Strider *GhostBuster*: Why It's A Bad Idea For Stealth Software To Hide Files

Yi-Min Wang, Binh Vo, Roussi Roussev, Chad Verbowski, and Aaron Johnson

Microsoft Research, Redmond

File-hiding through lying APIs [HTB03, NTI04] is an advanced stealth technique used by many popular system-monitoring software such as RootKits, Trojans, and keyloggers (collectively called *"ghostware"* in this paper) to make executables or data files invisible. Once the ghostware program is started, it intercepts all file queries at a very low level and uses filtering to ensure that a chosen subset of files are never revealed to any file query operations made by any program, not associated with the ghostware, running on the infected machine. This technique can defeat experienced system administrators who search the file system and Windows Registry for suspicious entries, as well as commonly used malware scanning tools that are based on known-bad file signatures.

Most of the existing ghostware detection tools exploit the *imperfection* of today's file-hiding implementations. Although such tools are necessary for combating today's ghostware, they may essentially provide testing resources to help the evolution towards perfect ghostware. In contrast, the Strider *GhostBuster* targets the fundamental weakness of the file-hiding behavior and turns the problem into its own solution [WVR+04].

The basic idea is very simple: since the hidden files are visible before the ghostware is started and become invisible after that, a diff of the two file-system scans before and after should precisely capture all hidden files. A straightforward implementation of this idea consists of three steps: (1) we first boot normally into the infected OS and invoke "dir /s /a" to scan the entire file system. We save the output in a file named, say, "Infected_Scan.txt" on a disk; (2) we reset the machine and this time boot into a clean WinPE CD [WPE] that contains a clean version of *WinDiff.exe*. We invoke "dir /s /a" again and save the output in the file "Clean_Scan.txt". The hidden file should appear in this output because the ghostware was not running during the scan; (3) we invoke *WinDiff.exe* to compare the two files "Infected_Scan.txt" and "Clean_Scan.txt". The diff result should contain all hidden files, plus some minor "noise files" that were updated between the two scans.

We have tested the tool on four file-hiding RootKits/Trojans (Hacker Defender, Aphex, Vanquish, and Msvsres.dll), two file-hiding commercial keyloggers (ActMon and ProBot SE), and four commercial file-hiding security software (Hide Files 3.3, Hide Folders XP, Advanced Hide Folders, File & Folder Protector). In all cases, *GhostBuster* deterministically, efficiently, and effectively detected all hidden files. One significant property of the *GhostBuster* approach is that it does not require a pre-generated, cat-and-mouse known-bad file signature; it simply relies on the hiding behavior as a "precise and universally-bad behavior signature" because no good software should try to hide from the machine administrators.

References

[HTB03] "How to become unseen on Windows NT," http://rootkit.host.sk/knowhow/hidingen.txt.

[NTI04] "NTIllusion: A portable Win32 userland rootkit," Phrack Magazine, July 2004. http://www.phrack.org/phrack/62/p62-0x0c_Win32_Portable_Userland_Rootkit.txt.

[WPE] Microsoft Windows Preinstallation Environment (Windows PE), http://www.microsoft.com/licensing/programs/sa/support/winpe.mspx.

[WVR+04] Y. M. Wang, et al., "*Strider GhostBuster*: Why It's A Bad Idea For Stealth Software To Hide Files," Microsoft Research Technical Report MSR-TR-2004-71, July 2004.