Kernel mode hooks or user mode hooks – what's best for the firewall?

Introduction

There has been much discussion in online forums and the media concerning the ability of firewalls to prevent in-the-wild malware from impacting system security and stability. A newly-published set of leaktests has intensified arguments as to whether the kernel mode hook or the user mode hook provides better user protection. This document is Agnitum's contribution to that debate, from the perspective of a leading firewall developer.

The document is in two parts: the first one spells out the potential benefits of both approaches, and the second explains briefly why we chose one particular approach when designing and developing Outpost.

Techniques for controlling application interactivity

Applications frequently interact with each other on a PC. It happens, for example, when you open up a PDF document from a My Computer location by double-clicking on the filename; Windows Explorer calls up the default PDF viewer in order to render the file in its context. In order for the firewall to be able to detect such an interaction and prevent the unauthorized practice of one program using another's access credentials for nefarious purposes by hijacking this process, the firewall (and other proactive defense applications) needs to place an intermediary, known as a function hook, to intercept intra-program commands in order to monitor the mutual interaction.

In order to provide effective protection, security applications therefore need to gain control over running processes, for example to check their permissions for interaction before allowing them to attempt any activity. To achieve this, the firewall replaces a number of internal system functions calls with its own functions. This process of function interception is called *hooking*.

There are two widely-accepted methods for intercepting program functions— *user mode hooks* and *kernel mode hooks* (such as, for instance, the System Service Descriptor Table (SSDT) hook). User mode is a special method for processing tasks that are visible to the user, and carries fewer access privileges than kernel mode. Kernel mode, on the other hand, is a more global implementation which operates directly from the Windows kernel and processes system-specific commands. In simple terms, they could be referred to as "front end" and "back end" modes respectively.

Let's take a look at the upsides and downsides of each approach from the system security perspective. Below, you'll see a simple tabular representation of the differences, which should help you to better understand their implications as described in the text that follows.

Pros and cons of user-mode and kernel-mode hooks

Comparison criterion	User-mode hooks	Kernel-mode hooks
Security		
1) Hook disabling (unhooking).	Can be compromised if user mode hooks are not properly protected. Each method of user-mode hook protection requires specialized countermeasure to dismantle defenses.	Can be compromised when the Administrator account is used. Most users have administrator rights as the default setting on their PCs.
2) Local inter-process communication.	Can prompt for inter-process communication in real time, which prevents: • The disruption of legitimate applications • Data integrity compromise or loss due to target application modification.	Cannot prompt in real time because some inter-process communication cannot be put "on hold" in kernel mode without affecting system stability.
Stability		
3) Reliability of operation.	Good, rarely causes errors in operation. Most-case scenario would be a single-application crash.	Not so good. Stability depends on a variety of factors such as software/hardware configuration. Instability frequently results in system-wide crashes like reboots, blue screens and system freezes.
Compatibility		
4) 64-bit Windows compatible.	Yes	No, because of Microsoft PatchGuard requirements
5) Windows Vista compatible.	Yes	No, most events cannot be hooked in kernel mode by design.

Because of the way they are designed, some kernel-mode hooks do not allow a process to be halted (frozen) pending a decision from the user. That's why kernel-mode firewalls are forced to assume a predetermined outcome, which will not always be correct. In cases of inter-process communication, that decision would likely be to "Allow" first and then prompt if/when the modified application attempts network access. This is akin to inviting the thief in through the front door and then making sure he doesn't leave through an open back door. User mode hooks, on the other hand, provide for better flexibility to make a correct and informed decision and to proactively block malicious activity, thus preventing computer infestation at earlier stage.

Agnitum's approach

At Agnitum, we believe that a healthy symbiosis of the two approaches creates a win-win situation for the user.

In Outpost Firewall Pro, we use a complex approach that combines both user mode and kernel mode hooks working in conjunction to create stronger, broader protection.

Still, it's important to understand that it's absolutely wrong to assume that kernel-mode hooking in alone can provide the full scope of protection. Similar leaktests can be created to expose breaches in kernel-mode hooking as well. If your account has administrative rights, kernel mode hooks can be subverted by both malware and legitimate code (for example, a legitimate SDTRestore program which is able to remove such hooks unless denied a low-level memory access; AVZ is another example).

Therefore, it is to our view more appropriate for the firewall to combine both types of hooking than using one preferred method. This will make it more proactive, reliable and capable of resisting real-world threats. No single approach is as powerful as the two wisely used together. That's the vision we foster at Agnitum.

Summary

Both the kernel mode hooks and user mode hooks require sophisticated programming techniques to bypass but both can be vulnerable to some degree.

The best security advice we can give is that there is no 100% solution. The most effective techniques are those that can deal with real-world threats today by mixing together both techniques of controlling inter-process communication. That's always been our guiding principle and why we continue to believe that Outpost Firewall Pro is the best protection you can get.