Windows XP SP 3 Winlogon.exe Registry Handling Local Privilege Escalation

by Gynvael Coldwind and Matthew Jurczyk
Hispasec

1. Basic information

Name	Microsoft Windows XP Service Pack 3 Winlogon Registry Handling Local Privilege Escalation
Class	Design error
Impact	High
Credits	Gynvael Coldwind, Matthew Jurczyk
Discovered	2009-01-07
Published	2010-05-29

2. Abstract

Microsoft Windows XP is a well known desktop operating system, currently released with Service Pack 3.

The system process responsible for handling user logon and loading the profile is called winlogon.exe. This process can be forced to perform a certain specific operation on the Windows registry, that in consequence allows executing attacker-provided applications with privileges of another user (for example, the administrator), at the time that user logs in.

The vulnerability exploitation requires the attacking user to be able to log in into the system twice, and the administrator user to log into the system once after that (the time of the administrator login is not important).

Older versions of Microsoft Windows (2000, NT4) might also be affected. However, Microsoft Windows Vista is not affected, due to changes in both Winlogon and Windows registry.

3. Vulnerability details

Microsoft Windows Registry possesses a not well known ability to create link keys, that link to another key – it works the same way as file/directory links in a file systems. In Microsoft Windows XP SP3, any user is able to create a link from his HKCU (Current User) registry key to any other key in the system (HKU, HKLM, etc). The link does inherit the rights of the target, meaning if the

link points to a key, where the user has read-only access, the user will still have only read-only access when using the link to access the target.

However, some SYSTEM processes create keys and sets values in the HKCU of the user. One of such processes is the Winlogon.exe proccess. It creates the following keys:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

And inside this key, it sets the following string values:

- **Local AppData**
- **AppData**
- **Cookies**
- **Desktop**
- **Favorites**
- **NetHood**
- **Personal**
- **PrintHood**
- **Recent**
- **SendTo**
- **Start Menu**
- **Templates**
- **Programs**
- **Startup**
- **Local Settings**
- **Local AppData**
- **Cache**
- **History**

The content of these string values is very similar – almost every value leads to a folder within the users home folder. For example, the **AppData** contains the following value:

C:\Documents and Settings\test\Application Data

Winlogon sets these values without dropping the privileges to match the privileges of the user that logs in – it creates these values with SYSTEM privileges.

The attacker can **delete** the **Explorer\Shell Folders** key, and **create a link** with the same name (Shell Folders in HKCU\...\Explorer) to the **HKLM\Software\Microsoft\Windows\CurrentVersion\Run** key (the user does not have write access to this key, but Winlogon process has the required access to write). This will cause the Winlogon, when the attacker will log in again, to follow the link, and create the above values (AppData, and so on)

inside the **Run** key. Every value inside this key is treated as the path and name of an executable, and is executed when any (since it's the HKLM key) user logs in.

The attacker can now create a backdoor application, and name it the following way:

C:\Documents and Settings\test\Application Data.exe

This will cause the backdoor to be run whenever any user logs in, since the path is places inside the Run key. The “Application Data” folder does not have to be removed, since Windows will append the “.exe” extension to the name by default.

An example exploit would retrace the above steps. Please note that it is required for the attacker user to relog after creating the link, since the Winlogon will create the key and values when the attacker logs in. Please also note that it is a good idea to place the backdoor executable where ever possible (according to the Shell Folder list), since if an executable is not found, Windows will open an Explorer window for the given folder. Another thing, is that the backdoor will be run by any logging in user – it should be instructed to act only when a user with higher privileges logs in.

4. Impact

This vulnerability allows a local attacker to get the privileges of another user, but only if that user logs into his account on the same computer.

The impact of a single attack, considering the above requirements, is considered as high. However, if malware would implement this exploit, the impact may change to very high.

5. Disclaimer