Threat Thursday: China-Based APT Plays Auto-Updater Card to Deliver WinDealer Malware

blogs.blackberry.com/en/2022/06/threat-thursday-china-based-apt-plays-auto-updater-card-to-deliver-windealer-malware

The BlackBerry Research & Intelligence Team

1. BlackBerry Blog
2. Threat Thursday: China-Based APT Plays Auto-Updater Card to Deliver WinDealer Malware

Auto-updaters can help close the door on known attacks and security vulnerabilities by allowing software vendors to patch their own wares automatically. But recently, threat actors from the LuoYu group have been turning this commonly used feature into a weapon to compromise victims' machines.

These attackers have been using a technique mentioned in documents leaked by famed NSA defector Edward Snowden, called Quantum Insert or Man-on-the-Side (MotS) attacks. These are used in conjunction with watering-hole attacks on popular Android messaging apps to deploy malware, including WinDealer.

Attacks such as the ones we'll describe in this post have recently been observed in East Asia, but the threat group LuoYu has broadened its focus to hit Chinese-speaking targets in other countries as well, with the apparent motive of stealing dissident information.

Operating System

Windows	MacOS	Linux	Android
Yes	No	No	No

Risk & Impact

Impact	High
Risk	Low

LuoYu Threat Group Background

LuoYu is a sophisticated APT group linked to China, which has been active since 2008. First discovered and christened by researchers at teamT5, LuoYu's malicious activities and campaigns have been primarily confined to Eastern Asia. The group's previous targets have included Hong Kong, Japan, Korea, Taiwan, and China.

According to teamT5, China has been the group's primary theater of operation, with the apparent goal of targeting Chinese dissidents through messaging applications. This group has also attacked Chinese-speaking victims in the United States, Russia, and Europe. LuoYu initially focused its attacks on the technology, media, and education sectors of China and Eastern Asia, but has since branched out to other industries including financial, government, military, telecommunications, and logistics sectors.

In 2019, Fortinet highlighted a new campaign against a U.S.-hosted Chinese-language news site by a (then-unnamed) threat group. The backdoor application used in that campaign was named "PPTV(pplive)_forap_1084_9333.exe." This exact same file, combined with the same filename and hosting, was also present in recent research published by Kaspersky, which identifies the threat as WinDealer and names LuoYu as the threat actor responsible for delivering the malware. The presence of the same file, filename, hosting, and the similar nature of the target likely indicates that LuoYu was behind the earlier Chinese news site attack.

LuoYu currently deploys several different types of malware via a number of different methods. The group also deploys Demsty and SpyDealer, which are information stealers, by using watering-hole attacks. A watering-hole attack targets a specific group of users by infecting websites that members of the group are known to visit, or which the threat group has observed them visiting by covert means. For example, it's an easy guess that employees of any company are likely to visit their own organization's website most often, and periodically those of their competitors. These websites are then compromised via zero-day vulnerabilities on browsers or other software.

LuoYu's watering hole attacks use compromised news sites and malicious versions of popular Android messaging applications like WeChat, WhatsApp, QQ, and Weibo. These tactics were employed in 2019, exploiting interest in protests over the Hong Kong anti-extradition bill to steal information from suspected dissidents.

Other researchers have been dutifully following LuoYu's use of the MotS technique to deliver WinDealer malware. Our purpose in this post is to analyze the latest iteration of WinDealer malware, and its notable addition of a new infection vector.

New Attack Vector

LuoYu can now target the automatic update functionality of legitimate applications, using MotS attack techniques. Unlike a Man-in-the-Middle attack (MitM) attack, where an attacker is positioned as a node between the client connection and its destination, MotS gives an attacker direct control of the response from both client and server, allowing the threat actor to directly modify the information requested or sent, and to delete messages or deny responses entirely. The difference between the two attack methods lies in how the attacker controls the flow of information.

In a MotS attack, the attacker only has access to see and read the communication channel, but no ability to modify or deletepackets sent between server and client. The goal of this technique is to send messages in the communication channel before a legitimate response from the server is returned. The attacker must get the timing exactly right to beat the legitimate server response.

This is where auto-updaters come in. Auto-updaters tend to make a lot of requests to servers, to check if there are any new updates to their host's software. These types of applications would provide an attacker with many opportunities to win "the race" in getting compromised messages back to the client. This could explain why automatic updaters are a popular target.

A MotS attack is also known as a Quantum Insert. This term was used in documentation from the NSA leaked by Edward Snowden in 2014. This attack technique has been in use since 2005.

Technical Analysis

WinDealer requires two portable executable (PE) files to work properly, both of which are carried within the host executable:

- **Main Executable** (SHA256: 08530e8280a93b8a1d51c20647e6be73795ef161e3b16e22e5e23d88ead4e226)
  This does the initial configuration and decrypts the embedded library.
- **Second Executable** (SHA256: aa7a43b30025e849a9912c873fb0225e8354fc895533fd8136af8379902aea27)
  This file is masked as a bitmap (BMP) image in the resource section, and loaded into memory on the system.

Both files are used to execute WinDealer's capabilities, which include the following activities:

- File manipulation
- Registry manipulation
- Taking screenshots
- Process enumeration
- Collecting host environment information
- Data exfiltration
- Keylogging
- Command-and-control (C2) backdoor creation

The main executable is an unpacked, 32-bit Microsoft Visual C++ compiled application. The file itself masquerades as Microsoft's Internet Explorer, including having the trademarked blue lowercase "e" icon and golden ring halo.

Copyright information in the file description lists the copyright as belonging to "Mozille Developers." The file itself is not signed. The original filename is listed as "explorer.exe," as shown in Figure 1.

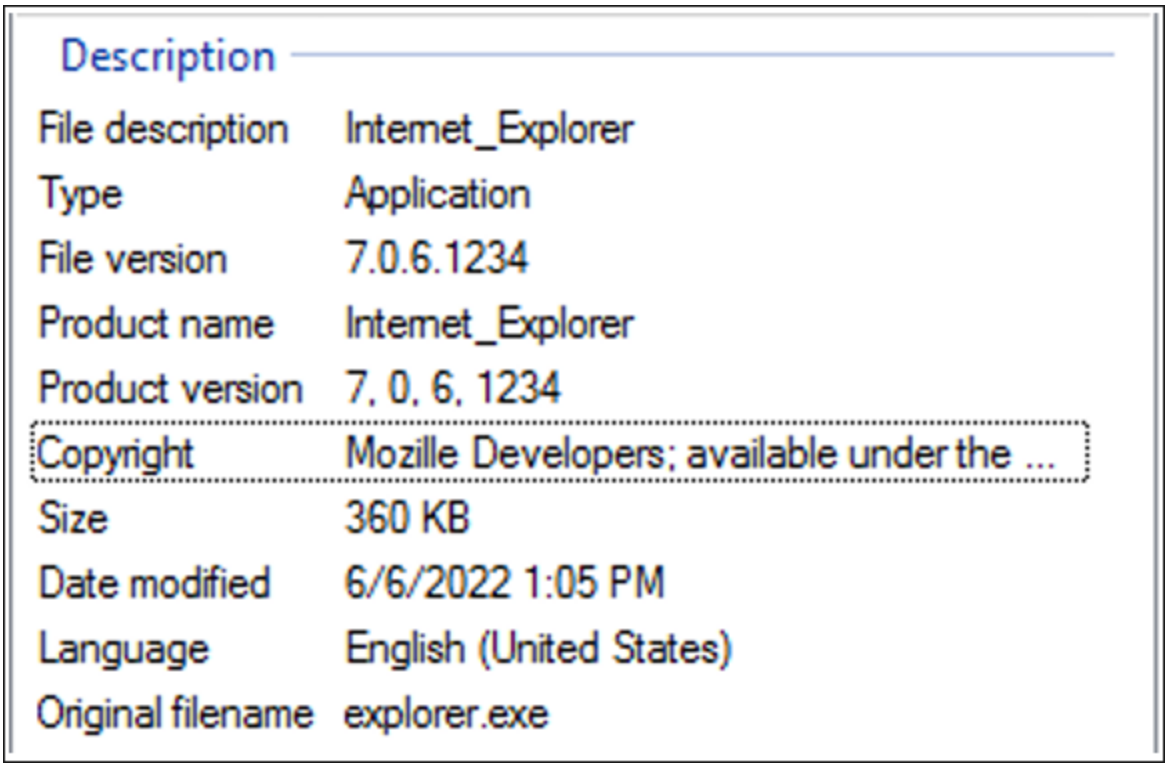


Figure 1 - Details for the main executable (SHA256: 08530e8280a93b8a1d51c20647e6be73795ef161e3b16e22e5e23d88ead4e226)

WinDealer begins its execution by querying the user profile of the victim's machine with the getenv() command. If an environment is found, a hash value is created to uniquely identify the victim.

This identifier is then used with CreateMutexA as both a named mutex object and a shared resource. A mutex is a synchronization object that prevents multiple threads from accessing an object at the same time. This is to prevent race conditions such as two threads trying to change data at the same time.

In addition to the mutex, a unique identifier is also stored in the registry under "\CurrentVersion\Internet Settings\5.0\User Agent\." This key is used to identify the browser to the server host site being accessed.

The hash value used to identify the victim is in MD5 format. It combines the MAC address, physical drive info, and the username of the system. Creation of this identifier is shown as a snippet from the decompiler in Figure 2, below.

WinDealer achieves persistence by editing the "\SOFTWARE\Microsoft\CurrentVersion\Run\" key on the host's machine. This registry key is loaded when the user logs on to the system and executes the application "firebrow.exe."

This malware also contains a hard-coded string that indicates which version of Windealer it is. The hash in the sample used in this particular analysis is version “18.20.1225.” The last two numbers represent the creation date; in this case, December 25. The middle number is the two-digit year, 2020. We believe the first number is the iteration of this version of WinDealer.

```
mutex_test_object = CreateMutexA((LPSECURITY_ATTRIBUTES)0x0,0,identifier);
if (mutex_test_object != (HANDLE)0x0) {
  last_error_code = GetLastError();
  if (last_error_code != 0xb7) {
    Windows_version_Internet_Drive_info();
```

Figure 2 - Mutex creation and registry key function

Once the mutex and persistence mechanisms are set, WinDealer creates three files in the “C:\ProgramData\” folder. These files are named 1c76cbfe, 923b5fd7, and f46d373b. The files are used by WinDealer for its C2 configuration. Configuration settings include RemoteIP, RemoteDomain, Password, and a Remark variable for the configuration, which are shown below in figure 3. Configuration settings received from the server are stored in these files.

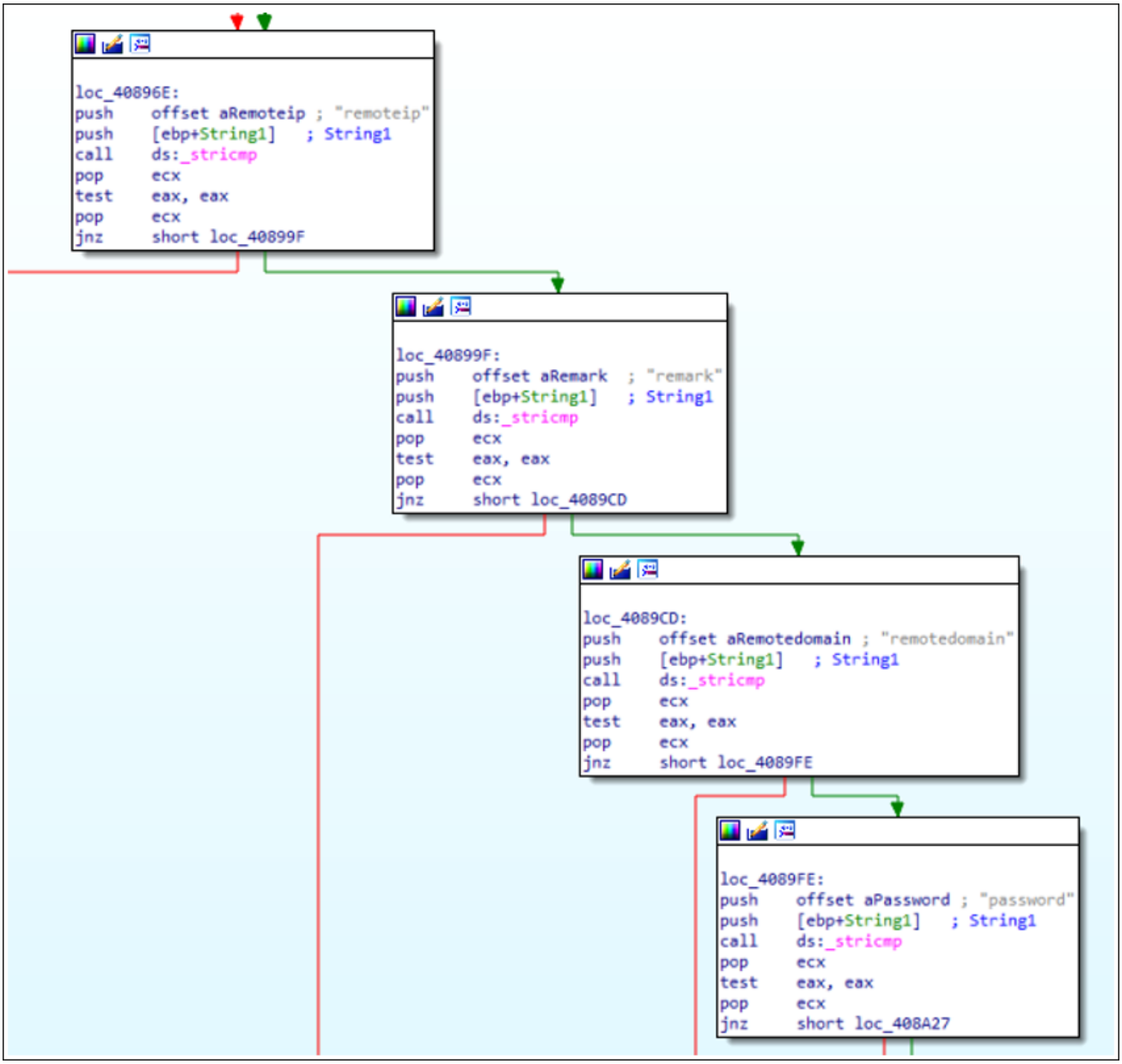


Figure 3 - RemoteIP, Remark, RemoteDomain, and Password configuration

Next, three folders are created in the “C:\User\[hostname]\Appdata\Local\temp\” directory. The folders—named 8e98-fb8010fb, 070a-cf37dcf5, and 28e4-20a6acec—are used to exfiltrate information stolen from the target host.

Folder 8e98-fb8010fb has logs of the directory information for C:/, Program Files(x86), OneDrive, Boot, and the Recycle Bin, all stored as text files. Folder 070a-cf37dcf5 contains “.S” files that are composed of hex values. At this time, the purpose of the .S files is unclear.

Folder 28e4-20a6acec has files with the extension “.A.” These files are XORed to obfuscate the file contents, using the key “YYYY.” Using a bitwise XOR with the key produces the decrypted contents of the .A files. There are five .A files in this directory, each containing

specific data extracted from the host. Contents of the .A files include the running process tree, adapter information, key inputs, disk information, and local area network configuration. Figure 4 shows the decrypted output of the process tree file.

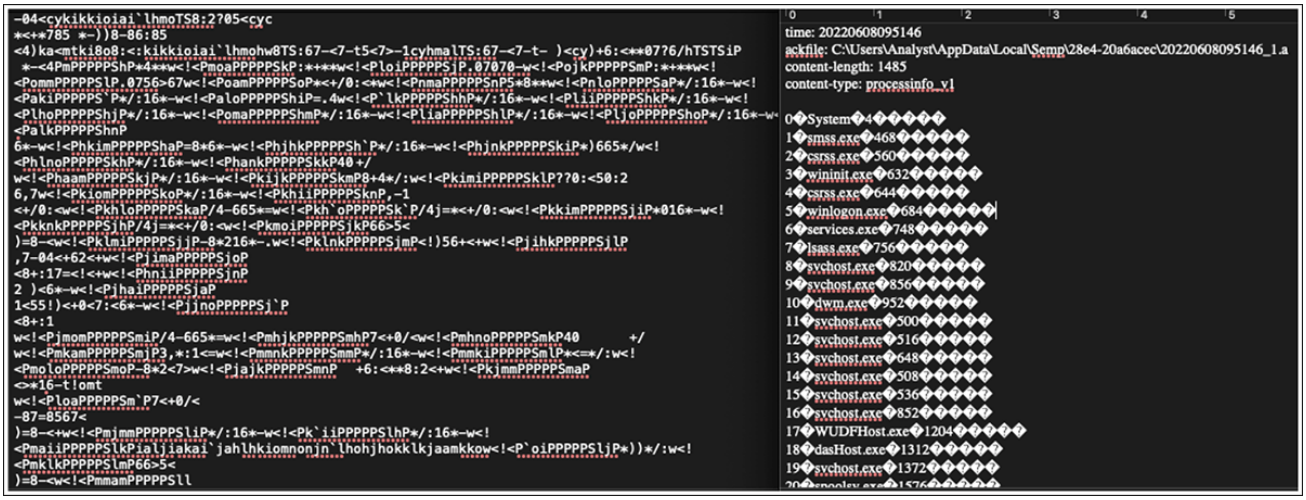
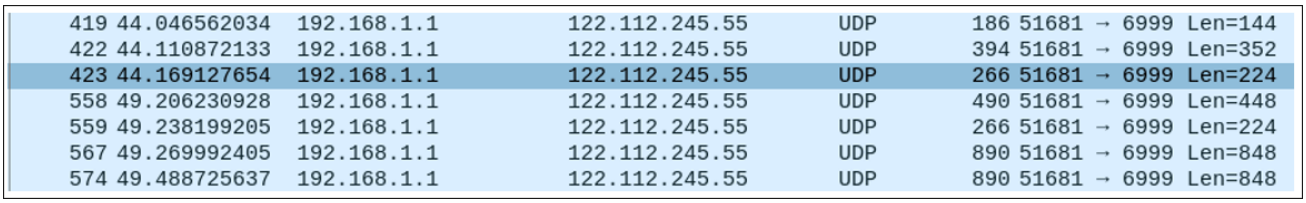


Figure 4 - Decrypted processInfo file with key "YYYY"

For the exfiltration of this data to the server, WinDealer uses rand() to generate a random IP for communication. Shown in Figure 5, the function works in the range of 113[.]62[.]0[.]0 – 113[.]63[.]255[.]255 and 111[.]120[.]0[.]0 – 111[.]123[.]255[.]255 respectively. This IP address is generated when no existing configuration is available.

```
if (DAT_0041bd1c == 0) {
  iVar2 = 62;
  do {
    if (local_104[iVar2] == 0) break;
    iVar2 = iVar2 + 1;
  } while (iVar2 < 255);
  random_number_2 = rand();
  DAT_0041bd1c = 1;
  random_number = random_number + 113 + (random_number_2 % 2 + iVar2) * 256;
}
else if (DAT_0041bd1c == 1) {
  iVar2 = 120;
  do {
    if (local_104[iVar2] == 0) break;
    iVar2 = iVar2 + 1;
  } while (iVar2 < 255);
  random_number_2 = rand();
  random_number = random_number + 111 + (random_number_2 % 4 + iVar2) * 256;
  DAT_0041bd1c = 0;
}
```
Figure 5 - IP snippet of address generator function

In addition to the IP generation, the sample we used for this analysis also attempted connections with non-existent URLs www[.]microsoftcom and icanhazip[.]com. Included with those DNS queries, there is an IP connection on port 80 via TCP from one of two addresses: 104[.]18[.]115[.]97, or 104[.]18[.]114[.]97. There is also a connection to a Chinese-hosted ISP over port 6999 at 122[.]112[.]245[.]55. Traffic from this communication is shown in Figure 6.



419	44.046562034	192.168.1.1	122.112.245.55	UDP	186 51681 → 6999 Len=144
422	44.110872133	192.168.1.1	122.112.245.55	UDP	394 51681 → 6999 Len=352
423	44.169127654	192.168.1.1	122.112.245.55	UDP	266 51681 → 6999 Len=224
558	49.206230928	192.168.1.1	122.112.245.55	UDP	490 51681 → 6999 Len=448
559	49.238199205	192.168.1.1	122.112.245.55	UDP	266 51681 → 6999 Len=224
567	49.269992405	192.168.1.1	122.112.245.55	UDP	890 51681 → 6999 Len=848
574	49.488725637	192.168.1.1	122.112.245.55	UDP	890 51681 → 6999 Len=848

Figure 6 – UDP traffic over 6999 to a Chinese-hosted ISP

After the IP has been generated and the connection is established, WinDealer proceeds to exfiltrate data from the host.

The threat also edits the registry value “\Microsoft\Windows\CurrentVersion\Internet Settings” ProxyEnable and ProxyServer. Both these keys are used to facilitate a proxy connection from a Windows device.

After it has completed this action, the malware begins to extract the victim’s information that has been stored in the files created in AppData\Local\Temp. The contents of the .A files are XORed, packeted, and sent to the IP address configured by the executable. Figure 7 shows a list of functions facilitating some of the information extraction, and the target information.

```
extract_vic_info                                XREF[1]:     main:00403022(c)
    PUSH        ESI
    MOV         ESI,param_1
    CALL        Get_Computer_Name                              undefined4 Get_

    MOV         param_1,ESI
    CALL        userProfile_env                                undefined4 user

    MOV         param_1,ESI
    CALL        CPU_Type                                       undefined4 CPU_

    MOV         param_1,ESI
    CALL        kernel32_OS_info                               undefined4 kern

    MOV         param_1,ESI
    CALL        net_adapter_and_MacAddr                        undefined4 net_
```

Figure 7 - GetComputerName, UserProfile, CPU, Kernel32, and network-related function calls

The malicious application comes embedded with a DLL that uses a steganography technique to avoid detection. Steganography is a method of hiding data within another message or object. In this case, the object is a bitmap image stored in the resource section of the executable.

The bitmap image, which is the embedded DLL, is XOR'd with a 10-byte key. Once decrypted, a call to GetProcAddress, LoadLibraryA, and Virtuallalloc loads the library into memory. The strings for the library loads are built as character arrays and represented in hexadecimal to obscure the calls. Figure 8 shows the bitmap .RSRC section, a snapshot of the decompiler call to the function that XORs the library, and a snippet of the function that XORs the library.

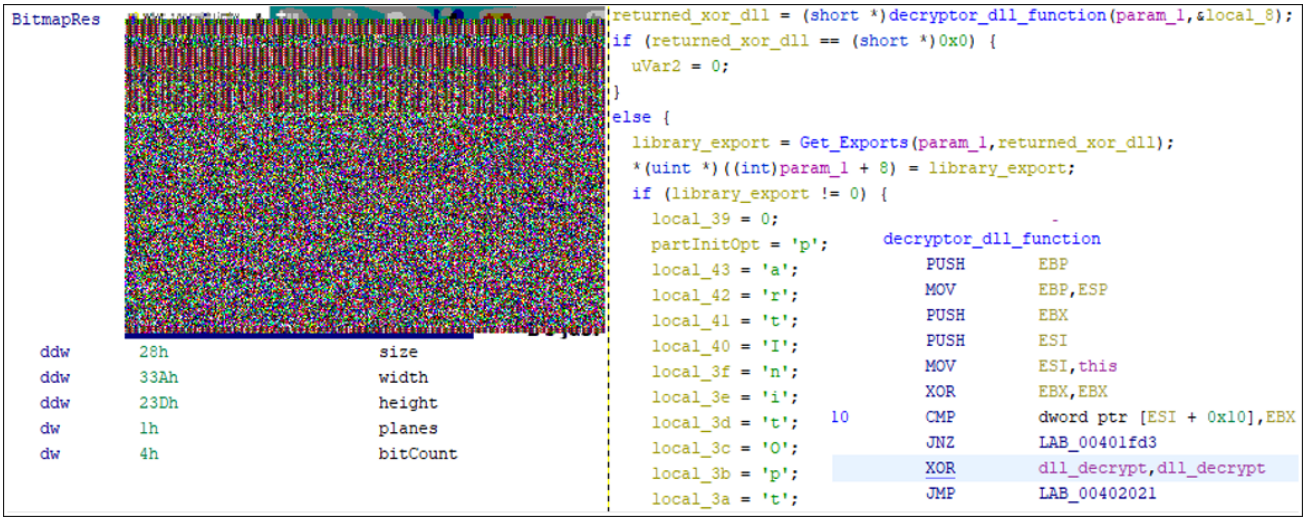


Figure 8 - Embedded DLL bitmap and XOR-related functions

WinDealer uses the executable and DLL in combination to contact its C2 server and exfiltrate data. The loaded library is used to steal data and configure access to the host, as well as handling thread creation to exfiltrate data from the victim.

The decrypted DLL (SHA256: aa7a43b30025e849a9912c873fb0225e8354fc895533fd8136af8379902aea27) is named "MozillaDll.dll" and has three exports: AutoGetSystemInfo, GetConfigInfo, and partInitOpt.

The AutoGetSytemInfo export facilitates the extraction of information from the victim. This function searches the victim's system for the presence of registry keys and configuration files that are used by popular chat applications. Targeted applications are WeChat, WangWang, and Microsoft's Skype messenger.

The directory information of the "My Documents" folder and other user-specific contents are written to the log files for exfiltration as well. Any information found in this folder is written to the appdata\Local\Temp directory and sent to the attacker's remote destination. A side-by-side display of disassembly and parent function calls from the decompiler of AutoGetSystemInfo are shown in Figure 9.

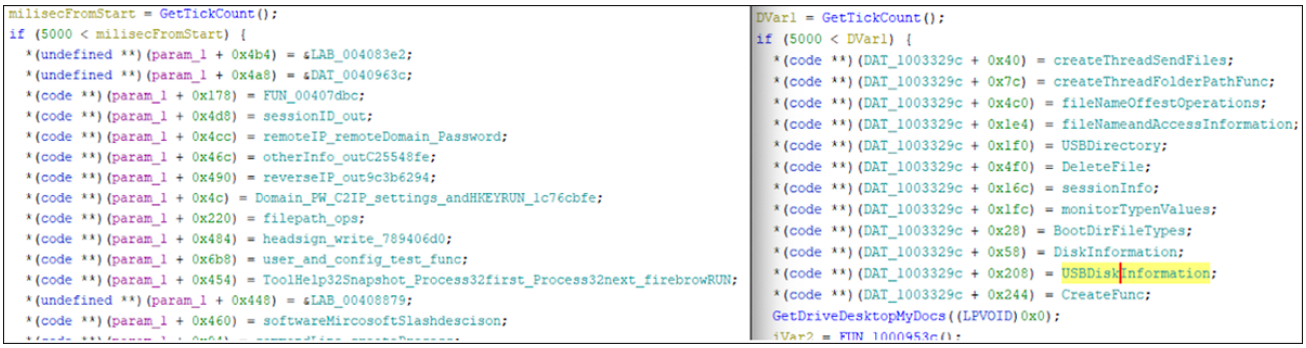
```
68 d8 e1        PUSH        s_FileSavePath_1002e1d8                 = "FileSavePath"
02 10
68 c0 e1        PUSH        s_Software\Tencent\WeChat_1002e1c0      = "Software\\Tencent\\WeChat"
02 10
57              PUSH        EDI
e8 48 28        CALL        openKeyandRead                          undefined4 openKeyandRead(HKEY
ff ff
83 c4 20        ADD         ESP,0x20
80 bd ec        CMP         byte ptr [EBP + local_118],0x0
fe ff ff 00
74 17           JZ          LAB_1000f62e
8d 85 ec        LEA         EAX=>local_118,[EBP + 0xfffffeec]
fe ff ff
68 b4 e1        PUSH        myDocuments                             = "MyDocument:"
```

```
undefined4 __fastcall stealChatInformation(void *param_1)

{
  AppData_LocalAppDataKeys((int)param_1);
  SkypeConfig(param_1);
  TencentConfig();
  WeChatConfig();
  WangWangConfig(param_1);
  WrapToCreateWriteFile(param_1);
  return 1;
}
```

Figure 9 - AutoGetSystemInfo snippet showing WeChat and MyDocuments targets (left) and parent function call (right)

The partInitOpt export contains embedded functions for the C2 aspect of the malware. The commands provide the executable with functionality to pull host configuration data to send to the attacker. The commands can change session, display any USB drives, monitor the session type, change filenames, and get disk information. This command set is split between the executable and dropped library, as shown in Figure 10.



```
milisecFromStart = GetTickCount();
if (5000 < milisecFromStart) {
  *(undefined **)(param_1 + 0x4b4) = &LAB_004083e2;
  *(undefined **)(param_1 + 0x4a8) = &DAT_0040963c;
  *(code **)(param_1 + 0x178) = FUN_00407dbc;
  *(code **)(param_1 + 0x4d8) = sessionID_out;
  *(code **)(param_1 + 0x4cc) = remoteIP_remoteDomain_Password;
  *(code **)(param_1 + 0x46c) = otherInfo_outC25548fe;
  *(code **)(param_1 + 0x490) = reverseIP_out9c3b6294;
  *(code **)(param_1 + 0x4c) = Domain_PW_C2IP_settings_andHKEYRUN_1c76cbfe;
  *(code **)(param_1 + 0x220) = filepath_ops;
  *(code **)(param_1 + 0x484) = headsign_write_789406d0;
  *(code **)(param_1 + 0x6b8) = user_and_config_test_func;
  *(code **)(param_1 + 0x454) = ToolHelp32Snapshot_Process32first_Process32next_firebrowRUN;
  *(undefined **)(param_1 + 0x448) = &LAB_00408879;
  *(code **)(param_1 + 0x460) = softwareMircosoftSlashdescison;
```

```
DVar1 = GetTickCount();
if (5000 < DVar1) {
  *(code **)(DAT_1003329c + 0x40) = createThreadSendFiles;
  *(code **)(DAT_1003329c + 0x7c) = createThreadFolderPathFunc;
  *(code **)(DAT_1003329c + 0x4c0) = fileNameOffestOperations;
  *(code **)(DAT_1003329c + 0x1e4) = fileNameandAccessInformation;
  *(code **)(DAT_1003329c + 0x1f0) = USBDirectory;
  *(code **)(DAT_1003329c + 0x4f0) = DeleteFile;
  *(code **)(DAT_1003329c + 0x16c) = sessionInfo;
  *(code **)(DAT_1003329c + 0x1fc) = monitorTypenValues;
  *(code **)(DAT_1003329c + 0x28) = BootDirFileTypes;
  *(code **)(DAT_1003329c + 0x58) = DiskInformation;
  *(code **)(DAT_1003329c + 0x208) = USBDiskInformation;
  *(code **)(DAT_1003329c + 0x244) = CreateFunc;
  GetDriveDesktopMyDocs((LPVOID)0x0);
```

Figure 10 - Command set for C2 functionality, split between the executable and library. The image on the left is from the executable "explorer.exe," and on the right is "MozillaDLL.dll"

The GetConfigInfo export is used to get the configuration for the malware. Its functionality consists of copying the string data set by the host executable, and setting the configuration based on the information in created config files. The settings can also be configured by an active connection through the C2 aspect of the malware.

Conclusion

WinDealer is Windows-based malware that splits functionality between two executables. One Trojan .EXE handles installation, network configurations, and houses a split section of the C2 commands. The encrypted DLL that is dropped upon execution handles configuration, plus the bulk of data extraction, as well as the other half of the C2 commands. Windealer XORs the stolen data before sending it over a TCP connection via port 6999.

The group behind this threat, LuoYu, uses a MotS attack to deliver WinDealer to any target for which they have access to the communication channel. By targeting auto-update processes, the group can persist until they succeed in exfiltrating potentially sensitive data from their target.

This type of attack could become more frequent in the future, as many legitimate applications now come with auto-update features enabled by default. This could be a tempting target for other sophisticated APT groups that have the patience and timing to execute this type of sophisticated and complex attack.

Who is Affected?

WinDealer is most commonly deployed to Chinese mainland targets, due to its association with the APT group LuoYu. Occasional infections in other countries, such as the United States, Russia, and India, are also possible. If an infection occurs outside of China, it is likely intended to target Chinese-speaking individuals. Affected sectors include tech, media, education, government, telecom and logistics.

Mitigation Tips

- Monitor IP traffic for a remote terminal session (*https://d3fend.mitre.org/technique/d3f:RemoteTerminalSessionDetection/*).
- Monitor IP traffic for extracted files (https://d3fend.mitre.org/technique/d3f:FileCarving/).
- Use a filter for traffic from the generated IP address WinDealer creates, and any attempted connections (*https://d3fend.mitre.org/technique/d3f:ConnectionAttemptAnalysis/*).

YARA Rule

The following YARA rule was authored by the BlackBerry Research & Intelligence Team to catch the threat described in this document:

```
rule Windealer_executable{
    meta:
        description = "Detects WinDealer Executable"
        author = "BlackBerry Threat Research Team"
        date = "2022-06-14"
        license = "This Yara rule is provided under the Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as long as you use it under this license and ensure originator credit in any derivative to The BlackBerry Research & Intelligence Team"

    strings:
        $s1 = "28e4-20a6acec"
        $s2 = "5a7e-42ccdb67"
        $s3 = "632c-0ef22957"
        $s4 = "63ae-a20cf808"
        $s5 = "65ce-731bffbb"
        $a1 = "remoteip"
```

```
        $a2 = "sessionid"
        $a3 = "remotedomain"
        $a4 = "remark"

    condition:
        uint16(0) == 0x5a4d and 2 of ($s*) and 1 of ($a*)
}
rule Windealer_Library{
    meta:
        description = "Detects WinDealer Loaded DLL"
        author = "BlackBerry Threat Research Team"
        date = "2022-06-14"
        license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"

    strings:
        $s1 = "C:\\Users\\Public\\Documents\\Tencent\\QQ\\UserDataInfo.ini"
        $s2 = "Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders"
        $s3 = "SOFTWARE\\SogouInput\\red"
        $s4 = "SOFTWARE\\SogouDesktopBar"
        $s5 = "MozillaDll.dll"
        $s6 = "Tencent Files"
        $s7 = "wangwang"
        $s8 = "WeChat Files"
        $s9 = "MyDocument"
        $s10 = "Skype"
        $e1 = "AutoGetSystemInfo"
        $e2 = "GetConfigInfo"
        $e3 = "partInitOpt"
    condition:
        uint16(0) == 0x5a4d and all of them
}
```

Indicators of Compromise (IoCs)

Explorer.exe -
08530e8280a93b8a1d51c20647e6be73795ef161e3b16e22e5e23d88ead4e226

MozillaDLL.dll -
aa7a43b30025e849a9912c873fb0225e8354fc895533fd8136af8379902aea27

References

LuoYu: The Eavesdropper Sneaking in on Multiple Platforms *(PDF)*

Malware WinDealer Used by LuoYu Attack Group

LuoYu: Continuous Espionage Activities Targeting Japan With the New Version of WinDealer in 2021 *(PDF)*

WinDealer Dealing on the Side

Tricky Chinese-Targeted Trojan Bypasses Authentication

BlackBerry Assistance

If you're battling this malware or a similar threat, you've come to the right place, regardless of your existing BlackBerry relationship.

The BlackBerry Incident Response team is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you, providing around-the-clock support where required, as well as local assistance. Please contact us here: https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment

Related Reading:

About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

Back