# macOS 10.14 Security Features

macOS 10.14 Mojave is Apple's new OS version expected to ship later this year. It's available in the public beta program now. At Apple's Worldwide Developers Conference (WWDC18), Apple had a session "Your Apps and the Future of macOS Security" [1] where they gave a technical overview of new security features in the OS. These are my notes from that video and what I've found while experimenting with the 10.14 Beta.

- **macOS Platform Security Improvements**
    - Starting with 10.11 El Capitan, macOS has System Integrity Protection (SIP) where platform binaries are protected from modification on disk and at runtime
    - With 10.14 Mojave, the OS will strengthen SIP by enforcing validity of code signatures throughout the lifetime of system processes and abort any process that doesn't match its code signature or attempts to execute any code not signed by Apple
    - Dynamically loaded libraries/frameworks/plugins loaded by system processes must be signed by Apple too
    - Exceptions for some legacy features that use non-Apple extensions
    - Already default on iOS

- **User Consent with Approve/Deny Dialogs**
    - High Sierra already prompted user to allow an app to access location services, contacts, calendars, reminders, photos
        - Let "well intentioned apps honor user preferences"
    - On Mojave, this is enforced, even for apps accessing the data via the filesystem
    - Dialogs are blocking
    - User can whitelist apps to avoid numerous repeated prompts
        - Apple gave example of an app iterating over files in Photo collection triggering many dialogs
    - Mojave additionally restricts access to
        - Mail, Messages, Safari History, HTTP Cookies, iTunes Backups, Time Machine Backups
            - Doesn't prompt for access to these, only system apps or whitelisted apps in security/privacy prefs can access

- **Camera and Microphone Access**
    - Apps built with 10.14 SDK require plist entries in the .app in order to access camera or mic
        - Access triggers an approve/deny dialog
    - For Apps built with earlier SDK's, the new entries are optional

- Aps can add description strings that are included in the dialog to explain why the access is needed
- Can enumerate devices without prompt
- Applies to all devices supported by system drivers
- Can query authorization status
    - states: { notDetermined, restricted (user can not consent), denied (user previously denied), authorized }
- Apps can request access programatically ahead of time, API is asynchronous
- OS remembers user decision
- “tccutil reset” command (appears to reset approval for every app)

- **Enhanced Runtime Protection**
    - Is opt-in and requires building with the 10.14 SDK
    - Enables protections similar to those on system binaries
    - Configurable with entitlements (“unrestricted entitlements” — available to anybody without any prior approval)
    - Manageable from Xcode or with the codesign "—options runtime” flag
    - Safe to enable on binaries deployed on older versions of macOS
    - Versioning scheme in place to support future revisions of the policies
    - Code Signing
        - Every executable page must match what shipped with your app
        - Enforces code signature of dynamically loaded libraries at runtime
            - Must be signed by Apple or same Team ID
                - Same team requirement can be disabled
    - Debugging other apps, being debugged, dyld env vars require exceptions
    - Resource access
        - Accessing resources not listed in your entitlement list will make the app crash
            - Entitlements for mic, camera, location, contacts, etc.
    - Exception entitlements used to opt-in to allowing more risky behavior
        - com.apple.security.cs.allow-jit works
            - Enable access to regions mapped as MAP_JIT
        - com.apple.security.cs.allow-unsigned-executable-memory
            - Enable executable mapping without a signature, for loading unsigned libraries
        - com.apple.security.cs.disable-executable-page-protection
            - Disables all code signing protection
        - com.apple.security.cs.disable-library-validation
            - Allow loading of libraries signed by different Team ID's
        - com.apple.security.cs.debugger
            - For apps that are debuggers
        - com.apple.security.cs.allow-dyld-environment-variables
        - com.apple.security.get-task-allow

            - To allow the app to be debugged
    - Is it inherited by child processes? TBD
    - Is it compatible with custom sandbox profiles like we use? TBD

- **Notarized Apps**
    - Apple already scans App Store apps
    - This is an optional feature apps distributed outside of the App Store using developer ID program
    - Apps are pre-submitted to Apple, Apple runs automated malware checks
    - When a Notarized App is downloaded and run for the first time by the user, the GateKeeper dialog is less of a warning
        - See screenshots on bug 1470607[5] and in video[1]
    - Available already
    - Notary service issues a ticket
    - Ticket “stapled” to app
    - May be required in the future
    - “Not a review”
    - Targeting a turnaround time of less than an hour
    - Requirements
        - Enhanced runtime
        - Therefore 10.14 SDK? TBD
        - All binaries be codesigned, opted into enhanced runtime
    - Command line tools available

- **Security Model Changes for Allowing Apps To Do Things on Users’s Behalf**
    - New approval mechanism for users to enable software controlling the UI on their behalf
        - Configurable in the Security and Privacy preferences pane in the Accessibility list
        - Some API’s for controlling the UI now only allowed by whitelisted Apps in the settings, setMouseLocation(), tapCreate().
            - Impact on our tests? TBD.
            - We call CGEventTapCreate() which won’t work on 10.14 unless running as root or "Access for assistive devices is enabled”
                - Seems OK because we use kCGEventTapOptionListenOnly which is mentioned in the video
                - https://developer.apple.com/documentation/coregraphics/1454426-cgeventtapcreate

- **Safari / Webkit**
    - Sandbox adoption for the app and satellite processes
    - Web processes no longer need access to Dock, WindowServer, network[7]

    - Audit and removal of risky dependencies from existing sandboxes
    - Heap and JIT hardening (isolated heaps)

**References:**


1. Your Apps and the Future of macOS Security
2. Bug 1470597 - Investigate enabling Enhanced Runtime for the parent process
3. Bug 1474447 - Use MAP_JIT for all executable JIT pages for 10.14 Enhanced Runtime
4. Bug 1474451 - Deploy Enhanced Runtime on Mac without com.apple.security.cs.allow-jit
5. Bug 1470607 - Use the Apple Notary Service for signed builds
6. Bug 1470833 - Add NSCameraUsageDescription and NSMicrophoneUsageDescription Info.plist Keys for 10.14
7. WebKit web process sandbox profile (search for 101400 for 10.14-specific code)
8. Bug 1474462 - Intermittent "plugin-container" would like to access the microphone