INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-22 06:46 UTC

macOS Living-Off-The-Land: Native Primitives Weaponized for Stealthy Execution and Lateral Movement

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0199
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	macOS (all enterprise deployments); Apple Terminal.app, Finder, Spotlight, AppleEventsD daemon, osascript; socat utility
Published	2026-04-21T10:00:29+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

Cisco Talos has documented a set of attack techniques targeting macOS endpoints that require no malware, attackers repurpose Apple's own built-in tools to move between systems, execute code remotely, and establish persistence. Every enterprise macOS device is a potential target, with developer and DevOps environments carrying the highest exposure given their elevated privileges and network access. Standard endpoint detection tools largely miss these techniques, meaning the threat may already be present in environments that appear clean.

Technical Analysis

Cisco Talos documented macOS living-off-the-land (LOTL) techniques exploiting Apple-signed, natively present binaries and daemons, no novel malware or CVE required. Techniques documented include: (1) osascript (T1059.002) driving Remote Apple Events over the AppleEventsD daemon for cross-host remote code execution; (2) mdfind Spotlight metadata queries (T1083) for host and file enumeration without generating network telemetry; (3) socat as a TCP relay or reverse shell conduit (T1071, T1571); (4) LaunchAgent plist-based persistence (T1543.001, T1543.004). Lateral movement paths leverage Remote Desktop (T1021.001) and SSH (T1021.005). File and artifact hiding techniques (T1564, T1564.001) reduce forensic visibility. Relevant CWEs: CWE-78 (OS Command Injection via osascript), CWE-284 (Improper Access Control over AppleEventsD), CWE-693 (Protection Mechanism Failure, EDR coverage gap). No patch exists because no vulnerability is being exploited; the attack surface is architectural. macOS-specific LOTL detection coverage

in MITRE ATT&CK and commercial EDR tooling remains materially less mature than Windows equivalents. Environments relying solely on SSH telemetry and static file scanning have no visibility into several documented attack paths. Source: Cisco Talos Blog (T3).

Action Checklist

1. Containment, Disable Remote Apple Events: On all macOS endpoints, verify System Preferences > Sharing > Remote Apple Events is disabled. If Remote Apple Events must be enabled for specific use cases, implement network-level segmentation and EDR-based monitoring to restrict AppleEventsD connections to approved internal hosts only. Use lsof or netstat to verify which hosts are connecting to TCP 3031.
2. Detection, Deploy behavioral detection rules for: osascript invocations with remote host arguments; mdfind executions outside expected developer workflows; socat process launches; LaunchAgent plist writes or modifications under ~/Library/LaunchAgents/ and /Library/LaunchAgents/. Cross-reference against MITRE ATT&CK techniques T1059.002, T1083, T1543.001, T1071, and T1571. Query EDR telemetry for socat parent-child process chains and osascript spawning network connections.
3. Eradication, Remove unauthorized LaunchAgent plists from ~/Library/LaunchAgents/ and /Library/LaunchAgents/ on all affected hosts. Terminate active socat processes not attributable to approved tooling. Review and tighten TCC (Transparency, Consent, and Control) permissions for Automation and Accessibility, which gate osascript remote execution.
4. Recovery, After remediation, validate that AppleEventsD is not accepting inbound connections on TCP port 3031 from unauthorized hosts using netstat or lsof. Confirm LaunchAgent directories contain only approved plists. Monitor for re-establishment of persistence within 72 hours post-remediation. Validate EDR telemetry is now capturing osascript, mdfind, and socat invocations.
5. Post-Incident, This campaign exposes a systemic macOS detection gap. Conduct a coverage assessment of your EDR against the documented MITRE techniques (T1059.002, T1083, T1543.001, T1543.004, T1071, T1571, T1021.001, T1021.005, T1564). Establish macOS-specific threat hunting cadence. Evaluate whether macOS endpoints are enrolled in MDM with enforced baseline configurations, including Remote Apple Events disabled by policy.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to CISO and legal counsel immediately if lsof or Unified Log analysis confirms active inbound AppleEventsD sessions from external IPs, any LaunchAgent plist contains encoded payloads with network callbacks, or if affected hosts process PII, PHI, or PCI-scoped data — triggering breach notification assessment under applicable regulations.

Recovery Notes	After eradication, maintain active monitoring of TCP port 3031, LaunchAgent directories, and TCC database state for a minimum of 72 hours, as Living-off-the-Land persistence via LaunchAgent re-registration can re-establish silently within minutes if the initial access vector (compromised user session or lateral movement path via Remote Apple Events) is not fully closed. Validate that MDM-enforced configuration profiles preventing Remote Apple Events re-enablement are applied to all endpoints before returning hosts to production. Retain all forensic artifacts — Unified Logs, TCC database snapshots, LaunchAgent copies, and network captures — for a minimum of 90 days to support any subsequent forensic review or regulatory inquiry.
Forensic Artifacts	Unified Log stream entries for com.apple.appleevents subsystem and eppc protocol — captures all inbound Remote Apple Events connection attempts, source IPs, and authenticated principals: `log show --predicate 'subsystem == "com.apple.appleevents" OR process == "osascript"' --last 14d --style json` | TCC.db SQLite databases at /Library/Application Support/com.apple.TCC/TCC.db and per-user ~/Library/Application Support/com.apple.TCC/TCC.db — records all Automation (AppleEvents) and Accessibility grants that permitted osascript remote execution, including client bundle ID, grant timestamp, and authorization value | LaunchAgent plist files from /Library/LaunchAgents/ and ~/Library/LaunchAgents/ — attacker-deployed persistence in this campaign manifests as plists with ProgramArguments containing osascript -e payloads or socat EXEC/TCP-LISTEN configurations; examine Label, ProgramArguments, and RunAtLoad keys | socat process command-line arguments captured via `ps auxwww | grep socat` or osquery processes table — in this campaign socat is used to tunnel AppleScript execution channels and establish covert C2; the cmdline will reveal EXEC:osascript, TCP4-LISTEN, or OPENSSL configurations indicating the lateral movement method | macOS Unified Log entries for mdfind process execution filtered outside Spotlight indexing context — T1083 (File and Directory Discovery) via mdfind in this campaign produces log entries showing targeted metadata queries (e.g., mdfind -name id_rsa or mdfind kMDItemKind=Script) that reveal attacker reconnaissance scope: `log show --predicate 'process == "mdfind"' --last 7d --style syslog`

Per-Action IR Details

Containment — Audit and restrict Remote Apple Events: on all macOS endpoints, verify System Preferences > Sharing > Remote Apple Events is disabled unless explicitly required. Identify and isolate any host where AppleEventsD is accepting inbound connections from non-approved sources.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST AC-3 (Access Enforcement), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Run the following on each macOS endpoint via remote shell or MDM script: `sudo systemsetup -getremoteappleevents` to confirm status. To disable: `sudo systemsetup -setremoteappleevents off`. For network-level blocking without EDR, deploy a host-based pf firewall rule blocking inbound TCP 3031: add `block in proto tcp from any to any port 3031` to /etc/pf.conf and activate with `sudo pfctl -ef /etc/pf.conf`. For fleet-scale audit without MDM, use osquery with the query: `SELECT * FROM sharing_preferences WHERE remote_apple_events = 1;` to identify exposed hosts.

Evidence: Before disabling AppleEventsD, capture: (1) active network connections on TCP port 3031 via `sudo lsof -iTCP:3031 -nP` and `sudo netstat -an | grep 3031` — document all remote IPs connected to appleeventsD; (2) full AppleEventsD process state via `sudo ps aux | grep appleevents`; (3) /var/log/system.log and /var/log/install.log filtered for 'AppleEvents' and 'eppc' entries documenting inbound connection history; (4) unified log stream entries: `log show --predicate 'subsystem == "com.apple.appleevents"' --last 7d` capturing all remote Apple Events activity; (5) network pcap of TCP 3031 traffic via `sudo tcpdump -i any -w /tmp/appleevents_capture.pcap port 3031` — run for 5 minutes before containment to capture active session patterns.

Detection — Deploy behavioral detection rules for: osascript invocations with remote host arguments; mdfind executions outside expected developer workflows; socat process launches; LaunchAgent plist writes or modifications under ~/Library/LaunchAgents/ and /Library/LaunchAgents/. Cross-reference against MITRE ATT&CK techniques T1059.002, T1083, T1543.001, T1071, and T1571. Query EDR telemetry for socat parent-child process chains and osascript spawning network connections.

NIST Phase: Detection Analysis

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

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Without EDR, deploy Endpoint Security Framework-based detection using osquery scheduled queries. Key queries: (1) osascript with network args: `SELECT pid, name, cmdline, parent FROM processes WHERE name = 'osascript' AND cmdline LIKE '%-e%' OR cmdline LIKE '%remote%';` (2) socat launches: `SELECT pid, name, cmdline, parent FROM processes WHERE name = 'socat';` (3) LaunchAgent plist writes: `SELECT * FROM file WHERE path LIKE '/Users/%/Library/LaunchAgents/%' OR path LIKE '/Library/LaunchAgents/%' AND ctime > (strftime('%s','now') - 86400);` Enable macOS Unified Logging and pipe to a local log aggregator: `log stream --predicate 'process == "osascript" OR process == "socat" OR process == "mdfind"' --style syslog`. For Sigma-based detection on collected logs, use the rule matching on osascript parent-process anomalies (osascript spawned by launchd outside /Applications) and socat TCP_LISTEN or EXEC configurations. Use YARA rules scanning ~/Library/LaunchAgents/ for plist keys containing 'osascript', 'socat', or base64-encoded payloads.

Evidence: Collect before deploying detection rules to establish a pre-rule baseline: (1) full process tree snapshot: `sudo ps auxwww > /tmp/process_baseline_$(date +%Y%m%d%H%M).txt`; (2) existing LaunchAgent inventory: `find /Library/LaunchAgents /Users/*/Library/LaunchAgents -name '*.plist' -exec ls -la {} \; > /tmp/launchagent_inventory.txt`; (3) Unified Log history for osascript, mdfind, and socat: `log show --predicate 'process == "osascript" OR process == "socat" OR process == "mdfind"' --last 14d --style json > /tmp/lol_process_history.json`; (4) network socket state for any socat-established tunnels: `sudo lsof -i -n -P | grep -E 'socat|LISTEN' > /tmp/socat_sockets.txt`; (5) TCC database snapshot documenting existing Automation and Accessibility grants: `sudo sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db 'SELECT client, service, auth_value, last_modified FROM access WHERE service IN ("kTCCServiceAppleEvents", "kTCCServiceAccessibility");'`.

Eradication — Remove unauthorized LaunchAgent plists from ~/Library/LaunchAgents/ and /Library/LaunchAgents/ on all affected hosts. Terminate active socat processes not attributable to approved tooling. Revoke any Remote Apple Events permissions granted to non-administrative users. Review and tighten TCC (Transparency, Consent, and Control) permissions for Automation and Accessibility, which gate osascript remote execution.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-3 (Malicious Code Protection), NIST CM-6 (Configuration Settings), CIS 2.3 (Address Unauthorized Software), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: For LaunchAgent removal across a fleet without MDM: create a shell script that (1) iterates `/Library/LaunchAgents` and `~/Library/LaunchAgents`, (2) computes SHA-256 of each plist and compares against a known-good allowlist, and (3) unloads and removes unknowns via `launchctl bootout gui/$(id -u) /path/to/agent.plist && rm /path/to/agent.plist`. For socat termination: `sudo pkill -9 socat && sudo pkill -9 osascript` — verify no socat re-spawning via `watch -n 5 'ps aux | grep socat'`. For TCC remediation without MDM, use the tccutil CLI: `tccutil reset AppleEvents` to revoke all Automation grants system-wide, forcing re-authorization. Verify TCC database state post-reset with the sqlite3 query from the evidence step. For Remote Apple Events permission revocation, re-run `sudo systemsetup -setremoteappleevents off` and confirm with `sudo systemsetup -getremoteappleevents`.

Evidence: Preserve before eradicating — this is forensic-critical: (1) full binary copy of all suspicious LaunchAgent plists: `cp -p /Users/*/Library/LaunchAgents/*.plist /tmp/forensic_launchagents/` — examine plist content for osascript -e payloads, socat EXEC strings, or base64-encoded commands under ProgramArguments key; (2) memory image of

active socat processes (if available via third-party tool) or at minimum the full cmdline: `sudo cat /proc/$(pgrep socat)/cmdline` equivalent: `ps -p $(pgrep socat) -o pid,ppid,comm,args`; (3) TCC database before reset: `sudo cp /Library/Application\ Support/com.apple.TCC/TCC.db /tmp/forensic_TCC_$(date +%Y%m%d).db`; (4) user-level TCC databases: `find /Users/*/Library/Application\ Support/com.apple.TCC/ -name 'TCC.db' -exec cp {} /tmp/ \;`; (5) launchd service state: `launchctl dumpstate > /tmp/launchd_dumpstate_$(date +%Y%m%d).txt` capturing all loaded agents at time of eradication.

Recovery — After remediation, validate that AppleEventsD is not accepting inbound connections on TCP port 3031 from unauthorized hosts using netstat or lsof. Confirm LaunchAgent directories contain only approved plists. Monitor for re-establishment of persistence within 72 hours post-remediation. Validate EDR telemetry is now capturing osascript, mdfind, and socat invocations.

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 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: For validation without commercial EDR: (1) AppleEventsD port check: `sudo lsof -iTCP:3031 -nP` — expected output is no results; any LISTEN or ESTABLISHED state is a re-compromise indicator; (2) LaunchAgent integrity verification using a plist allowlist: generate SHA-256 hashes of all plists against your approved baseline: `find /Library/LaunchAgents /Users/*/Library/LaunchAgents -name '*.plist' | xargs shasum -a 256 > /tmp/current_launchagents_hashes.txt` then diff against pre-incident approved baseline; (3) deploy a 72-hour osquery scheduled query pulsing every 15 minutes for new plist writes: `SELECT path, filename, ctime, mtime FROM file WHERE directory IN ('/Library/LaunchAgents/', '/Users/%/Library/LaunchAgents/') AND mtime > (strftime('%s','now') - 900);`; (4) validate telemetry by running a benign osascript invocation: `osascript -e 'display notification "EDR test"'` and confirming the process appears in your log stream or osquery results within 60 seconds.

Evidence: Document the validated clean state for post-incident record: (1) timestamped output of `sudo lsof -iTCP:3031 -nP` confirming no AppleEventsD listeners; (2) full directory listing with hashes of both LaunchAgent paths post-eradication; (3) Unified Log export for the 72-hour monitoring window filtered for osascript, socat, mdfind, and eppc: `log show --predicate 'process IN {"osascript","socat","mdfind"} OR subsystem == "com.apple.appleevents"' --start YYYY-MM-DD --end YYYY-MM-DD --style json`; (4) screenshot or exported output of TCC database showing no unauthorized AppleEvents or Accessibility grants; (5) network flow logs or tcpdump output confirming no TCP 3031 traffic during the 72-hour watch period — retain for minimum 90 days per NIST AU-11 (Audit Record Retention).

Post-Incident — This campaign exposes a systemic macOS detection gap. Conduct a coverage assessment of your EDR against the documented MITRE techniques (T1059.002, T1083, T1543.001, T1543.004, T1071, T1571, T1021.001, T1021.005, T1564). Establish macOS-specific threat hunting cadence. Evaluate whether macOS endpoints are enrolled in MDM with enforced baseline configurations, including Remote Apple Events disabled by policy.

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 SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: For EDR coverage gap assessment without commercial tooling: use the MITRE ATT&CK Navigator (free, browser-based) to map your current detection capability against T1059.002, T1083, T1543.001, T1543.004, T1071, T1571, T1021.001, T1021.005, and T1564 — color-code each as detected, partially detected, or blind. For macOS-specific threat hunting without EDR, establish a bi-weekly osquery hunt using the Santa binary authorization tool (free, Google) to allowlist approved binaries and alert on unapproved osascript, socat, or mdfind executions. For MDM baseline enforcement without Jamf/commercial MDM, use Apple Configurator 2 (free) to deploy configuration profiles that set `com.apple.RemoteAppleEvents` to disabled and enforce TCC policies via PPPC (Privacy Preferences

Policy Control) profiles. Document findings in a formal lessons-learned report per NIST IR-4 (Incident Handling), addressing why AppleEventsD was exposed and why initial detection failed.

Evidence: Produce the following post-incident documentation: (1) ATT&CK Navigator layer export mapping technique coverage gaps revealed by this campaign — attach to lessons-learned report; (2) MDM compliance report (or manual audit output) showing current Remote Apple Events configuration state across the full macOS fleet; (3) threat hunting query library document capturing all osquery, log stream, and plist inspection queries developed during this incident — version-controlled for reuse; (4) delta analysis comparing LaunchAgent inventory before and after the campaign (if pre-incident baseline existed) to establish dwell-time estimate; (5) Cisco Talos advisory reference and IOC list archived in your threat intelligence platform or shared mailbox for future hunting correlation.

Detection Guidance

Primary behavioral indicators to query in EDR and endpoint log sources: (1) osascript process launches with arguments referencing remote hostnames or IP addresses, flag any osascript invocation that includes 'remote' or contains a target host parameter; (2) mdfind executions outside standard user desktop search context, particularly in automated or scripted contexts or during off-hours; (3) socat process creation from any parent other than an explicitly approved developer tool, socat spawning a shell (sh, bash, zsh) is high-confidence malicious; (4) new or modified plist files written to ~/Library/LaunchAgents/ or /Library/LaunchAgents/, hash new plists against a known-good baseline; (5) AppleEventsD (com.apple.appleeventsd) establishing outbound or accepting inbound connections on TCP 3031 to/from non-local hosts. Unified Log (log stream / log show) on macOS captures appleeventsd and osascript activity. ESF (Endpoint Security Framework) telemetry provides process exec and file creation events for LaunchAgent writes. Hunting hypothesis: search for osascript parent processes that are non-interactive (launchd, cron, or unknown) as a high-signal lateral movement indicator.

Framework Mappings

MITRE-ATTACK

- **T1059.002** — AppleScript
- **T1072** — Software Deployment Tools
- **T1105** — Ingress Tool Transfer
- **T1083** — File and Directory Discovery
- **T1543.004** — Launch Daemon
- **T1071** — Application Layer Protocol
- **T1571** — Non-Standard Port
- **T1021.001** — Remote Desktop Protocol
- **T1021.005** — VNC
- **T1564.001** — Hidden Files and Directories
- **T1543.001** — Launch Agent
- **T1018** — Remote System Discovery
- **T1564** — Hide Artifacts

NIST-800-53R5

- **CA-7** — Continuous Monitoring

- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-10** — Information Input Validation
- **CM-7** — Least Functionality
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A03:2021** — Injection
- **A01:2021** — Broken Access Control

CIS-V8

- **2.5** — Allowlist Authorized Software
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **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(a)(1)** — Access Control

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059.002	AppleScript	Execution
T1072	Software Deployment Tools	Execution
T1105	Ingress Tool Transfer	Command-And-Control
T1083	File and Directory Discovery	Discovery
T1543.004	Launch Daemon	Persistence
T1071	Application Layer Protocol	Command-And-Control
T1571	Non-Standard Port	Command-And-Control
T1021.001	Remote Desktop Protocol	Lateral-Movement

Technique ID	Technique Name	Tactic
T1021.005	VNC	Lateral-Movement
T1564.001	Hidden Files and Directories	Defense-Evasion
T1543.001	Launch Agent	Persistence
T1018	Remote System Discovery	Discovery
T1564	Hide Artifacts	Defense-Evasion

Sources

Source	URL	Tier
Cisco Talos Blog	https://blog.talosintelligence.com/bad-apples-weaponizing-native-ma...	T3
	https://blog.talosintelligence.com/bad-apples-weaponizing-native-ma...	T3
Security Bite: Apple takes aim at cybercriminals' more ... - 9to5Mac	https://9to5mac.com/2026/03/28/security-bite-apple-takes-aim-at-cyb...	T3
Protecting against malware in macOS - Apple Support	https://support.apple.com/guide/security/protecting-against-malware...	T3
Mac Security Threats in 2025: Enterprise Defense Strategies - Jamf	https://www.jamf.com/blog/mac-security-threats-enterprise-defense-s...	T3

Generated 2026-04-22 06:46 UTC by TJS Security Command Center