INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:33 UTC

# AI-Assisted Malware Enters Ransomware Attack Chains: Interlock’s Slopoly Backdoor Signals a Tactical Shift

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

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0078 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Windows systems, scheduled tasks (schtasks.exe), cmd.exe, Windows Restart Manager API; no specific enterprise product CVEs identified |
| **Published** | 2026-03-22 |

## Executive Summary

IBM X-Force identified a financially motivated threat actor, Hive0163, deploying an AI-generated backdoor called Slopoly as part of an Interlock ransomware attack chain targeting Windows environments. The actor maintained persistent access for over one week before beginning data exfiltration, indicating a dwell-time window sufficient for full network reconnaissance and staging. The business risk is material: AI-assisted malware development is lowering the skill threshold for custom tool creation, enabling threat actors to produce bespoke malware faster and at lower cost, thereby expanding the pool of adversaries capable of executing ransomware campaigns.

## Technical Analysis

Hive0163 deployed Slopoly, a PowerShell backdoor assessed by IBM X-Force as AI-generated, based on code characteristics consistent with LLM-assisted development: verbose inline comments, semantically descriptive variable names, and structured error handling. No CVEs are attributed to this campaign; exploitation relies on Windows-native mechanisms rather than patched vulnerabilities. The attack chain maps to: initial access via phishing (T1566) and malicious user execution (T1204.001); persistence via scheduled tasks using schtasks.exe (T1053.005); execution via cmd.exe and PowerShell (T1059.001); masquerading techniques (T1036, T1036.004); deobfuscation/decoding (T1140); C2 communication over HTTP/S (T1071.001); ingress tool transfer (T1105); and exfiltration (T1041), culminating in data encryption via Interlock ransomware (T1486). The backdoor also leverages the Windows Restart Manager API. Relevant weaknesses are CWE-77 (command injection via cmd.exe invocation) and CWE-494 (download of code without integrity verification). No CVSS

vector string was provided with the source data; the 7.5 base score is an input estimate, not a verified NVD-sourced value. EPSS data is not available for this campaign item. Primary source: IBM X-Force via BleepingComputer.

## Action Checklist

1. Step 1 (Immediate): Hunt for Slopoly indicators in your environment, search PowerShell execution logs and scheduled task creation events for anomalous entries created by cmd.exe or schtasks.exe outside of known administrative windows. Prioritize endpoints with recent phishing exposure.
2. Step 2 (Detection): Enable and review Windows Event IDs 4698 (scheduled task created), 4702 (task updated), 4104 (PowerShell script block logging), and 4688 (process creation) for chains involving schtasks.exe spawning cmd.exe or powershell.exe with encoded or obfuscated arguments.
3. Step 3 (Assessment): Inventory endpoints where PowerShell execution policy permits unrestricted or bypass mode. Identify systems with internet-facing exposure where ingress tool transfer (T1105) or C2 beaconing (T1071.001) could go undetected. Confirm EDR coverage on all Windows endpoints.
4. Step 4 (Communication): Notify the SOC and IR team of the Hive0163/Interlock TTPs. If your organization received recent phishing campaigns targeting Windows users, escalate to incident response triage. Brief leadership on ransomware dwell-time risk, over one week of undetected access is the confirmed baseline for this actor.
5. Step 5 (Long-term): Review detection rule coverage against the full MITRE technique set listed in this campaign. Assess whether behavioral analytics can identify LLM-generated PowerShell patterns (structured comments, verbose variable names) as an anomaly signal. Update phishing simulation and user awareness programs to address T1566/T1204.001 entry vectors. Consider restricting PowerShell to Constrained Language Mode on endpoints where full functionality is not required.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |
| **Escalation Criteria** | Escalate to external IR firm or CISA if any endpoint shows evidence of scheduled task execution by schtasks.exe or cmd.exe outside normal admin windows within the past 30 days, or if phishing telemetry indicates Hive0163-attributed emails successfully delivered to >5 users in the past 7 days. |
| **Recovery Notes** | Post-containment: (1) Force-reset all administrative account passwords (domain and local) to eliminate lateral movement via cached credentials. (2) Restore all affected endpoints from clean backups dated prior to the earliest detected schtasks.exe/cmd.exe anomaly, or perform full reimaging if backup integrity cannot be verified. (3) Conduct full network scan for persistence mechanisms (scheduled tasks, WMI Event Subscriptions via 'wmic /namespace:\\root\subscription class __EventFilter list brief', Registry Run keys, BITS jobs via 'bitsadmin /list /verbose') and remove all non-whitelisted entries. |

| | |
|---|---|
| **Forensic Artifacts** | Windows Security Event Log (C:\Windows\System32\winevt\Logs\Security.evtx) — Events 4688 (process creation with command-line), 4698 (scheduled task created), 4702 (scheduled task updated), 106 (task scheduler service events) \| Windows PowerShell Event Log (C:\Windows\System32\winevt\Logs\Windows PowerShell.evtx) and Script Block Logging (Event ID 4104) — captures executed PowerShell commands and obfuscation patterns \| Registry hive: HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree — persistence via scheduled tasks; export with 'reg export' \| Scheduled Tasks directory (C:\Windows\System32\Tasks and C:\Windows\SysWOW64\Tasks) — task definition XML files contain execution parameters, triggers, and command-line arguments \| Process execution artifacts (if EDR unavailable): Windows Prefetch files (C:\Windows\Prefetch*.pf) contain execution timeline and command-line history; analyze with WinPrefetchView (Nirsoft) or Prefetch Parser |

### Per-Action IR Details

**Step 1 (Immediate): Hunt for Slopoly indicators in your environment — search PowerShell execution logs and scheduled task creation events for anomalous entries created by cmd.exe or schtasks.exe outside of known administrative windows. Prioritize endpoints with recent phishing exposure.**

**NIST Phase:** Detection Analysis

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

**Controls:** NIST IR-4(1), CIS 6.2 (Audit Log Centralization), CIS 8.2 (User Account Logging)

**Compensating:** On systems without SIEM: Export Windows Event Log 4698 and 4688 from all endpoints using wevtutil.exe (e.g., 'wevtutil qe Security /q:*[System[(EventID=4698 or EventID=4688)]] /f:text > tasks.txt'). Parse for schtasks.exe or cmd.exe process chains using grep or PowerShell filter-objects. Establish baseline of legitimate admin scheduling windows (e.g., Tuesday 2-4 AM) and flag outliers manually. Use Autoruns (Sysinternals) to dump scheduled tasks on each endpoint for manual review.

**Evidence:** Capture before hunting: (1) Windows Event Log Security hive (C:\Windows\System32\winevt\Logs\Security.evtx) from all endpoints; (2) Registry key HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree for persistence indicators; (3) PowerShell Operational log (C:\Windows\System32\winevt\Logs\Windows PowerShell.evtx) and Script Block logging (Event ID 4104); (4) cmd.exe and powershell.exe process execution history via Get-WinEvent or Security event log 4688 with command-line arguments enabled; (5) System.evtx for task scheduler service events (Event ID 106).

**Step 2 (Detection): Enable and review Windows Event IDs 4698 (scheduled task created), 4702 (task updated), 4104 (PowerShell script block logging), and 4688 (process creation) for chains involving schtasks.exe spawning cmd.exe or powershell.exe with encoded or obfuscated arguments.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2.3 (Indicators and Artifacts)

**Controls:** NIST AU-12, NIST SI-4(1), CIS 8.5 (Log Alert Responses)

**Compensating:** Enable Script Block Logging via Group Policy (gpedit.msc: Computer Configuration > Policies > Administrative Templates > Windows PowerShell > Turn on Module Logging and Script Block Logging). For Event ID 4688, enable command-line capture via Registry (HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Audit: ProcessCreationIncludeCommandLine = 1). Without central logging, configure local Event Viewer alerts using Task Scheduler to trigger on 4698/4702/4104/4688 and email SOC. Use free tool Chainsaw (GitHub: WithSecurityLabs/chainsaw) to hunt Event Logs offline: 'chainsaw hunt -e /path/to/Security.evtx sigma-rules/ --json'.

**Evidence:** Capture before enabling: (1) Baseline of legitimate scheduled task creation patterns (last 30 days of Event IDs 4698/4702) to establish whitelist; (2) Current PowerShell execution policy and Module Logging state (Get-ExecutionPolicy -List; Get-ItemProperty 'HKLM:\SOFTWARE\Policies\Microsoft\Windows\PowerShell*'); (3) Current 4688 configuration state (reg query

HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Audit); (4) Snapshot of existing scheduled tasks (tasklist /v /fo csv) before enabling logging; (5) Sample of cmd.exe and powershell.exe process arguments from last 7 days if 4688 is already enabled.

**Step 3 (Assessment): Inventory endpoints where PowerShell execution policy permits unrestricted or bypass mode. Identify systems with internet-facing exposure where ingress tool transfer (T1105) or C2 beaconing (T1071.001) could go undetected. Confirm EDR coverage on all Windows endpoints.**

**NIST Phase:** Preparation

**Reference:** NIST 800-61r3 §2 (Preparation Phase); NIST 800-53 CA-6 (Security Assessment and Authorization)

**Controls:** NIST SI-2 (Flaw Remediation), NIST AC-6 (Least Privilege), CIS 2.3 (Address Unauthorized Software), CIS 6.1 (Enable MFA)

**Compensating:** Use native PowerShell for asset inventory: Create GPO-less inventory by executing 'Get-ExecutionPolicy' and 'Get-NetFirewallRule' via WinRM/PSRemoting to all systems (requires WinRM enabled; if not, use PSExec from PsTools: 'psexec -s \\ powershell "Get-ExecutionPolicy"'). Restrict PowerShell with Constrained Language Mode (CLM) using AppLocker (Local Security Policy > Application Control Policies > AppLocker > Executable Rules > set powershell.exe to audit/block). For internet-facing assessment: Query firewall egress rules (netsh advfirewall show rule name=all dir=out action=allow) and DNS logs to identify C2 communication patterns. Cross-reference MITRE T1071.001 TTPs (HTTP beaconing, DNS tunneling) against your DNS/proxy logs manually.

**Evidence:** Capture before assessment: (1) Baseline execution policy state across all endpoints (Get-ExecutionPolicy -List > baseline.txt from each system); (2) Current AppLocker and Windows Defender Application Control (WDAC) policy exports (Get-AppLockerPolicy -Effective); (3) Firewall egress rule baseline (netsh advfirewall show rule name=all dir=out action=allow > firewall_baseline.txt); (4) EDR/endpoint protection agent presence inventory (wmic logicaldisk get name | find EDR deployment log); (5) Network segmentation and internet-facing port inventory (netstat -ano, firewall inbound rules snapshot).

**Step 4 (Communication): Notify the SOC and IR team of the Hive0163/Interlock TTPs. If your organization received recent phishing campaigns targeting Windows users, escalate to incident response triage. Brief leadership on ransomware dwell-time risk — over one week of undetected access is the confirmed baseline for this actor.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.1 (Preparation); NIST 800-53 IR-6 (Incident Reporting)

**Controls:** NIST IR-1, NIST IR-6, CIS 6.3 (Email Protection)

**Compensating:** Create incident ticket using free ticketing system (e.g., Zulip, Taiga, or shared spreadsheet with version control). Document dwell-time risk: 'Hive0163 maintains 7+ days undetected access before data exfil — priority is detection of access after phishing email receipt (T1566.002) within 24-48 hour window.' Cross-reference phishing logs: Query mail logs for sender/domain reputation and user click-rates on malicious links. Use threat intelligence feeds (free: AlienVault OTX, abuse.ch, URLhaus) to identify IoCs from Interlock/Slopoly campaigns. Brief IT leadership using NIST 800-61r3 §6 (Post-Incident Activities) framework: quantify ransomware dwell-time financial impact if breach occurred (encryption + exfil + negotiation timeline).

**Evidence:** Capture before communication: (1) Phishing email headers and payloads (full EML files) from recent campaigns targeting Windows users; (2) Email logs showing delivery/click timestamps and user interaction (Delivered At, Opened At, Clicked At fields); (3) Baseline of known-good administrative actions during the suspect window (admin logons, patch deployments, legitimate task scheduling); (4) External threat intelligence reports on Hive0163/Interlock (IBM X-Force, CISA alerts); (5) Current incident response contact list and escalation authority (name, role, phone).

**Step 5 (Long-term): Review detection rule coverage against the full MITRE technique set listed in this campaign. Assess whether behavioral analytics can identify LLM-generated PowerShell patterns (structured comments, verbose variable names) as an anomaly signal. Update phishing simulation and user awareness programs to address T1566/T1204.001 entry vectors. Consider restricting PowerShell to Constrained Language Mode on endpoints where full functionality is not required.**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §6.1 (Lessons Learned); NIST 800-53 SA-3 (System Development Life Cycle).

**Controls:** NIST SI-4(1) (System Monitoring), NIST AT-3 (Security Awareness Training), CIS 11.4 (Automated Controls)

**Compensating:** Map Slopoly/Interlock TTPs to Sigma rules (open source, GitHub: SigmaHQ/sigma) and YARA rules (GitHub: Yara-Rules/rules) for free detection engineering. Test detection rules against Slopoly indicators using Chainsaw or bulk event log analysis. For LLM-pattern detection: Create baseline of legitimate PowerShell scripts by parsing first 100 lines of comment structure and variable naming conventions; flag scripts with verbose comments (LLM signature: 'This script is designed to...') or variable names like '$ProcessInformation', '$CommandLineArguments' (non-standard verbosity). Implement Constrained Language Mode via AppLocker or WDAC on non-admin endpoints (test in audit mode first). Conduct phishing simulation monthly targeting Windows users with Slopoly-like payloads (e.g., .LNK + scheduled task executables) and measure click-rates; adjust training for high-risk groups.

**Evidence:** Capture before remediation: (1) Existing detection rule inventory (SIGMA, custom SIEM rules, EDR detection policies); (2) Sample of 50 legitimate PowerShell scripts for baseline comparison (comment structure, variable naming); (3) Historical phishing simulation results (click-rate by department, user demographics); (4) Current AppLocker/WDAC policy state and enforcement mode (audit vs. block); (5) MITRE ATT&CK technique coverage matrix for Hive0163 campaign (map: T1566.002 → T1204.001 → T1547 → T1071.001 → T1005 → T1020.001).

## Detection Guidance

Detection should focus on behavioral chains rather than static signatures, given the bespoke nature of AI-generated tooling. Key signals: (1) schtasks.exe creating or modifying tasks outside of patch windows or known automation, filter Event ID 4698/4702 for tasks with actions invoking powershell.exe or cmd.exe with Base64-encoded or long-argument strings; (2) PowerShell script block logs (Event ID 4104) containing structured inline comment blocks, verbose variable declarations, or try/catch error handling in scripts not sourced from known repositories, this is a behavioral marker consistent with LLM-generated code; (3) outbound HTTP/S connections from powershell.exe or cmd.exe processes (T1071.001 C2 pattern), correlate with process tree to confirm parent-child anomalies; (4) Windows Restart Manager API calls (rstrtmgr.dll loaded by non-system processes) in proximity to file encryption activity, a pre-ransomware staging indicator; (5) ingress file transfers (T1105) to user-writable directories from PowerShell processes. No confirmed IOC hashes, IPs, or domains were published in the available source data at the time of this report. Monitor IBM X-Force Threat Intelligence and BleepingComputer for IOC updates as the campaign disclosure matures.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| TECHNIQUE | `T1053.005 — schtasks.exe persistence` | Slopoly uses Windows scheduled tasks for persistence. No specific task name IOC confirmed in available sources. | HIGH |
| TECHNIQUE | `T1059.001 — PowerShell execution with LLM-style code structure` | Slopoly backdoor is PowerShell-based. Behavioral indicator: verbose inline comments and structured error handling in script block logs. | HIGH |

| Type | Value | Context | Confidence |
|---|---|---|---|
| TECHNIQUE | `T1071.001 — C2 over HTTP/S` | C2 communication method identified in IBM X-Force analysis. Specific domains or IPs not confirmed in available T3 sources. | MEDIUM |
| TECHNIQUE | `T1105 — Ingress tool transfer` | Secondary payloads transferred to compromised host during dwell period. No specific file hashes confirmed in available sources. | MEDIUM |

## Framework Mappings

### MITRE-ATTACK

- **T1036.004** — Masquerade Task or Service
- **T1053.005** — Scheduled Task
- **T1140** — Deobfuscate/Decode Files or Information
- **T1036** — Masquerading
- **T1071.001** — Web Protocols
- **T1059.001** — PowerShell
- **T1105** — Ingress Tool Transfer
- **T1566** — Phishing
- **T1587.001** — Malware
- **T1486** — Data Encrypted for Impact
- **T1204.001** — Malicious Link
- **T1041** — Exfiltration Over C2 Channel

### NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **AT-2** — Literacy Training and Awareness
- **SI-8** — Spam Protection
- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **SI-10** — Information Input Validation
- **CM-3** — Configuration Change Control
- **IR-4** — Incident Handling

### OWASP-TOP10-2021

- **A03:2021** — Injection
- **A08:2021** — Software and Data Integrity Failures

### CIS-V8

- **16.10**
- **2.5**
- **2.6**

### NIST-CSF-2

- **RS.MI-01** — Incidents are contained

### HIPAA-SECURITY

- **164.308(a)(7)(ii)(A)** — Data Backup Plan
- **164.308(a)(6)(ii)** — Response and Reporting

### ISO-27001-2022

- **A.5.29** — Information security during disruption
- **A.5.34** — Privacy and protection of personal information

### SOC2-TSC

- **CC7.4** — Responds to identified security incidents

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1036.004 | Masquerade Task or Service | Defense-Evasion |
| T1053.005 | Scheduled Task | Execution |
| T1140 | Deobfuscate/Decode Files or Information | Defense-Evasion |
| T1036 | Masquerading | Defense-Evasion |
| T1071.001 | Web Protocols | Command-And-Control |
| T1059.001 | PowerShell | Execution |
| T1105 | Ingress Tool Transfer | Command-And-Control |
| T1566 | Phishing | Initial-Access |
| T1587.001 | Malware | Resource-Development |
| T1486 | Data Encrypted for Impact | Impact |
| T1204.001 | Malicious Link | Execution |

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1041 | Exfiltration Over C2 Channel | Exfiltration |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://www.bleepingcomputer.com/news/security/ai-generated-slopoly... | T3 |
| **Task Scheduler– New Vulnerabilities for schtasks.exe - Cymulate** | https://cymulate.com/blog/task-scheduler-new-vulnerabilities-for-sc... | T3 |
| **Exploiting the Windows Task Scheduler Through CVE-2019-1069** | https://www.thezdi.com/blog/2019/6/11/exploiting-the-windows-task-s... | T3 |
| **New Windows Task Scheduler Bugs Let Attackers Bypass UAC and ...** | https://thehackernews.com/2025/04/experts-uncover-four-new-privileg... | T3 |
| **Windows Task Scheduler Vulnerability Allows Attackers to Gain ...** | https://cyberpress.org/windows-task-scheduler-vulnerability/ | T3 |



Generated 2026-03-29 18:33 UTC by TJS Security Command Center