if (feof(f))
                errx(1, "Corrupt patch\n");
        err(1, "fread(%s)", argv[3]);
}

if (memcmp(header, "BSDIFF40", 8) != 0)
        errx(1, "Corrupt patch\n");

bzctrllen = offtin(header + 8);
bzdatalen = offtin(header + 16);
newsize = offtin(header + 24);
if ((bzctrllen < 0) || (bzdatalen < 0) || (newsize < 0))
        errx(1, "Corrupt patch\n");

if (fclose(f))
        err(1, "fclose(%s)", argv[3]);
if ((cpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(cpf, 32, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)32);
```

```
if ((cpfbz2 = BZ2_bzReadOpen(&cbz2err, cpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", cbz2err);
if ((dpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(dpf, 32 + bzctrllen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen));
if ((dpfbz2 = BZ2_bzReadOpen(&dbz2err, dpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", dbz2err);
if ((epf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(epf, 32 + bzctrllen + bzdatalen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen + bzdatalen));
if ((epfbz2 = BZ2_bzReadOpen(&ebz2err, epf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", ebz2err);
```

bspatch(1) - Step 1: code reorganization

```
@@ -89,0 +90,11 @@ int main(int argc, char *argv[])
+       if ((cpf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((dpf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((epf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((oldfd = open(argv[1], O_RDONLY | O_BINARY, 0)) < 0)
+               err(1, "open(%s)", argv[1]);
+       if ((newfd = open(argv[2], O_CREAT | O_TRUNC | O_WRONLY | O_BINARY,
+           0666)) < 0)
+               err(1, "open(%s)", argv[2]);
@@ -126,2 +177,0 @@ int main(int argc, char *argv[])
-       if ((cpf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -133,2 +182,0 @@ int main(int argc, char *argv[])
-       if ((dpf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -140,2 +187,0 @@ int main(int argc, char *argv[])
-       if ((epf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -148,3 +193,0 @@ int main(int argc, char *argv[])
-       oldfd = open(argv[1], O_RDONLY | O_BINARY, 0);
-       if (oldfd < 0)
-               err(1, "%s", argv[1]);
@@ -218,3 +260,0 @@ int main(int argc, char *argv[])
-       newfd = open(argv[2], O_CREAT | O_TRUNC | O_WRONLY | O_BINARY, 0666);
-       if (newfd < 0)
-               err(1, "%s", argv[2]);
```

bspatch(1) - Capsicumize

```
@@ -89,0 +90,11 @@ int main(int argc, char *argv[])
+       if ((cpf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((dpf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((epf = fopen(argv[3], "rb")) == NULL)
+               err(1, "fopen(%s)", argv[3]);
+       if ((oldfd = open(argv[1], O_RDONLY | O_BINARY, 0)) < 0)
+               err(1, "open(%s)", argv[1]);
+       if ((newfd = open(argv[2], O_CREAT | O_TRUNC | O_WRONLY | O_BINARY,
+           0666)) < 0)
+               err(1, "open(%s)", argv[2]);
@@ -126,2 +177,0 @@ int main(int argc, char *argv[])
-       if ((cpf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -133,2 +182,0 @@ int main(int argc, char *argv[])
-       if ((dpf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -140,2 +187,0 @@ int main(int argc, char *argv[])
-       if ((epf = fopen(argv[3], "rb")) == NULL)
-               err(1, "fopen(%s)", argv[3]);
@@ -148,3 +193,0 @@ int main(int argc, char *argv[])
-       oldfd = open(argv[1], O_RDONLY | O_BINARY, 0);
-       if (oldfd < 0)
-               err(1, "%s", argv[1]);
@@ -218,3 +260,0 @@ int main(int argc, char *argv[])
-       newfd = open(argv[2], O_CREAT | O_TRUNC | O_WRONLY | O_BINARY, 0666);
-       if (newfd < 0)
-               err(1, "%s", argv[2]);
```

cap_enter()

bspatch(1) - Step 2: read more code

```
if ((f = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);

if (fread(header, 1, 32, f) < 32) {
        if (feof(f))
                errx(1, "Corrupt patch\n");
        err(1, "fread(%s)", argv[3]);
}

if (memcmp(header, "BSDIFF40", 8) != 0)
        errx(1, "Corrupt patch\n");

bzctrllen = offtin(header + 8);
bzdatalen = offtin(header + 16);
newsize = offtin(header + 24);
if ((bzctrllen < 0) || (bzdatalen < 0) || (newsize < 0))
        errx(1, "Corrupt patch\n");

if (fclose(f))
        err(1, "fclose(%s)", argv[3]);
if ((cpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(cpf, 32, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)32);
```

```
if ((cpfbz2 = BZ2_bzReadOpen(&cbz2err, cpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", cbz2err);
if ((dpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(dpf, 32 + bzctrllen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen));
if ((dpfbz2 = BZ2_bzReadOpen(&dbz2err, dpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", dbz2err);
if ((epf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(epf, 32 + bzctrllen + bzdatalen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen + bzdatalen));
if ((epfbz2 = BZ2_bzReadOpen(&ebz2err, epf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", ebz2err);
```

bspatch(1) - Step 2: read more code

```
if ((f = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);

if (fread(header, 1, 32, f) < 32) {
        if (feof(f))
                errx(1, "Corrupt patch\n");
        err(1, "fread(%s)", argv[3]);
}

if (memcmp(header, "BSDIFF40", 8) != 0)
        errx(1, "Corrupt patch\n");

bzctrllen = offtin(header + 8);
bzdatalen = offtin(header + 16);
newsize = offtin(header + 24);
if ((bzctrllen < 0) || (bzdatalen < 0) || (newsize < 0))
        errx(1, "Corrupt patch\n");

if (fclose(f))
        err(1, "fclose(%s)", argv[3]);
if ((cpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(cpf, 32, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)32);
```

```
if ((cpfbz2 = BZ2_bzReadOpen(&cbz2err, cpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", cbz2err);
if ((dpf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(dpf, 32 + bzctrllen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen));
if ((dpfbz2 = BZ2_bzReadOpen(&dbz2err, dpf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", dbz2err);
if ((epf = fopen(argv[3], "rb")) == NULL)
        err(1, "fopen(%s)", argv[3]);
if (fseeko(epf, 32 + bzctrllen + bzdatalen, SEEK_SET))
        err(1, "fseeko(%s, %lld)", argv[3],
            (long long)(32 + bzctrllen + bzdatalen));
if ((epfbz2 = BZ2_bzReadOpen(&ebz2err, epf, 0, 0, NULL, 0)) == NULL)
        errx(1, "BZ2_bzReadOpen, bz2err = %d", ebz2err);
```

bspatch(1) - Step 3: Capsicumize

```
@@ -82,0 +95,3 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       cap_rights_t rights_ro, rights_wr;
+#endif
@@ -90,0 +105,17 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       if (cap_enter() < 0 &&errno != ENOSYS) {
+                       err(1, "failed to enter security sandbox");
+       } else {
+               cap_rights_init(&rights_ro, CAP_READ, CAP_FSTAT, CAP_SEEK);
+               cap_rights_init(&rights_wr, CAP_WRITE);
+
+               if (cap_rights_limit(fileno(f), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(cpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(dpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(epf), &rights_ro) < 0 ||
+                   cap_rights_limit(oldfd, &rights_ro) < 0 ||
+                   cap_rights_limit(newfd, &rights_wr) < 0)
+                       err(1, "cap_rights_limit() failed, could not restrict"
+                           " capabilities");
+       }
+#endif
```

bspatch(1) - Step 3: Capsicumize

```
@@ -82,0 +95,3 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       cap_rights_t rights_ro, rights_wr;
+#endif
@@ -90,0 +105,17 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       if (cap_enter() < 0 &&errno != ENOSYS) {
+                       err(1, "failed to enter security sandbox");
+       } else {
+               cap_rights_init(&rights_ro, CAP_READ, CAP_FSTAT, CAP_SEEK);
+               cap_rights_init(&rights_wr, CAP_WRITE);
+
+               if (cap_rights_limit(fileno(f), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(cpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(dpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(epf), &rights_ro) < 0 ||
+                   cap_rights_limit(oldfd, &rights_ro) < 0 ||
+                   cap_rights_limit(newfd, &rights_wr) < 0)
+                       err(1, "cap_rights_limit() failed, could not restrict"
+                           " capabilities");
+       }
+#endif
```

bspatch(1) - Step 3: Capsicumize

```
@@ -82,0 +95,3 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       cap_rights_t rights_ro, rights_wr;
+#endif
@@ -90,0 +105,17 @@ int main(int argc, char *argv[])
+#ifdef HAVE_CAPSICUM
+       if (cap_enter() < 0 &&errno != ENOSYS) {
+                       err(1, "failed to enter security sandbox");
+       } else {
+               cap_rights_init(&rights_ro, CAP_READ, CAP_FSTAT, CAP_SEEK);
+               cap_rights_init(&rights_wr, CAP_WRITE);
+
+               if (cap_rights_limit(fileno(f), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(cpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(dpf), &rights_ro) < 0 ||
+                   cap_rights_limit(fileno(epf), &rights_ro) < 0 ||
+                   cap_rights_limit(oldfd, &rights_ro) < 0 ||
+                   cap_rights_limit(newfd, &rights_wr) < 0)
+                       err(1, "cap_rights_limit() failed, could not restrict"
+                           " capabilities");
+       }
+#endif
```

caph_enter()

Casper - dhclient(8)

```
Starting devd.
Starting dhclient.
pid 336 (dhclient), uid (65):  Path `/var/crash/dhclient.65.0.core' failed on initial open
test, error = 2
pid 336 (dhclient), uid 65: exited on signal 5
Trace/BPT trap
/etc/rc.d/dhclient: WARNING: failed to start dhclient
add host 127.0.0.1: gateway lo0 fib 0: route already in table
Script /etc/rc.d/defaultroute interrupted
Creating and/or trimming log files.
Starting syslogd.
```

Casper - dhclient(8)

```
Starting devd.
Starting dhclient.
pid 336 (dhclient), uid (65):  Path `/var/crash/dhclient.65.0.core' failed on initial open
test, error = 2
pid 336 (dhclient), uid 65: exited on signal 5
Trace/BPT trap
/etc/rc.d/dhclient: WARNING: failed to start dhclient
add host 127.0.0.1: gateway lo0 fib 0: route already in table
Script /etc/rc.d/defaultroute interrupted
Creating and/or trimming log files.
Starting syslogd.
```

Casper - dhclient(8)

```
Starting program: /sbin/dhclient vtnet1

Program received signal SIGTRAP, Trace/breakpoint trap.
0x0000000800bbdd1a in _connect () from /lib/libc.so.7
Current language:  auto; currently minimal
(gdb) bt
#0  0x0000000800bbdd1a in _connect () from /lib/libc.so.7
#1  0x0000000800bb0499 in connectlog ()
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:379
#2  0x0000000800bb0090 in vsyslog (pri=<value optimized out>,
    fmt=<value optimized out>, ap=0x7fffffffe9c0)
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:254
#3  0x0000000800bafcdd in syslog (pri=<value optimized out>,
    fmt=<value optimized out>)
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:128
#4  0x000000000040cf7b in note (fmt=0x41056d "")
    at /usr/home/oshogbo/git/freebsd/sbin/dhclient/errwarn.c:132
#5  0x0000000000405178 in send_discover (ipp=0x80066a000)
    at /usr/home/oshogbo/git/freebsd/sbin/dhclient/dhclient.c:1285
#6  0x00000000004037a2 in main (argc=<value optimized out>, argv=<value optimized out>)
```

Casper - dhclient(8)

```
Starting program: /sbin/dhclient vtnet1

Program received signal SIGTRAP, Trace/breakpoint trap.
0x0000000800bbdd1a in _connect () from /lib/libc.so.7
Current language:  auto; currently minimal
(gdb) bt
#0  0x0000000800bbdd1a in _connect () from /lib/libc.so.7
#1  0x0000000800bb0499 in connectlog ()
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:379
#2  0x0000000800bb0090 in vsyslog (pri=<value optimized out>,
    fmt=<value optimized out>, ap=0x7fffffffe9c0)
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:254
#3  0x0000000800bafcdd in syslog (pri=<value optimized out>,
    fmt=<value optimized out>)
    at /usr/home/oshogbo/git/freebsd/lib/libc/gen/syslog.c:128
#4  0x000000000040cf7b in note (fmt=0x41056d "")
    at /usr/home/oshogbo/git/freebsd/sbin/dhclient/errwarn.c:132
#5  0x0000000000405178 in send_discover (ipp=0x80066a000)
    at /usr/home/oshogbo/git/freebsd/sbin/dhclient/dhclient.c:1285
#6  0x00000000004037a2 in main (argc=<value optimized out>, argv=<value optimized out>)
```

Casper - dhclient(8)

```
359         /* Initially, log errors to stderr as well as to syslogd. */
360         openlog(__progname, LOG_PID | LOG_NDELAY, DHCPD_LOG_FACILITY);
361         setlogmask(LOG_UPTO(LOG_DEBUG));

521         if (cap_enter() < 0 && errno != ENOSYS)
522                 error("can't enter capability mode: %m");
```

```
void openlog(const char *ident, int logopt, int facility);

The routines closelog(), openlog(), syslog() and vsyslog() return no value.
```

Casper - dhclient(8)

```
Starting devd.
Starting dhclient.
pid 336 (dhclient), uid (65):  Path `/var/crash/dhclient.65.0.core' failed on initial open
test, error = 2
pid 336 (dhclient), uid 65: exited on signal 5
Trace/BPT trap
/etc/rc.d/dhclient: WARNING: failed to start dhclient
add host 127.0.0.1: gateway lo0 fib 0: route already in table
Script /etc/rc.d/defaultroute interrupted
Creating and/or trimming log files.
Starting syslogd.
```

Casper - dhclient(8)

```
Starting devd.
Starting dhclient.
pid 336 (dhclient), uid (65):  Path `/var/crash/dhclient.65.0.core' failed on initial open
test, error = 2
pid 336 (dhclient), uid 65: exited on signal 5
Trace/BPT trap
/etc/rc.d/dhclient: WARNING: failed to start dhclient
add host 127.0.0.1: gateway lo0 fib 0: route already in table
Script /etc/rc.d/defaultroute interrupted
Creating and/or trimming log files.
Starting syslogd.
```

Casper - dhclient(8)

```
--- sbin/dhclient/dhclient.c
+++ sbin/dhclient/dhclient.c
@@ -345,6 +347,21 @@ routehandler(struct protocol *p)
        exit(1);
 }

+static void
+init_casper(void)
+{
+       cap_channel_t           *casper;
+
+       casper = cap_init();
+       if (casper == NULL)
+               error("unable to start casper");
+
+       capsyslog = cap_service_open(casper, "system.syslog");
+       cap_close(casper);
+       if (capsyslog == NULL)
+               error("unable to open system.syslog service");
+}
+
 int
 main(int argc, char *argv[])
 {
```

Casper - dhclient(8)

```
@@ -356,9 +373,11 @@ main(int argc, char *argv[])
        pid_t                   otherpid;
        cap_rights_t            rights;

+       init_casper();
+
        /* Initially, log errors to stderr as well as to syslogd. */
-       openlog(__progname, LOG_PID | LOG_NDELAY, DHCPD_LOG_FACILITY);
-       setlogmask(LOG_UPTO(LOG_DEBUG));
+       cap_openlog(capsyslog, __progname, LOG_PID | LOG_NDELAY, DHCPD_LOG_FACILITY);
+       cap_setlogmask(capsyslog, LOG_UPTO(LOG_DEBUG));

        while ((ch = getopt(argc, argv, "bc:dl:p:qu")) != -1)
                switch (ch) {
@@ -518,7 +537,7 @@ main(int argc, char *argv[])

        setproctitle("%s", ifi->name);

-       if (cap_enter() < 0 && errno != ENOSYS)
+       if (caph_enter_with_casper() < 0)
                error("can't enter capability mode: %m");

        if (immediate_daemon)
```

Casper - dhclient(8)

```
@@ -78,7 +78,7 @@ error(char *fmt, ...)
                write(2, "\n", 1);
        }

-       syslog(LOG_CRIT, "exiting.");
+       cap_syslog(capsyslog, LOG_CRIT, "exiting.");
        if (log_perror) {
                fprintf(stderr, "exiting.\n");
                fflush(stderr);
```

File system

cap_fileargs

- Easy sandbox applications which operates on argc/argv

```
fileargs_t *fileargs_init(int argc, char *argv[], int flags, mode_t mode,
    cap_rights_t *rightsp);
fileargs_t *fileargs_cinit(cap_channel_t *cas, int argc, char *argv[],
    int flags, mode_t mode, cap_rights_t *rightsp);
void fileargs_free(fileargs_t *fa);

int fileargs_open(fileargs_t *fa, const char *name);
FILE *fileargs_fopen(fileargs_t *fa, const char *name, const char *mode);
```

Let’s try it!
https://reviews.freebsd.org/D14408

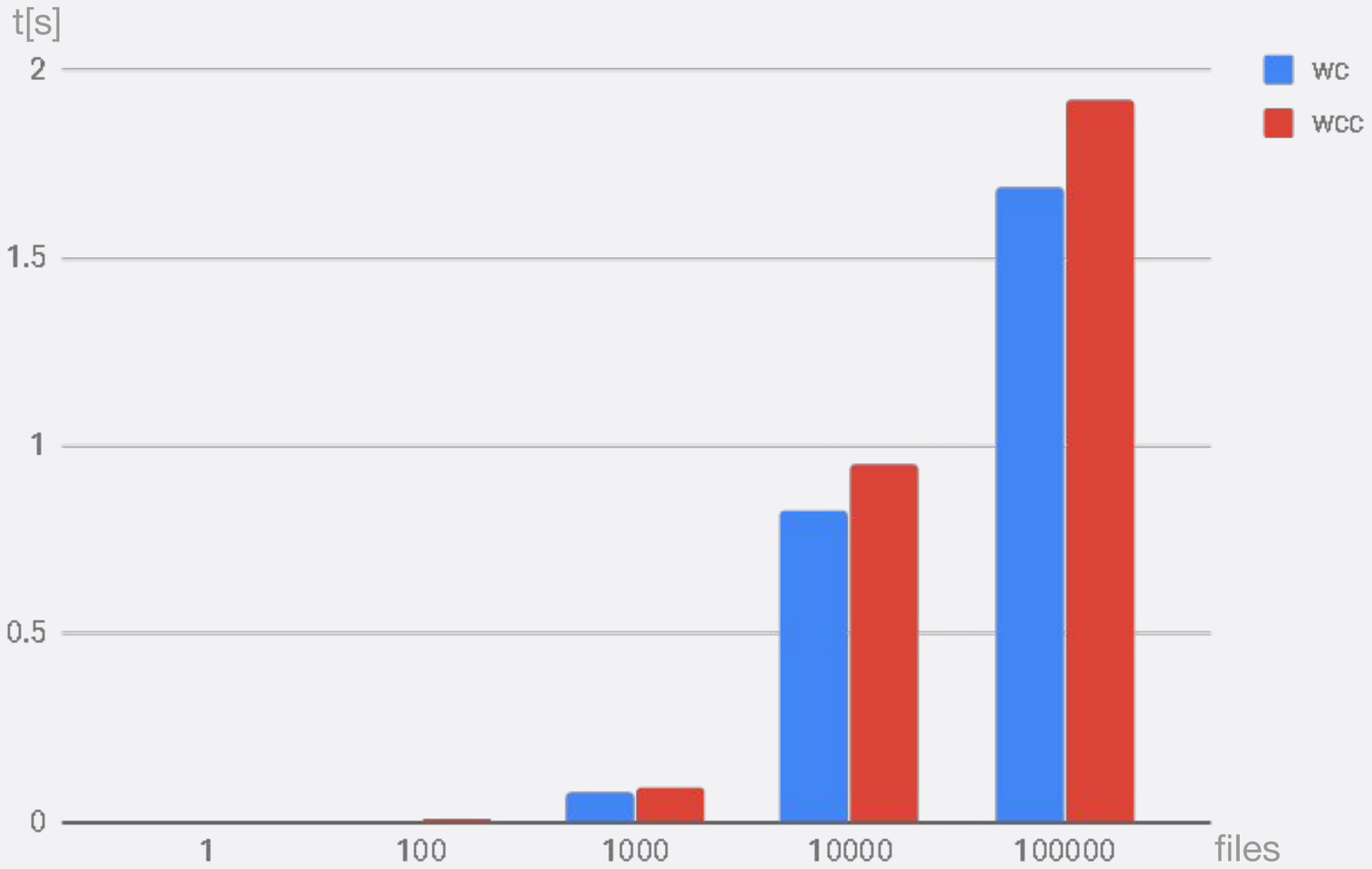
```
@@ -125,6 +132,26 @@

        (void)signal(SIGINFO, siginfo_handler);

+       fa = fileargs_init(argc, argv, O_RDONLY, 0,
+           cap_rights_init(&rights, CAP_READ, CAP_FSTAT));
+       if (fa == NULL) {
+               xo_warn("Unable to init casper");
+               exit(1);
+       }
+
+       caph_cache_catpages();
+       if (caph_limit_stdio() < 0) {
+               xo_warn("Unable to limit stdio");
+               fileargs_free(fa);
+               exit(1);
+       }
+
+       if (caph_enter_with_casper() < 0) {
+               xo_warn("Unable to enter capability mode");
+               fileargs_free(fa);
+               exit(1);
+       }
+
        /* Wc's flags are on by default. */
        if (doline + doword + dochar + domulti + dolongline == 0)
                doline = doword = dochar = 1;
```

```
@@ -206,7 +234,7 @@
        linect = wordct = charct = llct = tmpll = 0;
        if (file == NULL)
                fd = STDIN_FILENO;
-       else if ((fd = open(file, O_RDONLY, 0)) < 0) {
+       else if ((fd = fileargs_open(fa, file)) < 0) {
                xo_warn("%s: open", file);
                return (1);
        }
```

Performance



Need more hands…

- usr.bin/login
- usr.bin/newgrp
- usr.bin/opiepasswd
- usr.bin/chpass
- usr.bin/bluetooth/btsockstat
- usr.bin/lock
- usr.bin/passwd
- usr.bin/su
- usr.bin/netstat
- usr.bin/at
- usr.bin/opieinfo
- usr.bin/wall
- sbin/shutdown

https://wiki.freebsd.org/Capsicum

Thank you!

Mariusz Zaborski

https://oshogbo.vexillium.org

oshogbo@FreeBSD.org

@oshogbovx