Detecting Sticky Key Backdoors

Article • 10/03/2016

If there is one thing I've learned about cyber security over the last fifteen years, it is that we are stronger as a community than alone. In that spirit, I wanted to share a PowerShell scanner I wrote to detect the presence of a Sticky Key backdoor on a Windows system. You can find a copy of the scanner here:

https://github.com/TrullJ/sticky-keys-scanner

You can run the PowerShell script locally or remotely via the Invoke-Command cmdlet.

Details of Backdoor:

The original attack involved replacing the C:\Windows\System32\sethc.exe binary, or one of the other accessibility suite binaries, with something that could access the underlying OS, like cmd.exe. When the executables are switched, you can bypass the login and get a system level command prompt by pushing the associated "sticky key" - e.g., Shift-key five times in a row quickly if sethc.exe is replaced. Sethc.exe and the other accessibility binaries are executed pre-login, so the attacker would have an immediate backdoor to the system without requiring authentication.

Another variant of the attack works by setting cmd.exe, or another program, as a debugger to one of the accessibility suite binaries. This requires one addition to the registry and the attacker doesn’t need to worry about the file replacement described previously.

For sethc.exe, the registry change might look like the following:

REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /t REG_SZ /v Debugger /d “C:\windows\system32\cmd.exe” /f

Another common variant leverages Utilman, a different part of the accessibility suite. This is identical to the previous attack but sets cmd.exe as the debugger for utilman.exe. The attacker would then push the Windows Key + U and a system level command prompt would present itself. For this particular persistence method, the registry change would look like the following:

REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe" /t REG_SZ /v Debugger /d “C:\windows\system32\cmd.exe” /f

One key note is that these are post compromise indicators. The attacker would have already compromised the systems or had physical access to the machine.

Detecting the Attacks:

There are two approaches for detecting the presence of this particular backdoor.

1. Match Hashes – backdoor exists if one of the binaries in the Microsoft accessibility suite matches a binary that provides system-level access - e.g., cmd.exe or explorer.exe.

2. Registry Analysis – The next method for detecting this backdoor is to review the registry and determine if cmd.exe or another program providing system-level access has been set as the debugger for utilman.exe or sethc.exe. You can also review Windows event logs for a history of registry changes that match this signature; however, this technique is not currently included in the scanner's functionality.

The scanner looks for both backdoor variants - binary replacement and registry modification.

Happy hunting and please share any improvements to the code via github so we can all be stronger together![bing_translator]