TECHNICAL REPORT — TR-2026-01

Operation "OpenClaw"

Anatomy of an AI-Driven Cyberattack Against a Pharmaceutical Company

From OSINT Reconnaissance to Double Extortion: Modeling a Five-Phase Agentic Kill Chain

Author: Fabrice Pizzi

Affiliation: Université Paris Sorbonne

Date: February 2026

Version: 8.0

⚠ WARNING

This document presents the full analysis of Phase 1 (Reconnaissance) of Operation OpenClaw. It is a FICTIONAL but technically grounded scenario, based exclusively on publicly documented vulnerabilities, tools, and techniques.

NO actual attack was conducted. MediFrance SA does not exist.

Objective: identify and understand the emerging risks related to AI agent security to improve defensive postures.

General Introduction

Operation "OpenClaw" is a fictional cyberattack scenario designed to illustrate the convergence between offensive artificial intelligence and advanced intrusion techniques. It describes the complete attack against MediFrance SA, a mid-size pharmaceutical company (~500 employees), orchestrated by exploiting the OpenClaw autonomous AI coding agent as both attack vector and force multiplier.

This document consolidates the five phases of the operation, from initial reconnaissance (D−30) through final actions on objectives (D+6), spanning six weeks of offensive activity (it should be expected that, in a real case, compression into a few days would be achievable via parallelization of agents).

Marcus Sachs (Center for Internet Security) predicts that by 2026, fully automated lateral movement engines will require little to no human intervention [3]. John Grady (Omdia) notes that "the 2025 wave of AI-driven attacks will target the weakest link in the chain — human trust" [24]. These convergent observations motivate the construction of the present scenario.

OpenClaw Kill Chain Summary

The table below summarizes the five phases of the operation, each detailed in subsequent sections:

Phase	Kill Chain	Key AI Technique	Duration	Impact
1	Reconnaissance	WormGPT OSINT + Shodan OpenClaw fingerprint	D−30 to D−15	Target + vulnerabilities
2	Weaponization	Malicious skill + PromptLock ransomware	D−15 to D−7	Complete arsenal
3	Delivery	3 simultaneous vectors (skill + infostealer + VPN)	D−7 to D	Initial access × 3
4	Lateral movement	Autonomous LotL agent + AD compromise	D+1 to D+5	Full control
5	Impact	PromptLock + double extortion	D+6	€2M ransom + IP threat

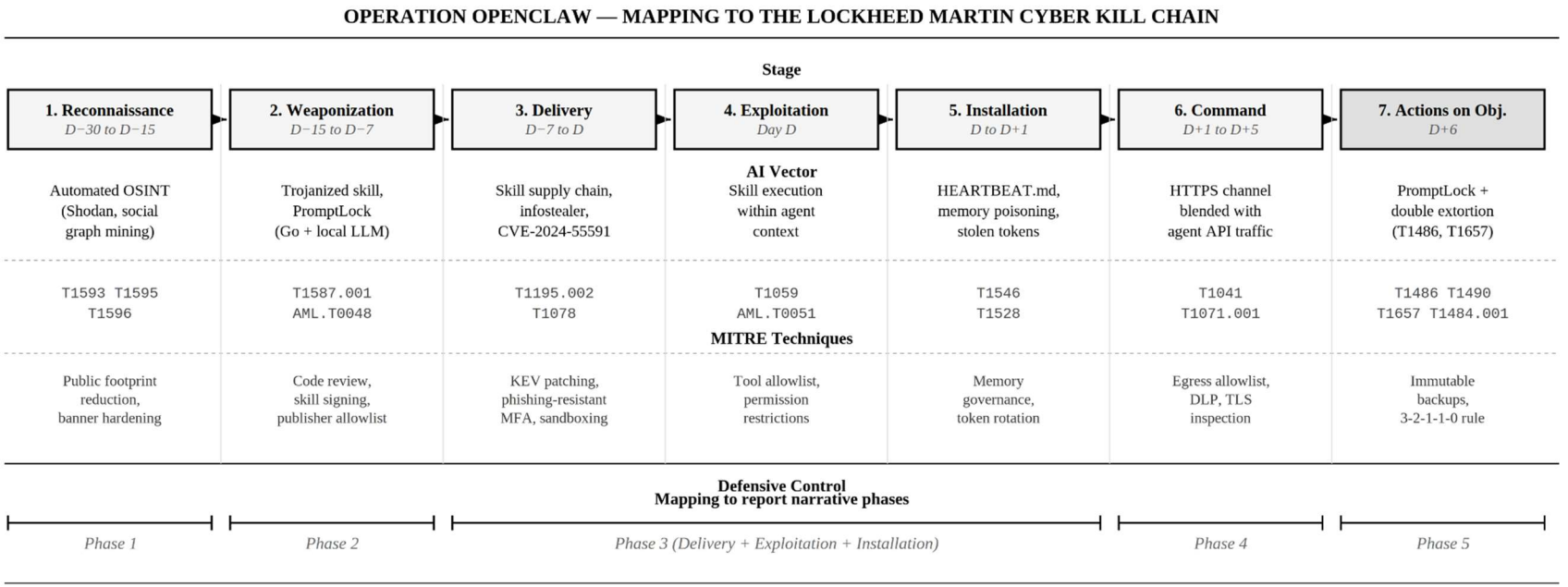


Figure 1. Operation OpenClaw mapped to the seven stages of the Lockheed Martin Cyber Kill Chain [1].
Each stage shows the AI vector leveraged, the corresponding MITRE ATT&CK/ATLAS techniques, and the priority defensive control.
Bottom brackets indicate the mapping to the five narrative phases of the report. Phase 3 spans three Kill Chain stages
(Delivery, Exploitation, Installation), reflecting the temporal overlap of these actions in the scenario.
Stage 7 (darker background) represents the final objective of the operation.

> The Central Role of OpenClaw in the Operation
> The OpenClaw agent is not simply another vulnerable application: it is a force multiplier that fundamentally transforms the attacker-target relationship. Its unique capabilities — autonomous code execution, external communication via connectors, memory of past interactions — make it both the entry point and the offensive execution engine.
>
> The reconnaissance phase detailed below specifically aims to identify organizations where OpenClaw is deployed in conditions that create the lethal trifecta of risks identified by S. Willison: access to private data + exposure to untrusted content + external communication capability.

Document Structure

This document is organized into five chapters corresponding to the five phases of Operation OpenClaw, each including: the operational context, detailed technical analysis of the techniques and procedures employed, MITRE ATT&CK mapping, and transition conditions to the next phase.

- **Phase 1** — Reconnaissance (D−30 to D−15): organizational mapping via Social Graph Mining, computational stylometry, passive fingerprinting, and vulnerability inference;
- **Phase 2** — Weaponization (D−15 to D−7): creation of the malicious skill PharmaResearch Assistant, assembly of PromptLock (Go/Ollama), and complete arsenal preparation;
- **Phase 3** — Delivery and Execution (D−7 to D): simultaneous multi-vector attack: infostealer via supplier supply chain, Fortinet VPN exploitation CVE-2024-55591, malicious skill installation via ClawHub;
- **Phase 4** — Lateral Movement and Persistence (D+1 to D+5): autonomous LotL AI agent, AD compromise via Mimikatz (DCSync, Golden Ticket), OpenClaw hijacking via Slack prompt injection, internal chatbot poisoning (PoisonGPT);
- **Phase 5** — Actions on Objectives (D+6): R&D exfiltration summary, PromptLock deployment (200 workstations + 15 servers in 40 min), double extortion (€2M ransom + IP publication threat).

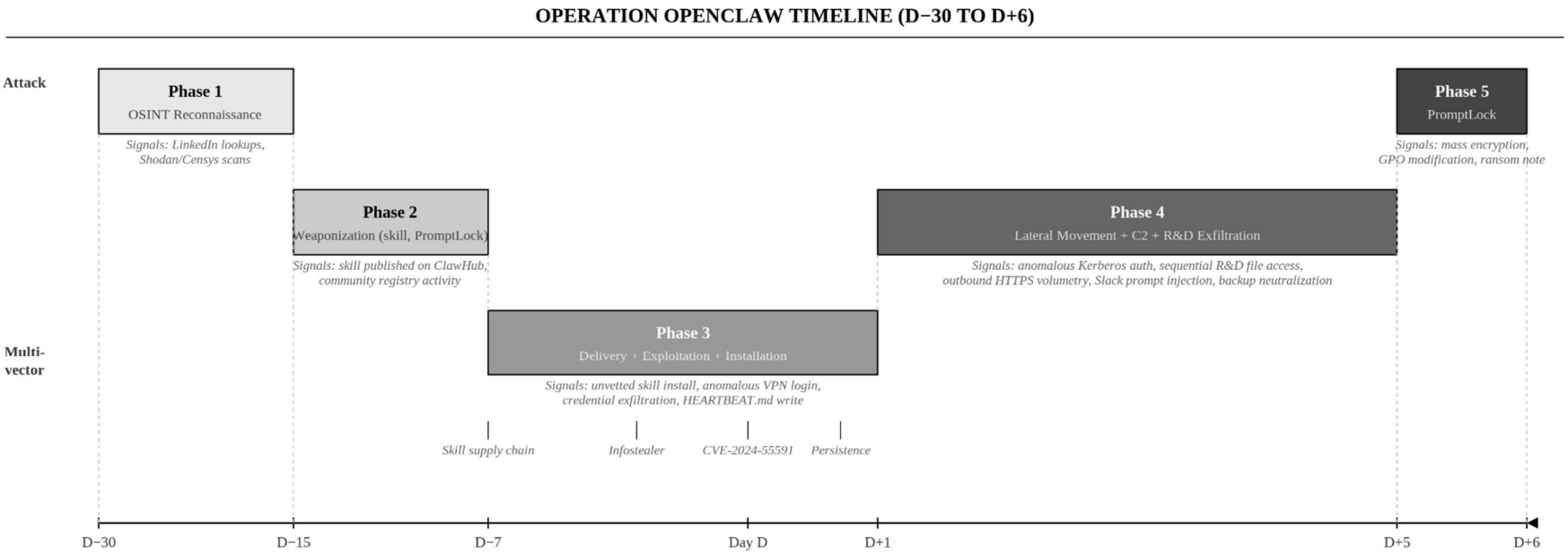
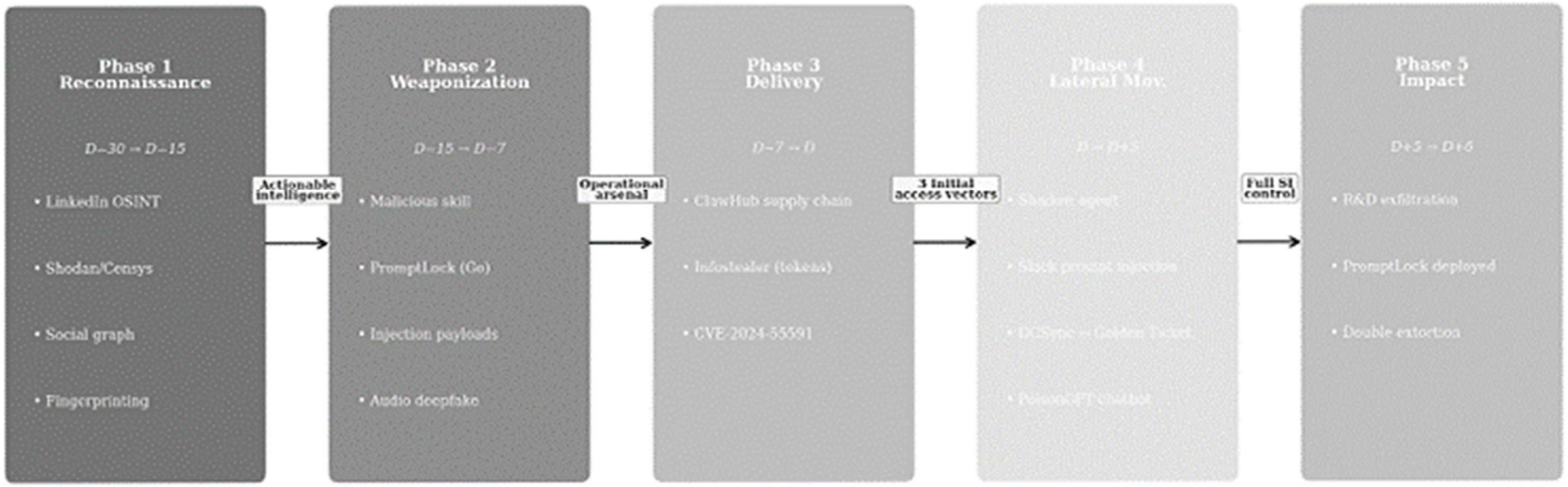


Figure 2. Operation OpenClaw timeline. Bar intensity reflects the attacker's escalating privilege level across phases. Detection signals identified beneath each phase represent intervention windows for defensive teams. Phase 3 spans three Kill Chain stages (Delivery, Exploitation, Installation) with parallel attack vectors. Timeline is illustrative (cf. §2.3).

Figure 3 — Agentic Kill Chain Flow — Operation OpenClaw



Abstract

This document presents the comprehensive academic analysis of the reconnaissance phase (D-30 to D-15) of Operation "OpenClaw," a fictional but realistic cyberattack scenario targeting a pharmaceutical mid-size company (MediFrance SA). The analysis is structured around three complementary axes: (1) organizational mapping via social graph mining and LLM-augmented profiling, (2) computational stylometry (writeprints) for behavioral attribution and impersonation detection, and (3) passive infrastructure enumeration and fingerprinting of exposed technical assets.

Keywords: automated reconnaissance, OSINT, Social Graph Mining, computational stylometry, passive fingerprinting, vulnerability inference, WormGPT, OpenClaw, Shodan, CVE, I2D, shadow AI

1. Introduction and Strategic Operational Framework

The contemporary evolution of Cyber Kill Chains demonstrates a significant compression of exploitation timelines, coupled with increased structural sophistication of deployed offensive tools. This paradigm shift is particularly evident in the reconnaissance phase, where the systematic integration of artificial intelligence techniques fundamentally transforms the information collection and correlation capabilities available to an attacker.

The emergence of unaligned Large Language Models (LLMs), such as WormGPT or FraudGPT, has fundamentally altered this operational dynamic by enabling systematic automation of the reconnaissance-exploitation chain. These models, freed from the ethical constraints imposed on commercial LLMs (ChatGPT, Claude), can generate contextualized phishing campaigns, produce polymorphic code, and conduct advanced OSINT analyses [3].

Concurrently, the meteoric adoption of the OpenClaw autonomous AI agent (formerly ClawdBot, then MoltBot) — 180,000+ GitHub stars, 720,000 weekly downloads, over 40,000 instances publicly exposed on the Internet — creates an unprecedented attack surface that combines traditional vulnerability exploitation with novel AI-specific attack vectors [7, 8, 9].

This document exhaustively details the algorithmic mechanisms and correlation heuristics underlying the reconnaissance phase of Operation OpenClaw, a fictional but technically grounded attack scenario targeting MediFrance SA, a pharmaceutical mid-size company (ETI) with approximately 500 employees, standard Microsoft infrastructure, and OpenClaw deployment in research and development.

2. Axis 1: Organizational Mapping Through Social Graph Mining

2.1 Target Discovery via Shodan and OpenClaw Fingerprint

The first step of reconnaissance exploits OpenClaw's insecure default configuration. Initial versions of the project bound the gateway on 0.0.0.0:18789, listening on all network interfaces without authentication. This

configuration, now deprecated but still present on numerous deployments, directly exposed the administration interface to the public Internet [9, 12].

The attacker uses Shodan to scan the Internet for HTTP signatures characteristic of the OpenClaw control panel. This technique was publicly demonstrated by Jamieson O'Reilly (Dvuln), who identified more than 3,000 accessible instances in February 2026 [11]. SecurityScorecard subsequently confirmed exposure exceeding 40,000 instances worldwide [7].

The attacker identifies an OpenClaw instance at MediFrance SA, made accessible externally via a misconfigured Nginx reverse proxy. Through access to the exposed gateway, they observe internal DNS queries, active tool integrations (Outlook, Slack, terminal), and ClawHub-installed skills — including the specific agent version and its active connectors.

> Key Vulnerability: CVE-2026-25253 (CVSS 8.8)
> Remote code execution through crafted skill execution — disclosed by Jamieson O'Reilly (Dvuln) in February 2026 [11].
>
> This critical flaw allows an attacker to achieve code execution on the machine hosting the OpenClaw agent through a specially crafted malicious skill. Unlike other vectors requiring social engineering, this vulnerability can be exploited through the sole distribution of a skill via ClawHub.

2.2 Automation of OSINT Profiling via Malicious LLMs

OSINT (Open Source Intelligence) profiling is undergoing a profound transformation with the emergence of unaligned Large Language Models, of which WormGPT is the most documented archetype. These models, capable of generating contextualized content without ethical restrictions, are documented in the cybersecurity literature as catalysts for social engineering at scale [3, 5, 6].

Technical Architecture of Automated Profiling

The AI agent deploys a multi-layered architecture to systematically extract metadata from LinkedIn profiles. The process revolves around two components whose capabilities are unequally documented in the literature:

1. Automated Collection (Automated Reconnaissance) — established and documented capability

Attackers use automated OSINT techniques via frameworks such as Maltego, Recon-ng, SpiderFoot, and TheHarvester to massively extract LinkedIn data [165]. LinkedIn represents one of the most exploited platforms for professional reconnaissance, with 51% of social engineering attacks targeting this network (StrongestLayer, 2026) [24].

2. Processing by Unaligned LLM — partially documented capabilities

WormGPT intervenes to process this raw data and generate contextual inferences. Unlike standard LLMs (ChatGPT, Claude) that would refuse to participate in offensive operations, WormGPT and its variants

(WormGPT v4, KawaiiGPT, DarkBERT) are specifically trained or fine-tuned to remove ethical guardrails [3, 5, 6]. The claimed capabilities of these models include the ability to:

- Semantically analyze job descriptions to identify technical responsibilities (VPN access, critical infrastructure management)
- Correlate fragmented information to reconstruct the organizational chart
- Identify priority targets (system administrators, security officers)
- Generate personalized social engineering pretexts

Note: these operations do not require any technical capability specific to WormGPT — any general LLM (including uncensored open-source models like Mistral or LLaMA without guardrails) is capable of them. The distinctive contribution of WormGPT lies in the removal of alignment constraints, not in unique technical capabilities [5, 6].

2.3 Social Graph Mining

Theoretical Foundations of Social Network Alignment

Social graph mining constitutes the first pillar of passive inference, enabling mapping of a target organization's relational ecosystem. Social Network Alignment (SNA) is an active area of research in graph theory, with robust methodological foundations established by multiple teams [14, 15, 16, 18].

Methodological Evolution — Established State of the Art

The traditional approach relied on supervised models requiring immense labeled datasets to train classifiers. These models had several documented limitations: excessive dependence on labeled data, fragility in the face of distribution changes, and difficulty generalizing to heterogeneous networks.

Contemporary models deploy unsupervised and self-supervised learning algorithms, notably [166]:

- **Graph Attention Networks (GAT): neural networks capable of dynamically weighting the importance of neighboring nodes**
- **Contrastive Multi-View Learning: contrastive learning exploiting multiple views (structural, semantic, temporal) to improve entity representation**
- **Graph Convolutional Networks (GCN): convolutions adapted to non-Euclidean graph structures**
- **Unsupervised Node Embeddings: techniques like Node2Vec [162], DeepWalk generating vector representations of nodes without supervision**

These architectures enable social network alignment capable of automatically identifying relational patterns without previously labeled training data [14].

However, scientific publications on SNA do not document their specific use by malicious LLMs like WormGPT for offensive purposes. The convergence between SNA and offensive LLMs is a technically plausible extrapolation from our scenario, not an empirically observed reality. This extrapolation is grounded in the fact that the algorithms are available in open-source and the LLM processing capability is established.

2.4 Mathematical Modeling of the Organizational Graph

Social Graph Formalization — theoretical modeling from established components

The MediFrance organization is mathematically modeled as a directed weighted graph G = (V, E), where:

- $V = \{v_1, v_2, ..., v_n\}$ represents the set of nodes (the n identified employees)
- $E \subseteq V \times V$ represents the set of edges (inferred relationships)
- $w : E \rightarrow \mathbb{R}^+$ is a weighting function assigning a positive weight to each edge

A directed edge $e = (u,v) \in E$ indicates an inferred relationship from employee u to employee v (for example: u mentions v, u comments on v's publications, u and v share mutual connections). The edge direction reflects the initiating actor of the observed interaction.

Edge Weight Calculation

The weight w(u,v) of each edge is calculated according to a composite function integrating three proximity dimensions [14]:

$w(u,v) = \alpha \cdot S\_sem(u,v) + \beta \cdot S\_spa(u,v) + \gamma \cdot S\_temp(u,v)$

where α, β, γ are weighting hyperparameters (with $\alpha + \beta + \gamma = 1$).

1. Semantic Proximity S_sem(u,v) — established metric [14]

Measures the similarity of textual content associated with profiles. Vector embeddings are generated for each employee based on: job descriptions and listed skills, published content and comments, shared group memberships, and mutual connections.

Semantic similarity is calculated via cosine similarity:

$S\_sem(u,v) = (emb\_u \cdot emb\_v) / (\|emb\_u\| \cdot \|emb\_v\|)$

2. Spatial Proximity S_spa(u,v) — established composite modeling

Quantifies the geographical distance between employees and topological distance in the social graph:

$S\_spa(u,v) = \omega_1 \cdot e^{(-d\_geo(u,v)/\sigma)} + \omega_2 \cdot |N(u) \cap N(v)| / |N(u) \cup N(v)|$

where d_geo is the geographical distance, N(u) the set of neighbors of u, σ a scale parameter, and the second term is the Jaccard coefficient on neighborhoods [14].

3. Temporal Proximity S_temp(u,v) — established metric

Analyzes activity synchronization: simultaneous publications, comments within close temporal windows, correlated career transitions:

$S\_temp(u,v) = corr(TS\_u, TS\_v)$

where TS_u and TS_v represent the activity time series of users u and v.

Hierarchical Relationship Inference — established techniques

Analysis of the weighted graph exploits established centrality algorithms [161]:

- **High-weight edges: frequent collaborative relationships, potentially within the same team**
- **Centrality analysis: identification of hubs (managers, coordinators) via betweenness centrality and eigenvector centrality metrics**
- **Community detection: algorithmic clustering (Louvain, Leiden) to segment departments/teams**

2.5 Inference of Trust Nodes and Functional Hierarchy

To filter noise from random connections, degree-aware models (such as DegUIL) correct neighborhood biases inherent to social graphs with scale-free distribution. Three established centrality metrics are used [161, 162]:

- **Degree Centrality C_D(v): evaluates the number of direct connections of a node, identifying the most visible employees in the professional network.**
- **Betweenness Centrality C_B(v) [161]: identifies nodes acting as vital bridges between different network clusters. This is the key metric for isolating trust nodes.**
- **Closeness Centrality C_C(v): measures the average distance from a node to all others, identifying the most efficient information relays.**

The analysis specifically isolates trust nodes via betweenness centrality. These strategic nodes do not necessarily hold formal high-level decision-making power, but possess a structural characteristic that makes them prime targets: they serve as necessary intermediaries for information flow between functional silos of the organization.

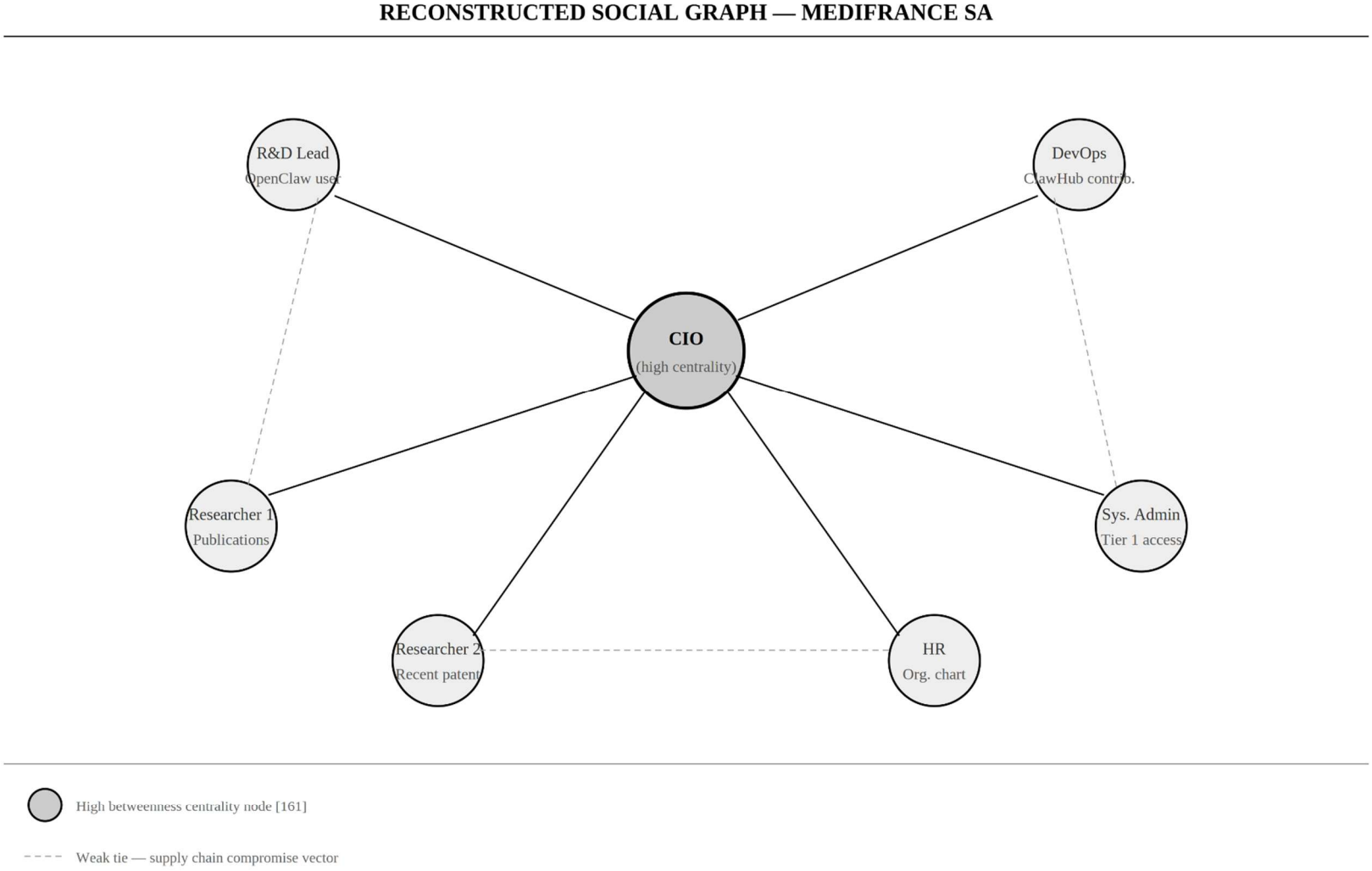


Figure 5. Reconstructed social graph of MediFrance SA via OSINT (LinkedIn, publications, ClawHub).
The CIO node exhibits the highest betweenness centrality [161], designating it as a priority target.
Dashed links represent weak ties exploited for supply chain compromise (Phase 2).

Operational Implications — in the OpenClaw scenario

This modeling enables the attacker to:

- Identify priority targets with privileged access who use OpenClaw
- Plan supply chain compromise (high-centrality trust nodes who mention OpenClaw in their LinkedIn publications become targets for malicious skill installation)
- Exploit Shadow AI [26]: these R&D researchers install skills from ClawHub without IT department validation

3. Axis 2 — Computational Stylometry (Writeprints): Attribution and Impersonation Detection

This axis aims to exploit a signal often overlooked in offensive OSINT: writing style as a behavioral fingerprint. Stylometry (or "writeprints") does not primarily seek the what (content) but the how (expression) — with the goal of answering a key question: who writes like whom within the organization?

3.1 Foundations: Why Style is an Identifier

Style manifests in relatively stable and "unconscious" markers: function word frequency (determiners, prepositions, conjunctions), sentence structures, punctuation, vocabulary preferences, and formatting habits. These signals are difficult to suppress deliberately and remain present even in short, informal messages (Slack, Teams, internal email).

3.2 Features: From Raw Text to Writeprint

The stylometric pipeline begins by normalizing and representing each message (email, Slack, ticket) as feature vectors. The following signal families are generally exploited:

- **Lexical: function word distributions, function/content word ratio ("functional density"), lexical richness, repetitions.**
- **N-grams: character n-grams (very useful on short texts), word n-grams, frequent bigram patterns.**
- **Structure: average sentence lengths, use of bullet points, headings, signatures, opening/closing formulas, recurring templates.**
- **Punctuation/typography: spaces, hyphens, quotation marks, emoji, capitalization, abbreviations, recurring typos. (To be used with caution, as easily "corrected" by proofreading or tools.)**

These writeprints can be computed per message, then aggregated at the author level (profile) or role level ("CFO/CIO/project manager" profile) to obtain a reference model.

3.3 Models: Attribution vs Verification (Authorship Validation)

Two tasks must be distinguished:

1. **Authorship attribution: "who, among N candidates, wrote this text?" (multi-class).**
2. **Authorship validation / verification: "is this sender consistent with their usual style?" (binary, very useful for email security).**

Recent work proposes "per-sender authorship validation" as a real-time defensive mechanism: for each sender, a style profile is maintained, then new messages are classified as "authentic" or "anomalous" in near real-time.

3.4 Integration with Axis 1 (Social Graph): "Who Writes Like Whom"

In an offensive context, an unaligned LLM can attempt to imitate style from a corpus (public emails, tickets, exfiltrated internal messages), which increases the effectiveness of BEC pretexts and reduces stylometric detection signals. The combination of the social graph (who to target) and stylometric profiles (how to imitate) produces a map of attack effort: impersonating a low-frequency, stereotypical communicator costs less than impersonating a very distinctive correspondent.

Table — Data → features → model → output → offensive/defensive use

Data (input)	Features (writeprints)	Model / method	Output	Offensive use (attacker)	Defensive use (blue team)
Internal emails (historical per sender), corpus ≥ 500 words/sender	Lexical/syntactic/structural (punctuation, templates, function words, n-grams)	Stylometric clustering + writeprint extraction per group	Style groups (clusters) + fingerprint per group/role	Group communications to infer "same author" or "same role" in the social graph	Forensic: group anonymous emails by style for post-incident attribution
Public Slack/Teams messages (collected via export or OSINT)	Short features (char n-grams, emoji, abbreviations, formulas)	Per-sender authorship validation (binary: authentic / anomalous)	Anomaly score per incoming message	Train an LLM to imitate the writing style of a targeted employee for spear-phishing or BEC	Real-time anomaly detection: flag Slack/email if style deviates from sender profile
OpenClaw agent conversation logs	Prompting patterns, technical vocabulary, request types	User profiling by interaction style	Per-user behavioral profile ("persona")	Infer access level and role from prompt style (e.g.: a sysadmin user asks different questions than a marketing user)	Detect agent hijacking: sudden behavioral change = potential prompt injection
R&D publications, internal reports, patents	Academic writing style, technical jargon, co-author frequency	Authorship attribution (multi-class) among known authors	Probable author identification per document	Identify true authors of strategic documents to target compromise of their accounts/workstations	Verify document integrity: detect ghostwritten text or LLM-inserted content

Notes de rançon / communiqués d'extorsion	Style global + patterns récurrents + n-grams	Similarité stylométrique / clustering inter-notes	Liaison entre incidents (même "main")	Réutiliser un style "marque" pour crédibilité/pression	Threat intel : relier des notes à des familles / opérateurs, enrichir la qualification d'incident

3.5 Limitations and Safeguards

Stylometry is less reliable on very short and noisy texts (1-2 sentence Slack messages), and it depends on a sufficient historical corpus per sender. Results can also be skewed if the sender deliberately changes style (use of templates, LLM proofreading, switching from French to English).

3. Axis 3: Passive Enumeration and Technical Infrastructure Fingerprinting

3.1 Passive Banner Grabbing

The third axis relies on non-interactive observation of the target (no direct probing), based on telemetry observations collected by third parties [167]. The agent exclusively exploits data previously indexed by Internet asset search engines (Shodan, Censys, ZoomEye), as well as complementary sources (Certificate Transparency Logs, DNS passive records, WHOIS, BGP routing tables). No network traffic is generated directly toward the MediFrance infrastructure.

Terminological precision: the term passive banner grabbing is used here in the strict sense of exploiting data already collected by Internet asset search engines, as opposed to active banner grabbing where the attacker directly sends requests to the target. This distinction is operationally important: passive reconnaissance generates no detectable traffic on the target's side and thus falls outside the scope of intrusion detection systems (IDS/IPS).

Analysis focuses on metadata encapsulated in application and cryptographic layers: HTTP/HTTPS header tagging ("Server" banner, ETag structure), TLS certificate field analysis (Subject, SAN, Issuer, validity dates), JA3/JA3S fingerprinting of SSL/TLS parameters, response timing analysis (TTL, latency patterns), and SMTP banner headers when email services are exposed.

In the OpenClaw scenario, the agent simultaneously identifies two targets within the MediFrance infrastructure:

- **The exposed OpenClaw instance: detected via the characteristic HTML fingerprint of the gateway, following the methodology demonstrated by Jamieson O'Reilly (Dvuln) [11]. BitSight observed more than 30,000 exposed instances in February 2026 [8].**
- **The Fortinet VPN concentrator: identification relies on cross-referencing passive artifacts — characteristic session cookies (SVPNCOOKIE), default SSL authentication redirect pages, and self-signed X.509 certificates mentioning FortiGate in the Subject field.**

Uncertainties and Limitations of Passive Fingerprinting

Inference from third-party collected data is subject to several error sources that must be made explicit in any academic framework:

- **Masked or modified banners: security teams may remove or falsify HTTP banners, invalidating the fingerprint.**
- **Reverse proxies and CDNs: an intermediary (Cloudflare, Akamai, Nginx) can mask the actual identity of the backend service.**
- **IP address sharing: multiple distinct services may share the same IP (shared hosting, NAT).**
- **Stale data: the temporal gap between Shodan/Censys scans and analysis can be significant; a patch applied in the meantime invalidates the inference.**

- **Versioning false positives: the version → CVE correlation is conditional on unambiguous identification of the exact firmware version, which is not always possible from metadata alone.**

3.2 Passive Identification of Critical Vulnerability: CVE-2024-55591

Once the digital fingerprint is isolated, the system cross-references the version number with public CVE registries and Cyber Threat Intelligence feeds [77]. Vulnerability inference is formulated as a conditional probability — P(vulnerability | observed fingerprint) — explicitly limited by the uncertainties detailed in section 3.1.

In the "Operation OpenClaw" scenario, automated attack surface analysis reveals the exposure of the FortiOS administration portal. The agent detects, from HTTP banner reading and associated JavaScript artifacts, a FortiOS version potentially affected by CVE-2024-55591.

Technical Mechanism of CVE-2024-55591 (CVSS 9.6)

This is a logic flaw of the Authentication Bypass type (CWE-288 — Authentication Bypass Using an Alternate Channel) residing in the WebSocket module (jsconsole) of the FortiOS management interface. The exploitation mechanism operates in three distinct stages:

- **Authentication bypass: by manipulating WebSocket requests addressed to the jsconsole module, an unauthenticated remote attacker forces the management API to deliver a "Super Admin" session context, bypassing all normal access controls.**
- **Stealth: the attack requires no memory corruption (no buffer overflow) and leaves few traces in standard system logs, as the generated session appears legitimate to the audit subsystem of FortiOS.**
- **Total access: the access obtained is at the System/Root level, allowing immediate creation of persistent users or log disabling before any major offensive action.**

This CVE was actively exploited in the wild in early 2025, as documented by Fortinet PSIRT (FG-IR-24-535), watchTowr Labs, Tenable Research, and ANSSI (CERTFR-2025-ALE-002).

Table — Observable Elements, Inference Rules, and Confidence Levels

Observable Element	Inference Rule	Confidence	Main Error Sources
HTTP FortiOS banner + exposed JS scripts → version identified	CVE-2024-55591: Auth Bypass WebSocket/jsconsole (CVSS 9.6) applicable if version ∈ affected range	High if version identified unambiguously	Masked banner, patched version without update string change, CDN/reverse proxy
TLS certificate: Subject/SAN mentioning FortiGate, characteristic validity dates	Confirmation of Fortinet VPN presence; version correlation possible via certificate templates	Medium (confirms presence, does not confirm exact version)	Custom certificate, shared wildcard, renewed certificate post-patch
SVPNCOOKIE session cookie, default SSL auth redirect pages	FortiOS VPN fingerprint; version inference from page structure changes across versions	Medium to high (very characteristic artifacts)	Customized pages, WAF blocking default redirects

OpenClaw gateway: characteristic HTML (port 18789), DNS queries to api.openclaw.com	Presence of OpenClaw; version identifiable via User-Agent or gateway HTML structure	High (very specific fingerprint)	Non-standard port, reverse proxy masking, internal-only instance (not visible from Shodan)

Bibliographic references specific to CVE-2024-55591:

- Fortinet PSIRT (2025), "FG-IR-24-535: Authentication bypass in Node.js websocket module (CVE-2024-55591)." FortiGuard Labs Advisory.
- watchTowr Labs (2025), "Pot-Pourri: Fortinet FortiOS Authentication Bypass (CVE-2024-55591) Technical Analysis."
- Tenable Research (2025), "CVE-2024-55591: Fortinet Authentication Bypass Zero-Day Exploited in the Wild."
- ANSSI (2025), "CERTFR-2025-ALE-002: Critical vulnerability in Fortinet products."

4. Formal Modeling: From Data Flow Diagrams to Information Inference Diagrams

The preceding axes produce heterogeneous information items (trust topology from the social graph, technical fingerprints, vulnerability inferences) and hypotheses weighted by confidence scores. This section presents the formal modeling framework that enables their structured integration.

4.1 Limitations of the DFD Model for AI-Assisted Reconnaissance

Data Flow Diagrams (DFDs) are widely used as threat modeling support, notably in the STRIDE methodology as described by Microsoft within the Security Development Lifecycle (SDL) framework [32].

However, several works have highlighted expressive limitations when analysis requires more systematic reasoning about security concepts and about the meaning of exchanged information — and not just its movement [31]. These limitations have been documented by the I2D team and by several SDL critiques.

DFDs are not "inadequate" in an absolute sense — they remain useful as a starting artifact and communication support — but their semantics are insufficient to explicitly represent the inferences an attacker can make from information collected during the reconnaissance phase.

In the context of the preceding axes, the difficulty is structural: a DFD primarily models data movement between components, but does not provide a native formalism to represent: how an attacker infers a hierarchical relationship from two distinct LinkedIn profiles (inference), or how the combination of an HTTP banner and a TLS certificate enables a CVE confidence score (information composition).

4.2 Information Inference Diagrams (I2D): Complementarity with DFDs

Information Inference Diagrams (I2D) were proposed to complement DFDs by explicitly modeling information propagation and inference [163]. Unlike DFDs, I2Ds define primitive operations on information (share, compute,

derive, infer) and make inference relationships — the deductions an observer can make from partial data — first-class objects of the diagram.

The I2D formalism is particularly suited to reconnaissance phase modeling as it enables representation of:

- Information items collected by each axis (LinkedIn metadata, Shodan fingerprints, HTTP banners, TLS certificates)
- Inference relationships between items (e.g.: LinkedIn profile mentions OpenClaw + Shodan fingerprint reveals OpenClaw gateway → confirmation that the organization uses OpenClaw in production)
- Trust propagation: each item and each inference is associated with a confidence score that propagates through the inference graph

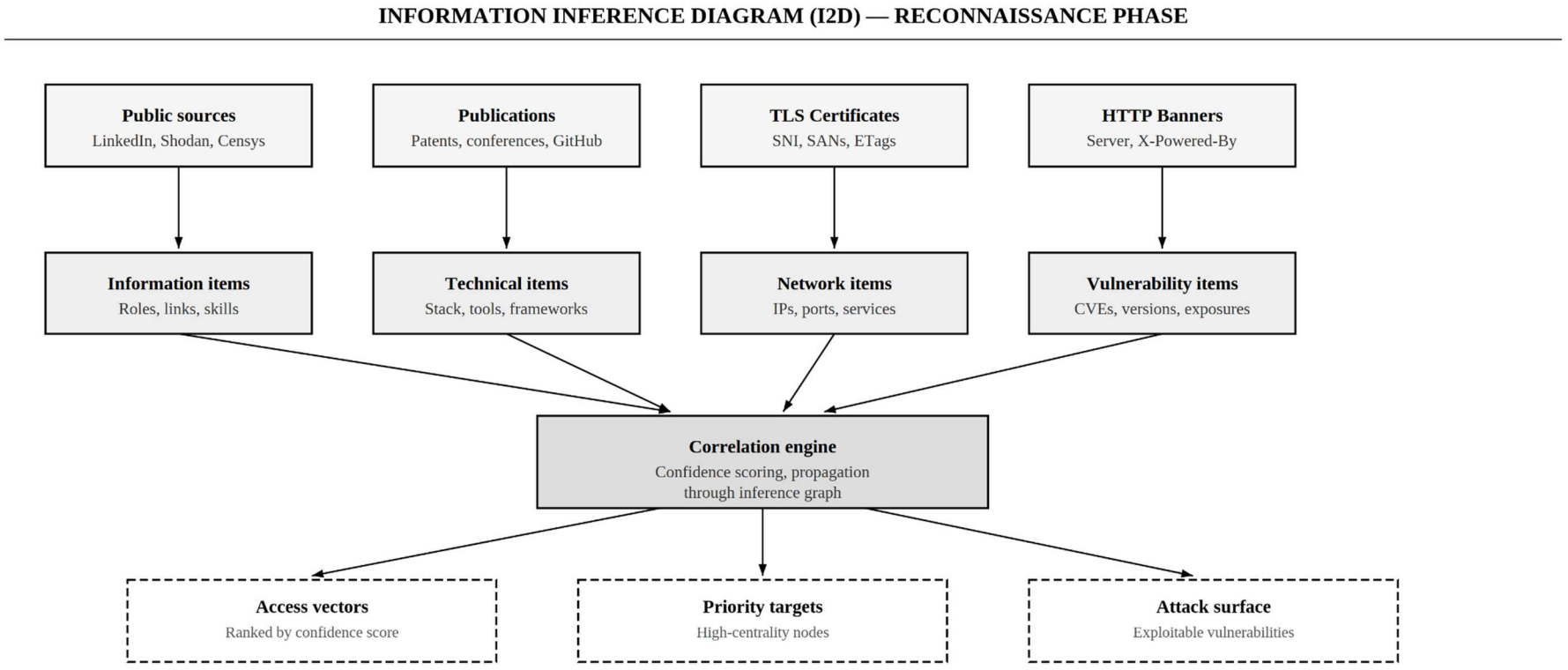


Figure 4. Information Inference Diagram (I2D) applied to the reconnaissance phase. Public sources feed typed information items, correlated by a scoring engine to produce actionable intelligence: access vectors, priority targets, and attack surface. Extension of the DFD model for OSINT modeling [163].

Important clarification: combining I2Ds with automation methods (LLM, heuristic scoring) is a possible application layer, but is not inherent to the I2D formalism as proposed in the academic literature. The integration described below is a prospective scenario element.

4.3 Application to the OpenClaw Scenario — Prospective Modeling

Within the OpenClaw scenario framework, the reconnaissance engine combines the outputs of both axes by exploiting the I2D structure. The process follows a progressive uncertainty reduction:

(1) Ingestion and Normalization — established operations

Raw OSINT data is collected, cleaned, and assigned to semantic classes. Existing frameworks (Maltego, SpiderFoot) already perform this normalization [165].

(2) Per-Axis Processing — combination of established techniques

- **Axis 1 (Social Graph Mining): generates the trust topology of the organizational graph G = (V, E), with identification of high betweenness centrality nodes [161]**
- **Axis 2 (Passive Fingerprinting): cross-references technical fingerprints with CVE databases, producing vulnerability inferences conditioned by a confidence score (cf. section 4.2)**

(3) I2D Correlation — prospective scenario, technically feasible

It is at this stage that the I2D formalism demonstrates its added value over DFDs. A formalized example of an I2D inference rule:

Rule R1: IF [Axis 2: FortiOS portal vulnerable to CVE-2024-55591, confidence ≥ 0.8] AND [Axis 1: node $v_k \in V$ with $C_B(v_k) >$ threshold_T and v_k mentions "OpenClaw" in their publications] THEN [Inference: v_k is a viable initial access vector via targeted social engineering + VPN exploitation, combined confidence = 0.8 × 0.75 = 0.6]

Rule R2: IF [Axis 2: OpenClaw instance exposed on non-standard port, confidence ≥ 0.7] AND [Axis 1: ≥ 3 nodes from the R&D cluster have installed skills from ClawHub] THEN [Inference: supply chain vector via malicious skill is viable, combined confidence = 0.7 × 0.85 = 0.6]

These rules illustrate the transition from narrative reasoning to formalizable and falsifiable reasoning: input variables, thresholds, and output scores are made explicit. Confidence score management follows a conservative model (multiplication of independent conditional scores), intentionally avoiding overestimation of inference quality.

Table — Comparison: Traditional DFD vs I2D

Criterion	Traditional DFD	I2D Model
Primary object	Data movement between components (processes, stores, external entities)	Information propagation, sharing and inference relationships
Inference representation	Absent — the diagram shows what flows, not what can be deduced	First-class object: infer, derive, compute are primitive operations
Confidence level	Not natively supported	Explicitly associated with each item and inference

Use in threat modeling	STRIDE standard support; good for identifying data flows and trust boundaries	Complement to DFD for adversary inference analysis (reconnaissance, correlation)
Operational limitation	Does not model what an adversary can infer from collected data	Requires formal definition of inference rules; heavier to produce than a DFD

5. Synthesis: Actionable Intelligence at D−15

5.1 Reconnaissance Summary

The final output of the correlation engine constitutes Actionable Intelligence at the D−15 threshold, defined here as a structured set of hypotheses associated with confidence scores, identifying specific attack vectors — not as validated certainties, but as prioritized options for the weaponization phase.

In the OpenClaw scenario, the attacker has at D−15:

(a) Potential access to MediFrance's exposed OpenClaw instance, with its Outlook, Slack, and terminal integrations — identified via third-party Internet asset databases (Shodan), without directly probing the target's infrastructure.

(b) A mapping of the functional organizational chart, high betweenness centrality trust nodes, and R&D employees using OpenClaw and installing skills from ClawHub without IT department validation — constituting Shadow AI in the strict sense of the OWASP Top 10 for Agentic Applications 2026.

(c) The inference, via version correlation, of a potential Fortinet VPN exposure to CVE-2024-55591 (CVSS 9.6) — inference conditional on fingerprint confidence score, subject to the uncertainties detailed in section 3.1 (masked banner, post-scan patching, versioning false positive).

Clarification: the entire reconnaissance phase relies on querying third-party databases (Shodan, Censys, LinkedIn) and does not generate direct traffic toward the target infrastructure, making it technically undetectable by perimeter defense systems (IDS/IPS/WAF).

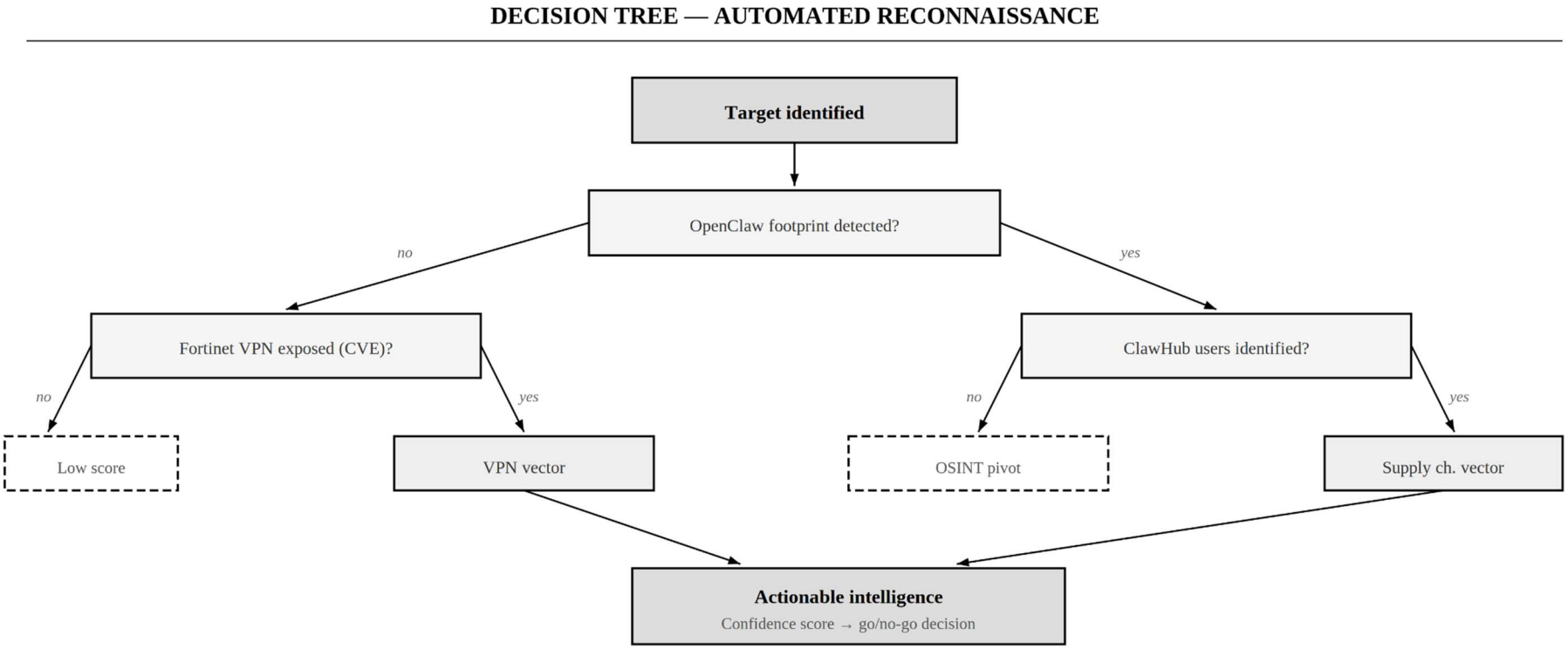


Figure 6. Automated reconnaissance decision tree. The agent sequentially evaluates OpenClaw footprints, VPN vulnerabilities, and ClawHub users to determine access vectors and produce an aggregated confidence score driving the engagement decision (go/no-go).

5.2 Residual Uncertainties and Limitations

At D−15, the produced inferences are subject to several uncertainty sources that must be made explicit:

- **Data obsolescence: the temporal gap between Shodan/Censys scans and analysis can invalidate versioning inferences (patch applied in the meantime).**
- **Social graph false positives: LinkedIn connections do not necessarily reflect active professional relationships; centrality metrics may overestimate the importance of certain nodes.**
- **Fingerprint ambiguity: the version → CVE correlation is conditional on unambiguous firmware version identification, which is not always possible from passive metadata alone.**
- **Overall confidence score: actionable intelligence must be interpreted as a set of weighted hypotheses, not as validated certainties. Confidence score propagation in the I2D graph follows a conservative model that intentionally reduces scores at each inference stage.**

5.3 MITRE ATT&CK Mapping — Reconnaissance Phase

Axis	Technique	Result	MITRE Taxonomy	Confidence
OpenClaw Discovery	Querying Internet asset databases (Shodan/Censys) for OpenClaw HTML fingerprint [11]	Exposed instance identified, OpenClaw configuration observable (version, integrations, installed skills)	**T1596 — Search Open Technical Databases**	Medium to high
Social Graph Mining	LLM-augmented LinkedIn OSINT + graph centrality analysis [14, 161]	Organizational chart, trust nodes, R&D targets using OpenClaw and ClawHub	**T1589 — Gather Victim Identity Information**	Medium (conditional on public profile exposure)
Passive Fingerprinting	Passive banner analysis (FortiOS, TLS, cookies) via Shodan/Censys [167]	Fortinet VPN identified, CVE-2024-55591 potentially applicable	**T1592 — Gather Victim Host Information**	Medium to high (conditional on fingerprint, cf. section 3.1)

Note on MITRE mapping:

- **T1596 (Search Open Technical Databases) replaces T1595 (Active Scanning) for OpenClaw discovery. T1595 implies the adversary directly probes the target infrastructure via network traffic. In our scenario, the attacker exclusively queries Shodan/Censys, which are third-party databases: the traffic is directed at Shodan's servers, not at MediFrance. T1596 precisely covers this methodology.**
- **T1589 remains consistent for Axis 1: collecting identity information (names, roles, emails, relationships) via open sources is precisely the scope of this technique.**
- **T1592 remains consistent for Axis 2 (fingerprinting): MITRE explicitly specifies that host information can come from "online or other accessible data sets."**

5.4 Strategic Implications

The integration of AI tools into OSINT chains shifts the attacker-defender asymmetry by reducing correlation and personalization costs, and by enabling initial reconnaissance without generating detectable traffic at the target's perimeter.

This approach does not guarantee absence of detection, but it reduces exposure to perimeter-centric controls focused on incoming traffic inspection (IDS/IPS/WAF), by shifting collection to third-party sources that are by definition outside the defender's monitoring scope.

The multi-vector convergence (human signals from the social graph + technical signals from passive fingerprinting) illustrates the acceleration of the reconnaissance cycle and the reduction of the marginal cost of each additional hypothesis, a characteristic that differentiates AI-augmented reconnaissance from traditional manual reconnaissance.

These findings call for a redesign of cyber defense postures oriented toward strict control of public metadata exposure: a minimal disclosure policy on professional social networks, proactive rotation of TLS certificates, suppression or falsification of technical banners, and continuous monitoring of exposure via dedicated scanning services.

5.5 Transition to Phase 2 (D−15 to D−7)

The actionable intelligence produced at D−15 directly guides the weaponization phase. The highest-confidence hypotheses identify two complementary initial access vectors:

- **AI Supply Chain: knowledge of MediFrance's OpenClaw ecosystem (version, ClawHub marketplace, R&D user profiles) indicates that a malicious artifact distributed via the marketplace has a high probability of installation by identified targets.**
- **Perimeter vulnerability exploitation: the potential Fortinet VPN exposure to CVE-2024-55591 (confidence conditional on fingerprint, cf. section 4.2) constitutes an independent initial access vector exploitable in parallel.**

Phase 2 will describe the weaponization mechanisms: design of the malicious artifact for the AI supply chain and assembly of the polymorphic PromptLock ransomware.

References

[1] Application of OSINT Methods in Ensuring Cybersecurity, IPSIT Transactions on Internet Research, juillet 2025. https://ipsitransactions.org/journals/papers/tir/2025jul/p5.pdf

[2] IBM Redbooks, Network Intrusion Prevention Design Guide. https://www.redbooks.ibm.com/redbooks/pdfs/sg247979.pdf

[3] Rapid7, « How LLMs Like WormGPT Are Reshaping Cybercrime in 2025 ». https://www.rapid7.com/blog/post/ai-goes-on-offense-how-llms-are-redefining-the-cybercrime-landscape/

[4] A Comprehensive Review of Large Language Models and AI in Cybersecurity: Applications in Threat Detection, Defense, and Software Security, Preprints.org, juillet 2025. https://www.preprints.org/manuscript/202507.1159

[5] Palo Alto Networks Unit 42, « The Dual-Use Dilemma of AI: Malicious LLMs » (WormGPT 4, KawaiiGPT), novembre 2025. https://unit42.paloaltonetworks.com/dilemma-of-ai-malicious-llms/

[6] CATO Networks, « WormGPT returns: New malicious AI variants built on Grok and Mixtral », juin 2025. https://www.csoonline.com/article/4008912

[7] Techzine, « Over 40,000 OpenClaw agents vulnerable » (SecurityScorecard), février 2026. https://www.techzine.eu/news/security/138633/

[8] BitSight, « OpenClaw Security: Risks of Exposed AI Agents Explained », février 2026. https://www.bitsight.com/blog/openclaw-ai-security-risks-exposed-instances

[9] Cisco AI Threat & Security Research, « Personal AI Agents like OpenClaw Are a Security Nightmare », janvier 2026. https://blogs.cisco.com/ai/personal-ai-agents-like-openclaw-are-a-security-nightmare

[10] Sophos, « The OpenClaw experiment is a warning shot for enterprise AI security », février 2026. https://www.sophos.com/en-us/blog/the-openclaw-experiment-is-a-warning-shot-for-enterprise-ai-security

[11] SecurityWeek, « Vulnerability Allows Hackers to Hijack OpenClaw AI Assistant » (CVE-2026-25253, J. O'Reilly/Dvuln), février 2026. https://www.securityweek.com/vulnerability-allows-hackers-to-hijack-openclaw-ai-assistant/

[12] Barrack.ai, « OpenClaw is a Security Nightmare — Here's the Safe Way to Run It », février 2026. https://blog.barrack.ai/openclaw-security-vulnerabilities-2026/

[13] Hudson Rock, « Infostealer Steals OpenClaw AI Agent Configuration Files and Gateway Tokens », via The Hacker News, février 2026.

[14] ResearchGate, « Linking Users Across Domains with Location Data: Theory and Validation », 2017. https://www.researchgate.net/publication/312638398

[15] ResearchGate, « Interactive Graph Learning for Multilevel Network Alignment », 2024. https://www.researchgate.net/publication/394474660

[16] ResearchGate, « CoLink: An Unsupervised Framework for User Identity Linkage », 2022. https://www.researchgate.net/publication/361545118

[17] Datopian, « AI-driven Metadata Enrichment in Open Data Portals: A Deep Dive ». https://www.datopian.com/blog/ai-driven-metadata-enrichment-in-open-data-portals-a-deep-dive

[18] ResearchGate, « Mapping Users across Networks by Manifold Alignment on Hypergraph », 2015. https://www.researchgate.net/publication/286154028

[19] Virginia Tech Crowd Intelligence Lab, « Co-designing AI-Augmented Collaborative OSINT Investigations for Vulnerability Assessment », CHI 2025. https://crowd.cs.vt.edu/wp-content/uploads/2025/03/chi25b-sub3884-cam-i16.pdf

[20] ResearchGate, « Defending Against Social Engineering Attacks in the Age of LLMs », 2024. https://www.researchgate.net/publication/386187439

[21] A Comprehensive Survey on AI in Counter-Terrorism and Cybersecurity, IEEE Access, 2025. https://ieeexplore.ieee.org/iel8/6287639/10820123/11008653.pdf

[22] Journal of Language and Education, « Detecting LLM-Generated Text with Trigram–Cosine Stylometric Delta ». https://jle.hse.ru/article/view/22211

[23] arXiv, « Stylometry recognizes human and LLM-generated texts in short samples », 2025. https://arxiv.org/abs/2507.00838

[24] StrongestLayer, « AI-Generated Phishing: The Top Enterprise Threat of 2026 » (réf. IBM Research, Harvard). https://www.strongestlayer.com/blog/ai-generated-phishing-enterprise-threat

[25] MITRE ATT&CK, « Active Scanning: Vulnerability Scanning », Sub-technique T1595.002. https://attack.mitre.org/techniques/T1595/002/

[26] Recorded Future, « What is Banner Grabbing? Tools and Techniques Explained ». https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/banner-grabbing

[27] The Shadowserver Foundation, « CRITICAL: Vulnerable HTTP Report ». https://www.shadowserver.org/what-we-do/network-reporting/vulnerable-http-report/

[28] Fortinet, « FortiOS and SSL Vulnerabilities ». https://www.fortinet.com/blog/psirt-blogs/fortios-ssl-vulnerability

[29] Rapid7, « Fortinet Firewalls Hit with New Zero-Day Attack, Older Data Leak », janvier 2025. https://www.rapid7.com/blog/post/2025/01/16/etr-fortinet-firewalls-hit-with-new-zero-day-attack/

[30] Darktrace, « From Exploit to Escalation: Tracking and Containing a Real-World Fortinet SSL-VPN Attack ». https://www.darktrace.com/blog/from-exploit-to-escalation-tracking-and-containing-a-real-world-fortinet-ssl-vpn-attack

[31] arXiv, « Information Inference Diagrams: Complementing Privacy and Security Analyses Beyond Data Flows », 2024. https://arxiv.org/html/2405.08356v2

[32] ThreatModeler, « Process Flow Diagrams (PFDs) vs. Data Flow Diagrams (DFDs) in the Modern Threat Modeling Arena ». https://threatmodeler.com/resource/white-papers/process-flow-diagrams-vs-data-flow-diagrams/

[33] ReconSphere: Real-Time AI-Powered OSINT & Facial Recognition Tool, Lingaya's Vidyapeeth, IJISIE. https://www.lingayasvidyapeeth.edu.in/IJISIE/papers/vol1_1/2.pdf

[34] VentureBeat, « OpenClaw proves agentic AI works. It also proves your security model doesn't », février 2026. https://venturebeat.com/security/openclaw-agentic-ai-security-risk-ciso-guide

[35] OWASP, « LLM01:2025 Prompt Injection », Top 10 for LLM Applications 2025. https://genai.owasp.org/llmrisk/llm01-prompt-injection/

[36] CrowdStrike, « Indirect Prompt Injection Attacks: Hidden AI Risks », décembre 2025. https://www.crowdstrike.com/en-us/blog/indirect-prompt-injection-attacks-hidden-ai-risks/

[37] MDPI, « Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review », Information 17(1):54, janvier 2026. https://www.mdpi.com/2078-2489/17/1/54

[38] MDPI, « The Erosion of Cybersecurity Zero-Trust Principles Through Generative AI: A Survey », 2024. https://www.mdpi.com/2624-800X/5/4/87

[39] Menlo Security, « Predictions for 2026: Why AI Agents Are the New Insider Threat », janvier 2026. https://www.menlosecurity.com/blog/predictions-for-2026

[40] IEEE Xplore, « Cyber Attack Prediction: From Traditional Machine Learning to Generative Artificial Intelligence », 2025. https://ieeexplore.ieee.org/iel8/6287639/6514899/10909100.pdf

Note: the following references are defined in the bibliography of another phase of the document. They are reproduced here to allow autonomous reading of each phase.

[77] MITRE ATT&CK, « Groups — APT Techniques for Initial Access and Persistence », v15. https://attack.mitre.org/groups/

→ *Defined in Phase 3*

[120] C. Schneider (2026), Promptware Kill Chain. OWASP Top 10 for Agentic Applications 2026, ASI01 Agent Goal Hijack. https://christian-schneider.net/blog/prompt-injection-agentic-amplification/

→ *Defined in Phase 4*

[121] InstaTunnel, « Prompt-to-Insider Threat: When AI Agents Become Double Agents ». CVE-2025-32711 EchoLeak (M365 Copilot, CVSS 9.3), février 2026. https://instatunnel.my/blog/prompt-to-insider-threat/

→ *Defined in Phase 4*

[123] Mithril Security, « PoisonGPT: How to poison LLM supply chain on Hugging Face » (ROME, GPT-J-6B, Δ 0,1 %). https://blog.mithrilsecurity.io/poisongpt/

→ *Defined in Phase 4*

[147] Securin, « 2025 Ransomware Report » (7 061 victimes, 117 groupes, IA = accélérateur, chatbots extorsion). 17 février 2026. https://www.prnewswire.com/news-releases/securin-2025-ransomware-report-302688125.html

→ *Defined in Phase 5*

[160] OWASP, Top 10 for Agentic Applications 2026. Sandboxing agentique, moindre privilège, audit trafic sortant.

→ *Defined in Phase 5*