Cracking .msi Files
By mMhCkB for RET

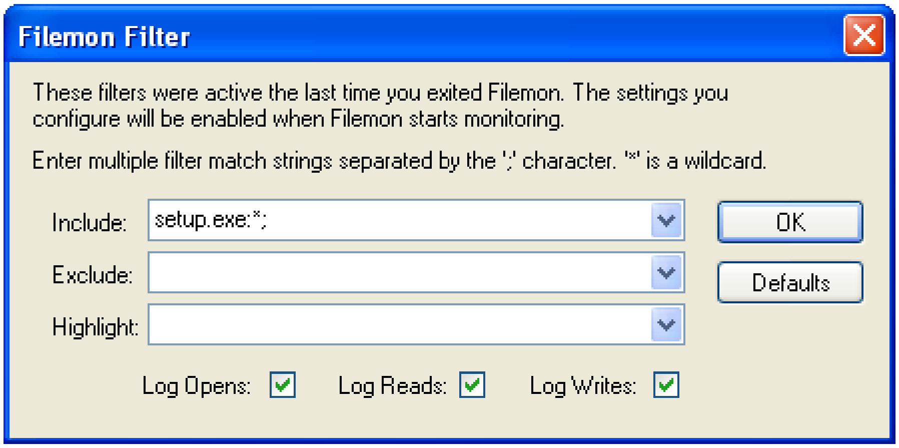
Tools required:

Wise for Windows Installer
Filemon (Optional)

Today, we are discussing how to bypass serial number protections built in to windows binary installer files (.msi). Commonly, registration number protections are embedded within an InstallShield script, so we are going to make sure this is not the case before we delve into the .msi file.

If the installer is only a .msi file, or rather, if there is no .exe installer within the package, skip the entire filemon/inx hunting procedure.

Start Filemon and add setup.exe:*; (or whatever the name of your installer exe is) into the include list of the filter and hit ok.



Filemon Filter

These filters were active the last time you exited Filemon. The settings you configure will be enabled when Filemon starts monitoring.

Enter multiple filter match strings separated by the ';' character. '*' is a wildcard.

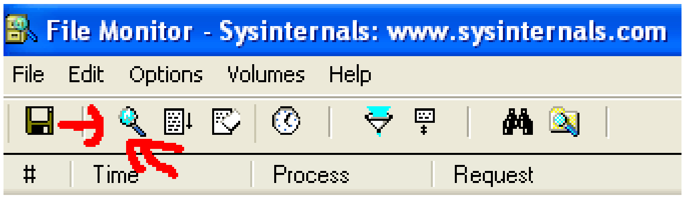
Include: setup.exe:*;

Exclude:

Highlight:

Log Opens: ☑ Log Reads: ☑ Log Writes: ☑

Run your setup exe until you reach the screen which asks for you to input the serial number. Back in filemon, hit the capture button.



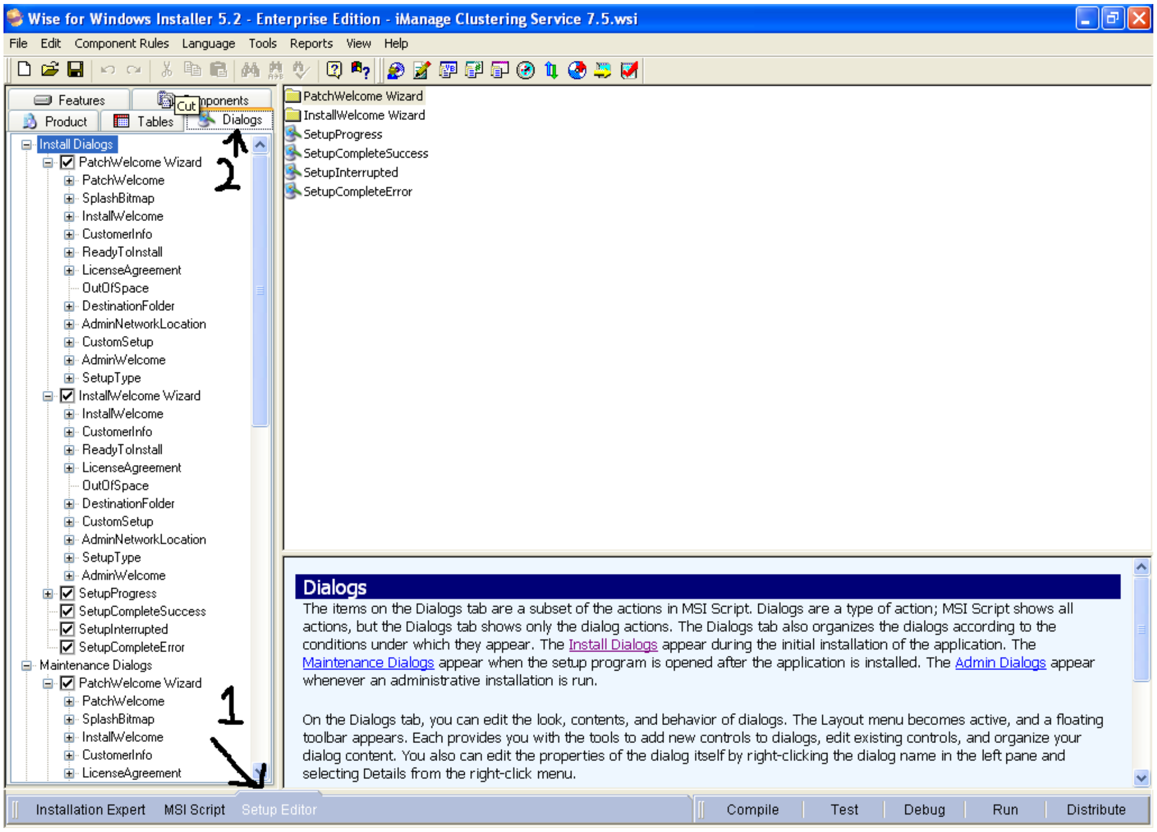
Edit->Find. Search for ".inx" and ".ins" . Those are the file extensions for InstallShield script files. Commonly, you can skip the filemon procedure and find it directly by looking in your temp folder, but not always. You can access the temp folder by going to Start -> Run: %temp% snoop around there and you may find an InstallShield script file. If a script file is found, odds are the protection lies within the script file and you can close this tut right now, if not, continue on. ☺

If you skipped the filemon step, continue from here:

Alright, on to the .msi cracking. In our standard generic example, the .msi installer loads, yet refuses to install until a valid serial number is entered into an input box. We need to bypass this.

Load up Wise for Windows Installer.
File, Open, choose your .msi file.
It will ask you if you wish to convert the .msi file to a .wsi wise installer file. Choose yes. Choose your source directory (doesn’t matter what you pick, when we’re done, we’ll delete it) and hit ok. Keep hitting next until it pops up and says the directory you’re trying to use doesn’t exist and if you want to create it, hit yes and continue. Wise will proceed to “decompile” your .msi file.

1. In the lower left, hit the Setup Editor tab.
2. Up top, hit the Dialogs tab.

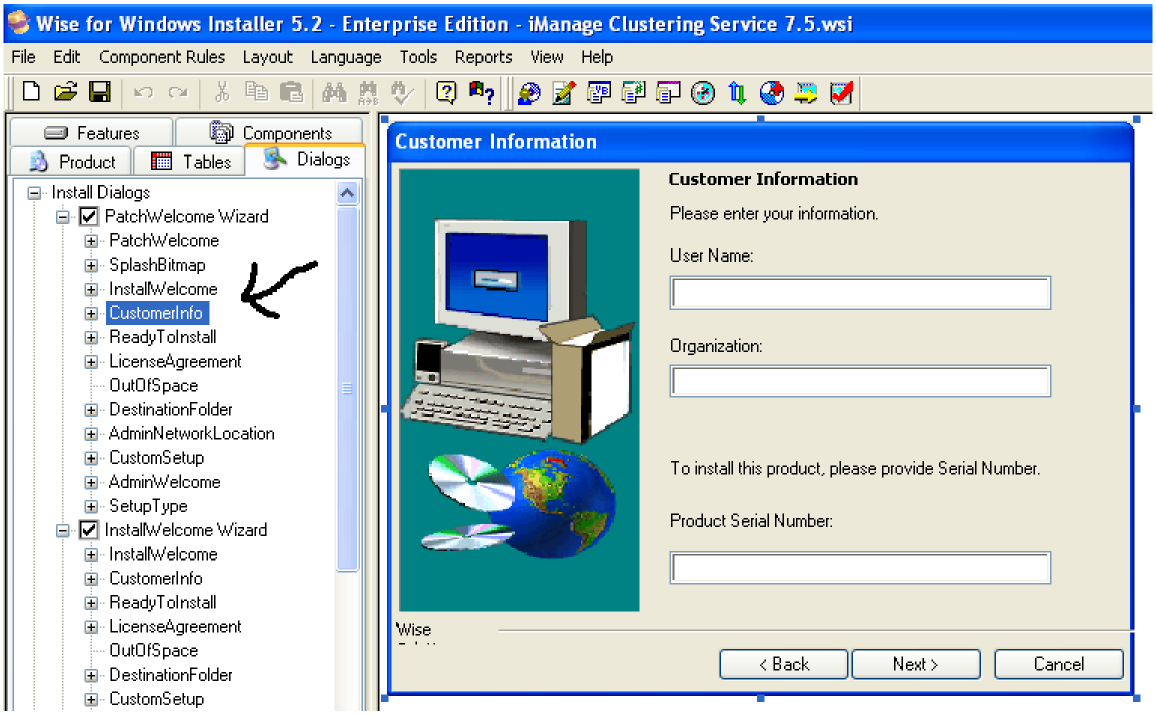


Dialogs

The items on the Dialogs tab are a subset of the actions in MSI Script. Dialogs are a type of action; MSI Script shows all actions, but the Dialogs tab shows only the dialog actions. The Dialogs tab also organizes the dialogs according to the conditions under which they appear. The Install Dialogs appear during the initial installation of the application. The Maintenance Dialogs appear when the setup program is opened after the application is installed. The Admin Dialogs appear whenever an administrative installation is run.

On the Dialogs tab, you can edit the look, contents, and behavior of dialogs. The Layout menu becomes active, and a floating toolbar appears. Each provides you with the tools to add new controls to dialogs, edit existing controls, and organize your dialog content. You also can edit the properties of the dialog itself by right-clicking the dialog name in the left pane and selecting Details from the right-click menu.

On the left bar, click down through the different dialogs until you find the one which contains the serial number input box.



Customer Information

Please enter your information.

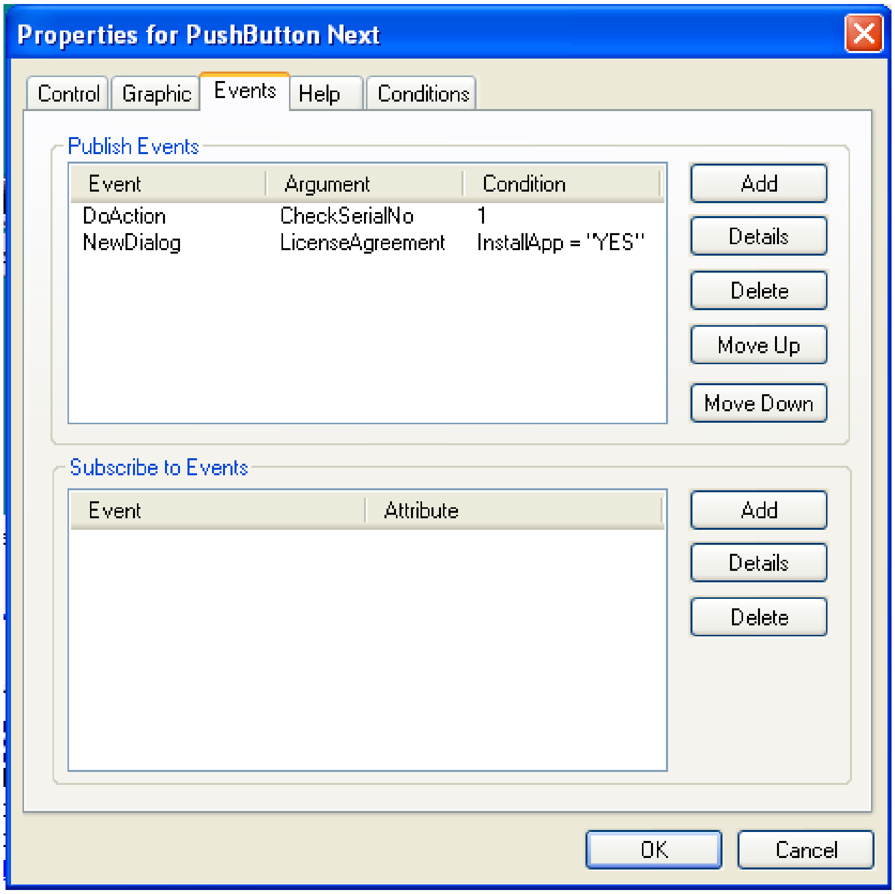
User Name:

Organization:

To install this product, please provide Serial Number.

Product Serial Number:

Double click on the next button, and it will bring up items properties window.



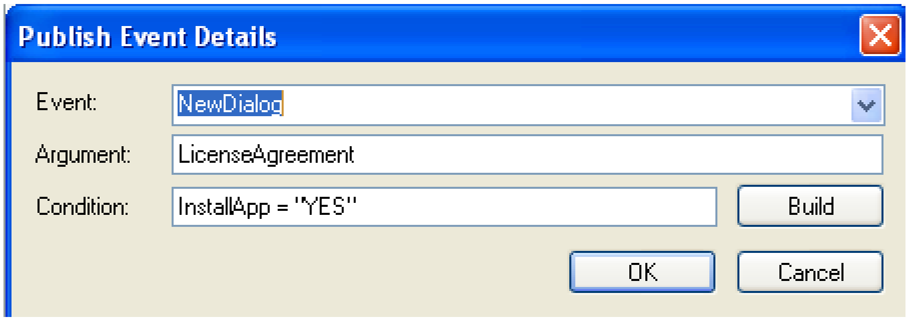
Event	Argument	Condition
DoAction	CheckSerialNo	1
NewDialog	LicenseAgreement	InstallApp = "YES"

Event	Attribute

So, when you hit that next button, it calls a function called CheckSerialNo with the condition set to 1. The condition set to one means, always true, thus, it will always perform this check. You can think of this similar to an infinite loop you might write in C that looks something like:

While(1) blah;

After the CheckSerialNo call, it moves on to the next event. The event is a NewDialog (Proceeding on to the next screen instead of saying bad serial), which calls the dialog named LicenseAgreement. The condition is set to InstallApp = "YES". Apparently, in the CheckSerialNo function, it sets a variable called InstallApp equal to "YES" if it's a valid serial and "NO" if the serial is invalid. So, basically, it would move to the next dialog if the CheckSerialNo function had set InstallApp equal to "YES", unfortunately, since we don't have a real serial number, this isn't the case. So, in order to fix this, double click on the "New Dialog" line.

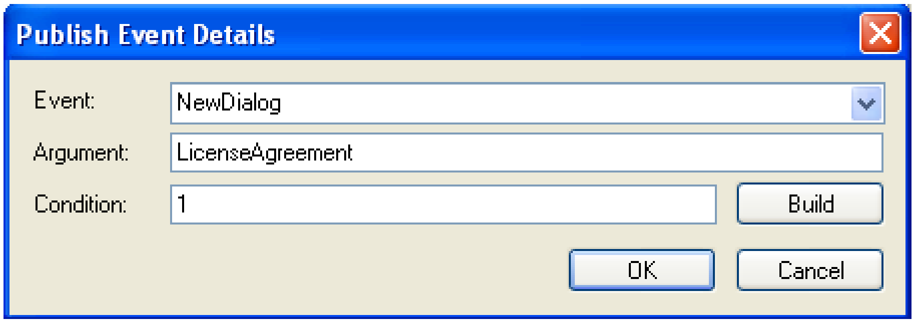


Event: NewDialog

Argument: LicenseAgreement

Condition: InstallApp = "YES"

Change the condition to 1, so it is always true, and hit ok.



Event: NewDialog

Argument: LicenseAgreement

Condition: 1

We've now successfully recoded our .msi file to accept any serial entered. All that is left is to recompile it. So, hit the Compile button down in the lower right. Go to the directory you originally chose to use and inside should be your new .msi file. If the original installer used a .cab file, there will also be a new .cab file in this directory, which has the same filename as the .msi. Copy these two (or one if no cab file was used) files to the original installation file folder overwriting the old .msi file and the old .cab file. Run your installer and enjoy entering any serial number that you wish.

Greetings to: pale, zero, the rest of the nts guys, everyone in ret, everyone else I talk to regularly and all the other guys out there who make the 0day scene a competition.

mM