HOW TO MAKE A BYTE PATCHER

by Daniel Middelhede alias Thefool
danielm@mail.dk

This is my first tutorial. Its just a short one, but its pretty easy to understand, even though my English is not the best. I tried to make it for newbies.

You will need the following tools:

-Purebasic
Shareware
This is the programming language i chosed. More about that later.
Trial from http://www.purebasic.com

Why i choose PureBasic

A nice and easy language. Very advanced, but easy for the beginner.
If it wasnt advanced, how could you make an byte patcher :)
It makes !SMALL! Executables!
The bytepatcher is 5.53 kb in size (real size). Thats what i call small.
And !FAST! Too! Comparable to C++, if not faster.
If you like programming, give it a try. I am VERY happy with that language.

The program to test it on

Its very important to have something to test it on. The source is pretty simple:

```
;Open an msg dialog:
MessageRequester("daniels software","REGISTER")
;Get the serial
serial.s=InputRequester("daniels software","serial number:","")
;Test
If serial.s="fjolle"
MessageRequester("daniels software","Serial OK")
Else
MessageRequester("daniels software","Wrong serial")
EndIf
```

That was the test program. Pretty easy. I added some comments. Easy to understand. It just stores a keyboard input in the variable serial.s and compares is. This is just a simple example, and is not at all meant to be secure. It isnt. Just for testing! Compile it. Tested and works on 3.81.

Finding the place to place the bytes

This is not a cracking tutorial. So i will just tell you a little. Using an Disassembler, you can see the assembler commands. When an equation is made, and jump command if often seen in assembler. I've found the offset(position in file) for the IF equation in the test file. The offsets are:

1.one: 6FB
2. one: 6FC

Both need to be replaced with the NOP command, that dosent do anything.
To do that, we need to write the byte-length hex value 90 to the two offsets.

Thats why i've asked the purebasic comunity, and Paul answered with a simple one. I was too stupid to figure this simple one out myself. Thanks Paul :)
It is simple, take a look:

```
;This is a procedure made with help from Paul, that patches the file.
Procedure.l Patch(file.s,location.l,byte.b)
;Make an backup of the file
CopyFile(file,file+".bak")
;Open file
  If OpenFile(0,file)
   ;Find the place with the old bytes
    FileSeek(location)
    ;Write new bytes
    WriteData(@byte,1)
    ;And close it
    CloseFile(0)
    ;Return 1 if this was ok.
    ProcedureReturn 1
  EndIf
EndProcedure

;How to use the procedure:
;Patch(filename,offset,byte)
;When writing HEX in purebasic, put an $ before
;the value. Simple!

;Debug is written before patch, so the returned value
;gets by the debugger, wich reports the number to the
;programmer. This can easly be removed.
Debug Patch("test.exe",$6FB,$90)
Debug Patch("test.exe",$6FC,$90)
```

A lot of comments in this code. The real code is only a few lines!
Pretty simple.- Opens file. Finds the place and puts the data.

Try it

Try to compile the test file and this patcher. Try the test before running the patcher.
The test should only say Nice Serial, if you write the serial, that is fjolle. If not,
it returns a Wrong Serial message. Try run the patcher. It should now return Serial OK to all serials! It should work. Be sure the test is named test.exe and the patcher is in the same directory.
There is no gui and info to the user. This is a Binary Patcher tutorial.
Thanks for reading. Hope you enjoyed :)

Thanks:

Fred, for making purebasic
All others who maked purebasic.
Paul, giving me a patcher snippet, the idea for this tutorial.
And all members of the purebasic comunity, youre the best!
Anyone else in the world who ever helped me and was nice to me :)

-Daniel middelhede [THEFOOL]
danielm@mail.dk