# Investigation into PlugX Uncovers Unique APT Technique

By Gilbert Sison, Abraham Camba ( words)

Published: 2021-01-20 · Archived: 2026-04-05 21:40:53 UTC

Advanced persistent threats (APT) are known — and are universally dreaded — for their stealth. Actors behind such attacks actively innovate their techniques to evade detection and ensure that they maintain a foothold inside an environment as long as possible. Through the Apex One with Endpoint Sensor (iES)products, we discovered one such incident wherein an attacker utilized sophisticated techniques in an attempt to exfiltrate sensitive information from a company. The unique tactics, techniques, and procedures (TTPs) used in this attack highlight the importance of cross-layered detection and response solutions.

## Technical analysis

### Detection

We noticed the execution of *schtasks.exe* with the command line parameter "*schtasks /create /tn <name> c:\programdata\<software name>\<file name>.bat /sc /once /st <time> /ru <user account>*". The scheduled task was not created for persistence. The batch file that was to be executed had a suspicious name that stood out. This prompted us to dig deeper.

### Evasion

The image file of the process residing in the Windows directory that triggered the *schtasks* command was marked as normal. However, a quick check on VirusTotal showed that the file's name and usual location differed from what was found in the victim machine.

### PlugX and malicious BLOBs

The normal file loads a seemingly normal dynamic-link library (DLL) named after a common Microsoft DLL. However, we saw that the hash of the DLL does not match any of the known hashes of a normal DLL.

Reversing the file revealed that it was the PlugX loader, a remote access tool (RAT) that has been historically used in attacks targeting government-related industries and organizations. We then observed that it decrypts, loads, and executes another DLL file named after another Microsoft DLL file, but is actually an encrypted Binary Large Object (BLOB). Figure 1 below shows the relationship between these three files.

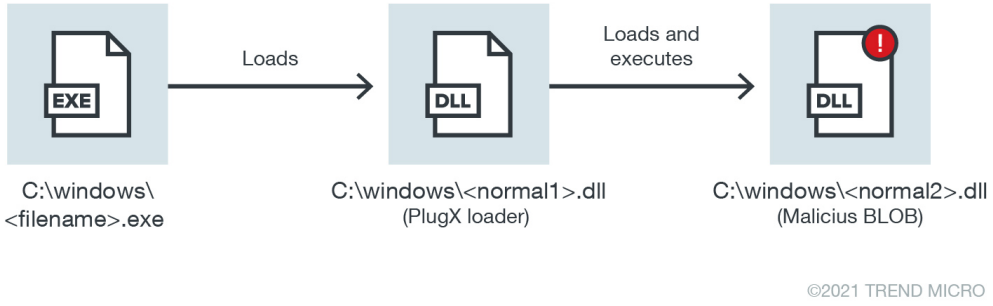


Figure 1. The relationship between the normal file used, the PlugX, and the malicious BLOB

Despite being around for quite some time, PlugX is still effective at evading detection. The variant used in this attack has three parts:

1. A normal file
2. A DLL loader that the normal file expects to be present
3. An encrypted BLOB file containing the malicious code

Of the three, the loader would be the easiest to detect. However, because the file is so simple and straightforward, it is easy for attackers to create multiple versions of the said file to evade detection.

Detection support for the BLOB is probably low. Because of its being "free form," detection engines would have a hard time parsing through its contents to find the malicious code which potentially made detection for EPPs and EDR difficult. Aside from this, changing the encryption mechanism or packer would result in a totally different BLOB. It should be noted that we have also observed an increase use of BLOB files since 2020.

The malicious code would run in the address space of the normal file. This could allow the malicious code to evade behavior-monitoring features as endpoint protection platforms (EPP) would see the process as normal. This is detailed in the next section.

## A closer look at PlugX

### *svchost* injection

This variant of PlugX (detected by Trend Micro as Backdoor.Win32.PLUGX.DYSGUT), including the code in the BLOB, launches an *svchost.exe* process and injects to it. Usually, *svchost* is launched by *services.exe*, so the action done by PlugX could be used as a trigger for investigation.

*Svchost* injection is not new — in fact, Trickbotnews- cybercrime-and-digital-threats and other malware variants use this technique. The difference between Trickbot and PlugX is that Trickbot is much easier to detect. Trickbot's binary, which is not whitelisted, does the injection; in PlugX's case, a normal process that is possibly whitelisted, does the injection. It is possible for EPP behavior-monitoring features to allow the action of the whitelisted process to go through. This gives the attacker access to a RAT running inside a normal process.

### A unique way of launching tools

Our investigation shows that the attacker used other tools during the attack. The tools used varied from network scanners to account harvesters. The simplest way of using these tools is to have the injected code in *svchost* download, drop, and execute the tools.

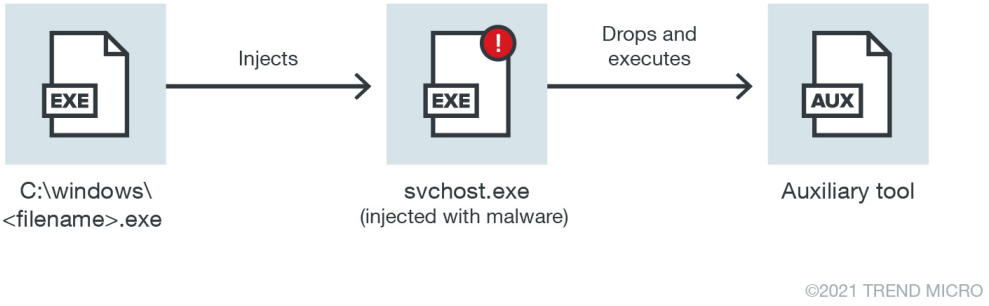


Figure 2. A straightforward approach to launching a malicious tool

The approach above, although simple, could potentially generate a lot of suspicious telemetry events – *svchost.exe* suddenly launching an unknown application could trigger alerts. To avoid this, the attacker chose a different approach. For every tool that needs to be run, it creates three things:

1. The auxiliary tool file
2. <filename>.bat file
3. *Cmd.exe* with a *schtasks.exe* command to create a scheduled task to run the batch file above

Instead of directly launching the auxiliary tool, it makes use of a scheduled task that runs the batch file that, in turn, executes the tool.

The attacker's usage of a tool can be divided into two separate events: the dropping of the tool by the *svchost.exe* process that runs the malicious code and the execution of the tool via the scheduled task.

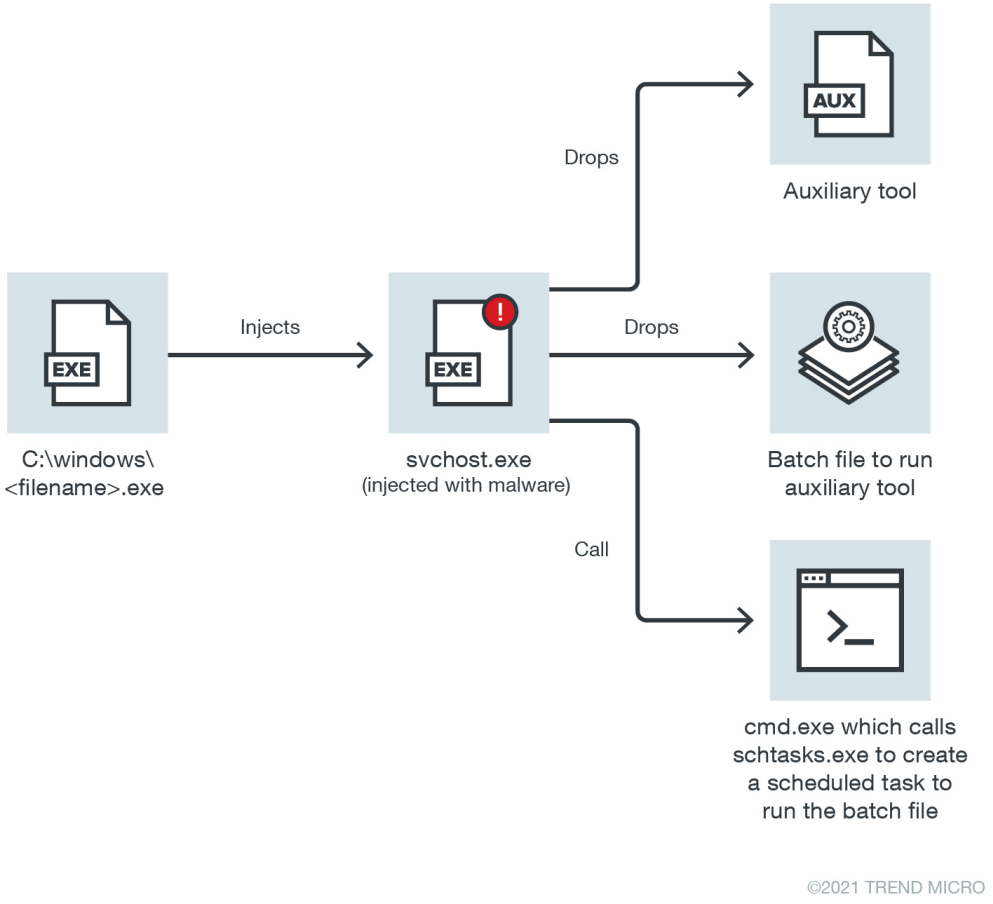


Figure 3. A diagram showing the dropping of the auxiliary tool

If the creation of the tool, batch file, and scheduled task is disregarded, it would seem that there is no connection between PlugX and the execution of the auxiliary tool. However, a deeper investigation proved otherwise.

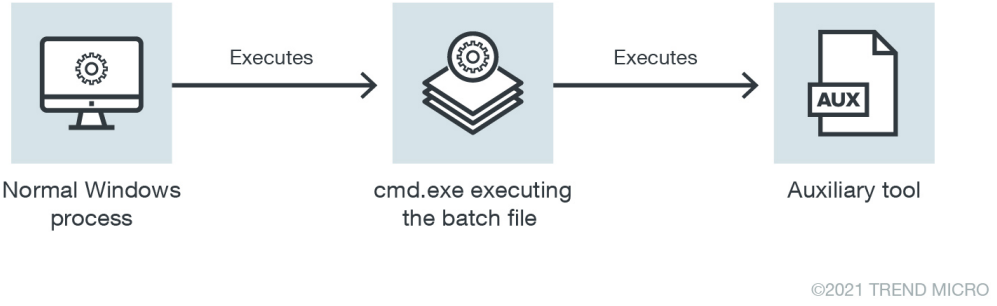


Figure 4. Launching the auxiliary tool via the scheduled task

Scheduled tasks are usually associated with persistence or privilege escalation, which is not the case in this incident. Having the scheduled task run only once removes the notion of persistence. Meanwhile, having the same user account launch the PlugX process and the auxiliary tool shows that the scheduled task is not for privilege

escalation. One possible reason why the attacker chose this technique is to make analysis and forensics, such as via EDR, more difficult.

The disjoint between the dropping and the execution of the auxiliary tool might cause problems in EDR's rendering of RCA graphs, specifically in providing important information about the attack. Security analysts will have to rely on telemetry data to provide a more complete picture of said attack.

In this incident, the attacker used a specific tool to gather information kept by the organization. However, the attacker was able to delete it before we were able to get our hands on the tools used for investigation purposes. An analysis of these tools could be used to better understand what is happening or what has happened in the environment. Whenever a tool is used, the attacker deletes it from the disk as soon as it has done its purpose – which usually does not last for more than 15 minutes. This means that acquiring the tools used would be more difficult. For the attacker, immediately deleting the tool once it performs its intended purpose means being able to extend its shelf life.

## What enterprises and organizations should look out for

In incidents like this, wherein the attacker makes sure that the tools used are not detected, threat hunting is a good weapon to have for defense. Sifting through suspicious events and trying to piece them together to understand what has transpired could help identify such attacks.

Based on our investigation of this specific attack, we provide helpful tips for enterprises and organizations to spot PlugX and lateral movement in their environments.

**How to identify PlugX**

PlugX's use of a normal file makes it hard for antivirus (AV) or EPP to detect it. Threat hunters can use this to possibly identify other machines that have PlugX installed: Using attributes that could identify the normal file, such as the hash and the digital signer, a sweeping task could be carried out to identify instances of the normal file. Chances are, the normal file that loads the rest of PlugX uses a file name that is different from what it normally uses or is in a location where it is not usually found. For example, Microsoft's *outlook.exe* is usually found in the Program Files directory. Finding a normal *outlook.exe* in the Windows directory makes it suspicious. Finding such entries that stand out could help identify how wide the infection is.

**How to spot lateral movement**

In this particular attack, we observed that to perform lateral movement, the attacker mapped Windows admin shares and used remotely created scheduled tasks to launch malware. Both events never or rarely happened in the victim's environment.

### The importance of looking into anomalous events

When identifying PlugX and lateral movement techniques utilized in attacks, it's important to be able to determine anomalous events. It was relatively easy for us to flag the events related to the breach because they were anomalies; after looking into these anomalies, we were able to map them out to known techniques that attackers previously used. Keeping an eye on anomalous events could help detect incidents earlier. In order to have

effective visibility and correlation at all times, enterprises need to have an integrated view of all of their interconnected security solutions. Automated protection and efficient threat hunting, investigation, and response, such as that provided by Trend Micro XDRservices, will help ensure that enterprises are secured from advanced threats.

## Trend Micro Solutions

Trend Micro's comprehensive XDRservices solution applies the most effective expert analytics to the deep data sets collected from Trend Micro solutions across the enterprise — including email, endpoints, servers, cloud workloads, and networks — making faster connections to identify and stop attacks. Powerful artificial intelligence (AI) and expert security analytics correlate data from customer environments and Trend Micro's global threat intelligence to deliver fewer, higher-fidelity alerts, leading to better, early detection. One console with one source of prioritized, optimized alerts supported with guided investigation simplifies the steps needed to fully understand the attack path and impact on the organization.

Tags

---