INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-22 13:40 UTC

# Harvester APT Brings GoGra Backdoor to Linux, Hides C2 Inside Microsoft Outlook

**THREAT CAMPAIGN** | **HIGH** | CVSS 7.5

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0201 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Linux systems (servers and workstations), Microsoft Azure Active Directory, Microsoft Graph API, Microsoft Outlook, targets in telecommunications, government, and IT sectors in South Asia |
| **Published** | 2026-04-22T06:00:00 |
| **Discovery Source** | Rss |

## Executive Summary

The Harvester espionage group has extended its GoGra backdoor to Linux systems, targeting telecommunications, government, and IT organizations in South Asia. The malware hides command-and-control traffic inside Microsoft Outlook inboxes via the Microsoft Graph API, making it indistinguishable from legitimate cloud traffic to most security controls. Organizations running Linux infrastructure that rely on Microsoft 365 services face elevated risk of undetected long-term access and data exfiltration.

## Technical Analysis

Harvester's Linux GoGra variant communicates exclusively through Microsoft Graph API calls to a dedicated Outlook mailbox, receiving AES-encrypted commands from attacker-controlled emails and exfiltrating data via email replies. The malware shares hardcoded AES encryption keys with the Windows variant (CWE-798), uses a potentially weak or misapplied cryptographic implementation (CWE-327), and bypasses perimeter controls by abusing the legitimacy of Microsoft cloud services (CWE-693). No CVE has been assigned; the technique exploits legitimate API functionality rather than a platform vulnerability. MITRE ATT&CK techniques include T1102.002 (Bidirectional Communication via Web Service), T1071.001 (Web Protocols), T1573.001 (Symmetric Cryptography), T1041 (Exfiltration Over C2 Channel), T1105 (Ingress Tool Transfer), T1078.004 (Cloud Accounts), T1543.002 (Systemd Service persistence), T1547.013 (XDG Autostart), T1027 (Obfuscated Files), T1560 (Archive Collected Data), T1070.003 (Clear Command History), T1059 (Command and Scripting Interpreter), T1566.001 (Spearphishing Attachment), and T1204.002 (Malicious File execution). Cross-platform

codebase parity, identical function names and shared keys across Windows and Linux builds, indicates deliberate porting rather than opportunistic adaptation. No vendor patch exists; mitigation is detection and access control focused. CVSS scoring is not applicable; severity is assessed qualitatively based on attack scope, target criticality, and operational impact. Source: BleepingComputer, based on Symantec threat research.

## Action Checklist

1. Step 1: Containment, Audit Microsoft Graph API OAuth application registrations in your Azure AD / Entra ID tenant. Revoke or disable any unrecognized app registrations with Mail.Read, Mail.Send, or Mail.ReadWrite permissions. Restrict Graph API access to approved, inventoried applications only. Priority targets: Linux servers and workstations in telecom, government, and IT environments.
2. Step 2: Detection, Search endpoint logs on Linux hosts for unusual systemd service creation (T1543.002) or XDG autostart entries (T1547.013) associated with unknown binaries. Review Microsoft 365 Unified Audit Logs for OAuth app consent events, Graph API mail access from non-human identities, and Outlook mailbox activity by service principals not in your approved inventory. Prerequisite: Ensure Microsoft 365 Unified Audit Logs are enabled for all mailbox and application activity. If not enabled, enable immediately and allow 24 hours for historical log population before searching. Look for periodic, low-volume email sends and receives by app identities, consistent with polling C2 behavior. Correlate with EDR telemetry for process execution chains spawning from unknown binaries on Linux hosts.
3. Step 3: Eradication, Remove any identified GoGra binaries and associated persistence mechanisms (systemd unit files, XDG autostart entries). Revoke and rotate all OAuth tokens and client credentials associated with compromised or unrecognized Graph API app registrations. Reset credentials for any cloud accounts (T1078.004) accessed by the malware. Audit and clear command history on affected Linux hosts (attacker clears history per T1070.003, gaps in history logs are themselves an indicator).
4. Step 4: Recovery, After credential rotation and binary removal, monitor Graph API audit logs for resumed polling behavior. Confirm no new unauthorized app registrations appear in Entra ID. Validate systemd services and XDG autostart entries on affected Linux hosts against a known-good baseline. Monitor outbound HTTPS to Microsoft Graph API endpoints (graph.microsoft.com) from Linux hosts, legitimate use should be limited and attributable.
5. Step 5: Post-Incident, This campaign exposes three control gaps: insufficient OAuth app governance, absence of behavioral monitoring for Graph API abuse on Linux endpoints, and no baseline for legitimate vs. anomalous Graph API usage. Implement Conditional Access policies restricting Graph API app consent to admin-approved applications. Deploy audit logging for all OAuth consent events. Extend EDR coverage to Linux infrastructure if not already present. Review MITRE ATT&CK T1102.002 detection coverage in your SIEM.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |

| | |
|---|---|
| **Escalation Criteria** | Escalate to senior IR leadership and legal/compliance immediately if UAL evidence confirms Harvester accessed mailboxes containing personally identifiable information, government classified communications, or customer data subject to GDPR, sector-specific breach notification requirements (e.g., FCC for telecom), or contractual disclosure obligations — the long-dwell nature of this campaign means exfiltration volume may be significant before discovery. |
| **Recovery Notes** | After credential rotation and GoGra binary removal, maintain continuous monitoring of Entra ID AuditLogs and Microsoft 365 UAL for at least 90 days given Harvester's documented long-dwell TTPs — the group is known for re-entry after partial remediation using previously staged credentials. Validate every Linux host's systemd service inventory and XDG autostart entries against a clean baseline before returning systems to production, as GoGra may install multiple persistence mechanisms to survive partial cleanup. Confirm that all OAuth app registrations active in the tenant are accounted for in your application inventory before standing down the incident — any gap means a potential surviving foothold. |
| **Forensic Artifacts** | Microsoft 365 Unified Audit Log — MailItemsAccessed and Send operations by UserType:Application service principals, specifically periodic low-volume access events with sub-10-minute polling intervals that are the behavioral signature of GoGra's Graph API C2 channel (T1102.002) \| Entra ID AuditLogs — Add app role assignment to service principal and Consent to application events establishing when GoGra's OAuth app identity was first registered and granted Mail.Read/Mail.Send/Mail.ReadWrite permissions, providing the initial access timestamp (T1078.004) \| Linux systemd unit files in /etc/systemd/system/ and /usr/lib/systemd/system/ — specifically unit files created outside of package manager activity with ExecStart directives pointing to binaries in non-standard paths (/tmp, /var/tmp, user home directories), representing GoGra's persistence mechanism (T1543.002) \| XDG autostart .desktop files in ~/.config/autostart/ and /etc/xdg/autostart/ — entries created by GoGra for user-context persistence (T1547.013) that survive system reboots and may persist after systemd service removal if cleanup is incomplete \| Linux auditd logs (/var/log/audit/audit.log) — execve syscall records for the GoGra binary process, including child process spawning chains and outbound connect syscalls to Microsoft ASN 8075 (graph.microsoft.com), plus any evidence of bash_history truncation or deletion as T1070.003 anti-forensic activity |

### Per-Action IR Details

**Step 1: Containment — Audit Microsoft Graph API OAuth application registrations in your Azure AD / Entra ID tenant. Revoke or disable any unrecognized app registrations with Mail.Read, Mail.Send, or Mail.ReadWrite permissions. Restrict Graph API access to approved, inventoried applications only. Priority targets: Linux servers and workstations in telecom, government, and IT environments.**

**NIST Phase:** Containment

**Reference:** NIST 800-61r3 §3.3 — Containment Strategy

**Controls:** NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST AC-17 (Remote Access), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

**Compensating:** Run the following Azure CLI commands to enumerate all app registrations with mail-scope OAuth permissions: `az ad app list --all --query "[].{AppId:appId, DisplayName:displayName, ReplyUrls:replyUrls}" -o table` then cross-reference against `az ad sp list --all --query "[].{AppId:appId, DisplayName:displayName}"`. For each unfamiliar service principal, pull its OAuth2 permission grants via `az ad app permission list --id ` and look for `Mail.Read`, `Mail.Send`, or `Mail.ReadWrite` delegated or application permissions. Revoke with `az ad app permission delete --id --api `. A 2-person team can complete this audit in one session using the free Azure CLI; no SIEM required.

**Evidence:** Before revoking any tokens, snapshot the full OAuth app registration state: export all app registrations and their permission grants via `az ad app list --all -o json > app_inventory_snapshot.json`. Capture the Microsoft 365

Unified Audit Log entries for `Add app role assignment to service principal` and `Consent to application` operations — these record when GoGra's Graph API app identity was first registered and granted mail permissions, establishing the initial access timeline (MITRE T1078.004). Also preserve the `az ad app show --id ` output including `createdDateTime` and `signInAudience` fields for each suspicious registration before deletion.

**Step 2: Detection — Search endpoint logs on Linux hosts for unusual systemd service creation (T1543.002) or XDG autostart entries (T1547.013) associated with unknown binaries. Review Microsoft 365 Unified Audit Logs for OAuth app consent events, Graph API mail access from non-human identities, and Outlook mailbox activity by service principals not in your approved inventory. Look for periodic, low-volume email sends and receives by app identities — consistent with polling C2 behavior. Correlate with EDR telemetry for process execution chains spawning from unknown binaries on Linux hosts.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2 — Detection and Analysis

**Controls:** NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

**Compensating:** On each Linux host, run `systemctl list-units --type=service --state=enabled | grep -v 'loaded active'` to surface non-standard enabled services, then compare unit file paths against a known-good baseline — GoGra persistence via systemd would appear in `/etc/systemd/system/` or `/usr/lib/systemd/system/` with an unfamiliar binary path in the `ExecStart` directive. For XDG autostart, check `~/.config/autostart/*.desktop` and `/etc/xdg/autostart/*.desktop` for entries pointing to unknown executables. For Graph API polling detection without a SIEM, use the M365 Audit Log search in the Microsoft Purview Compliance Portal, filtering on `MailItemsAccessed` and `Send` operations by non-human `UserType` (value: `Application`), sorted by `CreationTime` to identify regular polling intervals (GoGra's C2 loop produces a distinctive rhythmic access pattern). Use the free `o365-extractor` or `hawk` PowerShell tool to bulk-pull UAL entries offline for timeline analysis.

**Evidence:** Collect the following before any remediation action: (1) `/etc/systemd/system/` and `/usr/lib/systemd/system/` — dump all unit files modified in the past 90 days via `find /etc/systemd/system /usr/lib/systemd/system -newer /etc/passwd -exec ls -la {} \;`. (2) `/var/log/auth.log` and `/var/log/syslog` on affected Linux hosts — filter for `systemd` entries referencing newly created service names. (3) Microsoft 365 Unified Audit Log — specifically `MailItemsAccessed` records with `AppId` values not in your approved inventory, timestamped to identify first-seen Graph API access (establishes C2 channel establishment date). (4) Process execution logs from auditd (`/var/log/audit/audit.log`) — filter on `SYSCALL` records for `execve` calls from the GoGra binary path to capture child process spawning. (5) Network connection records — run `ss -tnp` or `netstat -tnp` and check for established HTTPS connections to `graph.microsoft.com` (140.82.x.x or Microsoft ASN 8075) originating from unexpected Linux process PIDs.

**Step 3: Eradication — Remove any identified GoGra binaries and associated persistence mechanisms (systemd unit files, XDG autostart entries). Revoke and rotate all OAuth tokens and client credentials associated with compromised or unrecognized Graph API app registrations. Reset credentials for any cloud accounts (T1078.004) accessed by the malware. Audit and clear command history on affected Linux hosts (attacker clears history per T1070.003 — gaps in history logs are themselves an indicator).**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AC-2 (Account Management), CIS 5.3 (Disable Dormant Accounts), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

**Compensating:** Before deleting GoGra binaries, collect a SHA-256 hash (`sha256sum `) and file metadata (`stat `) for IOC submission and post-incident reporting. Use `systemctl stop && systemctl disable ` followed by `rm /etc/systemd/system/.service && systemctl daemon-reload` to fully remove the persistence mechanism. For OAuth token revocation without an enterprise identity platform, use Azure CLI: `az ad app credential delete --id --key-id ` to rotate client secrets, and `az ad app delete --id ` for confirmed-malicious registrations. Use

`revoke-AzureADUserAllRefreshToken` (PowerShell AzureAD module, free) to invalidate all active sessions for any cloud accounts the malware accessed. Document all command history gaps (`cat ~/.bash_history` — missing timestamps or truncated sequences) as forensic evidence of T1070.003 before clearing.

**Evidence:** Preserve the following forensic evidence BEFORE eradication: (1) A full memory dump of the running GoGra process if still active — use `gcore ` or `dd if=/proc//mem` with the maps file to capture in-memory strings including any embedded OAuth client secrets or C2 configuration. (2) The GoGra binary itself, hashed and stored offline — Harvester is known to use custom tooling with limited public signatures, making the binary a high-value threat intelligence artifact. (3) All command history files with gaps — `~/.bash_history`, `~/.zsh_history` — where missing entries or zero-byte files indicate T1070.003 cleanup; photograph or copy before any remediation. (4) The full list of Outlook mailbox items (subject lines, sender/recipient metadata only) accessed by the malicious app identity from the UAL `MailItemsAccessed` records — this establishes what intelligence Harvester may have exfiltrated. (5) `journalctl -u --no-pager -o export > service_journal.txt` to preserve systemd journal entries for the malicious service before unit file removal.

**Step 4: Recovery — After credential rotation and binary removal, monitor Graph API audit logs for resumed polling behavior. Confirm no new unauthorized app registrations appear in Entra ID. Validate systemd services and XDG autostart entries on affected Linux hosts against a known-good baseline. Monitor outbound HTTPS to Microsoft Graph API endpoints (graph.microsoft.com) from Linux hosts — legitimate use should be limited and attributable.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure), CIS 7.2 (Establish and Maintain a Remediation Process)

**Compensating:** Generate a systemd service baseline on clean hosts using `systemctl list-unit-files --type=service --state=enabled -o json > baseline_services.json` and diff against recovering hosts with `diff baseline_services.json recovered_host_services.json`. For XDG autostart, use `md5sum /etc/xdg/autostart/*.desktop ~/.config/autostart/*.desktop` and compare hashes to baseline. For Graph API monitoring without a SIEM, configure Microsoft Entra ID `Diagnostic Settings` to stream `AuditLogs` and `SignInLogs` to a Log Analytics workspace (free tier available) and create a simple alert rule for any new `Add app role assignment to service principal` event. Use `tcpdump -i any -n host graph.microsoft.com -w graph_traffic.pcap` on Linux hosts to capture any resumed Graph API polling — GoGra's polling interval produces a recognizable time-series pattern in packet captures.

**Evidence:** During the recovery monitoring window, continuously collect: (1) Microsoft Entra ID AuditLogs filtered on `Add application` and `Update application` operations — any new registration within 30 days post-eradication is a strong re-infection indicator. (2) Microsoft 365 UAL `MailItemsAccessed` records filtered to `UserType: Application` — resumed periodic access by a service principal after credential rotation indicates Harvester re-deployed GoGra with new credentials. (3) Linux host `auditd` records for new file creation events in `/etc/systemd/system/`, `/usr/lib/systemd/system/`, and all user `~/.config/autostart/` directories — rule: `-w /etc/systemd/system -p wa -k systemd_persistence`. (4) Outbound network flows to Microsoft ASN 8075 (`graph.microsoft.com` resolves to this ASN) from Linux hosts — any process other than known, inventoried applications initiating these connections is anomalous.

**Step 5: Post-Incident — This campaign exposes three control gaps: insufficient OAuth app governance, absence of behavioral monitoring for Graph API abuse on Linux endpoints, and no baseline for legitimate vs. anomalous Graph API usage. Implement Conditional Access policies restricting Graph API app consent to admin-approved applications. Deploy audit logging for all OAuth consent events. Extend EDR coverage to Linux infrastructure if not already present. Review MITRE ATT&CK T1102.002 detection coverage in your SIEM.**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §4 — Post-Incident Activity

**Controls:** NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-4 (System Monitoring), NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.5 (Require MFA for Administrative Access), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

**Compensating:** Without enterprise EDR, deploy Auditd on all Linux hosts with rules targeting GoGra-relevant behaviors: file writes to systemd unit directories (`-w /etc/systemd/system -p wa`), execve syscalls from non-standard paths (`-a always,exit -F arch=b64 -S execve -F dir=/tmp -k suspicious_exec`), and outbound connections from unexpected processes (`-a always,exit -F arch=b64 -S connect -k outbound_conn`). Write a Sigma rule targeting the Graph API polling pattern: periodic `MailItemsAccessed` events from `UserType: Application` with inter-event deltas under 10 minutes from a single `AppId` not in your allowlist — this is the behavioral signature of GoGra's C2 loop (T1102.002). For OAuth governance without Entra ID P2 licensing, enable the free `User consent settings` in Entra ID to require admin approval for all app consent requests, eliminating the self-consent vector Harvester exploited.

**Evidence:** Compile the following into the post-incident report for threat intelligence sharing and future detection improvement: (1) SHA-256 hashes of all GoGra Linux binaries recovered, with file path, compile timestamp (from `readelf -p .comment ` if available), and associated systemd service name — submit to VirusTotal and share with sector ISAC (telecom, government). (2) The `AppId` and `createdDateTime` of all malicious Entra ID app registrations, including the OAuth permission scope requested — these are reusable IOCs across other organizations in the same sector. (3) The polling interval extracted from UAL `MailItemsAccessed` timestamps — Harvester's GoGra C2 loop timing is a behavioral fingerprint for future detection rules. (4) The full attack timeline from first Graph API consent event to discovery, mapped to MITRE ATT&CK: Initial Access (T1078.004), Persistence (T1543.002, T1547.013), C2 (T1102.002), Defense Evasion (T1070.003) — this mapping supports lessons-learned and detection gap analysis.

## Detection Guidance

Primary detection surface is Microsoft 365 Unified Audit Logs and Azure AD / Entra ID sign-in and audit logs. Query for: (1) OAuth application consent grants with Mail.Read, Mail.Send, or Mail.ReadWrite scopes granted within the last 90 days, flag any not in your approved application inventory; (2) Graph API calls to /v1.0/me/messages or /v1.0/users/{id}/messages originating from service principals on Linux hosts or from hosts not expected to access mail programmatically; (3) Periodic low-volume email activity (sends and receives at regular intervals) by non-human identities, GoGra polls the Outlook inbox for commands on a schedule. On Linux endpoints: monitor for new systemd unit file creation in /etc/systemd/system/ or /lib/systemd/system/ by non-root processes, and new entries in ~/.config/autostart/ (XDG). Flag command history gaps or explicit history-clearing commands (history -c, truncation of ~/.bash_history). AES-encrypted payloads over Graph API are not inspectable at the network layer for content-based signatures. Detection must rely on behavioral indicators and Graph API audit log analysis, not payload inspection. Behavioral indicators: Linux process spawning network connections exclusively to graph.microsoft.com with no corresponding legitimate application owner, combined with periodic execution cadence.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `graph.microsoft.com` | Legitimate Microsoft Graph API endpoint abused by GoGra for C2 polling and data exfiltration — presence alone is not malicious; look for access from Linux hosts with no legitimate application owner | LOW |

| Type | Value | Context | Confidence |
|---|---|---|---|
| URL | `https://graph.microsoft.com/v1.0/me/messages` | Graph API mail endpoint used by GoGra to retrieve attacker commands from a dedicated Outlook inbox — flag access from Linux processes not associated with approved applications | MEDIUM |

## Framework Mappings

### MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter
- **T1566.001** — Spearphishing Attachment
- **T1071.001** — Web Protocols
- **T1070.003** — Clear Command History
- **T1102.002** — Bidirectional Communication
- **T1078.004** — Cloud Accounts
- **T1543.002** — Systemd Service
- **T1560** — Archive Collected Data
- **T1547.013** — XDG Autostart Entries
- **T1027** — Obfuscated Files or Information
- **T1071.003** — Mail Protocols
- **T1204.002** — Malicious File
- **T1078** — Valid Accounts
- **T1573.001** — Symmetric Cryptography
- **T1041** — Exfiltration Over C2 Channel
- **T1105** — Ingress Tool Transfer

### NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CA-7** — Continuous Monitoring

- **SC-13** — Cryptographic Protection

**OWASP-TOP10-2021**

- **A07:2021** — Identification and Authentication Failures
- **A02:2021** — Cryptographic Failures

**CIS-V8**

- **16.10** — Apply Secure Design Principles in Application Architectures

**ISO-27001-2022**

- **A.8.28** — Secure coding
- **A.8.24** — Use of cryptography
- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

**HIPAA-SECURITY**

- **164.312(e)(1)** — Transmission Security

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1059 | Command and Scripting Interpreter | Execution |
| T1566.001 | Spearphishing Attachment | Initial-Access |
| T1071.001 | Web Protocols | Command-And-Control |
| T1070.003 | Clear Command History | Defense-Evasion |
| T1102.002 | Bidirectional Communication | Command-And-Control |
| T1078.004 | Cloud Accounts | Defense-Evasion |
| T1543.002 | Systemd Service | Persistence |
| T1560 | Archive Collected Data | Collection |
| T1547.013 | XDG Autostart Entries | Persistence |
| T1027 | Obfuscated Files or Information | Defense-Evasion |
| T1071.003 | Mail Protocols | Command-And-Control |
| T1204.002 | Malicious File | Execution |
| T1078 | Valid Accounts | Defense-Evasion |
| T1573.001 | Symmetric Cryptography | Command-And-Control |

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1041 | Exfiltration Over C2 Channel | Exfiltration |
| T1105 | Ingress Tool Transfer | Command-And-Control |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://www.bleepingcomputer.com/news/security/new-gogra-malware-fo... | T3 |
| **Critical Exposure of Azure AD Data via Unauthenticated Microsoft ...** | https://kudelskisecurity.com/research/critical-exposure-of-azure-ad... | T3 |
| **User sent messages in Microsoft Teams to multiple conversations ...** | https://docs-cortex.paloaltonetworks.com/r/Cortex-XDR/Cortex-XDR-An... | T3 |
| **A pair of flaws in Microsoft's Entra ID identity and access ...** | https://www.facebook.com/wired/posts/a-pair-of-flaws-in-microsofts-... | T3 |
| **This Microsoft Entra ID Vulnerability Could Have Been Catastrophic** | https://www.reddit.com/r/sysadmin/comments/1nlbl8r/this_microsoft_e... | T3 |



Generated 2026-04-22 13:40 UTC by TJS Security Command Center