

Hijacking the Hijacker

Eduardo 'edu' Prado
edu@secumania.net

Secumania Security and Vulnerability Research Lab

January 2010

On Windows OS, most of the times people get infected by malware, they can only blame themselves for opening "sexy-pictures.exe" and the like, received by e-mail. Not only can these malwares steal sensitive data and give the creator(s) remote access as well as make the computer a zombie of a botnet at times, but also make the infection obvious by hijacking the web browser and the user's desktop. In the case of hijacking the desktop they usually display crappy porn banners, sometimes they put a message telling the user she/he needs to purchase specific Antivirus software to get rid of the malware installed on the computer as well, so that they are able to steal money and credit card information for example.

Not satisfied in making the infection obvious, malware creators also like to add a startup entry to popular registry locations such as:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

In the "Shell" string value they append data so it will be set to something like this: "[explorer.exe malware.exe](#)" and then they place the file "malware.exe" in the system32 directory and it will be started by Windows Explorer on every boot. As these malwares can, for instance, hijack the desktop, nothing is more fair than hijacking them as a reward, right?

Well, sure! By abusing the creator's lack of knowledge or lack of patience to write code properly/safely, we can hijack their execution in case they edit the above entry (which is widely used by malwares) without even touching the registry or even having administrative privileges, on Windows XP, Win2k, and Win3k versions. All we have to do is to place an executable called "malware.exe" in the current logged on user's base directory (*%userprofile%*). Because Windows Explorer on the above versions of Windows starts in the user's base directory and malware writers usually don't provide a full path to their malwares, thus we're able to successfully hijack the execution of it, rendering it "dead" in the system. It's obviously clear that this approach doesn't necessarily work if the malware writer has added additional startup entries.

This issue happens because of a common vulnerability in Windows OS: "**Relative Paths**"! This arised security problems in the past and is also the cause for another vulnerability that we published in March of 2010, in the HTML Help Control's (hhctrl.ocx) function "HTMLHelpA()" that loads CHM help files from the same directory where the program invoking help starts in.

© Secumania Security and Vulnerability Research Lab
All Rights Reserved