The Apple Sandbox

Dionysus Blazakis
dion@securityevaluators.com
Blackhat DC 2011

Where to find stuff

https://github.com/dionthegod/XNUSandbox

http://www.semantiscope.com/research/
BHDC2011/BHDC2011-Paper.pdf

http://www.semantiscope.com/research/
BHDC2011/BHDC2011-Slides.pdf

I’m Dion

I work for ISE as a reverser/cracker/developer/exploiter.

I’m not a security old salt (embedded developer by trade.)

Software is hard.

I used to fuzz Adobe Reader all the time.

It broke a lot.

Later, I learned most software breaks a lot.

We should *totally* do something about this.

static analysis tools

large scale fuzz testing

developer training

change control

(formal methods)

Suppose # bugs are going to zero

How long will it take?

What happens for the next 5 (50) years?

Assume an attacker can, for the near future, always find a bug cheaply.

Got a bug, now what?

OS exploit mitigations. Written by security people that are developers (!!!?!?)

Mitigations make exploitation much more expensive, but still relatively cheap.

Client apps are behind

Separating privileges is nothing new for server applications.

Maybe it’s a good idea for client applications to be explicit about privileges.

(i.e. your browser’s HTML parser doesn’t need to execute calc.exe)

A simile

Exploitation is like a chase scene.

You need to get to through an alley, but there is always that barbed-wire fence.

Client apps (.NET or Flash or any info leaks) keep stacking cardboard boxes against the first fence (OS mitigations).

The sandboxes are coming!

MS Internet Explorer and Office Protected View

Google Chrome

Adobe Reader X

iOS AppStore

OS Support

Fine-grained control via process syscall filtering:

Linux: SELinux, AppArmor

FreeBSD, XNU: TrustedBSD

This Talk

A top-down walkthrough of the XNU Sandbox

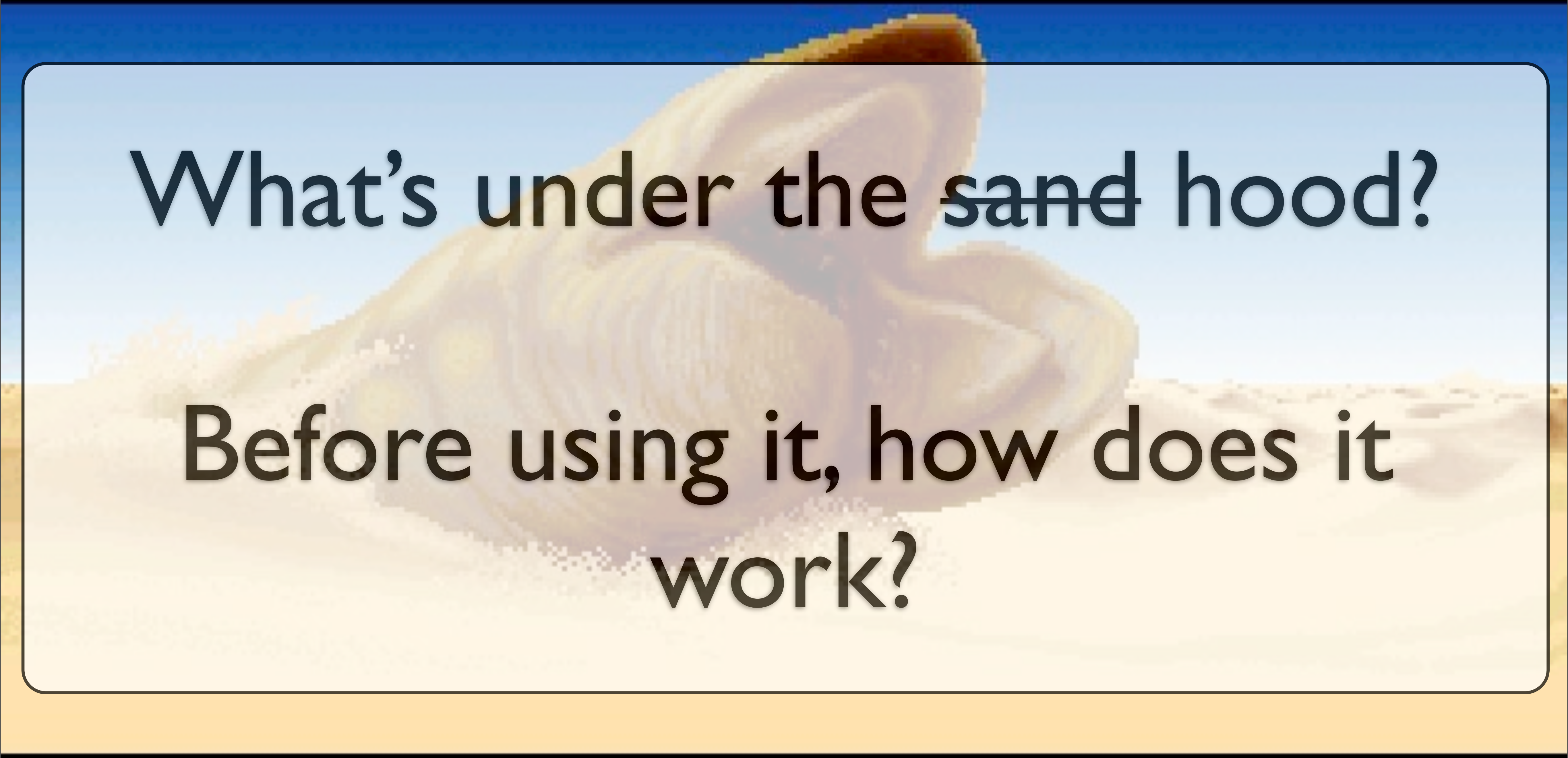
Not this talk

Some sandbox escape. If you were expecting me to give you one, feel free to be let down.

Why do you care?

Giant sandworms!

~~Giant sandworms!~~



What’s under the ~~sand~~ hood?

Before using it, how does it work?

XNU Sandbox

Previously, codenamed “Seatbelt”

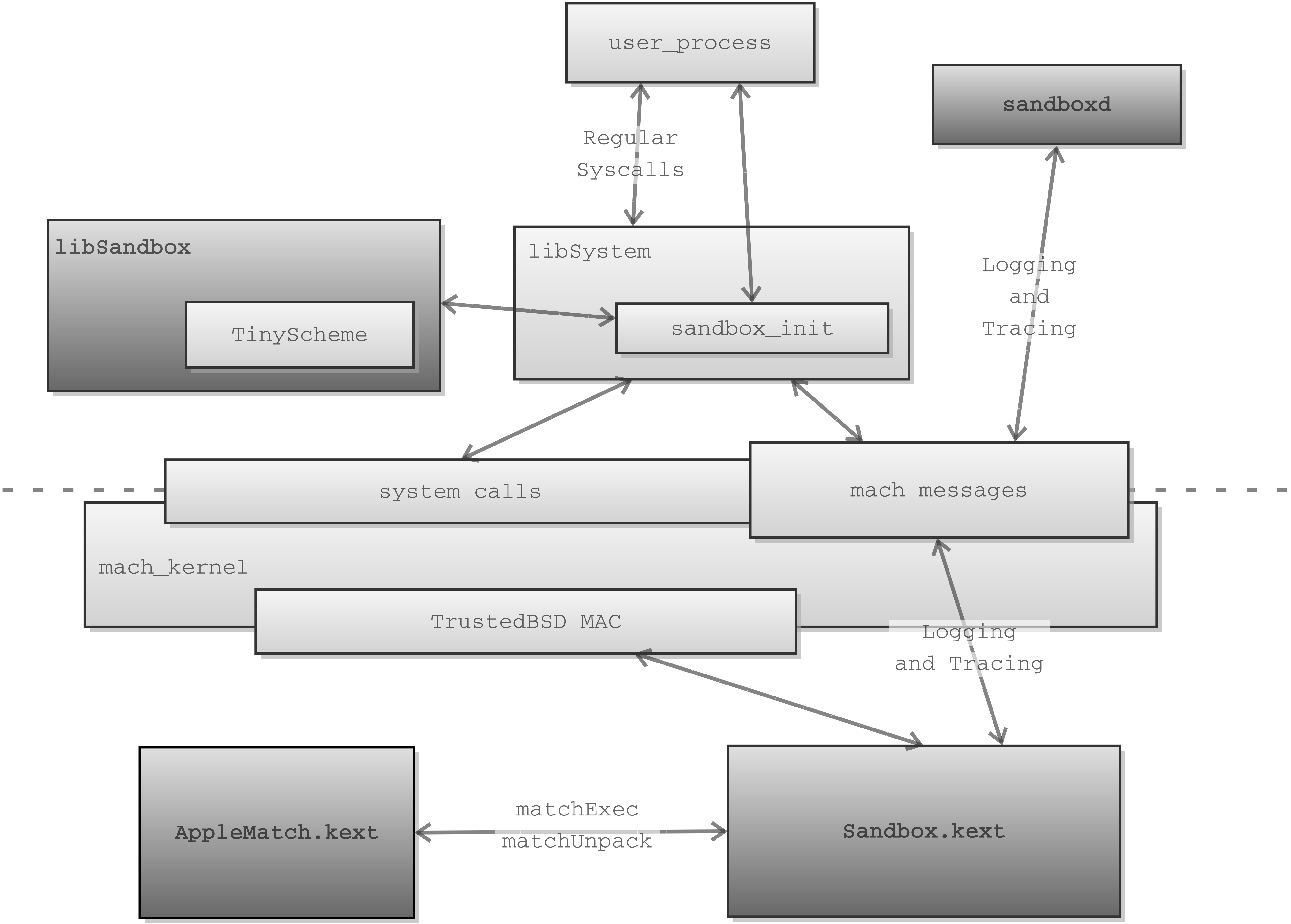
For XNU systems, implemented as a TrustedBSD policy module

Runtime configurable, per-process access control policy

Used to contain AppStore application on iOS

Example: restricting network usage

```
fluffy:tmp dion$ sandbox-exec -n no-internet /bin/sh
sh-3.2$ file /etc/passwd
/etc/passwd: ASCII English text
sh-3.2$ ping www.eff.org
PING eff.org (64.147.188.3): 56 data bytes
ping: sendto: Operation not permitted
^C
--- eff.org ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
sh-3.2$ exit
```


user_process
sandboxd
Regular
Syscalls
libSandbox
TinyScheme
libSystem
sandbox_init
Logging
and
Tracing
system calls
mach messages
mach_kernel
TrustedBSD MAC
Logging
and Tracing
AppleMatch.kext
matchExec
matchUnpack
Sandbox.kext

Public interface

“Documented” interfaces:

sandbox-exec(1)

sandbox_init(3)

sandbox-exec

```
NAME
    sandbox-exec -- execute within a sandbox

SYNOPSIS
    sandbox-exec [-f profile-file] [-n profile-name] [-p profile-string]
                 [-D key=value ...] command [arguments ...]

DESCRIPTION
    The sandbox-exec command enters a sandbox using a profile specified by
    the -f, -n, or -p option and executes command with arguments.

    The options are as follows:

    -f profile-file
            Read the profile from the file named profile-file.

    -n profile-name
            Use the pre-defined profile profile-name.

    -p profile-string
            Specify the profile to be used on the command line.

    -D key=value
            Set the profile parameter key to value.
```

sandbox-exec

```
NAME
    sandbox-exec -- execute within a sandbox

SYNOPSIS
    sandbox-exec [-f profile-file] [-n profile-name] [-p profile-string]
                 [-D key=value ...] command [arguments ...]

DESCRIPTION
    The sandbox-exec command enters a sandbox using a profile specified by
    the -f, -n, or -p option and executes command with arguments.

    The options are as follows:

    -f profile-file
            Read the profile from the file named profile-file

    -n profile-name
            Use the pre-defined profile profile-name

    -p profile-string
            Specify the profile to be used on the command line.

    -D key=value
            Set the profile parameter key to value.
```

sample files? where?

what are these names??

Example: restricting network usage

```
fluffy:tmp dion$ sandbox-exec -n no-internet /bin/sh
sh-3.2$ file /etc/passwd
/etc/passwd: ASCII English text
sh-3.2$ ping www.eff.org
PING eff.org (64.147.188.3): 56 data bytes
ping: sendto: Operation not permitted
^C
--- eff.org ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
sh-3.2$ exit
```

sandbox_init

```
NAME
     sandbox_init -- set process sandbox

SYNOPSIS
     #include <sandbox.h>

     int
     sandbox_init(const char *profile, uint64_t flags, char **errorbuf);

DESCRIPTION
     sandbox_init() places the current process into a sandbox(7). The
     NUL-terminated string profile specifies the profile to be used to config-
     ure the sandbox.  The flags specified are formed by or'ing the following
     values:

     SANDBOX_NAMED           The profile argument specifies a sandbox profile
                             named by one of the constants given in the
                             AVAILABLE PROFILES section below.
```

sandbox_init (cont.)

```
AVAILABLE PROFILES
     The following are brief descriptions of each available profile.  Keep in
     mind that sandbox(7) restrictions are typically enforced at resource
     acquisition time.

     kSBXProfileNoInternet              TCP/IP networking is prohibited.

     kSBXProfileNoNetwork               All sockets-based networking is pro-
                                        hibited.

     kSBXProfileNoWrite                 File system writes are prohibited.

     kSBXProfileNoWriteExceptTemporary  File system writes are restricted to
                                        the temporary folder /var/tmp and the
                                        folder specified by the confstr(3)
                                        configuration variable _CS_DAR-
                                        WIN_USER_TEMP_DIR.

     kSBXProfilePureComputation         All operating system services are pro-
                                        hibited.
```

/usr/include/sandbox.h

```
/*
 * Available Sandbox profiles.
 */

/* TCP/IP networking is prohibited. */
extern const char kSBXProfileNoInternet[];

/* All sockets-based networking is prohibited. */
extern const char kSBXProfileNoNetwork[];

/* File system writes are prohibited. */
extern const char kSBXProfileNoWrite[];

/* File system writes are restricted to temporary folders /var/tmp and
 * confstr(_CS_DARWIN_USER_DIR, ...).
 */
extern const char kSBXProfileNoWriteExceptTemporary[];

/* All operating system services are prohibited. */
extern const char kSBXProfilePureComputation[];
```

Too lazy for IDA

```
fluffy:tmp dion$ cat /tmp/dump.c
#include <stdio.h>
#include <sandbox.h>
main() { printf("%s\n", kSBXProfileNoInternet); }
fluffy:tmp dion$ gcc -o /tmp/dump /tmp/dump.c
fluffy:tmp dion$ /tmp/dump
no-internet
```

/usr/include/sandbox.h

```
#ifdef __APPLE_API_PRIVATE

/* The following flags are reserved for Mac OS X.  Developers should not
 * depend on their availability.
 */

/*
 * @define SANDBOX_NAMED_BUILTIN   The `profile' argument specifies the
 * name of a builtin profile that is statically compiled into the
 * system.
 */
#define SANDBOX_NAMED_BUILTIN   0x0002

/*
 * @define SANDBOX_NAMED_EXTERNAL   The `profile' argument specifies the
 * pathname of a Sandbox profile.  The pathname may be abbreviated: If
 * the name does not start with a `/' it is treated as relative to
 * /usr/share/sandbox and a `.sb' suffix is appended.
 */
#define SANDBOX_NAMED_EXTERNAL  0x0003
```

Existing profiles

```
fluffy:tmp dion$ ls /usr/share/sandbox/
awacsd.sb            ntpd.sb
bsd.sb               portmap.sb
cvmsCompAgent.sb     quicklookd-job-creation.sb
cvmsServer.sb        quicklookd.sb
fontmover.sb         sshd.sb
kadmind.sb           syslogd.sb
krb5kdc.sb           xgridagentd.sb
mDNSResponder.sb     xgridagentd_task_nobody.sb
mds.sb               xgridagentd_task_somebody.sb
mdworker.sb          xgridcontrollerd.sb
named.sb
```

Existing profiles

```
fluffy:tmp dion$ cat /usr/share/sandbox/named.sb

...

(deny default)
(allow process*)
(deny signal)
(allow sysctl-read)
(allow network*)

;; Allow named-specific files
(allow file-write* file-read-data file-read-metadata
  (regex "^(/private)?/var/run/named\\.pid$"
         "^/Library/Logs/named\\.log$"))

(allow file-read-data file-read-metadata
  (regex "^(/private)?/etc/rndc\\.key$"
         "^(/private)?/etc/resolv\\.conf$"
         "^(/private)?/etc/named\\.conf$"
         "^(/private)?/var/named/"))
```

Trying our hand at it

```
fluffy:tmp dion$ sandbox-exec -p'
(version 1)
(allow default)
(deny file-read-data
    (regex "^/private/tmp/sand-fixie$"))
' /bin/sh
sh-3.2$ echo "Sandy McGee" > /tmp/sand-fixie
sh-3.2$ ls -l /tmp/sand-fixie
-rw-r--r--  1 dion  wheel  12 Jan 16 12:49 /tmp/sand-fixie
sh-3.2$ cat /tmp/sand-fixie
cat: /tmp/sand-fixie: Operation not permitted
sh-3.2$ exit
exit
fluffy:tmp dion$ cat /tmp/sand-fixie
Sandy McGee
```

Questions so far...

What is the full language supported by those profiles?

Which operations may be restricted?

(attacker thoughts:)

How is the profile enforced?

Is this interpreted in the kernel?

Userspace

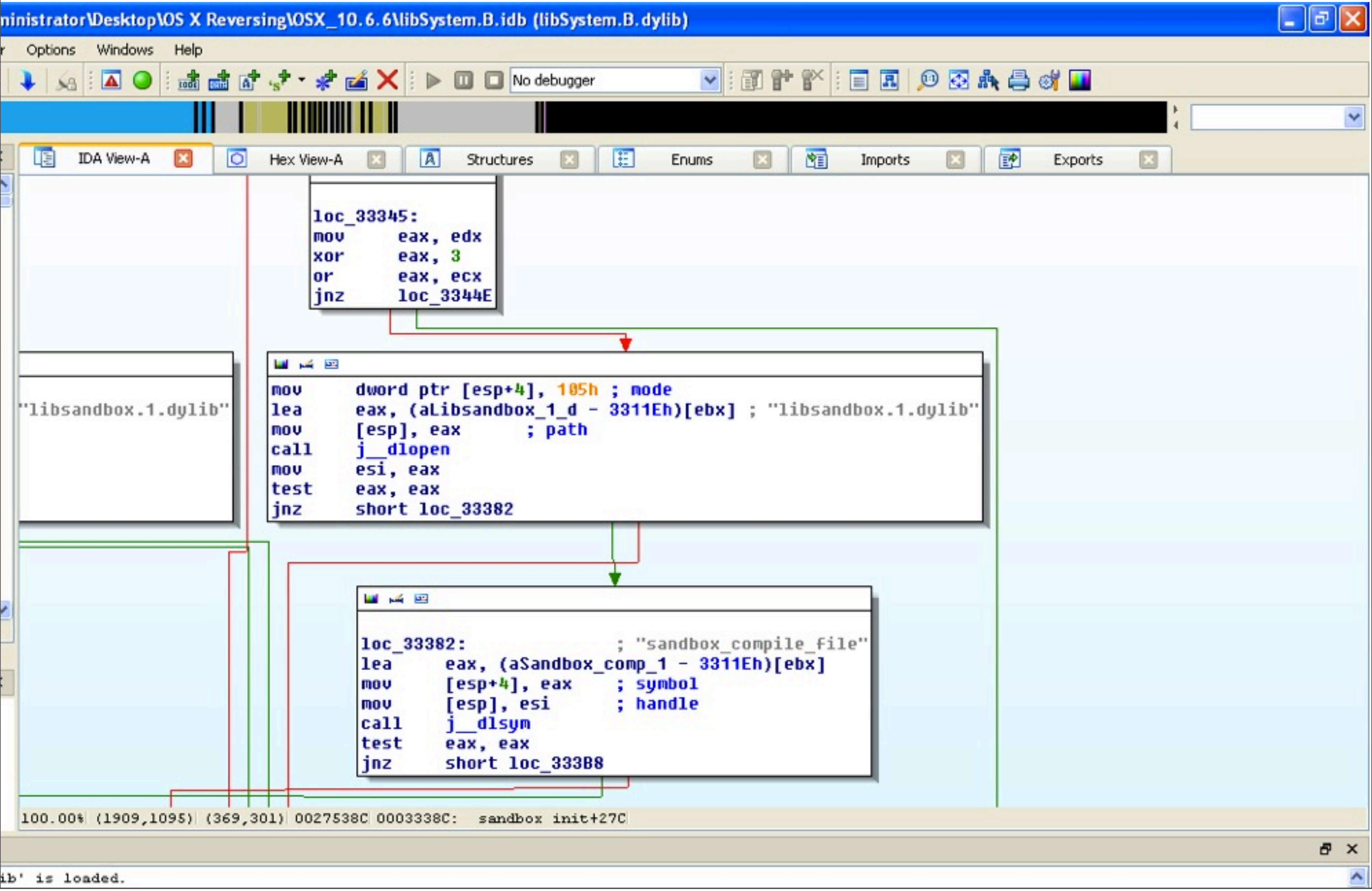
In an attempt to answer our questions, we'll start at the start: `sandbox_init`

Userspace

In an attempt to answer our questions, we'll start at the start: `sandbox_init`

```
fluffy:tmp dion$ cat i_call_sandbox_init.c
#include <sandbox.h>
int main(int argc, char *argv[]) {
  sandbox_init("", 0, NULL);
  return 0;
}
fluffy:tmp dion$ dyldinfo -lazy_bind i_call_sandbox_init
lazy binding information (from lazy_bind part of dyld info):
segment section address index dylib symbol
__DATA __la_symbol_ptr 0x100001038 0x0000 libSystem  _exit
__DATA __la_symbol_ptr 0x100001040 0x000C libSystem  _sandbox_init
```

sandbox_init



ministrator\Desktop\OS X Reversing\OSX_10.6.6\libSystem.B.idb (libSystem.B.dylib)
Options
Windows
Help
No debugger
IDA View-A
Hex View-A
Structures
Enums
Imports
Exports

```
loc_33345:
mov     eax, edx
xor     eax, 3
or      eax, ecx
jnz     loc_3344E
```

```
"libsandbox.1.dylib"
```

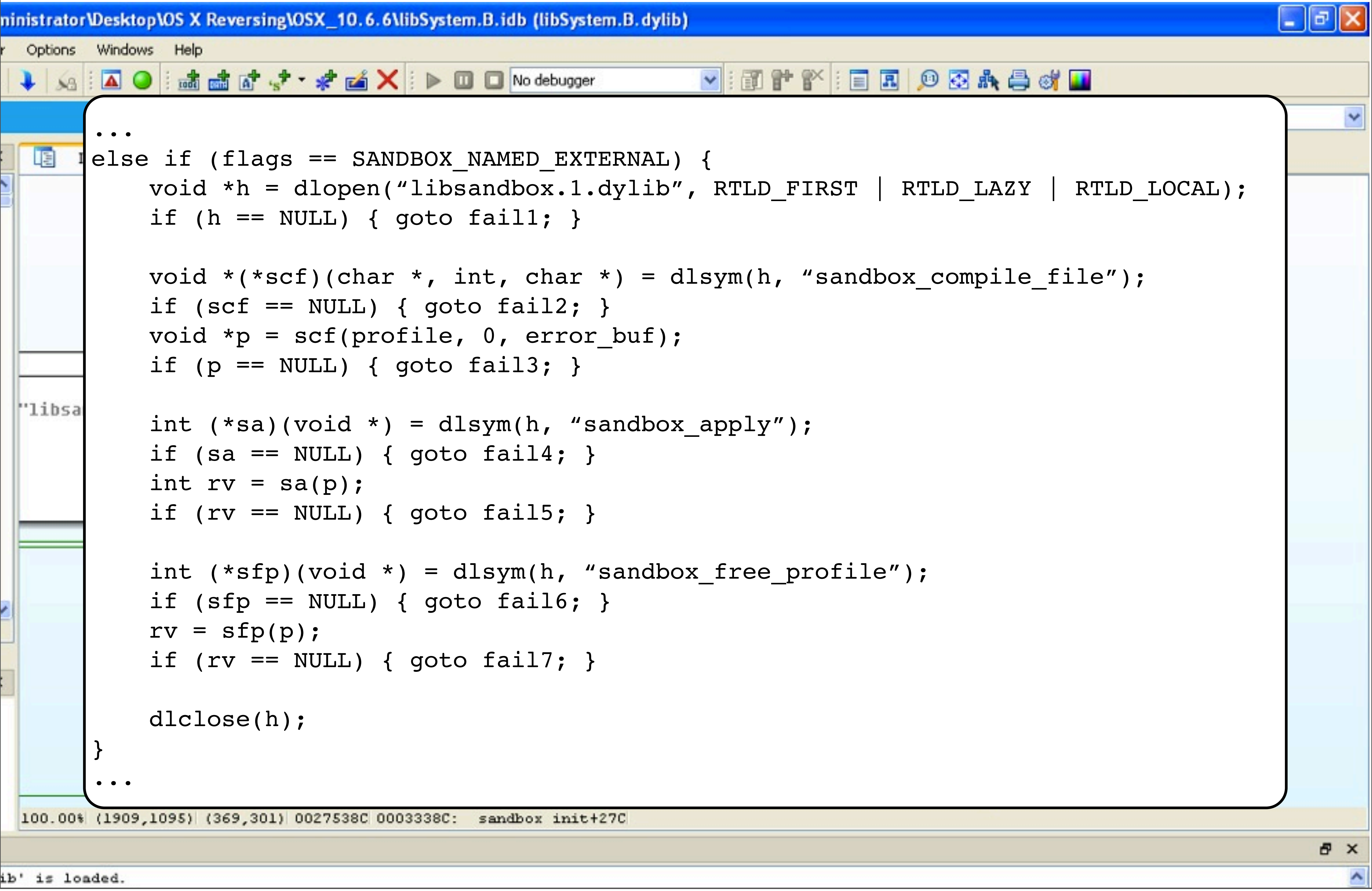
```
mov     dword ptr [esp+4], 105h ; mode
lea     eax, (aLibsandbox_1_d - 3311Eh)[ebx] ; "libsandbox.1.dylib"
mov     [esp], eax      ; path
call    j__dlopen
mov     esi, eax
test    eax, eax
jnz     short loc_33382
```

```
loc_33382:              ; "sandbox_compile_file"
lea     eax, (aSandbox_comp_1 - 3311Eh)[ebx]
mov     [esp+4], eax    ; symbol
mov     [esp], esi      ; handle
call    j__dlsym
test    eax, eax
jnz     short loc_333B8
```

```
100.00% (1909,1095) (369,301) 0027538C 0003338C: sandbox_init+27C
```

```
ib' is loaded.
```

sandbox_init

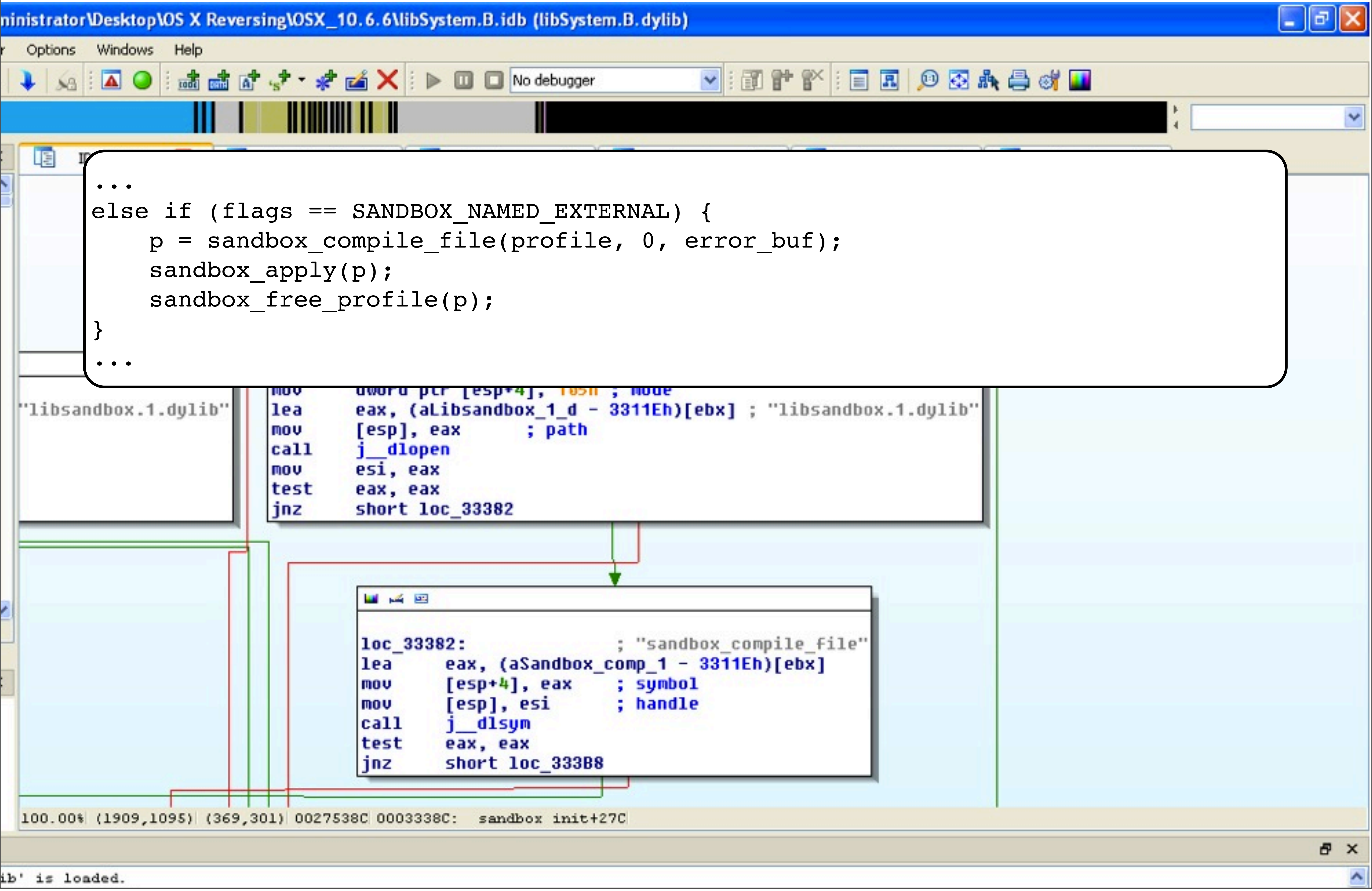

ministrator\Desktop\OS X Reversing\OSX_10.6.6\libSystem.B.idb (libSystem.B.dylib)
Options
Windows
Help
No debugger
"libsa

```
...
else if (flags == SANDBOX_NAMED_EXTERNAL) {
    void *h = dlopen(“libsandbox.1.dylib”, RTLD_FIRST | RTLD_LAZY | RTLD_LOCAL);
    if (h == NULL) { goto fail1; }

    void *(*scf)(char *, int, char *) = dlsym(h, “sandbox_compile_file”);
    if (scf == NULL) { goto fail2; }
    void *p = scf(profile, 0, error_buf);
    if (p == NULL) { goto fail3; }

    int (*sa)(void *) = dlsym(h, “sandbox_apply”);
    if (sa == NULL) { goto fail4; }
    int rv = sa(p);
    if (rv == NULL) { goto fail5; }

    int (*sfp)(void *) = dlsym(h, “sandbox_free_profile”);
    if (sfp == NULL) { goto fail6; }
    rv = sfp(p);
    if (rv == NULL) { goto fail7; }

    dlclose(h);
}
...
```

```
100.00% (1909,1095) (369,301) 0027538C 0003338C: sandbox_init+27C
```

```
ib' is loaded.
```

sandbox_init



```
...
else if (flags == SANDBOX_NAMED_EXTERNAL) {
    p = sandbox_compile_file(profile, 0, error_buf);
    sandbox_apply(p);
    sandbox_free_profile(p);
}
...
```

ninistrator\Desktop\OS X Reversing\OSX_10.6.6\libSystem.B.idb (libSystem.B.dylib)

```
"libsandbox.1.dylib"
```

```
mov     dword ptr [esp+4], 105h ; mode
lea     eax, (aLibsandbox_1_d - 3311Eh)[ebx] ; "libsandbox.1.dylib"
mov     [esp], eax      ; path
call    j__dlopen
mov     esi, eax
test    eax, eax
jnz     short loc_33382
```

```
loc_33382:              ; "sandbox_compile_file"
lea     eax, (aSandbox_comp_1 - 3311Eh)[ebx]
mov     [esp+4], eax    ; symbol
mov     [esp], esi      ; handle
call    j__dlsym
test    eax, eax
jnz     short loc_333B8
```

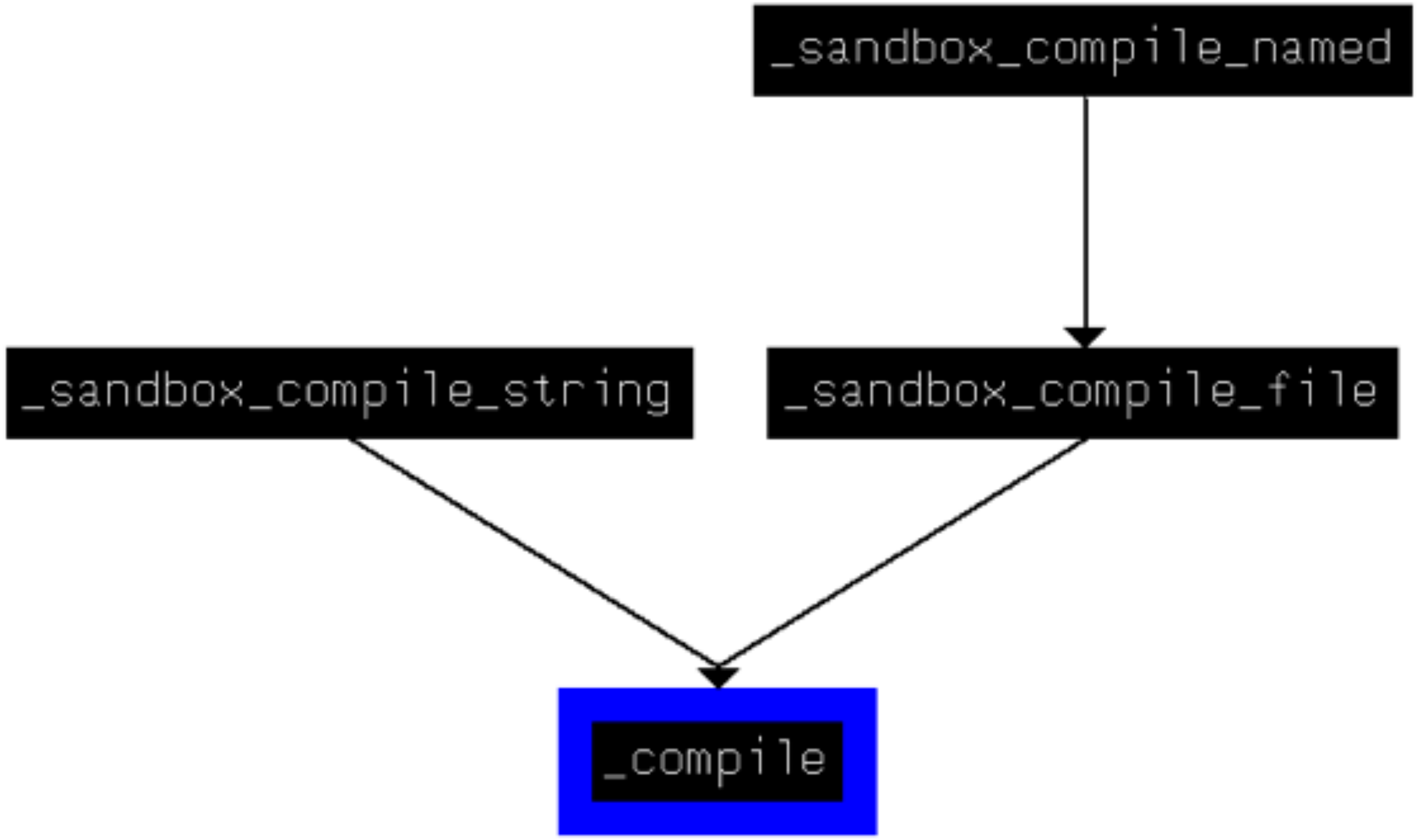
```
100.00% (1909,1095) (369,301) 0027538C 0003338C: sandbox_init+27C
```

```
ib' is loaded.
```

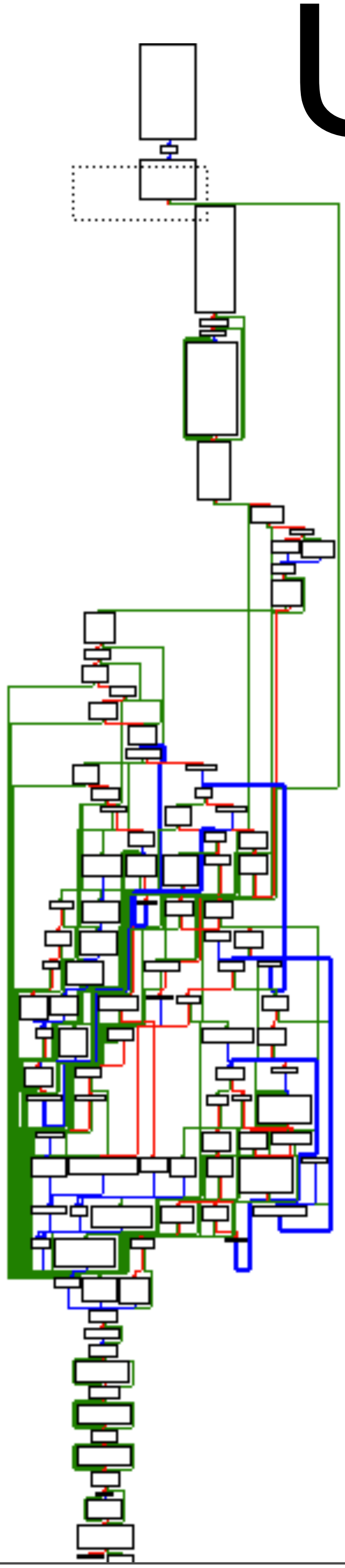
Userspace:
`sandbox_compile_file`

Next step is to open libsandbox.1.dylib in IDA and examine `sandbox_compile_file`

In IDA, the most interesting call is to `compile`



Userspace: `compile`



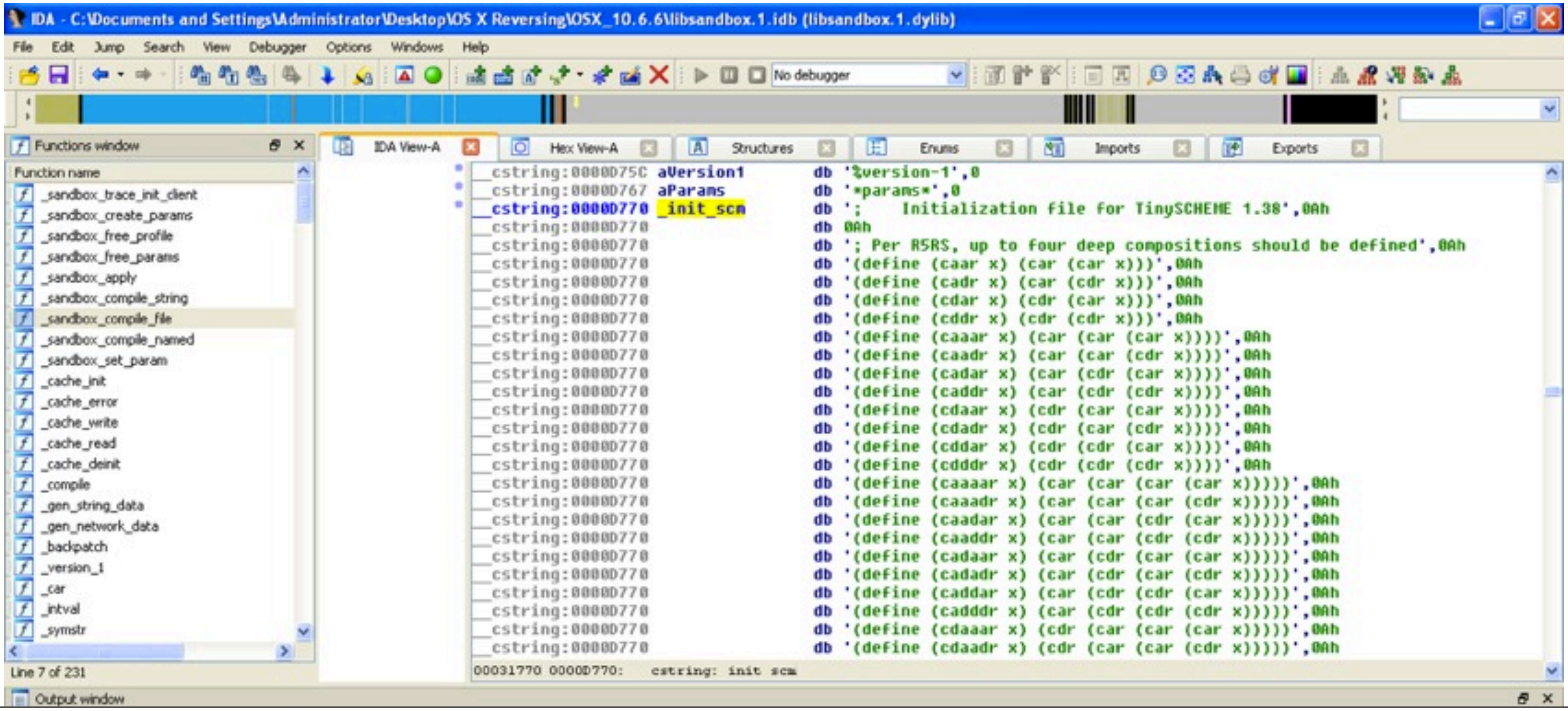
`compile` is a nice large-ish function with lots of logic.

Examining the first few basic blocks shows a call to `scheme_init_new`

This must be where the Scheme evaluation (the sandbox profiles are Scheme scripts) takes place...

Userspace: `compile`

Looking further into libsandbox.1.dylib reveals the init script is based on one from TinyScheme:



```
__cstring:0000D75C aVersion1        db '%version-1',0
__cstring:0000D767 aParams          db '*params*',0
__cstring:0000D770 _init_scm        db ';    Initialization file for TinySCHEME 1.38',0Ah
__cstring:0000D770                  db 0Ah
__cstring:0000D770                  db '; Per R5RS, up to four deep compositions should be defined',0Ah
__cstring:0000D770                  db '(define (caar x) (car (car x)))',0Ah
__cstring:0000D770                  db '(define (cadr x) (car (cdr x)))',0Ah
__cstring:0000D770                  db '(define (cdar x) (cdr (car x)))',0Ah
__cstring:0000D770                  db '(define (cddr x) (cdr (cdr x)))',0Ah
__cstring:0000D770                  db '(define (caaar x) (car (car (car x))))',0Ah
__cstring:0000D770                  db '(define (caadr x) (car (car (cdr x))))',0Ah
__cstring:0000D770                  db '(define (cadar x) (car (cdr (car x))))',0Ah
__cstring:0000D770                  db '(define (caddr x) (car (cdr (cdr x))))',0Ah
__cstring:0000D770                  db '(define (cdaar x) (cdr (car (car x))))',0Ah
__cstring:0000D770                  db '(define (cdadr x) (cdr (car (cdr x))))',0Ah
__cstring:0000D770                  db '(define (cddar x) (cdr (cdr (car x))))',0Ah
__cstring:0000D770                  db '(define (cdddr x) (cdr (cdr (cdr x))))',0Ah
__cstring:0000D770                  db '(define (caaaar x) (car (car (car (car x)))))',0Ah
__cstring:0000D770                  db '(define (caaadr x) (car (car (car (cdr x)))))',0Ah
__cstring:0000D770                  db '(define (caadar x) (car (car (cdr (car x)))))',0Ah
__cstring:0000D770                  db '(define (caaddr x) (car (car (cdr (cdr x)))))',0Ah
__cstring:0000D770                  db '(define (cadaar x) (car (cdr (car (car x)))))',0Ah
__cstring:0000D770                  db '(define (cadadr x) (car (cdr (car (cdr x)))))',0Ah
__cstring:0000D770                  db '(define (caddar x) (car (cdr (cdr (car x)))))',0Ah
__cstring:0000D770                  db '(define (cadddr x) (car (cdr (cdr (cdr x)))))',0Ah
__cstring:0000D770                  db '(define (cdaaar x) (cdr (car (car (car x)))))',0Ah
__cstring:0000D770                  db '(define (cdaadr x) (cdr (car (car (cdr x)))))',0Ah
00031770 0000D770:    cstring: init scm
```

Userspace: `compile`

Another `scheme_load_string` call is passed a script defining the functions and macros used in profile scripts.

Apple calls this the SBPL -- SandBox Profile Language.

Userspace: `compile`

```
;;;;;; Sandbox Profile Language stub
;;; This stub is loaded before the sandbox profile is evaluated.  When version
;;; is called, the SBPL prelude and the appropriate SBPL version library are
;;; loaded, which together implement the profile language.  These modules build
;;; a *rules* table that maps operation codes to lists of rules of the form
;;;   RULE -> TEST | JUMP
;;;   TEST -> (filter action . modifiers)
;;;   JUMP -> (#f . operation)
;;; The result of an operation is decided by the first test with a filter that
;;; matches.  Filter can be #t, in which case the test always matches.  A jump
;;; causes evaluation to continue with the rules for another operation.  The
;;; last rule in the list must either be a test that always matches or a jump.
```

Userspace: SBPL

```
;;;;;; Sandbox Profile Language stub
;;; This stub is loaded before the sandbox profile is evaluated.  When version
;;; is called, the SBPL prelude and the appropriate SBPL version library are
;;; loaded, which together implement the profile language.  These modules build
;;; a *rules* table that maps operation codes to lists of rules of the form
;;;   RULE -> TEST | JUMP
;;;   TEST -> (filter action . modifiers)
;;;   JUMP -> (#f . operation)
;;; The result of an operation is decided by the first test with a filter that
;;; matches.  Filter can be #t, in which case the test always matches.  A jump
;;; causes evaluation to continue with the rules for another operation.  The
;;; last rule in the list must either be a test that always matches or a jump.
```

rules

```
#( ((#t deny))
   ((#f . 0))
   ((#f . 1))
   (((filter path 0 regex
             ^/dev/dtracehelper$) allow)
    (#f . 1))

   ((#f . 1))
   (((filter path 0 regex
             ^/dev/null$
             ^(/private)?/var/run/syslog$
             ^/dev/u?random$
             ^/dev/autofs_nowait$
             ^/dev/dtracehelper$
             ...) allow)
    (#f . 4))
...
```

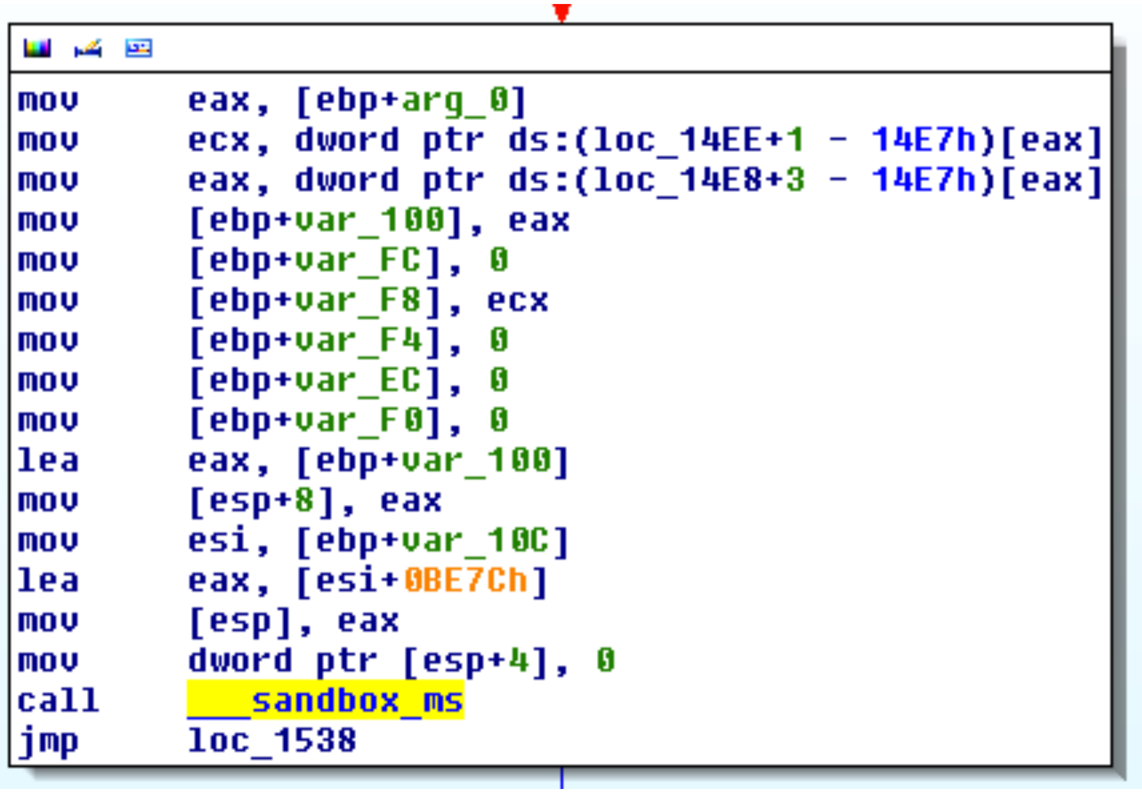
rules

```
#( ((#t deny))
   ((#f . 0))
   ((#f . 1))
   (((filter path 0 regex
            ^/dev/dtracehelper$) allow)
    (#f . 1))

   ((#f . 1))
   (((filter path 0 regex
            ^/dev/null$
            ^(/private)?/var/run/syslog$
            ^/dev/u?random$
            ^/dev/autofs_nowait$
            ^/dev/dtracehelper$
            ...) allow)
    (#f . 4))
...
```

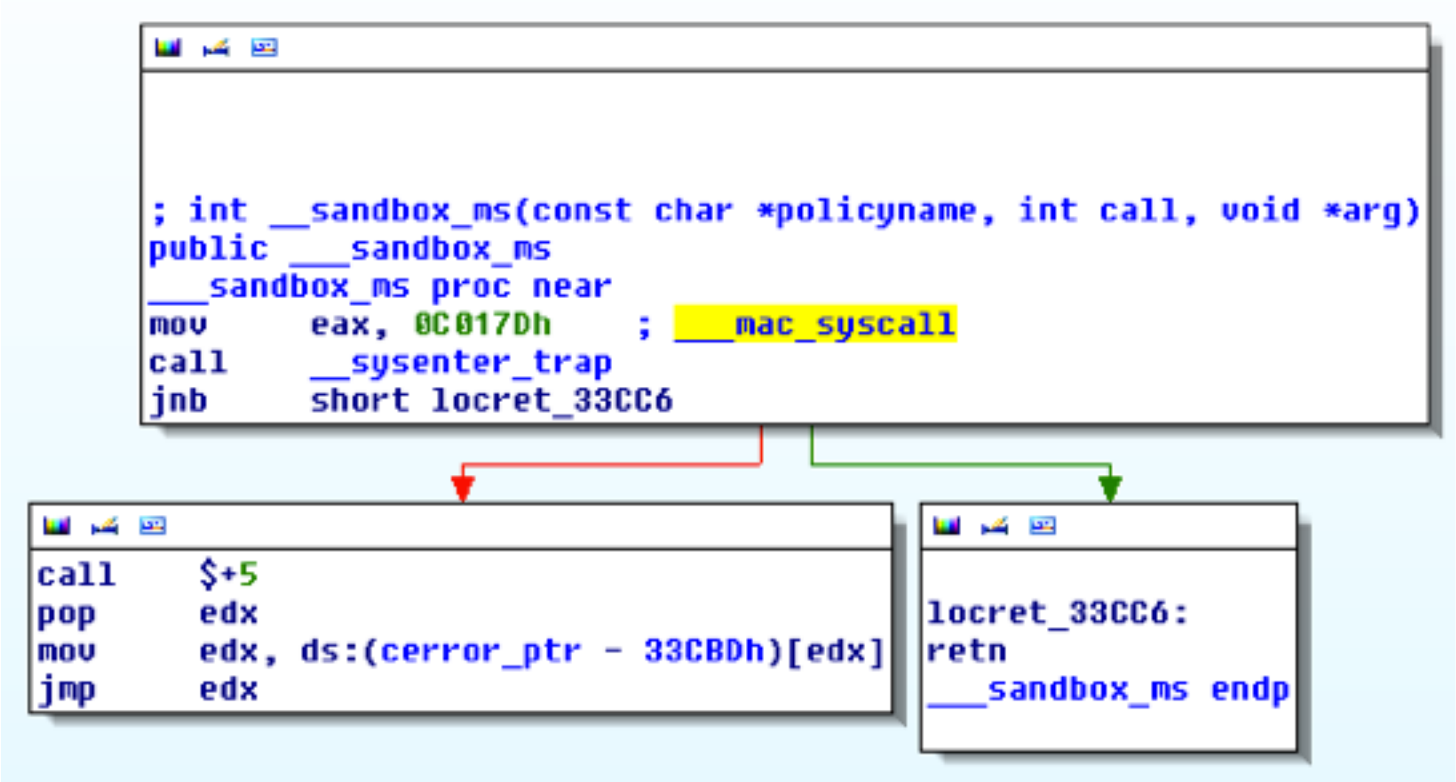
```
0: deny()
1: goto 0
2: goto 1
3: if regex.match(“^/dev/dtracehelper$”):
       allow()
   else:
       goto 1
4: goto 1
5: if regex.match(“^/dev/null$”) or \
      regex.match(“^(/private)?/var/run/syslog$”) o
      regex.match(“^/dev/u?random$”) or \
      regex.match(“^/dev/autofs_nowait$”) or \
      regex.match(“^/dev/dtracehelper$”) or \
      ...:
       allow()
   else:
       goto 4
...
```

Userspace: `sandbox_apply`

Following the magic of `compile`, `sandbox_apply` is called -- in this function we



```
mov     eax, [ebp+arg_0]
mov     ecx, dword ptr ds:(loc_14EE+1 - 14E7h)[eax]
mov     eax, dword ptr ds:(loc_14E8+3 - 14E7h)[eax]
mov     [ebp+var_100], eax
mov     [ebp+var_FC], 0
mov     [ebp+var_F8], ecx
mov     [ebp+var_F4], 0
mov     [ebp+var_EC], 0
mov     [ebp+var_F0], 0
lea     eax, [ebp+var_100]
mov     [esp+8], eax
mov     esi, [ebp+var_10C]
lea     eax, [esi+0BE7Ch]
mov     [esp], eax
mov     dword ptr [esp+4], 0
call    ___sandbox_ms
jmp     loc_1538
```



```
; int __sandbox_ms(const char *policyname, int call, void *arg)
public ___sandbox_ms
___sandbox_ms proc near
mov     eax, 0C017Dh    ; ___mac_syscall
call    __sysenter_trap
jnb     short locret_33CC6
```

```
call    $+5
pop     edx
mov     edx, ds:(cerror_ptr - 33CBDh)[edx]
jmp     edx
```

```
locret_33CC6:
retn
___sandbox_ms endp
```

Kernel Entry

We left userspace through `sandbox_apply` (via `__sandbox_ms`)

`__sandbox_ms` entered the kernel syscall 0x017D which is, as IDA commented, the `mac_syscall` syscall

Kernel Entry

`__mac_syscall` is implemented on line 2119 of `xnu-1504.7.4/security/mac_base.c`

Hooray, some open source!

It turns out the XNU kernel contains a port of Robert Watson's TrustedBSD MAC framework for FreeBSD.

TrustedBSD

TrustedBSD provides hooks for kernel extensions.

An extension registers a large list of function pointers with TrustedBSD.

These FPs are called for most syscalls and for many kernel object lifecycle events (vnode created/destroyed).

It’s fairly complex. More recently, see Capsicum.

Sandbox.kext

Sandbox.kext is the kernel extension that registers with TrustedBSD to enforce the Sandbox profile semantics.

Finally! We made it.

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2[] op_table

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2[] op_table

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2[] op_table

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[0]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))
```

0x0012 * 8 = 0x90

```
fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[0]
```

0 = “default” operation

```
ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

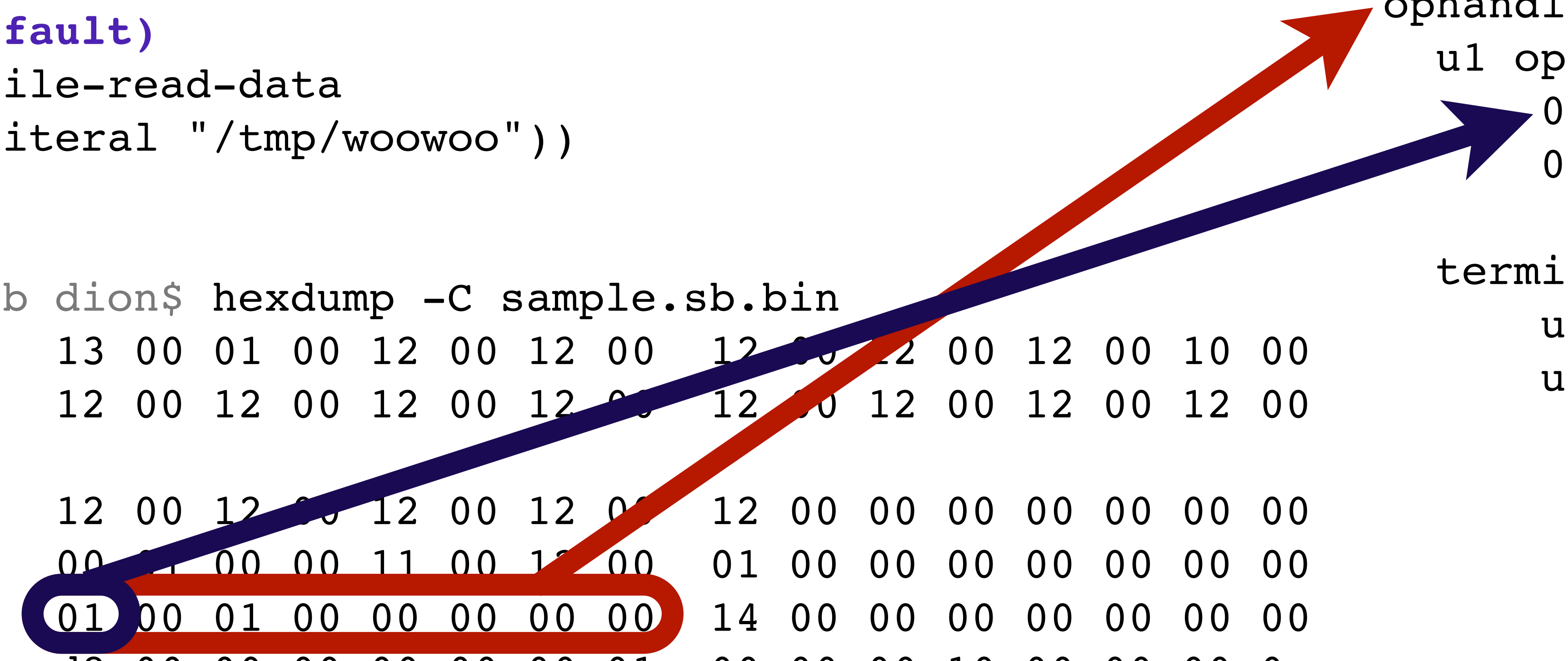
sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))
```



```
fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
     01: terminal
     00: non-terminal

  terminal:
     u1: padding
     u1: result
       00: allow
       01: deny
       02: allow-with-log
       03: deny-with-log

  non-terminal:
     u1 filter
         01: path
         02: xattr
         03: file-mode
         04: mach-global
         05: mach-local
         06: socket-local
         07: socket-remote
         08: signal
     u2 filter_arg
     u2 transition_matched
     u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
   u1 opcode
         01: terminal
         00: non-terminal

   terminal:
         u1: padding
         u1: result
            00: allow
            01: deny
            02: allow-with-log
            03: deny-with-log

   non-terminal:
         u1 filter
               01: path
               02: xattr
               03: file-mode
               04: mach-global
               05: mach-local
               06: socket-local
               07: socket-remote
               08: signal
         u2 filter_arg
         u2 transition_matched
         u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))
```

0x0010 * 8 = 0x80

```
fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[5]
```

5 = “file-read-data” operation

```
ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

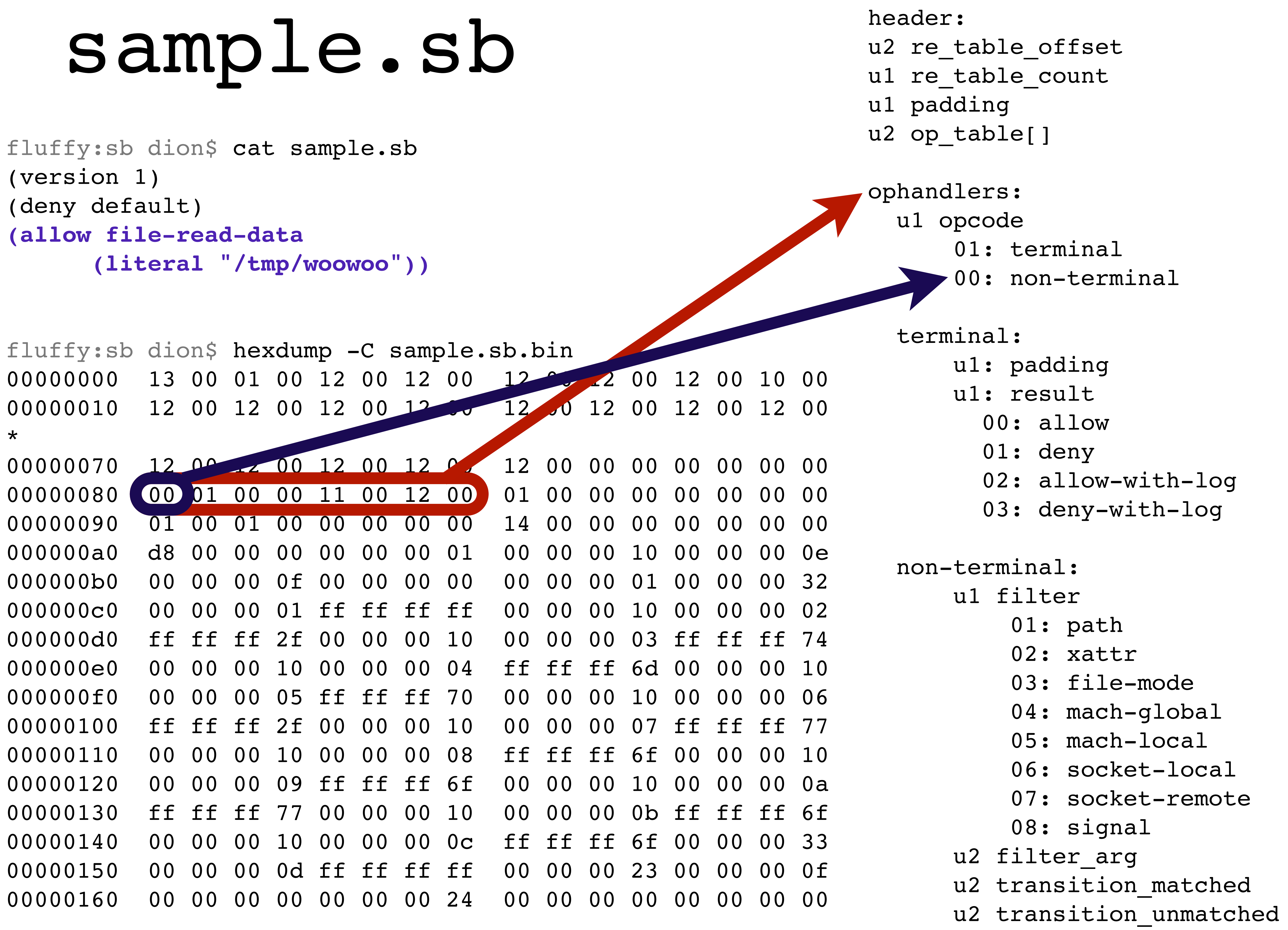
sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.b

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))
```



```
fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
        01: path
        02: xattr
        03: file-mode
        04: mach-global
        05: mach-local
        06: socket-local
        07: socket-remote
        08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

sample.sb

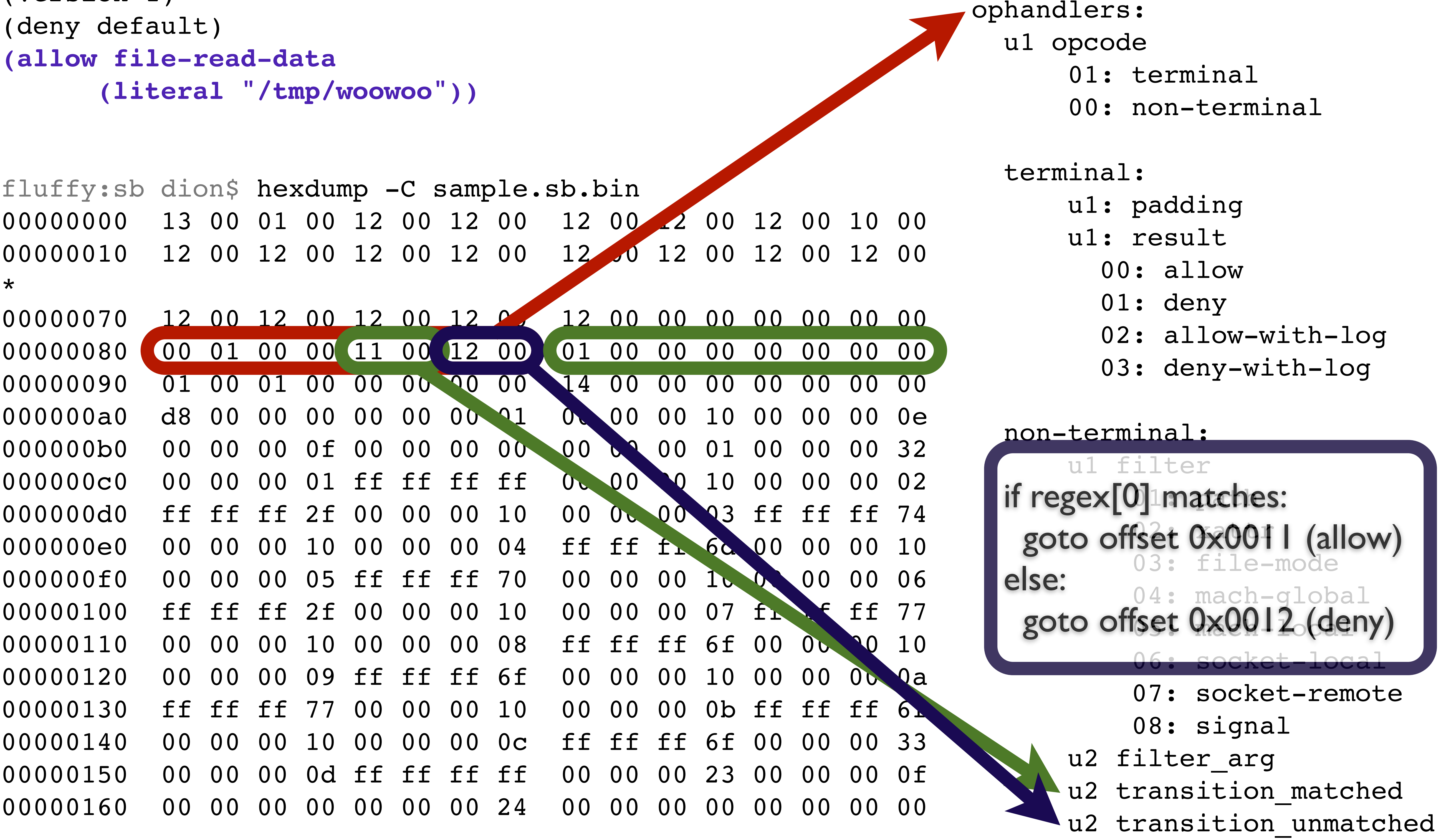
```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))


fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
  u1 opcode
      01: terminal
      00: non-terminal

  terminal:
      u1: padding
      u1: result
        00: allow
        01: deny
        02: allow-with-log
        03: deny-with-log

  non-terminal:
      u1 filter
          01: path
          02: xattr
          03: file-mode
          04: mach-global
          05: mach-local
          06: socket-local
          07: socket-remote
          08: signal
      u2 filter_arg
      u2 transition_matched
      u2 transition_unmatched
```

regex[0]

sample.sb

```
fluffy:sb dion$ cat sample.sb
(version 1)
(deny default)
(allow file-read-data
      (literal "/tmp/woowoo"))
```



```
fluffy:sb dion$ hexdump -C sample.sb.bin
00000000  13 00 01 00 12 00 12 00  12 00 12 00 12 00 10 00
00000010  12 00 12 00 12 00 12 00  12 00 12 00 12 00 12 00
*
00000070  12 00 12 00 12 00 12 00  12 00 00 00 00 00 00 00
00000080  00 01 00 00 11 00 12 00  01 00 00 00 00 00 00 00
00000090  01 00 01 00 00 00 00 00  14 00 00 00 00 00 00 00
000000a0  d8 00 00 00 00 00 00 01  00 00 00 10 00 00 00 0e
000000b0  00 00 00 0f 00 00 00 00  00 00 00 01 00 00 00 32
000000c0  00 00 00 01 ff ff ff ff  00 00 00 10 00 00 00 02
000000d0  ff ff ff 2f 00 00 00 10  00 00 00 03 ff ff ff 74
000000e0  00 00 00 10 00 00 00 04  ff ff ff 6d 00 00 00 10
000000f0  00 00 00 05 ff ff ff 70  00 00 00 10 00 00 00 06
00000100  ff ff ff 2f 00 00 00 10  00 00 00 07 ff ff ff 77
00000110  00 00 00 10 00 00 00 08  ff ff ff 6f 00 00 00 10
00000120  00 00 00 09 ff ff ff 6f  00 00 00 10 00 00 00 0a
00000130  ff ff ff 77 00 00 00 10  00 00 00 0b ff ff ff 6f
00000140  00 00 00 10 00 00 00 0c  ff ff ff 6f 00 00 00 33
00000150  00 00 00 0d ff ff ff ff  00 00 00 23 00 00 00 0f
00000160  00 00 00 00 00 00 00 24  00 00 00 00 00 00 00 00
```

```
header:
u2 re_table_offset
u1 re_table_count
u1 padding
u2 op_table[]

ophandlers:
   u1 opcode
         01: terminal
         00: non-terminal

   terminal:
         u1: padding
         u1: result
            00: allow
            01: deny
            02: allow-with-log
            03: deny-with-log

   non-terminal:
         u1 filter
            01: path
            02: xattr
            03: file-mode
            04: mach-global
            05: mach-local
            06: socket-local
            07: socket-remote
            08: signal
         u2 filter_arg
         u2 transition_matched
         u2 transition_unmatched
```

```
if regex[0] matches:
  goto offset 0x0011 (allow)
else:
  goto offset 0x0012 (deny)
```

AppleMatch.kext

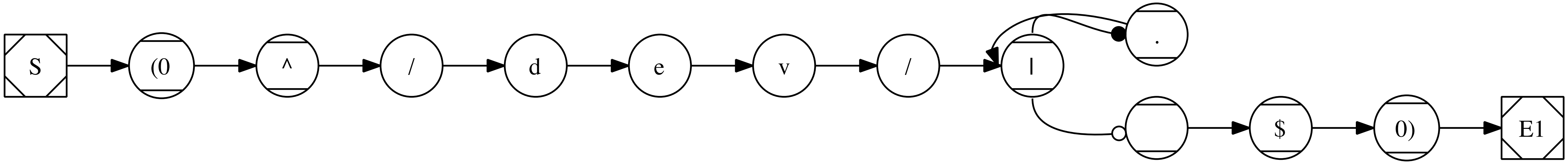
One of the most popular filters for Sandbox profiles is the pathname regular expression match.

This means there needs to be a regex engine in the kernel(!)

AppleMatch.kext provides this.

AppleMatch.kext

The regular expression are compiled into NFAs in userspace first:



Is the resulting NFA from “`^/dev/.*$`”

Utilities

libsandcall: wrapping the OS X subsyscalls

sbsnarf: convert a Scheme profile into a binary profile

resnarf: extract all regexs from a binary profile

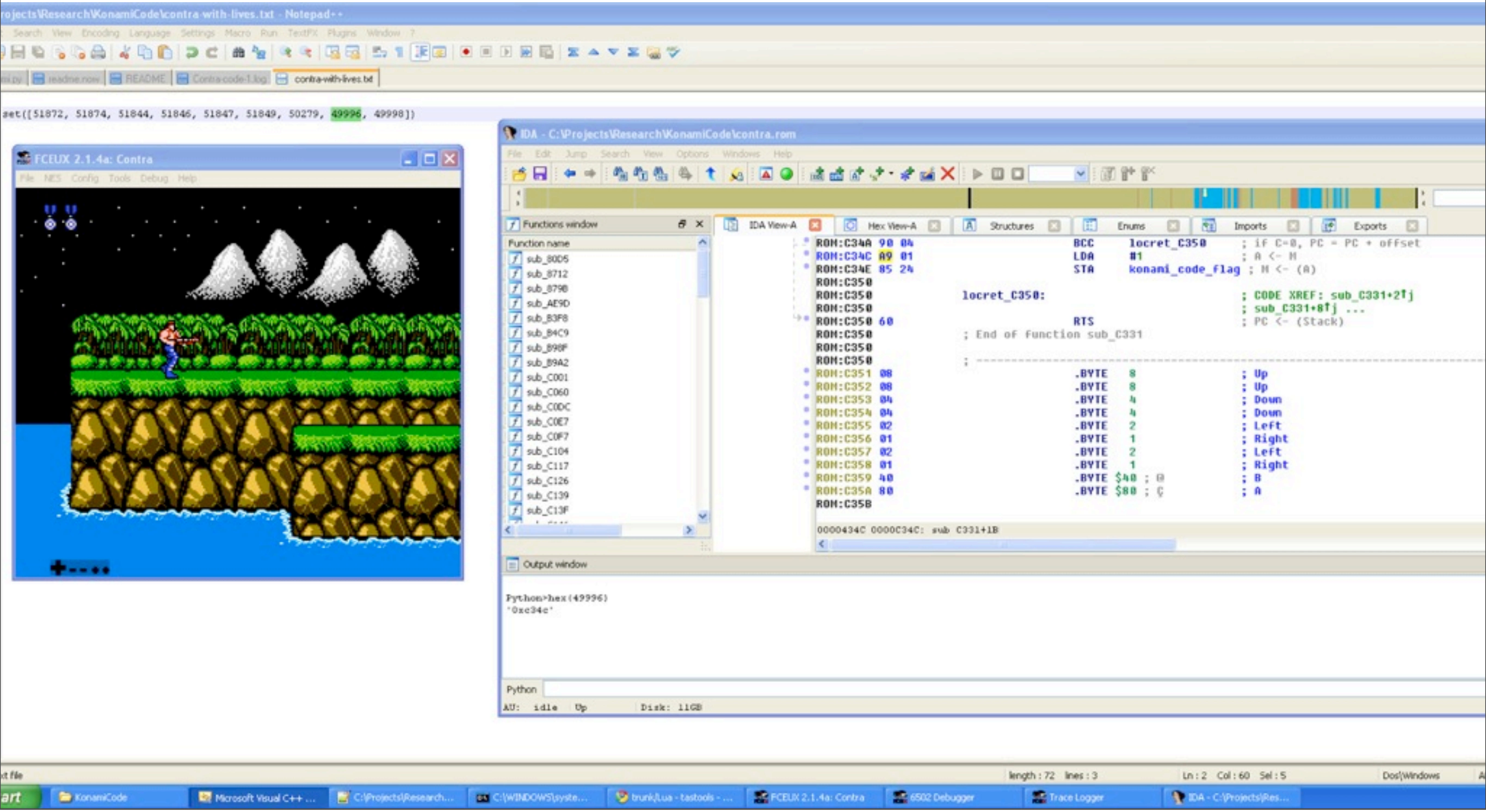
apple-scheme: dlopen()'s libsandbox and uses the embedded TinyScheme to evaluate scripts using their patched interpreter

re2dot: converts a compiled regex to graphviz dot of the NFA

sbdis: disassemble a binary profile to human readable form

The iOS 4.2.1 AppStore (“containter”) profile

Questions?



```
set([51872, 51874, 51844, 51846, 51847, 51849, 50279, 49996, 49998])
```

```
ROM:C34A 90 04                      BCC     locret_C350      ; if C=0, PC = PC + offset
ROM:C34C A9 01                      LDA     #1               ; A <- M
ROM:C34E 85 24                      STA     konami_code_flag ; M <- (A)
ROM:C350
ROM:C350              locret_C350:                           ; CODE XREF: sub_C331+2↑j
ROM:C350                                                     ; sub_C331+8↑j ...
ROM:C350 60                         RTS                      ; PC <- (Stack)
ROM:C350              ; End of function sub_C331
ROM:C350
ROM:C350              ; ---------------------------------------------------------------
ROM:C351 08                         .BYTE   8                ; Up
ROM:C352 08                         .BYTE   8                ; Up
ROM:C353 04                         .BYTE   4                ; Down
ROM:C354 04                         .BYTE   4                ; Down
ROM:C355 02                         .BYTE   2                ; Left
ROM:C356 01                         .BYTE   1                ; Right
ROM:C357 02                         .BYTE   2                ; Left
ROM:C358 01                         .BYTE   1                ; Right
ROM:C359 40                         .BYTE $40 ; @            ; B
ROM:C35A 80                         .BYTE $80 ; Ç            ; A
ROM:C35B

0000434C 0000C34C: sub C331+1B
```

```
Python>hex(49996)
'0xc34c'
```