Search

Posted on **2017-07-31**
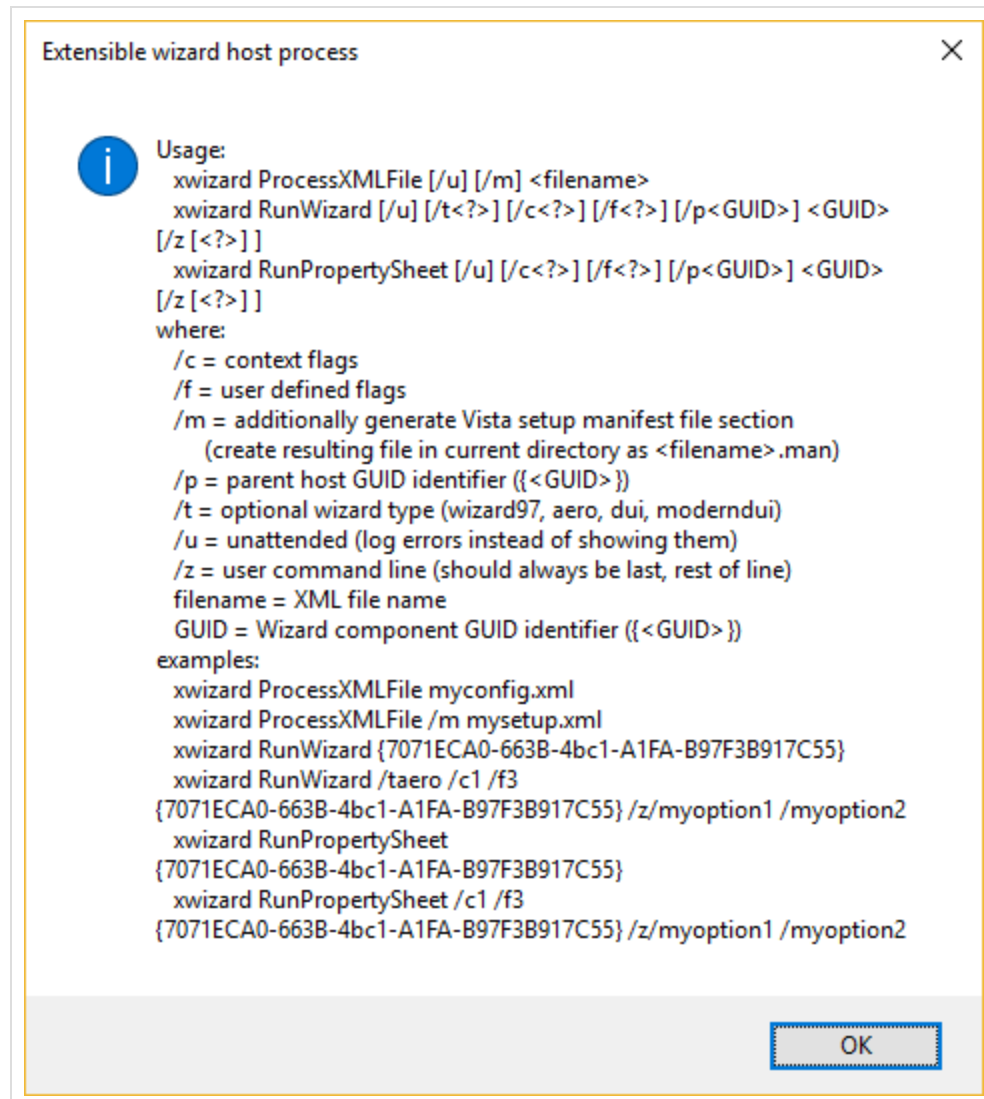
The Wizard of X – Oppa PlugX style

Xwizard is an 'Extensible wizard host process'. While I am not 100% sure what it is doing I know for certain that – whatever it is – PlugX guys would approve.

Why?

When you run it with a '/h' command line parameter, you will get this info:

```
Extensible wizard host process                                    ✕

  ⓘ   Usage:
         xwizard ProcessXMLFile [/u] [/m] <filename>
         xwizard RunWizard [/u] [/t<?>] [/c<?>] [/f<?>] [/p<GUID>] <GUID>
       [/z [<?>] ]
         xwizard RunPropertySheet [/u] [/c<?>] [/f<?>] [/p<GUID>] <GUID>
       [/z [<?>] ]
       where:
        /c = context flags
        /f = user defined flags
        /m = additionally generate Vista setup manifest file section
            (create resulting file in current directory as <filename>.man)
        /p = parent host GUID identifier ({<GUID>})
        /t = optional wizard type (wizard97, aero, dui, moderndui)
        /u = unattended (log errors instead of showing them)
        /z = user command line (should always be last, rest of line)
        filename = XML file name
        GUID = Wizard component GUID identifier ({<GUID>})
       examples:
         xwizard ProcessXMLFile myconfig.xml
         xwizard ProcessXMLFile /m mysetup.xml
         xwizard RunWizard {7071ECA0-663B-4bc1-A1FA-B97F3B917C55}
         xwizard RunWizard /taero /c1 /f3
       {7071ECA0-663B-4bc1-A1FA-B97F3B917C55} /z/myoption1 /myoption2
         xwizard RunPropertySheet
       {7071ECA0-663B-4bc1-A1FA-B97F3B917C55}
         xwizard RunPropertySheet /c1 /f3
       {7071ECA0-663B-4bc1-A1FA-B97F3B917C55} /z/myoption1 /myoption2


                                                    [    OK    ]
```

Something about the unusual command line parameters described there caught my eye.

After a quick inspection I discovered why. The arguments are actually… names of functions exported from xwizards.dll!

Very nice!

And even nicer is the fact the LoadLibraryEx that loads that xwizards.dll finds its conveniently in the current path…

Ouch…

So… all you have to do is copy c:\WINDOWS\system32\xwizard.exe to your folder, drop your xwizards.dll DLL there and call xwizard.exe with at least two arguments.

And the Microsoft-signed xwizards.exe will load xwizards.dll of your choice…

This entry was posted in **Anti-***, **Compromise Detection**, **Forensic Analysis**, **Incident Response**, **Living off the land**, **Malware Analysis** by **adam**. Bookmark the **permalink**.