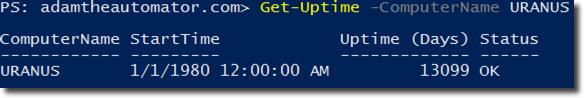
November 2015 Scripting Games Puzzle

Server uptime is the lifeblood of system administrators. We strive on it, get addicted to it..we *need*…*more* server uptime! Don't you think something as addictive and important as server uptime be measured? How do we know we're getting our uptime fix? As that famous quote goes, "Reality does not exist until it's measured.". Let's measure it not only for our own sake but also to give a pretty report to our manager with all those whizbang, doohickey Excel juju that they love to see!

For this month's challenge, I want you to create a PowerShell function that you can remotely point to a Windows server to see how long it has been up for. Here's an example of what it should output.



Requirements:

1. Support pipeline input so that you can pipe computer names directly to it.
2. Process multiple computer names at once time and output each computer's stats with each one being a single object.
3. It should not try to query computers that are offline. If an offline computer is found, it should write a warning to the console yet still output an object but with Status of OFFLINE.
4. If the function is not able to find the uptime it should show ERROR in the Status field.
5. If the function is able to get the uptime, it should show 'OK' in the Status field.
6. It should include the time the server started up and the uptime in days (rounded to 1/10 of a day)
7. If no ComputerName is passed, it should default to the local computer.

Bonus:

1. The function should show a MightNeedPatched property of $true ONLY if it has been up for more than 30 days (rounded to 1/10 of a month). If it has been up for less than 30 days, MightNeedPatched should be $false.