INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-03 06:49 UTC

ToddyCat APT Deploys Umbrij Malware to Steal Gmail OAuth Tokens via Chrome Remote Debugging

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0617
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Google Chrome (remote debugging interface), Microsoft Edge, Gmail/Google Workspace (OAuth API), Bitdefender ConnectAgent (BDSubWiz.exe, abused as signed binary loader), Microsoft Visual Studio (VSTestVideoRecorder.exe, abused as signed binary loader), Google Desktop (GoogleDesktop.exe, abused as signed binary loader)
Published	2026-07-02T09:04:13
Discovery Source	Rss

Executive Summary

The ToddyCat APT group has deployed a new malware family, Umbrij, that silently steals Gmail OAuth bearer tokens from active Chrome browser sessions by hijacking Chrome's built-in remote debugging interface, no user credentials or MFA bypass required. Organizations running Google Workspace in corporate environments are the reported target, with stolen tokens granting persistent, API-level access to Gmail, Google Drive, Contacts, Calendar, and Tasks. According to Kaspersky researchers, Umbrij disguises itself using legitimately signed executables from Bitdefender, Microsoft, and Google to evade endpoint detection, raising the risk of undetected, long-duration email and document exfiltration.

Technical Analysis

Umbrij is a .NET malware family attributed by Kaspersky to the ToddyCat APT group (corroborated by The Hacker News). The malware abuses the Chrome DevTools Protocol (CDP) by launching a headless browser instance attached to an existing authenticated Chrome profile. Through this debugging channel, Umbrij intercepts OAuth token issuance and exfiltrates bearer tokens granting API-level access to Gmail, Google Drive, Contacts, Calendar, and Tasks (MITRE T1528, Steal Application Access Token; T1539, Steal Web Session Cookie; T1185, Browser Session Hijacking). No CVE has been assigned; the technique exploits legitimate CDP functionality rather than a patched software flaw. Relevant CWEs: CWE-732 (Incorrect Permission Assignment

for Critical Resource), CWE-287 (Improper Authentication), CWE-522 (Insufficiently Protected Credentials). To evade detection, Umbrij abuses three legitimately signed binaries as execution proxies: Bitdefender ConnectAgent (BDSubWiz.exe), Microsoft Visual Studio (VSTestVideoRecorder.exe), and Google Desktop (GoogleDesktop.exe), consistent with MITRE T1218 (System Binary Proxy Execution) and T1036.004 (Masquerading: Match Legitimate Name or Location). Additional mapped techniques include T1560 (Archive Collected Data), T1114.002 (Remote Email Collection), T1114.003 (Email Forwarding Rule), T1134.001 (Token Impersonation/Theft), T1574.002 (DLL Side-Loading), T1550.001 (Application Access Token), and T1053.005 (Scheduled Task/Job). Confidence in core campaign attribution and technique description is HIGH per dual-source reporting. Confidence in victim count and geographic scope is LOW, single-source (Kaspersky) without independent victim confirmation. Patch status: no vendor patch available; the attack surface is legitimate browser functionality.

Action Checklist

1. Step 1: Containment. Audit all corporate endpoints for anomalous Chrome remote debugging activity. Check for processes launching Chrome or Edge with '--remote-debugging-port' flags or CDP connections originating from unexpected parent processes. Temporarily disable or restrict Chrome remote debugging capability via Group Policy where not operationally required. Flag and isolate any host where BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe executed outside their expected installation paths (NIST IR-4, Incident Handling; CIS 4.6, Securely Manage Enterprise Assets and Software).
2. Step 2: Detection. Query EDR and process-creation logs for execution of BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe from non-standard directories or with unusual parent processes. Search endpoint logs for Chrome launched with '--remote-debugging-port' or '--remote-debugging-address' arguments. Review Google Workspace audit logs (Admin Console > Reports > Audit) for OAuth token grants to unrecognized applications or unusual API access patterns across Gmail, Drive, Contacts, Calendar, and Tasks. Correlate with MITRE T1528 (token theft) and T1218 (signed binary proxy) behavioral indicators. Reference: NIST AU-2 (Event Logging), AU-6 (Audit Record Review), CIS 8.2 (Collect Audit Logs).
3. Step 3: Eradication. Revoke all active OAuth tokens for affected Google Workspace accounts via the Google Admin Console (Users > Security > Connected Apps, revoke third-party access). Force re-authentication for affected users. Remove or quarantine any identified Umbrij-related binaries. Investigate DLL side-loading paths associated with the abused signed binaries and remove unauthorized DLLs (NIST SI-3, Malware and Unwanted Software Protection; SI-7, Software, Firmware, and Information Integrity; CIS 2.3, Address Unauthorized Software).
4. Step 4: Recovery. After token revocation, monitor Google Workspace audit logs for anomalous re-authorization attempts or residual OAuth app grants (NIST AU-6, AU-12). Validate that Chrome remote debugging is disabled or restricted across the fleet before restoring normal operations. Confirm no email forwarding rules were added to affected accounts (consistent with T1114.003). Re-image endpoints where Umbrij execution is confirmed. Apply multi-factor authentication review; note that MFA does not block this attack vector since tokens are stolen post-authentication. Verify conditional access policies are configured to detect anomalous token usage locations.
5. Step 5: Post-Incident. This campaign exposes gaps in signed-binary allowlisting and OAuth token lifecycle management. Implement application allowlisting that validates not just signature but expected execution path and parent process (CIS 2.3; NIST AC-6, Least Privilege). Review Google Workspace

OAuth application policies to restrict which applications can be granted API access (NIST AC-3, Access Enforcement; AC-20, Use of External Systems). Establish alerting on OAuth token grants to new or unrecognized application client IDs. Evaluate Chrome enterprise policy to block remote debugging on managed devices.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to legal, privacy counsel, and executive leadership if Google Workspace audit logs confirm OAuth token grants to unrecognized client IDs with Gmail or Drive scopes for accounts with access to PII, PHI, financial data, or regulated customer communications, as this constitutes unauthorized access to cloud-stored data and may trigger breach notification obligations under GDPR, HIPAA, or applicable state breach laws; additionally escalate if more than five endpoints show evidence of Umbrij DLL side-loading, indicating a broad ToddyCat campaign footprint rather than an isolated incident.
Recovery Notes	Before restoring any reimaged endpoint to production, independently verify via Group Policy Results (`gpresult /H`) that the Chrome enterprise policy 'RemoteDebuggingAllowed' is set to Disabled and that AppLocker or equivalent allowlisting rules are enforced and logging. Monitor Google Workspace Admin Audit logs daily for a minimum of 30 days post-eradication for new OAuth token grants to any client ID not on the approved allowlist, paying particular attention to the Gmail and Drive API scopes that Umbrij targeted, as ToddyCat actors have demonstrated persistent re-access attempts after initial eviction. Verify that no Gmail delegation, forwarding rules, or IMAP/POP access settings were modified on affected accounts during the compromise window, as these provide durable secondary access channels that survive OAuth token revocation.

Forensic Artifacts	Chrome DevTools Protocol (CDP) session logs and chrome.debugger API call records: Umbrij uses Chrome's remote debugging interface on port 9222 (default) to invoke the `Network.getResponseBody` and `Fetch.requestPaused` CDP methods to intercept OAuth token responses in flight — look for CDP WebSocket frames in a Wireshark capture on loopback (127.0.0.1:9222) or in any Chrome debug log enabled via '--enable-logging --v=1' capturing `chrome_debug.log` in the User Data directory. | DLL side-loading artifacts in signed binary directories: The unauthorized DLL loaded by BDSubWiz.exe (Bitdefender ConnectAgent), VSTestVideoRecorder.exe (Visual Studio), or GoogleDesktop.exe will be present in the same filesystem directory as the signed host binary rather than its legitimate system or vendor path — capture SHA-256 hashes and PE metadata (compile timestamp, import table, section entropy) of all DLLs in those directories for comparison against vendor-published file manifests. | Google Workspace Admin SDK Token audit log (`TOKEN_ISSUED`, `AUTHORIZE` events): These records show the exact OAuth client ID, scopes requested, and the user account that granted access — the client ID registered by Umbrij's infrastructure will appear here and will not match any application in the organization's approved OAuth app inventory; export via Admin Console Reports API before any account remediation destroys the 180-day rolling audit window. | Windows Prefetch files for the three abused signed binaries: `C:\Windows\Prefetch\BDSUBWIZ.EXE-{hash}.pf`, `VSTESTVIDEOREORDER.EXE-{hash}.pf`, and `GOOGLEDESKTOP.EXE-{hash}.pf` record execution count, last run time, and up to 1024 file references including the side-loaded DLL path — parse with WinPrefetchView or the `PECmd` tool from Eric Zimmerman's toolkit to establish the execution timeline and confirm the malicious DLL load occurred during the same execution event. | Chrome SQLite Cookies database and in-memory OAuth bearer token artifacts: The `Cookies` database at `%LOCALAPPDATA%\Google\Chrome\User Data\Default\Cookies` may retain residual OAuth token metadata, but more critically, a live RAM acquisition (WinPmem/DumpIt) of the chrome.exe process taken before host isolation will contain plaintext OAuth bearer tokens in heap memory that Umbrij extracted via CDP — these tokens can be used to establish the exact API access scope the attacker obtained and to validate whether exfiltration of Gmail, Drive, Contacts, Calendar, or Tasks data occurred.

Per-Action IR Details

Step 1: Containment — Audit all corporate endpoints for anomalous Chrome remote debugging activity. Check for processes launching Chrome or Edge with '--remote-debugging-port' flags or CDP connections originating from unexpected parent processes. Temporarily disable or restrict Chrome remote debugging capability via Group Policy where not operationally required. Flag and isolate any host where BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe executed outside their expected installation paths (NIST IR-4 — Incident Handling; CIS 4.6 — Securely Manage Enterprise Assets and Software).

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 — not in knowledge base; omitted, CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Without EDR: Deploy Sysmon with EventID 1 (Process Create) configured to log full command-line arguments; filter for '--remote-debugging-port' or '--remote-debugging-address' in chrome.exe or msedge.exe command lines using a PowerShell query: `Get-WinEvent -LogName 'Microsoft-Windows-Sysmon/Operational' | Where-Object {$_.Message -match 'remote-debugging-port'}`. For signed-binary abuse, query Sysmon EventID 7 (Image Load) to flag DLLs loaded by BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe from paths other than their vendor installation directories. Use Group Policy (Computer Configuration > Administrative Templates > Google > Google Chrome > Remote Debugging) to set 'RemoteDebuggingAllowed' to Disabled across the fleet.

Evidence: Before isolating any flagged host, capture: (1) full RAM image using WinPmem or DumpIt to preserve in-memory CDP session state and any injected Umbrij DLL artifacts; (2) live network connections via `netstat -ano` or `Get-NetTCPConnection` to document active CDP connections on port 9222 (or configured debug port) and their remote endpoints; (3) running process tree via `Get-Process` and `Get-CimInstance Win32_Process | Select-Object Name,ProcessId,ParentProcessId,CommandLine` to capture the full parent-child chain showing BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe as the loader; (4) prefetch files from C:\Windows\Prefetch\ for BDSubWiz.exe, VSTestVideoRecorder.exe, and GoogleDesktop.exe before any reimaging. These artifacts are destroyed upon host isolation or power-off.

Step 2: Detection — Query EDR and process-creation logs for execution of BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe from non-standard directories or with unusual parent processes. Search endpoint logs for Chrome launched with '--remote-debugging-port' or '--remote-debugging-address' arguments. Review Google Workspace audit logs (Admin Console > Reports > Audit) for OAuth token grants to unrecognized applications or unusual API access patterns across Gmail, Drive, Contacts, Calendar, and Tasks. Correlate with MITRE T1528 (token theft) and T1218 (signed binary proxy) behavioral indicators. Reference: NIST AU-2 (Event Logging), AU-6 (Audit Record Review), CIS 8.2 (Collect Audit Logs), D3-LAM (Local Account Monitoring), D3-SFA (System File Analysis).

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 8.2 (Collect Audit Logs)

Compensating: Without SIEM/EDR: Enable Windows Security Event Log Event ID 4688 (Process Creation) with command-line auditing via Group Policy (Audit Process Creation + Include command line in process creation events). Query with: `Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4688 -and $_.Message -match 'remote-debugging-port'}`. For signed-binary DLL side-loading, use Sysmon EventID 7 to compare loaded DLL paths against known-good Bitdefender, Visual Studio, and Google Desktop installation paths stored in a baseline CSV. For Google Workspace OAuth visibility without a SIEM, export the Admin SDK Reports API activity log via `gam report token` (using the free GAM CLI tool) and filter for `app_name` fields not matching an approved application allowlist. Cross-reference OAuth grant timestamps against endpoint process execution timestamps from Sysmon logs.

Evidence: No live-state alteration occurs in this detection step; no pre-capture is required before querying logs. Key artifacts to examine: (1) Windows Security EventID 4688 or Sysmon EventID 1 showing chrome.exe spawned with '--remote-debugging-port=9222' (or any non-zero port) with BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe as the parent process; (2) Google Workspace Admin Audit log entries for token grants (`TOKEN_ISSUED` events) to client IDs not present in the approved OAuth application registry; (3) Sysmon EventID 7 (Image Load) showing unsigned or mismatched-path DLLs loaded into the address space of the three signed loader binaries — check specifically for DLLs in the same directory as the executable rather than system32 or the vendor path; (4) Chrome User Data directory (`%LOCALAPPDATA%\Google\Chrome\User Data\Default\`) for unexpected modifications to Preferences or any files written by a non-Chrome process.

Step 3: Eradication — Revoke all active OAuth tokens for affected Google Workspace accounts via the Google Admin Console (Users > Security > Connected Apps — revoke third-party access). Force re-authentication for affected users. Remove or quarantine any identified Umbrij-related binaries. Investigate DLL side-loading paths associated with the abused signed binaries and remove unauthorized DLLs (NIST SI-3 — no mapped control for DLL side-loading specifically in the provided knowledge base; CIS 2.3 — Address Unauthorized Software; D3-CRO — Credential Rotation).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 2.3 (Address Unauthorized Software), NIST AC-12 (Session Termination)

Compensating: Without enterprise MDM for token revocation: Use the Google Admin SDK Directory API with a service account to enumerate and revoke tokens programmatically — `gam user [email] delete token [clientid]` via

GAM CLI for each affected account. For binary removal without EDR response capability, use Sysinternals Autoruns to identify persistence mechanisms tied to BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe (check Image Hijacks and AppInit DLLs tabs), then manually delete identified side-loaded DLLs and quarantine to a password-protected zip with SHA-256 hash documented before deletion. Verify removal with `Get-FileHash` on the quarantined samples and submit to VirusTotal or an internal sandbox for family confirmation.

Evidence: Before revoking OAuth tokens and before removing or quarantining binaries, capture: (1) a full list of active OAuth tokens for affected accounts via Google Admin Console export or `gam user [email] show tokens` — document all client IDs, scopes (particularly https://mail.google.com/, https://www.googleapis.com/auth/drive), and grant timestamps before revocation destroys the record of attacker access windows; (2) a forensic copy (SHA-256 hashed) of all Umbrij-related DLLs from their side-loading locations (directory of BDSubWiz.exe, VSTestVideoRecorder.exe, GoogleDesktop.exe) before quarantine; (3) Sysmon EventID 11 (File Create) records showing when the malicious DLLs were written to disk and by what parent process — this establishes the initial access timeline; (4) a YARA scan output (using a rule matching Umbrij PE characteristics if available from Kaspersky's disclosure) against the full endpoint before removal so that no persistence copy is overlooked.

Step 4: Recovery — After token revocation, monitor Google Workspace audit logs for anomalous re-authorization attempts or residual OAuth app grants (NIST AU-6, AU-12). Validate that Chrome remote debugging is disabled or restricted across the fleet before restoring normal operations. Confirm no email forwarding rules were added to affected accounts (consistent with T1114.003). Re-image endpoints where Umbrij execution is confirmed. Apply D3-MFA (Multi-factor Authentication) review — note that MFA does not block this attack vector since tokens are stolen post-authentication; verify conditional access policies are configured to detect anomalous token usage locations.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 6.3 (Require MFA for Externally-Exposed Applications)

Compensating: Without enterprise DLP or CASB for OAuth monitoring: Configure Google Workspace Alert Center (free, included in all Workspace tiers) to alert on 'Suspicious activity' and 'Token grant to new application' events. Use `gam report token` on a daily scheduled task to diff OAuth grants against a known-good baseline CSV — new entries trigger manual review. For Gmail forwarding rule detection without a SIEM, run `gam user [email] show filters` and `gam user [email] show forwardingaddresses` for all accounts flagged as potentially compromised. Before restoring an endpoint to production, boot from a known-good read-only USB baseline (e.g., CAINE or TSURUGI Linux) and compare installed files in Bitdefender, Visual Studio, and Google Desktop directories against SHA-256 hashes from the vendor's official distribution.

Evidence: Before reimaging confirmed Umbrij endpoints, capture: (1) a full disk image using dd or FTK Imager to preserve the complete forensic record of the DLL side-loading artifacts, Umbrij binary, and any staged exfiltration data; (2) the Chrome User Data directory (`%LOCALAPPDATA%\Google\Chrome\User Data\`) in its entirety — particularly the SQLite databases (Cookies, Login Data, Web Data) and any DevTools protocol logs that may record CDP session commands Umbrij issued to extract OAuth bearer tokens from chrome.debugger API calls; (3) the Windows Registry hive `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` and `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` exported via `reg export` before wipe, to document any persistence keys Umbrij or its loader established; (4) a copy of Gmail forwarding rules and filter configurations from affected accounts before re-authentication forces any rule reconfiguration.

Step 5: Post-Incident — This campaign exposes gaps in signed-binary allowlisting and OAuth token lifecycle management. Implement application allowlisting that validates not just signature but expected execution path and parent process (CIS 2.3; NIST AC-6 — Least Privilege). Review Google Workspace OAuth application policies to restrict which applications can be granted API access (NIST AC-3 — Access Enforcement; AC-20 — Use of External Systems). Establish alerting on OAuth token grants to new or unrecognized application client IDs. Evaluate Chrome enterprise policy to block remote debugging on managed devices. Reference D3-UAP (User Account Permissions) and D3-CH (Credential Hardening) for ongoing hardening posture.

NIST Phase: Post Incident

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

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), NIST AC-20 (Use Of External Systems), CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a CASB or enterprise app-governance platform: Enforce Google Workspace OAuth application restrictions via Admin Console (Security > API Controls > App Access Control) — set to 'Restrict which third-party apps can access Google Workspace data' and require admin approval for new OAuth client IDs accessing Gmail, Drive, Contacts, Calendar, and Tasks scopes. For signed-binary side-loading hardening without a commercial allowlisting tool, deploy AppLocker rules (available in Windows 10/11 without additional cost) that enforce execution only from vendor-specific installation paths for BDSubWiz.exe, VSTestVideoRecorder.exe, and GoogleDesktop.exe, and block DLL loading from those same directories if the DLL is not present in a verified hash baseline. Document lessons learned per NIST 800-61r3 §4 including the gap that signed-binary proxy execution evaded existing AV controls, and submit IOCs (DLL hashes, OAuth client IDs used by Umbrij's C2 registration) to your ISAC or to CISA's automated indicator sharing feed.

Evidence: No volatile evidence is at risk in this post-incident phase as systems should already be contained and reimaged. Focus on documentary evidence for lessons learned: (1) the complete timeline reconstructed from Sysmon, Windows Security, and Google Workspace audit logs showing dwell time from initial DLL drop to first OAuth token exfiltration; (2) the list of OAuth client IDs and API scopes that Umbrij's infrastructure registered or abused, preserved from the pre-eradication token export, for threat intelligence sharing and future detection rule development; (3) a diff of Chrome enterprise policy GPO settings before and after remediation to document the control gap that permitted '--remote-debugging-port' invocation on managed endpoints.

Detection Guidance

Primary behavioral indicators: (1) Chrome or Edge processes spawned with '--remote-debugging-port' or '--remote-debugging-address' command-line arguments by a parent process other than a known developer tool, query EDR process-creation telemetry for this pattern. (2) BDSubWiz.exe, VSTestVideoRecorder.exe, or GoogleDesktop.exe executing from paths inconsistent with their vendor installation directories, flag in EDR and SIEM. (3) Unusual child processes or DLL loads associated with the above signed binaries, consistent with T1574.002 (DLL Side-Loading). (4) Google Workspace Admin audit logs showing OAuth token grants to unrecognized application client IDs, or API calls to Gmail, Drive, Contacts, Calendar, or Tasks from IP addresses inconsistent with the user's normal access pattern, check Admin Console Reports > OAuth Token Audit. (5) Scheduled tasks (T1053.005) created by unexpected processes, review Windows Task Scheduler logs (Event ID 4698, task created; Event ID 4702, task updated). (6) Email forwarding rules created on corporate accounts without corresponding user action (T1114.003), audit via Google Workspace Gmail audit log. For confirmed IOCs (file hashes, C2 domains, IP addresses), consult the Kaspersky Securelist Part 2 report directly; this summary focuses on behavioral detection patterns applicable across environments.

Framework Mappings

MITRE-ATTACK

- **T1528** — Steal Application Access Token
- **T1560** — Archive Collected Data
- **T1059.002** — AppleScript
- **T1114.002** — Remote Email Collection
- **T1218** — System Binary Proxy Execution

- **T1134.001** — Token Impersonation/Theft
- **T1574.002** — DLL Side-Loading
- **T1550.001** — Application Access Token
- **T1053.005** — Scheduled Task
- **T1114.003** — Email Forwarding Rule
- **T1036.004** — Masquerade Task or Service
- **T1185** — Browser Session Hijacking
- **T1539** — Steal Web Session Cookie

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A07:2021** — Identification and Authentication Failures
- **A04:2021** — Insecure Design

NIST-800-53R5

- **AC-3** — Access Enforcement
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **IA-5** — Authenticator Management
- **SI-4** — System Monitoring

CIS-V8

- **3.3** — Configure Data Access Control Lists
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **5.2** — Use Unique Passwords
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(ii)(D)** — Password Management

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

- **A.5.34** — Privacy and protection of personal information

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1528	Steal Application Access Token	Credential-Access
T1560	Archive Collected Data	Collection
T1059.002	AppleScript	Execution
T1114.002	Remote Email Collection	Collection
T1218	System Binary Proxy Execution	Defense-Evasion
T1134.001	Token Impersonation/Theft	Defense-Evasion
T1574.002	DLL Side-Loading	Persistence
T1550.001	Application Access Token	Defense-Evasion
T1053.005	Scheduled Task	Execution
T1114.003	Email Forwarding Rule	Collection
T1036.004	Masquerade Task or Service	Defense-Evasion
T1185	Browser Session Hijacking	Collection
T1539	Steal Web Session Cookie	Credential-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/07/toddycat-linked-umbrij-malware-ab...	T2
ToddyCat: your hidden email assistant. Part 2	https://securelist.com/toddycat-apt-umbrij-tool-and-oauth/120251/	T1
I have Bitdefender Security on my Laptop and PC and am ...	https://support.google.com/chrome/thread/41467993/i-have-bitdefende...	T3
Repeatedly getting threats detected in google chrome	https://forums.malwarebytes.com/topic/271895-repeatedly-getting-thr...	T1

Source	URL	Tier
Online threat detection - repeated warnings	https://community.bitdefender.com/en/discussion/105423/online-threa...	T1

Generated 2026-07-03 06:49 UTC by TJS Security Command Center