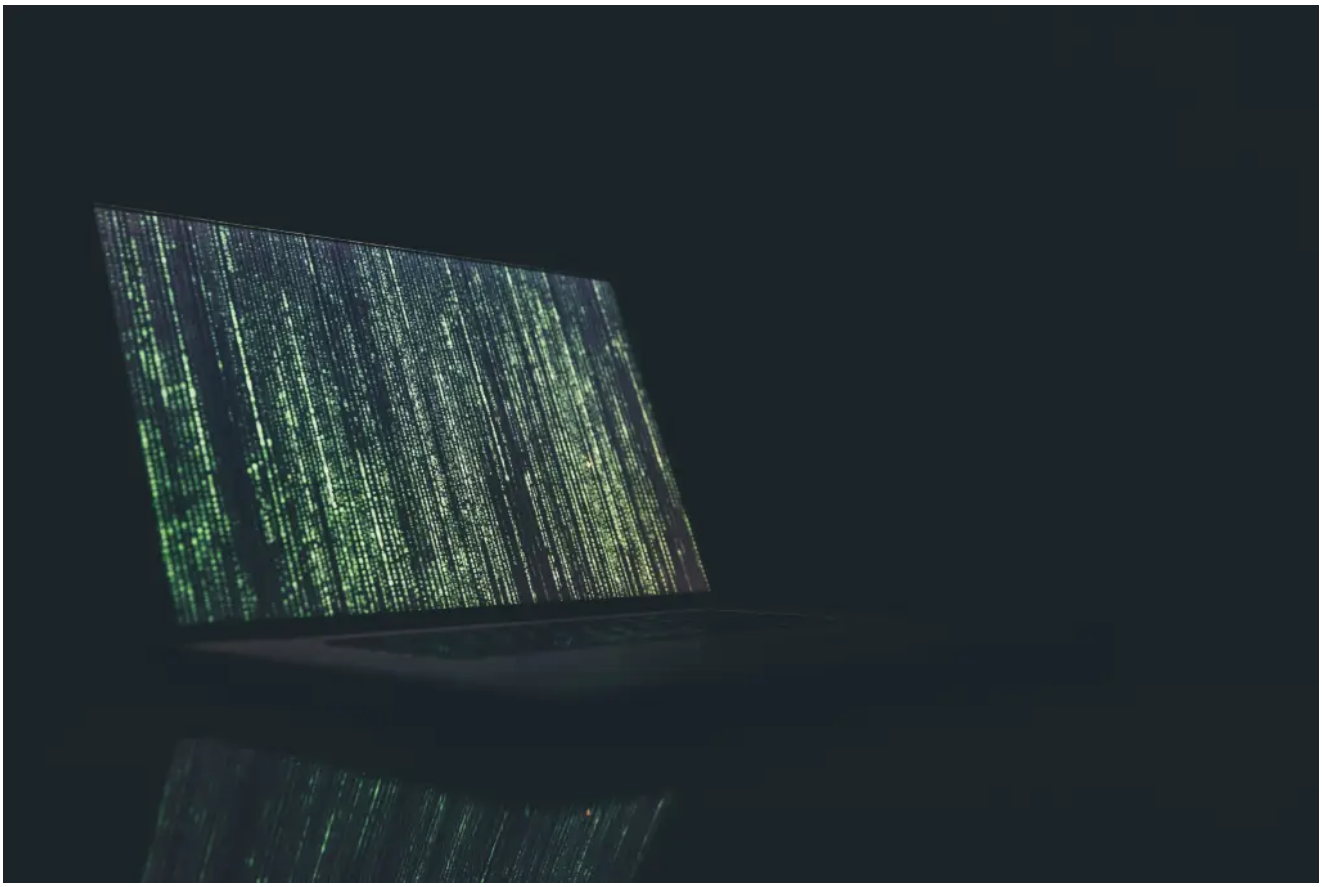
A “GULP” of PlugX – Cyber&Ramen

cyberandramen.net/2022/01/06/a-gulp-of-plugx/

January 6, 2022



Often attributed to Chinese-speaking threat actors, PlugX a remote access trojan(RAT), was identified by security researchers in 2012. With several variants of the RAT identified by vendors over the year, many techniques used to compromise systems have remained the same.

While perusing public malware sandboxes for interesting new samples, I stumbled upon a Windows executable that at the time, had a VirusTotal score of 9 out of 68 anti-virus vendors.

As this sample was found via a sandbox, the delivery method is unknown, and will not be covered in this post.

Dropper

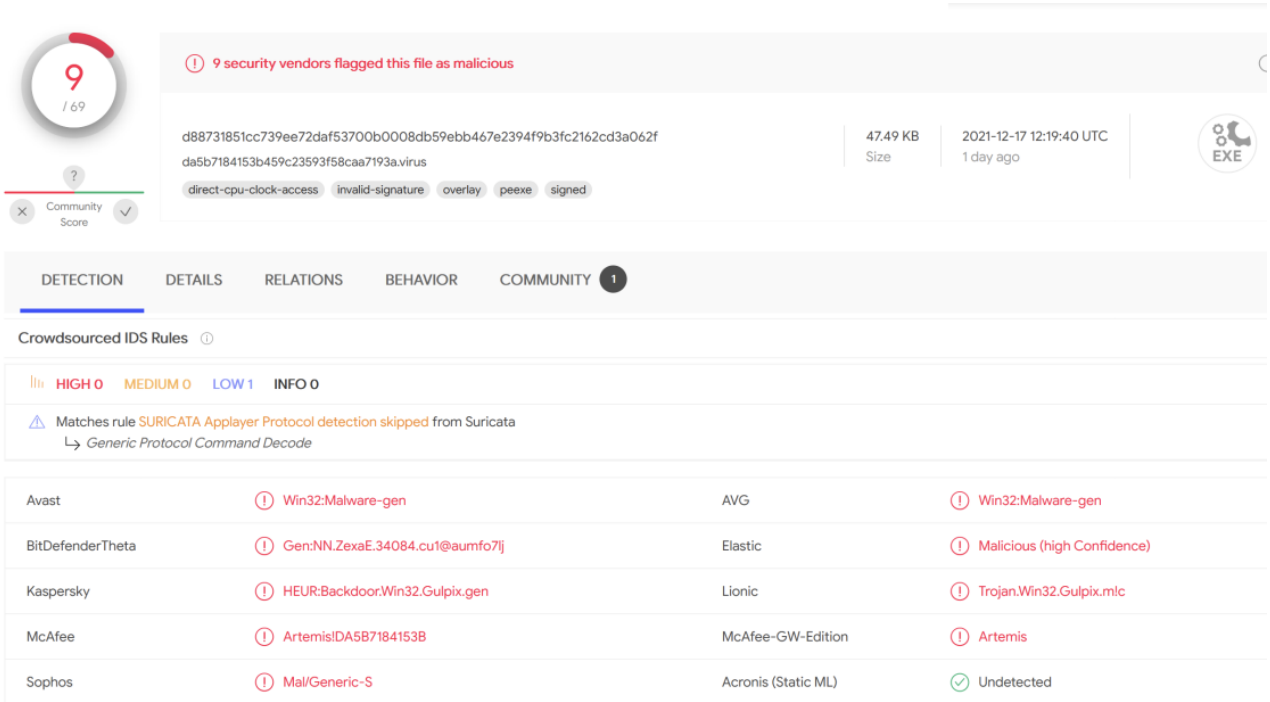


Figure 1
SHA256: d88731851cc739ee72daf53700b0008db59ebb467e2394f9b3fc2162cd3a062f

This sample was identified by VT user PerMorten as a dropper for the reflective loading of PlugX. Looking a little closer at the supposed dropper file, three additional files within the PE are identified:

- WinHelp32.exe
- rscom.dll
- rscom.dll.dat

```
call dword [GetWindowsDirectoryW]  ; 0x406020 ; UINT GetWindowsDirectoryW(LPWSTR lpBuffer, UINT u...
push edi
lea eax, [esi + 0x10ad10]
push ebx
push eax
call fcn.00401aab
push str.emp___scom.dll.dat        ; 0x409058
lea eax, [esi + 0x10ad10]
push ebx
push eax
call fcn.00401aab
push edi
lea eax, [esi + 0x10af18]
push ebx
push eax
call fcn.00401aab
push str.emp__WinHelp32.exe        ; 0x409030
lea eax, [esi + 0x10af18]
push ebx
push eax
call fcn.00401aab
push edi
push ebx
lea edi, [esi + 0x10b120]
push edi
call fcn.00401aab
push str.emp___scom.dll            ; 0x409010
```

Figure 2

WinHelp32.exe is a legitimate software application that will be described further below. For PlugX aficionados, the above trio of documents likely looks familiar. A well-known technique of PlugX is to utilize a dropper or self-extracting RAR PE file to extract files on the victim system for execution.

The Legitimate App

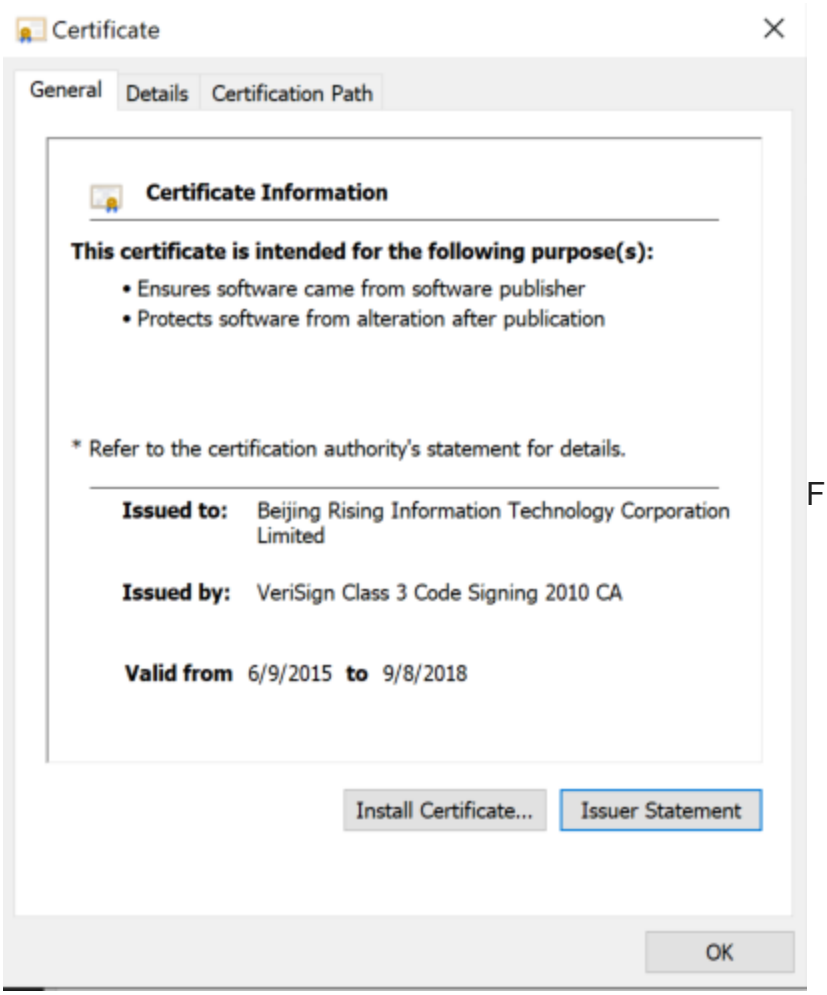


Figure 3

SHA256: ec200f75e4884933a56e82531f3f52e64e73a3347ad4a3b9e6318df82cdca92a

Winhelp32.exe is a legitimate application from the Beijing Rising IT company, a Chinese software company that develops Rising Antivirus among other computer security software.

As the network infrastructure utilized with this malware was only recently registered as of November 2021, the reasoning for using an outdated application is unknown. The threat actor, in this case, may have purposefully utilized a Rising Antivirus executable in the targeting of the intended victim or picked a random executable for their malware.

Rscom.dll.dat

The rscom.dll file does not contain much to write about other than its main purpose is to load the .dat file, which is the compressed/encoded PlugX payload.

As the payload is what everyone is here for, let's dive a bit deeper into the data file.

The well-known magic "GULP" is visible in the .dat file through a hex editor. Additionally, within the file, MZ and PE headers are also visible.

```
0002CCB0  89 46 14 8B 45 1C 89 46 18 C7 06 47 55 4C 50 8B  ‰F.‹E.‰F.Ç.GULP‹
0002CCC0  47 28 03 C6 89 46 1C 0F B7 47 14 33 C9 8D 44 38  G(.Æ‰F..·G.3É.D8
0002CCD0  18 66 3B 4F 06 73 27 8D 58 14 8B 03 03 45 FC FF  .f;O.s'.X.‹..Eüÿ
0002CCE0  73 FC 50 8B 43 F8 03 C6 50 FF 55 B4 0F B7 47 06  süP‹Cø.ÆPÿU´.·G.
0002CCF0  83 C4 0C FF 45 10 83 C3 28 39 45 10 7C DC 8B 87  ƒÄ.ÿE.ƒÃ(9E.|Ü‹‡
0002CD00  A0 00 00 00 85 C0 0F 84 9C 00 00 00 83 BF A4 00   ....…À.„œ...ƒ¿¤.
0002CD10  00 00 00 0F 84 8F 00 00 00 03 C6 EB 7D 83 65 10  ....„.....Æë}ƒe.
```

Figure 4

The .dat file is likely padded/compressed to evade antivirus engines. Upon execution, the file is decompressed via a call to the Windows API, RtlDecompressBuffer, and run in memory.

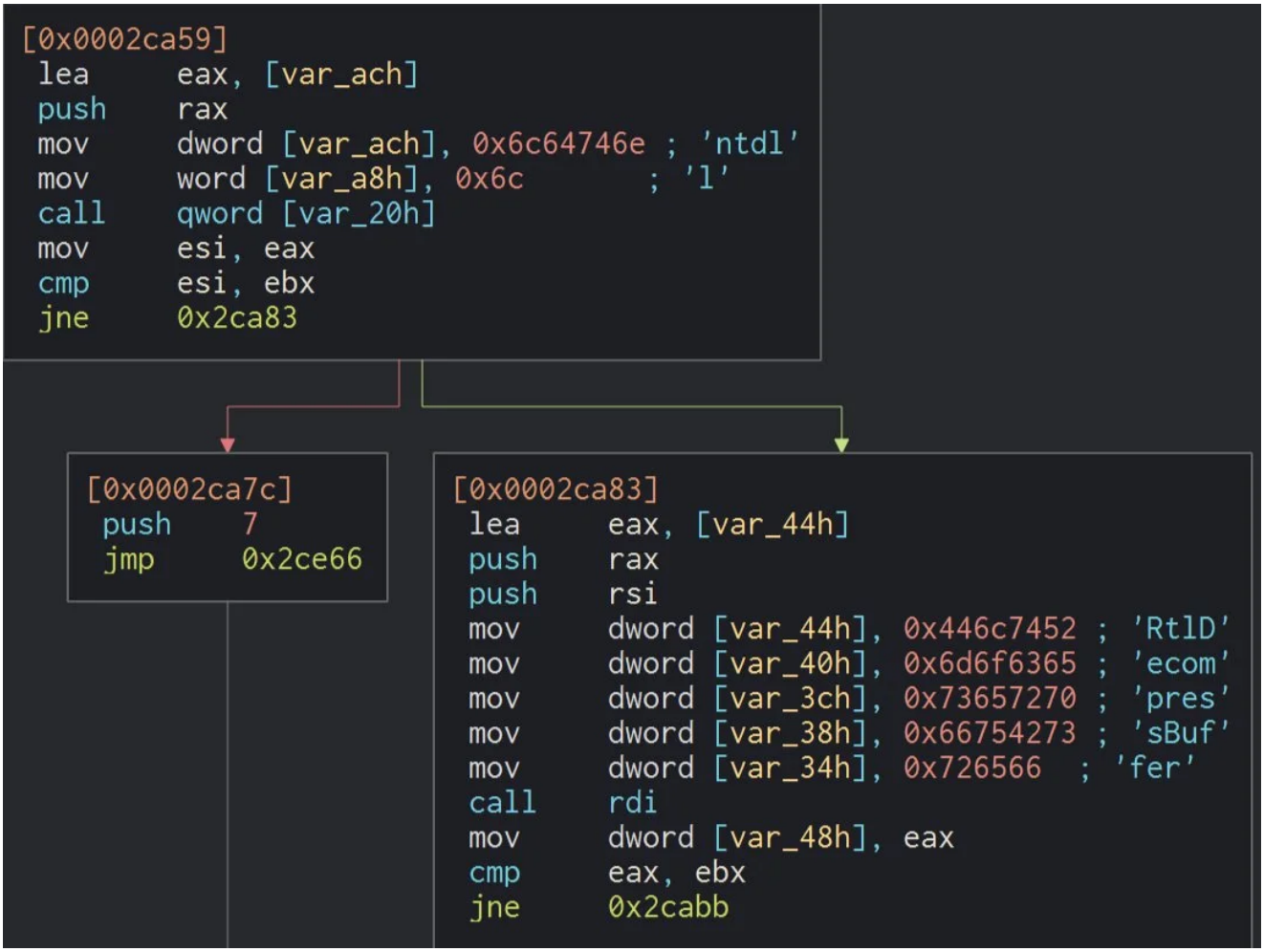


Figure 5

Identified in a number of reports on network intrusions involving PlugX, a familiar decryption routine (Figure 6) is also seen in rscom.dll.dat. The decryption routine contains multiple keys and shift operations, identified by the shr and shl calls below.

```
[0x0002cb02]
 mov     edx, ecx
 shr     edx, 3
 lea     ecx, [rcx + rdx - 0x11111111]
 mov     edx, eax
 shr     edx, 5
 lea     edx, [rax + rdx - 0x22222222]
 mov     eax, dword [var_8h]
 shl     eax, 7
 mov     ebx, 0x33333333             ; '3333'
 sub     ebx, eax
 add     dword [var_8h], ebx
 mov     eax, dword [var_4h]
 shl     eax, 9
 mov     ebx, 0x44444444             ; 'DDDD'
 sub     ebx, eax
 add     dword [var_4h], ebx
 lea     ebx, [rdx + rcx]
 add     bl, byte [var_8h]
 lea     eax, [rbp + rsi - 0x100]
 add     bl, byte [var_4h]
 mov     dword [var_14h], edx
 mov     edx, dword [var_ch]
 xor     bl, byte [rdx + rax]
 mov     byte [rax], r11b
 cmp     esi, 0x10
 jb      0x2caff
```

Figure 6

Malware Flow

The unnamed dropper places the three files into “C:\\ProgramData\Log’” in addition to a file named NvSmart.hlp (Figure 7). Upon running WinHelp32, the application deletes itself which is another interesting choice by the threat actor, as this would likely raise suspicions by the victim running the antivirus software.

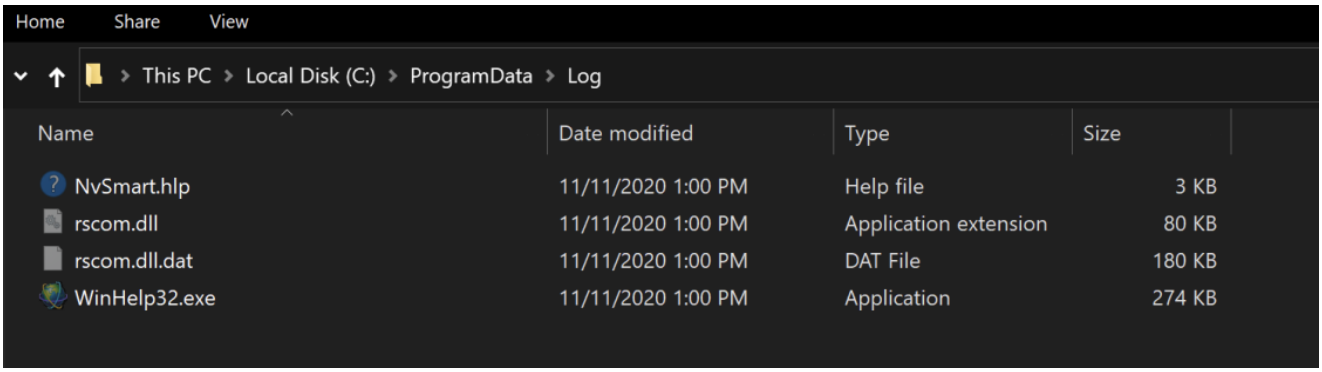


Figure 7
Watching the execution flow in your favorite Windows process monitoring software, old-school PlugX is in full effect. WinHelp32.exe injects itself into svchost.exe, with the usual second injected process, msiexec.exe not being seen in this case.

In most cases, if services.exe is not the process launching svchost.exe, this would be an easy win for defenders to detect. It is likely the threat actor is relying on the behavior of antivirus software injecting itself into a process that would not raise alarms.

Taking a look at the injected process read, write, executable (RWX) properties, we once again see that the MZ and PE headers have been replaced with GULP, or PLUG backward.

Figure 8
A number of hardcoded values including command and control (C2) information are located within the decoded configuration:

```
00042160  FF FF FF FF 01 00 39 30 78 69 67 75 61 6D 6F 6D  ÿÿÿÿ..90xiguamom
00042170  6F 6D 6F 2E 63 6F 6D 00 00 00 00 00 00 00 00 00  omo.com.........
00042180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000421A0  00 00 00 00 00 00 00 00 01 00 39 30 31 32 37 2E  ..........90127.
000421B0  30 2E 30 2E 31 00 00 00 00 00 00 00 00 00 00 00  0.0.1...........
000421C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000421D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000421E0  00 00 00 00 00 00 00 00 00 00 00 00 01 00 39 30  ..............90
000421F0  31 32 37 2E 30 2E 30 2E 31 00 00 00 00 00 00 00  127.0.0.1.......
00042200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042210  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042230  01 00 39 30 31 32 37 2E 30 2E 30 2E 31 00 00 00  ..90127.0.0.1...
00042240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042270  00 00 00 00 48 54 54 50 3A 2F 2F 00 00 00 00 00  ....HTTP://.....
00042280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042290  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000422F0  00 00 00 00 48 54 54 50 3A 2F 2F 00 00 00 00 00  ....HTTP://.....
00042300  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042310  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042320  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042330  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042340  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042350  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042370  00 00 00 00 48 54 54 50 3A 2F 2F 00 00 00 00 00  ....HTTP://.....
00042380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00042390  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000423F0  00 00 00 00 48 54 54 50 3A 2F 2F 00 00 00 00 00  ....HTTP://.....
```

Figure 9

Upon further research, an additional network indicator is located that appears to be a proxy for the C2.

```
00026E50  50 72 6F 78 79 2D 41 75 74 68 3A 20 00 00 00 00  Proxy-Auth: ....
00026E60  34 33 2E 31 32 39 2E 32 30 38 2E 32 32 36 00 00  43.129.208.226..
00026E70  34 33 2E 31 32 39 2E 32 30 38 2E 32 32 36 00 00  43.129.208.226..
00026E80  34 33 2E 31 32 39 2E 32 30 38 2E 32 32 36 00 00  43.129.208.226..
00026E90  34 33 2E 31 32 39 2E 32 30 38 2E 32 32 36 00 00  43.129.208.226..
```

Figure 10

So far we know the following about the network capabilities of this malware sample:

- A C2 domain of xiguamomomo[.]com
- Utilizes HTTP
- Communicates with a proxy server of 43.129.208[.]226

References to localhost, 127.0.0.1 can be seen in Figure 9, but the malware also seems to utilize the address for debug or anti-analysis purposes. This technique could possibly be utilized to slow researchers who may not be running the malware as needed for proper execution (running only the DLL file for example).

```
Protocol:[ TCP], Host: [127.0.0.1:12345], Proxy: [0::0::]
```

Figure 11

In addition to the possible debug strings seen in Figure 11, some 28 .cpp files indicating additional capabilities of the RAT were also found:

- XJoin.cpp
- XThreadManager.cpp
- XSoUdp.cpp
- XSoTcpHttp.cpp
- XSoTcp.cpp
- XSoPipe.cpp
- XSniffer.cpp
- XSetting.cpp
- XSessionImpersonate.cpp
- XPlugTelnet.cpp
- XPlugSQL.cpp
- XPlugShell.cpp
- XPlugService.cpp
- XPlugScreen.cpp
- XPlugRegEdit.cpp
- XPlugProcess.cpp
- XPlugPortMap.cpp
- XPlugOption.cpp
- XPlugNetstat.cpp
- XPlugNetHood.cpp
- XPlugKeyLogger.cpp
- XPlugDisk.cpp
- XPlugLoader.cpp
- XPacket.cpp
- XOnline.cpp
- XInstall.cpp
- XDList.cpp

- XBuffer.cpp

The following interesting PDB paths were also found:

```
Line 8058: 0x373beb4 (48): d:\work\plugx(32)\shellcode\shellcode\XSetting.h
Line 8067: 0x373bfe0 (45): d:\work\plugx(32)\shellcode\shellcode\XPlug.h
Line 8484: 0x37485d8 (43): D:\WORK\PLug 1.0\Plug\Release\ByPassUAC.pdb
```

Figure 12

Network Indicators

According to PassiveDNS information, the domain xiguamomomo[.]com resolves to 111.73.46[.]103, located in China, first seen 2021-10-12.

WHOIS information reveals the domain was registered through GoDaddy, with the registrant country listed as Cambodia, and the registrant identified as “ewrwer.”

In what could certainly be a coincidence, both xigua, and momo are popular apps originating from China. Xigua, an online video-sharing app with users across the world, boasts some 160 million users. Momo, currently only available in Chinese, is a social networking app with a large following.

It should be noted that not only are the delivery method of the RAT unknown, but also the targeting. The above should be taken as low confidence at best, but certainly interesting nonetheless.

An additional IP address of 111.73.46[.]30 (open ports: 3389, 8000, 5985, 5987, and 24681) was also identified through packet captures.

The ports 3389 (RDP), and 5985 are largely seen among many other suspected PlugX C2 infrastructure. This IP address belongs to the Chinanet-Backbone ASN.

The possible proxy address 43.129.208[.]226 (open ports: 22, 3306, and 8443) is located in Hong Kong and belongs to the TENCENT-NET-AP-CN ASN.

Multiple User-Agent values were also found within the decoded configuration data as seen in Figure 13.

```
0x6b3c46f0 (50): Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
0x6b3c4754 (41): Mozilla/4.0 (compatible; MSIE 8.0; Win32)
0x6b3c4780 (50): Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
0x6b3c47b8 (101): Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.16 Safari/537.36
0x6b3c4820 (63): Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
0x6b3c4860 (65): Mozilla/5.0 (Windows NT 6.2; rv:12.0) Gecko/20100101 Firefox/12.0
0x6b3c48a8 (99): Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5
0x6b3c4910 (94): Mozilla/5.0 (compatible; MSIE 10.0; Windows Phone 8.0; Trident/6.0; IEMobile/10.0; ARM; Touch)
0x6b3c4970 (64): Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)
0x6b3c49b8 (69): Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; Xbox)
0x6b3c4a00 (63): Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
0x6b3c4a40 (71): Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
0x6b3c4a88 (83): Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0)
0x6b3c4ae8 (125): Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
```

Figure 13

**Featured image: Photo by Markus Spiske on Unsplash

Conclusion

As there is quite a bit of information missing with this variant of PlugX, the fresh command and control infrastructure and domain naming indicate that even dated versions of this RAT still get the job done.

Please keep an eye out for updates to this post as I look deeper into the network infrastructure to possibly tie additional domains/malware to the above findings.

Indicators

Files:

- Dropper file: d88731851cc739ee72daf53700b0008db59ebb467e2394f9b3fc2162cd3a062f
- WinHelp32.exe (legitimate application): ec200f75e4884933a56e82531f3f52e64e73a3347ad4a3b9e6318df82cdca92a
- Rscom.dll (loader) : 7af30d3c192f3fb85e1cadbf5c01f049f11eb036ca8107abb3451ffa0cc218b7
- Rscom.dll.dat (PlugX payload): ec46e04df901d7ec76ff1ad9ad6ceb54f8c2ad5e3597173365e094c5602e0049

Network:

- xiguamomomo[.]com >> 111.73.46[.]103
- 111.73.46[.]30
- 43.129.208[.]226 (proxy)
- “/update?id=” (Callback URI in config)
- Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
- Mozilla/4.0 (compatible; MSIE 8.0; Win32)
- Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
- Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.16 Safari/537.36
- Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
- Mozilla/5.0 (Windows NT 6.2; rv:12.0) Gecko/20100101 Firefox/12.0
- Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5
- Mozilla/5.0 (compatible; MSIE 10.0; Windows Phone 8.0; Trident/6.0; IEMobile/10.0; ARM; Touch)
- Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)
- Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; Xbox)
- Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
- Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
- Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0)

- Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3