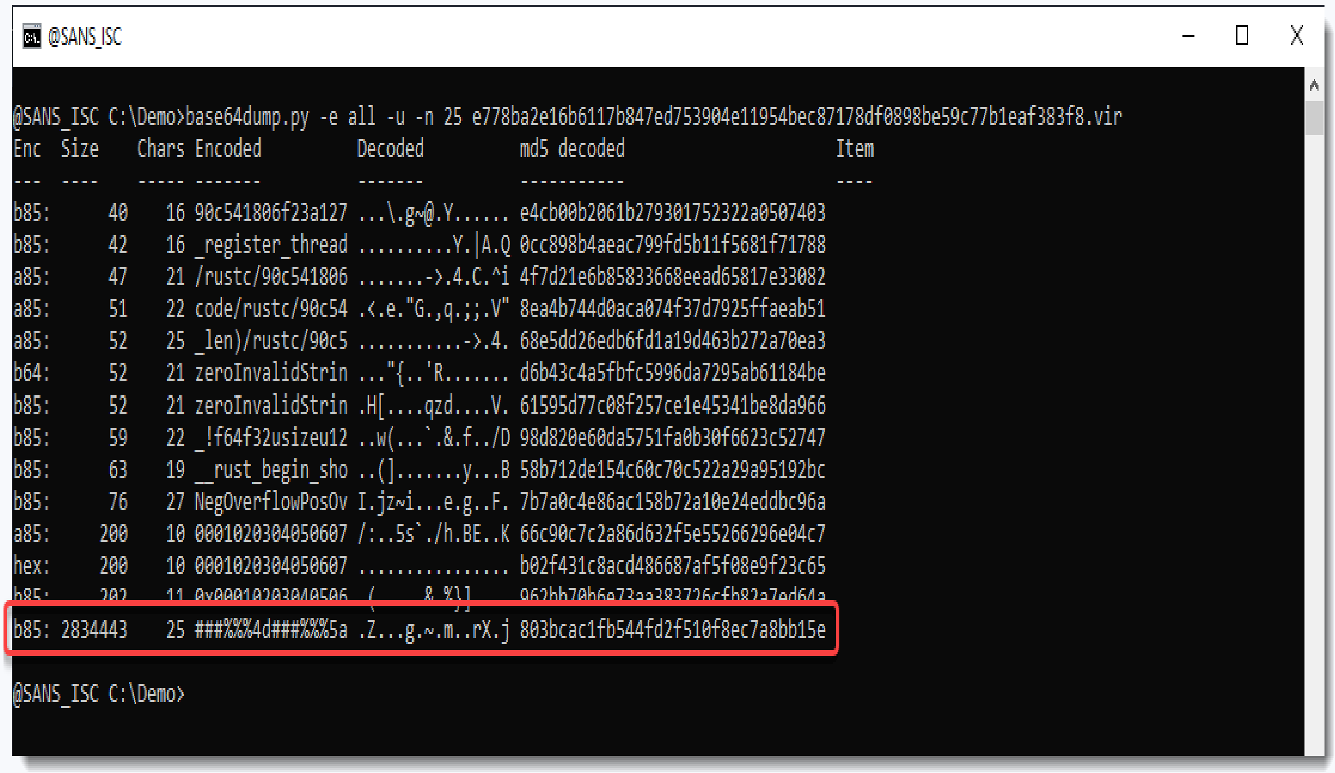
Obfuscated Hexadecimal Payload

Published: 2024-03-16
Last Updated: 2024-03-17 00:36:54 UTC
by Didier Stevens (Version: 1)

0 comment(s)

This PE file contains an obfuscated hexadecimal-encoded payload. When I analyze it with base64dump.py searching for all supported encodings, a very long payload is detected:

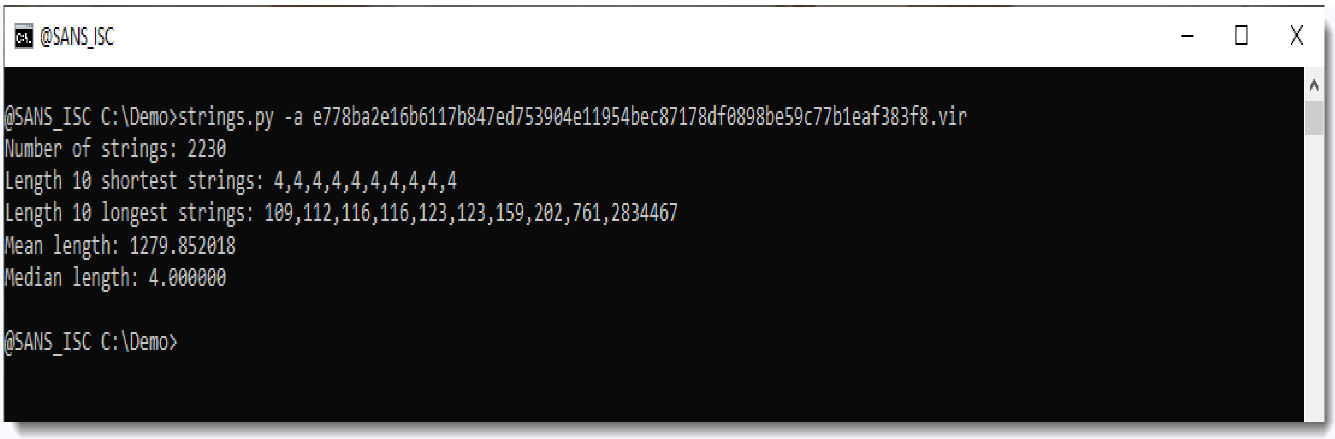


```
@SANS_ISC C:\Demo>base64dump.py -e all -u -n 25 e778ba2e16b6117b847ed753904e11954bec87178df0898be59c77b1eaf383f8.vir
Enc  Size    Chars Encoded          Decoded          md5 decoded                      Item
---  ----    ----- -------          -------          -----------                      ----
b85:      40    16 90c541806f23a127 ...\.g~@.Y...... e4cb00b2061b279301752322a0507403
b85:      42    16 _register_thread .........Y.|A.Q 0cc898b4aeac799fd5b11f5681f71788
a85:      47    21 /rustc/90c541806 .......->.4.C.^i 4f7d21e6b85833668eead65817e33082
a85:      51    22 code/rustc/90c54 .<.e."G.,q.;;.V" 8ea4b744d0aca074f37d7925ffaeab51
a85:      52    25 _len)/rustc/90c5 ...........->.4. 68e5dd26edb6fd1a19d463b272a70ea3
b64:      52    21 zeroInvalidStrin ..."{..'R....... d6b43c4a5fbfc5996da7295ab61184be
b85:      52    21 zeroInvalidStrin .H[....qzd....V. 61595d77c08f257ce1e45341be8da966
b85:      59    22 _!f64f32usizeu12 ..w(...`.&.f../D 98d820e60da5751fa0b30f6623c52747
b85:      63    19 __rust_begin_sho ..(]......y...B 58b712de154c60c70c522a29a95192bc
b85:      76    27 NegOverflowPosOv I.jz~i...e.g..F. 7b7a0c4e86ac158b72a10e24eddbc96a
a85:     200    10 0001020304050607 /:..5s`./h.BE..K 66c90c7c2a86d632f5e55266296e04c7
hex:     200    10 0001020304050607 ................ b02f431c8acd486687af5f08e9f23c65
b85:     202    11 0x00010203040506 .(.....&.%}]... 962bb70b6e73aa383726cfb82a7ed64a
b85: 2834443    25 ###%%%4d###%%%5a .Z...g.~.m..rX.j 803bcac1fb544fd2f510f8ec7a8bb15e

@SANS_ISC C:\Demo>
```

It's 2834443 characters long, and matches base85 encoding (b85), but this is likely a false positive, as base85 uses 85 unique characters (as its name suggests), but in this particular encoded content, only 23 unique characters are used (out of 85).

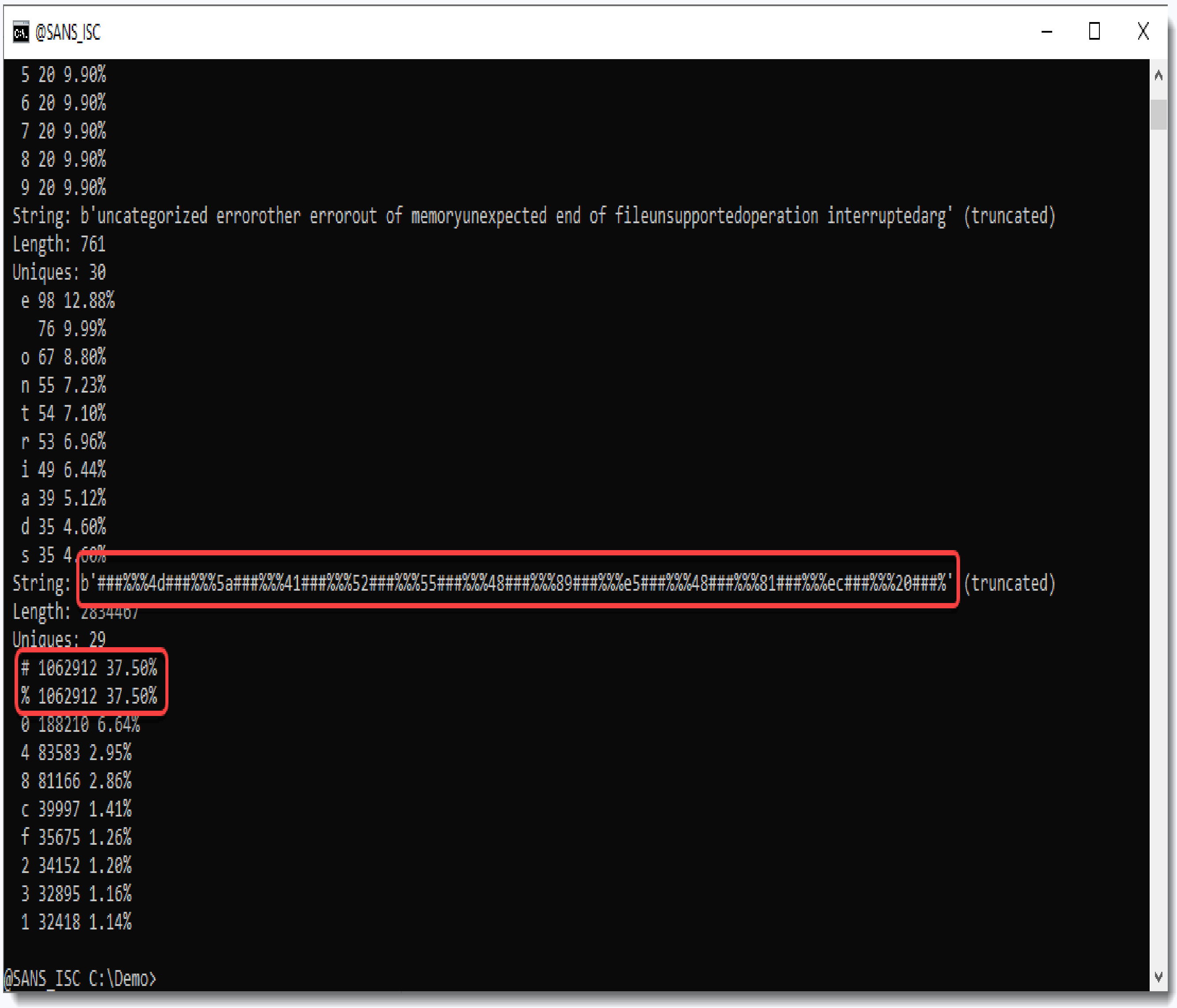
Analyzing the PE file with my strings.py tool (calculating statistics with option -a) reveals it does indeed contain one very long string:



```
@SANS_ISC C:\Demo>strings.py -a e778ba2e16b6117b847ed753904e11954bec87178df0898be59c77b1eaf383f8.vir
Number of strings: 2230
Length 10 shortest strings: 4,4,4,4,4,4,4,4,4,4
Length 10 longest strings: 109,112,116,116,123,123,159,202,761,2834467
Mean length: 1279.852018
Median length: 4.000000

@SANS_ISC C:\Demo>
```

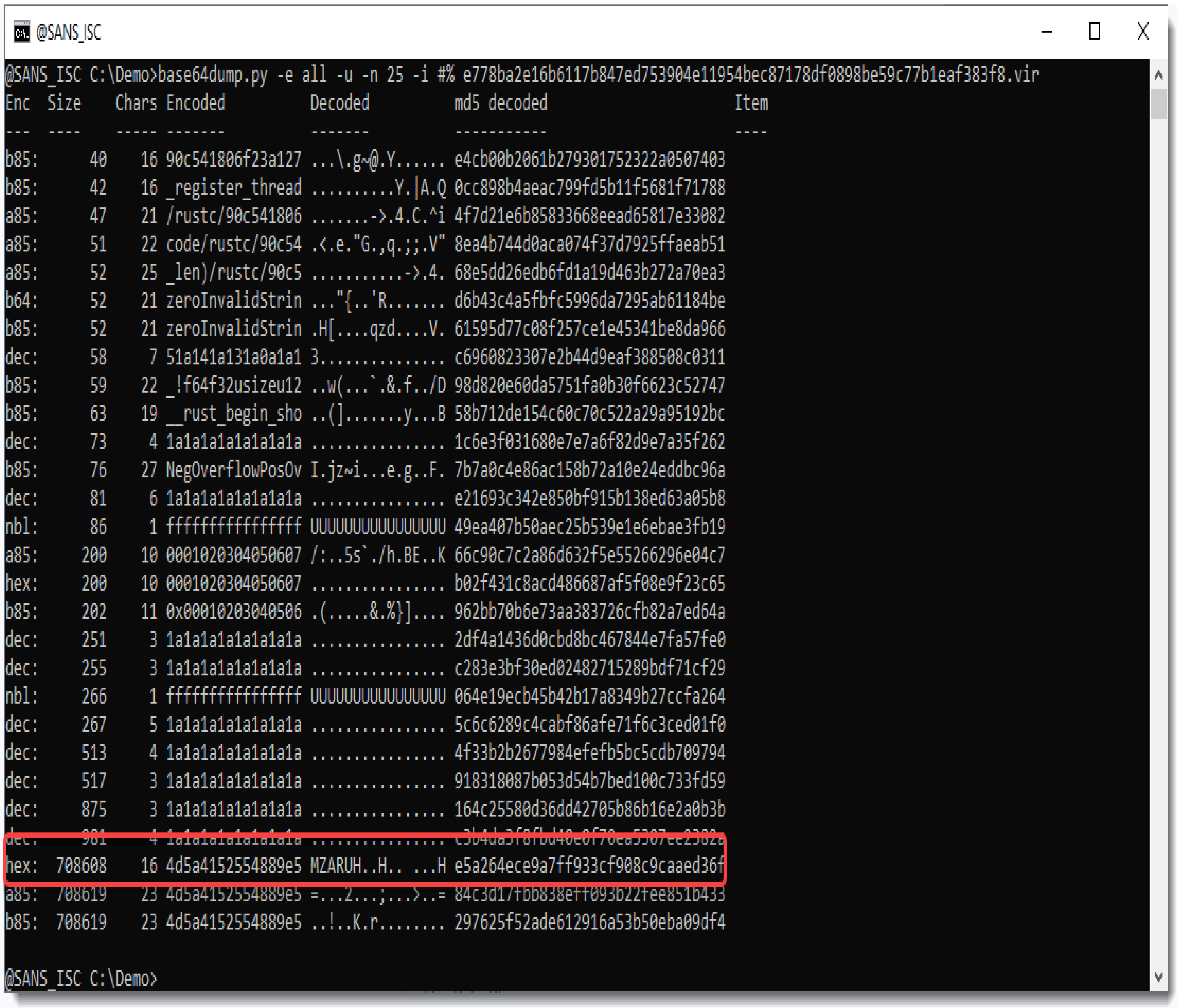
Verbose mode (-V) gives statistics for the 10 longests strings. We see that 2 characters (# and %) appear very often in this string, more than 75% of this long string is made up of these 2 characters:



```
 5 20 9.90%
 6 20 9.90%
 7 20 9.90%
 8 20 9.90%
 9 20 9.90%
String: b'uncategorized errorother errorout of memoryunexpected end of fileunsupportedoperation interruptedarg' (truncated)
Length: 761
Uniques: 30
 e 98 12.88%
   76 9.99%
 o 67 8.80%
 n 55 7.23%
 t 54 7.10%
 r 53 6.96%
 i 49 6.44%
 a 39 5.12%
 d 35 4.60%
 s 35 4.60%
String: b'###%%%4d###%%%5a###%%%41###%%%52###%%%55###%%%48###%%%89###%%%e5###%%%48###%%%81###%%%ec###%%%20###%' (truncated)
Length: 2834467
Uniques: 29
 # 1062912 37.50%
 % 1062912 37.50%
 0 188210 6.64%
 4 83583 2.95%
 8 81166 2.86%
 c 39997 1.41%
 f 35675 1.26%
 2 34152 1.20%
 3 32895 1.16%
 1 32418 1.14%

@SANS_ISC C:\Demo>
```

These 2 characters are likely inserted for obfuscation. Let's use base64dump.py and let it ignore these 2 characters (-i #%"):



```
@SANS_ISC C:\Demo>base64dump.py -e all -u -n 25 -i #% e778ba2e16b6117b847ed753904e11954bec87178df0898be59c77b1eaf383f8.vir
Enc  Size    Chars Encoded          Decoded          md5 decoded                      Item
---  ----    ----- -------          -------          -----------                      ----
b85:      40    16 90c541806f23a127 ...\.g~@.Y...... e4cb00b2061b279301752322a0507403
b85:      42    16 _register_thread .........Y.|A.Q 0cc898b4aeac799fd5b11f5681f71788
a85:      47    21 /rustc/90c541806 .......->.4.C.^i 4f7d21e6b85833668eead65817e33082
a85:      51    22 code/rustc/90c54 .<.e."G.,q.;;.V" 8ea4b744d0aca074f37d7925ffaeab51
a85:      52    25 _len)/rustc/90c5 ..........->.4. 68e5dd26edb6fd1a19d463b272a70ea3
b64:      52    21 zeroInvalidStrin ..."{..'R....... d6b43c4a5fbfc5996da7295ab61184be
b85:      52    21 zeroInvalidStrin .H[....qzd....V. 61595d77c08f257ce1e45341be8da966
dec:      58     7 51a141a131a0a1a1 3............... c6960823307e2b44d9eaf388508c0311
b85:      59    22 _!f64f32usizeu12 ..w(...`.&.f../D 98d820e60da5751fa0b30f6623c52747
b85:      63    19 __rust_begin_sho ..(]......y...B 58b712de154c60c70c522a29a95192bc
dec:      73     4 1a1a1a1a1a1a1a1a ................ 1c6e3f031680e7e7a6f82d9e7a35f262
b85:      76    27 NegOverflowPosOv I.jz~i...e.g..F. 7b7a0c4e86ac158b72a10e24eddbc96a
dec:      81     6 1a1a1a1a1a1a1a1a ................ e21693c342e850bf915b138ed63a05b8
nbl:      86     1 ffffffffffffffff UUUUUUUUUUUUUUUU 49ea407b50aec25b539e1e6ebae3fb19
a85:     200    10 0001020304050607 /:..5s`./h.BE..K 66c90c7c2a86d632f5e55266296e04c7
hex:     200    10 0001020304050607 ................ b02f431c8acd486687af5f08e9f23c65
b85:     202    11 0x00010203040506 .(.....&.%}].... 962bb70b6e73aa383726cfb82a7ed64a
dec:     251     3 1a1a1a1a1a1a1a1a ................ 2df4a1436d0cbd8bc467844e7fa57fe0
dec:     255     3 1a1a1a1a1a1a1a1a ................ c283e3bf30ed02482715289bdf71cf29
nbl:     266     1 ffffffffffffffff UUUUUUUUUUUUUUUU 064e19ecb45b42b17a8349b27ccfa264
dec:     267     5 1a1a1a1a1a1a1a1a ................ 5c6c6289c4cabf86afe71f6c3ced01f0
dec:     513     4 1a1a1a1a1a1a1a1a ................ 4f33b2b2677984efefb5bc5cdb709794
dec:     517     3 1a1a1a1a1a1a1a1a ................ 918318087b053d54b7bed100c733fd59
dec:     875     3 1a1a1a1a1a1a1a1a ................ 164c25580d36dd42705b86b16e2a0b3b
dec:     981     4 1a1a1a1a1a1a1a1a ................ c3b4da3f8fbd40e0f70ea5307ee2382a
hex:  708608    16 4d5a4152554889e5 MZARUH..H.. ...H e5a264ece9a7ff933cf908c9caaed36f
a85:  708619    23 4d5a4152554889e5 =...2...;...>..= 84c3d17fbb838eff093b22fee851b433
b85:  708619    23 4d5a4152554889e5 ..!..K.r........ 297625f52ade612916a53b50eba09df4

@SANS_ISC C:\Demo>
```

Now we have a hex encoded payload that decodes to a PE file (MZ), and most likely a Cobalt Strike beacon (MZARUH).

Didier Stevens
Senior handler
blog.DidierStevens.com