PART II

SYSTEM MONITORING

So far, I've covered programmatic methods of collecting data to generate snapshots of the system's state, then analyzed these snapshots to uncover symptoms of malicious activity. This approach limits the analysis to single points in time, however. Simple antivirus programs often provide such a feature in a "scan now" option, which can be useful for determining whether the system has already been infected and for creating a baseline of a known good state. The obvious downside to this approach is that it's reactive and, worse, could miss an infection altogether. For example, ransomware could infect a system and render it inoperable in the window of time between snapshots.

The solution is to expand upon the methods presented in Part I to provide real-time monitoring capabilities. In Part II, I'll explain how to monitor the system log, as well as network, filesystem, and process events, in real time. In some cases, we'll have to write code specific to the target of our monitoring; in other cases, Apple's Endpoint Security framework can serve

as the basis for a wide range of monitors capable of overseeing filesystem, process, and many other noteworthy events. To fully understand Endpoint Security's capabilities, I'll spend an entire chapter highlighting its advanced features, including authorization and muting. The most comprehensive malware detection solutions will include the approaches presented in Part I as well as the techniques I'll cover in Part II.

Also, the monitoring code can apply strategies covered in Part I for identifying anomalies. For example, the logic we wrote in Chapter 2 to detect that a running process's binary is packed can identify suspicious binaries in real time, such as when a process monitor intercepts a newly spawned process.