tinexta
defence

AI autonoma, sicurezza fragile: il caso OpenClaw

#TinextaDefenceBusiness

AI4Cyber

AI4Cyber studio, è il nuovo spazio di ricerca applicata di Tinexta Defence, dedicato alla ricerca e applicazione dell'**Intelligenza Artificiale in ambito cybersecurity**.

Al suo interno, diversi specialisti con competenze ed esperienze eterogenee collaborano per affrontare le sfide poste dal panorama delle minacce informatiche, accompagnare i clienti nell'adozione dell'AI nei propri processi aziendali e condividere le attività di ricerca con la comunità tecnico-scientifica.

Il team cura l'**intero ciclo di sviluppo dei sistemi basati su AI**: dalla raccolta e dal preprocessing dei dati, all'addestramento e validazione dei modelli, fino al deployment in produzione.

Le soluzioni proposte si fondano su tecnologie avanzate di **Machine Learning**, **Deep Learning** e **Large Language Models**, e possono essere personalizzate in base alle specifiche esigenze del cliente.

L'AI Team è costantemente impegnato in attività di ricerca e sperimentazione, con un'attenzione particolare agli **aspetti etici**, alla **trasparenza** e alla **tutela della privacy**.

L'obiettivo è proporre soluzioni innovative che siano in linea con i principi di **equità, inclusività e rispetto dei diritti fondamentali**.

Sommario

Abstract

Il presente report analizza la classe degli agenti AI autonomi self-hosted quale categoria emergente di rischio informatico, utilizzando come caso studio principale OpenClaw — un sistema open-source che nel periodo gennaio-marzo 2026 rappresenta il primo caso documentato su scala globale di compromissione diffusa in ambienti agentici autonomi self-hosted, per numero di istanze esposte, CVE formali e supply chain compromessa, evidenziando criticità strutturali ancora poco affrontate dai modelli di sicurezza tradizionali. L'analisi evidenzia come il modello di rischio degli agenti AI autonomi presenti caratteristiche qualitativamente diverse da quello delle applicazioni convenzionali: combinazione di accesso privilegiato al sistema operativo, elaborazione di contenuto non verificato, capacità di comunicazione esterna e memoria persistente inter-sessione genera una superficie di attacco che non può essere ridotta esclusivamente tramite patch o configurazione.

Lo studio documenta oltre 53 vulnerabilità formali (CVSS medio 7.3), decine di migliaia di istanze esposte su internet pubblico (con stime che variano tra 21.000 e 135.000 a seconda della metodologia di scansione adottata), una compromissione sistematica del marketplace di skill con stime di pacchetti malevoli che variano tra 341 e oltre 1.100 a seconda della fonte e del perimetro di analisi, e una serie di incidenti operativi che dimostrano la concreta possibilità di sfruttamento delle vulnerabilità identificate. Vengono analizzate le tecniche di attacco specifiche per sistemi agentici — incluse prompt injection indiretta, time-shifted attack e infostealer AI-aware — nonché le risposte istituzionali da parte di governi, grandi aziende tecnologiche e organizzazioni di sicurezza.

Lo studio propone indicatori di compromissione e raccomandazioni di detection articolati su tre livelli — rete, endpoint e comportamento — e formula raccomandazioni operative per le organizzazioni che devono gestire deployment agentici in assenza di standard di sicurezza maturi per questa categoria di sistemi.

Autori:

- Antonio Addabbo: AI4Cyber
- Alberico Ciriello: Malware Lab
- Maurizio Stoisman: Compliance

Introduzione

Gli agenti AI autonomi self-hosted rappresentano una discontinuità nel modello di sicurezza costruito negli ultimi decenni. A differenza dei modelli linguistici tradizionali, operano con accesso privilegiato al sistema operativo, mantengono memoria persistente tra le sessioni, comunicano con servizi esterni ed elaborano contenuti provenienti da fonti non verificate. Questa combinazione di capacità, necessaria al funzionamento dell'agente, genera una superficie di attacco qualitativamente diversa da quella delle applicazioni convenzionali.

Questo studio non analizza una singola vulnerabilità, ma una classe emergente di rischi legati all'autonomia operativa degli agenti AI. Il caso OpenClaw, agente AI open-source la cui adozione virale nel periodo gennaio – marzo 2026 ha prodotto oltre cinquanta CVE formali, decine di migliaia di istanze esposte e una supply chain sistematicamente compromessa, viene utilizzato come evidenza empirica per inquadrare il modello di rischio strutturale e fornire strumenti concreti di analisi, detection e risposta.

Il fenomeno non è isolato: secondo un'indagine Gartner su 302 CISO di grandi organizzazioni (ricavi >$250M), il 59% dichiara evidenza o sospetto di automazione AI non autorizzata [23]. La ricerca si basa sull'analisi di oltre sessanta fonti primarie ovvero advisory di sicurezza, audit indipendenti, report di threat intelligence, disclosure CVE, documentazione ufficiale, le quali sono state prodotte nel periodo novembre 2025 – marzo 2026.

1. L'Agente AI autonomo: architettura e modello di rischio

1.1 Definizione e caratteristiche operative

Un agente AI autonomo è un sistema software che utilizza un LLM come motore decisionale per interpretare obiettivi espressi in linguaggio naturale e scomporli in sequenze di azioni eseguibili. Le caratteristiche distintive rispetto a un chatbot tradizionale sono:

- Accesso privilegiato alle risorse di sistema: l'agente opera con i permessi dell'utente che lo esegue, includendo accesso completo al filesystem, esecuzione di comandi shell, lettura e scrittura di e-mail e calendari, controllo del browser;
- Integrazione OAuth con servizi enterprise: l'agente mantiene token di lunga durata verso piattaforme esterne (Google Workspace, Slack, Discord, GitHub, Microsoft 365), ereditando i relativi privilegi;
- Elaborazione di contenuto non verificato: l'agente legge e-mail, pagine web, documenti e messaggi da canali controllati da terze parti, potenzialmente ostili;
- Memoria persistente inter-sessione: contesto, istruzioni e storico sono conservati su disco tra una sessione e l'altra, rendendo possibili attacchi a detonazione differita;
- Estensione delle capacità tramite skill: funzionalità aggiuntive distribuite come pacchetti installabili da marketplace pubblici, eseguite con i medesimi privilegi dell'agente principale.

1.2 La trifecta letale e il quarto fattore

Simon Willison ha formalizzato la struttura di rischio degli agenti autonomi con il concetto di "trifecta letale" [1]: accesso a dati privati e credenziali, capacità di comunicazione con servizi esterni, ed elaborazione di contenuto proveniente da fonti non verificate. Questa combinazione non è un difetto implementativo ma è la condizione necessaria perché il sistema sia funzionale. Riteniamo che ridurre uno dei tre elementi possa degradare le capacità operative nella stessa misura in cui riduce la superficie di attacco.

Palo Alto Networks, invece, ha identificato un quarto fattore amplificante: la memoria persistente, la quale trasforma gli attacchi da exploit puntuali a exploit con stato, ritardati nel tempo. [2]

Trend Micro ha successivamente analizzato il “fenomeno” OpenClaw, fornendo un punto di vista interessante sulle implicazioni di sicurezza. La minaccia più importante è diventata l’Indirect Prompt Injection, perché OpenClaw può essere connesso a canali di comunicazione quali Whatsapp o Telegram al posto di un utente. In uno scenario in cui gli viene inviato un messaggio con un prompt nascosto (caratteri invisibili ad esempio), potrebbe eseguire un comando malevolo senza che l’utente ne sia consapevole. Il ricercatore di Trend Micro afferma che il futuro dell’intelligenza artificiale è in locale e agentico, ma attualmente è incredibilmente fragile. [3]

1.3 Il gap degli strumenti di sicurezza esistenti

I sistemi di protezione attualmente in uso nelle organizzazioni presentano una lacuna strutturale rispetto al comportamento agentico, ovvero alla capacità degli agenti AI di agire autonomamente, orchestrare azioni su più sistemi e interpretare contenuto esterno come istruzione. CrowdStrike [25] e Trend Micro [24] documentano indipendentemente lo stesso fenomeno: un EDR rileva i processi associati all'agente ma non può interpretarne il comportamento semantico; un network monitoring vede le API call verso servizi legittimi ma non distingue l'automazione autorizzata dalla compromissione; un identity management registra i grant OAuth ma non classifica le connessioni di un agente AI come anomalie. Il risultato è che un agente compromesso opera, dal punto di vista degli strumenti di difesa tradizionale, in modo indistinguibile da un'automazione legittima.

Cisco State of AI Security 2026 documenta che solo il 29% delle organizzazioni si dichiara preparato a proteggere deployment di AI agentica, in assenza di standard tecnici maturi e framework di governance specificamente progettati per questa categoria di sistemi. [4]

OpenClaw viene utilizzato come caso studio nei capitoli seguenti non perché sia un caso eccezionale, ma perché è il primo caso in cui questo modello di rischio si è manifestato su scala misurabile. I dati quantitativi (CVE, istanze esposte, percentuali di supply chain compromessa) non sono l'oggetto dell'analisi, sono la prova che il modello descritto in questo capitolo ha già prodotto conseguenze reali.

2. Il caso OpenClaw: storia e vulnerabilità

2.1 Profilo del sistema

OpenClaw è un agente AI autonomo open-source self-hosted, sviluppato dall'austriaco Peter Steinberger, fondatore di PSPDFKit, come prototipo personale e rilasciato pubblicamente nel novembre 2025 sotto il nome Clawdbot. L'architettura è composta da quattro componenti principali che interagiscono e definiscono collettivamente la superficie di attacco del sistema.

Il **Gateway** è il componente centrale: un processo Node.js che espone un'API WebSocket sulla porta 18789 (default), gestisce l'autenticazione, mantiene le sessioni, orchestra i tool disponibili e instrada le istruzioni verso l'LLM configurato. Il gateway è registrato come processo persistente — LaunchAgent su macOS, systemd su Linux, Scheduled Task su Windows — e sopravvive ai riavvii del sistema. Opera quindi con i permessi dell'utente che lo esegue, il che in molti deployment equivale a privilegi amministrativi completi.

I **Nodi** sono host di esecuzione remota, tipicamente macchine macOS, che si registrano al gateway ed espongono capacità operative: esecuzione di comandi shell, accesso al filesystem, controllo del browser, lettura di notifiche. Un'istanza OpenClaw può orchestrare più nodi simultaneamente.

I **Canali** sono le interfacce di comunicazione attraverso cui l'utente interagisce con l'agente: WhatsApp, Telegram, Discord, Slack, Signal, iMessage, Microsoft Teams. Ogni messaggio ricevuto da un canale viene elaborato dall'LLM come potenziale istruzione, inclusi i messaggi provenienti da mittenti non verificati, se la configurazione lo consente.

Le **Skill** sono pacchetti modulari che estendono le capacità dell'agente, distribuiti attraverso ClawHub, il marketplace pubblico del progetto. Una skill è tecnicamente una directory contenente un file SKILL.md con istruzioni in linguaggio naturale, script opzionali e metadati. Le skill vengono eseguite con i medesimi privilegi del gateway, non esiste isolamento tra il processo principale e le skill installate. Questo formato annulla strutturalmente la distinzione tra documentazione ed esecuzione: un file Markdown è culturalmente associato a contenuto statico, ma in questo contesto costituisce un installer con privilegi operativi completi.

L'LLM configurabile (Claude, GPT-4, DeepSeek, Gemini o modelli locali) riceve il contesto completo della sessione, inclusi i contenuti elaborati dai canali, le istruzioni delle skill installate e la memoria persistente inter-sessione. Non esiste separazione tra il contesto fidato (istruzioni dell'utente) e il contesto non fidato (contenuto esterno elaborato dall'agente): tutto confluisce nello stesso spazio di ragionamento del modello.

2.2 Difetti architetturali

Un'analisi di Immersive Labs [5] su OpenClaw evidenzia un insieme di problematiche di sicurezza e architetturali che rendono la superficie di attacco molto ampia:

- Da inizio 2026 sono state pubblicate diverse CVE critiche (vedi CVE-2026-25253) e sono seguiti anche diversi altri advisory che suggeriscono una base di codice che necessita di costante revisione.
- Molte skills di OpenClaw (estensioni che sono distribuite da ClawHub) sono risultate come malware mascherato come software legittimo.
- Migliaia di istanze di OpenClaw sono risultate esposte e configurate con impostazioni di default. Ad esempio, il deployment tramite Docker è configurato con impostazioni base su 0.0.0.0:18789. Ciò significa che il gateway è in ascolto su tutte le interfacce di rete ed esposto su Internet.

- Assenza di validazione dell'header Origin nelle connessioni WebSocket: qualsiasi sito web può stabilire una connessione al gateway locale senza restrizioni di origine. [6]
- agents.defaults.sandbox.mode = off: il sandboxing Docker non è attivo salvo configurazione esplicita. [5]
- Archiviazione credenziali in chiaro: OAuth token e chiavi API sono conservati in file JSON e Markdown non cifrati in ~/.openclaw/. [5]
- Sessione DM globale: in ambienti multiutente, i segreti di una sessione privata sono accessibili ad altri utenti dello stesso bot (parametro session.dmScope). [7]

La tabella seguente classifica gli elementi dell'architettura OpenClaw per tipo di rischio, distinguendo tra elementi intrinseci (legati alla natura stessa degli agenti autonomi), architetturali (dipendenti dalle scelte di design del sistema) e implementativi (difetti di codice o configurazione correggibili).

Elemento architetturale	Descrizione	Rischio associato	Tipo
Esecuzione autonoma di comandi	L'agente esegue azioni sul sistema operativo senza supervisione continua	Abuso operativo, lateral movement	**Intrinseco**
Elaborazione di contenuto non verificato	E-mail, pagine web, documenti da fonti potenzialmente ostili vengono interpretati come istruzioni	Prompt injection indiretta, esfiltrazione	**Intrinseco**
Memoria persistente inter-sessione	Contesto e istruzioni conservati su disco tra sessioni successive	Time-shifted attack, C2 persistente	**Intrinseco**
Marketplace skill aperto (ClawHub)	Skill distribuite come pacchetti installabili [8]	Supply chain compromise, infostealer	**Architetturale**
Integrazione OAuth con servizi enterprise	Token di lunga durata verso Google Workspace, Slack, GitHub, M365	Account takeover, credential theft	**Architetturale**

Binding su 0.0.0.0:18789 (Docker)	Gateway esposto su tutte le interfacce nei deployment Docker	Accesso non autorizzato, RCE	**Implementativo**
Credenziali in chiaro (~/.openclaw/)	OAuth token e chiavi API in file JSON/Markdown non cifrati [5]	Credential theft, account takeover	**Implementativo**
Sandboxing disattivo per default	agents.defaults.sandbox.mode = off [5]	Escape da container, esecuzione arbitraria	**Implementativo**
Sessione DM globale	Segreti di sessione privata accessibili ad altri utenti dello stesso bot [7]	Cross-session data leak	**Implementativo**

La classificazione per tipo (intrinseco, architetturale, implementativo) serve a distinguere cosa può essere corretto con patch e governance da cosa è strutturalmente legato alla natura degli agenti autonomi. Per la valutazione operativa degli stessi rischi in termini di impatto e priorità di mitigazione si rimanda alla matrice in Appendice A.

2.3 Catalogo delle vulnerabilità principali

La tabella seguente raccoglie le vulnerabilità principali identificate nel periodo di analisi, selezionate per rilevanza tecnica, impatto operativo e disponibilità di evidenza verificabile. Per ciascuna voce sono indicati l'identificatore o il nome, il punteggio CVSS (ove presente una CVE), la categoria di attacco, la descrizione tecnica della vulnerabilità e la versione in cui è stata corretta. Esiste un tracker, **jgamblin/OpenClawCVEs** [18], che registra numerose CVE su OpenClaw con criticità elevata. Le voci all'interno della tabella seguente non saranno aggiornate durante la pubblicazione del seguente report, ma è possibile vedere le nuove direttamente dal riferimento. Queste vulnerabilità hanno un grande valore analitico per un threat model.

Vulnerabilità	CVSS / Criticità	Categoria	Descrizione tecnica	Stato
CVE-2026-25253	8.8	RCE WebSocket	La Gateway Control UI di OpenClaw accetta il parametro gatewayUrl dalla query string senza validazione e stabilisce automaticamente una connessione WebSocket verso l'URL specificato, trasmettendo il token di autenticazione nel payload di connessione (CWE-669). Un attaccante può sfruttare questa flaw per esfiltrare il token tramite Cross-Site WebSocket Hijacking (CSWSH), quindi invocare in sequenza: 3. exec.approvals.set con parametro ask: "off" per disabilitare ogni approvazione umana; 4. config.patch con tools.exec.host: "gateway" per forzare l'esecuzione dei comandi direttamente sull'host, bypassando il container Docker. Risultato: RCE completa con i privilegi del processo gateway in millisecondi, senza ulteriore interazione della vittima. PoC pubblico: [20]	**Corretto in v2026.1.29**
CVE-2026-25593	8.4	RCE no-auth locale	Un client locale non autenticato può invocare l'API WebSocket del gateway tramite config.apply per scrivere configurazioni arbitrarie su disco. Valori cliPath malevoli non sono vincolati a nomi/percorsi di eseguibili sicuri e vengono successivamente utilizzati durante il command discovery tramite invocazione shell, consentendo OS command injection (CWE-78) come utente del processo gateway. Nessuna autenticazione richiesta. GitHub Security Advisory GHSA-g55j-c2v4-pjcg [18]	**Corretto in v2026.1.20**
CVE-2026-24763	8.8	Docker sandbox escape	Vulnerabilità di OS command injection (CWE-78) nel meccanismo di esecuzione Docker sandbox di OpenClaw. La funzione buildDockerExecArgs interpola la variabile d'ambiente PATH fornita dall'utente direttamente in una stringa di comando shell (export PATH="${user_input}: $PATH") senza sanitizzazione. Un utente autenticato con controllo sulle variabili d'ambiente può iniettare comandi arbitrari eseguiti nel contesto del container. N.B.: non si tratta di un sandbox escape diretto verso l'host, ma di command injection all'interno del container, con possibile escalation in presenza di configurazioni Docker non sicure (es. socket montato). GitHub Security Advisory GHSA-mc68-q9jw-2h3v: [18]	**Corretto in v2026.1.29**
CVE-2026-25157	7.5	Command injection macOS	Si tratta di due vulnerabilità correlate di OS command injection (CWE-78) nel componente SSH dell'applicazione macOS menubar di OpenClaw (modalità Remote/SSH only): ▪ La funzione sshNodeCommand costruisce uno script shell senza effettuare l'escaping del percorso di progetto fornito dall'utente; quando il comando cd fallisce, il percorso non sanitizzato viene interpolato in un'istruzione echo, consentendo l'esecuzione arbitraria di comandi sull'host SSH remoto. ▪ La funzione parseSSHTarget non valida correttamente le stringhe di target SSH per fare in modo che non inizino con un trattino; un target malevolo come -oProxyCommand=... viene interpretato come flag di configurazione SSH, consentendo l'esecuzione di comandi sulla macchina locale. GitHub Security Advisory GHSA-q284-4pvr-m585: [18] Non interessati: CLI (npm), gateway web, app iOS/Android.	**Corretto in v2026.1.29**

ClawJacked (Oasis)	**Alta**	WebSocket hijack	Catena di vulnerabilità nel core del gateway OpenClaw (no plugin/estensioni). JavaScript su qualsiasi sito web visitato dalla vittima può aprire una connessione WebSocket verso localhost (non bloccata dalle policy cross-origin del browser), effettuare brute-force della password del gateway (rate limiter esonera loopback con centinaia di tentativi/s senza throttling né logging), e registrarsi automaticamente come dispositivo trusted (auto- approvazione da localhost senza prompt utente). Post- autenticazione: sessione admin con accesso completo all'agente, dump configurazione, enumerazione nodi, lettura log, esecuzione comandi. Nessuna interazione utente è richiesta oltre alla visita del sito. [6]	**Corretto in v2026.2.25**
CVE-2026-26322	**7.6**	SSRF	Server-Side Request Forgery (CWE-918) nel componente Gateway tool. Il client WebSocket del gateway accettava override gatewayUrl forniti dalle invocazioni dei tool senza validazione o allowlisting, consentendo connessioni in uscita verso target arbitrari (servizi interni, localhost, endpoint di metadati cloud). Richiede la capacità di invocare tool con override gatewayUrl (operatori autenticati, automazione trusted oppure deployment con tool calls esposti a non-operatori). GitHub Security Advisory GHSA-g6q9-8fvw-f7rf: [18]	**Corretto in v2026.2.14**
CVE-2026-25475	**6.5**	LFI / Path Traversal	Vulnerabilità di Path Traversal (CWE-22) nella funzione isValidMedia() in src/media/parse.ts. Il componente non valida correttamente i percorsi file forniti, accettando percorsi assoluti, riferimenti alla home directory e sequenze di traversal (../). Un agente può leggere file arbitrari sul filesystem (es. chiavi SSH, /etc/passwd, file di configurazione) emettendo output nel formato MEDIA:/percorso/al/file, con conseguente esfiltrazione dei dati verso l'utente o il canale di comunicazione. GitHub Security Advisory GHSA-r8g4-86fx-92mq: [18]	**Corretto in v2026.1.30**
CVE-2026-26319	**7.5**	Auth bypass webhook	Assenza di autenticazione (CWE-306) nell'handler webhook Telnyx del plugin opzionale @openclaw/voice- call. La funzione TelnyxProvider.verifyWebhook() effettua un "fail open" quando telnyx.publicKey (o la variabile d'ambiente TELNYX_PUBLIC_KEY) non è configurata: qualsiasi richiesta HTTP POST all'endpoint webhook viene trattata come evento Telnyx legittimo, senza verifica della firma Ed25519. Impatto limitato ai deployment con il plugin Voice Call installato, abilitato e con l'endpoint raggiungibile dall'attaccante. GitHub Security Advisory GHSA-4hg8-92x6-h2f3: [18]	**Corretto in v2026.2.14**
CVE-2026-26326	**5.3**	Info Disclosure	Inserimento di informazioni sensibili nei dati trasmessi dall'endpoint skills.status (CWE-201, più preciso rispetto a CWE-200 presentato nell'Advisory). In versioni precedenti alla 2026.2.14, l'endpoint restituisce i valori di configurazione raw risolti nel campo configChecks[].value per i percorsi requires.config delle skill, accessibili da client con scope operator.read. Ciò consente l'esfiltrazione di segreti di configurazione, inclusi token Discord e altre credenziali di servizi integrati, da parte di utenti con permessi di sola lettura. GitHub Security Advisory GHSA-8mh7-phf8-xgfm: [18]	**Corretto in v2026.2.14**
Log poisoning WebSocke	**Media**	Indirect prompt injection	Vulnerabilità di indirect prompt injection tramite log poisoning via WebSocket. Se un'istanza di OpenClaw fosse esposta su Internet, un attaccante potrebbe iniettare codice arbitrario nei log dell'agente attraverso messaggi WebSocket. Poiché l'agente elabora i propri log come contesto, il contenuto malevolo viene interpretato come istruzione operativa, consentendo la manipolazione del comportamento dell'agente. [9]	**Corretto in v2026.2.13**

session.dmScope flaw	**Alta**	Cross-session data leak	Cross-session data leak nel contesto DM dell'agente (CWE-488). Il parametro session.dmScope in configurazione default (main) condivide un'unica sessione a lunga durata tra tutti i DM. In deployment multiutente, variabili d'ambiente, chiavi API e cronologia delle conversazioni caricate nella sessione DM "privata" risultano accessibili a chiunque possa messaggiare il bot. [7]	**Mitigato via hardening della configurazione**
Group chat isolation	**Alta**	Improper Access Control	Assenza di tool isolation nelle sessioni di gruppo (CWE-862). I workspace non sono segregati per gruppo: file e dati di una sessione possono riaffiorare in un'altra. Qualsiasi membro del gruppo può invocare tutti i tool concessi all'agente senza restrizioni per utente. Limitazione architetturale riconosciuta da OpenClaw. [7]	**Mitigato via hardening della configurazione**

3. Vettori di compromissione: tecniche e incidenti

3.1 Vettori di compromissione

1Password ha documentato come, nell'ecosistema di OpenClaw, il formato SKILL.md trasformi la documentazione in un vettore di esecuzione, poiché le istruzioni contenute nei file markdown vengono seguite con la stessa fiducia riservata alla documentazione ufficiale, innescando catene d'installazione malevole mascherate da legittimi requisiti.

Come osservato da Meller, autore dell'articolo, questo vettore non è specifico di OpenClaw ma è individuabile in qualsiasi agent ecosystem che adotti il formato Agent Skills standard (SKILL.md + script opzionali), inclusi i sistemi documentati da Anthropic e OpenAI. [10]

3.2 Tecniche di attacco specifiche per sistemi agentici

3.2.1 Prompt injection indiretta (IDPI)

Nella prompt injection indiretta le istruzioni malevole sono incorporate nel contenuto che l'agente elabora: e-mail ricevute, pagine web visitate, documenti aperti. PromptArmor, invece, ha dimostrato un vettore di esfiltrazione zero-click tramite le anteprime link di Telegram e Slack: l'agente viene manipolato per costruire un URL attaccante con dati sensibili come query parameter. Il fetch dell'anteprima avviene automaticamente, senza interazione utente. [11]

3.2.2 Indirect Prompt Injection via messaggistica

Un vettore specifico di IDPI sfrutta la natura “always-on” dell'agente e la sua integrazione diretta con canali di messaggistica come WhatsApp e Telegram. Tucci (Trend Micro) descrive uno scenario esemplificativo: un messaggio apparentemente innocuo come "Good morning! Check-out this recipe" viene ricevuto dall'agente, che lo elabora automaticamente, ma il messaggio potrebbe contenere testo nascosto (caratteri invisibili o link) con istruzioni malevole. Nel caso documentato, vi è il comando di comprimere il contenuto della directory ~/.ssh e inviarlo tramite POST a un IP controllato dall'attaccante. Poiché l'agente opera con i privilegi dell'utente (spesso equivalenti a root), il comando viene eseguito senza alcuna interazione attiva da parte della vittima. Le pericolosità sono tre: nessun click richiesto all'utente finale (l'agente elabora il messaggio autonomamente), sfruttamento di canali percepiti come personali e tipicamente meno monitorati, la natura dell'attacco che non richiede vulnerabilità software e non è rilevabile da scanner tradizionali. [3]

3.2.3 Time-shifted prompt injection

L’espansione della trifecta ad una quarta capacità, la memoria persistente, trasforma gli attacchi da exploit puntuali a exploit con stato, ritardati nel tempo. Palo Alto Networks descrive come input non fidati, apparentemente benigni in isolamento, possano essere scritti nella memoria a lungo termine dell'agente e successivamente assemblati in un set di istruzioni eseguibili. Questo meccanismo abilita quella che gli autori definiscono "logic bomb-style activation": l'exploit viene creato nella fase di ingestion ma viene detonato solo quando lo stato interno dell'agente, i suoi obiettivi o la disponibilità di strumenti soddisfano le condizioni codificate.

Come sintetizzato dagli autori: con la memoria persistente, gli attacchi non sono più exploit point-in-time ma diventano attacchi a esecuzione differita. [2] Riteniamo che le tempistiche in cui viene distribuito questo vettore lo rendono particolarmente insidioso per i SIEM tradizionali.

3.2.4 HEARTBEAT.md – C2 injection

L'heartbeat di OpenClaw è un processo proattivo che si attiva a intervalli regolari (il default è di 30 minuti), legge le istruzioni contenute nel file HEARTBEAT.md nel workspace dell'agente e le esegue senza richiedere input dall'utente. Quattro proprietà architetturali lo rendono un bersaglio ideale per stabilire persistenza: è ciclico e automatico, opera in background, le sue risposte possono essere soppresse tramite i token HEARTBEAT_OK e NO_REPLY (rendendolo invisibile all'utente) e viene spesso configurato con modelli più economici e meno capaci per contenere i costi. Rehberger [13] ha dimostrato come sfruttare queste proprietà per costruire un canale di command and control interamente basato su prompt. Tramite indirect prompt injection, veicolata ad esempio attraverso un'e-mail o un documento elaborato dall'agente, l'attaccante induce la scrittura di istruzioni malevole nel file HEARTBEAT.md. Da quel momento, ad ogni ciclo di heartbeat, l'agente compromesso contatta un server C2 (Agent Commander) per ricevere nuovi obiettivi formulati in linguaggio naturale: ricognizione dell'host, esfiltrazione di screenshot della posta, monitoraggio di repository o qualsiasi altro compito l'agente sia in grado di svolgere con i propri privilegi. Poiché il modello più debole assegnato all'heartbeat effettua meno controlli, l'injection ha maggiore probabilità di successo e, poiché l'agente esegue ripetutamente le stesse istruzioni malevole, il comportamento si normalizza nel tempo, riducendo ulteriormente la possibilità che venga riconosciuto come anomalo. Rehberger, quindi, sintetizza un cambio di paradigma, ovvero non è necessario installare malware tradizionale poiché l'agente stesso diventa il vettore di esecuzione, mentre i comandi sono prompt. [13]

3.2.5 Evoluzione verso il social engineering

OpenAI ha documentato come gli attacchi più efficaci stiano assumendo sempre più le caratteristiche del social engineering. I primi attacchi di prompt injection erano relativamente diretti, ad esempio, l'inserimento di istruzioni in una pagina Wikipedia visitata dall'agente. Le varianti attuali, invece, sfruttano invece contenuto piuttosto plausibile, progettato per risultare credibile all'interno del flusso operativo dell'agente. La rilevanza di questa osservazione è amplificata dalla fonte, il quale, non è un vendor che opera nell'ambito della sicurezza ma, come sviluppatore di uno dei principali LLM, ha un reale interesse commerciale nel segnalare rischi relativi ad OpenClaw.

La superficie di attacco si amplia proporzionalmente all'autonomia operativa dell'agente, poiché ogni capacità aggiuntiva (navigazione web, recupero informazioni, esecuzione di azioni per conto dell'utente) introduce un nuovo vettore. OpenAI riconosce esplicitamente che la prompt injection è una minaccia persistente ed evolutiva che può essere mitigata e gestita, ma non risolta in modo definitivo, analogamente alle truffe online che colpiscono quotidianamente le persone. L'approccio difensivo proposto si concentra, pertanto, sul contenimento delle conseguenze, ovvero, progettando il sistema affinché l'impatto di una manipolazione riuscita resti circoscritto, anche quando l'attacco supera le difese di primo livello. [12]

3.3 Incidenti documentati

La tabella seguente raccoglie incidenti documentati nel periodo novembre 2025 – marzo 2026, classificati in diverse categorie: **autonomia incontrollata** (l'agente esegue azioni non autorizzate per comportamento inatteso o malfunzionamento), **compromissione esterna** (exploit di vulnerabilità, prompt injection o supply chain da parte di attaccanti) e **data breach** (esposizione di dati per difetti architetturali del sistema).

Incidente	Tipo	Fonte	Descrizione
Cancellazione casella e-mail	Autonomia incontrollata	X [26]	Summer Yue (Safety and alignment at Meta Superintelligence): il suo agente, istruito con direttiva esplicita "confirm before acting", ha cancellato autonomamente oltre 200 e-mail dalla casella principale. La causa tecnica è stata la context window compaction: l'inbox reale, molto più grande di quella di test, ha superato il limite dei token, per cui, l'agente ha compresso la cronologia perdendo l'istruzione di sicurezza originale. Yue ha tentato di interrompere l'operazione inviando in sequenza "Do not do that", "Stop don't do anything" e "STOP OPENCLAW". Tutti i comandi sono stati ignorati, rendendo possibile l'interruzione solo accedendo fisicamente alla macchina.

Incidente	Tipo	Fonte	Descrizione
Spam massivo su iMessage	Autonomia incontrollata	Bloomberg [14]	Chris Boyd, dopo aver concesso a OpenClaw accesso a iMessage per automatizzare un digest quotidiano (calendario, task e notizie), l'agente ha inviato oltre 500 messaggi a sé stesso, alla moglie e a tutti i contatti Apple recenti della rubrica iMessage. Il bug aveva due componenti: l'integrazione iMessage trattava la lista dei contatti recenti come lista di destinatari senza verificare l'utente autorizzato e il loop di conferma non prevedeva condizione di uscita né timeout. Boyd ha dovuto staccare fisicamente l'alimentazione per interrompere l'agente, per poi correggere il codice e prevenire il ripetersi del problema.
Moltbook breach	Data breach	Wiz [15]	Database Supabase di Moltbook, social network per agenti AI, privo di Row Level Security. Una chiave API esposta nel JavaScript client-side garantiva accesso completo in lettura e scrittura al database di produzione, senza autenticazione. Esposti 35.000 indirizzi e-mail e 1,5 milioni di token API associati agli agenti registrati sulla piattaforma, incluse le credenziali di agenti di ricercatori AI di alto profilo (tra cui Andrej Karpathy). Il numero di agenti registrati al momento della scoperta era di circa 1,5 milioni, riconducibili a circa 17.000 proprietari umani.
Accesso completo a istanze esposte	Compromis-sione esterna	LinkedIn [16]	Jamieson O'Reilly (Dvuln) ha identificato tramite Shodan centinaia di istanze OpenClaw esposte su Internet, prive di autenticazione. Il gateway OpenClaw, configurato per fidarsi di localhost per default, risultava accessibile dall'esterno quando posto dietro reverse proxy mal configurati. O'Reilly ha dimostrato accesso a chiavi API Anthropic, token Telegram, credenziali Slack OAuth, signing keys e cronologia chat completa, nonché esecuzione di comandi con privilegi di amministratore di sistema, capacità intrinseca dell'architettura OpenClaw.
Agente interno Meta che espone dati	Autonomia incontrollata	The Information [17]	Agente AI interno di Meta (descritto come simile a OpenClaw in ambiente enterprise sicuro) invocato da un ingegnere per analizzare una domanda tecni-ca su un forum aziendale, genera e pubblica autono-mamente una risposta contenente indicazioni errate, senza approvazione esplicita dell'operatore. Un altro ingegnere esegue il consiglio dell'agente, modificando permessi di accesso in modo da esporre dati aziendali e utente sensibili a dipendenti non autorizzati per circa due ore.

4. Strumenti open-source per la sicurezza agentica

I seguenti strumenti open-source sono stati sviluppati specificamente in risposta all'ecosistema OpenClaw nel periodo febbraio - marzo 2026. La tabella non è esaustiva, ma rappresenta le soluzioni con maggiore adozione documentata.

Tool	Sviluppatore	Funzionalità
Cisco Skill Scanner	Cisco AI Defense	Scanner open-source multi-engine per skill di agenti AI (OpenClaw, Claude Skills, OpenAI Codex). Combina quattro livelli di analisi: statica, comportamentale, semantica assistita da LLM e scansione VirusTotal. Rileva minacce in descrizioni, metadata e codice implementativo delle skill. [28]
SecureClaw	adversa-ai	Piattaforma di sicurezza open-source con architettura a doppio livello. Il plugin (livello codice) esegue 56 audit check, 5 moduli di hardening e 3 monitor in background, operando su gateway, permessi, credenziali e configurazione. Non può essere aggirato tramite prompt injection. La skill (livello LLM) inietta 15 regole comportamentali nel contesto dell'agente (~1.230 token) per il rilevamento di injection, scansione PII, monitoraggio di integrità e risposta emergenza. [19]
ClawSec	Prompt Security	Suite completa di security skill per piattaforme di agenti AI, con supporto per OpenClaw (MoltBot, Clawdbot e cloni) e NanoClaw (bot WhatsApp containerizzato). Offre sei funzionalità core: ▪ installer con un solo comando dell'intera suite con verifica dell'integrità; ▪ protezione dei file cognitivi dell'agente (SOUL.md, IDENTITY.md, etc.) tramite drift detection e auto-restore; ▪ advisory di sicurezza con polling automatico NVD/CVE e threat intelligence da community, pubblicati su clawsec.prompt.security; ▪ audit automatizzati con script di self-check per marker di prompt injection e vulnerabilità; ▪ verifica checksum SHA-256 su tutti gli artefatti delle skill; ▪ health check con aggiornamenti automatici e verifica di integrità delle skill installate. [30]

ClawNet	Silverfort	Plugin open-source per OpenClaw che intercetta le richieste di installazione delle skill da ClawHub, recupera il contenuto della skill e lo sottopone a review tramite l'endpoint chat-completions del gateway OpenClaw. L'LLM analizza la skill cercando istruzioni sospette o fuorvianti, comportamenti nascosti o insicuri, furto di credenziali o esposizione di segreti, pattern di esecuzione remota rischiosi e altri indicatori di pericolosità. La review restituisce un verdetto (suspicious, severity, reason). Se suspicious è true l'installazione viene bloccata, altrimenti procede normalmente. Implementato come plugin (non come skill) per garantire l'esecuzione del controllo in runtime, indipendentemente dal comportamento del modello. [29]
NemoClaw	NVIDIA	Stack open-source di riferimento che installa il runtime NVIDIA OpenShell (parte dell'NVIDIA Agent Toolkit) per eseguire agenti OpenClaw in un ambiente sandboxed. Funzionalità: controllo egress dichiarativo, isolamento filesystem e processo a livello kernel (Landlock/seccomp), privacy Router per l'instradamento dell'inferenza verso endpoint verificati. [21]

I tool si differenziano per livello di intervento:

- Cisco Skill Scanner e SecureClaw operano sulla supply chain (preinstallazione);
- ClawSec combina verifica di integrità della supply chain e monitoraggio in runtime continuo;
- ClawNet intercetta l'installazione delle skill direttamente nel loop dell'agente tramite plugin;
- NemoClaw isola l'intero runtime in una sandbox con policy enforcement a livello kernel.

4.1 NemoClaw: analisi critica (stato al marzo 2026)

NemoClaw rappresenta la risposta strutturalmente più coerente ai rischi architetturali di OpenClaw, ma il suo stato di maturità attuale richiede cautela prima di adottarlo come standard operativo. Il repository ha raggiunto 15.900 stelle e 1.700 fork in meno di dieci giorni dal rilascio — un tasso di adozione che replica il pattern di OpenClaw stesso e conferma l'urgenza percepita dal mercato per una soluzione di sandboxing, ma segnala anche il rischio che la pressione verso l'adozione preceda la stabilizzazione del prodotto.

Dall'analisi del repository pubblico (github.com/NVIDIA/NemoClaw) emergono quattro elementi rilevanti:

- **Stato alpha genuino.** Rilasciato il 16 marzo 2026, il repository presenta al momento dell'analisi 180 issue aperte e 194 pull request aperte. Un numero di pull request superiore alle issue è indicatore di una base di codice in rapida evoluzione con integrazione non ancora stabilizzata. Nessuna release taggata è stata pubblicata: tutto il codice risiede sul branch principale, rendendo impossibile il blocco a una versione stabile tramite i meccanismi standard di GitHub.
- **Installer non verificabile.** Il metodo di installazione raccomandato è `curl -fsSL https://www.nvidia.com/nemoclaw.sh | bash` — esecuzione diretta di codice remoto senza checksum, firma crittografica o procedura di verifica alternativa documentata. Per uno strumento che si posiziona esplicitamente come soluzione di sicurezza per agenti AI, questa contraddizione è rilevante e non viene discussa nella documentazione ufficiale.
- **Dipendenza infrastrutturale.** Il provider di inferenza predefinito è `nvidia/nemotron-3-super-120b-a12b` tramite NVIDIA Endpoint API (chiave da build.nvidia.com obbligatoria); le alternative locali (Ollama, vLLM) sono dichiaratamente sperimentali. OpenShell è anch'esso prodotto NVIDIA. La soluzione risolve i rischi di OpenClaw adottando l'intera infrastruttura NVIDIA, con implicazioni di dipendenza che le organizzazioni devono valutare esplicitamente.
- **Copertura macOS limitata.** Su macOS — piattaforma su cui opera una parte significativa della base utenti di OpenClaw — i runtime supportati si limitano a Colima e Docker Desktop; Podman non è supportato e l'inferenza locale (Ollama, vLLM) è dichiaratamente sperimentale, dipendente da un supporto host-routing di OpenShell non ancora stabilizzato. La soluzione di sandboxing strutturalmente più solida disponibile è quindi parzialmente inaccessibile alla fascia di utenti potenzialmente più esposta.

Raccomandazione. NemoClaw è la direzione corretta e va monitorato attivamente. Non è adatto a deployment in produzione allo stato attuale. La sperimentazione è appropriata esclusivamente in ambienti sandbox con dati sintetici, in linea con quanto già indicato da Gartner [22] per OpenClaw stesso.

5. Mapping Kill Chain: MITRE ATT&CK + ATLAS

La tabella seguente mappa le principali tecniche osservate finora nel report, integrando ATT&CK v18 con le tecniche ATLAS specifiche per sistemi AI. La colonna TID riporta gli identificatori tecnici verificabili direttamente su attack.mitre.org e atlas.mitre.org.

Tattica	TID	Tecnica
Initial Access	T1190	Exploit Public-Facing Application
Initial Access	T1195.001	Supply Chain: Compromise Software Dependencies
Execution	AML.T0051*	LLM Prompt Injection
Execution	T1059.007	Command and Scripting Interpreter: JavaScript
Persistence	T1543.001 T1543.002 T1053.005	Create/Modify System Process: Launch Agent (macOS) Systemd Service (Linux) Scheduled Task/Job: Scheduled Task (Windows)
Credential Access	T1528	Steal Application Access Token
Collection	T1114.002	Email Collection: Remote Email Collection
Exfiltration	T1567	Exfiltration Over Web Service
C2	T1071.001	Application Layer Protocol: Web Protocols

TID = Technique ID - identificatori verificabili su attack.mitre.org e atlas.mitre.org [27]

**Nota: Tattica ufficiale ATLAS: Initial Access. Mappata a Execution per applicazione funzionale nel contesto OpenClaw.*

6. Conclusioni

Le conclusioni che seguono non riguardano OpenClaw come prodotto ma riguardano la classe di sistemi di cui OpenClaw è il primo caso documentato su scala globale: gli agenti AI autonomi con accesso privilegiato, memoria persistente, elaborazione di contenuto non verificato e integrazione OAuth con servizi enterprise. Il caso OpenClaw non è riducibile a una crisi di sicurezza generata da un singolo prodotto affrettato nel rilascio. Rappresenta il primo caso documentato di un problema strutturale che accompagnerà l'intera fase di adozione dell'AI agentica: la combinazione di capacità operative ampie con un modello di sicurezza non maturo.

L'analisi condotta evidenzia cinque conclusioni principali.

1. Il modello di rischio degli agenti AI autonomi è qualitativamente diverso da quello delle applicazioni tradizionali. Le tecniche di prompt injection risultano difficilmente mitigabili con i modelli di sicurezza tradizionali, in quanto sfruttano la natura stessa dell'interazione linguistica e dell'interpretazione contestuale. I time-shifted attack sulla memoria persistente e la supply chain dei marketplace introducono vettori per i quali gli strumenti di difesa esistenti non sono progettati.

2. L'attribuzione delle azioni agentiche è un problema aperto con implicazioni normative dirette. Quando un agente AI opera con le credenziali dell'utente — token OAuth, sessioni browser, accesso a caselle e-mail — le azioni eseguite risultano indistinguibili da quelle compiute dall'utente stesso nei log di audit. Come osservato da Gartner [22], questa condizione introduce potenziali criticità in ambito responsabilità operativa (a chi è imputabile un'azione dell'agente?), attribuzione delle azioni (come distinguere attività umana da attività agentica nei log?) e verificabilità delle decisioni (come ricostruire il percorso decisionale dell'agente?). Queste questioni meritano approfondimento specifico in relazione ai framework NIS2 (art. 21), GDPR (artt.5(1)(f),25,32), DORA (art. 9) e EU AI Act (Allegato III e art. 50) — per i quali si rimanda alla mappatura normativa in Appendice A.

3. La distinzione tra agenti AI personali e agenti AI enterprise è strutturale, non solo terminologica. Un assistente personale come OpenClaw opera in un paradigma "agency first, security second": privilegi amministrativi, connettori non verificati, skill da marketplace non moderati, esecuzione su infrastruttura non controllata dall'organizzazione.

L'alternativa enterprise richiede un'inversione di principio — compliance e sicurezza come prerequisiti, agency come risultato — con controlli centralizzati, identità agentiche dedicate con least privilege, connettori approvati e tracciabilità nativa.

4. La velocità di adozione supera sistematicamente la maturità dei framework di sicurezza. In meno di due mesi, le stime di esposizione di OpenClaw hanno raggiunto tra 21.000 e oltre 135.000 istanze a seconda della metodologia di scansione, con il 22–50% delle organizzazioni enterprise che presentavano deployment non approvati [24]. Al momento della redazione del presente report non esistono standard tecnici consolidati né framework di governance specificamente progettati per questa categoria di sistemi. Finché le organizzazioni non rendono disponibili alternative enterprise con controlli vincolanti e verificabili — non semplici istruzioni in linguaggio naturale — la pressione verso l'adozione di soluzioni personali non autorizzate continuerà ad aumentare.

5. La risposta deve essere multidimensionale e non può ridursi alla gestione dei CVE. Le patch ai singoli CVE — pur necessarie — non risolvono le vulnerabilità architetturali. Le raccomandazioni operative sono le seguenti:

- **Detection:** implementare IoC di rete, endpoint e comportamentali specifici per agenti AI nella pipeline SOC, con priorità agli interventi classificati come critici nella matrice in appendice A;
- **Governance:** definire un processo formale di approvazione per qualsiasi agente AI con accesso a filesystem, e-mail o sistemi aziendali; implementare SSPM per il rilevamento di integrazioni OAuth non autorizzate; strutturare un programma di cybersecurity agentica articolato sulle cinque aree di intervento identificate da Gartner per i deployment agentici [22];
- **Risposta agli incidenti:** trattare qualsiasi istanza agentica precedentemente attiva su endpoint aziendali come potenziale incidente; revocare esplicitamente tutti i token OAuth collegati — nota: la disinstallazione standard non è sufficiente, si veda la nota operativa in Appendice A;
- **Architettura sicura per deployment controllati:** binding su 127.0.0.1, autenticazione obbligatoria, sandboxing attivo, Human-in-the-Loop per azioni ad alto rischio, aggiornamento continuativo tramite openclaw secrets audit.

Il fenomeno OpenClaw è un indicatore precoce di una transizione più ampia. L'industria AI sta costruendo agenti con capacità operative crescenti — accesso a sistemi enterprise, esecuzione autonoma di workflow, integrazione con infrastrutture critiche. La finestra temporale in cui le organizzazioni possono dotarsi di strumenti, processi e competenze adeguati prima che questa categoria raggiunga una diffusione enterprise di massa è limitata.

Appendice A – Strumenti Operativi per i Team di Sicurezza

La presente appendice raccoglie gli strumenti operativi derivati dall'analisi condotta nello studio. Tali strumenti non sostituiscono un'analisi specifica per la singola organizzazione, ma forniscono una base strutturata per la prioritizzazione degli interventi e la verifica della conformità normativa. Le valutazioni di conformità riportate nella mappatura normativa sono indicative e non sostituiscono un'analisi legale specifica.

Matrice di priorità degli interventi

La matrice seguente traduce i rischi identificati nell'analisi in interventi operativi concreti, ordinati per urgenza. La priorità è determinata dalla combinazione di due fattori: impatto potenziale in caso di mancata azione e complessità di implementazione. Gli interventi classificati come "Critica" richiedono esecuzione immediata indipendentemente dalle altre attività in corso; quelli "Alta" vanno pianificati nel breve termine; quelli "Media" rientrano in un programma strutturato di hardening.

Priorità	Intervento	Impatto	Complessità
Critica	Eliminare credenziali in chiaro; ruotare tutti i token OAuth se OpenClaw ha avuto accesso al sistema	Alto	Bassa
Critica	Abilitare approvazioni human-in-the-loop per azioni distruttive e transazioni — canale di conferma out-of-band	Alto	Bassa
Alta	Migrare a runtime NemoClaw o equivalente per deployment controllati (solo ambienti sandbox allo stato attuale)	Alto	Media
Alta	Implementare monitoraggio IoC di rete, endpoint e comportamentali nel SIEM e EDR esistenti	Medio	Media
Media	Audit completo e revoca delle skill installate; whitelist delle skill approvate	Medio	Bassa
Media	Integrare guardrail semantici attivi per ispezione dell'input/output LLM	Alto	Alta

Nota: la disinstallazione standard di OpenClaw non revoca le credenziali sui server terzi. La rotazione esplicita di tutti i token OAuth collegati è prerequisito indipendente dagli altri interventi.

Mappatura delle non conformità normative

La mappatura seguente identifica le non conformità normative generate dall'utilizzo di OpenClaw nelle versioni precedenti alle patch correttive, in deployment privi delle misure di hardening descritte in questo studio. Non si tratta di una valutazione dello stato attuale del progetto, ma di un'analisi delle implicazioni regolatorie per le organizzazioni che abbiano avuto OpenClaw attivo sui propri endpoint nel periodo novembre 2025–marzo 2026. I quattro framework considerati — NIS2, GDPR, EU AI Act e DORA — sono quelli con applicabilità diretta al contesto enterprise europeo.

Norma	Articolo	Violazione specifica
NIS2	Art. 21(2), lett. (b), (e)	Inadeguatezza delle procedure di incident handling: impossibilità documentata di arrestare l'agente tramite comandi diretti, fallimento del meccanismo di stop anche in presenza di istruzioni esplicite dell'utente (b). Gestione delle vulnerabilità esclusivamente reattiva: 53+ CVE con CVSS medio 7.3, nessun SDLC documentato, zero risorse dedicate al vulnerability management (e). (EUR-Lex NIS2: https://eur-lex.europa.eu/legal-content/IT/TXT/?uri=CELEX%3A32022L2555)
NIS2	Art. 21(2), lett. (h), (j)	Credenziali e token OAuth in chiaro in ~/.openclaw/ (h). Gateway esposto senza autenticazione nei deployment Docker, assenza di validazione header Origin nelle connessioni WebSocket (j).
GDPR	Artt. 5(1)(f), 25, 32	Memorizzazione di credenziali, token OAuth e chiavi API in chiaro in ~/.openclaw/credentials/*.json, sandboxing disattivo per default, gateway esposto su tutte le interfacce di rete: violazione dell'obbligo di protezione tecnica adeguata dei dati trattati (32). Assenza di separazione tra contesto fidato e contesto non fidato, sessione DM globale con segreti accessibili ad altri utenti dello stesso bot: violazione dell'obbligo di data protection by design (25). Le due condizioni integrate violano il principio di integrità e riservatezza (5(1)(f)) (EUR-Lex GDPR: https://eur-lex.europa.eu/legal-content/IT/TXT/?uri=CELEX%3A32016R0679)
EU AI Act	Art. 6(2) + Allegato III	Se il deployment rientra in un caso d'uso elencato nell'Allegato III (es. cat. 2 — infrastrutture critiche digitali), si attivano obblighi di human oversight (Art. 14) e cybersecurity (Art. 15) documentati come assenti nell'architettura OpenClaw. (EUR-Lex EU AI Act: https://eur-lex.europa.eu/legal-content/IT/TXT/?uri=CELEX%3A32024R1689)
EU AI Act	Art. 50	Il sistema non è progettato per informare le persone fisiche che interagiscono con un sistema AI (Art. 50, comma 1). L'agente risponde autonomamente su forum aziendali e canali di messaggistica senza che i destinatari siano consapevoli della natura agentica dell'interlocutore.
DORA	Art. 9	Assenza di meccanismi di autenticazione sul gateway e di validazione delle connessioni WebSocket (comma 4, lett. d). Azioni dell'agente non distinguibili da quelle dell'utente nei log di audit — impatto sull'autenticità dei dati; credenziali in chiaro e sessione DM globale con segreti accessibili ad altri utenti — impatto sulla riservatezza dei dati (comma 2). (EUR-Lex DORA: https://eur-lex.europa.eu/legal-content/IT/TXT/?uri=CELEX%3A32022R2554)

Le valutazioni di conformità riportate sono indicative e non sostituiscono un'analisi legale specifica per la singola organizzazione.

Bibliografia e Fonti

[1] Willison, S. (2025, giugno 16). The lethal trifecta for AI agents: private data, untrusted content, and external communication. simonwillison.net.
https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/

[2] Palo Alto Networks — Mishra, S.; Morgan, S.P. (2026, febbraio 4). OpenClaw May Signal the Next AI Security Crisis.
https://www.paloaltonetworks.com/blog/network-security/why-moltbot-may-signal-ai-crisis/

[3] Trend Micro — Tucci, F. (2026, marzo 10). CISOs in a Pinch: A Security Analysis of OpenClaw.
https://www.trendmicro.com/en_us/research/26/c/cisos-in-a-pinch-a-security-analysis-openclaw.html

[4] Cisco. (2026). State of AI Security 2026.
https://www.cisco.com/site/us/en/products/security/state-of-ai-security.html

[5] Immersive Labs — Breen, K. (2026, febbraio 19). Why You Should Uninstall OpenClaw AI Immediately.
https://www.immersivelabs.com/resources/c7-blog/openclaw-what-you-need-to-know-before-it-claws-its-way-into-your-organization

[6] Oasis Security. (2026, febbraio). ClawJacked: OpenClaw Vulnerability Enables Full Agent Takeover. https://www.oasis.security/blog/openclaw-vulnerability

[7] Giskard. (2026, febbraio). OpenClaw security issues include data leakage and prompt injection.
https://www.giskard.ai/knowledge/openclaw-security-vulnerabilities-include-data-leakage-and-prompt-injection-risks

[8] Koi Security — Yomtov, O. (2026, febbraio). ClawHavoc: 341 Malicious Skills Found.
https://www.koi.ai/blog/clawhavoc-341-malicious-clawedbot-skills-found-by-the-bot-they-were-targeting

[9] Eye Security. (2026, febbraio). Log Poisoning — OpenClaw AI Agent Injection Risk.
https://www.eye.security/blog/log-poisoning-openclaw-ai-agent-injection-risk

[10] 1Password — Meller, J. (2026, febbraio 1). From Magic to Malware: How OpenClaw's Agent Skills Become an Attack Surface.
https://1password.com/blog/from-magic-to-malware-how-openclaws-agent-skills-become-an-attack-surface

[11] PromptArmor. (2026, febbraio). LLM Data Exfiltration via URL Previews — OpenClaw example.
https://www.promptarmor.com/resources/llm-data-exfiltration-via-url-previews-(with-openclaw-example-and-test)

[12] OpenAI. (2026, marzo). Designing Agents to Resist Prompt Injection. https://openai.com/index/designing-agents-to-resist-prompt-injection/

[13] Rehberger, J. (2026, marzo 16). Agent Commander: Promptware-Powered Command and Control. Embrace The Red. https://embracethered.com/blog/posts/2026/agent-commander-your-agent-works-for-me-now/

[14] Bloomberg. (2026, febbraio 4). AI Agent Goes Rogue, Spamming OpenClaw User With 500 Messages. https://www.bloomberg.com/news/articles/2026-02-04/openclaw-s-an-ai-sensation-but-its-security-a-work-in-progress

[15] Wiz. (2026, febbraio). Exposed Moltbook Database Reveals Millions of API Keys. https://www.wiz.io/blog/exposed-moltbook-database-reveals-millions-of-api-keys

[16] O'Reilly, J. (2026). Hacking Clawdbot: Eating Lobster Souls. LinkedIn. https://www.linkedin.com/pulse/hacking-clawdbot-eating-lobster-souls-jamieson-o-reilly-whhlc/

[17] The Information. (2026, marzo 18). Inside Meta, a Rogue AI Agent Triggers Security Alert. https://www.theinformation.com/articles/inside-meta-rogue-ai-agent-triggers-security-alert

[18] Gamblin, J. (2026). OpenClaw CVE & Security Advisory Tracker — monitoraggio automatico CVE e GHSA, aggiornato ogni ora. GitHub. https://github.com/jgamblin/OpenClawCVEs

[19] Adversa AI — Polyakov, A. (2026, febbraio 18). SecureClaw: Dual Stack Open-Source Security Plugin. Help Net Security. https://www.helpnetsecurity.com/2026/02/18/secureclaw-open-source-security-plugin-skill-openclaw/
Repository: https://github.com/adversa-ai/secureclaw

[20] ethiack/moltbot-1click-rce. (2026). PoC CVE-2026-25253. https://github.com/ethiack/moltbot-1click-rce

[21] NVIDIA. (2026, marzo). NemoClaw — Reference Stack for Running OpenClaw in OpenShell. GitHub. https://github.com/NVIDIA/NemoClaw

[22] Gartner — Watts, J.; Khandabattu, H. (2026, febbraio 9). Block Personal AI Assistants Before They Control Your Organization. G00847608.

[22] Gartner — Khandabattu, H.; Brethenoux, E.; D'Hoinne, J.; Watts, J.; Dekate, C.; Olliffe, G. (2026, gennaio 30). First Take: OpenClaw (Formerly Moltbot, Clawdbot): Agentic Productivity Comes With Unacceptable Cybersecurity Risk. G00847299.

[23] Gartner. (2025). Cybersecurity Innovations in AI Risk Management and Use Survey. Gartner, Inc. (accesso riservato ad abbonati).

[24] Trend Micro — Gariuolo, S.; Ciancaglini, V.; Tucci, F. (2026, febbraio 6). Viral AI, Invisible Risks: What OpenClaw Reveals About Agentic Assistants. https://www.trendmicro.com/en_us/research/26/b/what-openclaw-reveals-about-agentic-assistants.html

[25] CrowdStrike. (2026, febbraio). What Security Teams Need to Know About OpenClaw. https://www.crowdstrike.com/en-us/blog/what-security-teams-need-to-know-about-openclaw-ai-super-agent/

[26] Yue, S. (2026, gennaio). Post su X (@summeryue0). https://x.com/summeryue0/status/2025774069124399363

[27] MITRE ATLAS — Adversarial Threat Landscape for AI Systems. https://atlas.mitre.org/

[28] Cisco AI Defense — Chang, A.; Narajala, V.S.; Habler, I. (2026, gennaio 28). Personal AI Agents like OpenClaw Are a Security Nightmare. https://blogs.cisco.com/ai/personal-ai-agents-like-openclaw-are-a-security-nightmare

[29] Silverfort — Gazit, N. (2026, marzo 24). Hijacking trust: ClawHub vulnerability enables attackers to manipulate rankings to become the #1 skill. https://www.silverfort.com/blog/clawhub-vulnerability-enables-attackers-to-manipulate-rankings-to-become-the-number-one-skill/

[30] Prompt Security. (2026). ClawSec — Security Suite for AI Agent Platforms. GitHub. https://github.com/prompt-security/clawsec

tinexta
defence

Next | Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 – 00131 Roma
tel. 06.45752720 – info@defencetech.it
www.tinextadefence.it

#TinextaDefenceBusiness