

# Analysing the HotKeyCamo

-A GUI shell for compiling AutoHotKey scripts -

Related Forum/download:

<http://www.autohotkey.com/forum/topic49952.html>

Okay I launched a Diff on the same script.

Once compiled with the 'normal' AHK-Compiler and the other time with HotKeyCamo.

(Make sure the ahkExe is uncompressed. May be decompress it with 'Upx -d \*.exe')

Okay here are the changes...

...and how to deal with them to be able to decompile the script with myAutToExe

Hmm well for better understand I put that part on the top(even when it comes last in the File)

## The Script (Camo vs Normal)

64DF0: 00 00 00 00 00 00 00 00	64DF0: 00 00 00 00 00 00 00 00
64DF8: 00 00 00 00 00 00 00 00	64DF8: 00 00 00 00 00 00 00 00
64E00: 7A FF 02 72 C2 D9 55 17  zyOrÅÜÜÜ	64E00: A3 48 4B BE 98 6C 4A A9  fHK*~lJ@
64E08: 63 46 5C 2E 9D D0 2F 97  cF\..00/-	64E08: 99 4C 53 0A 86 D6 48 7D  "LSD+ÜH)
64E10: 03 B7 BF 84 94 15 89 DA  □•¿~□%Ú	64E10: 03 C1 FA 00 00 FF 6D B0  □Áú ým"
64E18: 1E 58 A6 54 95 FB FA 83  □XIT•0uf	64E18: CE AF 29 00 00 C3 47 1A  î~) ÅC□
64E20: CC 96 F1 E7 78 D9 E3 26  î-ĤgxÜ&	64E20: 79 A1 FF 9A DC CB 0F 21  y;ÿšÜE□!

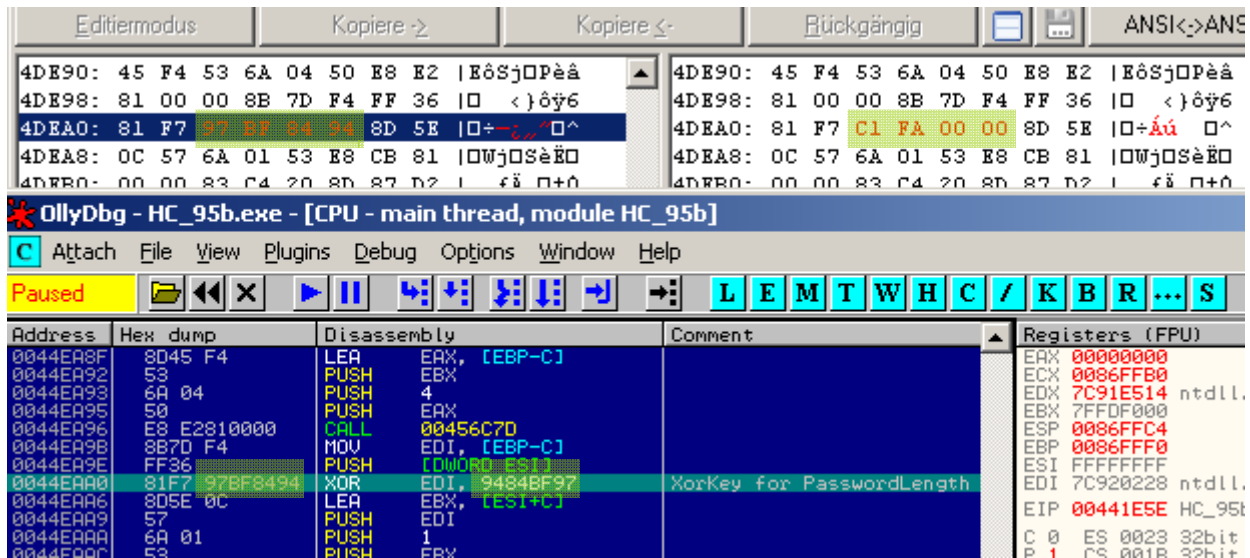
The Compiled Script AutoIT File format:

-----

AutoIt_Signature	size 0x10 Bytes	String "fHK..."
SubType	size 0x1 Byte	Should be 0x03 (0x01 AutoIT2; )
PassphraseLen	size 0x4 Bytes	[XorKey=0x000FAC1]
Passphrase	size (depends on PassphraseLen)	[StrKey=C3D2]
ResType	size 0x4 Byte	eString: "FILE" [
StrKey=16FA]		
ScriptType	eString ">AUTOIT SCRIPT<"	[LenKey=29BC,
StrKey=A25E]		
CompiledPathName	eString "C:\...\Temp\aut26A.tmp"	[LenKey=29AC,
StrKey=F25E]		
IsCompressed	size 0x1 Byte	
ScriptSize	Compressed size 0x4 Byte	[XorKey=45AA]

Now the changes in the AHK-Interpreter stub that's at the beginning of each AHKExe

## 1. LenKeyXorValue Change



Critical Mod:

This Value is used to get the len of the Passphrase. (see below for FileFormat details)

MyAutToExe needs to know that length to correctly read in the Password and the data that follows.

So here you'll need to go into the source code

Search for 'FAC1' in the whole Project until you get here:

```
' ==> Get Script Password
Dim MD5PassphraseHash As New StringReader
If bIsOldScript Then
    ' Old AutoIT Script if branch...
    ' Move three bytes back since SubType is only 1 Byte but before we read 4
byte
    .Move -3
    MD5PassphraseHash = GetEncryptStr(64193, 50130, File) '&HFAC1, &HC3D2
```

Change the last line to

```
MD5PassphraseHash = GetEncryptStr(&H9484BF97, 50130, File) '&HFAC1,
&HC3D2
```

Now it should work.

**Decompiled script should look like that:**

```
; HotkeyCamo ~0.9.5.0>
DetectHiddenText, On
SetTi...
```

**Hint on finding the '9484BF97' value.**

Note that this is the len of the Passphrase. Usually that value will be in a range of 0 to 255 (0x000000 00 to 0x000000 FF).

So the last three bytes will be the same nearly all the time xx 00 00 00 .

Go to ScriptStart +0x11 there is 'B7 BF 84 94' or '9484BFB7'. So a search for the hexstring 'BF 84 94' in the uncompressed \*.ahkExe will reveal that there is a 97 before them and so the Full XorKeyValue is 97 BF 84 94 -> '9484BF97'.

And well there's as well an alternative in case you somehow can't find this '9484BF97' value (or I explained it too messy)

Change the code as the following:

```
' ==> Get Script Password
Dim MD5PassphraseHash As New StringReader
```

```

If bIsOldScript Then
    ' Old AutoIT Script if branch...
    ' Move three bytes back since SubType is only 1 Byte but before we read 4
byte
    .Move -3
    'MD5PassphraseHash = GetEncryptStr(64193, 50130, File) '&HFAC1, &HC3D2
    'MD5PassphraseHash = GetEncryptStr(&H9484BF97, 50130, File) '&HFAC1,
&HC3D2

    .Move 4

    Dim StrLen&
    StrLen = 32

    MD5PassphraseHash = Decrypt(.FixedString(StrLen), 50130 + StrLen)
-> guess/changing the StrLen = 32 that long till it fits.

```

Okay now the rest of the changes:

## 2. MainScript FileName messed up

5AE50: 4A 6F 69 6E 00 00 00 00  Join	5AE50: 4A 6F 69 6E 00 00 00 00  Join
5AE58: 2A 2F 00 00 2F 2A 00 00  */ /*	5AE58: 2A 2F 00 00 2F 2A 00 00  */ /*
5AE60: D9 4C 97 95 6B 61 51 DE  ÜL-*kaQf	5AE60: 3E 41 48 4B 20 57 49 54  >AHK WIT
5AE68: 3A B8 A8 8B 84 EC E9 00  :,"<,,ié	5AE68: 48 20 49 43 4F 4E 3C 00  H ICON<
5AE70: F8 42 64 09 14 23 D6 5C  Bd 0#0\	5AE70: 3E 41 55 54 4F 48 4F 54  >AUTOHOT
5AE78: 8A 95 08 34 85 0C A4 FA  Š•□4...mú	5AE78: 4B 45 59 20 53 43 52 49  KEY SCRI
5AE80: 40 95 FA 00 45 58 45 20  @•ú EXE	5AE80: 50 54 3C 00 45 58 45 20  PT< EXE
5AE88: 63 6F 72 72 75 70 74 65  corrupte	5AE88: 63 6F 72 72 75 70 74 65  corrupte
5AE90: 64 00 00 00 52 45 47 5F  d REG_	5AE90: 64 00 00 00 52 45 47 5F  d REG_
5AE98: 52 45 53 4F 55 52 43 45  RRSNTTRCR	5AE98: 52 45 53 4F 55 52 43 45  RRSNTTRCR

Uncritical:

Hmm well it will mess up detection what script 'flavor' this script is and so the decompiled file will get the extension \*.au3 (since AutoIT is the standard). But I think you can handle that ;)

Hint: Rename \*.au3 -> \*.ahk

## 3. AHK-Script Start Pattern; FILE Start Pattern and Start Pattern for compressed data are messed

628C8: 01 00 00 00 04 00 00 00  □ □	628C8: 01 00 00 00 04 00 00 00  □ □
628D0: 06 00 00 00 02 00 00 00  □ □	628D0: 06 00 00 00 02 00 00 00  □ □
628D8: 04 00 00 00 63 46 5C 2E  □ cF\.	628D8: 04 00 00 00 99 4C 53 0A  □ "LS□
628E0: 9D D0 2F 97 7A FF 02 72  □□/-zy□x	628E0: 86 D6 48 7D A3 48 4B BE  +0H)ΔHK34
628E8: C2 D9 55 17 72 62 00 00  &00□rb	628E8: 98 6C 4A A9 72 62 00 00  "1J@rb
628F0: 77 2B 62 00 F1 60 85 78  w+tb Ĥ`...x	628F0: 77 2B 62 00 46 49 4C 45  w+tb FILE
628F8: 00 00 00 00 19 6B E5 B8  □kĤ,	628F8: 00 00 00 00 4A 42 30 31  JB01
62900: 00 00 00 00 34 EA 43 00  4êC	62900: 00 00 00 00 34 EA 43 00  4êC
62908: F1 DF 43 00 34 EA 43 00  ĤBC 4êC	62908: F1 DF 43 00 34 EA 43 00  ĤBC 4êC

Uncritical:

Check if myAutToExe correctly found the start of the script. (It's using the heuristic EndOf\_PE-ExeFile => Start of Script)

So in the log there should be:

---> ScriptStartOffset: 00064E00

... else enter '64E00' in the Textbox (for the start offset of the script) manually

Click on yes if myAutToExe complains about invalid File marker.

(This JB01 thing don't comes into games since the script don't gets compressed)