Custom Kernel Mode Network Solutions from JW Secure

Generally, software applications avoid the use of custom kernel mode code due to the danger that poorly written code will cause the computer to crash. Strictly, the only code that must reside in the kernel is that which talks directly to the hardware. However, the lure of better performance has forced some code like network stacks and authentication into the kernel. Offsetting this desire for better performance is a desire for more stability of the computer, which has pushed many traditional kernel functions into user mode, including device drivers that do not require direct hardware access.

> **Sherlock Holmes** said that "when all other contingencies fail, whatever remains, however improbable, must be the truth." Likewise for kernel-mode code; it should be considered an improbable solution until all user-mode options fail.

Crashing the computer is bad, so why consider a solution using kernel mode code? Server performance is a major factor. For example, if Internet Information Server 7 is deployed, it will use kernel-mode transport layer security (TLS aka SSL) and the associated kernel-based authentication module. The typical performance problems that demand kernel-mode solutions are those that involve a large number of user-kernel transitions, high-volume data transfer, or real-time responses to external events. In the case of high-performance secure web transactions, a worst-case scenario is that data must be moved back and forth between user and kernel multiple times for each packet in order to perform cryptographic functions. A kernel-only solution doesn't have that problem.

JW Secure has solved performance and security problems for several customers by creating custom solutions which include kernel components. A typical scenario requires network packet data to be intercepted in the kernel. The intercepted data is inspected, modified, and retransmitted without a throughput-killing transition to user-mode. Such was the case for the DARPA MilNet program, as well as for the Microsoft RODC Kerberos Filter project. In both cases, when network data needed additional security and flow control, a kernel-mode driver developed by JW Secure was the answer.

Consider the problem of routing video streams through the variety of network gateways that are typical of today's home or business networks. For many such networks UDP has been shown to be unsuccessful. JW Secure has the know-how to provide an even rate of packet flows for TCP that is able to traverse all known network configurations, something that is not possible with typical TCP/IP driver stacks.

In designing, implementing, and testing custom code for the kernel, rigorous review is required. Only senior systems-level software engineers with extensive kernel-mode development experience have the skill to deliver crash-proofed code that will be able to protect critical computers from downtime. Whatever the data flow or security problem your organization might face, if performance absolutely must be improved, an analysis of the flows between user and kernel mode is most likely to reveal the problem and suggest a solution. Let us run that analysis for you.

These resources offer additional background on kernel-mode code:
Understanding User and Kernel Mode
Windows Programming / User Mode vs. Kernel Mode