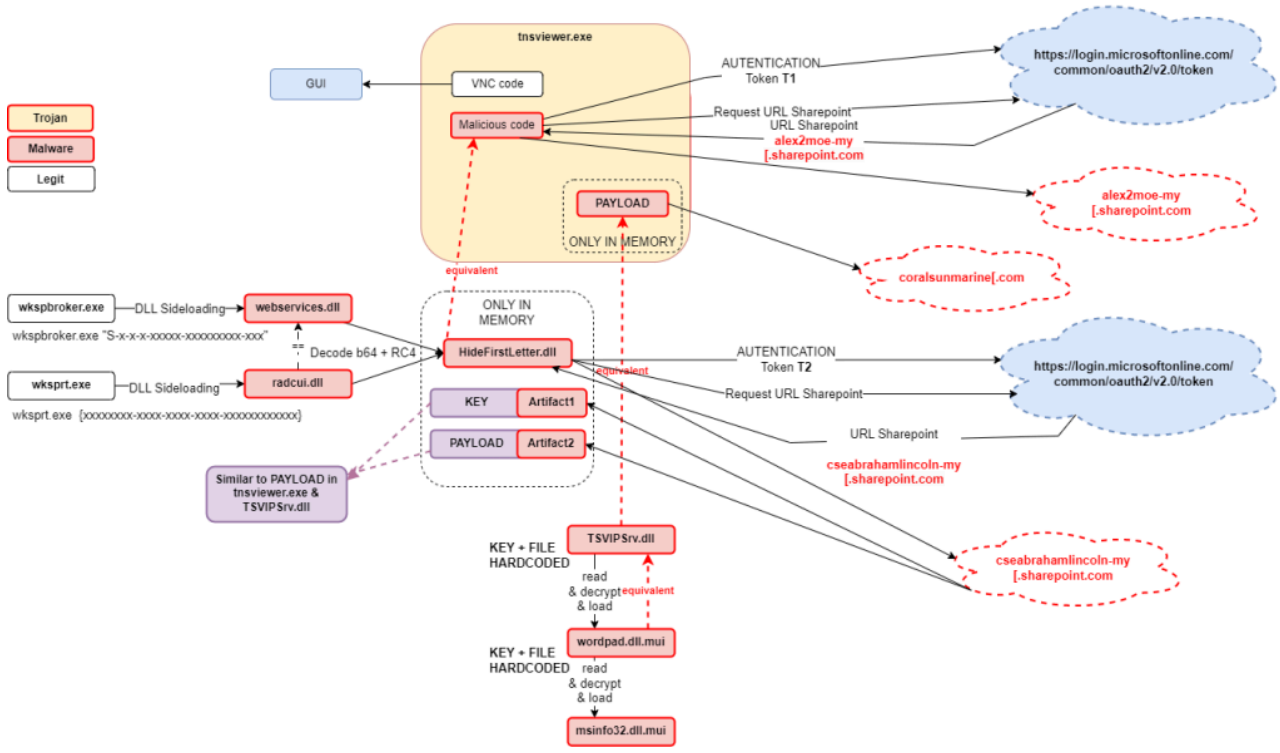
From Dream Job to Malware: DreamLoaders in Lazarus' Recent Campaign

10ba ⋮

During August 2025, Lab52 gained access to artifacts linked to Lazarus through DreamJob campaigns. Some of these artifacts and their operational details were recently highlighted by ESET (e.g., *radcui.dll*, *HideFirstLetter.dll*).
From our perspective, one of the most notable aspects of this campaign is the use of various types of loaders — components capable of deploying different payloads depending on the actors' needs.
These loaders are used in the DreamJob campaign, but we believe they could also appear in other operations. For us, they truly are ***dream loaders***.
The operational flow observed by our team is shown in the following diagram.



In this article, we describe the relationship between these artifacts and also detail the characteristics of **TSVIPSrv.dll**, a loader used by the group in the analyzed case.

During the investigation, two deployment methods were observed, one of them involving the use of **legitimate system executables** to load the various loaders through **DLL sideloading**.

Given the diversity of loading mechanisms and their connection to the **DreamJobs** campaigns, we internally coined the term **“DreamLoaders”** to refer to this type of loader.

Findings summary

As in other **DreamJob** cases, the attackers—in this case, the **Lazarus** group—aim for administrators of the targeted organizations to execute malware in order to extract credentials or other types of information, which are later used to gain access and perform further actions on the compromised systems.

The analyzed case is particularly interesting because **three deployment variants of the same loader** were observed, which is the focus of this report.

Tnsviewer.exe

A trojanized version of the TightVNC client, distributed inside a **password-protected ZIP file** along with a README.txt file containing instructions about the IP address the victim (i.e., the administrator) should connect to.

This binary was already available on VirusTotal at the start of the analysis. Behavioral data, contacted domains, and a memory dump of the process (executed via sandbox) were collected. This allowed analysts to retrieve a payload whose structure and characteristics are very similar to another identified artifact: **TSVIPSrv.dll**.

Executing this artifact creates registry keys used by TightVNC and triggers the malware’s operation, which is equivalent to the functionality of the **HideFirstLetter.dll** artifact described below.

It is designed as a **decoy**, intended to be executed by an administrator.

Webservices.dll and radcui.dll

Webservices.dll and radcui.dll were identified on compromised user systems. They are **equivalent DLL loaders** that are executed through **DLL sideloading**, using the legitimate binaries **wkspbroker.exe** and **wksprt.exe** (copied beforehand from C:\Windows\System32) and a password following a **SID-like pattern**, likely to blend in.

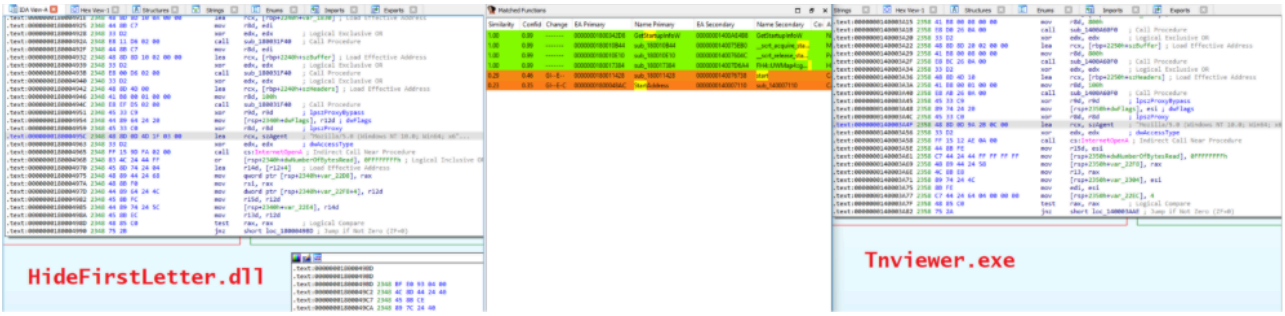
These DLLs differ only in that each contains an **encrypted payload**, encoded in Base64 and decrypted using a different key (the SID-pattern argument passed at execution). Once decrypted and injected into memory, the payload is a DLL named **HideFirstLetter.dll**.

HideFirstLetter.dll performs the same malicious activity observed when executing tnsviewer.exe:

- It attempts to authenticate to the tenant using the legitimate Microsoft URL. https://login.microsoftonline.com/common/oauth2/v2.0/token, leveraging an access token embedded in the binary.

```
.data:000000018004C35E                  db    0
.data:000000018004C35F                  db    0
.data:000000018004C360 a1Atqarazm57e8o db '1.ATQAraZm57E8OEO1BmepIHY1GAXupyYCVnZNp7rq6Le2eUE0AMc0AA.AgABAwEA'
.data:000000018004C360                                      ; DATA XREF: sub_180007610+48↑o
.data:000000018004C360                                      ; .data:off_18004CD98↓o ...
.data:000000018004C360                  db 'AABVrSpeuWamRam2jAF1XRQEAwDs_wUA9P8Tzpl2eHiQJH99CUT30uUuIKnI_cWhU'
.data:000000018004C360                  db '-KI7zbVb6qM-niNdqakqMNpWoJYAGy2S7cJDXHAan3cnrYHlf0lS8RHn-DB6Eykf0'
.data:000000018004C360                  db '2FA5wi2_GjPg1Isw77gLfVndmynYYaLMhsVXnWuu5s7O2IOh8oNO305Qx6qk_oTo7'
.data:000000018004C360                  db 'KTPiQK2s3WRQhLrOddUIm_QrJW3qkZDq1ckXAPOBcIInHS8V7FkEVYUjPoLS5yJk6'
```

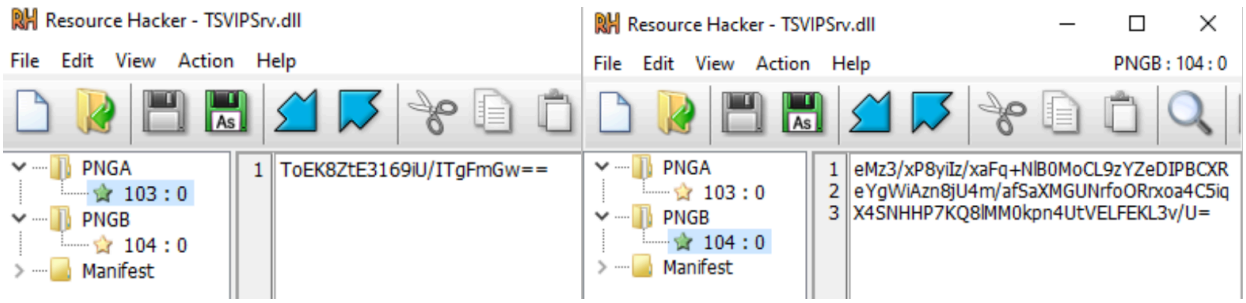
- Then, it sends a request to the **Microsoft Graph API** to retrieve the URL of the compromised **SharePoint** server.
- Two Microsoft API URLs were found, although only one was reachable at the time of analysis.
- The returned value could serve as a key to decompress another payload, presumably obtained from the second URL. This hypothesis is based on the behavior of the other artifacts and their relationships —particularly between the payload extracted from tnsviewer.exe and **TSVIPSrv.dll**.

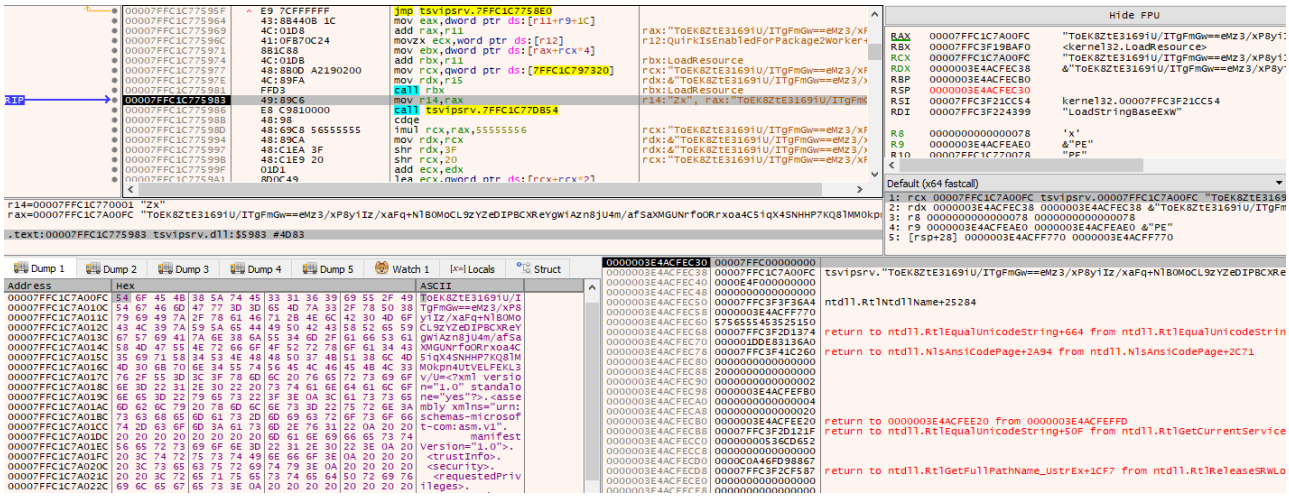


TSVIPSrv.dll

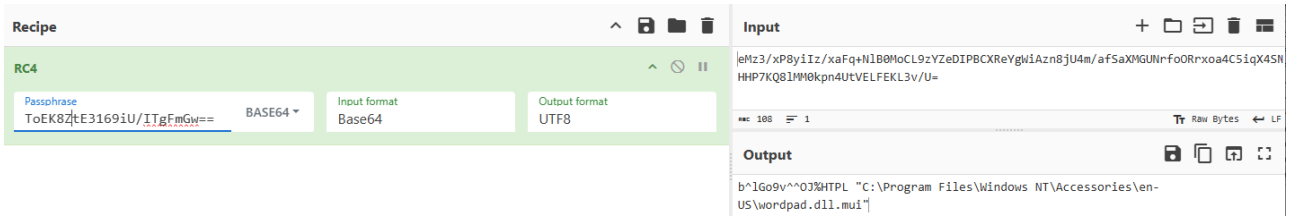
TSVIPSrv.dll is a loader identified on compromised servers. It is a DLL that is executed via a **malicious service created by the attackers**, named **sessionenv**. It also relies on two additional files previously placed by the attackers—**wordpad.dll.mui** and **msinfo32.dll.mui**—which contain the payloads to be loaded. During the investigation, only **wordpad.dll.mui** was accessible.

It includes two Base64-encoded resources: the first contains a password, and the second contains a file path to the next payload.



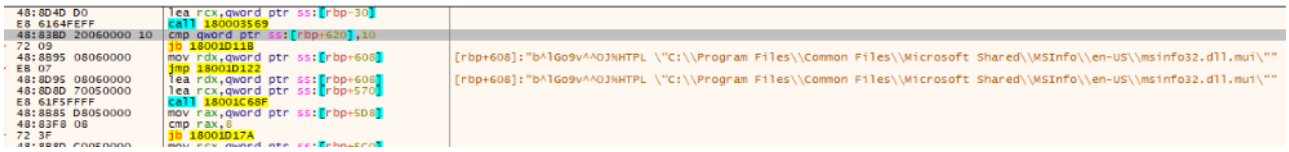


It uses **RC4** to decrypt the file path (pointing to **wordpad.dll.mui**).



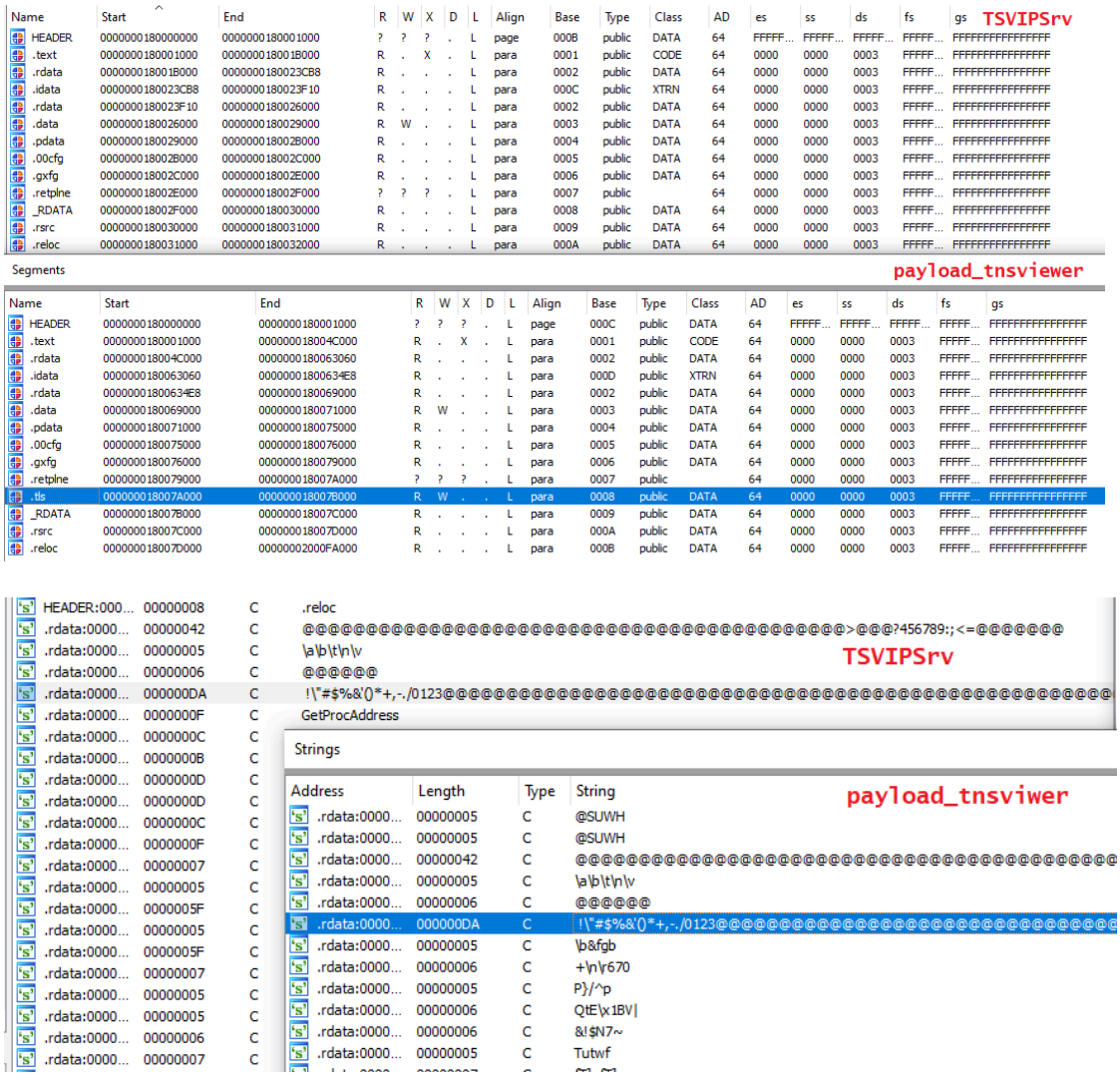
It then decrypts and loads the contents of **wordpad.dll.mui**, which is again a variant of **TSVIPSrv.dll** that performs the same actions: read its own resources (this time in a different order), decrypt them with RC4, access the file, and attempt to load it.

Similarly, when this resource is decrypted, a path will be obtained that will load the following file into memory: **C:\Program Files\Common Files\Microsoft Shared\MSInfo\en-US\msinfo32.dll.mui**.



An important detail is that TSVIPSrv.dll, can be used to load different modular payloads, since **the content is independent and stored in other files (the .mui files)**. Therefore, it is significant that, for the two DLLs found on two different machines, the collected .mui files are identical. This implies that the same payload was deployed on both machines.

So, TSVIPSrv.dll decrypts the file wordpad.dll.mui, which in turn is another DLL very similar to TSVIPSrv.dll. A comparison between that DLL once decrypted and the payload of tnsviewer.exe shows an 85% code similarity, suggesting that the final DLL might be even more similar.



Conclusion

The investigation into Lazarus group's DreamJobs campaign reveals a sophisticated and modular malware deployment strategy, leveraging legitimate system binaries and encrypted payloads to evade detection. Through the analysis of artifacts like TSVIPSrv.dll, HideFirstLetter.dll, and trojanized tools such as tnsviewer.exe, it is possible to see a high degree of code reuse and stealthy sideloading techniques. The presence of identical payloads across different systems is part of a coordinated and targeted effort.

This case underscores a critical lesson: attackers continuously refine their methods, often hiding in plain sight by mimicking legitimate processes. Therefore, vigilance, proactive threat hunting, and deep technical analysis remain essential in defending against advanced persistent threats.

IOC

Files

Name	Hash sha256
tnviewer.exe	aefc12b500b58fbc09ebbf34fe64b34cb32a27513478f4769447280ad23af4d2 0fdd97a597380498f6b2d491f8f50da8f903def4ea6e624b89757456c287f92d
radcui.dll	fa014db2936da21af5943cc8f3656adb9800173ad86af196f71c6052295fff97
webservices.dll	26bd4aab63563e77ca426c23b11d18d894eef9a727e111be79336e099b22bdd1
TSVIPSrv.dll	473726dd9bc034564c4c7b951df12d102ff24f7b17b8356f55d36ed6d908882d
wordpad.dll.mui	b3d7a3c3dedaa873e81b1676b6c0027ae1fd164587299bf65c02bd067ae1a972
wordpad.dll.mui decyphered (only in memory)	855baa2ff0c3e958a660ae84a048ce006e07cf51ce5192c0de364ee62873980c

Comunications

Artifact name	Domain
tnviewer.exe	alex2moe-my.sharepoint[.com coralsunmarine[.com
HideFirstLetter.exe	cseabrahamlincoln-my[.sharepoint.com