jamf

WHITE PAPER

Deprecating Kernel Extensions: How to Transition to Kextless Workflows

At the 2019 Apple Worldwide Developers Conference (WWDC), Apple announced a plan to begin deprecating the usage of kernel extensions (KEXTs) as a part of an ongoing effort to modernize macOS, improve security, and create reliability with third-party software and security providers to ensure compatibility with operating systems upgrades.

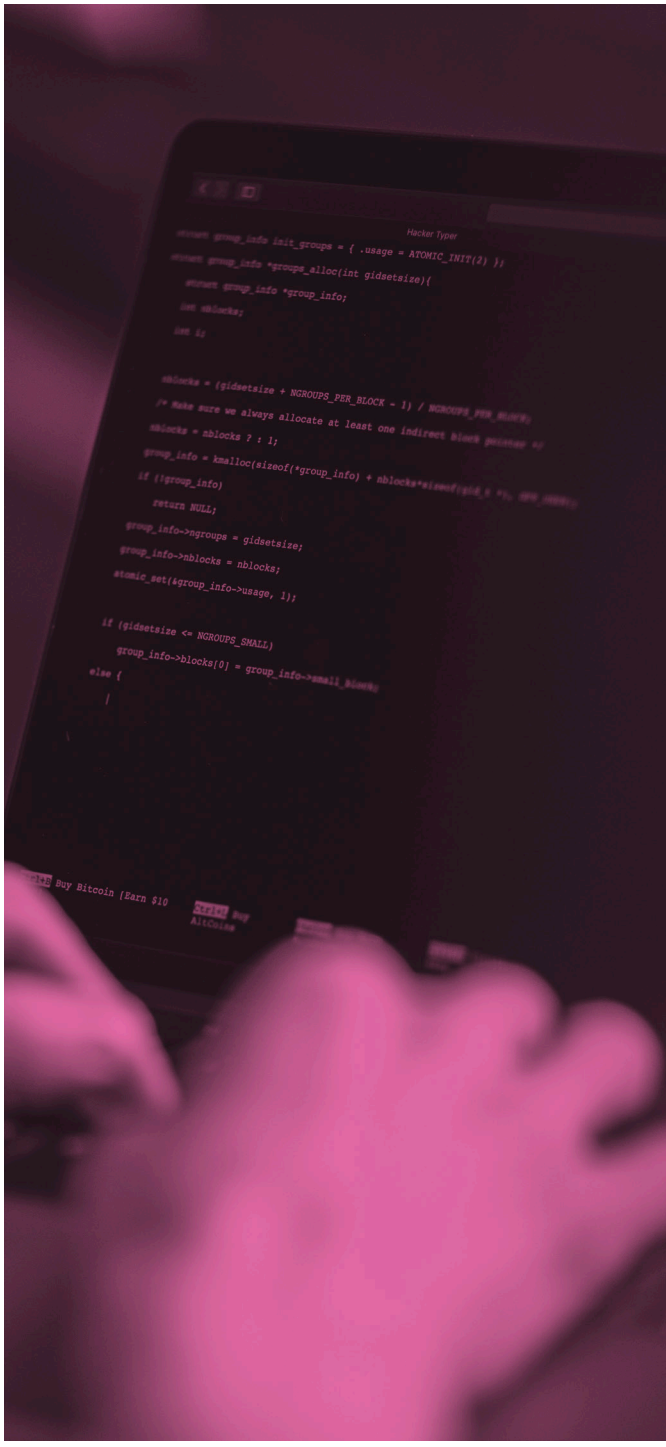
KEXT deprecation began with the release of macOS Big Sur, which means now is the time to understand how this affects your environment and find compliant partners to help with the transition.

This white paper covers:

- What kernel extensions are and why they are being deprecated
- Why Jamf Protect was built to be “kextless”
- The benefits of kextless for IT, Information Security and users

What are kernel extensions?

Within a system, there are two “levels” or “modes” — user mode and kernel mode. User mode is where many applications coexist simultaneously with inherent boundaries between them. These boundaries limit the access apps have within their specific, preset boundaries. The whole of user mode is also restricted from interacting with anything running in kernel mode, outside of very specific capabilities (APIs), or making any changes. This ensures that no single application brings down the whole system and has a difficult time interfering with other applications that are operating within its own boundaries.



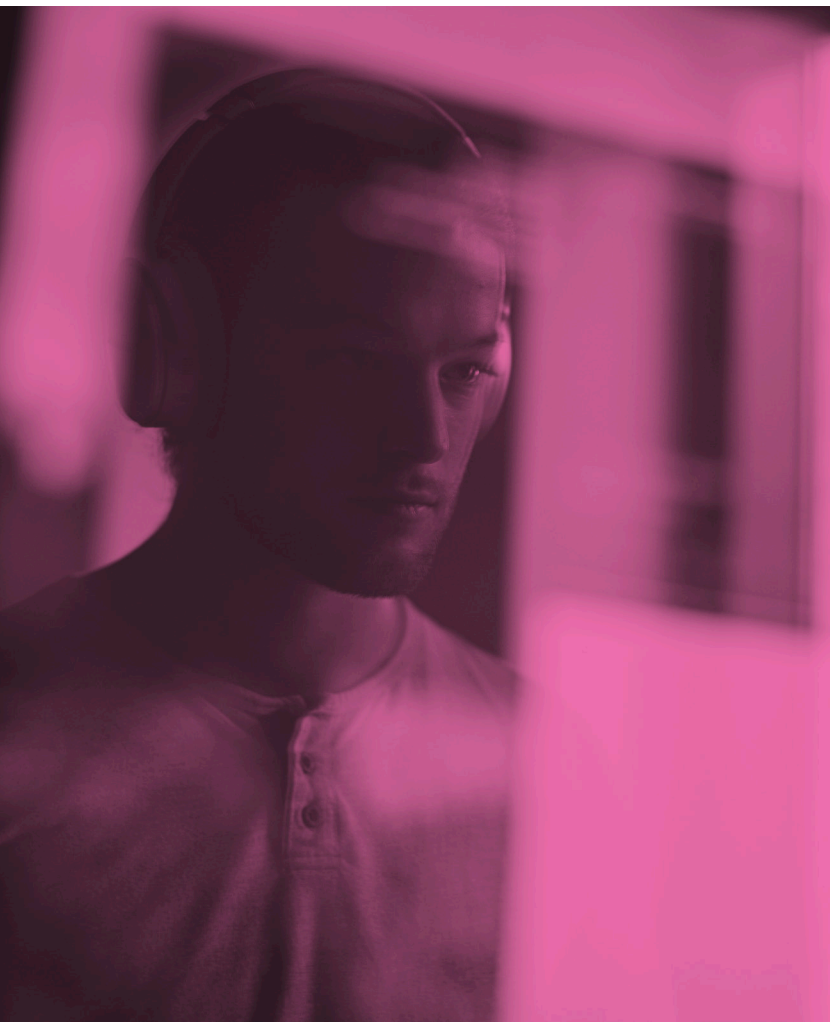
Kernel mode on the other hand has no boundaries set in place. It can interfere with any running application, modify any piece of data, has administrative privileges by design and can even interfere with the operating system itself. After all, kernel space is designed to only contain the operating system. As a result, anything running in kernel mode is inherently a stability risk, a security risk and a privacy risk. Kernel mode software has stricter expectations on quality control than normal applications.

KEXTs offer developers the ability to tap into the kernel space and develop within the kernel. From this space, developers can monitor processes, prevent applications from launching and generally monitor all actions on a device. In legacy OSes, many implementations of security monitoring and application’s controls leveraged this technology. However, vendors are still fallible.

Why is Apple deprecating kernel extensions?

Coding within the kernel is a ‘double-edged sword’. It allows third-party apps and software to have direct access to make substantial changes, but if something goes wrong, it often goes very wrong. Creating extensions into the kernel is not an aspect that is supported by Apple because you are often interfering with the operating system itself, potentially in a way that impacts its ability to function properly. Because there are no boundaries within the kernel, when things break, there is little there to prevent it from breaking aspects that affect the rest of the device’s functioning. Additionally, this makes all modifications dependent on Apple not altering the kernel mode portions of future OS releases in a way that third-party applications are not expecting.

Apple ardently advises caution when it comes to KEXT usage because of their intentions to alter the operating system as a part of future releases. If Apple does change an aspect of the OS a KEXT is dependent on and the device is upgraded to the new OS version – regardless of it being a major upgrade, minor update, or supplemental patch – it could create a lot of issues, a hole in security and even render the device unusable.



Some of the prime users of KEXTs are third-party security providers. Because of the access and capabilities KEXTs can offer, many security providers rely on a KEXT to function. This, in turn, can make security a major source of instability and frustration for companies leveraging a security provider that is dependent on KEXTs when it comes time to patch or upgrade the OS as vendors need time to verify their product on the new OS.

This model of kernel mode, user mode and kernel extensions is not unique to Apple. Windows calls kernel extensions “drivers” which are the only supported way many applications function. This plays a key role in understanding why security providers naturally turn to KEXT. Many security providers have a main product that was built for Windows first, and they looked to make that same framework work for Apple devices. As people have discovered with unified endpoint management (UEM), you can’t leverage one tool and have them magically work for all operating systems. Having a product built directly for Apple, in this case Apple security, is the only way to unlock the countless benefits of Apple.

There is an alternative option to still achieve these security measures without this large amount of risk. One that was built with Apple device security as the top priority. Since applications in user mode never interact with those internal structures and systems, they are safe from these deprecative changes.

Forward-thinking solution built without KEXT

At the onset of Jamf Protect, an endpoint security solution built purposefully for Mac, it was designed to use Apple-supported aspects and never required the use of kernel extensions. This also means when Apple announced at WWDC19 the deprecation and eventual elimination of KEXT, while also unveiling the addition of a new System Extensions with the Endpoint Security Framework, Network Extensions and DriverKit, Jamf was ready to embrace this concept, without having to undo any reliance on kernel extensions. This resulted in lower device impact, better stability and strong privacy for any applications adopting these new frameworks.

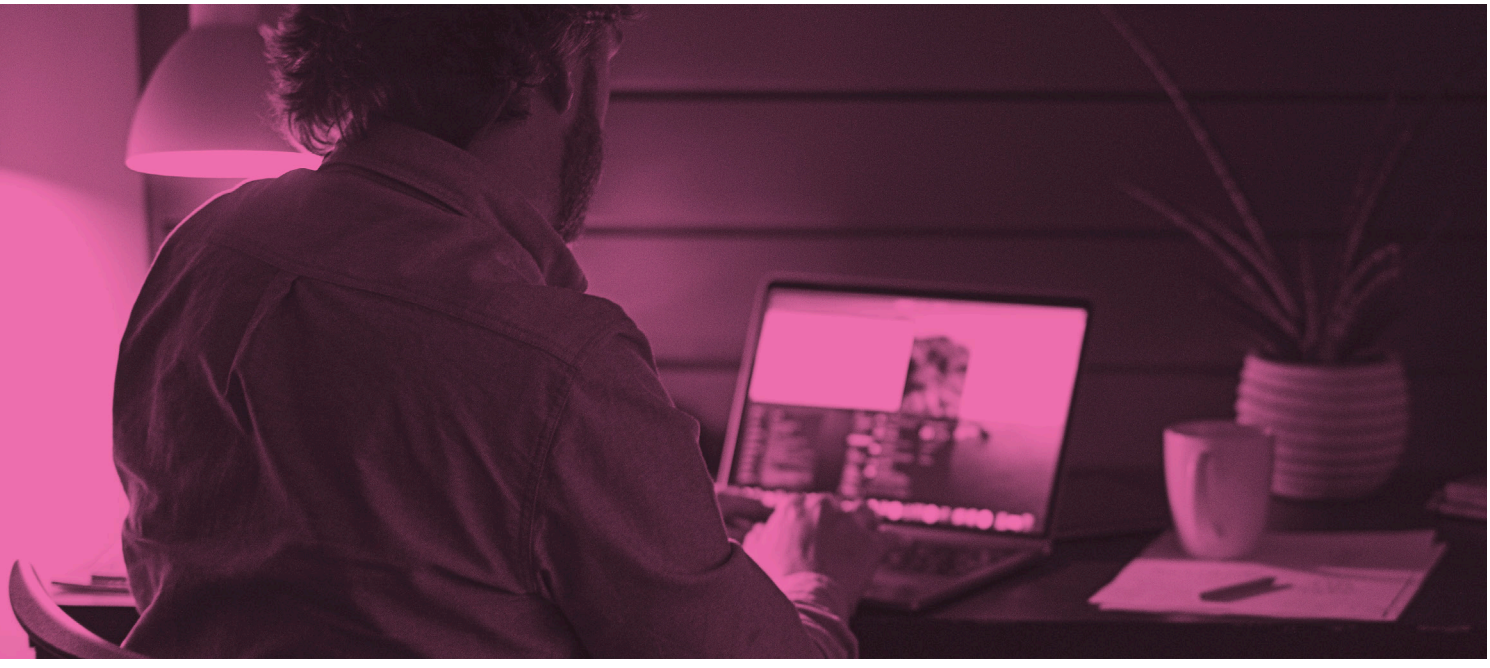
The announcement of System Extensions marked the first time Apple supported a method to build software in a way that would achieve similar results as kernel extensions, without access to kernel mode. It offered a definitive way for security providers to gain the rich information and data they needed in a stable way, while backing up Apple's message that organizations should seek third-party providers that do not leverage KEXT.

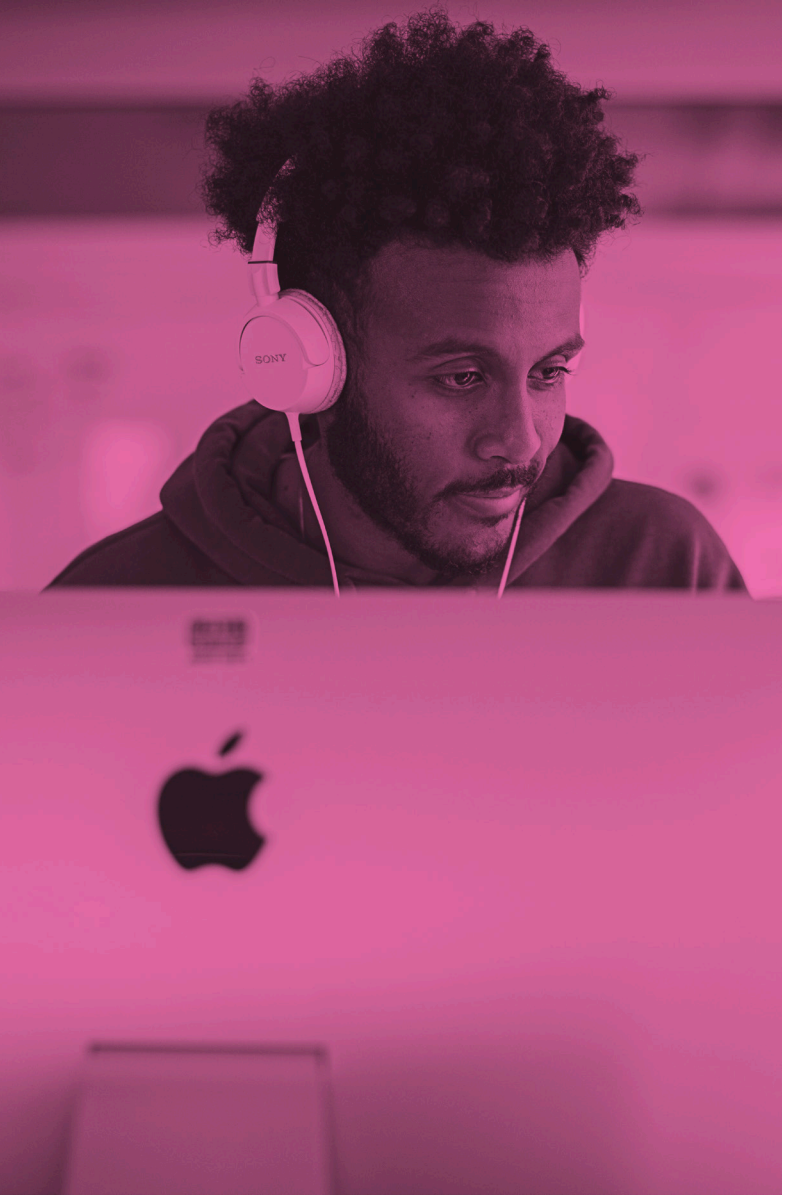
This is fantastic news for Jamf customers, but a bit of a quandary for many providers that were told they would need to overhaul their product for Apple devices. Jamf Protect didn't need to change a thing for its customers and immediately embraces the new power Apple.

Why a kextless provider is the way to go

Whether it happens over the course of a few years or within the next few rollouts of macOS, Apple has said they are eliminating KEXT. This means there is a point in the near future when switching to the new user mode frameworks instead of a KEXT will become mandatory. Apple made it clear during WWDC 2020 that vendors have to switch now and even encouraged customers to switch vendors if they don't support the new model.

As mentioned before, when issues with kernel extensions go bad, they can go very bad and kernel panics are potentially extremely difficult and time-consuming to diagnose and fix with a remote workforce. Some situations may not even be fixable when remote, and require devices be shipped to IT or Apple for recovery. This leads to more IT Tickets, lengthier IT ticket resolutions, and can drastically hinder the user experience with Mac.





With the release of Apple's newest Mac operating system, macOS Big Sur, it's important to mention that a kextless security provider and kextless third-party apps remove a major aspect of stress that revolves around upgrading devices. Since many security providers are still testing their platforms to make sure it will function as desired with the new OS, it causes unnecessary stress on organizations looking to keep their devices compliant with the most up-to-date system and can result in costly downtime. Jamf doesn't have that issue as it offers same-day Apple OS support and compliance with every product, including Jamf Protect.

Finally, now that Apple has introduced their own silicon in the form of the M1 chip, tools relying on KEXTs face new end-user experience challenges. Any macOS Big Sur device running a KEXT already required that either an en-duser confirms that the KEXT was allowed to run at each device boot or that a mobile device management (MDM) solution was used to authorize the KEXT via a profile. However, with M1 devices, users must first reboot their machine into recovery mode to allow it to attempt loading KEXTs and vendors have to provide M1 specific versions of the KEXT. It's an arduous process and just one more nail in the coffin for KEXTs on macOS.

Ready for a better, more secure way with kextless workflows?

Relying and leveraging kernel extensions have always been a short-term solution that brings instability and unreliability to your environment. It was sufficient at the time for many security providers, but there is now a better way. With Apple deprecating KEXT, that superior method has never been more needed. Turning to a security provider that provides endpoint protection built specifically for Mac with Apple users in mind puts you one step ahead of security threats and keeps your team's success at the forefront.

Try Jamf Protect