

Google Sheets-Powered RAT Targets Diplomatic Organizations

securonix.com/blog/sheetcreep-evolved-google-sheets-rat

June 11, 2026



Blog

Analyzing SHEET#CREEP: SHEETCREEP is up again with different config obfuscation

Threat Intelligence, Threat Research, Threat Security

Analyzing SHEET#CREEP: SHEETCREEP is up again with different config obfuscation

By Securonix Threat Research: Shikha Sangwan, Akshay Gaikwad, Aaron Beardslee

tldr:

The Securonix Threat Research team has identified an ongoing espionage campaign, tracked as SHEETCREEP, where threat actors deliver a C# remote access trojan through a diplomatic-themed ISO phishing lure. Building upon the initial discovery and excellent research of the SHEETCREEP malware family by Zscaler's ThreatLabz, we observed that the RAT abuses the Google Sheets API as its command-and-control (C2) channel authenticating via an embedded GCP service account private key and using individual spreadsheet tabs per victim for bidirectional communication. Our team successfully extracted the embedded credentials, authenticated to the live C2 spreadsheet, and identified 91 active victim tabs including a high-confidence target located in Pakistan.



Campaign Evolution:

The SHEETCREEP malware family was initially documented by Zscaler ThreatLabz in January 2026. However, our latest analysis reveals that the campaign shows signs of being a continuation and evolution of previous threat activity. The threat actors have actively hardened their tooling to bypass traditional security defenses and evade detection. Specifically, this is a new, evolved variant where the C2 configuration strings, which were previously stored in plaintext, are now heavily obfuscated using an XOR-based encryption routine (with the key "discrete"). This shift

highlights active development by the threat actors to complicate static analysis and indicates they are adapting their tactics in response to public threat intelligence reporting.

What makes this campaign particularly interesting is not just the technique, it's the telemetry. By extracting the hardcoded GCP service account credentials from the RAT binary and authenticating to the live C2 spreadsheet, we were able to map out 91 active victim tabs, identify sandbox environments, research labs, and critically, isolate a high-confidence target with physical hardware in Islamabad, Pakistan.

Key Takeaways:

1. Diplomatic-themed phishing lure: The campaign uses an ISO file themed around the "UAE-India Strategic Partnership Week" to lure targets in the Indian diplomatic and foreign affairs space into executing a malicious LNK shortcut.
2. Multi-stage C# infection chain: The LNK triggers a C# dropper that extracts a bait PDF, drops the RAT payload into the Windows Vault directory, and establishes persistence through a scheduled task, before melting (self-deleting) to remove forensic traces.
3. Google Sheets API as C2: The RAT authenticates to the Google Sheets API using a hardcoded GCP service account (sheet5@sheet5-495707.iam.gserviceaccount.com) and an embedded RSA-2048 private key. Each victim is assigned a dedicated spreadsheet tab identified by a unique hash of username-hostname. Commands are written to Column A, and outputs are returned in Column B, all Base64-encoded.
4. XOR-based config obfuscation: C2 configuration strings (spreadsheet ID, service account email) are obfuscated using a simple XOR cipher with the key "discrete" and are decrypted at runtime via a function named JIT().
5. Anti-analysis and counter-forensics: The threat actor actively profiles victim machines by running commands to inspect open window titles, installed applications, and desktop files. When analyst tools (dnSpy, Wireshark, Network Monitor) are detected, the operator executes Restart-Computer -Force to break the debugging session.
6. Persistence through Scheduled Tasks: The malware creates a scheduled task named WindowsVaultSyncService with a deliberately misleading description ("Windows Edge Core Update Task Machine Discord Update") set to run on every logon with no execution time limit.

Initial Infection

The infection chain begins when the user opens the ISO file UAE-India_Strategic_Partnership_Week.iso. When mounted, the ISO presents the user with a single LNK shortcut file that masquerades as a legitimate document.

The LNK file UAE-India_Strategic_Partnership-Week.lnk uses the PDF document icon to deceive the victim into believing they are opening a PDF. Unfortunately, this is a common and effective technique. When the user double-clicks the LNK, it silently executes:

```
C:\Windows\System32\cmd.exe /c start "" "Document_11052026-03578240540350-93.exe"
```

This spawns the C# dropper binary packaged alongside the LNK inside the ISO.

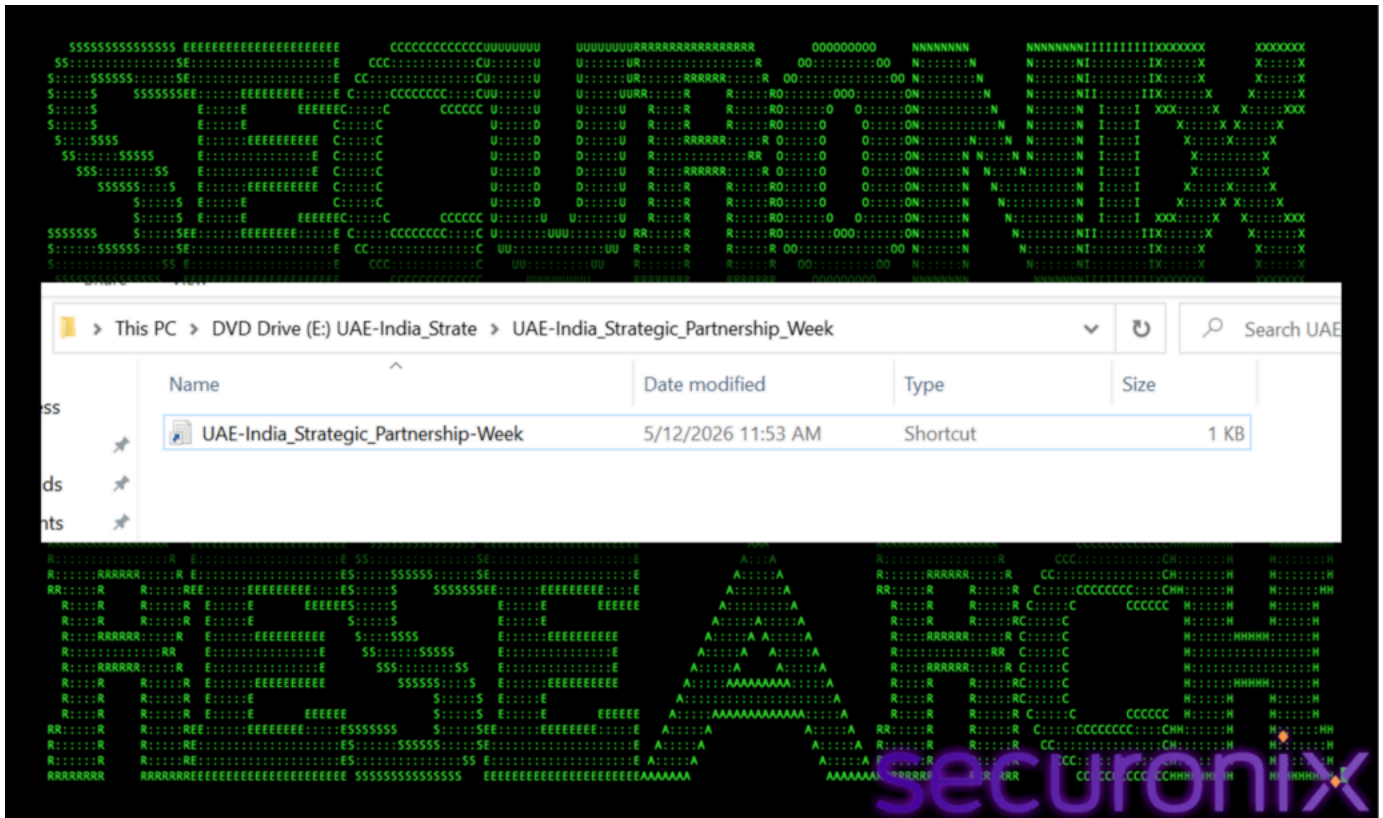


Figure 1 UAE-India_Strategic_Partnership-Week style themed iso malware

Dropper Analysis: CoreDeployment

The dropper binary Document_11052026-03578240540350-93.exe is a compact C# .NET executable with a single class named CoreDeployment. Despite its small footprint, the dropper performs several critical operations in rapid succession

The dropper contains two embedded .NET manifest resources:

InternalBait : The decoy PDF document, extracted to %TEMP%\<GUID>.pdf and immediately spawned using Process.Start() to display a legitimate-looking document to the victim.

InternalExe : The actual RAT payload, extracted to %LOCALAPPDATA%\Microsoft\Vault\vaultsvc.exe.

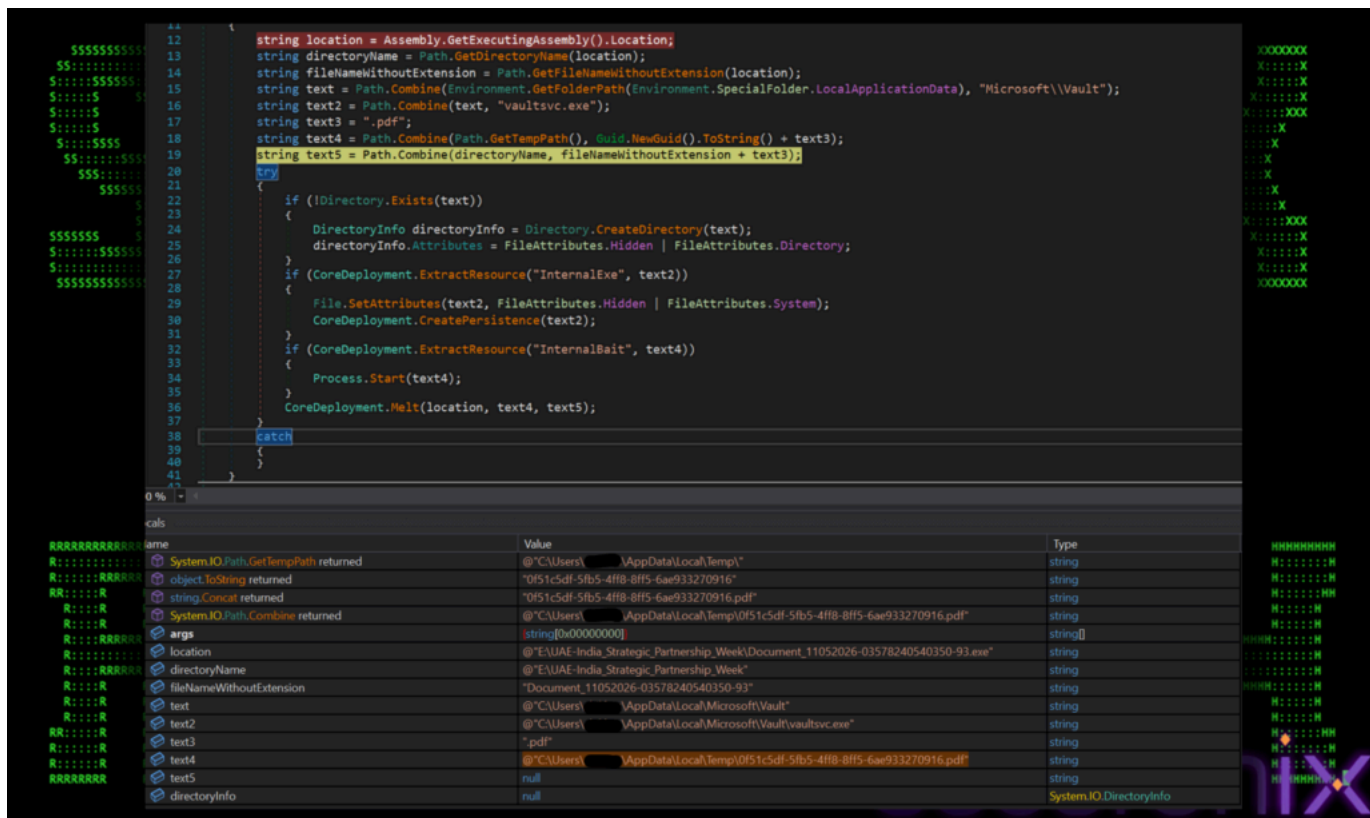


Figure 2 Resource Extraction

The choice of drop path is deliberate. The %LOCALAPPDATA%\Microsoft\Vault\ directory is a legitimate Windows Credential Vault path, allowing the malware binary to blend in with legitimate system files and evade casual directory inspection.

Once the RAT is written to disk, the dropper immediately sets the file attributes to Hidden | System:

`File.SetAttributes(text2, FileAttributes.Hidden | FileAttributes.System);`

Persistence via COM-based Scheduled Task

Rather than using the schtasks.exe command line utility (which would be easily logged), the dropper creates persistence programmatically through the Task Scheduler COM API (Schedule.Service). The relevant decompiled code is shown below:

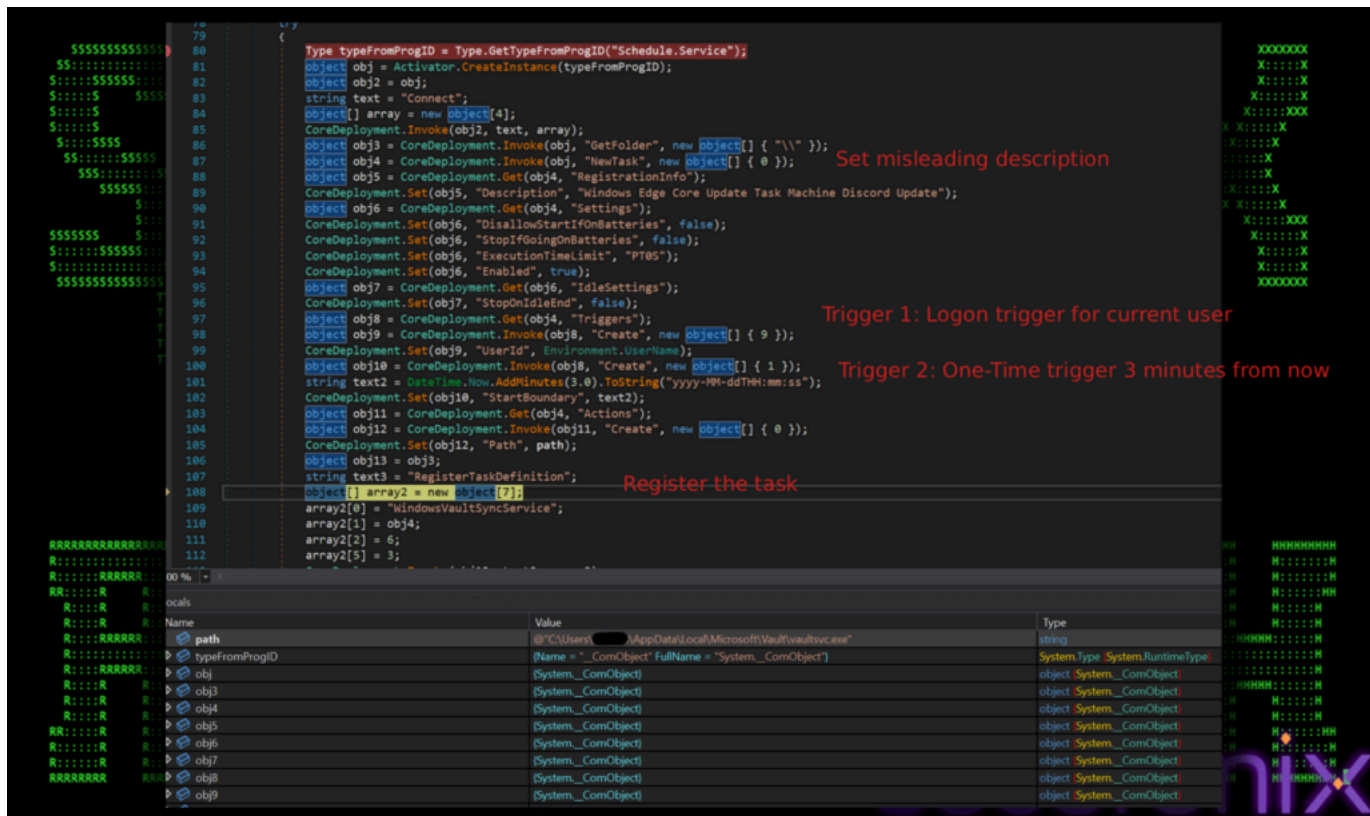


Figure 3 Persistence via COM-based scheduled task

The task is registered with the name `WindowsVaultSyncService` and the deliberately confusing description "Windows Edge Core Update Task Machine Discord Update" mixing "Edge," "Windows," and "Discord" terminology to appear benign during manual inspection. The execution time limit is set to `PT0S` (unlimited), ensuring the RAT runs indefinitely once triggered.

Self-deletion (Melt)

After completing its payload drop and persistence setup, the dropper executes a self-deletion routine:

C2 infrastructure: Google Sheets API

This is where the campaign gets particularly interesting. The RAT does not use a traditional C2 server. Instead, it leverages the Google Sheets API v4 as a bidirectional command-and-control channel. All C2 traffic flows through `sheets.googleapis.com` over HTTPS, making it virtually indistinguishable from normal Google Workspace traffic at the network level.

The C2 configuration strings are not stored in plaintext, they are encrypted using a simple XOR cipher and decrypted at runtime by a function named `JIT()`:

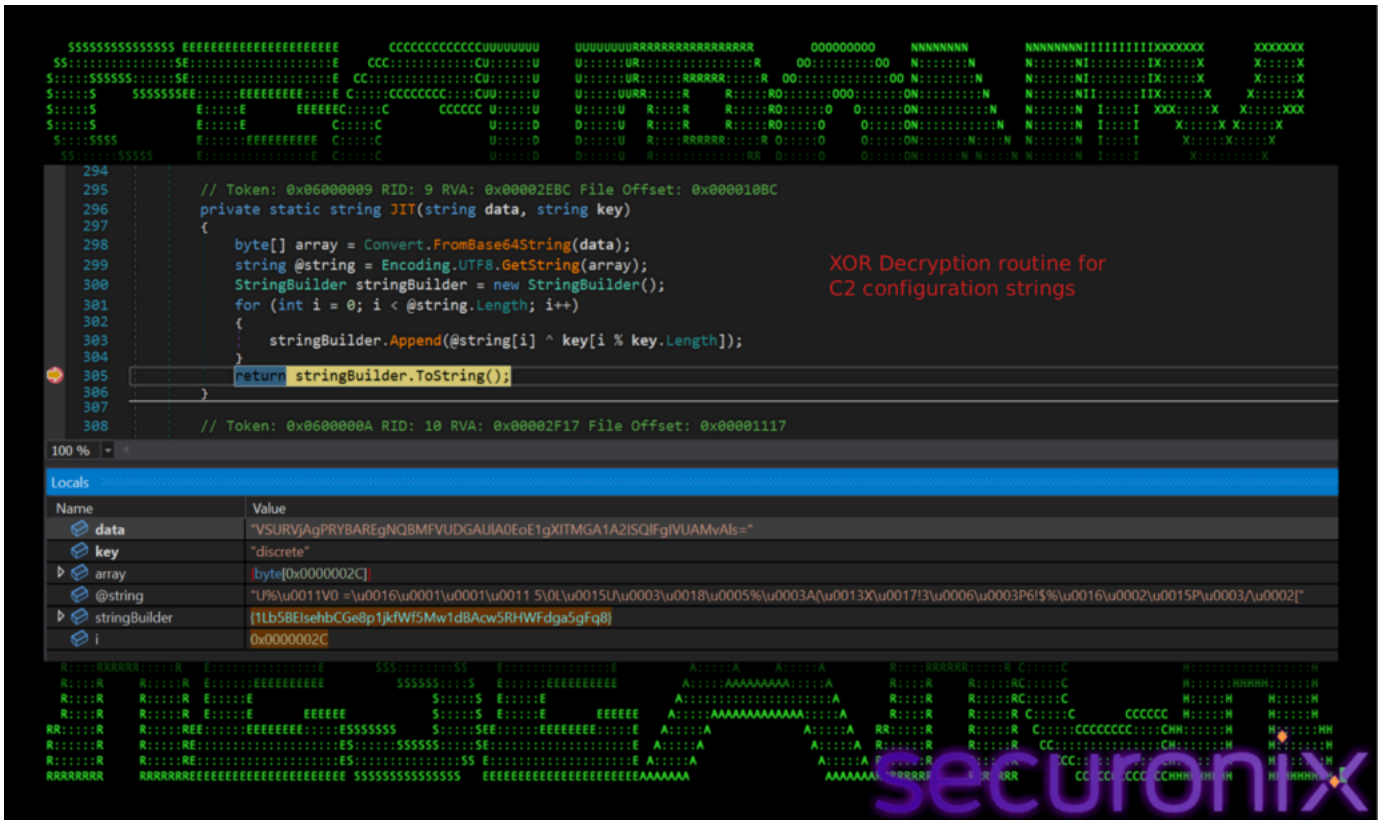


Figure 7 XOR decryption routine for C2 configuration strings

The XOR key used is "discrete". Using this key, we decrypted the following configuration values:

Encrypted (Base64)	Decrypted Value
VSURVjAgPRYBAREgNQBMFVUDGAUIA0E0E1gXITMGA1A2ISQIFgIVUAMvAls=	1Lb5BElsehbCGe8p1jKfWf5Mw1dBAcw5RHWfDga5gFq8
FwEWBgZQNBYMDBYXR0hAXFFeQ1RcDBUISg4ABgATHQYBCBAAHRAaEUoKHA4=	sheet5@sheet5-495707.iam.gserviceaccount.com

This represents a notable evolution from prior SHEETCREEP variants where these values were stored as plaintext strings, indicating the threat actors are actively hardening their tooling against static analysis.

OAuth2 authentication via JWT

The RAT authenticates to the Google Sheets API by constructing and signing a JSON Web Token (JWT) using an embedded RSA-2048 PKCS#8 private key:

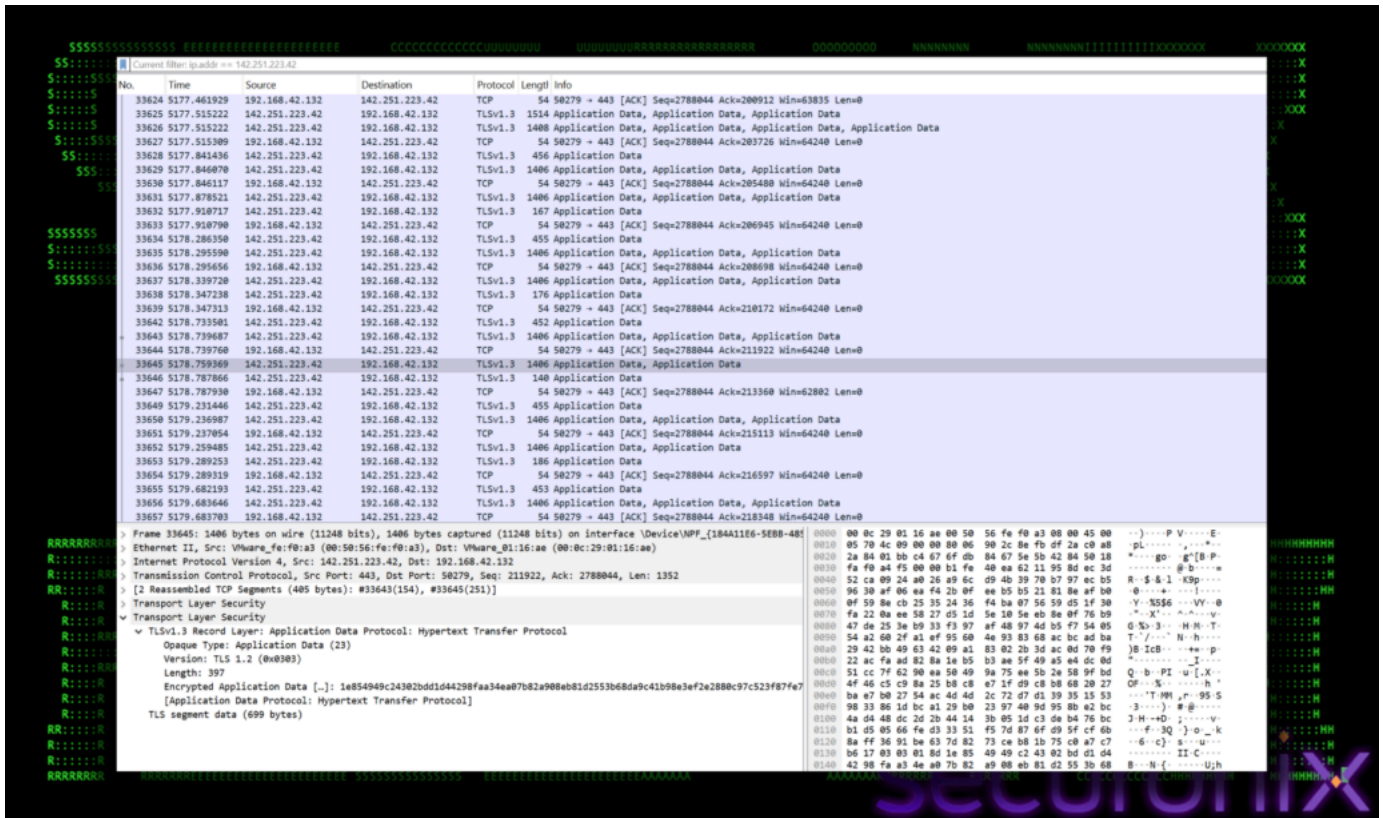


Figure 9 C2 beaconing to 142.251.223.42

By extracting the embedded GCP service account credentials and authenticating to the live C2 spreadsheet, we were able to enumerate all victim tabs and analyze the telemetry data. During initial triage, the spreadsheet contained 79 victim tabs. By the time of our live C2 interrogation on May 26, 2026, this had grown to 91 active tabs, including our own analysis VM checking in as *username*-DESKTOP-R32QGHR-9CDE.

Victim categorization

Category	Count	Key Indicators
Automated Sandboxes	28	Randomized usernames (FD1HVy, UZ6YMRX6LFot), synchronized check-in bursts (7 VMs within 17 seconds), legacy hardware spoofing (Fujitsu, Toshiba)
Security Researcher Labs	17	Analysis tools in process list (wireshark.exe, x64dbg.exe, dnspy.exe, netmon.exe). Includes Gigabyte lab clusters with hostnames IDKMAN / BFFC-PC and a Flare VM (BoJack)
Potential True Targets	17	Physical hardware, no sandbox indicators. 1 system confirmed as high-confidence target
Empty Sheets	16	Initialized but no data — likely immediate AV block or network teardown
Default Sheet	1	Default Sheet1 from spreadsheet creation
Our Analysis VM	1	*username*-DESKTOP-R32QGHR-9CDE
Total	91	

The heavy concentration of systems in UTC-08:00 and UTC+01:00 checking in within short bursts strongly suggests that public uploads to file-scanning portals (Virus Total, Hybrid Analysis) triggered automatic dynamic execution across cloud-based multi-sandbox systems.

Figure below shows the decoded systeminfo output exfiltrated from this victim's C2 tab. The data was Base64-encoded in cell B3 of the victim's sheet and decoded during our investigation.

Figure 11 In-process PowerShell command execution

This is a significant evasion technique. Because the PowerShell execution occurs entirely within the vaultsvc.exe process memory space, there is no powershell.exe child process for EDR solutions to detect. During our dynamic analysis monitoring window, the only child process observed under the RAT was a hidden conhost.exe instance.

Victimology and Attribution:

Based on the lure theming (“UAE-India Strategic Partnership Week”), the targeting of Indian diplomatic interests, and the identification of a high-confidence target in Pakistan (a common staging ground for APT36 operations), we assess with moderate confidence that this campaign is attributable to APT36 (Transparent Tribe), a Pakistan-aligned threat group that has historically targeted Indian government and military entities.

This assessment is further supported by:

- The use of ISO-based delivery mechanisms, consistent with prior APT36 campaigns.
- The Google Sheets C2 technique, which was previously documented in Zscaler’s SHEETCREEP reporting attributed to APT36.
- The geopolitical targeting profile (India-UAE diplomatic relations).

Conclusion

The SHEET#CREEP campaign demonstrates a sophisticated abuse of trusted cloud infrastructure for command-and-control operations. By routing all C2 traffic through the Google Sheets API, the threat actors ensure that network-level detection is extremely difficult, the traffic is legitimate TLS to Google’s IP space, authenticated with valid GCP service account credentials, and structurally identical to normal Sheets API calls.

The multi-layered evasion approach is notable: COM-based scheduled task creation to avoid command-line logging, in-process PowerShell execution to avoid spawning detectable child processes, XOR-encrypted configuration strings, hidden file attributes in a legitimate Windows directory, and active analyst detection and countermeasures. Combined, these techniques present a significant challenge for both automated detection systems and manual incident responders.

Another notable aspect of this campaign is the evolution from previous SHEETCREEP variants. The introduction of XOR-based configuration obfuscation indicates that the threat actors are actively developing and hardening their tooling in response to public threat intelligence reporting. Defenders should expect continued iteration on this approach.

The overall impact of the SHEET#CREEP campaign enables persistent, stealthy access to victim machines with full PowerShell execution capabilities. With the C2 channel active, the threat actors can execute arbitrary commands, exfiltrate data, enumerate networks, and move laterally, all while blending into normal cloud API traffic.

Securonix Recommendations

- Avoid downloading files or attachments from external sources, especially if the source was unsolicited. Malicious payloads from phishing emails can be delivered as ISO images containing LNK shortcuts that masquerade as legitimate documents.
- Monitor for the creation of scheduled tasks with unusual or misleading descriptions, particularly those registered via COM interop (Schedule.Service) rather than schtasks.exe. The task name WindowsVaultSyncService and description “Windows Edge Core Update Task Machine Discord Update” are specific indicators for this campaign.
- Monitor the %LOCALAPPDATA%\Microsoft\Vault\ directory for unexpected executables. The Windows Credential Vault directory should not contain .exe files. Look for hidden/system-attributed binaries in this path.
- Be aware that Google Sheets API traffic to sheets.googleapis.com may be used as a C2 channel. While blocking Google Sheets outright may not be practical, anomalous patterns — such as a non-browser process making sustained HTTPS connections to Google API endpoints — should be flagged.
- Since the RAT uses in-process PowerShell execution via System.Management.Automation, traditional PowerShell logging (ScriptBlock, Module logging) may not capture the activity. We strongly recommend deploying robust endpoint logging capabilities including Sysmon and AMSI-based detections for .NET-hosted PowerShell.
- Securonix customers can scan endpoints using the Securonix hunting queries below.

MITRE ATT&CK Matrix

Tactics	Techniques
---------	------------

Initial Access	T1566.001: Phishing: Spearphishing Attachment
Execution	T1204.002: User Execution: Malicious File T1059.001: Command and Scripting Interpreter: PowerShell T1053.005: Scheduled Task/Job: Scheduled Task
Persistence	T1053.005: Scheduled Task/Job: Scheduled Task
Defense Evasion	T1564.001: Hide Artifacts: Hidden Files and Directories T1036.005: Masquerading: Match Legitimate Name or Location T1070.004: Indicator Removal: File Deletion T1027.013: Obfuscated Files or Information: Encrypted/Encoded File
Discovery	T1082: System Information Discovery T1057: Process Discovery T1083: File and Directory Discovery
Collection	T1005: File and Directory Discovery
Command and Control	T1102.002: Web Service: Bidirectional Communication T1573.002: Encrypted Channel: Asymmetric Cryptography
Exfiltration	T1041: Exfiltration Over C2 Channel

Relevant Securonix detections

- WOS-212-ER
- EDR-ALL-73-ER
- CEDR-ALL-829-ERR

Relevant hunting queries

(remove square brackets “[]” for IP addresses or URLs)

- index = activity AND rg_functionality = “Next Generation Firewall” AND destinationaddress = “142.251.223[.]42 “

- index = activity AND rg_functionality = "Next Generation Firewall" AND destinationhostname CONTAINS "sheets.googleapis[.com]" AND sourceprocessname NOT IN ("chrome.exe", "msedge.exe", "firefox.exe", "excel.exe")
- index = activity AND rg_functionality = "Firewall" AND destinationhostname CONTAINS "sheets.googleapis[.com]" AND sourceprocessname NOT IN ("chrome.exe", "msedge.exe", "firefox.exe", "excel.exe")
- index = activity AND rg_functionality = "Web Proxy" AND destinationhostname CONTAINS "sheets.googleapis[.com]" AND sourceprocessname NOT IN ("chrome.exe", "msedge.exe", "firefox.exe", "excel.exe")
- index = activity AND rg_functionality = "Web Application Firewall" AND destinationhostname CONTAINS "sheets.googleapis[.com]" AND sourceprocessname NOT IN ("chrome.exe", "msedge.exe", "firefox.exe", "excel.exe")
- index = activity AND rg_functionality = "Endpoint Management Systems" AND deviceaction = "File created" AND filename IN ("\\Microsoft\\Vault\\vaultsvc.exe")
- index = activity AND rg_functionality = "Endpoint Management Systems" AND (deviceaction = "Process Create" OR deviceaction = "Process Create (rule: ProcessCreate)" OR deviceaction = "ProcessRollup2" OR deviceaction = "Procstart" OR deviceaction = "Process" OR deviceaction = "Trace Executed Process") AND customstring49 ENDS WITH "\\Microsoft\\Vault\\vaultsvc.exe"

C2 and infrastructure

Type	Indicator
Auth Endpoint	oauth2.googleapis[.]com
C2 Endpoint	sheets.googleapis[.]com
C2 IP (observed)	142.251.223.42 (Google LLC, AS15169)
GCP Project ID	sheet5-495707
Service Account	sheet5@sheet5-495707.iam.gserviceaccount[.]com
C2 Spreadsheet ID	1Lb5BEIsehbCGe8p1jkfWf5Mw1dBAcw5RHWfDga5gFq8
Scheduled Task	WindowsVaultSyncService
Mutex	Global\\WinSync_<username>-<hostname>-<4char-hash>

Analyzed files/hashes

File Name	SHA256
UAE-India_Strategic_Partnership_Week.iso	1ba67bb1cfad42446880cca53cbd05fe66d7514b2bb139b48e5c63adff14be7b
Document_11052026-03578240540350-93.exe (Dropper)	2cc7c2d8653c98e5bac32caf5e45b861efb4bb87df3b3f96285edb475e75bba
vaultsvc.exe (RAT)	62d62950ff7a0e43550a5d0ba55d32d5083b9de5538e0f012e406b6d951e16aa

References:

1. Zscaler ThreatLabz – SHEETCREEP: C# RAT using Google Sheets as C2

<https://www.zscaler.com/blogs/security-research/apt-attacks-target-indian-government-using-sheetcreep-firepower-and>

2. MITRE ATT&CK – Web Service: Bidirectional Communication (T1102.002)

<https://attack.mitre.org/techniques/T1102/002/>

[Previous Article](#)



[Threat Research](#)

[CHAMELEON#NET: A Deep Dive into Multi-Stage .NET Malware Leveraging Reflective Loading and Custom Decryption for Stealthy Operations](#)

[Learn More](#)



[Threat Research](#)

[Analyzing SERPENTINE#CLOUD: Threat Actors Abuse Cloudflare Tunnels to Infect Systems with Stealthy Python-Based Malware](#)

[Learn More](#)

securonix | **THREAT LABS**

INTELLIGENCE INSIGHTS



November 2023

[Threat Research](#)

[Securonix Threat Labs Monthly Intelligence Insights – November 2023](#)

[Learn More](#)

securonix | **THREAT LABS**

INTELLIGENCE INSIGHTS



October 2023

[Threat Research](#)

[Securonix Threat Labs Monthly Intelligence Insights – October 2023](#)

[Learn More](#)