

Apple's Sandbox Guide

v0.1

Table of Contents

1 – Introduction	3
2 – What are we talking about?.....	3
3 – How can it be used or implemented?	4
4 - Anatomy of a profile	5
5 - Commands Reference	5
5.1 – Actions	5
5.2 – Operations	5
5.3 – Filters	24
5.4 – Modifiers	27
5.5 - Other keywords.....	28
6 – Special hardcoded cases	28
7 – A sample profile	28

1 – Introduction

Apple's sandbox technology was introduced in Leopard version of Mac OS X. A few years have passed and documentation is still lacking.

Dionysus Blazakis published a great paper and presentation where he reversed the implementation of this feature. Otherwise, the other references are Apple's own sandbox profiles and some attempts by other users to create new profiles.

This document tries to help in closing this gap, by documenting the available options and actions available in this technology.

Due to its nature, it is an incomplete and work in progress type of document. It is possible that mistakes and wrong assumptions exist so all contributions and fixes are more than welcome!

You can contact me at reverser@put.as. The latest version is always available at <http://reverse.put.as>.

Enjoy,

fG!

2 – What are we talking about?

Using the definition from Apple's website we have:

“Sandboxing protects the system by limiting the kinds of operations an application can perform, such as opening documents or accessing the network. Sandboxing makes it more difficult for a security threat to take advantage of an issue in a specific application to affect the greater system.”

The implementation found in Mac OS X can limit the following type of operations:

- File: read, write, with many different operations
- IPC: posix and sysv
- Mach
- Network: inbound, outbound
- Process: execution, fork
- Signals
- Sysctl
- System

It has a rich set of operations that can help to improve the security of applications and mitigate potential attacks, especially on network-enabled applications such as web browsers, Flash or applications that process untrusted input such as pdf, word/excel/powerpoint documents, etc.

3 – How can it be used or implemented?

There are two ways to use this feature. The first is to execute an application within a sandbox, using the command “sandbox-exec”. The second one is to use the sandbox feature inside your code, calling “sandbox_init” and using one of the pre-defined profiles (also available to sandbox-exec).

These profiles are:

- kSBXProfileNoInternet : TCP/IP networking is prohibited.
- kSBXProfileNoNetwork : All sockets-based networking is prohibited.
- kSBXProfileNoWrite : File system writes are prohibited.
- kSBXProfileNoWriteExceptTemporary : File system writes are restricted to the temporary folder /var/tmp and the folder specified by theconfstr(3) configuration variable _CS_DARWIN_USER_TEMP_DIR.
- kSBXProfilePureComputation : All operating system services are prohibited.

With sandbox-exec command we can use custom profiles, written in SBPL – Sandbox Profile Language (a “Scheme embedded domain specific language” using Dion’s definition). You can find some examples at “/usr/share/sandbox”. These are currently used to sandbox some system daemons. The syntax for this command is:

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

The option -f should be used for loading custom profiles. Either you can use the absolute path to the profile or just the name of the profile, as long it is located at the following folders:

/Library/Sandbox/Profiles

/System/Library/Sandbox/Profiles

/usr/share/sandbox

Full path example:

```
sandbox-exec -f /usr/share/sandbox/bsd.sb /bin/lis
```

Name example:

```
sandbox-exec -f bsd /bin/lis
```

where bsd.sb is located at /usr/share/sandbox

The -n is used to load one of the pre-defined profiles. Instead of using the same names as in sandbox_init, you should use these:

Sandbox_init	Sandbox-exec
kSBXProfileNoInternet	no-internet
kSBXProfileNoNetwork	no-network
kSBXProfileNoWriteExceptTemporary	no-write-except-temporary
kSBXProfileNoWrite	no-write
kSBXProfilePureComputation	pure-computation

4 - Anatomy of a profile

A profile is composed of actions on operations, modifiers, filters, options and (optionally) comments. To simplify, I will call these commands except for comments.

The first thing to be configured is the version of SBPL. For now there's only version 1 so this should be common to all scripts.

Then you can configure the logging option, with the “debug” command. Two options are available, all, which should log all operations (allowed or not), and deny, which logs only denied operations.

Other profiles can be included using the import command.

The default action can be configured either to deny or to allow. This will depend on the type of profile you are interested to achieve.

Comments should start with semicolon (;) and are valid until the end of the line.

5 - Commands Reference

All commands are enclosed into parenthesis. In each example, the “\$” symbol means command execution at the shell.

5.1 – Actions

There are two available actions, allow or deny.

Actions apply only to the operations defined below.

Syntax:

(action operation [filter modifiers])

Example:

- (deny default)

All operations will be denied unless explicitly allowed (default is an operation). This is a whitelist mode.

- (allow default)

All operations will be allowed unless explicitly denied. In this case we have a blacklist mode.

5.2 – Operations

As previously described, the sandbox supports different type of operations. Almost all operations have global and granular modes. Global means that the whole category of operation can be configured. For example, the “file*” operation will control all type of file related operations. But we can also be more granular and allow file reads and deny file writes (and even be a little more specific in these two operations).

The following table shows the global operations, including the ones without granular modes.

Default	File*	Ipc*	Mach*	Network*	Process*
Signal	Sysctl*	System*	Job-creation	Mach-per-user-lookup	

Operations can have filters and modifiers. Modifiers apply to all operations (except the mach ones) while filters don't.

All the available operations are now described.

Default

As the name implies, this is the default action if no other operation matches. It doesn't matter where this operation is configured, either at the beginning or the end of the profile. The engine will only hit the default operation if no explicit match can be found. Searching for operations will stop when the first explicit match is hit. This means that a deny action followed by an allow action to the same operation and target will never trigger the allow action, it will always be denied.

Syntax:	(action default [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow default)

If you wish to create a blacklist type of profile.

- (deny default)

To create a whitelist profile.

- (deny default (with no-log))

To create a whitelist profile without logging.

File*

This operation will control file related operations such as reads, writes, extended attributes, etc.

Syntax:	(action file* [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file*)

This will deny all file related operations to any file.

- (deny file* (literal "/mach_kernel"))

This will deny all file related operations that have /mach_kernel as target.

File-chroot

Control whether the target should be allowed or not to chroot() into the specified directory.

Syntax:	(action file-chroot [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode

Modifiers: send-signal no-log

Example:

- (deny file-chroot (literal "/"))

```
# sandbox-exec -f ls2 /usr/sbin/chroot -g nobody / /bin/ls
```

```
chroot: /: Operation not permitted
```

Log output:

```
Sep 2 18:45:02 macbox sandboxd[40841]: chroot(40840) deny file-chroot /
```

File-ioctl

Determine whether the target can perform the ioctl operation.

Warning: Since ioctl data is opaque from the standpoint of the MAC framework, and since ioctls can affect many aspects of system operation, policies must exercise extreme care when implementing access control checks.

Syntax: (action file-ioctl [filter] [modifier])

Actions: allow deny

Filters: path file-mode

Modifiers: send-signal no-log

Example:

- (allow file-ioctl (literal "/dev/dtracehelper"))

File-read*

Controls all available read operations described below.

Syntax: (action file-read* [filter] [modifier])

Actions: allow deny

Filters: path file-mode

Modifiers: send-signal no-log

Example:

- (deny file-read* (literal "/mach_kernel"))

```
$ sandbox-exec -f ls2 cat /mach_kernel
```

```
cat: /mach_kernel: Operation not permitted
```

Log output:

```
Sep 2 00:13:12 macbox sandboxd[24486]: cat(24485) deny file-read-data /mach_kernel
```

```
$ sandbox-exec -f ls2 ls /mach_kernel
```

```
ls: /mach_kernel: Operation not permitted
```

Log output:

```
Sep 2 00:13:46 macbox sandboxd[24498]: ls(24504) deny file-read-metadata /mach_kernel
```

```
$ sandbox-exec -f ls2 xattr /mach_kernel
```

```
xattr: No such file: /mach_kernel
```

Log output:

```
Sep 2 00:13:38 macbox sandboxd[24498]: Python(24497) deny file-read-xattr /mach_kernel
Sep 2 00:13:38 macbox sandboxd[24498]: Python(24497) deny file-read-metadata
/mach_kernel
```

File-read-data

Give or refuse read access to the contents of the target file.

Syntax:	(action file-read-data [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-read-data (literal "/mach_kernel"))

```
$ sandbox-exec -f ls2 ls /mach_kernel
/mach_kernel
$ sandbox-exec -f ls2 cat /mach_kernel
cat: /mach_kernel: Operation not permitted
```

Log output:

```
Sep 2 00:18:59 macbox sandboxd[24653]: cat(24652) deny file-read-data /mach_kernel
```

File-read-metadata

Control read access to the files-system metadata. For example "ls" will not work against the target (if action is deny) while a "cat" will (because it is accessing the contents, not the metadata).

Syntax:	(action file-read-metadata [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

- **Example:**

(deny file-read-metadata (literal "/mach_kernel"))

```
$ cat /mach_kernel
????uZ$
$ sandbox-exec -f ls2 cat /mach_kernel
cat: /mach_kernel: Operation not permitted
```

Log output:

```
Sep 2 00:24:11 macbox sandboxd[24809]: ls(24808) deny file-read-metadata /mach_kernel
```

File-read-xattr

This operation will control read access to the file extended attributes.

Syntax:	(action file-read-xattr [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode xattr
Modifiers:	send-signal no-log

Example:

- (deny file-read-xattr (literal “/mach_kernel”))

Result without sandbox:

```
$ xattr /mach_kernel  
com.apple.FinderInfo
```

Result with sandbox:

```
$ sandbox-exec -f ls2 xattr /mach_kernel  
xattr: [Errno 1] Operation not permitted: '/mach_kernel'
```

File-revoke

Controls access to revoke().

Syntax:	(action file-revoke [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

N/A

File-write*

Controls all available write operations described below.

Syntax:	(action file-write* [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-write* (literal “/test”))

```
$ sandbox-exec -f ls2 touch /test  
touch: /test: Operation not permitted
```

Log output:

```
Sep  2 21:05:46 macbox sandboxd[45341]: touch(45340) deny file-write* /test
```

File-write-data

Asdasd

Syntax:	(action file-write-data [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

File-write-flags

Control access to file flags (check manpage for chflags).

Syntax:	(action file-write-flags [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-write-flags (literal “/private/tmp/test”))

```
$ sandbox-exec -f ls2 chflags nohidden /tmp/test
```

```
chflags: /tmp/test: Operation not permitted
```

Log output:

```
Sep  2 19:29:59 macbox sandboxd[42198]: chflags(42197) deny file-write-flags  
/private/tmp/test
```

File-write-mode

Control access to file modes.

Syntax:	(action file-write-mode [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-write-mode (literal “/private/tmp/test”))

```
$ sandbox-exec -f ls2 chmod 777 /tmp/test
```

```
chmod: Unable to change file mode on /tmp/test: Operation not permitted
```

Log output:

```
Sep  2 19:54:35 macbox sandboxd[43051]: chmod(43050) deny file-write-mode  
/private/tmp/test
```

File-write-mount

Access control check for mounting a file system.

Syntax:	(action file-write-mount [filter] [modifier])
Actions:	allow deny

Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

N/A (tried different combinations and mount still works!)

File-write-owner

Control access to file ownership changes.

Syntax:	(action file-write-owner [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-write-owner (literal "/private/tmp/test"))

```
# sandbox-exec -f ls2 chown nobody /tmp/test
```

```
chown: /tmp/test: Operation not permitted
```

Log output:

```
Sep  2 20:05:48 macbox sandboxd[43419]: chown(43418) deny file-write-owner  
/private/tmp/test
```

File-write-setugid

Not implemented???

Syntax:	(action file-write-setugid [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

File-write-times

Access timestamps?

“Determine whether the subject identified by the credential can set the passed access timestamps on the passed vnode.”

Syntax:	(action file-write-times [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

N/A

File-write-unmount

Access control check for unmounting a filesystem.

Syntax:	(action file-write-unmount [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode
Modifiers:	send-signal no-log

Example:

- (deny file-write-unmount (literal "/Volumes/Mac OS X Install ESD"))

```
# sandbox-exec -f ls2 umount /Volumes/Mac\ OS\ X\ Install\ ESD/  
umount: unmount(/Volumes/Mac OS X Install ESD): Operation not permitted
```

Log output:

```
Sep  2 20:21:19 macbox sandboxd[43908]: umount(43911) deny file-write-unmount  
/Volumes/Mac OS X Install ESD
```

File-write-xattr

This operation will control write access to the file extended attributes.

Syntax:	(action file-write-xattr [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode xattr
Modifiers:	send-signal no-log

Example:

- (deny file-write-xattr (literal "/test"))

```
$ xattr -w test 123 /test  
$ xattr -l /test  
test: 123  
$ sandbox-exec -f ls2 xattr -w test2 123 /test  
xattr: [Errno 1] Operation not permitted: '/test'
```

Log output:

```
Sep  2 00:38:13 macbox sandboxd[25217]: Python(25216) deny file-write-xattr /test
```

ipc*

This operation will IPC related operations described below.

Syntax:	(action ipc* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-posix*

This operation will IPC POSIX related operations described below.

Syntax:	(action ipc-posix* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-posix-sem

Controls access to POSIX semaphores (create, open, post, unlink, wait).

Syntax:	(action ipc-posix-sem [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-posix-shm

Controls access to POSIX shared memory region (create, mmap, open, stat, truncate, unlink).

Syntax:	(action ipc-posix-shm [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-sysv*

This operation will IPC POSIX related operations described below.

Syntax:	(action ipc-sysv* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-sysv-msg

Controls access to System V messages (enqueue, msgrcv, msgrmid, msqctl, msqget, msqrcv, msqsnd).

Syntax:	(action ipc-sysv-msg [modifier])
Actions:	allow deny

Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-sysv-sem

Controls access to System V semaphores (semctl, semget, semop).

Syntax:	(action ipc-sysv-sem [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

ipc-sysv-shm

Controls access to mapping System V shared memory (shmat, shmctl, shmdt, shmget).

Syntax:	(action ipc-sysv-shm [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach*

Aaa

Syntax:	(action mach* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-bootstrap

Aaa

Syntax:	(action mach-bootstrap [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-lookup

Aa

Syntax:	(action mach-lookup [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-priv*

Aaa

Syntax:	(action mach-priv* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-priv-host-port

Aaa

Syntax:	(action mach-priv-host-port [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-priv-task-port

Aaa

Syntax:	(action mach-priv-task-port [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Mach-task-name

Aaa

Syntax:	(action mach-task-name [modifier])
Actions:	allow deny
Filters:	n/a

Modifiers: send-signal no-log

Example:

Network*

Controls all available network operations described below.

Syntax:	(action network* [filter] [modifier])
Actions:	allow deny
Filters:	network path file-mode
Modifiers:	send-signal no-log

Example:

- (deny network* (remote ip "*" :80"))

```
$ sandbox-exec -f ls2 nc www.google.com 80
```

Log output:

```
Sep 2 21:12:00 macbox sandboxd[45542]: nc(45540) deny network-outbound 74.125.39.99:80
```

Network-inbound

Control network inbound operations.

"A socket has a queue for receiving incoming data. When a packet arrives on the wire, it eventually gets deposited into this queue, which the owner of the socket drains when they read from the socket's file descriptor."

Syntax:	(action network-inbound [filter] [modifier])
Actions:	allow deny
Filters:	network path file-mode
Modifiers:	send-signal no-log

Example:

- (allow network-inbound (local ip4 "*" :22))

Network-bind

Control access to socket bind().

Syntax:	(action network-bind [filter] [modifier])
Actions:	allow deny
Filters:	network path file-mode
Modifiers:	send-signal no-log

Example:

- (deny network-bind (local ip "*" :7890))

```
$ sandbox-exec -f ls2 nc -l 7890
```

```
nc: Operation not permitted
```


Log output:

Sep 2 21:08:41 macbox sandboxd[45438]: nc(45437) deny network-bind 0.0.0.0:7890

Network-outbound

Controls access to send data to the socket.

Syntax:	(action network-outbound [filter] [modifier])
Actions:	allow deny
Filters:	network path file-mode
Modifiers:	send-signal no-log

Examples:

- (deny network-outbound)

This will deny any packets going out from the target application.

- (deny network-outbound (remote ip “*:80”))

```
$ sandbox-exec -f ls2 nc www.google.com 80
```

Log output:

Sep 2 22:29:03 macbox sandboxd[47760]: nc(47758) deny network-outbound
74.125.39.106:80

- (allow network-outbound (remote unix-socket (path-literal "/private/var/run/syslog")))

Allow access to the syslog unix socket.

Process*

Controls all available process operations described below. One important detail is that no filters are available here but are on process-exec.

Syntax:	(action process* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny process*)

```
$ sandbox-exec -f ls2 ls
```

```
sandbox-exec: ls: Operation not permitted
```

Log output:

Sep 2 22:36:09 macbox sandboxd[47975]: sandbox-exec(47980) deny process-exec /bin/ls

Process-exec

Control process execution.

Syntax:	(action process-exec [filter] [modifier])
Actions:	allow deny
Filters:	path file-mode

Modifiers: send-signal no-log no-sandbox
--

Example:

- (deny process-exec (literal “/bin/ls”))

```
$ sandbox-exec -f ls2 /bin/ls
```

```
sandbox-exec: /bin/ls: Operation not permitted
```

```
$ sandbox-exec -f ls2 ls
```

```
sandbox-exec: ls: Operation not permitted
```

Log output:

```
Sep  2 01:16:57 macbox sandboxd[26360]: sandbox-exec(26359) deny process-exec /bin/ls
```

```
Sep  2 01:17:00 macbox sandboxd[26360]: sandbox-exec(26363) deny process-exec /bin/ls
```

Process-fork

Control access to fork and vfork.

Syntax: (action process-fork [modifier])
--

Actions: allow deny

Filters: n/a

Modifiers: send-signal no-log

Example:

- (deny process-fork)

```
$ ./forktest
```

```
child!
```

```
parent!
```

```
$ sandbox-exec -f ls2 ./forktest
```

```
parent!
```

Log output:

```
Sep  2 01:23:52 macbox sandboxd[26677]: forktest(26676) deny process-fork
```

Signal

Control if program can send signals to itself, processes in the same group or all other processes.

Syntax: (action signal [filter] [modifier])

Actions: allow deny

Filters: signal

Modifiers: send-signal no-log

Example:

- (deny signal (target others))

The sandboxed process will not be able to send signals to other processes.

```
$ sandbox-exec -f ls2 kill -ALRM 10229
```

```
kill: 10229: Operation not permitted
```

Log output:

Sep 2 10:45:01 macbox sandboxd[31416]: kill(31418) deny signal

Sysctl*

Control all access to sysctl() and its variants, sysctlbyname and sysctlnametomib.

Syntax:	(action sysctl* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny sysctl*)

```
$ sandbox-exec -f ls2 sysctl debug
```

Log output:

Sep 2 01:33:50 macbox sandboxd[26952]: sysctl(26960) deny sysctl-read

```
# sandbox-exec -f ls2 sysctl -w debug.bpf_bufsize=1024
```

second level name bpf_bufsize in debug.bpf_bufsize is invalid

This happens because sysctl-read is also denied so it can't read the name.

Sysctl-read

Control read access to sysctl() and its variants, sysctlbyname and sysctlnametomib.

Syntax:	(action sysctl-read [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny sysctl-read)

```
$ sandbox-exec -f ls2 sysctl debug
```

Log output:

Sep 2 01:40:01 macbox sandboxd[27171]: sysctl(27170) deny sysctl-read

```
# sandbox-exec -f ls2 sysctl -w debug.bpf_bufsize=1024
```

second level name bpf_bufsize in debug.bpf_bufsize is invalid

Sysctl-write

Control write access to sysctl() and its variants, sysctlbyname and sysctlnametomib.

Syntax:	(action sysctl-write [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Note: there seems to be a bug in this implementation (Snow Leopard at least), where a (deny sysctl-write) requires a (allow sysctl-read), even if we have a (allow default).

Test command:

```
# sandbox-exec -f ls2 sysctl -w debug.bpf_bufsize=1024
```

Test profile:

(version 1)

(debug all)

(allow default)

(deny sysctl-write)

But it works if written this way:

(version 1)

(debug all)

(allow default)

(deny sysctl-write)

(allow sysctl-read)

System*

Controls all available system operations described below.

Syntax:	(action system* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system*)

```
# sandbox-exec -f ls2 date 01212200
```

```
date: settimeofday (timeval): Operation not permitted
```

Log output:

```
Sep  2 22:49:30 macbox sandboxd[48428]: date(48435) deny system-set-time
```

System-acct

Determine whether the target should be allowed to enable accounting, based on its label and the label of the accounting log file. See acct(5) for more information.

Syntax:	(action system-acct [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow system-acct)

System-audit

Determine whether the target can submit an audit record for inclusion in the audit log via the audit() system call.

Syntax:	(action system-audit [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow system-audit)

System-fsctl

Control access to fsctl().

Warning: The fsctl() system call is directly analogous to ioctl(); since the associated data is opaque from the standpoint of the MAC framework and since these operations can affect many aspects of system operation, policies must exercise extreme care when implementing access control checks.

Syntax:	(action process* [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system-fsctl)

System-lcid

Determine whether the target can relabel itself to the supplied new label (newlabel). This access control check is called when the mac_set_lctx/lcid system call is invoked. A user space application will supply a new value, the value will be internalized and provided in newlabel.

Syntax:	(action system-lcid [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow system-lcid)

System-mac-label

Determine whether the target can perform the mac_set_fd operation. The mac_set_fd operation is used to associate a MAC label with a file.

Syntax:	(action system-mac-label [modifier])
Actions:	allow deny

Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system-mac-label)

System-nfssvc

Determine whether the target should be allowed to call nfssrv(2).

Syntax:	(action system-nfssvc [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow system-nfssvc)

System-reboot

Controls if target can reboot system.

Note: doesn't seem to work!

Syntax:	(action system-reboot [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system-reboot)

System-set-time

Controls access to the system clock.

Syntax:	(action system-set-time [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system-set-time)

```
# sandbox-exec -f ls2 date 01212200
```

```
date: settimeofday (timeval): Operation not permitted
```

Log output:

```
Sep 2 22:49:30 macbox sandboxd[48428]: date(48435) deny system-set-time
```

System-socket

Control access to create (raw?) sockets.

Syntax:	(action system-socket [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (deny system-socket)

System-swap

Access control check for swap devices (swapon/swapoff).

Syntax:	(action system-swap [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

- (allow system-swap)

System-write-bootstrap

Not implemented???

Syntax:	(action system-write-bootstrap [modifier])
Actions:	allow deny
Filters:	n/a
Modifiers:	send-signal no-log

Example:

Job-creation

Not implemented ???

Syntax:	(action job-creation [filter] [modifier])
Actions:	allow deny
Filters:	path
Modifiers:	send-signal no-log

Example:

Mach-per-user-lookup

???

Syntax:	(action mach-per-user-lookup [modifier])
Actions:	allow deny

Filters:	n/a
Modifiers:	send-signal no-log

Example:

5.3 – Filters

Filters can be applied to the operations that support them, allowing better control and granularity. The filters can be path names, file names, IP addresses, extended attributes, file modes. Regular expressions are supported in some.

The following table resumes the existing filters:

path	network	file-mode	xattr	mach	signal
------	---------	-----------	-------	------	--------

Anything included in square braces “[]” is optional.

Path

Match filenames or paths.

Three different modes are supported, regular expressions, literal, and subpath.

Symlinks are resolved so a path filter (literal or regex matching the beginning) to “/tmp/testfile” will fail because “/tmp” is a symbolic link to “/private/tmp”. In this case the correct filter should be “/private/tmp/testfile”.

1. Regular Expressions

Syntax:

(regex EXPRESSION)

Example:

- (allow file-read* (regex #"^/usr/lib/*"))

This will allow file reading access to all files available under /usr/lib/.

Multiple regular expressions are supported, so the operation can apply to multiple paths and/or files.

Example:

```
• (allow file-read*  
(regex  
  #"^/usr/lib/*"  
  #"^/dev/*"  
  #"^/System/Library/Frameworks/*"  
)  
)
```

2. Literal

Syntax:

(literal PATH)

Example:

- (deny file-read* (literal "/dev"))

This will deny all file read access to /dev only, but everything else inside /dev isn't protected by this operation.

```
$ sandbox-exec -f ls2 ls /dev
```

```
ls: /dev: Operation not permitted
```

```
$ sandbox-exec -f ls2 ls /dev/dtrace
```

```
/dev/dtrace
```

3. Subpath

Syntax:

(subpath PATH)

Example:

- (deny file-read* (subpath "/dev"))

In this case, everything under /dev will be denied read access (including /dev itself).

Network

Syntax:

(local ip|ip4|ip6|tcp|tcp4|tcp6|udp|udp4|udp6 ["IP:PORT"])

(remote ip|ip4|ip6|tcp|tcp4|tcp6|udp|udp4|udp6 ["IP:PORT"])

(remote unix|unix-socket [path-literal PATH])

The default "IP:PORT" is "*:*". Aliases "from", "to", and "unix-socket" can be used instead of "local", "remote", and "unix". The ICMP protocol is included in the IP and UDP options.

Note:

In this case, PATH must be "path-literal" instead of "regex", "literal", or "subpath".

Example:

- (deny network* (remote ip "*:*"))

Deny IP access to any remote host.

```
$ sandbox-exec -f ls2 ping www.google.com
```

```
PING www.l.google.com (74.125.39.147): 56 data bytes
```

```
ping: sendto: Operation not permitted
```

Log output:

```
Sep  2 11:00:17 macbox sandboxd[31870]: ping(31869) deny network-outbound
```

```
74.125.39.147:0
```

Example:

- (deny network* (remote tcp "*:*"))

Deny TCP access to any remote host.

```
$ sandbox-exec -f ls2 telnet www.google.com 80
```

```
Trying 74.125.39.147...
```

```
telnet: connect to address 74.125.39.147: Operation not permitted
```

Log output:

```
Sep 2 11:02:20 macbox sandboxd[31937]: telnet(31935) deny network-outbound
74.125.39.147:80
```

Example:

- (deny network* (local tcp "*:*"))

Deny TCP access to localhost ports.

```
$ telnet localhost 22
```

```
Trying 127.0.0.1...
```

```
telnet: connect to address 127.0.0.1: Connection refused
```

Log output:

```
Sep 2 11:04:49 macbox sandboxd[32011]: telnet(32010) deny network-outbound 127.0.0.1:22
```

Example:

- (allow network* (remote unix-socket (path-literal "/private/var/run/syslog")))

File-mode

Syntax:

(file-mode FILEMODE)

Example:

???

Xattr

Match the extended attribute name, not content.

Syntax:

(xattr REGEX)

Example:

- (deny file-write-xattr (xattr "test_xattr"))

```
$ xattr -w test_xattr aaaa /tmp/xattr
```

```
$ xattr -l /tmp/xattr
```

```
test_xattr: aaaa
```

```
$ sandbox-exec -f ls2 xattr -w test_xattr aaaa /tmp/xattr
```

```
xattr: [Errno 1] Operation not permitted: '/tmp/xattr'
```

Log output:

```
Sep 2 11:48:02 macbox sandboxd[33295]: Python(33294) deny file-write-xattr
/private/tmp/xattr
```

Mach

These are needed for things like getpwnam, hostname changes, & keychain.

Syntax:

(global-name REGEX|LITERAL)

(local-name REGEX|LITERAL)

Example:

```
(allow mach-lookup (global-name
    "com.apple.bsd.dirhelper"
    "com.apple.distributed_notifications.2"))
```

Signal

Syntax:

(target self | pgrp | others)

Where:

self: sandboxed process itself

pgrp: group processes ?

others: all processes

Example:

(deny signal (target others))

The sandboxed process will not be able to send signals to other processes.

\$ sandbox-exec -f ls2 kill -ALRM 10229

kill: 10229: Operation not permitted

Log output:

Sep 2 10:45:01 macbox sandboxd[31416]: kill(31418) deny signal

5.4 – Modifiers

There are three available modifiers, although one just applies to a single operation. The modifiers are send-signal, no-log, and no-sandbox. To use them you will need the keyword “with”.

1. Send-signal

The best description is found in Apple’s scripts:

“To help debugging, “with send-signal SIGFPE” will trigger a fake floating-point exception, which will crash the process and show the call stack leading to the offending operation.

For the shipping version “deny” is probably better because it vetoes the operation without killing the process.”

There is a special exception, where send-signal doesn’t apply to mach-* operations.

It can be applied to allow and deny actions.

Syntax:

(with send-signal SIGNAL)

Example:

(deny file-read* (with send-signal SIGFPE))

The target binary will crash with a floating point exception when it tries to read any file.

\$ sandbox-exec -f ls2 cat /tmp/test

Floating point exception

2. No-log

Do not log denied operations. Applies only to deny action.

Syntax:

© 2011, rgl

(with no-log)

Example:

(deny file-read* (subpath “/tmp”) (with no-log))

3. No-sandbox

Applies only to allow action and process-exec operation.

Syntax:

(with no-sandbox)

Example:

???

5.5 - Other keywords

6 – Special hardcoded cases

The following special cases can be found inside the code:

- Allow mach-bootstrap if mach-lookup is ever allowed.
- Allow access to webdavfs_agent if file-read* is always allowed.
- Never allow a sandboxed process to open a launchd socket.c

7 – A sample profile for Outlook 2011