ICT DIVISION
FUTURE IS HERE

BANGLADESH COMPUTER COUNCIL
COMPUTER FOR EVERYTHING

BGD e-GOV CIRT

TLP:CLEAR

SideCopy Campaign Using MSHTA, Weaponized LNK and Fileless RAT Execution

TLP: CLEAR
Distribution: Public
Advisory on: SideCopy Campaign Using MSHTA, Weaponized LNK and Fileless RAT Execution
Severity: High
Threat Type: APT / Cyber Espionage, RAT Deployment
Potentially Affected Sectors: Government, Defense & Security, Academic/Research, Critical Infrastructure
Date: 23 September 2026

Executive Summary

BGD e-GOV CIRT has reviewed recent reporting on a **SideCopy** campaign employing spear-phishing, weaponized Windows shortcut (`.LNK`) files, `mshta.exe`, multilayer script obfuscation, .NET deserialization and reflective DLL loading to deploy a Remote Access Trojan (RAT).

The campaign begins with a weaponized ZIP archive containing an LNK file disguised as a legitimate document. Execution of the LNK retrieves an HTA payload from attacker-controlled infrastructure through `mshta.exe`. The HTA stage subsequently extracts additional components, establishes Registry Run Key persistence and loads the final RAT directly into the memory of `mshta.exe`. The observed RAT provides extensive remote-control and information-stealing capabilities, including **system reconnaissance, password theft, clipboard collection, screenshots, file operations, command execution and file upload**. Data is exfiltrated through an encrypted TCP connection to attacker infrastructure over **port 5863**.

The use of legitimate Windows components and fileless execution significantly reduces reliance on traditional malicious executable files and therefore **requires behavioral, endpoint, network and email-based detection**.

Key Findings

- Weaponized ZIP archives are used for initial delivery.
- LNK files are disguised as legitimate DOCX documents.
- `mshta.exe` is abused to retrieve and execute remote HTA content.
- HTA payloads use multiple layers of obfuscation.
- Malicious components are extracted from compressed resources.
- Registry Run Keys provide persistence.
- `.NET BinaryFormatter` is abused for object reconstruction.
- The final RAT is reflectively loaded into `mshta.exe`.
- RAT functionality includes credential, clipboard, screen and file collection.
- Encrypted C2 communication occurs over TCP port **5863**.
- The campaign demonstrates a combination of **living-off-the-land and fileless execution technique.**

Attack Chain

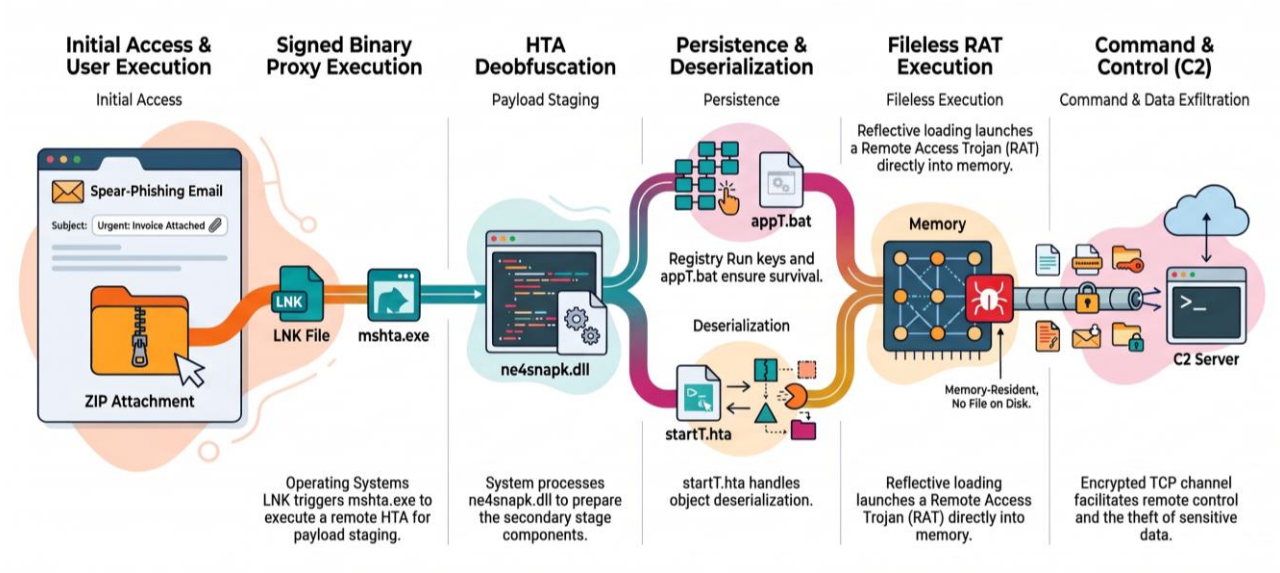

Figure: Complete SideCopy Attack Chain (from phishing to data theft)

The documented infection chain follows this progression from spear-phishing through LNK execution, HTA staging, persistence, in-memory payload activation and encrypted data exfiltration.

Initial Access — Weaponized ZIP

The campaign uses spear-phishing emails containing a **weaponized ZIP archive**. The archive contains:

- a malicious `.LNK` file;
- a decoy GIF/document;
- deceptive naming;
- a spoofed PDF/document icon.

The LNK is designed to appear as a legitimate document, increasing the probability that a recipient will execute it.

Malicious LNK Execution

The LNK contains a URL pointing to attacker-controlled infrastructure:

```
hxxps://docsportal[.]in/public/reps/com/161[.]php
```

When executed, the shortcut invokes Windows-native functionality to retrieve the next-stage HTA payload. The LNK therefore acts as a bridge between the phishing document and the `mshta.exe` execution stage.

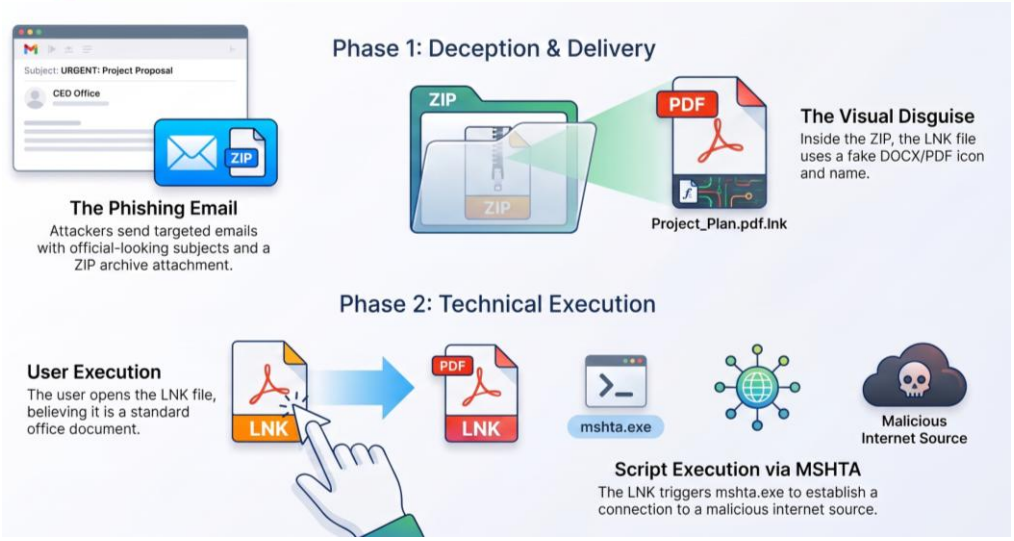

Figure: Weaponized ZIP and LNK Delivery

MSHTA-Based Payload Retrieval

The LNK retrieves an obfuscated HTA file identified as: `com.hta`
The payload is cached on the Windows system and subsequently executed using: `mshta.exe`

After the next stages are initialized, the HTA performs an anti-forensic deletion routine to remove itself from disk. This behavior is particularly important because `mshta.exe` is a legitimate Windows component but can be abused to execute malicious HTML Application content.

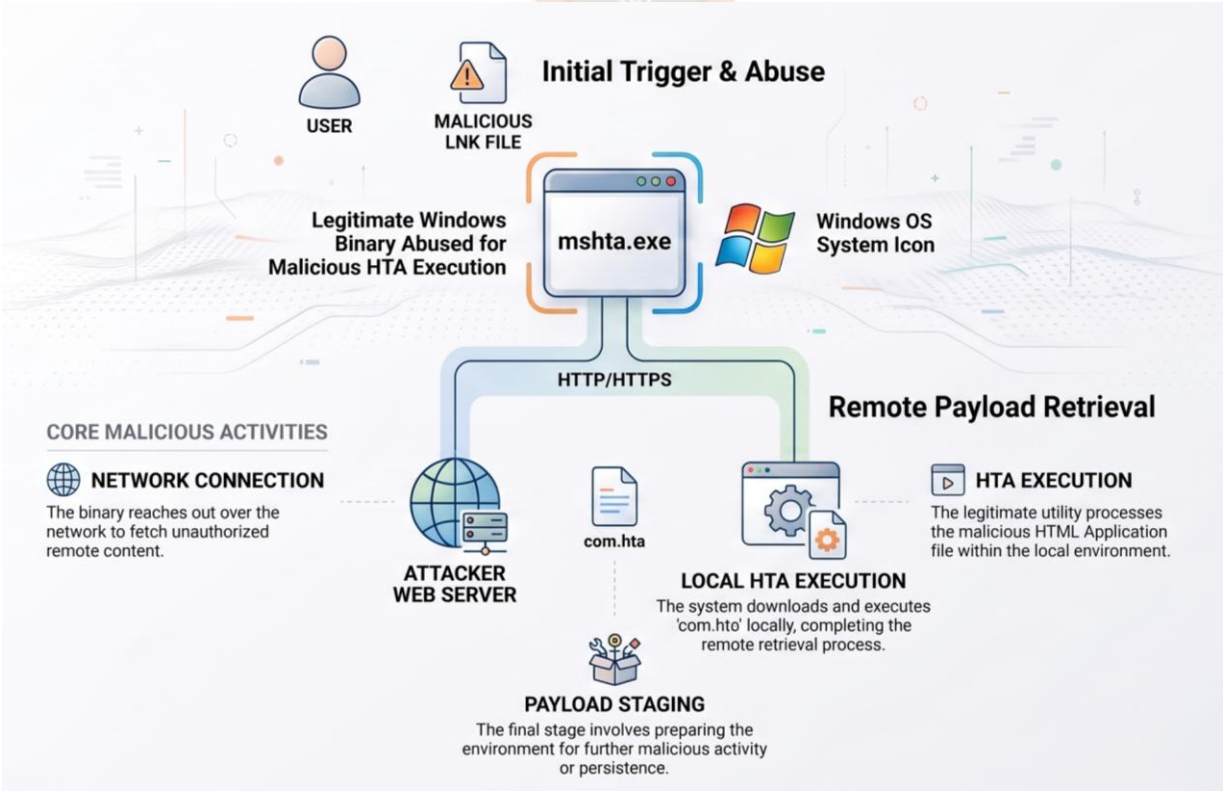

Figure: MSHTA Remote Payload Retrieval

HTA Staging and Payload Extraction

The HTA contains an embedded DLL: `ne4snapk.dll`

The DLL contains compressed components that include:

- `appT.bat`
- `startT.hta`
- `Ioluegnt.dll`

The components are stored in GZIP-compressed form and extracted during execution. The decoy document is also deployed to make the activity appear legitimate.

The resulting architecture is:

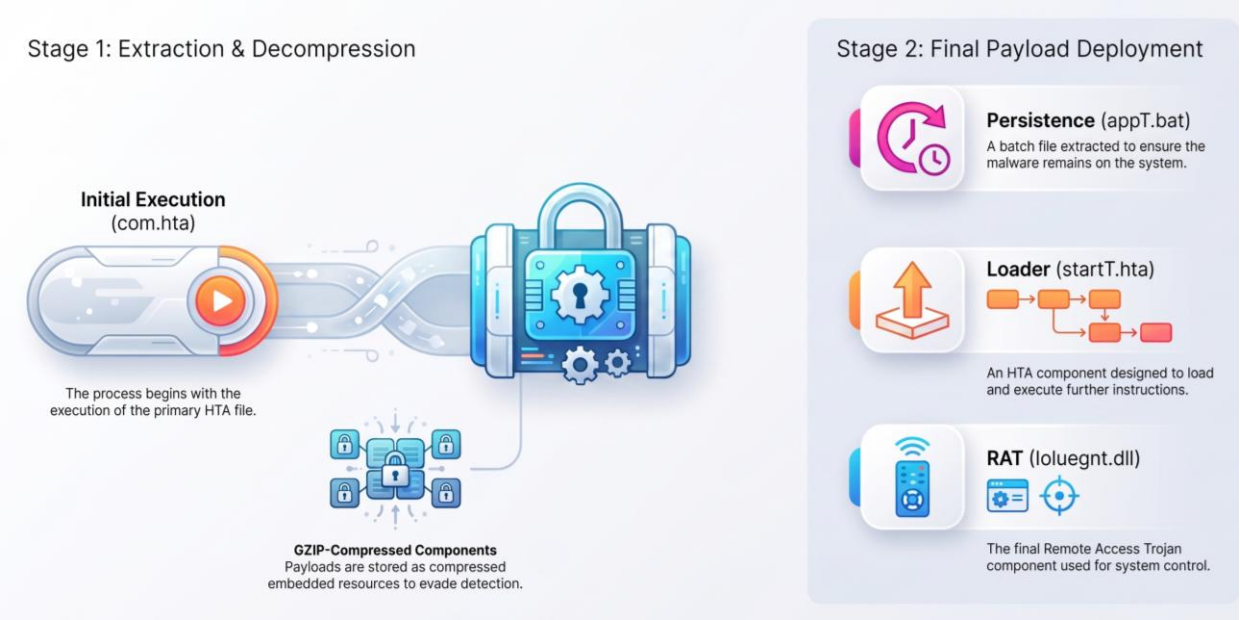

Figure: HTA Multi-Stage Payload Extraction.

Persistence Through Registry Run Key

Persistence is established through the Windows Registry. The observed location is:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

The registry entry points to: `C:\Users\Public\User\appT.bat`

The batch file launches:

```
start /b mshta.exe "C:\Users\Public\User\startT.hta"
```

The `/b` parameter allows the process to execute without opening a visible console window.

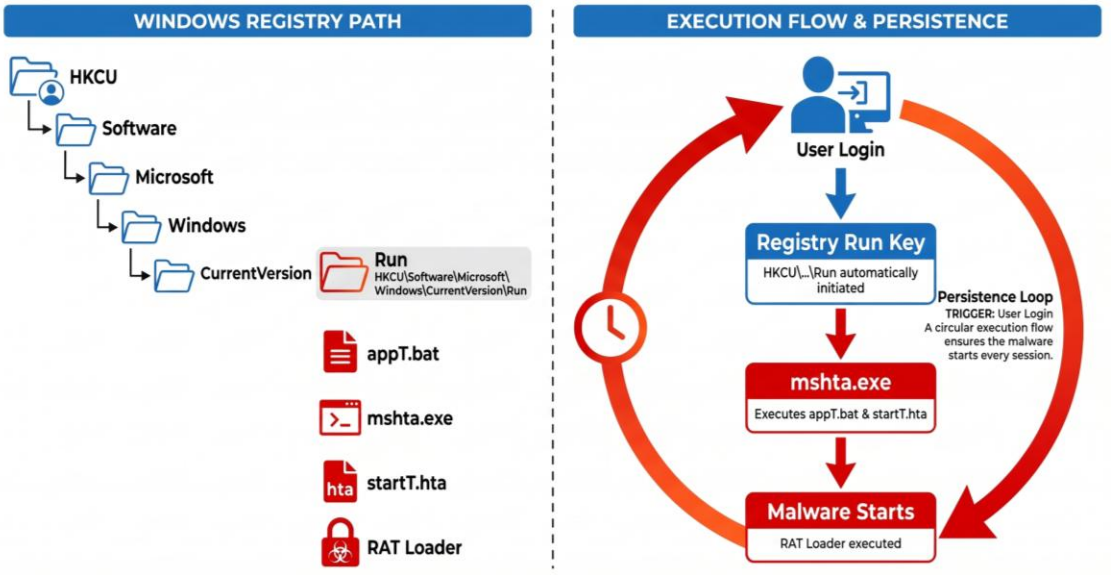

Figure: Registry Run Key Persistence (The Malware Execution Chain)

.NET Deserialization and Fileless Execution

The second-stage HTA reconstructs a payload in memory using several techniques:

- Base64 decoding
- Secondary transformation
- MemoryStream creation
- Binary object reconstruction
- .NET deserialization
- Reflective DLL loading

The campaign abuses:

`System.Runtime.Serialization.Formatters.Binary.BinaryFormatter`

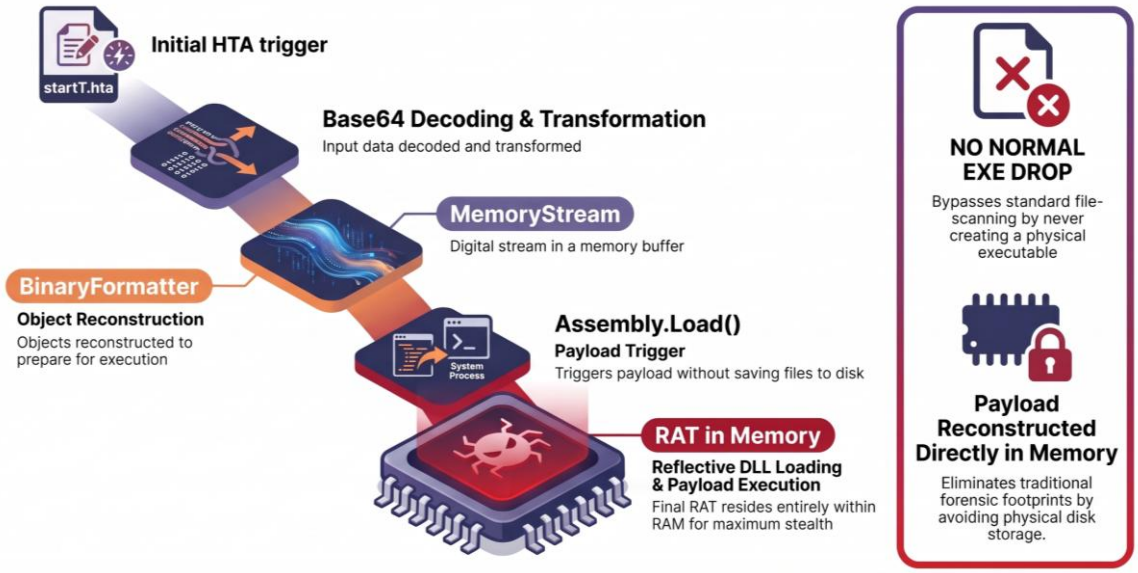

Figure: .NET Deserialization and Fileless Execution

The resulting payload does not need to be written as a conventional executable file before execution.

RAT Activation in Memory

The final payload: `Ioluegnt.dll` is reflectively loaded into the `mshta.exe` process.

The malware uses .NET reflection functionality including:

- `System.Reflection.Assembly.Load`
- `Activator.CreateInstance`

The RAT then begins execution from memory. This technique reduces traditional disk-based detection opportunities and makes **process-memory and behavioral telemetry** particularly important.

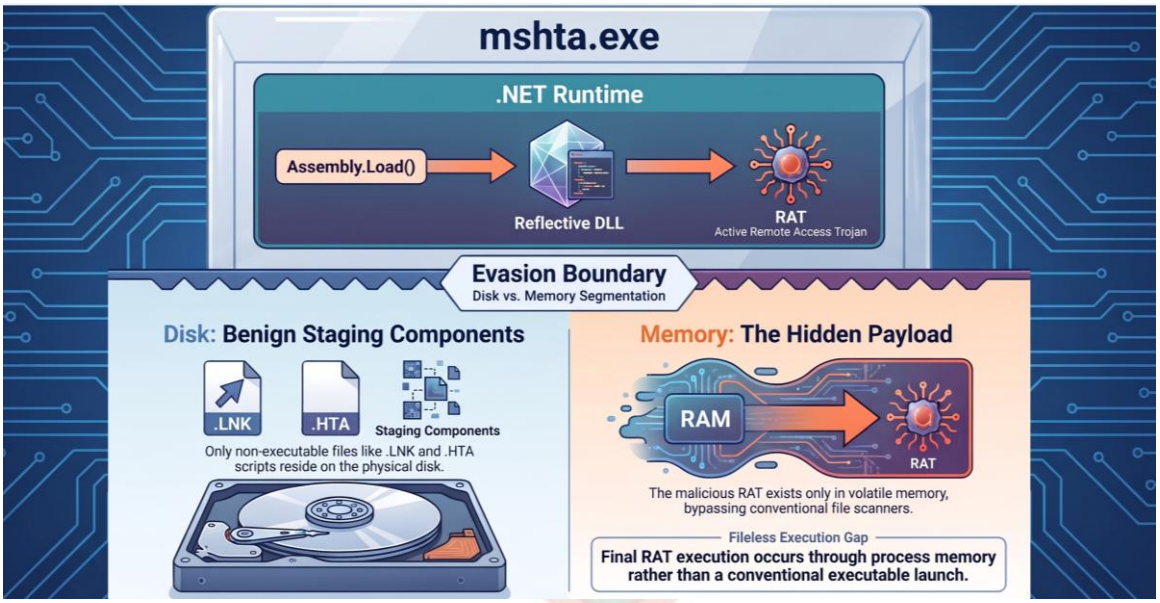

Figure: Fileless RAT Execution Inside MSHTA

RAT Capabilities

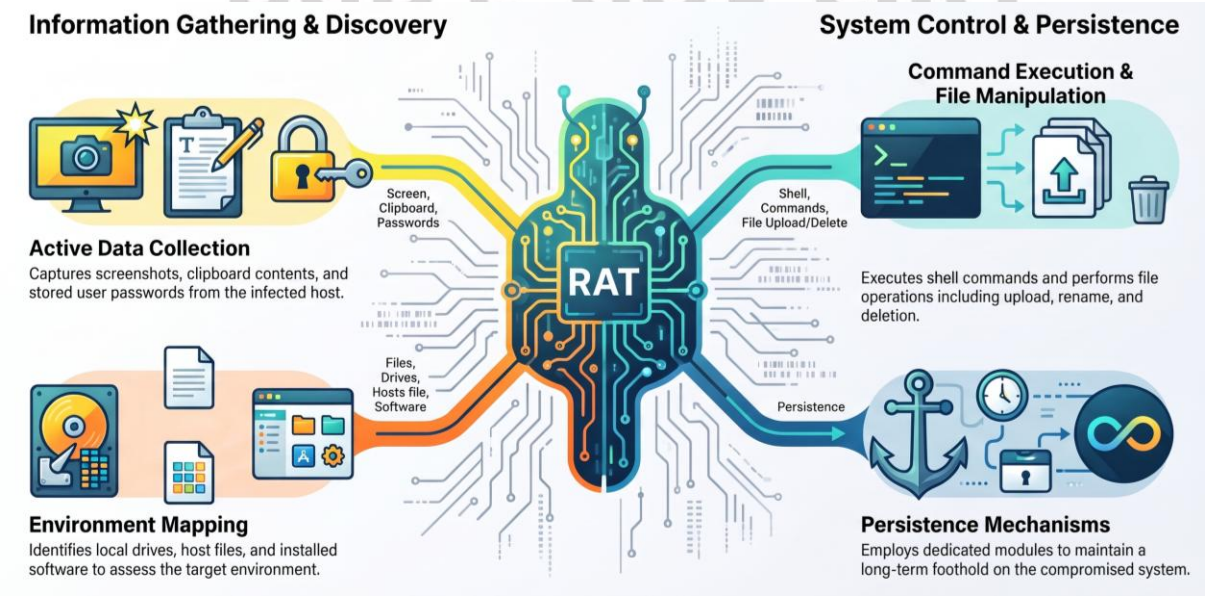

Figure: RAT Capability Architecture

The observed RAT provides extensive remote-control functionality.

Capability	Function
SystemInformation	Collect system information
Software	Enumerate installed software
RD	Capture screenshots
Passwords	Collect passwords
GetCPText	Collect clipboard data
GetHostsFile	Obtain hosts file
ListFiles	Enumerate files
ListDrives	Enumerate drives
Run	Execute files
Execute	Execute commands
Shell	Open shell session
fileupload	Upload files
Rmfile	Delete files
rnfile	Rename files
Mkdir	Create directories
rmdir	Remove directories
addSys	Establish persistence

These capabilities provide the operator with both **surveillance and interactive remote-control functionality.**

Figure: RAT Data Collection and Exfiltration (The Path of Stolen Data)

Command-and-Control and Data Exfiltration

The RAT communicates with attacker infrastructure using encrypted TCP communication.

Observed infrastructure:
Domain: `dns[.]educationportals[.]biz`
IP: `45[.]61[.]157[.]22`
Port: `5863`

The malware uses AES encryption for communications. A hardcoded cryptographic key was also identified in the analyzed sample: `NMXIKS09?:709,!~lnsYUS`. The use of an uncommon outbound port combined with encrypted traffic provides useful network-level detection opportunities.

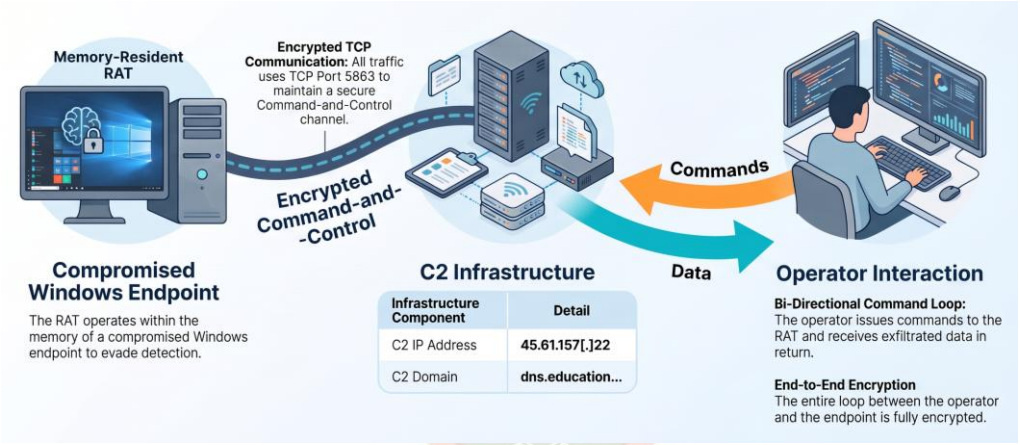

Figure: RAT Command-and-Control Architecture

MITRE ATT&CK Mapping

Technique	ID	Observed Activity
Phishing	T1566	Spear-phishing delivery
User Execution	T1204.001	Malicious LNK execution
Mshta	T1218.005	HTA execution
Ingress Tool Transfer	T1105	Remote payload retrieval
Command Shell	T1059.003	BAT execution
Registry Run Keys	T1547.001	Persistence
.NET Deserialization	T1218 / related execution behavior	BinaryFormatter abuse
Reflective Loading	T1620	In-memory DLL loading
File and Directory Discovery	T1083	File enumeration
System Information Discovery	T1082	Host reconnaissance
Clipboard Data	T1115	Clipboard collection
Screen Capture	T1113	Screenshot collection
Credentials	T1555 / related	Password collection
Command Execution	T1059	Remote command execution
Exfiltration Over C2	T1041	Data transmission through C2

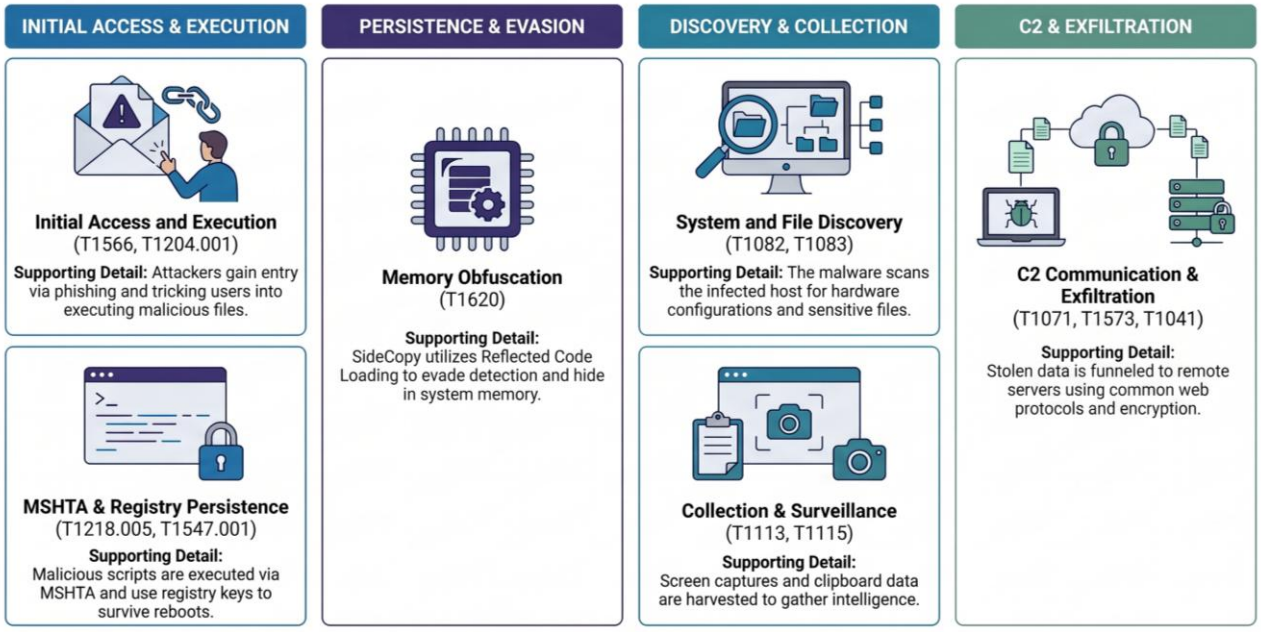

Figure: SideCopy Attack Lifecycle / MITRE ATT&CK Mapping

Indicators of Compromise (IoC)

File Hashes

Type	SHA-256
ZIP	0647336477bdd277450b0c36f104f3f82da721e98d56b67bbeeec05cc48f2d1c
LNK	0e0b77f79fe5d06f11de2959559379e94d62512e073c267b481a58d19d265240
HTA	34c20f5abc04375822f3f68e3e9915c7e388e8adb1ade597672920d53f2c067c
DLL	ac340859805220f97f98b299b32d82b1ccbb2c04c8c09516f3937feda937af80
BAT	8435ec938ca225c3131a624715832845ad9adc5faa93488486bc19c094b4d3ec
HTA	a5e36cf05bcc9ac4e9ebf2a13f95aef8e3847086fe7340c36d6aba6616ae1174
DLL	cceee5c983360842351ffdb8979676fd2fccd4e4c387ac77e4506291d8083c5c

Network Indicators

IP Address: 45[.]61[.]157[.]22
Port: TCP/5863
URL:

- dns[.]educationportals[.]biz
- hxxps[://]docsportal[.]in/public/reps/com/com[.]hta
- hxxp[://]docsportal[.]in/public/reps/com/com[.]hta
- hxxps[://]docsportal[.]in/public/reps/com/161[.]php

Detection and Monitoring Recommendations

Endpoint

Monitor:

- `*.lnk` → `mshta.exe`
- `mshta.exe` → network connection
- `mshta.exe` → PowerShell
- `mshta.exe` → child process
- `mshta.exe` → unusual DLL/module loading

Monitor creation or modification of:

`HKCU\Software\Microsoft\Windows\CurrentVersion\Run`

Network

Monitor:

- outbound TCP/5863;
- connections to known malicious domains/IPs;
- `mshta.exe` making external connections;
- unusual encrypted traffic from user workstations;
- HTTP/HTTPS requests for `.hta`, `.php` or suspicious document resources

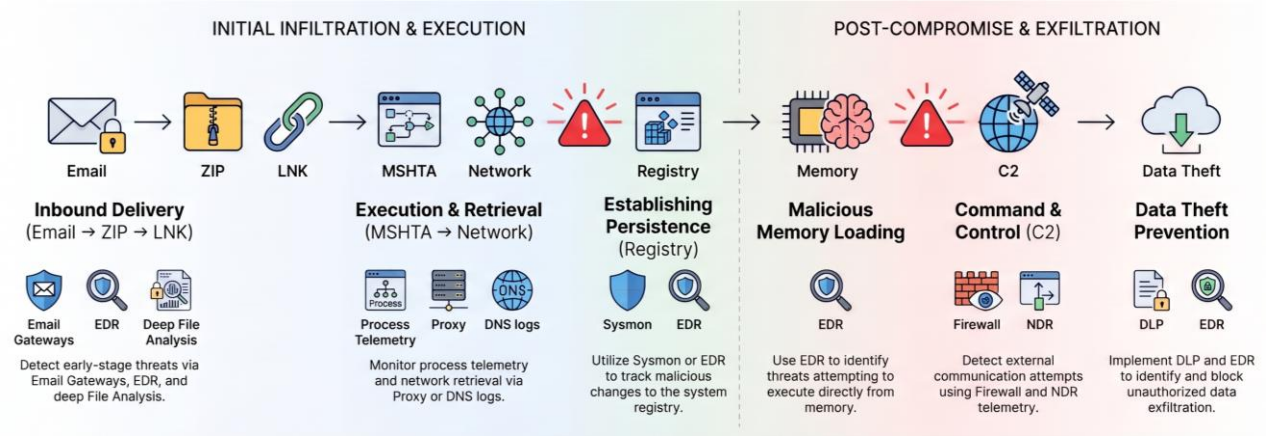

Figure: Detection Opportunities Across the Attack Chain

Behavioral Correlation

Multiple events occurring within a short period should generate a high-priority alert

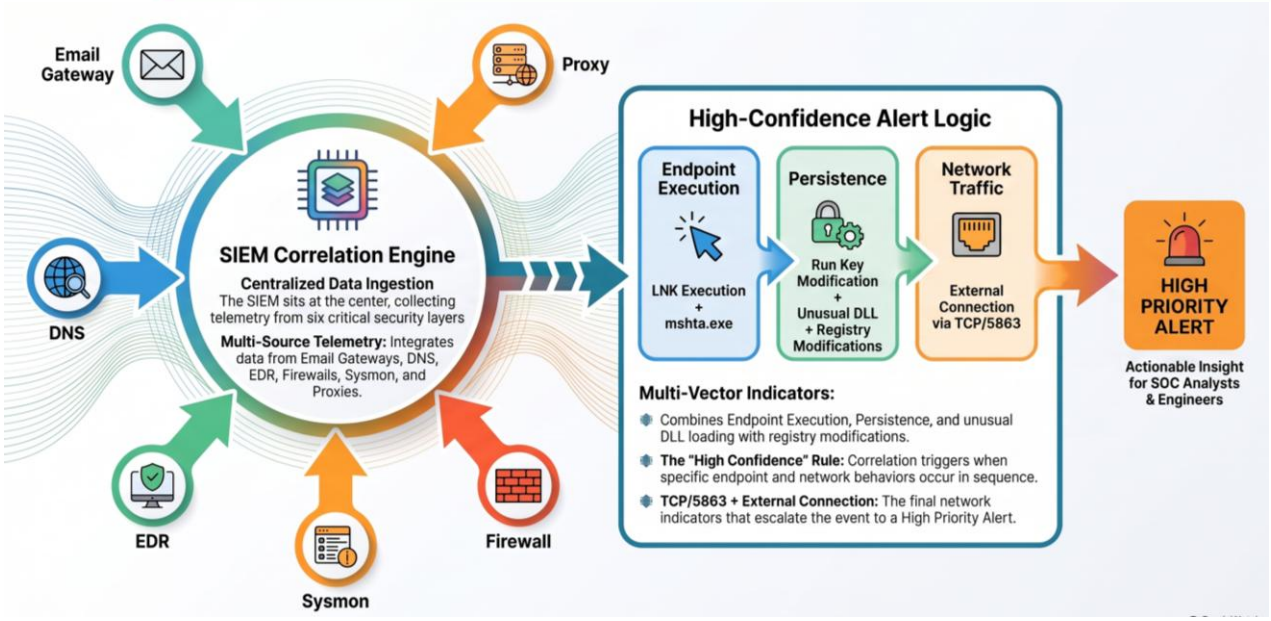

Figure: SIEM Correlation Model (Turning Raw Logs into High-Confidence Alerts)

Recommended Actions

Organizations should prioritize the following activities:

- Block or restrict mshta.exe where it is not operationally required.
- Restrict execution of `.LNK`, `.HTA` and other high-risk attachment types from email.
- Deploy behavioral EDR monitoring for `mshta.exe` and unusual child processes.
- Monitor Registry Run Key modifications.
- Apply application allow-listing where feasible.
- Implement outbound network filtering and restrict unnecessary non-standard ports.
- Inspect suspicious ZIP/LNK attachments before execution.
- Enable centralized Windows process, PowerShell, Sysmon and network logging.
- Hunt for the provided IOCs across DNS, proxy, firewall, EDR and SIEM infrastructure.
- Rotate credentials immediately if password theft or RAT activity is confirmed.

Bangladesh-Specific Considerations

Given the use of targeted spear-phishing against government and strategic personnel, organizations in Bangladesh should pay particular attention to:

- government email systems;
- ministries and government agencies;
- defense and security-related personnel;
- critical infrastructure;
- academic and research institutions;
- telecommunications;

- financial institutions;
- organizations handling sensitive government information.

Users should treat unexpected committee, meeting, recruitment, policy, administrative or official-document themed ZIP/DOCX files as potentially suspicious, particularly when the archive contains an LNK file.

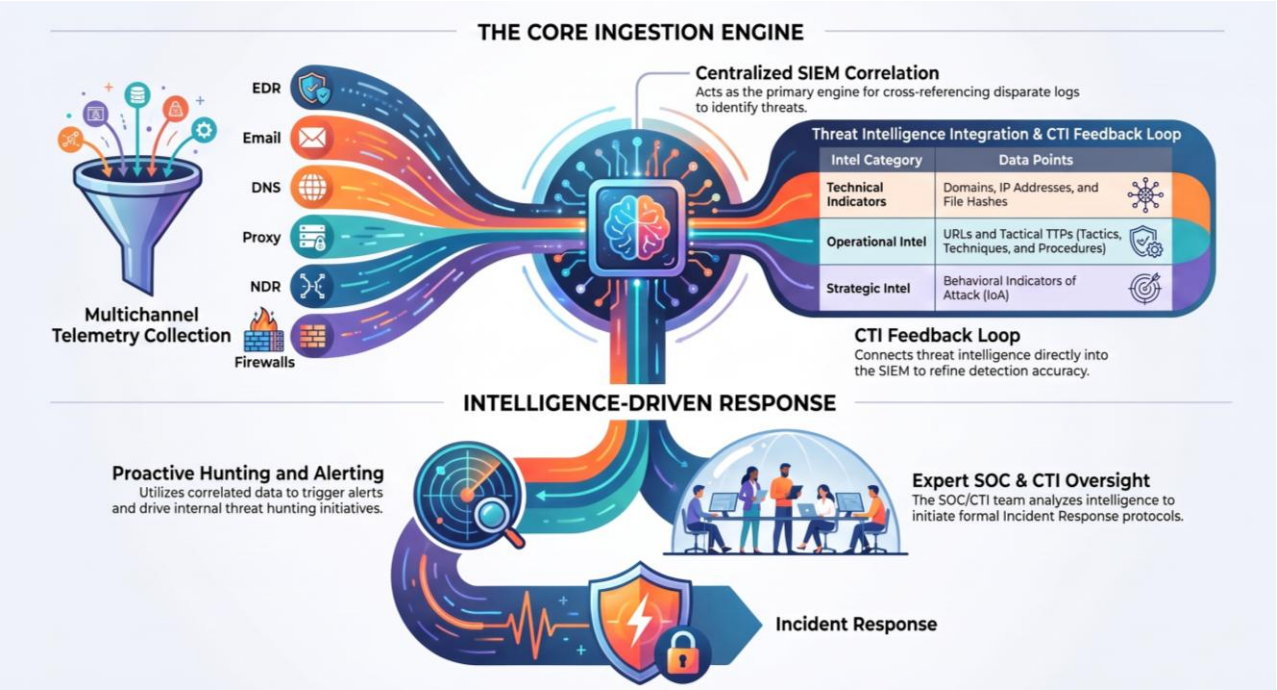

Figure: Detection Architecture (From Ingestion to Incident Response)

The observed SideCopy campaign demonstrates a multi-stage Windows intrusion chain combining spear-phishing, malicious LNK files, MSHTA execution, Registry persistence, .NET deserialization, reflective DLL loading and encrypted RAT communications.

The most important defensive focus should therefore move beyond traditional malware signatures toward behavioral detection and attack-chain correlation.

References

- https://www.trellix.com/blogs/research/sidecopy-threat-intel-mshta-execution-rat-deployment/
- https://learn.microsoft.com/en-us/dotnet/api/system.runtime.serialization.formatters.binary.binaryformatter?view=net-10.0
- https://www.seqrite.com/blog/umbrella-of-pakistani-threats-converging-tactics-of-cyber-operations-targeting-india/