Stages of Power Management Transition in Windows 7

Internal APIs research for OS Virtualization project

¡Manage system at power state transition: force it to the Sleep state and hook at hardware abstraction level “at the last moment”!

The Client’s Project

The Client is working with Apriorit team on the big OS virtualization project to allow several operating systems simultaneously run on a single hardware platform with a run-time switching between them.

Run-time switching supposes that user is returned to the same system state (activity, running applications, etc.) as before the initial switching. This functionality always provides Apriorit Research and Reverse Engineering Department with some interesting non-trivial tasks.

Project Task

This time, it was the implementation of Android x86 – Windows 7 x86 run-time switching. The strategy, which the team chose for this task, was to force currently inactive Windows OS to the Sleep state and hook it then – it would let to store the current system state and provide quick restoration. This approach supposed thorough research of the power management methods in Windows 7.

Documented power management methods are all high-level ones and can simply force the system to Sleep, while the task required the system to be hooked when it was already in the Sleep (Suspend-to-RAM) state. The only way was to hook some particular low level - hardware abstraction level – method at a certain power state transition stage. Obviously, these process and methods are closed in Windows system.

How to “catch” the system on the edge of Sleep – that was the question to Apriorit Research and Reverse Engineering Department.

Research Tasks

The object for reverse engineering was clear – it was general halmacpi.dll file. There were 2 research goals formulated:

1. Find and describe the particular steps of transition from one power state to another.
2. Detect the particular function in this scheme to hook.

Working On

This reversing task was not usual. The key difficulty was that the methods to be researched could not be debugged, as the system switches off everything when goes to the Sleep state – including debug COM port. So it was simple code browsing without any breakpoints, pure interaction and relation analysis.

The task was even more difficult as halmacpi.dll is a general file with a huge amount of code to analyze.

Researcher managed to reduce the scope for analysis, using the fact that power management methods work only in 16-bit mode, so only methods of such type had to be considered.

Results

The research task was completed in 2 man-weeks. The report described the step-by-step process of transition of Windows 7 x86 from working power state to the suspend-to-RAM state, including low-level instruction to be called.

According to the general task of OS switching and returning requirements, researcher chose the low-level function to hook.

The solution was successfully implemented and helped the Client to make the next stage of the project.

Researched OS characteristics would also help to implement such tasks as the system power state custom optimization, or transfer of running system context to another hardware device after suspending to some removable data storage.