Mac OS X : System Integrity Protection

Nicolas RUFF - nruff(at)google(dot)com

Introduction

What is SIP?

SIP: "System Integrity Protection", a.k.a. "rootless".

SIP restricts capabilities, even for the `root` user.

- No write access to:
  - `/System`, `/bin`, `/sbin`, `/usr` (except `/usr/local`)
- No access to Apple-signed processes.
  - Includes memory dumping, ptrace() and DTrace access.
- No unsigned kernel extension (`kext`) loading.
- No write access to boot- and SIP-related NVRAM settings.
- ... plus a few other goodies
  - Protects symbolic links inside `/etc`, `/tmp`, `/var`
  - Protects system apps under `/Applications`
  - Protects against removal of selected `launchd` services.
  - Etc.

How is SIP implemented?

- Configuration file under:
  - `/System/Library/Sandbox/rootless.conf`
- Backward compatibility list under:
  - `/System/Library/Sandbox/Compatibility.bundle/Contents/Resources/paths`
- Individual setting bits stored in NVRAM.
- Can be selectively disabled in Recovery Mode using `csrutil` command.
- Live controlled by syscall `0x1e3`.
- `ls -O` displays "protected" files.

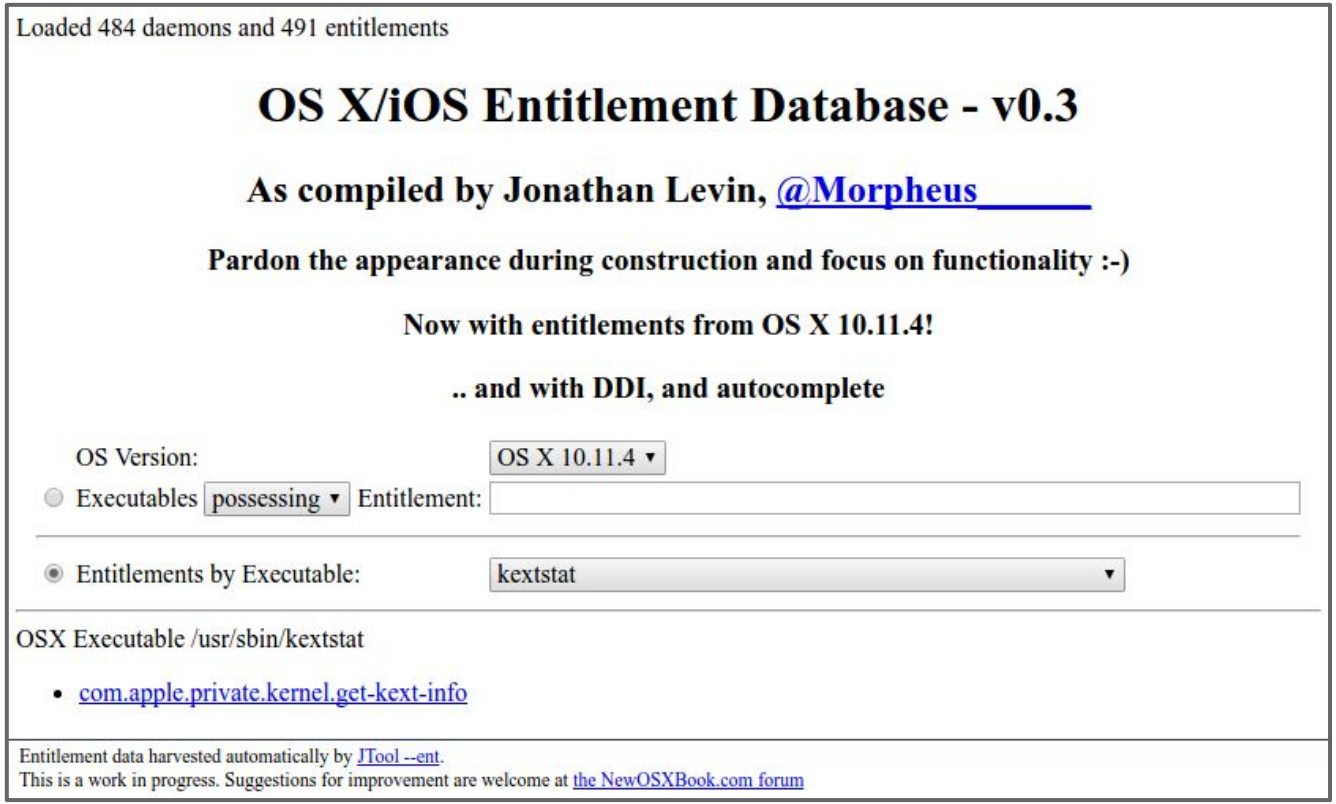
What is SIP goal?

- SIP aims at protecting the core OS against permanent loss of integrity.
- Threat model is `root` to kernel and/or protected location escalation.
  - Local users already have sudo access on OS X.
- SIP is application-based access control rather than user-based.
- Applications are identified by:
  - Signing authority (Apple) + signature + entitlement(s)

"Entitlements"

Strings, essentially.

(in XML form)



Source: http://newosxbook.com/ent.jl

SIP Shortcomings

SIP shortcomings

Existing extensions:

- `/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist` contains a whitelist of 11,000+ unsigned-yet-allowed extensions.
  - Identified by Bundle SHA-1.
  - The revocation list is silently updated by default.
- Signed `kext` with known bugs ... or features.
  - E.g. https://www.spyresoft.com/dockmod or `AppleHWAccess.kext`
  - Both blacklisted.
- `kext` signing certificate costs $99.

Fixed in OS X 10.11: whitelist not honored anymore.

SIP shortcomings

`kext` signature check is implemented in userland (`kextd` and `kextload`).

Fixed in OS X 10.11:

- Require `com.apple.rootless.kext-management` entitlement.
- Prevent the debugging of system processes.

SIP shortcomings

Misbehaving "entitled" application.

- E.g. `fsck_cs -l <logfile>`
- https://twitter.com/i0n1c/status/714261458851221504

This particular one has been fixed in OS X 10.11.5.

"Entitled" applications should be considered as dangerous as `suid` binaries.

SIP shortcomings

Kernel debugger.

- Requires physical access.

`gdb-i386-apple-darwin`

- Can run (but not attach to) protected processes [Now fixed].

SIP shortcomings

Kernel bugs.

- Writing a single `NULL` byte over the policy global var.
- Calling `_csr_set_allow_all(1)`.

Note: `kas_info()` leaks ASLR offset to the `root` user (before OS X 10.11.3).

Conclusion

Conclusion

SIP tries to replace user-based permissions by application-centric permissions.

Adding security to a decade-old design is challenging to get right ; expect more bugs.

Kernel attack surface is still huge ; a single bug defeats the whole model.

References

Apple Documentation
https://developer.apple.com/library/mac/documentation/Security/Conceptual/System_Integrity_Protection_Guide/Introduction/Introduction.html

External analysis
http://www.slideshare.net/i0n1c/syscan360-stefan-esser-os-x-el-capitan-sinking-the-ship
http://go.sentinelone.com/rs/327-MNM-087/images/SyScan360%20SG%202016%20-%20Memory%20Corruption%20is%20for%20wussies.pdf

Also relevant to OS X Security
http://reverse.put.as/
https://objective-see.com/blog.html
https://bugs.chromium.org/p/project-zero/issues/list?can=1&q=OS+X