INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-19 06:44 UTC

OpenClaw AI Agent Framework Patched for Credential Theft and Persistence Vulnerabilities

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0145
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	OpenClaw (AI agent framework), specific vulnerable versions not confirmed from available sources
Published	2026-05-18T17:24:59
Discovery Source	Rss

Executive Summary

Multiple critical vulnerabilities in OpenClaw, an AI agent framework seeing growing enterprise adoption, allowed attackers to steal credentials, escalate privileges, and establish persistent footholds in affected environments. The flaws are not isolated bugs; they reflect a pattern of security immaturity across the emerging category of enterprise AI agent frameworks, where speed-to-market has outpaced secure design. Organizations running unpatched OpenClaw should treat this as a systemic architecture risk, not a routine patch event, and audit how AI agent frameworks are integrated with privileged systems and credential stores.

Technical Analysis

The OpenClaw vulnerabilities span three CWE categories that, taken together, describe a near-complete attack chain: insufficiently protected credentials (CWE-522), improper privilege management (CWE-269), and improper access control (CWE-284). An attacker who exploits the credential exposure flaw gains usable secrets, API keys, service account tokens, or session credentials, without needing to crack or brute-force them. From there, the privilege management flaw allows elevation beyond the agent's intended operational scope. The access control deficiency then enables persistence, with MITRE ATT&CK techniques T1543 (Create or Modify System Process) and T1546 (Event Triggered Execution) both mapped to this incident, suggesting attackers could survive reboots and credential rotations by embedding within the agent's execution environment. The post-exploitation path maps cleanly to T1078 (Valid Accounts): once credentials are stolen and privileges elevated, the attacker operates as a legitimate identity, making detection significantly harder. The supply chain angle (T1195) is worth flagging separately. AI agent frameworks like OpenClaw often mediate access between

enterprise data, external APIs, and internal services. A compromised agent is not merely a compromised endpoint, it is a compromised intermediary with trusted relationships to multiple upstream and downstream systems. Barracuda's coverage and the arxiv taxonomy paper cited in source material both situate this incident within a broader, documented pattern of agentic AI security failures, suggesting the OpenClaw findings are representative rather than exceptional. The CVSS base score of 7.5 (High) is taken from raw source data and has not been independently confirmed against NVD or a vendor advisory. No CVE identifiers were available in the source data.

Action Checklist

1. Step 1: Assess exposure, audit your environment for any deployment of OpenClaw or dependent frameworks; check both direct installations and third-party products that may bundle OpenClaw as a component
2. Step 2: Apply vendor patches if available; check OpenClaw's official security advisory or vendor repository for patch status and version numbers. If patches are not yet released, prioritize isolation of OpenClaw instances with access to privileged credentials, internal APIs, or sensitive data stores
3. Step 3: Review credential exposure, audit where OpenClaw stores, accesses, or transmits credentials; check for plaintext secrets in config files, environment variables, or agent memory; rotate any credentials the framework has touched
4. Step 4: Audit agent privilege scope, apply least-privilege principles to all AI agent service accounts; revoke any permissions that exceed the agent's documented operational requirements
5. Step 5: Hunt for persistence indicators, review process creation logs, scheduled task registries, and service modifications in environments where OpenClaw was deployed; look for anomalous entries consistent with T1543 and T1546
6. Step 6: Update threat model, add AI agent frameworks as a privileged intermediary category in your threat register; model the lateral movement paths available from a compromised agent to connected systems
7. Step 7: Monitor developments, track for CVE assignment, updated CVSS scoring, and follow-on vendor advisories; the absence of CVE identifiers in initial reporting is not confirmation that formal assignments will not follow

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to senior IR leadership and legal/compliance if forensic review reveals that OpenClaw accessed, transmitted, or cached credentials associated with regulated data stores (PII, PHI, PCI-scoped systems), if any persistence indicators (T1543, T1546) are confirmed on systems with domain admin or cloud-root-level access, or if the team lacks the forensic tooling to perform memory acquisition and process-level credential auditing without external support.

Recovery Notes	Before restoring OpenClaw to production, verify the patched version hash matches the vendor's published release checksum and that all rotated credentials have been propagated and validated end-to-end across connected APIs, databases, and secret stores. Monitor OpenClaw service account authentication events in your identity provider (Windows Security Event ID 4624/4648 or equivalent IdP audit logs) for 30 days post-recovery for anomalous access patterns that may indicate a missed persistence mechanism or a credential that was exfiltrated and is being replayed by an external actor. Re-run the privilege scope audit (Step 4) at 7 and 30 days post-recovery to detect any permission creep reintroduced through application updates or infrastructure automation.
Forensic Artifacts	OpenClaw configuration files and .env files (e.g., ~/.openclaw/config.yaml, /etc/openclaw/settings.json, app-relative ./config/.env): primary evidence of plaintext credential storage, the central vulnerability mechanism in this advisory | Process memory dump of the OpenClaw agent process (via procdump or gcore): captures in-memory API tokens, session credentials, and agent task queues that the credential theft vulnerability would expose before they are flushed | Windows Security Event Log Event ID 4688 (Process Creation) and Sysmon Event ID 1 filtered on OpenClaw service as parent process: identifies any child processes spawned for persistence installation consistent with T1543 (Create or Modify System Process) or T1546 (Event Triggered Execution) | Scheduled task XML exports (Windows) and crontab/systemd unit files (Linux) created or modified during the OpenClaw deployment window: primary forensic evidence for T1543/T1546 persistence mechanisms that an agent framework with execution capability would leverage | Network packet capture (pcap) of OpenClaw agent outbound traffic to connected APIs and data stores: reveals plaintext credential transmission, unexpected lateral connection attempts, or exfiltration of credentials to attacker-controlled endpoints

Per-Action IR Details

Step 1: Assess exposure — audit your environment for any deployment of OpenClaw or dependent frameworks; check both direct installations and third-party products that may bundle OpenClaw as a component

NIST Phase: Detection Analysis

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

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST SI-5 (Security Alerts, Advisories, and Directives), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run `pip show openclaw` and `pip list | grep -i openclaw` on all Python environments (including virtualenvs under /opt, /srv, /home). On Windows: `Get-WmiObject Win32_Product | Where-Object {$_.Name -like '*openclaw*'}` and search for openclaw in `%APPDATA%`, `%PROGRAMFILES%`, and `%LOCALAPPDATA%`. Cross-reference against package-lock.json, requirements.txt, pyproject.toml, and Pipfile.lock for transitive dependency pulls. Use osquery: `SELECT name, version, install_time FROM python_packages WHERE name LIKE '%openclaw%';`

Evidence: Before any changes, snapshot the installed package manifest: `pip freeze > pip_snapshot_$(hostname)_$(date +%F).txt`. Capture the OpenClaw configuration directory (commonly `~/.openclaw/`, `/etc/openclaw/`, or app-relative `./config/`), including any `.env` files, `config.yaml`, or `settings.json` that define agent credentials, API keys, or data store connections. These files represent the primary credential exposure surface this vulnerability class targets.

Step 2: Apply vendor patches immediately — vendor patches have been released; prioritize patching for any OpenClaw instance with access to privileged credentials, internal APIs, or sensitive data stores

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: short-term containment and system patching

Controls: NIST SI-2 (Flaw Remediation), NIST IR-4 (Incident Handling), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: If immediate patching of production OpenClaw instances is not feasible, implement network-layer isolation first: block inbound external access to any port serving the OpenClaw agent API using host firewall rules (`iptables -A INPUT -p tcp --dport [agent_port] -s [trusted_cidr] -j ACCEPT && iptables -A INPUT -p tcp --dport [agent_port] -j DROP` or Windows Firewall equivalent). Apply patch via `pip install --upgrade openclaw` in each affected virtualenv and verify with `pip show openclaw`. Document pre- and post-patch version hashes from `pip show openclaw | grep -E 'Version|Location'` for the change record.

Evidence: Before patching, capture a full memory dump of the running OpenClaw process using `procdump -ma [pid]` (Windows) or `gcore [pid]` (Linux) to preserve any in-memory credentials or agent session tokens that the credential theft vulnerability may have exposed or cached. Record all active OpenClaw process IDs (`ps aux | grep openclaw` or `Get-Process | Where-Object {$_.Name -like '*openclaw*'}`), open network connections (`ss -tlnp | grep [pid]` or `netstat -ano | findstr [pid]`), and loaded modules before the patch is applied.

Step 3: Review credential exposure — audit where OpenClaw stores, accesses, or transmits credentials; check for plaintext secrets in config files, environment variables, or agent memory; rotate any credentials the framework has touched

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: eliminating components of the incident and credential remediation

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST IA-5 (Authenticator Management), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 5.2 (Use Unique Passwords), CIS 3.2 (Establish and Maintain a Data Inventory)

Compensating: Search OpenClaw config paths for plaintext secrets using: `grep -rE '(password|secret|api_key|token|credential)\s*[:=]\s*.+' ~/.openclaw/ /etc/openclaw/ ./config/ --include='*.yaml' --include='*.json' --include='*.env' --include='*.ini'`. On the host, dump environment variables for the OpenClaw service process: `cat /proc/[pid]/environ | tr '\0' '\n' | grep -iE 'key|token|secret|pass'` (Linux) or `(Get-Process -Id [pid]).StartInfo.EnvironmentVariables` (Windows). Cross-reference any discovered credentials against your identity provider (AD, Okta, etc.) to identify which service accounts and API tokens require immediate rotation.

Evidence: Capture `/proc/[pid]/environ` (Linux) or a process environment dump (Windows Sysinternals Process Explorer → Properties → Environment) before process termination. Review OpenClaw's agent memory log or session state files (commonly in `/tmp/openclaw-*`, `%TEMP%\openclaw-*`, or the framework's configured `workspace_dir`) for cached credentials or API tokens written to disk. Pull network capture (Wireshark or `tcpdump -i any -w openclaw_traffic.pcap host [agent_ip]`) covering any plaintext credential transmission to connected APIs or data stores.

Step 4: Audit agent privilege scope — apply least-privilege principles to all AI agent service accounts; revoke any permissions that exceed the agent's documented operational requirements

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: limiting attacker access and blast radius reduction

Controls: NIST AC-6 (Least Privilege), NIST AC-2 (Account Management), NIST IR-4 (Incident Handling), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.1 (Establish an Access Granting Process), CIS 6.2 (Establish an Access Revoking Process)

Compensating: On Linux, audit OpenClaw service account permissions: `id [openclaw_service_user]`, `sudo -l -U [openclaw_service_user]`, and review sudoers with `grep -r [openclaw_service_user] /etc/sudoers /etc/sudoers.d/`. On Windows, enumerate the service account's group memberships and token privileges: `whoami /all` (run as the service account) and `net user [account] /domain`. For database or API access, pull active permissions from your database (e.g., `SHOW GRANTS FOR 'openclaw_user'@'%';` in MySQL) and API gateway policy assignments. Revoke any admin-tier or cross-system permissions not explicitly required by OpenClaw's documented functionality.

Evidence: Before revoking permissions, document the current effective privilege set as your baseline and evidence of potential privilege escalation abuse: export AD group memberships (`Get-ADGroupMember` for each group the agent account belongs to), pull IAM policy attachments for cloud service accounts (e.g., `aws iam list-attached-user-policies --user-name [openclaw_service_user]`), and record database GRANTS. This establishes the over-permissioned state that the privilege escalation vulnerability may have been exploited to reach or exploit from.

Step 5: Hunt for persistence indicators — review process creation logs, scheduled task registries, and service modifications in environments where OpenClaw was deployed; look for anomalous entries consistent with T1543 and T1546

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: incident scope determination and IOC identification

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

Compensating: Deploy Sysmon with SwiftOnSecurity config if not present (`sysmon -accepteula -i sysmonconfig.xml`). Query Windows Security Event Log for Event ID 4688 (Process Creation) filtering on child processes spawned by the OpenClaw service process: `Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4688} | Where-Object {$_.Message -like '*openclaw*'}`. For T1543 (Create or Modify System Process): query `Get-ScheduledTask | Where-Object {$_.TaskPath -notlike '\Microsoft\*'} | Select TaskName, TaskPath, @{N='Actions';E={$_.Actions}}` for tasks created or modified during the OpenClaw deployment window. For T1546 (Event Triggered Execution): check `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` and `HKCU\Software\Microsoft\Windows\CurrentVersion\Run` for unexpected entries. On Linux: `crontab -l -u [openclaw_service_user]`, `ls -la /etc/cron.*`, `systemctl list-units --type=service --state=enabled | grep -v snap` for new or modified services.

Evidence: Collect Sysmon Event ID 1 (Process Create) logs for the full deployment lifetime of OpenClaw, focusing on child process trees rooted in the OpenClaw service. Pull Windows Scheduled Tasks XML exports: `Get-ScheduledTask | Export-Clixml scheduled_tasks_$(Get-Date -Format yyyyMMdd).xml`. On Linux, capture `/var/log/auth.log`, `/var/log/syslog`, and `journalctl -u [openclaw_service_name] --since [deployment_date]` for service installation events. These are the primary forensic artifacts for T1543 and T1546 persistence mechanisms that an AI agent framework with code-execution capability would use to survive reboot.

Step 6: Update threat model — add AI agent frameworks as a privileged intermediary category in your threat register; model the lateral movement paths available from a compromised agent to connected systems

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: lessons learned and threat model improvement

Controls: NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST IR-4 (Incident Handling), NIST SI-5 (Security Alerts, Advisories, and Directives), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Map OpenClaw's integration points using documented configuration: list every API endpoint, database connection string, secret store, and messaging queue in the OpenClaw config files. For each, apply the MITRE ATT&CK framework lateral movement paths: an agent with T1078 (Valid Accounts) access to an internal API can pivot to T1550 (Use Alternate Authentication Material) or T1210 (Exploitation of Remote Services). Document this as a new threat scenario in your risk register using the MITRE ATT&CK Navigator (free, browser-based) to visualize the kill chain from initial OpenClaw compromise through credential theft to lateral movement. This is a pattern finding — apply it to any other AI agent frameworks (LangChain, CrewAI, AutoGPT derivatives) in your inventory.

Evidence: Compile the full integration map from OpenClaw configs as the supporting evidence base for the threat model update: collect all outbound connection destinations from network capture (`tcpdump` output or firewall logs filtered on the OpenClaw service account's source IP), the list of secrets vault paths or environment variable names accessed, and any OAuth or API token scopes granted to the agent service account. This integration map constitutes the blast radius documentation and should be preserved as a formal incident record per NIST IR-5 (Incident Monitoring).

Step 7: Monitor developments — track for CVE assignment, updated CVSS scoring, and follow-on vendor advisories; the absence of CVE identifiers in initial reporting is not confirmation that formal assignments will not follow

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: intelligence sharing and continuous improvement

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Configure a free RSS or email alert for the OpenClaw vendor's GitHub releases page and security advisory channel. Set a NVD CPE watch for `openclaw` via the NVD API (`https://services.nvd.nist.gov/rest/json/cves/2.0?keywordSearch=openclaw`) with a weekly cron job that emails the output. Subscribe to CISA's Known Exploited Vulnerabilities (KEV) catalog RSS feed. Assign a specific team member to review these alerts weekly and update the internal risk register entry. If a CVE is formally assigned with CVSS >= 9.0 or KEV listing, treat it as a re-trigger of this incident's containment phase.

Evidence: Maintain a dated log of vendor advisory versions, patch release notes, and CVSS score changes as a living document tied to this incident ticket. If a CVE is subsequently assigned, immediately cross-reference its CWE classification against the vulnerability mechanisms already observed (credential storage flaws, privilege escalation paths) to determine whether additional forensic re-examination of previously analyzed systems is warranted. This advisory tracking record is required supporting evidence for any regulatory or audit inquiry under NIST IR-6 (Incident Reporting).

Detection Guidance

Focus detection efforts on the agent's process lineage and credential access patterns. In environments running OpenClaw, alert on: (1) credential file reads or environment variable access from agent processes outside expected operational windows; (2) privilege escalation events originating from service accounts associated with AI agent workloads, specifically, token impersonation or group membership changes; (3) new scheduled tasks, services, or event-triggered execution entries created by agent process identities (T1543, T1546); (4) lateral movement from agent hosts using harvested credentials, particularly against internal APIs, secrets managers (e.g., HashiCorp Vault, AWS Secrets Manager), or identity providers. In SIEM, correlate agent process activity against T1552 (Unsecured Credentials) hunting queries, look for agent processes reading .env files, credential config files, or secrets outside normal initialization sequences. EDR telemetry should flag account manipulation events (T1098) originating from agent service accounts. If your environment uses centralized secrets management, audit access logs for the service accounts OpenClaw uses; unexpected read patterns against secrets paths are a high-confidence indicator of post-exploitation credential harvesting.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Pending — refer to Dark Reading and Barracuda blog coverage for published indicators`	Source reporting references specific attack techniques (T1543, T1546, T1552) but does not publish discrete IOC values in the available source text; Dark Reading and Barracuda advisories may contain hashes, file paths, or behavioral signatures	LOW

Framework Mappings

MITRE-ATTACK

- **T1068** — Exploitation for Privilege Escalation
- **T1078** — Valid Accounts
- **T1543** — Create or Modify System Process
- **T1548** — Abuse Elevation Control Mechanism
- **T1546** — Event Triggered Execution
- **T1552** — Unsecured Credentials
- **T1098** — Account Manipulation
- **T1195** — Supply Chain Compromise

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **AC-2** — Account Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CM-6** — Configuration Settings
- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

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

CIS-V8

- **5.2** — Use Unique Passwords
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **6.3** — Require MFA for Externally-Exposed Applications
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

HIPAA-SECURITY

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1078	Valid Accounts	Defense-Evasion
T1543	Create or Modify System Process	Persistence
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation
T1546	Event Triggered Execution	Privilege-Escalation
T1552	Unsecured Credentials	Credential-Access
T1098	Account Manipulation	Persistence
T1195	Supply Chain Compromise	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/application-security/claw-chain-vulnera...	T3
A Systematic Taxonomy of Security Vulnerabilities in the ...	https://arxiv.org/abs/2603.27517	T2

Source	URL	Tier
OpenClaw security risks: What security teams need to know ...	https://blog.barracuda.com/2026/04/09/openclaw-security-risks-agent...	T3
OpenClaw Security Risks: From Vulnerabilities to Supply ...	https://www.sangfor.com/blog/cybersecurity/openclaw-ai-agent-securi...	T3
Personal AI Agents like OpenClaw Are a Security Nightmare	https://blogs.cisco.com/ai/personal-ai-agents-like-openclaw-are-a-s...	T3

Generated 2026-05-19 06:44 UTC by TJS Security Command Center