

LOTUSLITE: Targeted espionage leveraging geopolitical themes

Archived: 2026-05-22 02:01:24 UTC

Authors: Ilia Dafchev, Subhajeet Singha

Summary

- Acronis Threat Research Unit (TRU) observed a targeted malware campaign against U.S. government entities leveraging a politically themed ZIP archive containing a loader executable and a malicious dynamic-link library (DLL). The executable is used to sideload and execute the DLL, which functions as the primary backdoor, tracked as **LOTUSLITE**.
- The backdoor, referred to as **LOTUSLITE**, is a custom C++ implant that communicates with a hard-coded IP-based command-and-control server and supports basic remote tasking and data exfiltration with a decent persistence technique indicating an espionage-focused capability set rather than financially motivated objectives.
- The loader demonstrates **low development maturity**, with minimal error handling and limited defensive evasion, suggesting rapid operational deployment rather than a long-term, well-maintained malware framework.
- Infrastructure analysis and execution patterns show **moderate-confidence overlap with Mustang Panda tradecraft**, including delivery style, loader–DLL separation and infrastructure usage. Attribution is assessed at a behavioral level and does not rely on code reuse alone.
- This campaign reflects a **continued trend of targeted spear phishing using geopolitical lures**, favoring reliable execution techniques such as DLL sideloading over exploit-based initial access.
- The observed targeting is limited to **U.S. government and policy-related entities**, indicating a focused victim set. While the overall scale appears limited, the nature of the targets increases the potential strategic impact.

Introduction

Acronis Threat Research Unit (TRU) has been actively monitoring malware campaigns and threat activity leveraging **recent geopolitical developments between the [United States and Venezuela](#)** as thematic lures. During this tracking, TRU identified a targeted campaign delivering a previously undocumented DLL-based backdoor, tracked as **LOTUSLITE**, aimed at U.S. government–related entities.

The activity stood out due to its use of politically themed lure material packaged within a ZIP archive and a simple execution chain involving a loader executable and a malicious DLL. These observations prompted further analysis to understand the purpose of the tooling, the relationship between its components and the infrastructure used to operate the backdoor. This report focuses on documenting the delivery mechanism, execution flow and command-

and-control behavior associated with LOTUSLITE, as well as examining how this activity aligns with broader espionage-oriented threat trends. While the malware itself demonstrates limited technical sophistication, its selective targeting and contextual lure usage indicate deliberate victim selection.

Through this research, TRU outlines the analytical basis for attributing this activity to **Mustang Panda with moderate confidence**, primarily based on observed **infrastructural overlaps**, deployment patterns, and operational characteristics consistent with previously documented campaigns. The report details the indicators and behaviors that informed this assessment, providing defenders with context for understanding how such overlaps can support attribution without relying on code-level similarities alone.

Background and context

Mustang Panda is a long-running espionage-oriented state-aligned threat entity, known for aligning its operations with **current geopolitical developments**. The group has consistently leveraged themes tied to [international conferences](#), [bilateral engagements](#) and [region-specific political events](#) to support targeted intrusion activity against government and policy-related entities.

Operationally, Mustang Panda favors medium-**complexity, repeatable execution techniques**, most notably the extensive use of **DLL sideloading** to deploy custom implants via benign or trusted executables. The group has also demonstrated repeated **reuse of infrastructure and tooling**, enabling analysts to cluster activity and assess attribution, even in the absence of direct malware code reuse. The campaign examined in this report reflects these established behaviors and introduces a **newly observed backdoor**, providing relevant context for the analytical conclusions presented in subsequent sections.

Technical details

Initial analysis and delivery mechanism

Our investigation began after identifying a spear phishing archive named **US now deciding what's next for Venezuela.zip** that was uploaded for automated malware analysis from an IP address geolocated in the United States. The archive included a legitimate executable and a hidden, nonstandard DLL, a combination frequently associated with Mustang Panda tradecraft. Execution of the binary resulted in the DLL being loaded through sideloading, enabling covert execution of the malicious code.

The activity in terms of the naming convention stood out due to the politically themed archive name and the minimalistic execution chain, which aligns with the patterns observed in a target intrusion activity. These observations prompted a deeper analysis of the loader behavior and its capabilities, payload execution flow and the infrastructure used to control the implant, with its attributional artifacts.

DLL sideloading

In the very same directory where the launcher was located, we could previously see only one single file; then, once the attribute for the files was changed, the implant, which is a dynamic load library (DLL) called **kugou.dll**, was clearly visible.

Further, looking into the executable launcher named as *Maduro to be taken to New York.exe*, which is basically a renamed and is a launcher binary for a [music streaming](#) service owned by Tencent, it explicitly loads the malicious DLL using [LoadLibraryW](#) and resolves the exported function via [GetProcAddress](#) and transfers execution without relying on the import table or perform implicitly loading it.

LOTUSLITE backdoor

Upon further inspection, we found that the *kugou.dll* which had been loaded by the launcher executable turned out to be acting as a backdoor, which we track as LOTUSLITE with the sole purpose of providing access to the threat actor, and execute certain commands depending on the circumstances.

Although the DLL file has multiple exports, the interesting function, which is *DataImporterMain* is being executed as part of the backdoor-based functions.

Initially, this interesting function performs a set of multiple and genuine functionalities such as loading the binary into memory and multiple string-concatenation-based operations further leading to two important functions out of them, one is responsible for establishing persistence and spawn a decoy file onto the target machine, while the other set is responsible for beaconing-based functionality.

Further, during analysis of the DLL initialization routine, we observed that a significant portion of the implant logic is executed **before DllMain is reached** through the Microsoft C Runtime (CRT) initialization mechanism by leveraging `_initterm`, which iterates over a linker-generated array of function pointers (located in `.CRT$XC*` sections) and invokes each non-null function.

The initializer array of functions contains multiple functions related to the boiler-plate code, along with a handful of interesting functions such as setting up values for Mutexes and directory location for persistence, as well as the address for the command-and-control (C2) server.

Further, looking into the function responsible for beaconing-based tasks, it starts with a Sleep function, followed by a loop which depends on the beacon update or simply depending on the threat actor's command, in case the communication needs to be reset.

Followed by which, there is another set of interesting function, which we have renamed as *BuildC2PacketFromString* and *BuildC2Packet* which are responsible for framing the packets, with a magic-ID header in hex, that is `0x8899AABB`, upon which the data is transferred to the C2 server. The packet structure looks like this:

The binary protocol in the POST request body supports multiple packets in the same POST request

Next, the implant goes ahead and enumerates the names of the machine, along with the target's username using APIs such as *GetComputerName* and *GetUserName*.

Once this initial set of basic enumeration steps is completed, the backdoor proceeds to execute a predefined set of instructions, which can be dynamically leveraged based on the threat actor's operational requirements.

The first capability of the backdoor is the creation of an interactive cmd.exe shell with redirected standard I/O over anonymous pipes, enabling remote command execution and real-time command output retrieval.

The very next feature is that it also allows the threat actor to terminate the interactive shell, which was initially spawned by the backdoor, using the [TerminateProcess](#) API.

Depending on the threat actor's command, the backdoor also can enumerate files in multiple different directories.

Another feature is the backdoor's ability to perform various file operations, including creating files, adding data and more.

The last, but not least feature of this implant is that it also checks the status of the current beacon and reports back to the command-and-control system.

Therefore, the action codes for this implant are as follows:

Command and control

The implant relies on Windows WinHTTP APIs to connect to its command-and-control server, which helps it blend in with normal web traffic. After setting up an HTTP session and applying long timeout values, it connects to the remote server and prepares a POST request to an API-like endpoint.

The request itself is carefully made to look harmless. It uses a Googlebot User-Agent string, sets the referrer to Google, and presents the Host header as a Microsoft domain. Along with this, a fixed-session cookie is included, which likely serves as a simple way for the server to recognize the infected host. Together, these details help the traffic appear routine and reduce the chances of it standing out in network logs.

And, as mentioned above, the implant uses a hardcoded header sequence that is a small "magic" marker. This marker appears to tell the server that the request is coming from a valid implant. Only after this marker is added does the malware attach to the actual POST body, which is kept deliberately small and is likely used for basic check ins or task responses.

Persistence

The implant persists by initially creating a directory using the CreateDirectoryA API under the location of C:\ProgramData\Technology360NB.

Also, it renames the executable launcher to DataTechnology.exe and passes the command line argument -DATA, which is an internal execution-based flag. Then, once it is done, it uses the SHSetValueA API to create a registry entry under the current user's **Run** key, which is a classic technique for establishing persistence on the target machine under the value named Lite360. So therefore, it makes sure that the implant is automatically executed each time the user logs in.

Messages via exports

Although the other set of exports are not quite as interesting compared to the one which contains the code for the beaconing tasks and persistence, the dynamic-link-library also exports two interesting dummy functions known as **EvtNext** and **EvtQuery**, although these functions are mainly responsible for event-queries-based tasks, but in this instance, it is just used to deliver a few messages.

Two distinct functions contain developer-inserted messaging that appears to reference national identity. One function includes text distancing the author from a Russian origin, while the other contains an explicit self-identification statement claiming Chinese identity.

Infrastructural artifacts

The samples we analyzed were observed communicating with a command-and-control server hosted at **172[.]81[.]60[.]97**, which resolves to the hostname **unassigned.172-81-60-97[.]sprit[.]net**. The infrastructure is in **Phoenix, Arizona (United States)** and is part of **ASN AS398019**, operated by **Dynu Systems Incorporated**, a provider commonly associated with dynamic DNS and hosting services.

According to our analysis, the implant establishes outbound connections to this C2 server over **TCP port 443**, consistent with attempts to blend malicious traffic within normal HTTPS activity. Further looking into the passive network observations indicates repeated connections to this endpoint across multiple dates, suggesting it is actively used as a staging or beaconing server.

Attribution and confidence

Based on our analysis, we assess with moderate confidence that this campaign is attributable to Mustang Panda. This assessment is based on multiple independent lines of analytical evidence as outlined below.

- **Using similar launcher executable techniques:** Upon pivoting from the launcher executable, we found a compressed file containing an executable named **ASEAN Leaders Meeting.exe**. This file, which also originally belongs to KuGou, was used by Mustang Panda to target entities across the European region, as published by Cisco Talos in 2022.
- Further, looking at the recent implant, which we track as **LotusLite**, mimics the behavioral shenanigans of **ClaimLoader** by embedding provocative messages. Similar behavior has been observed across multiple campaigns related to Mustang Panda, as the researchers from [IBM-XForce](#) tracked during May 2025.

Building on pivoting through with shared tradecraft, including anomalies such as leaving uncommon behavioral trails and focusing on the tactics, techniques, and procedures (TTPs), we assess with moderate confidence that this campaign falls within the Mustang Panda activity cluster.

Attack chain

Conclusion

This campaign demonstrates how simple and well-tested techniques can still be effective when paired with targeted delivery and relevant geopolitical lures. Although the LOTUSLITE backdoor lacks advanced evasion

features, its use of DLL sideloading, reliable execution flow and basic command-and-control functionality reflects a focus on operational dependability rather than sophistication. With moderate confidence, we attribute this activity to Mustang Panda based on overlapping tradecraft and behavioral patterns observed in earlier campaigns.

Detection by Acronis

This threat has been detected and blocked by Acronis EDR / XDR:

IoCs

SHA256 hashes:

Maduro to be taken to New York.exe

819f586ca65395bdd191a21e9b4f3281159f9826e4de0e908277518dba809e5b

Kugou.dll 2c34b47ee7d271326cfff9701377277b05ec4654753b31c89be622e80d225250

Paths used for persistence:

- C:\ProgramData\Technology360NB

Mutex:

- Global\Technology360-A@P@T-Team

C2 IPs:

- 172[.]81[.]60[.]97

Source: https://www.acronis.com/en/tru/posts/lotuslite-targeted-espionage-leveraging-geopolitical-themes/#-m_BvsUlu9