No, I do not want your damned applications in my <appdata> folders! (http://www.sandon.it/node/87)

Posted on: Tue, 03/18/2014 - 22:52 **By:** ldsandon

There is lately a trend, pionereed by applications like Skype and now commonly used by Chrome, that totally ignores Windows guidelines, rules and best practices, installing executables in folders like <appdata> instead of <program files>. The reason is simple: now Windows enforces proper security rights, and thereby unprivileged users are not able to write into folders designed to host executables.

Of course applications like Chrome, designed to steal as many user data as they can, needs a way to install everywhere to perfrom their main task, even if a system is correctly configured to avoid unprivileged users to mess with executable files. Unluckily Windows for compatibiliy reasons allows execution of files in <appdata> - especially because by default in Windows readable files are executable also (these are different privileges, but you need the "advanced" mode to set them separately), and this opened an hole in Windows security (Microsoft One-Click applications exploit as well) which is not properly closed by default yet.

Google spin doctors made a good job in making users believe Chrome is the "best browser" (it isn't, believe me), and Google developers - like others paid to steal your data - had to find a way to get user install Chrome on every machine they use, even those correctly configured not to allow it - and more and more bad developers are following that bad examples because it's easier than developing an application and installing it following Windows rules and best practices.

There are good reason to separate privileged users - those who can change the configuration and state of a machine - from unprivileged ones, that shoud and can not - they shall be able only to "use" a machine, not change it. There is a reason why they are called "privileged" users (administrators) and "unprivileged" users (plain users). That's, for example, why *nix is "more secure" than Windows. It's totaly useless to complain about Windows security if we actively work against it. Sure, Microsoft should configure different, better defaults, but then user and bad developers will complain that bad written applications no longer work, and of course it would be a Windows fault, not theirs.

One of the reason unprivileged users must not be allowed to change executables is it allows an easy escalation of privileges. Most effective attack techniques require elevated privileges to work - and that must be gained somehow. If unprivileged user U can change executables (by writing or modifying it), a malicious user M can lure U into writing a malicious executable E. E could not work under U privileges (i.e., install a kernel module to rootkit the machine and hide its payload activities), but if a privileged user P logs on, and runs the same executable installed by U, now E is able to perform its attack and compromise the system fully. This is especially simple if you install and add shortcuts in the "system" <appdata> folder, which is readable by evey user.

That's why you shoud not install anything outside folders where proper permissions are assigned (<program files> permissions are already set), and if for any reason you have to do (i.e. programs written in languages that still can't understand spaces in paths, because in 1970 nobody used folder

names longer than four letters), you have to ensure those folders have the right permissions, and not "everybody can do whatever he likes". I hope I will have some time to write a series of articles about using Windows security from Delphi code - VCL libraries unluckily still think they're targeting Windows 95 processes and FAT disks only, and there's very little built-in already.

If you need to install an application for a "single user" and not for every user accessing the system, you should use permissions, not attempt to write into folders that have been designed to store data, not executables. If you need application to roam with the user, well, the solution is not to write your executable into a roaming profile - your system administrators may not be happy at all when they discover lot of useless files are moved around - nor your user if ever they need to work over a slow connection. There are better ways using Active Directory to enable user to access the software they need on any machine they're allowed to log on to.

Sure, you may need to write a proper setup and a proper update mechanism. Another reason is the latter - lazy developers find easier to update files outside <program files> because they do not need a properly written setup to perform an update. Is ironic how many people blabble about the security of "app stores" - which force them to properly deploy applications - and then when a store doent's force them, immediately take lazy shortcuts and deploy in silly ways.

My advice is to avoid applications that install where they should not to - they have no any good reason to do so but exploit you - and especially, if you're a software developer, don't do so. Read Windows Guidelines (http://msdn.microsoft.com/en-us/library/windows/desktop/ee915058.aspx) to learn how to deploy your software properly, and follow them. They are often updated as Windows evolve, thereby keep your knowledge up to date.

BTW, we developed an application for the company I work for which is able to neuter any executable found outside allowed paths. And I hope Windows 9 will not allow lazy developers to cripple security any longer.

- Log in (http://www.sandon.it/user/login?destination=/comment/reply/node/87/comment_node_blog%23comment-form) to post comments