Enhancing Technology Fingerprinting Through Human-Generated OSINT: A Comparative Study Against Tool-Only Reconnaissance

Yash Mahesh Malpathak

A thesis submitted in partial fulfilment of the
requirements for the degree of

Master of Science

University of Washington
2026

Committee:

Dr. Min Chen
Dr. Marc Dupuis
Dr. Jeffrey Kim

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

ABSTRACT

Enhancing Technology Fingerprinting Through Human-Generated OSINT:
A Comparative Study Against Tool-Only Reconnaissance

Yash Mahesh Malpathak

Chair of the Supervisory Committee:

Dr. Min Chen

Computing and Software Systems

Technology fingerprinting, the identification of software frameworks, cloud platforms, programming languages, and infrastructure components used by an organization, is a foundational stage of cybersecurity reconnaissance. Automated scanning tools such as Nmap, WhatWeb, and Subfinder are widely used for this purpose. However, these tools share a structural limitation: they primarily detect technologies that produce externally observable signals at the network perimeter, while application-layer frameworks, backend infrastructure, and internal development tooling may remain invisible to tool-based reconnaissance.

This thesis investigates whether human-generated open-source intelligence (OSINT), specifically job postings, LinkedIn employee profiles, and GitHub public repositories, can serve as a complementary signal class for improving technology fingerprinting. An empirical study was conducted across twelve purposively selected organizations. Three integration methods were designed and evaluated: a union-based model for maximum recall, a source-strength weighted model using empirically derived reliability values, and a rule-based evidence-tier model requiring no manually assigned numeric scores.

Results show that the average Jaccard similarity between tool-detected and OSINT-detected technology sets is 0.005, with zero overlap in ten of twelve organizations. This confirms that the two signal classes are largely complementary rather than redundant. Human-generated OSINT contributed 857 organization-level detections absent from the tool-only baseline, representing an average coverage gain of 94.1%. The dominant OSINT-only categories were programming languages at 34.66% and application frameworks at 17.74%. All three integration methods produced substantially richer fingerprints than tool-only reconnaissance, with integration coverage gains ranging from 53.85% to 96.35%. Only five technologies across the full dataset achieved cross-source validation between tools and OSINT.

These findings demonstrate that human-generated OSINT provides a quantifiably valuable complement to active scanning in technology fingerprinting workflows.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Min Chen for her guidance, encouragement, and invaluable feedback throughout this research. Her expertise in research and her willingness to engage critically with the methodology and findings of this work have been instrumental in shaping this thesis into a defensible and rigorous piece of research.

I would also like to thank the members of my thesis committee, Dr. Marc Dupuis and Dr. Jeffrey Kim, for their time, thoughtful engagement, and constructive input throughout the review process.

I am grateful to the faculty and staff of the Cybersecurity Engineering program at the University of Washington, Bothell for providing an intellectually stimulating environment and for their support throughout my graduate studies.

Finally, I would like to thank my family for their unwavering patience and encouragement throughout this journey.

Table of Contents

Chapter 1 Introduction

1.1 Background and Motivation

1.1.1 The Role of Technology Fingerprinting in Cybersecurity

In modern cybersecurity practice, understanding the technological composition of a target organization is a foundational prerequisite for effective security assessment, threat modeling, and attack surface analysis. The process of identifying the software frameworks, web platforms, cloud infrastructure, programming languages, and third-party services used by an organization, commonly referred to as technology fingerprinting, constitutes the first stage of most penetration testing and vulnerability assessment workflows. Accurate technology fingerprinting enables security professionals to identify potentially vulnerable components, map exposed attack surfaces, prioritize remediation efforts, and establish a baseline understanding of organizational risk exposure [1].

The significance of technology fingerprinting is further underscored by its role within established adversarial frameworks. The MITRE ATT&CK framework classifies reconnaissance-phase activities, including active and passive technology identification, as the earliest stages of the attack lifecycle [18]. Threat actors who can accurately enumerate an organization's technology stack gain a measurable operational advantage: they can select exploits with higher success probability, identify unpatched software versions, and target known vulnerabilities in specific platform configurations before a defender is aware of the exposure [5]. From the defender's perspective, this same intelligence, when gathered proactively, enables pre-emptive hardening and informed vulnerability prioritization.

The attack surface of an organization, as formally characterized by Manadhata and Wing, is defined as the set of methods and data items that an unauthorized user can use to enter or extract data from a computing environment [25]. Technology fingerprinting directly informs this measurement by identifying the specific technologies that constitute entry points or data exposure vectors within that surface. As organizations increasingly distribute their infrastructure across cloud platforms, containerized services, and third-party integrations, the completeness and accuracy of technology fingerprinting have become correspondingly more critical and correspondingly more difficult to achieve through conventional means alone.

1.1.2 Limitations of Active Scanning Tools

Traditional approaches to technology fingerprinting rely on a class of automated reconnaissance tools that interrogate publicly exposed, internet-facing systems to infer their technology composition from observable signals. Widely adopted tools in this category include Nmap [19], which performs network port scanning and service version detection; WhatWeb [20], which identifies web technologies from HTTP response headers and page content patterns; BuiltWith [21], which aggregates technology signals from DNS records, HTTP headers, and script tags to produce technology profile reports for web properties; and Subfinder

[26], which enumerates subdomains to expand the observable attack surface. These tools have proven effective and widely adopted within professional security workflows for identifying surface-level technologies including web servers, content delivery networks, and publicly exposed application frameworks.

However, this class of tools shares a structural limitation that constrains the completeness of any fingerprinting effort conducted exclusively through their use. Active scanning tools can only identify technologies that produce observable, externally accessible signals, meaning technologies that manifest in HTTP headers, open ports, server banners, certificate metadata, or detectable behavioural patterns at the application boundary. Technologies deployed behind reverse proxies, API gateways, load balancers, or internal network segments produce none of these signals and are therefore systematically invisible to active scanning. Similarly, internal development frameworks, backend data processing pipelines, database management systems, authentication services, and cloud-native infrastructure components are rarely exposed at the perimeter layer and consequently remain undetected [3].

This limitation has direct consequences for attack surface completeness. An organization that relies on active scanning alone for technology fingerprinting will produce an inventory that accurately reflects its public-facing perimeter while remaining blind to potentially vulnerable technologies operating in its internal layers. Security professionals and threat actors alike are increasingly aware that high-value targets, including sensitive data stores, identity management systems, and internal APIs, frequently reside precisely in these internal layers [3]. A fingerprinting methodology that cannot reach these layers produces an incomplete and potentially misleading picture of organizational risk.

Beyond the perimeter visibility limitation, active scanning tools are also constrained to identifying technologies that their signature databases recognize. Emerging frameworks, custom-built internal tools, proprietary software, and recently released platforms may not yet appear in tool signature sets, resulting in false negatives that compound the coverage gap. Prior studies have documented detection rate ceilings for active scanning tools in enterprise environments, with tool-only methods consistently missing significant proportions of the full technology stack when compared against reference inventories [2].

1.2 Problem Statement

Despite the widespread adoption of automated technology fingerprinting tools in professional security workflows, a significant and consistently reproducible gap exists between the technology inventory produced by tool-based reconnaissance and the full technology stack actually in use by target organizations. This gap is not attributable to tool malfunction or misconfiguration; it is a structural consequence of the fundamental operating principle shared by all active scanning tools: they can only observe what the perimeter exposes.

At the same time, organizations routinely publish detailed information about their internal technology stacks through human-generated public channels as a natural consequence of their operational activities. Job postings advertise the specific frameworks, platforms, and languages

that engineering teams use and seek expertise in. LinkedIn profiles of employees enumerate technologies used in day-to-day roles. Public GitHub repositories contain dependency files, configuration manifests, and infrastructure-as-code scripts that directly disclose technology choices. This class of signals, referred to throughout this thesis as human-generated Open-Source Intelligence (OSINT), represents a category of technology disclosure that is both publicly accessible and largely orthogonal to the signals captured by active scanning tools [7].

The central problem addressed by this thesis is the absence of a systematic, empirically evaluated methodology for integrating human-generated OSINT signals with active scanning tool outputs to produce a more complete technology fingerprint of target organizations. While individual OSINT sources have been studied in isolation in the prior literature, and while active scanning tools have been extensively benchmarked against one another, no prior study has conducted a controlled empirical comparison of the technology coverage achieved by tool-only reconnaissance against the coverage achievable through structured integration of human-generated OSINT, nor has any prior work proposed and evaluated multiple integration methodologies for combining these two signal classes [9].

It is important to clarify the scope of this research. This thesis focuses exclusively on technology fingerprinting, the identification of what technologies an organization uses, and does not extend to vulnerability detection, exploitation assessment, or active security testing. The OSINT sources examined here are used solely to identify technology stack components, not to assess whether those technologies contain vulnerabilities or are exploitable. Vulnerability assessment is a subsequent and separate stage of the security assessment lifecycle that falls outside the scope of this work

1.3 Research Objectives

1.3.1 Primary and Secondary Research Questions

This thesis is organized around two primary research questions derived directly from the problem statement.

RQ1: To what extent do human-generated OSINT sources, specifically job postings, LinkedIn employee profiles, and GitHub repositories, reveal technology stack information about target organizations that traditional active scanning tools fail to identify?

RQ2: What methodologies can effectively integrate human-generated OSINT with traditional reconnaissance tools to improve technology fingerprinting?

RQ1 is fundamentally a measurement question. It requires constructing a reference technology inventory based on consolidated public evidence for each target organization and systematically measuring the proportion of that inventory detected by tool-only reconnaissance against the proportion revealed by human-generated OSINT sources. The answer to RQ1 establishes whether the problem identified in the problem statement is empirically real and quantifiably significant, or whether the gap between tool-based and OSINT-based coverage is too small to justify a new methodology.

RQ2 is a design and evaluation question. It presupposes that RQ1 confirms a meaningful coverage gap and asks which of several candidate integration approaches best closes that gap. Three integration models of increasing sophistication are designed, implemented, and evaluated: a union-based model, a source strength-weighted model, and an evidence-tier model that classifies each detected technology based on the number and type of independent sources that corroborate it. The comparative evaluation of these three models constitutes the quantitative core of the thesis.

1.3.2 Scope and Boundaries

This research is scoped to publicly accessible information about real organizations gathered through ethical, non-intrusive means. All active scanning was conducted only against publicly accessible endpoints using non-intrusive, rate-limited methods, and no authentication bypass, exploitation, or access-controlled probing was performed. All OSINT collection drew exclusively from publicly available sources without authentication bypass or scraping of access-controlled content.

The study was conducted across a purposively selected sample of twelve organizations spanning diverse industry verticals, organizational sizes, and expected OSINT visibility profiles. This purposive sampling strategy was chosen to maximize analytical diversity and to ensure that findings are not artifacts of a single industry's disclosure patterns. The study is explicitly framed as an exploratory empirical investigation; it does not seek to make statistical inference claims about a broader population but rather to demonstrate the existence and magnitude of the coverage gap and to evaluate integration methodologies against a representative and diverse sample.

1.4 Proposed Approach

This thesis proposes and evaluates a structured pipeline for enhancing technology fingerprinting through the systematic collection, normalization, and integration of human-generated OSINT signals alongside traditional active scanning tool outputs. The pipeline operates in four stages.

In the first stage, active scanning is performed against each target organization using a standardized tool suite comprising Nmap, WhatWeb, Subfinder, and curl header inspection. The outputs of these tools are parsed, deduplicated, and normalized into a common technology signal schema, producing a tool-only baseline technology inventory for each organization.

In the second stage, human-generated OSINT is collected from three source categories. Job posting data is gathered from publicly accessible employment platforms, with postings filtered to engineering and technical roles within each target organization. Structured technology tags, required skill keywords, and platform mentions are extracted from posting text using a defined parsing methodology. LinkedIn employee profile data is collected through structured manual sampling of technical employees, with skill endorsements and listed technologies extracted and mapped to the common schema. GitHub repository data is collected by identifying public repositories associated with each organization and extracting technology signals from dependency files, README documentation, and infrastructure manifests.

In the third stage, the collected OSINT signals are normalized against the same schema used for tool outputs. Technology names are standardized, duplicate signals are removed, and each signal is mapped to a common category such as programming language, framework, cloud service, database, CI/CD tool, or security technology. This normalization step ensures that outputs from heterogeneous sources can be compared and integrated consistently across organizations.

In the fourth stage, three integration models are applied to combine tool signals and OSINT signals into unified technology inventories. The union-based model aggregates all signals without weighting. The source-strength weighted model ranks technologies using empirically derived source-strength values computed from agreement rate and category coverage across the collected dataset. The evidence-tier model classifies each technology into one of four evidence categories based on observable, rule-based criteria: technologies detected by both tools and at least one OSINT source are classified as Cross-Validated; those detected by two or more independent OSINT sources are classified as Strong OSINT Evidence; those detected by tools only or a single source with strong contextual evidence are classified as Moderate Evidence; and those detected by LinkedIn only or a single low-context source are classified as Weak Evidence. No manually assigned numeric scores or bonuses are used in this classification. The performance of each model is evaluated against the reference inventory using Jaccard similarity and coverage percentage metrics.

1.5 Key Contributions

This thesis makes three contributions to the fields of cybersecurity reconnaissance and OSINT-based security analysis.

The first contribution is an empirical quantification of the technology coverage gap between tool-only active scanning and human-generated OSINT across a diverse sample of twelve real organizations. Prior work has asserted this gap theoretically; this thesis measures it directly and demonstrates that the Jaccard similarity between tool-detected and OSINT-detected technology inventories approaches zero across the majority of studied organizations, confirming that the two signal classes are largely complementary rather than redundant.

The second contribution is the design and empirical evaluation of three integration methodologies of increasing sophistication for combining tool-based and OSINT-based technology signals. The evaluation examines whether evidence-tier integration, which classifies detected technologies through rule-based multi-source corroboration criteria, provides stronger and more interpretable results than simpler alternatives. It also evaluates whether integration of any kind improves coverage completeness relative to tool-only baselines, with coverage increases observed consistently across the studied organizations.

The third contribution is a replicable data collection and normalization pipeline for human-generated OSINT in the technology fingerprinting domain. The pipeline's methodology for standardizing technology names, categorizing signals, removing duplicates, and integrating heterogeneous source outputs provides a defensible and replicable foundation for future work that seeks to extend or adapt the integration framework.

The primary beneficiaries of this research are security professionals conducting attack surface assessments and penetration testing engagements, threat intelligence analysts building organizational risk profiles, and security researchers studying the information disclosure implications of human-generated OSINT.

1.6 Thesis Structure Overview

The remainder of this thesis is organized as follows. Chapter 2 surveys the existing literature on technology fingerprinting, OSINT in cybersecurity, human-generated information disclosure, and multi-source data fusion techniques, positioning this research relative to prior work and identifying the specific gaps it addresses. Chapter 3 presents the research methodology in full, covering the data collection protocols for each source type, the reference inventory construction process, the normalization pipeline, the design of the three integration models, and the evaluation metrics. Chapter 4 describes the system design and research implementation of the data collection and integration pipeline, documenting each module's research purpose, inputs, outputs, and connection to the methodology described in Chapter 3. Chapter 5 presents the results and evaluation for both research questions, including the tool-versus-OSINT coverage comparison addressing RQ1 and the comparative evaluation of the three integration methods addressing RQ2. Chapter 6 discusses the findings in relation to the two research questions, addresses limitations of the study, and considers the ethical dimensions of OSINT-based reconnaissance research. Chapter 7 concludes the thesis with a summary of contributions and directions for future work.

Chapter 2 Literature Review

This chapter surveys the body of research relevant to the present study across five thematic areas: technology reconnaissance and attack surface measurement, open-source intelligence in cybersecurity, human-generated OSINT sources, cyber threat intelligence and threat hunting, and multi-source data fusion and evidence integration. Each section identifies how prior work informs, motivates, or situates the methodology and contributions of this thesis. The chapter concludes by identifying the specific research gap that the present study addresses.

2.1 Technology Reconnaissance and Attack Surface Measurement

A foundational body of literature addresses the techniques, tools, and limitations of internet-facing technology reconnaissance. These works collectively establish the operational context within which the present research is situated and provide the benchmarks against which OSINT-enhanced fingerprinting is evaluated.

Safaei Pour et al. [1] present a comprehensive survey of internet measurement techniques applied to cybersecurity, covering active and passive scanning methodologies, network probing, and service enumeration approaches. Their survey documents the evolution of scanning infrastructure and demonstrates that active measurement, while powerful for identifying publicly exposed services, is inherently bounded by the visibility of the perimeter layer. This limitation is central to the motivation of the present study. Safaei Pour et al. further note that the majority of measurement research focuses on internet-facing assets, leaving internal and application-layer technologies systematically understudied. Their work provides the empirical basis for treating active scanning as a necessary but insufficient foundation for comprehensive technology fingerprinting.

Tools such as Nmap [19], WhatWeb [20], BuiltWith [21], and similar platform-specific scanners represent the current state of practice in automated technology fingerprinting. Nmap performs network port scanning and service version detection, operating on the principle that observable port states and service banners disclose sufficient information to identify running software. WhatWeb extends this approach to the web application layer, analyzing HTTP response headers, HTML meta tags, and JavaScript inclusions to infer web server and application framework identities. BuiltWith and similar platform-specific scanners aggregate technology signals from DNS records, HTTP headers, and script tags to produce technology profile reports for web properties. In this thesis, the implemented tool-only baseline uses Nmap, WhatWeb, Subfinder, and curl header inspection specifically, to maintain full methodological reproducibility and avoid dependency on commercial access tiers, as described in Section 3.3.1.

Makrushin [3] specifically examines the use of open-source intelligence for attack surface analysis and monitoring, arguing that organizations' digital footprints extend well beyond what automated scanning tools can observe. Makrushin's framework for attack surface analysis through OSINT provides a conceptual bridge between the reconnaissance literature and the OSINT literature, and directly motivates the integration approach developed in this thesis. Notably, Makrushin identifies the human-generated layer of organizational information

disclosure as the least systematically exploited component of the attack surface, a gap this research explicitly targets.

2.2 Open-Source Intelligence in Cybersecurity

The use of open-source intelligence in cybersecurity contexts has received growing scholarly attention, with several systematic reviews and empirical studies establishing both the value and the limitations of OSINT-based approaches.

Browne et al. [2] conducted a systematic review of research utilizing artificial intelligence for OSINT applications in cybersecurity, analyzing a large corpus of peer-reviewed studies to identify application domains, methodological approaches, and open challenges. Their review establishes that OSINT is applied across a wide range of cybersecurity functions including threat intelligence, vulnerability identification, attribution, and attack surface mapping. Browne et al. find that the majority of OSINT research in cybersecurity focuses on technical signals, such as dark web postings, threat feeds, and vulnerability databases, and that comparatively little attention has been paid to human-generated OSINT as a source of organizational technology intelligence. This finding directly positions the present study as addressing a recognized gap in the literature.

Avrahami et al. [8] examine strategies and solutions for leveraging OSINT in proactive cybersecurity, arguing that organizations which adopt OSINT-informed security postures can identify exposure and risk factors that reactive approaches consistently miss. Their work is relevant to the present thesis in two respects: it establishes the operational value of OSINT for proactive defense, and it identifies the aggregation of individually non-sensitive public disclosures as a particular source of unrecognized organizational risk. The present study operationalizes this observation by systematically aggregating technology signals from job postings, LinkedIn profiles, and GitHub repositories, each of which represents a routine organizational disclosure that individually appears benign but collectively reveals meaningful intelligence about internal technology stacks.

Kassim et al. [9] review how national computer security incident response teams use public data and free tools in operational practice, finding that OSINT-based approaches have become standard components of organizational threat monitoring workflows. Their review notes, however, that OSINT collection methodologies in operational practice tend to be ad hoc and lack structured integration frameworks, a limitation that the present study addresses through its defined collection pipeline and three-model integration methodology. Rahman [10] further characterizes the opportunities and challenges inherent in OSINT practice, noting that the expanding volume of publicly available information simultaneously increases its intelligence value and the difficulty of systematic analysis. Both studies together underscore the practical relevance of the structured integration methodology developed in this thesis.

2.3 Human-Generated OSINT: Job Postings, LinkedIn, and GitHub

Among the most directly relevant prior work to this thesis is a set of studies examining specific human-generated OSINT sources as intelligence carriers. This literature establishes the theoretical and empirical basis for treating job postings, professional network profiles, and

public code repositories as valid and complementary sources of organizational technology intelligence.

Herranen [6] conducted an analysis of e-skills found in job postings, examining the degree to which job advertisement content reflects the actual technology requirements and stack characteristics of hiring organizations. Herranen's study demonstrates that job postings are among the most explicitly declarative technology signals available in the public domain: organizations directly enumerate the specific frameworks, platforms, and languages they require candidates to know, reflecting the technologies their engineering teams are actively using. This declarative quality distinguishes job postings from other OSINT sources and provides the primary justification for their treatment as high-confidence signals in the present study's integration models.

Duffy et al. [7] examine information reconnaissance through the accumulation of public information data sources, demonstrating that the aggregation of signals from multiple public channels produces a significantly more complete organizational profile than any single source can yield individually. Their work provides empirical support for the multi-source integration approach adopted in this thesis and establishes that sources of individually partial information can be combined to produce intelligence of substantially greater completeness and reliability. Duffy et al. also note that public data aggregation introduces ethical considerations regarding unintended disclosure, a dimension addressed in the scope and ethical boundaries discussion of Chapter 3.

GitHub repository data, accessed through the GitHub API [22], represents a third category of human-generated OSINT that is particularly valuable for technology fingerprinting. Unlike job postings and LinkedIn profiles, which reflect declarative or aspirational technology signals, GitHub repositories reflect implemented technology choices: the dependency files, configuration manifests, CI/CD pipeline definitions, and language compositions of public repositories directly document technologies in active use within development teams. Bazzell [16] discusses the use of public code repositories and professional network profiles within broader OSINT collection frameworks, noting that these sources collectively reveal significant organizational intelligence when examined systematically. The present study operationalizes this observation through a structured GitHub collection pipeline described in Chapter 4.

LinkedIn profile data [23] constitutes the third OSINT source category in this study. Unlike job postings, which are organizational-level disclosures, LinkedIn profiles provide employee-level technology signal data that, when aggregated across a technical workforce, yields a statistical approximation of organizational technology adoption patterns. The self-reported nature of LinkedIn skill data introduces a reliability differential relative to job postings and GitHub repositories, a consideration that is directly incorporated into the evidence-tier integration model by assigning isolated LinkedIn-only detections to a lower evidence tier unless corroborated by another independent source.

2.4 Cyber Threat Intelligence and Threat Hunting

The broader context of cyber threat intelligence and threat hunting research motivates the attack surface orientation of this thesis and situates technology fingerprinting within the operational workflows of modern security teams.

Mahboubi et al. [5] present a systematic review of evolving techniques in cyber threat hunting, tracing the development of threat hunting from reactive incident response toward proactive intelligence-driven operations. Their review identifies technology stack enumeration as a critical precondition for effective threat hunting, arguing that hunters cannot meaningfully search for adversarial activity within an environment whose technology composition they do not fully understand. This finding directly motivates the present study's focus on completeness of technology fingerprinting as a security outcome, rather than treating fingerprinting as a purely technical exercise. Mahboubi et al. also highlight the increasing sophistication of adversaries who deliberately avoid leaving signals on tools' observable surfaces, reinforcing the argument that tool-only reconnaissance provides insufficient coverage for adversarial environments.

Shafee et al. [4] evaluate the use of OSINT-based approaches for cyber threat awareness, examining how publicly available information can be systematically collected and analyzed to produce actionable threat intelligence. Their work is relevant to this thesis in demonstrating that structured OSINT collection pipelines can produce reliable intelligence outputs from heterogeneous public sources, providing methodological precedent for the collection and normalization pipeline developed in Chapter 4.

MITRE ATT&CK [18] provides the standard framework for characterizing adversarial techniques throughout the attack lifecycle. Within this framework, technology reconnaissance is classified under the Reconnaissance tactic, with specific techniques covering active scanning, gathering host information, and searching open technical databases. The present study's focus on technology fingerprinting maps directly to these MITRE ATT&CK reconnaissance techniques, situating OSINT-enhanced fingerprinting as a capability relevant to both red team and blue team workflows. The ENISA Threat Intelligence Sharing Reports [17] further contextualize the operational importance of comprehensive technology awareness, identifying incomplete asset and technology inventories as a recurrent factor in organizational security incidents.

2.5 Multi-Source Data Fusion and Evidence Integration

The integration methodology at the core of this thesis draws on a well-established body of literature in multi-source data fusion and evidence combination. This section reviews the theoretical foundations that underpin the three integration models evaluated in Chapter 5.

Shafer's foundational work, A Mathematical Theory of Evidence [11], establishes the Dempster-Shafer theory of evidence combination, which provides the conceptual inspiration for the evidence-tier integration model developed in this thesis. Dempster-Shafer theory's core principle, that evidence from independent sources should be combined to produce a belief measure proportional to the degree of cross-source corroboration, directly informs the

classification logic of Method 3. However, rather than implementing the full mathematical machinery of Dempster-Shafer combination, the present study operationalizes this principle through an explicit rule-based evidence classification scheme. Technologies corroborated by multiple independent source types are assigned to higher evidence tiers, and technologies detected by only a single low-confidence source are assigned to lower tiers. This approach retains the essential multi-source corroboration logic of evidence fusion theory while producing classifications that are fully transparent, rule-based, and free of manually assigned numeric parameters.

Hall and Llinas [12] provide a comprehensive introduction to multisensor data fusion that extends evidence fusion theory to practical multi-source intelligence integration systems. Their framework addresses the challenge of combining signals from sources with heterogeneous reliability profiles. This principle informs the present study's evidence-tier model by supporting the idea that corroboration across independent source types should be treated as stronger evidence than isolated single-source detection. However, the present study applies this principle through transparent rule-based tiers rather than through numeric source weights.

Manning et al. [14] provide the conceptual foundation for the union-based integration model, the simplest of the three approaches evaluated in this thesis. Their treatment of set operations in information retrieval establishes the theoretical properties of union-based aggregation, including its guarantee of maximum recall at the cost of potentially reduced precision. This theoretical framing informs the interpretation of the integration results presented in Chapter 5. Robertson and Zaragoza [13] extend this foundation to weighted relevance ranking through the probabilistic relevance framework underlying BM25, providing the conceptual basis for the source-strength weighted integration model's treatment of empirically derived source strength as a ranking signal. Mitchell [15] provides further conceptual support for data fusion fundamentals, grounding the integration pipeline's design in established data fusion principles.

2.6 Research Gap and Positioning

The literature reviewed in this chapter collectively establishes four findings that directly motivate and position the present research. First, active scanning tools provide reliable coverage of perimeter-facing technologies but share a structural inability to detect technologies operating behind access boundaries or producing no externally observable signals [1, 3]. Second, human-generated OSINT sources, including job postings, professional profiles, and public repositories, constitute legitimate and largely untapped sources of organizational technology intelligence that are structurally complementary to the signals captured by active scanning [6, 7, 8]. Third, multi-source data fusion techniques developed in the information retrieval and sensor fusion literature provide principled methodological foundations for combining heterogeneous evidence sources into integrated outputs [11, 12, 14]. Fourth, no prior study has empirically quantified the technology coverage gap between tool-only reconnaissance and human-generated OSINT across a diverse organizational sample, nor has any prior work proposed and comparatively evaluated multiple integration methodologies for closing that gap [2, 9].

The present thesis addresses this gap directly. It constructs a controlled empirical study across twelve organizations, measures the coverage characteristics of tool-only and OSINT-only fingerprinting, and evaluates three integration models of increasing sophistication against reference technology inventories. In doing so, it provides both an empirical quantification of the coverage gap that prior literature has theorized but not measured, and a comparative evaluation of integration methodologies that prior work has provided theoretical tools for but not applied to this specific domain.

Chapter 3 Research Methodology

This chapter describes the complete methodology used to collect, process, and analyze data in support of the two research questions. It covers the research design, the target organization selection process, the data collection protocols for each source type, the signal normalization pipeline, the reference inventory construction process, the source profile computation, the design of the three integration models, and the evaluation metrics applied to compare their outputs.

3.1 Research Design and Approach

This study adopts a quantitative, empirical research design grounded in the exploratory comparison of two signal classes: active scanning tool outputs and human-generated OSINT. The design is exploratory rather than confirmatory in the statistical sense. It does not seek to make population-level inference claims but rather to demonstrate, through systematic measurement across a diverse organizational sample, whether the coverage gap between tool-based and OSINT-based technology fingerprinting is real, measurable, and consistent, and whether structured integration methodologies can reliably close that gap.

The research follows a sequential pipeline structure consisting of four phases: data collection, normalization, integration, and evaluation. Each phase produces outputs that feed directly into the next, ensuring that the final evaluation results are fully traceable to the raw collected signals. All data collection was conducted through publicly accessible channels using non-intrusive methods. No intrusive testing, exploitation, authentication bypass, or access-controlled probing was performed at any point in the study.

The study is positioned as an empirical systems evaluation. This design is appropriate for research questions that ask whether a proposed technical approach produces measurably different and improved outputs relative to an existing baseline, and is consistent with established practice in applied cybersecurity research [1].

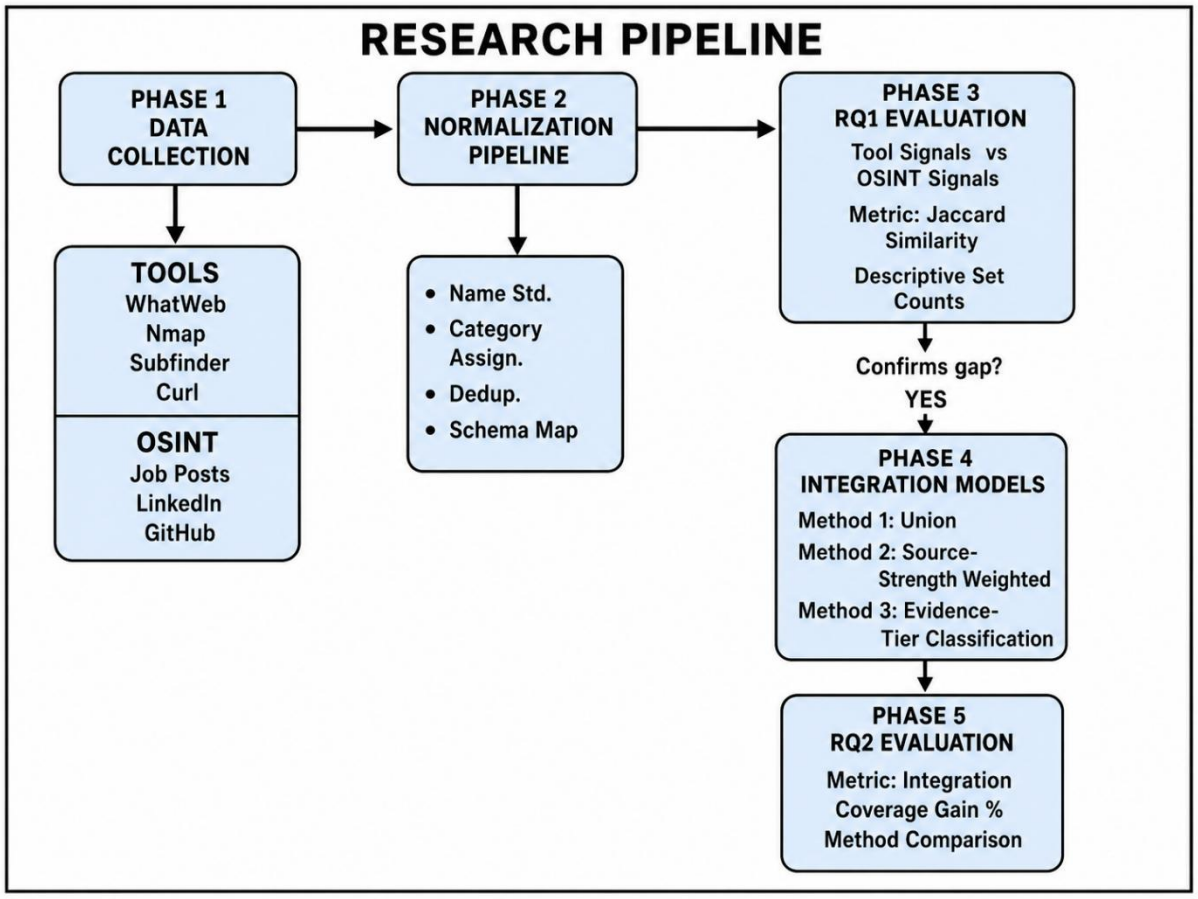

Figure 3.1 Research pipeline showing sequential execution of RQ1 evaluation before integration, confirming the coverage gap that motivates RQ2.

3.2 Target Organization Selection

Twelve organizations were selected as the target sample for this study using a purposive sampling strategy. Purposive sampling was chosen over random sampling because the research questions require diversity across several analytically relevant dimensions: organizational size, industry vertical, expected OSINT visibility, and technological profile. A randomly drawn sample from the population of all organizations would not guarantee adequate representation across these dimensions within a sample of twelve, whereas purposive selection allows the researcher to deliberately construct a sample that maximizes analytical diversity [5].

Each selected organization had sufficient publicly observable presence across the selected source categories to support cross-source comparison. The twelve organizations span financial technology, developer infrastructure, e-commerce, design tooling, healthcare technology, and enterprise software verticals, and range from growth-stage technology companies with extensive public engineering presence to established enterprise organizations with more limited public OSINT exposure.

Table 3.1 Twelve target organizations selected using purposive sampling, with industry vertical, size category, and expected OSINT visibility profile. Expected OSINT visibility was estimated before data collection based on publicly observable engineering content, job posting availability, public repository activity, and organizational web presence.

Organization	Industry Vertical	Size Category	Expected OSINT Visibility
Stripe	Financial Technology	Large	High
Shopify	E-Commerce	Large	High
Datadog	Developer Infrastructure	Large	High
Sentry	Developer Infrastructure	Mid-size	High
Coinbase	Financial Technology	Large	High
Figma	Design Tooling	Mid-size	High
Instacart	E-Commerce / Logistics	Large	Moderate
Brex	Financial Technology	Mid-size	High
Scale AI	Artificial Intelligence	Mid-size	Moderate
Asana	Enterprise Software	Mid-size	Moderate
Epic Systems	Healthcare Technology	Large	Low
One Medical	Healthcare Technology	Mid-size	Low

3.3 Data Collection

Data collection proceeded across two parallel tracks: active scanning tool execution and human-generated OSINT collection. Both tracks targeted the same twelve organizations and produced outputs normalized into a common signal schema.

3.3.1 Active Scanning Tool Collection

Active scanning was performed using four tools: WhatWeb [20], Nmap [19], Subfinder [26], and curl header inspection. The tool suite was selected based on three criteria: academic precedent and citation frequency in the reconnaissance literature, availability as open-source or freely accessible tools to ensure full methodological reproducibility, and complementary coverage across web application, network service, and subdomain layers. Commercial platforms such as Shodan and BuiltWith were excluded to maintain methodological reproducibility and to avoid dependency on access-tier constraints that would limit replication by other researchers within the scope of this study.

WhatWeb was applied to the primary and secondary web domains of each target organization to detect web server identities, application frameworks, content management systems, and front-end technologies from HTTP response headers and page content signals. Nmap was used for network port scanning and service version detection, targeting publicly accessible endpoints associated with each organization using a light, non-intrusive scan

configuration on standard web ports. Subfinder was used to enumerate publicly registered subdomains associated with each organization's primary domain, expanding the set of observable endpoints and identifying infrastructure-level technology signals embedded in subdomain naming patterns. Curl header inspection was applied to capture raw HTTP response headers, providing a lightweight complement to WhatWeb by extracting server software, framework hints, and security header signals directly from HTTP responses.

All active scanning was performed against publicly accessible web properties and network endpoints associated with each target organization using non-intrusive scanning settings and publicly available indexed data where applicable. The outputs of all four tools were collected, parsed into a structured format, and stored as individual source files prior to normalization.

3.3.2 Job Posting Collection

Job posting data was collected from publicly accessible employment platforms and the careers pages of each target organization. Collection was scoped to postings for engineering and technical roles, including software engineering, infrastructure, data engineering, security engineering, and DevOps positions. Non-technical postings were excluded, as they do not reliably contain technology stack signals.

For each organization, a structured set of job postings was collected and parsed for technology signals. Parsing targeted three content areas within each posting: explicitly listed required or preferred technical skills, named platforms and services mentioned in role descriptions, and tool or framework names appearing in responsibilities and qualifications sections. The self-reported and declarative nature of job posting content, in which organizations explicitly state the technologies their teams use and require candidates to know, makes this source type particularly direct as a technology disclosure mechanism [6]. Collection was supported through the Greenhouse and Lever public job board APIs where available, and through direct careers page collection for organizations not using these platforms.

3.3.3 LinkedIn Profile Collection

LinkedIn profile data was collected through structured manual sampling of technical employees at each target organization. Profiles were identified through LinkedIn's public search functionality using organization name filters and role-type keywords. For each organization, a set of employee profiles in technical roles was examined, and technology signals were extracted from the skills sections, role descriptions, and listed project and tool mentions of each profile.

LinkedIn data is structurally distinct from job posting data in that it reflects individual employee self-reports rather than organizational-level declarations. Skill endorsements on LinkedIn are self-selected and not systematically verified, which introduces a reliability differential relative to job postings. This characteristic is directly reflected in the source profile computed for LinkedIn in the preprocessing stage, which shows a lower agreement rate with other sources than job postings, and is incorporated into the evidence-tier classification by assigning isolated LinkedIn-only detections to a lower evidence tier unless corroborated by another independent source [23].

3.3.4 GitHub Repository Collection

GitHub repository data was collected using the GitHub API [22]. For each organization, the API was queried to retrieve the set of public repositories associated with the organization's GitHub account. From each repository, technology signals were extracted from four artifact types: dependency manifest files including package.json, requirements.txt, Gemfile, and pom.xml; infrastructure configuration files including Dockerfile and docker-compose.yml; CI/CD pipeline definition files including GitHub Actions workflow files and CircleCI configuration files; and README documentation containing explicit tool, framework, or platform mentions.

GitHub repository data represents the most implementation-level signal type among the three OSINT sources. Unlike job postings, which reflect declarative intent, or LinkedIn profiles, which reflect individual self-reports, GitHub dependency files and configuration manifests directly document technologies in active use within development workflows. This characteristic is reflected in GitHub's source profile, which shows the highest total signal count of any source type in the study at 850 signals across the full dataset, and a category coverage spanning multiple distinct technology domains. The lower agreement rate of GitHub signals relative to job postings reflects the unique implementation-level visibility of this source rather than unreliability, and is interpreted accordingly in the integration model design.

3.4 Signal Normalization Pipeline

Raw signals collected from all four tools and three OSINT sources were processed through a common normalization pipeline prior to integration. The normalization pipeline performs four operations: name standardization, category assignment, deduplication, and schema mapping. This normalization step ensures that outputs from heterogeneous sources can be compared and integrated consistently across organizations.

Name standardization resolves surface-level variation in technology references across sources. The same technology may appear as "React," "ReactJS," and "React.js" across different sources; normalization maps all variants to a single canonical name. Version-specific references such as "Python 3.9" are normalized to the base technology name "Python" to enable cross-source comparison. A standardized technology name dictionary was constructed from the collected dataset and applied consistently across all sources and organizations.

Category assignment maps each normalized technology name to one of thirteen technology categories. These categories were defined to cover the full range of technology signals observed in the dataset while maintaining sufficient granularity to support category-level analysis of coverage differences between source types. Technologies that could not be confidently assigned to any of the thirteen categories were placed in a residual category for reporting purposes.

Table 3.2 Thirteen technology categories used for signal normalization. All signals collected from tools and OSINT sources are mapped to one of these categories prior to integration.

Category	Description	Example Technologies
Language	Programming languages	Python, Go, TypeScript, Java
Framework	Application and web frameworks	React, Django, Spring Boot, FastAPI
Database	Relational and non-relational data stores	PostgreSQL, MongoDB, Redis, DynamoDB
Cloud	Cloud platforms and managed services	AWS, GCP, Azure, Kubernetes
CI/CD	Continuous integration and deployment tools	GitHub Actions, CircleCI, ArgoCD, Jenkins
DevOps	Infrastructure automation and orchestration	Terraform, Docker, Ansible, Helm
Infrastructure	Network and server infrastructure	Nginx, HAProxy, Cloudflare
Security	Security tooling and platforms	Snyk, Vault, SonarQube, OWASP ZAP
Data and Analytics	Data processing and analytics platforms	Spark, Snowflake, Kafka, Airflow
API and Integration	API gateways and integration services	GraphQL, gRPC, Stripe API, Twilio
Monitoring and Observability	System monitoring and alerting tools	Datadog, Prometheus, Grafana, Sentry
Identity and Authentication	Identity management and auth services	Okta, Auth0, LDAP, SAML
Other	Signals not fitting named categories	Varies

Deduplication removes redundant signals within each source-organization combination, ensuring that a technology mentioned multiple times within a single source contributes only one signal to the integration input. Deduplication is applied at the source level

prior to cross-source comparison, so that a technology mentioned independently by both job postings and GitHub contributes one signal from each source rather than being collapsed into a single signal.

Schema mapping produces a unified master signals file for each organization containing all normalized, categorized, and deduplicated signals from all sources, with each row recording the technology name, category, source type, and organization identifier. This file serves as the common input to all three integration models.

3.5 Reference Inventory Computation

A reference technology inventory was constructed for each of the twelve target organizations to serve as the basis for evaluating integration model outputs. The reference inventory is defined as a consolidated set of technology names that are plausibly attributable to each organization based on the totality of publicly available evidence collected across all source types.

The reference inventory is not an internal company-verified ground truth. It is a publicly consolidated evidence inventory constructed from the union of all signals that appear in at least one source with sufficient contextual specificity to be treated as a credible organizational technology disclosure. Signals that appeared only in tool outputs and that reflected infrastructure metadata rather than application-layer technology choices were reviewed for inclusion based on whether they represented meaningful technology fingerprint information. Signals appearing exclusively in LinkedIn profiles without corroboration from any other source were retained in the reference inventory for completeness but classified as weak evidence unless corroborated by another independent source.

This construction approach is consistent with the study's framing as an exploratory empirical investigation using publicly accessible data. The reference inventory enables systematic comparison of what each source type and integration model detects relative to the full set of publicly discernible technologies, which is the appropriate comparison basis for answering RQ1 and RQ2 as stated.

3.6 Approach to Answering RQ1

RQ1 asks: to what extent do human-generated OSINT sources reveal technology stack information about target organizations that traditional active scanning tools fail to identify?

This question is answered through a direct set-level comparison between the tool-detected technology inventory and the OSINT-detected technology inventory for each of the twelve target organizations. Following the normalization phase described in Section 3.4, the master signals file for each organization is split into two distinct sets: the tool set, comprising all technologies detected exclusively through active scanning, and the OSINT set, comprising all technologies detected through job postings, LinkedIn profiles, and GitHub repositories. These two sets are then compared across four measurement dimensions.

The first dimension is Jaccard similarity, which measures the degree of overlap between the tool set and the OSINT set. A value approaching zero confirms that the two signal classes

are detecting fundamentally different layers of the technology stack rather than providing redundant coverage of the same technologies.

The second dimension is OSINT coverage gain percentage, which measures the proportion of the combined technology fingerprint that is contributed exclusively by OSINT sources and is not detectable through tool-based reconnaissance alone. This directly quantifies the magnitude of the coverage gap that motivates the study.

The third dimension is descriptive set counts, which report for each organization the number of tool-only detections, OSINT-only detections, overlapping detections, and total reference inventory size. These counts provide the concrete empirical foundation underlying the derived ratios and allow the results to be verified and interpreted without dependence on any single metric.

The fourth dimension is category-level breakdown, which identifies which of the thirteen technology categories are dominated by OSINT-only signals versus tool-only signals. This analysis answers not just how much OSINT reveals beyond tools, but what specifically it reveals, directly connecting the coverage gap finding to the structural argument that tools are limited to perimeter-visible signals while OSINT reaches application-layer and internal technology signals.

Together these four dimensions provide a complete and multi-faceted answer to RQ1 that is robust to the limitations of any single metric.

3.7 Source Profile Computation

Prior to integration, a source profile was computed for each source type from the full dataset across all twelve organizations. The source profile characterizes each source's observed behavior across three metrics: agreement rate, category coverage, and source strength. These profiles were computed as descriptive statistics of the collected dataset in a preprocessing phase prior to integration, following established practice in data-driven parameter calibration where dataset structure informs integration parameters rather than having them assigned by expert judgment [14].

Agreement rate is defined as the proportion of signals produced by a given source that are also detected by at least one other independent source across the same organization. A high agreement rate indicates that a source tends to detect technologies corroborated by other sources, reflecting declarative and widely visible technologies. A low agreement rate indicates that a source tends to detect technologies not visible to other sources, reflecting either unique implementation-level visibility or lower-reliability self-reported signals depending on the source type.

Category coverage is defined as the proportion of the thirteen technology categories in which a given source produces at least one signal across the full dataset. A high category coverage indicates broad detection across multiple technology domains, while a lower value indicates specialization in a subset of domains.

Source strength is computed as the arithmetic mean of agreement rate and category coverage. Equal weighting was applied to avoid manually privileging either corroboration or breadth, as both dimensions contribute independently and meaningfully to a source's overall utility in multi-source integration. This formulation requires no manually chosen parameters beyond the two empirically measured components. Source strength is used only in Method 2 as a source-strength weighting factor. It is not used in the Method 3 evidence-tier classification, which is entirely rule-based and contains no numeric scoring.

The formula is:

Source Strength = (Agreement Rate + Category Coverage) / 2

The computed source profiles for the four source types are presented in Table 3.3.

Table 3.3 Source profiles computed empirically from the full dataset across all twelve organizations. Source strength is the arithmetic mean of agreement rate and category coverage. All values are derived from the data and no values were manually assigned.

Source	Total Signals	Agreement Rate	Category Coverage	Source Strength
Job Postings	203	0.7734	0.52	0.6467
LinkedIn	414	0.3720	0.32	0.3460
GitHub	850	0.1953	0.40	0.2977
Tools	110	0.0636	0.44	0.2518

The source profiles reveal several analytically significant patterns. Job postings exhibit the highest agreement rate at 0.7734, reflecting their declarative nature as explicit organizational technology disclosures. GitHub exhibits the highest total signal count at 850 but the lowest agreement rate among OSINT sources at 0.1953, reflecting its unique implementation-level visibility into technologies that no other source can observe. Tools exhibit the lowest agreement rate overall at 0.0636, confirming that the technologies detected by active scanning are largely distinct from those disclosed through human-generated OSINT. This pattern directly anticipates the near-zero Jaccard similarity finding reported in Chapter 5.

3.8 Integration Model Design

Three integration models were designed and implemented to combine tool signals and OSINT signals into unified technology inventories. The models are ordered by increasing analytical sophistication. A side-by-side comparison of all three models is provided in Table 3.4.

3.8.1 Method 1: Union-Based Integration

Method 1 aggregates all normalized signals from all source types without any weighting or filtering. Every technology detected by any source for a given organization is included in the integrated fingerprint, regardless of how many sources detected it or with what evidence strength. This model treats all signals as equally valid contributions to the integrated output.

The theoretical basis for union-based integration is drawn from set theory and information retrieval [14], where the union operation guarantees maximum recall by including all available evidence. Method 1 serves as a high-recall baseline in this study: it establishes the maximum number of technologies that could potentially be attributed to each organization from the totality of publicly available signals, and provides the comparison point against which more selective integration methods are evaluated. Its primary limitation is that it treats a single LinkedIn self-report with the same weight as a confirmed GitHub dependency, producing fingerprints that include weakly supported signals alongside well-evidenced detections.

3.8.2 Method 2: Source-Strength Weighted Aggregation

Method 2 ranks technologies within each organization's integrated fingerprint according to a cumulative source-strength score computed from the empirically derived source strength values presented in Table 3.3. Each signal in the master signals file carries the source strength value of its source type as its weight. For technologies detected by multiple sources, the source-strength scores of all detecting sources are summed to produce a cumulative score. Technologies are then ranked by their cumulative score, with higher-ranked technologies representing more strongly supported detections.

The theoretical basis for source-strength weighted aggregation draws on probabilistic relevance frameworks in information retrieval [13], which treat cumulative evidence weight as a ranking signal. Method 2 addresses the primary limitation of Method 1 by differentiating between technologies supported by high-strength, multi-source evidence and those appearing in only one low-strength source. Because source strength values are derived empirically from the dataset rather than assigned manually, the ranking produced by Method 2 reflects the observed agreement and coverage characteristics of each source type in the study dataset.

3.8.3 Method 3: Evidence-Tier Classification

Method 3 classifies each detected technology into one of four evidence tiers based on rule-based criteria that reflect the number and type of independent sources corroborating each detection. No manually assigned numeric scores, bonus values, or threshold parameters are used anywhere in the tier classification. Every classification decision is determined by observable, rule-based logic applied to the source composition of each detected technology, ensuring full transparency and reproducibility [11, 12].

The four tiers and their classification rules are as follows:

A technology is classified as **Cross-Validated** if it is detected by at least one tool and at least one OSINT source independently. This tier represents the highest evidence level, as it requires corroboration across the two structurally distinct signal classes that are the subject of this study.

A technology is classified as **Strong OSINT Evidence** if it is detected by two or more independent OSINT sources, meaning any combination of job postings, LinkedIn, and GitHub, without tool corroboration. This tier captures technologies that are well-evidenced within the OSINT layer but not externally visible to scanning tools, consistent with the expectation that application-layer and internal technologies will appear in this tier.

A technology is classified as **Moderate Evidence** if it is detected by tools only, or by a single source with strong contextual evidence. Strong contextual evidence refers to signals extracted from implementation-level artifacts such as dependency manifests, configuration files, CI/CD workflow files, or explicit technical requirements in job postings, rather than isolated keyword mentions.

A technology is classified as **Weak Evidence** if it is detected only by LinkedIn without corroboration from any other source. This tier reflects the lowest confidence classification, consistent with the self-reported and unverified nature of LinkedIn skill data as indicated by its source profile.

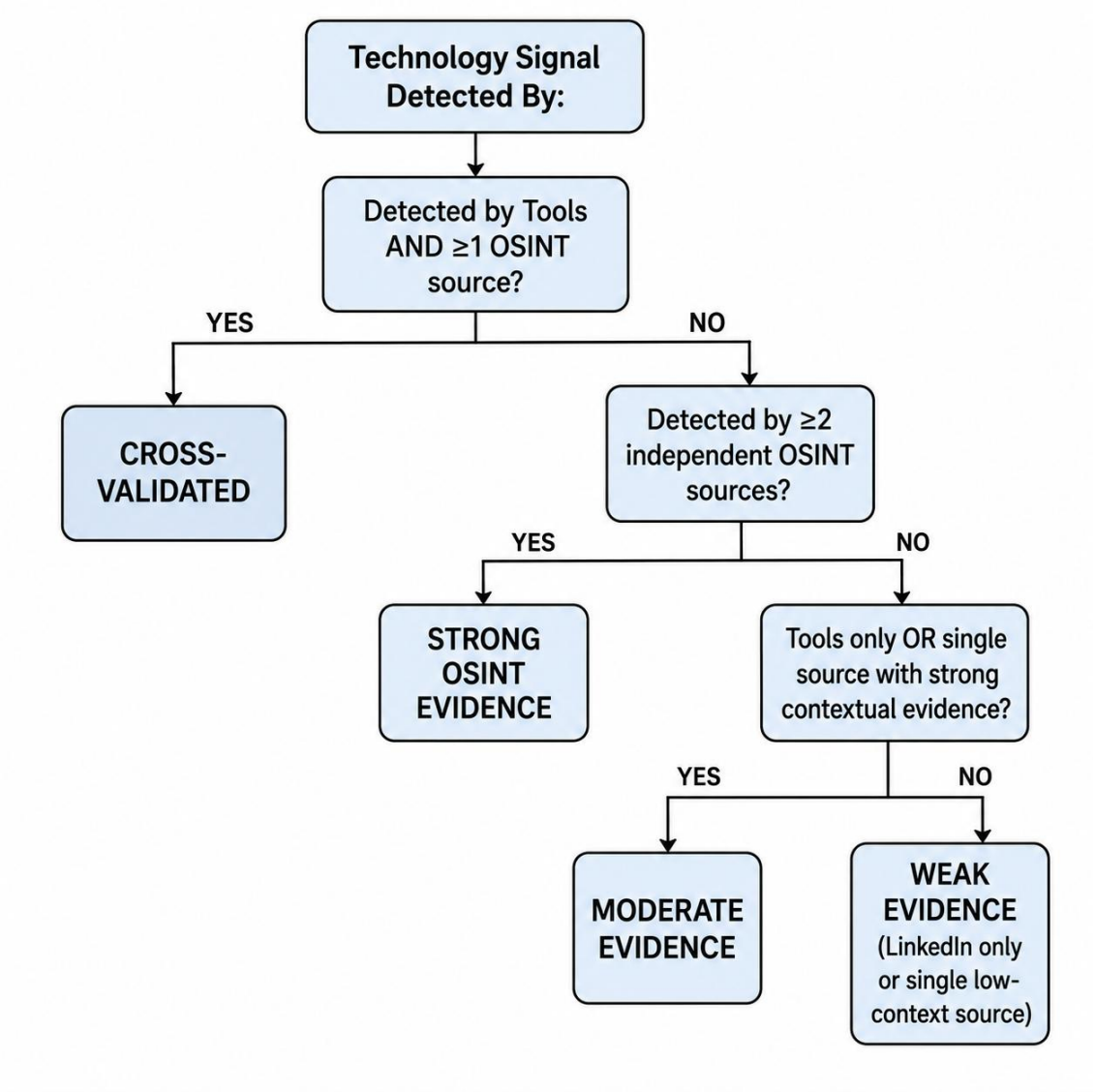

Figure 3.2 Rule-based decision flowchart for Method 3 evidence-tier classification. No numeric thresholds or manually assigned scores are used at any decision point.

Table 3.4 Side-by-side comparison of the three integration methods across key design dimensions.

Dimension	Method 1: Union-Based	Method 2: Source-Strength Weighted	Method 3: Evidence-Tier
Logic	Aggregate all signals equally	Weight signals by empirical source strength	Classify by rule-based multi-source corroboration
Numeric scores	None	Empirically derived source-strength scores	None
Output type	Flat technology list	Ranked technology list	Tiered technology classification
Primary strength	Maximum recall	Empirically ranked output	Interpretable evidence quality
Primary limitation	No signal quality differentiation	Sensitive to mention count inflation	No numeric ranking within tiers

Dimension	Method 1: Union-Based	Method 2: Source-Strength Weighted	Method 3: Evidence-Tier
Theoretical basis	Set union, information retrieval [14]	Probabilistic relevance weighting [13]	Evidence fusion and corroboration logic [11, 12]
Answers RQ2 by	Showing integration maximizes coverage	Showing empirical weighting improves reliability	Showing rule-based tiers provide interpretable output

3.9 Evaluation Metrics

The evaluation uses a combination of set-based comparison metrics and method-specific output measures. For RQ1, Jaccard similarity is used to measure the overlap between tool-detected and OSINT-detected technology sets, while OSINT coverage gain percentage is used to measure the proportion of the combined technology inventory contributed exclusively by OSINT sources. Descriptive set counts are also reported for each organization, including tool-only detections, OSINT-only detections, overlapping detections, and total reference inventory size. For RQ2, each integration method is evaluated according to its own output structure: Method 1 is evaluated through union-based coverage gain, Method 2 through source-strength weighted rankings and multi-source technology counts, and Method 3 through evidence-tier distributions. Together, these metrics provide the empirical basis for answering both research questions. Traditional accuracy metrics such as precision, recall, and F1-score were not applied because they require a verified ground truth technology inventory for each organization, which was not available in this study. The evaluation therefore relies on coverage and completeness metrics applied to a publicly consolidated reference inventory, which is the appropriate design given the data constraints of an OSINT-based study.

The two primary metrics are formally defined as follows. Let **T** denote the set of technologies detected by active scanning tools and **O** denote the set of technologies detected by human-generated OSINT sources for a given organization. Let **I** denote the integrated technology set produced by a given integration method.

$$\text{Jaccard Similarity}\ (\mathrm{T}, 0) = \frac{|\mathrm{T} \cap \mathrm{O}|}{|\mathrm{T} \cup \mathrm{O}|} \tag{1}$$

$$\text{Integration Coverage Gain}\ (\%) = \frac{|\mathrm{I} - \mathrm{T}|}{|\mathrm{I}|} \times 100 \tag{2}$$

In Equation (1), $| T \cap O |$ represents the number of technologies detected by both tool-based reconnaissance and OSINT, while $| T \cup O |$ represents the total number of unique technologies detected across both signal classes. In Equation (2), $| I - T |$ represents the number of technologies in the integrated fingerprint that were not detected by the tool-only baseline, while $| I |$ represents the total number of unique technologies in the integrated

fingerprint. A Jaccard similarity approaching zero indicates minimal overlap between the two signal classes. A coverage gain approaching 100% indicates that the integrated fingerprint is composed largely of technologies not visible through tool-based reconnaissance alone.

Jaccard similarity is defined as the size of the intersection of two sets divided by the size of their union. In this study, it is used to measure the overlap between tool-detected and OSINT-detected technology sets for each organization, directly answering RQ1 by quantifying how redundant or complementary the two signal classes are. A Jaccard similarity approaching zero indicates that the two sets are almost entirely non-overlapping, confirming the complementary nature of the two source types. Jaccard similarity is reported alongside descriptive set counts, including tool-only detections, OSINT-only detections, overlapping detections, and total inventory size, to provide the empirical foundation for interpreting OSINT coverage gain results.

Integration coverage gain percentage is defined as the number of technologies in the integrated fingerprint that were not detected by tools alone, divided by the total number of technologies in the integrated fingerprint, expressed as a percentage. This formulation uses the integrated total as the denominator rather than the tool-only count, avoiding denominator inflation effects that would arise when tool-only counts are very small relative to OSINT-only counts, as is consistently observed in this dataset.

3.10 Ethical Considerations

All data collected in this study was obtained from publicly accessible sources without authentication bypass, unauthorized system access, or any form of deception. Active scanning was performed only against publicly accessible endpoints using non-intrusive, rate-limited scanning configurations. OSINT collection drew exclusively from information voluntarily published by organizations and their employees through public channels.

The study does not publish individual employee data or personal identifying information. LinkedIn profile data was used only in aggregated form to derive organizational-level technology signals, and no individual profile data is reported in the results. The findings of this study are reported at the organizational level only, and no information is disclosed that would increase the security risk to any of the studied organizations beyond what is already publicly observable by any internet user [7, 8].

The research follows established norms for ethical OSINT research as documented in the cybersecurity research community, treating public data aggregation as a methodology for security improvement rather than as a vehicle for harm [16].

Chapter 4 System Design and Research Implementation

This chapter describes the design and implementation of the research pipeline used to operationalize the methodology presented in Chapter 3. The system was developed to collect, normalize, integrate, and evaluate technology signals from both tool-based reconnaissance and human-generated OSINT sources. Rather than serving as a standalone software product, the implementation functions as a research instrument for measuring the contribution of human-generated OSINT to technology fingerprinting, directly operationalizing the data collection, integration, and evaluation protocols specified in Chapter 3 in service of RQ1 and RQ2.

The pipeline is organized into six functional modules: a data collection layer, a data storage layer, a normalization module, a source profile computation module, an integration engine, and an evaluation and comparison module. Each module is described in terms of its research purpose, its inputs and outputs, and its role within the overall pipeline.

4.1 System Overview

The research pipeline implements the four-phase design described in Chapter 3 as a modular, sequential system in which the output of each module serves as the input to the next. The modular design was adopted to ensure that each research operation, collection, normalization, integration, and evaluation, could be executed, verified, and reproduced independently. This property is essential for a research instrument, as it allows the validity of each phase to be assessed separately and ensures that errors in one phase do not propagate silently through subsequent phases.

The pipeline operates on a per-organization basis across all twelve target organizations simultaneously, producing both organization-level outputs for detailed analysis and aggregate outputs for cross-organizational comparison. All intermediate and final outputs are stored as structured comma-separated value files to ensure accessibility, transparency, and reproducibility without dependency on proprietary data formats or database infrastructure.

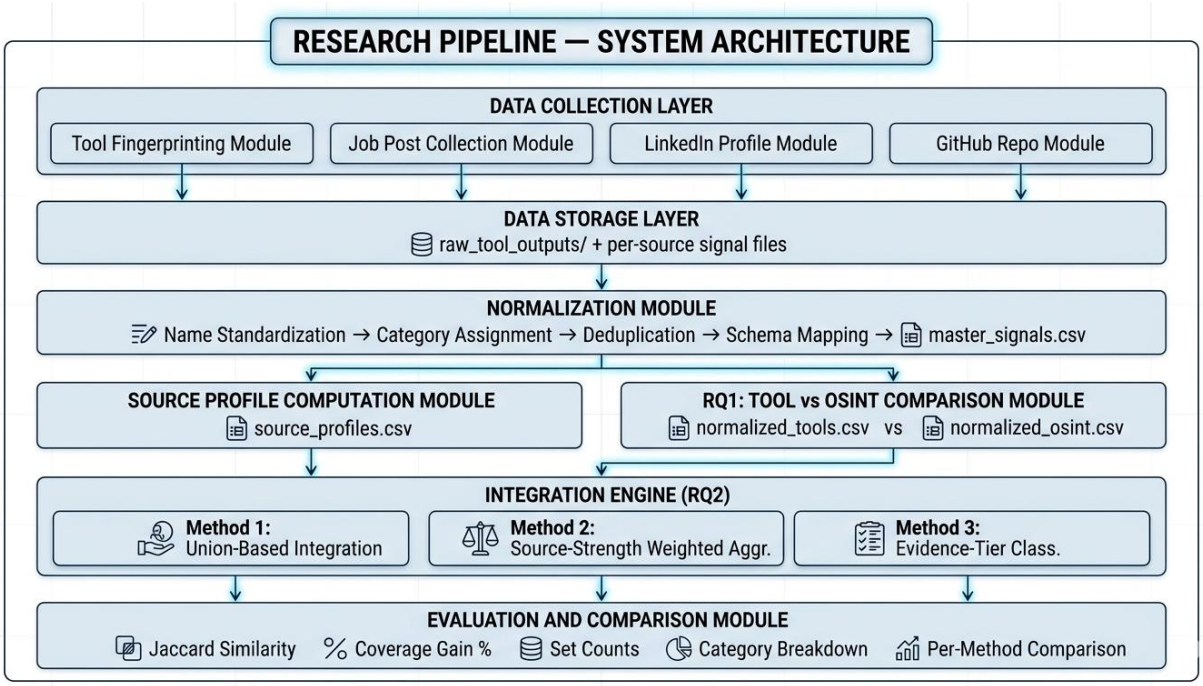

Figure 4.1 System architecture of the research pipeline. Each module operationalizes a corresponding phase of the research methodology described in Chapter 3. RQ1 is addressed through the Tool vs OSINT Comparison Module and RQ2 through the Integration Engine.

4.2 Data Collection Layer

The data collection layer operationalizes the collection protocols described in Section 3.3 by implementing four distinct collection modules, one for each signal source type. Each module was designed to produce structured technology signal outputs in a format compatible with the downstream normalization module, ensuring that signals from structurally heterogeneous sources could be processed through a common pipeline.

4.2.1 Tool Fingerprinting Module

The Tool Fingerprinting Module operationalizes the active scanning protocol described in Section 3.3.1 by orchestrating four scanning instruments against each target organization's publicly accessible endpoints: WhatWeb, Nmap, Subfinder, and curl header inspection. Each instrument was selected to contribute a distinct and complementary layer of technology signal coverage, collectively producing a standardized tool-based baseline fingerprint for each organization.

WhatWeb was applied to each organization's primary web domain to extract web server identities, application framework signatures, content management system indicators, and front-end technology patterns from HTTP response headers and page content. Nmap was used to perform light service version detection against standard web ports, producing network-layer technology signals including server software and protocol version information. Subfinder was applied to enumerate publicly registered subdomains, from which infrastructure-level technology signals were extracted by identifying technology-associated keywords embedded in subdomain naming patterns. Curl header inspection was applied as a lightweight complement to WhatWeb, extracting raw HTTP response header fields including server software declarations, framework hints, and security configuration signals.

The module was implemented to store raw outputs from each scanning instrument as structured JSON files organized by organization and tool, enabling full traceability from final normalized signals back to raw scanner output. These raw outputs were subsequently parsed and structured into a unified tool signal file serving as the tool-track input to the normalization module. The implementation was carried out through the tool fingerprinting pipeline script (tool_fingerprinting_script.py), which coordinates instrument execution, output parsing, and signal structuring across all target organizations.

4.2.2 Job Posting Collection Module

The Job Posting Collection Module operationalizes the job posting collection protocol described in Section 3.3.2 by extracting technology signals from publicly accessible job advertisements associated with each target organization. The module targeted engineering and technical role postings exclusively, as these postings contain direct declarative technology disclosures reflecting the organization's active technology stack.

Technology signal extraction was performed through pattern matching against a structured technology dictionary organized by category, covering programming languages, frameworks, cloud platforms, CI/CD tools, databases, and security technologies. The module supported collection from two structured job board APIs, the Greenhouse Boards API and the Lever API, which provide machine-readable job posting content without requiring HTML parsing, as well as from direct careers page collection through a configurable HTML scraping component for organizations not using these platforms. This multi-source collection strategy ensured comprehensive coverage of publicly posted technical roles across organizations with different recruitment infrastructure. The implementation was carried out through the job posting collection script (job_post_scraper.py).

4.2.3 LinkedIn Employee Profile Collection Module

The LinkedIn Employee Profile Collection Module operationalizes the LinkedIn collection protocol described in Section 3.3.3 through a two-phase approach designed to maintain full compliance with LinkedIn's terms of service and established ethical standards for human-subject data handling.

In the first phase, LinkedIn profiles of technical employees associated with each target organization were manually identified through LinkedIn's public search functionality and exported in PDF format using LinkedIn's native profile export feature. No automated access to LinkedIn's platform, API, or web interface was employed at any stage of this phase. Profile exports were organized per organization into structured local directories prior to processing.

In the second phase, a PDF parsing and signal extraction script (linkedin_employee_scraper.py) was applied to the locally stored profile PDFs. The script extracted organization-specific experience text from each profile by identifying and isolating blocks of text associated with the target organization's name and aliases. Technology signals were then detected through pattern matching against a structured technology dictionary spanning twenty-two technology categories. These extraction categories were subsequently mapped into the thirteen-category taxonomy used by the normalization module, ensuring consistency with the signal schema applied across all source types. During extraction, contextual labels were recorded to distinguish action-based technology mentions from isolated

keyword mentions. These labels were used only to support evidence interpretation and were not used as numeric confidence scores in the integration models. All employee identifiers were anonymized to sequential identifiers in the output, ensuring that no personally identifiable information was retained in the research dataset.

This two-phase approach balances the research need for employee-level technology signal data with the ethical requirement to avoid automated platform access and to protect individual privacy.

4.2.4 GitHub Repository Collection Module

The GitHub Repository Collection Module operationalizes the GitHub collection protocol described in Section 3.3.4 by extracting technology signals from public repositories associated with each target organization through the GitHub API [22]. The module was designed to capture implementation-level technology signals that are structurally inaccessible to active scanning tools, including programming language compositions, software dependencies, infrastructure definitions, and CI/CD pipeline configurations.

Signal extraction targeted four artifact types within each repository: dependency manifest files, infrastructure configuration files, CI/CD pipeline definition files, and README documentation. This multi-artifact extraction strategy was adopted because different technology layers within an organization's stack tend to be disclosed through different artifact types. For example, application framework choices tend to appear in dependency manifests, while infrastructure and deployment tooling tends to appear in configuration and CI/CD files. The implementation was carried out through the GitHub repository collection script (github_scraper.py).

4.3 Data Storage and Intermediate Outputs

The data storage layer organizes all intermediate and final research outputs within a structured directory hierarchy that reflects the sequential phases of the pipeline. This organization was designed to ensure that the provenance of every signal in the final dataset is traceable to its source collection module, supporting both research reproducibility and audit of the normalization and integration processes.

Raw outputs from the Tool Fingerprinting Module are stored as per-organization, per-instrument JSON files within a dedicated raw output directory. These files preserve the complete, unprocessed output of each scanning instrument and serve as the archival record of the collection phase. All subsequent signal files are derived from these raw outputs through the normalization pipeline.

Processed signal files are organized within a primary output directory containing the master signals file and all integration and evaluation outputs. Integration outputs are further organized into method-specific subdirectories, each containing the full results and company-level summary files for the corresponding integration method. Evaluation outputs are stored in a dedicated comparison subdirectory containing all RQ1 analysis files.

Table 4.1Complete summary of research pipeline modules with their corresponding implementation scripts, primary inputs, and primary outputs. Each module operationalizes a specific phase of the methodology described in Chapter 3.

Module	Implementation Script	Primary Inputs	Primary Outputs
Tool Fingerprinting	tool_fingerprinting_script.py	Target domain list	raw_tool_outputs/*.json, tool_signals.csv
Job Posting Collection	job_post_scraper.py	Organization careers pages, Greenhouse/Lever APIs	job_signals.csv
LinkedIn Profile Signal Extraction	linkedin_employee_scraper.py	Manually exported LinkedIn profile PDFs	linkedin_signals.csv
GitHub Repository Collection	github_scraper.py	GitHub API, organization repository list	github_signals.csv
Normalization	normalize_all_signals.py	All per-source signal files	master_signals.csv
OSINT Extraction	extract_osint.py	master_signals.csv	normalized_osint.csv
Source Profile Computation	compute_source_profiles.py	master_signals.csv	source_profiles.csv
RQ1 Comparison	compare_osintvtool.py	normalized_tools.csv, normalized_osint.csv	rq1_comparison_summary.csv, rq1_comparison_details.csv, rq1_category_diversity.csv, rq1_source_contribution.csv
Method 1: Union Integration	union_integration.py	master_signals.csv	union_integration.csv, union_summary_by_company.csv
Method 1: Evaluation	evaluate_union.py	master_signals.csv, union_integration.csv	evaluation_overall.csv, evaluation_by_company.csv
Method 2: Weighted Aggregation	integration_method2_weighted.py	master_signals.csv	weighted_results_full.csv, weighted_company_summary.csv
Method 3: Evidence-Tier	integration_method3.py	master_signals.csv, source_profiles.csv	method3_evidence_scores.csv, method3_company_summary.csv

4.4 Normalization Module

The Normalization Module implements the signal standardization protocol described in Section 3.4, transforming heterogeneous technology signals collected from four structurally distinct source types into a consistent, comparable representation suitable for cross-source analysis and integration. This module is the critical convergence point of the pipeline: all upstream collection outputs feed into it, and all downstream integration and evaluation modules depend on its output.

The module performs four sequential operations on the combined signal input. Name standardization resolves surface-level variation in technology references across sources by mapping all observed name variants to a single canonical form. A technology name dictionary was constructed from the full collected dataset and applied consistently, ensuring that identical technologies referenced differently across sources are recognized as the same signal in all subsequent analyses.

Category assignment maps each canonically named technology to one of the thirteen technology categories defined in Table 3.2. This operation was performed through rule-based matching against category membership lists, with manual review applied to resolve ambiguous cases. Technologies that could not be confidently assigned to any named category were assigned to the residual category to preserve signal completeness while clearly marking their categorical uncertainty.

Deduplication removes within-source redundancy by ensuring that each technology contributes exactly one signal per source per organization. This prevents high-frequency signals, such as a technology mentioned repeatedly across multiple job postings for the same organization, from artificially inflating that source's contribution to downstream aggregation. Each signal is assigned a source-weight field representing the empirically derived source-strength value computed from agreement rate and category coverage by the Source Profile Computation Module. This value is used only by Method 2 for signal weighting and is not used in Method 3 evidence-tier classification.

Schema mapping produces the master signals file, the unified research dataset that serves as the single input to all downstream modules. The schema of this file is documented in Table 4.2.

Table 4.2 Schema of the master signals file. The source_weight field stores the empirically derived agreement rate for each source type and is used only by Method 2.

Field	Data Type	Description	Example Value
company	String	Target organization name	Stripe
source	String	Signal source type	jobs
tech	String	Canonical technology name	PostgreSQL
category	String	Technology category from taxonomy	Database

Field	Data Type	Description	Example Value
source_weight	Float	The source_weight field stores the empirically derived source-strength value computed from agreement rate and category coverage and is used only by Method 2	0.6467

4.5 Source Profile Computation Module

The Source Profile Computation Module implements the source profile derivation protocol described in Section 3.6, computing three empirical metrics for each source type from the master signals file. These metrics characterize each source's observed behavior across the full dataset and provide the data-driven weighting foundation for Method 2 of the Integration Engine.

The module reads the master signals file and, for each source type, computes agreement rate as the proportion of that source's signals that are independently corroborated by at least one other source across the same organization. Category coverage is computed as the proportion of the thirteen technology categories represented by that source's signals across the full dataset. Source strength is then computed as the arithmetic mean of agreement rate and category coverage, as specified in Section 3.6.

The resulting source profiles are stored in source_profiles.csv and are used by Method 2 as signal weighting factors. Method 3 does not use source strength values for classification; its evidence-tier assignments are determined only by source composition rules. It is important to note the precise scope of these values within the pipeline: source strength values are used exclusively in Method 2 as signal weighting factors. This distinction preserves the methodological separation between Method 2 and Method 3 as described in Chapter 3.

The implementation was carried out through the source profile computation script (compute_source_profiles.py). The module was executed after normalization and before integration, ensuring that source strength values reflect the actual composition of the normalized dataset rather than assumptions made prior to collection.

4.6 Integration Engine

The Integration Engine implements the three integration models described in Section 3.7, applying each model to the master signals file to produce unified technology inventories for each organization. The engine operationalizes RQ2 by generating three distinct integrated outputs that can be compared against each other and against the tool-only baseline to evaluate which integration methodology most effectively combines tool and OSINT signals.

4.6.1 Method 1: Union-Based Integration Module

The Union-Based Integration Module implements Method 1 by aggregating all signals from the master signals file into a single unified technology inventory per organization, without applying any weighting, filtering, or confidence differentiation. Every technology detected by

any source for a given organization is included in the integrated output, regardless of the number of sources that detected it or the strength of evidence supporting the detection.

The module produces three outputs: a full integrated technology list per organization, a company-level summary comparing tool-only counts against integrated counts, and a category-level summary documenting integration gains by technology category. These outputs collectively establish Method 1 as the high-recall integrated baseline, demonstrating the maximum technology coverage achievable through full union aggregation of tool and OSINT signals. The implementation was carried out through the union integration script (union_integration.py), with evaluation of coverage gains relative to the tool-only baseline implemented through the union evaluation script (evaluate_union.py).

4.6.2 Method 2: Source-Strength Weighted Aggregation Module

The Source-Strength Weighted Aggregation Module implements Method 2 by ranking technologies within each organization's integrated fingerprint according to cumulative source-strength scores derived from the empirically computed source profiles. For each technology in the master signals file, the module retrieves the source-strength value associated with its source type from the source-weight field populated during normalization. Where a technology is detected by multiple sources, the source-strength values of all detecting sources are summed to produce a cumulative score. Technologies are then ranked in descending order of cumulative score within each organization, with higher-ranked technologies representing those supported by the strongest and most corroborated evidence.

The module additionally applies a deduplication step at the source level to prevent mention count inflation. This step ensures that a technology detected many times within a single source, such as a subdomain name repeated across hundreds of Subfinder results, contributes only one source-strength vote rather than inflating its cumulative score proportionally to mention frequency. This design decision preserves the empirical integrity of the source-strength weighting by ensuring that each source votes once per technology per organization. The implementation was carried out through the source-strength weighted aggregation script (integration_method2_weighted.py).

4.6.3 Method 3: Evidence-Tier Classification Module

The Evidence-Tier Classification Module implements Method 3 by classifying each technology in the master signals file into one of four evidence tiers based entirely on rule-based criteria reflecting the source composition of each detection. The module applies no numeric scores, bonus values, or threshold parameters at any stage of classification. All classification decisions are determined by the observable presence or absence of specific source types in the detection record for each technology-organization pair.

The module aggregates all signals per technology per organization, identifies which source types contributed to each detection, and applies the four-tier classification logic described in Section 3.7.3. Technologies are grouped within each organization by evidence tier. Within each tier, technologies are listed by source composition for reporting clarity. No numeric score is used to determine tier assignment.

The module produces two outputs: a detailed evidence-tier file recording the evidence tier, coverage type, and source composition for every detected technology across all organizations, and a company-level summary documenting the distribution of evidence tiers and coverage types for each organization. Although the output file is named method3_evidence_scores.csv, the reported classification is based on evidence tiers and source composition only, not numeric scoring. These outputs form the primary analytical basis for answering RQ2. The implementation was carried out through the evidence-tier classification script (integration_method3.py).

4.7 Evaluation and Comparison Module

The Evaluation and Comparison Module implements the evaluation protocols described in Section 3.8 and Section 3.6, producing the quantitative outputs used to answer both research questions. The module operationalizes RQ1 through direct set-level comparison of tool-detected and OSINT-detected technology inventories, and supports RQ2 evaluation by compiling and comparing outputs from all three integration methods against the reference inventory and tool-only baseline

For RQ1 analysis, the module separates the master signals file into a tool-normalized signal set and an OSINT-normalized signal set per organization, implemented through the OSINT signal extraction script (extract_osint.py) which filters and re-normalizes OSINT source signals from the master file. The direct comparison module (compare_osintvtool.py) then computes, for each organization, the Jaccard similarity between the tool and OSINT technology sets, the OSINT coverage gain percentage, the descriptive set counts of tool-only, OSINT-only, and overlapping detections, the category-level breakdown of OSINT-only signals, and the per-source contribution to OSINT-only detection counts. These outputs collectively address all four measurement dimensions specified in Section 3.6.

For RQ2 evaluation, the module compiles outputs from all three integration methods and compares their coverage, ranking, and evidence-quality characteristics against the reference inventory and tool-only baseline. Method 1 outputs are evaluated through the union evaluation script (evaluate_union.py) for coverage gain metrics. Method 2 outputs are assessed through ranked technology lists reflecting source-strength weighted aggregation. Method 3 outputs are assessed through evidence-tier distributions documenting the proportion of Cross-Validated, Strong OSINT Evidence, Moderate Evidence, and Weak Evidence classifications across each organization.

4.8 System Data Flow Summary

The complete end-to-end data flow of the research pipeline is summarized in Figure 4.2, which traces the transformation of raw collection outputs through normalization, source profiling, integration, and evaluation to final research outputs.

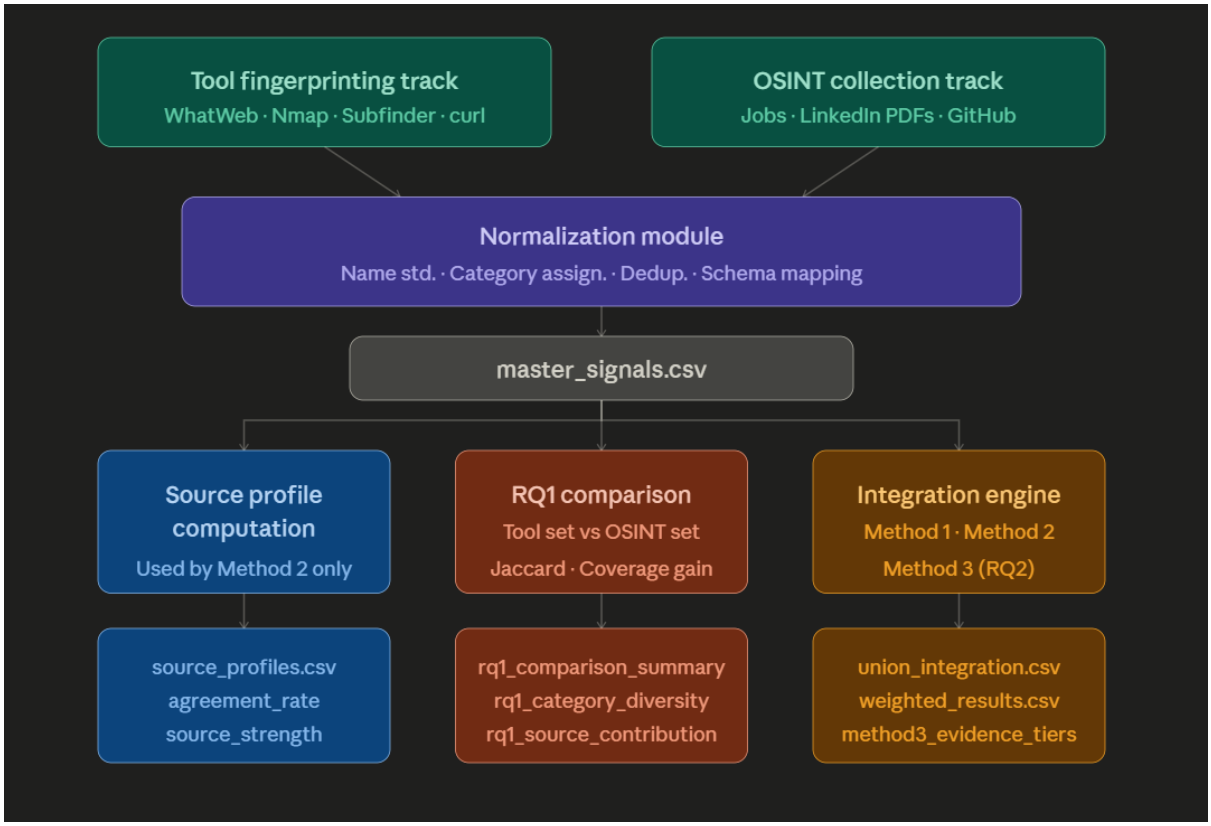

Figure 4.2 End-to-End data flow of the research pipeline, tracing the transformation of raw collection outputs through normalization, source profiling, RQ1 comparison, and three-method integration to final research outputs.

Chapter 5 Results and Evaluation

5.1 Overview

This chapter presents the empirical results of the study across both research questions. Sections 5.2 through 5.5 address RQ1 by documenting what active scanning tools and human-generated OSINT sources each detected, comparing their outputs across the twelve target organizations, and measuring the extent to which the two signal classes overlap or diverge. Sections 5.6 through 5.8 address RQ2 by evaluating the three integration methods, reporting their outputs in terms of the unit of analysis appropriate to each method, and comparing their characteristics across the studied organizations. Section 5.5 and 5.8 provide direct answers to both research questions derived from the evidence presented in this chapter.

Throughout this chapter, results are reported using the evaluation metrics defined in Chapter 3: Jaccard similarity for measuring signal overlap, integration coverage gain percentage for measuring OSINT contribution to integrated fingerprints, and descriptive set counts for characterizing the composition of each organization's technology inventory. All results are derived directly from the output files of the research pipeline described in Chapter 4.

5.2 Tool-Only Baseline Results

The tool-only baseline was established by running the four active scanning instruments against each of the twelve target organizations and recording the unique technologies detected after normalization and deduplication. The baseline represents the technology fingerprint achievable through tool-based reconnaissance alone, against which OSINT-enhanced approaches are subsequently compared. For the purposes of the RQ1 set comparison, three tool signal categories, specifically security headers, subdomain name patterns, and TLS security metadata, were excluded from the tool signal set as they represent infrastructure metadata rather than meaningful technology stack components. This filtering is applied consistently in the RQ1 comparison pipeline and is documented in the pipeline description in Chapter 4. This noise filtering approach represents the study's primary mechanism for handling low-quality tool signals, ensuring that the tool-only baseline reflects meaningful technology stack components rather than infrastructure metadata artifacts.

The tool-only detection counts for each organization are presented in Table 5.1.

Table 5.1 Tool-only detection counts after application of noise-category filtering for the RQ1 comparison pipeline. Tool completeness is the tool-detected count divided by the combined tool and OSINT technology count for each organization.

Organization	Tool-Detected Technologies	Tool Completeness
Asana	3	6.25%
Brex	1	1.35%
Coinbase	2	2.53%

Organization	Tool-Detected Technologies	Tool Completeness
Datadog	3	2.24%
Epic Systems	3	6.12%
Figma	2	3.70%
Instacart	3	4.11%
One Medical	3	30.00%
Scale AI	5	6.94%
Sentry	3	2.78%
Shopify	2	2.47%
Stripe	2	1.87%
Total	**32**	**Avg: 5.9%**

Table 5.1 shows that active scanning tools detected between one and five technologies per organization after filtering. The average tool completeness across the twelve organizations is 5.9%, indicating that tool-based reconnaissance alone captured less than six percent of the publicly detectable technology stack on average. Application-layer frameworks, programming languages, and internal infrastructure components were largely absent from the tool-only results for all organizations, consistent with the structural limitation of active scanning tools described in Chapter 2: tools can only observe technologies that produce externally visible network signals at the perimeter layer.

One Medical exhibits the highest tool completeness at 30%. This reflects not greater tool effectiveness but rather the comparatively small size of One Medical's OSINT-detectable technology inventory, which contains only seven technologies. With a smaller denominator, the three tool-detected technologies constitute a larger fraction of the combined inventory. This pattern underscores that tool completeness is a relative measure dependent on the total detectable inventory size rather than an absolute measure of tool capability.

5.3 Human-Generated OSINT Source Results

5.3.1 Source Profile Summary

Prior to integration, empirical source profiles were computed for each of the four source types from the full normalized dataset. The source profiles characterize each source's observed agreement rate, category coverage, and source strength, and serve as the empirical

foundation for the source-strength weighting applied in Method 2. Table 5.2 presents the source profile results.

Table 5.2 Empirically derived source profiles. Source strength is the arithmetic mean of agreement rate and category coverage. Tool signal count of 110 includes all categories; the filtered RQ1 pipeline count of 32 excludes noise categories.

Source	Total Signals	Agreement Rate	Category Coverage	Source Strength
Job Postings	203	0.7734	0.52	0.6467
LinkedIn	414	0.3720	0.32	0.3460
GitHub	850	0.1953	0.40	0.2977
Tools	110	0.0636	0.44	0.2518

Table 5.2 reveals several analytically significant patterns. Job postings exhibit the highest agreement rate at 0.7734, meaning approximately 77% of job posting signals were independently corroborated by at least one other source. This reflects the declarative nature of job posting content, in which organizations directly state the technologies their engineering teams use and require candidates to know.

GitHub produced the highest total signal count at 850, nearly four times the volume of job postings, yet exhibits the lowest agreement rate among OSINT sources at 0.1953. This reflects the implementation-level visibility of GitHub signals: dependency files, CI/CD configurations, and infrastructure manifests reveal technologies not disclosed through any other source channel and therefore show lower cross-source corroboration. This low agreement rate should be interpreted as evidence of unique visibility rather than unreliability.

Tools exhibit the lowest agreement rate of all source types at 0.0636, confirming that active scanning detects a set of technologies almost entirely disjoint from those revealed by OSINT sources. This finding directly anticipates the near-zero Jaccard similarity results reported in Section 5.4.

5.3.2 Per-Source Contribution to OSINT-Only Detections

Across the 857 organization-level OSINT-only detections, there were 1,126 source-level OSINT contributions because the same technology could be detected by more than one OSINT source independently. Table 5.3 presents the source-level contribution breakdown.

Table 5.3 Source-level OSINT contributions to OSINT-only detections. Percentages are over 1,126 total source-level contributions, not 857 unique detections, as a technology detected by multiple OSINT sources counts toward each source independently.

Source	Source-Level Contributions	Percentage of 1,126 Contributions
GitHub	516	45.83%
LinkedIn	411	36.50%
Jobs	199	17.67%
Total	**1,126**	**100%**

Table 5.3 shows that GitHub contributed the largest share of source-level OSINT contributions at 45.83%, followed by LinkedIn at 36.50% and job postings at 17.67%. The dominance of GitHub in OSINT-only detection volume is consistent with its implementation-level visibility, as public repositories directly expose technology choices through dependency manifests and configuration files that are not accessible through any other public source type.

Despite job postings exhibiting the highest source strength in Table 5.2, their lower contribution to OSINT-only source-level volume reflects the more targeted and role-specific nature of job posting content compared to the broad signal extraction possible across entire GitHub repository trees. Job postings contribute fewer but more reliably corroborated signals, while GitHub contributes larger volumes of signals with greater uniqueness.

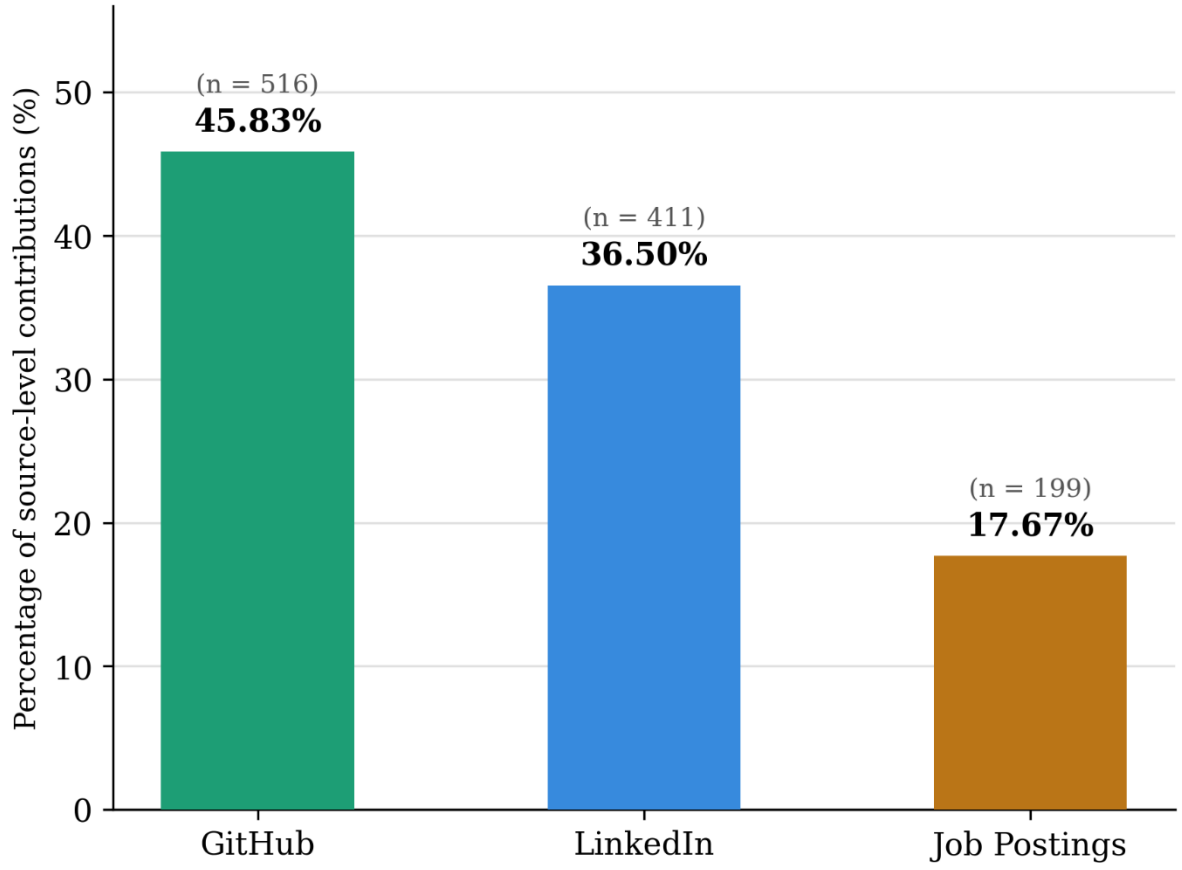

Figure 5.1 Source-level contribution of OSINT sources to OSINT-only detections.

5.4 RQ1: Tool vs OSINT Coverage Comparison

5.4.1 Jaccard Similarity and Coverage Gain Results

The core measurement for RQ1 compares the tool-detected and OSINT-detected technology sets for each organization using Jaccard similarity and OSINT coverage gain. The full per-organization comparison results are presented in Table 5.4.

Table 5.4 Per-organization RQ1 comparison results. Tool completeness and OSINT coverage gain do not sum to 100% as overlapping technologies are excluded from both. OSINT coverage gain = OSINT-only count / combined total.

Organization	Tool Techs	OSINT Techs	Combined	Common	Tool-Only	OSINT-Only	Tool Completeness	OSINT Coverage Gain	Jaccard
Asana	3	45	48	0	3	45	6.25%	93.75%	0.0000
Brex	1	73	74	0	1	73	1.35%	98.65%	0.0000
Coinbase	2	77	79	0	2	77	2.53%	97.47%	0.0000

Organization	Tool Techs	OSINT Techs	Combined	Common	Tool-Only	OSINT-Only	Tool Completeness	OSINT Coverage Gain	Jaccard
Datadog	3	131	134	0	3	131	2.24%	97.76%	0.0000
Epic Systems	3	46	49	0	3	46	6.12%	93.88%	0.0000
Figma	2	52	54	0	2	52	3.70%	96.30%	0.0000
Instacart	3	70	73	0	3	70	4.11%	95.89%	0.0000
One Medical	3	7	10	0	3	7	30.00%	70.00%	0.0000
Scale AI	5	71	72	4	1	67	6.94%	93.06%	0.0556
Sentry	3	105	108	0	3	105	2.78%	97.22%	0.0000
Shopify	2	79	81	0	2	79	2.47%	97.53%	0.0000
Stripe	2	106	107	1	1	105	1.87%	98.13%	0.0093
Average							**5.9%**	**94.1%**	**0.005**

Table 5.4 presents the central empirical finding of this study. Across all twelve organizations, the Jaccard similarity between tool-detected and OSINT-detected technology sets is 0.0000 for ten of twelve organizations. The two exceptions are Stripe, with a Jaccard similarity of 0.0093, and Scale AI, with a Jaccard similarity of 0.0556. Even these values represent near-zero overlap in absolute terms. The average Jaccard similarity across all twelve organizations is 0.005, confirming that tool-based and OSINT-based technology fingerprinting produce almost entirely disjoint signal sets across the studied sample.

OSINT-only detections account for between 70.00% and 98.65% of the combined technology inventory across the twelve organizations, with an average OSINT coverage gain of 94.1%. The lowest coverage gain is observed for One Medical at 70.00%, attributable to its limited public engineering footprint and the correspondingly small OSINT-detectable technology inventory of only seven technologies. The highest coverage gain is observed for Brex at 98.65%, where active scanning detected only one technology while OSINT sources revealed seventy-three additional organization-level detections not visible at the perimeter layer.

Scale AI represents the only organization with non-trivial overlap between tool and OSINT signal sets, with four technologies detected by both source types and a Jaccard similarity of 0.0556. This organization's active public technical presence across GitHub, job postings, and LinkedIn, combined with a comparatively larger tool-detected footprint of five technologies, accounts for this higher-than-average overlap. Even so, 67 of Scale AI's 72 combined technologies remained OSINT-only detections, confirming the structural separation of the two signal classes even in the most favorable case for overlap.

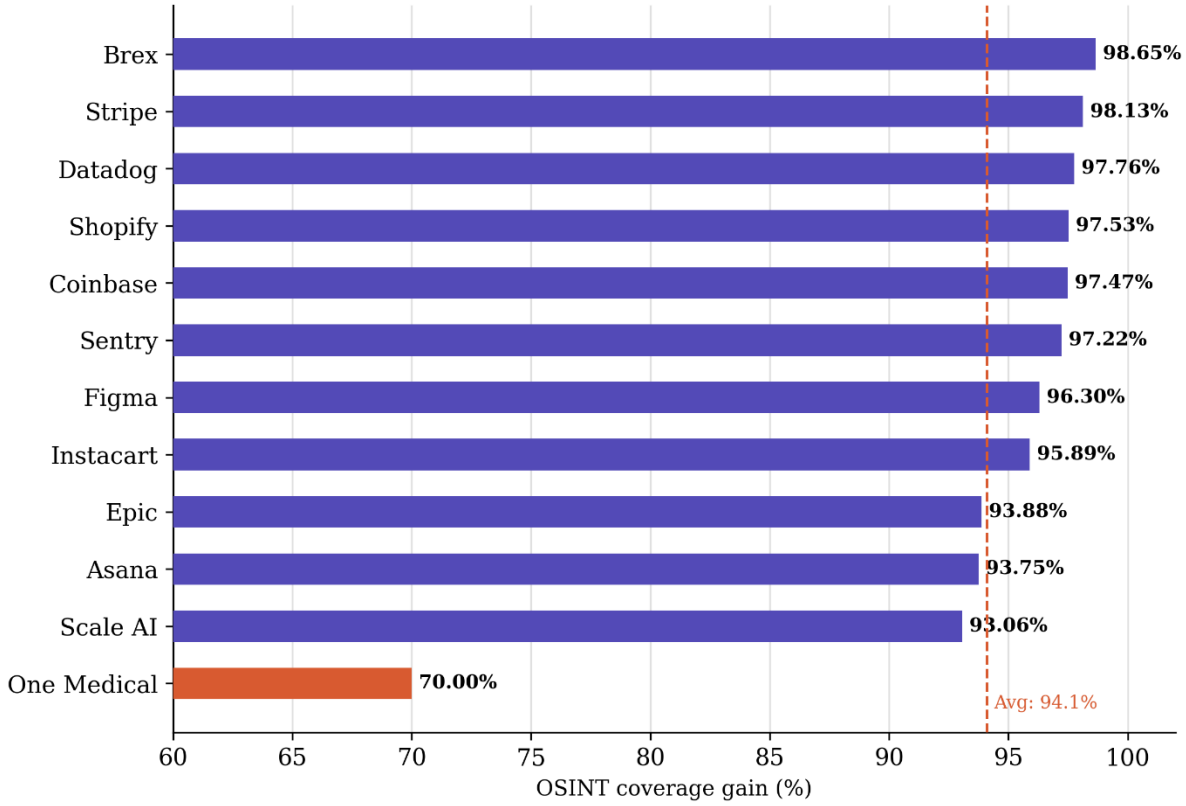

Figure 5.2 OSINT coverage gain by organization, with average coverage gain shown as a reference line.

5.4.2 Category-Level Analysis of OSINT-Only Detections

Beyond the quantitative coverage comparison, the category-level composition of OSINT-only detections reveals what kinds of technologies tools systematically miss. Table 5.5 presents the distribution of OSINT-only detections across the technology category taxonomy.

Table 5.5 Distribution of OSINT-only technology detections across the category taxonomy. Categories represent technologies detected exclusively by OSINT sources and absent from the filtered tool-only baseline across all twelve organizations.

Category	Tech Count	Percentage of OSINT-Only
Language	297	34.66%
Other	228	26.60%

Category	Tech Count	Percentage of OSINT-Only
Framework	152	17.74%
DevOps	41	4.78%
CI/CD	36	4.20%
Security	31	3.62%
Database	28	3.27%
Cloud	27	3.15%
Infrastructure	8	0.93%
Web Server	5	0.58%
Data	4	0.47%
Total	**857**	**100%**

Table 5.5 shows that the largest named category of OSINT-only detections is Language at 34.66%, followed by Framework at 17.74%. DevOps and CI/CD tooling together account for 8.98% of OSINT-only detections. These categories encompass precisely the technologies that active scanning tools cannot reach: programming languages are disclosed through GitHub dependency files and employee skill listings but produce no externally observable network signals, application frameworks are deployed behind load balancers and proxies that actively conceal their identity, and CI/CD tools operate entirely within internal development infrastructure.

The Other category accounts for 26.60% of OSINT-only detections, representing the second largest category in the dataset. This reflects the breadth of technology signals captured across LinkedIn profiles and GitHub repositories that do not map cleanly to the current named technology taxonomy. This finding suggests that the current taxonomy, while sufficient for structured comparison, does not fully capture the diversity of technology signals available in human-generated OSINT. This limitation is discussed further in Chapter 6.

The near-complete absence of tool-detectable categories such as Web Server and Infrastructure from the OSINT-only set confirms the structural complementarity of the two signal classes. Tools capture perimeter-layer infrastructure signals while OSINT reaches the application and development layers that tools structurally cannot observe.

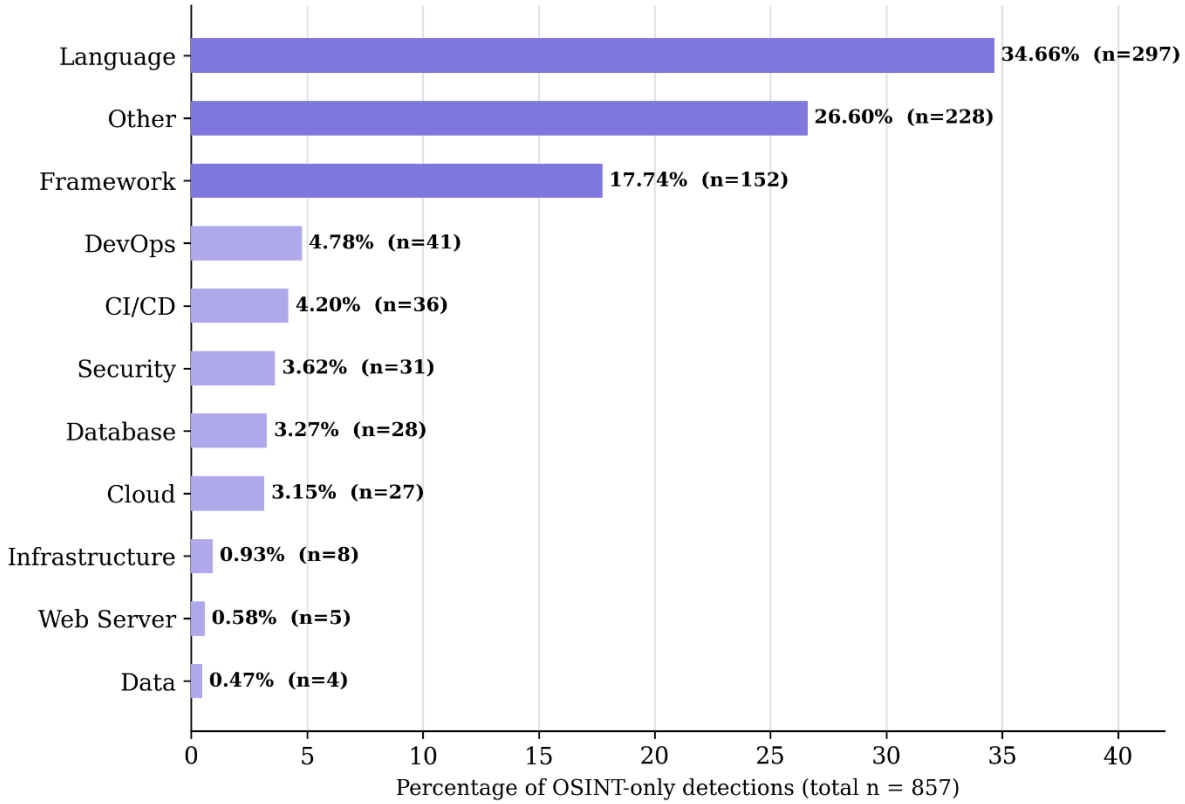

Figure 5.3 Distribution of OSINT-only detections across technology categories.

5.5 Summary Answer to RQ1

The results presented in Sections 5.2 through 5.4 provide a clear and consistent answer to RQ1. Human-generated OSINT sources reveal a substantially larger set of organization-level technology detections than traditional active scanning tools produce, and the two signal classes are largely non-overlapping. Across the twelve studied organizations, OSINT contributed 857 organization-level detections absent from the filtered tool-only baseline, representing an average coverage gain of 94.1%. The average Jaccard similarity of 0.005 confirms that tool-based and OSINT-based technology fingerprinting operate at fundamentally different layers of the organizational technology stack rather than providing redundant coverage of the same technologies. These findings strongly support the premise that human-generated OSINT constitutes a complementary and necessary signal class for comprehensive technology fingerprinting.

5.6 RQ2: Integration Method Results

Having established that tool and OSINT signals are largely complementary, the second part of the evaluation examines the three integration methods designed to combine these signal classes. All three methods integrate tool and OSINT signals into technology fingerprints substantially richer than the tool-only baseline, but they differ in output structure: Method 1 maximizes recall through union aggregation, Method 2 prioritizes signals by empirically derived source strength, and Method 3 classifies evidence quality through rule-based tiers.

Each method is reported in terms of its appropriate unit of analysis, and results should not be compared as equivalent raw counts across methods.

5.6.1 Method 1: Union-Based Integration

Method 1 aggregates all signals from all source types without weighting or filtering. Its output unit of analysis is unique company-technology pairs after union aggregation. Table 5.6 presents the Method 1 results for each organization.

Note that the tool-only counts in Table 5.6 differ from those in Table 5.4. Table 5.4 reports deduplicated tool technologies after application of the noise-category filter used in the RQ1 comparison pipeline, which excludes security header, subdomain, and TLS security signals. Table 5.6 reports the Method 1 pipeline's broader tool signal count, which retains all tool signal categories before union integration. Because these outputs reflect different units of analysis and different pipeline filtering stages, the counts should not be interpreted as directly interchangeable.

Table 5.6 Method 1 union-based integration results. Integration coverage gain = new OSINT technologies / total integrated count. Tool-only counts include all signal categories without noise filtering, explaining the difference from Table 5.4.

Organization	Tool-Only Count	After Integration	New from OSINT	Integration Coverage Gain
Asana	9	61	52	85.25%
Brex	4	87	83	95.40%
Coinbase	6	117	111	94.87%
Datadog	7	192	185	96.35%
Epic Systems	13	61	48	78.69%
Figma	10	77	67	87.01%
Instacart	10	113	103	91.15%
One Medical	6	13	7	53.85%
Scale AI	14	93	79	84.95%
Sentry	9	212	203	95.75%
Shopify	11	113	102	90.27%
Stripe	11	158	147	93.04%
Total	**110**	**1,297**	**1,187**	

Table 5.6 shows that union-based integration produced 1,297 unique company-technology pairs across all twelve organizations, compared to an unfiltered tool-only baseline of 110 signals. The integration coverage gain ranges from 53.85% for One Medical to 96.35% for Datadog, with all twelve organizations showing substantial OSINT contribution to the integrated fingerprint.

Datadog and Sentry exhibit the highest absolute gains, with 185 and 203 new technologies contributed by OSINT respectively, reflecting their large public engineering presence across GitHub repositories and active job posting programs. One Medical again

represents the lower boundary, consistent with its limited public technology disclosure observed across all stages of the analysis. The variation in integration coverage gain across organizations reflects differences in OSINT visibility rather than differences in integration method performance, since Method 1 applies no filtering or quality differentiation and functions as a high-recall integrated baseline.

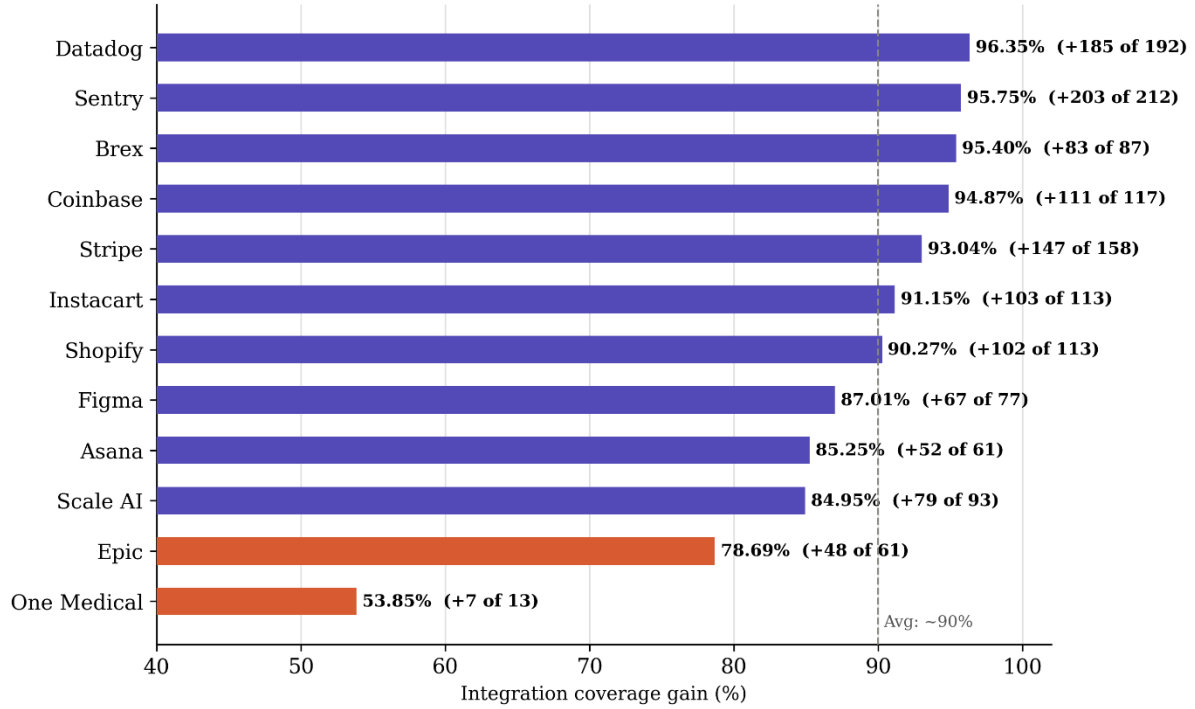

Figure 5.4 Method 1 union-based integration coverage gain by organization.

5.6.2 Method 2: Source-Strength Weighted Aggregation

Method 2 ranks technologies within each organization's integrated fingerprint according to cumulative source-strength scores derived from the empirically computed source profiles. Its output unit of analysis is weighted source-technology records before final ranking aggregation. Table 5.7 presents the Method 2 company-level summary.

Table 5.7 Method 2 source-strength weighted aggregation results. Total of 1,330 reflects source-technology combinations before aggregation, differing from Method 1's 1,297 unique company-technology pairs due to different units of analysis.

Organization	Total Weighted Records	Multi-Source Technologies	Avg Weighted Score	Max Weighted Score
Asana	61	11	0.4196	0.9687
Brex	87	12	0.4109	1.3407

Organization	Total Weighted Records	Multi-Source Technologies	Avg Weighted Score	Max Weighted Score
Coinbase	117	17	0.3900	1.3407
Datadog	192	16	0.3251	1.3407
Epic Systems	61	9	0.3919	1.1454
Figma	77	11	0.3529	1.3407
Instacart	113	16	0.3773	1.3407
One Medical	13	0	0.1790	0.7734
Scale AI	93	19	0.4288	1.4043
Sentry	212	26	0.3171	1.3407
Shopify	113	14	0.3507	1.3407
Stripe	158	23	0.3793	1.3407
Total	**1,330**	**173**		

Table 5.7 shows that Method 2 produced 1,330 weighted source-technology records and identified 173 multi-source technologies across all twelve organizations. The total of 1,330 differs from Method 1's 1,297 because Method 2 counts weighted source-technology records before aggregation while Method 1 counts unique company-technology pairs after union aggregation. Both methods operate on the same normalized input dataset and the difference reflects the different units of analysis rather than any inconsistency in the underlying data.

The maximum weighted score of 1.4043 is achieved by Scale AI for both AWS and Google Cloud, representing detection across all four source types simultaneously. This cumulative score equals the sum of all four empirically derived source-strength values: 0.7734 for job postings, 0.3720 for LinkedIn, 0.1953 for GitHub, and 0.0636 for tools. All other organizations achieve a maximum score of 1.3407, reflecting three-source corroboration from GitHub, job postings, and LinkedIn without tool confirmation, a pattern directly consistent with the near-zero tool-OSINT overlap established in RQ1.

One Medical stands as a distinct outlier with zero multi-source technologies and the lowest average weighted score of 0.1790. This result is consistent with its limited public engineering footprint observed consistently across all methods, confirming that the absence of multi-source corroboration is a property of One Medical's public disclosure pattern rather than a limitation of the integration methodology.

5.6.3 Method 3: Evidence-Tier Classification

Method 3 classifies each detected technology into one of four evidence tiers based entirely on rule-based criteria reflecting source composition. Its output unit of analysis is evidence-tier classifications generated after source-composition grouping. Table 5.8 presents the per-organization evidence-tier distribution.

Table 5.8 Method 3 evidence-tier distribution. Totals differ across methods due to different units of analysis; cross-method comparisons use Chapter 3 metrics.

Organization	Total	Cross-Validated	Strong OSINT	Moderate	Weak	CV Ratio	Strong Ratio
Asana	61	0	11	50	0	0.000	0.180
Brex	91	0	12	44	35	0.000	0.132
Coinbase	121	0	17	79	25	0.000	0.140
Datadog	200	0	16	143	41	0.000	0.080
Epic Systems	62	0	9	26	27	0.000	0.145
Figma	80	0	11	63	6	0.000	0.138
Instacart	114	0	16	67	31	0.000	0.140
One Medical	13	0	0	13	0	0.000	0.000
Scale AI	95	4	15	58	18	0.042	0.158
Sentry	218	0	26	171	21	0.000	0.119
Shopify	115	0	14	67	34	0.000	0.122
Stripe	161	1	22	93	45	0.006	0.137
Total	**1,331**	**5**	**169**	**874**	**283**		

Table 5.8 presents the Method 3 classification results. Across all twelve organizations, 5 technologies are Cross-Validated, 169 are Strong OSINT Evidence, 874 are Moderate Evidence, and 283 are Weak Evidence.

The most analytically significant finding from Method 3 is the near-zero Cross-Validated count. Only five technologies across the entire dataset achieved Cross-Validated status, concentrated in Scale AI with four and Stripe with one. This result should not be interpreted as a limitation of the evidence-tier model. Rather, it provides direct confirmation of the RQ1 finding: because tool-detected and OSINT-detected signal sets are almost entirely

disjoint, technologies confirmed by both source classes must be rare by definition. The low Cross-Validated count is structurally predicted by the near-zero Jaccard similarity values observed in Section 5.4 and reinforces the central thesis that tools and OSINT operate at complementary rather than overlapping layers of the technology stack.

The majority of technologies, 874 of 1,331 or 65.7%, fall into the Moderate Evidence tier, representing technologies detected by tools only or by a single OSINT source with strong contextual evidence. The Strong OSINT Evidence tier captures 169 technologies or 12.7% of the total, representing technologies corroborated by two or more independent OSINT sources without tool confirmation. These detections represent the most analytically reliable OSINT-exclusive findings in the dataset, as they achieve multi-source corroboration within the OSINT layer alone.

One Medical exhibits a unique pattern, with all thirteen technologies classified as Moderate Evidence and zero in either Strong OSINT or Weak Evidence tiers. This distribution confirms the limited public engineering footprint finding: One Medical's technologies are detectable through individual sources but do not achieve the multi-source corroboration required for Strong OSINT classification.

Asana presents a contrasting pattern, with zero Weak Evidence classifications, meaning every technology in Asana's integrated fingerprint is supported by at least moderate contextual evidence. Similarly, Asana's Strong OSINT ratio of 0.180 is the highest in the dataset, indicating that a greater proportion of its technologies are corroborated by multiple independent OSINT sources relative to its total integrated inventory.

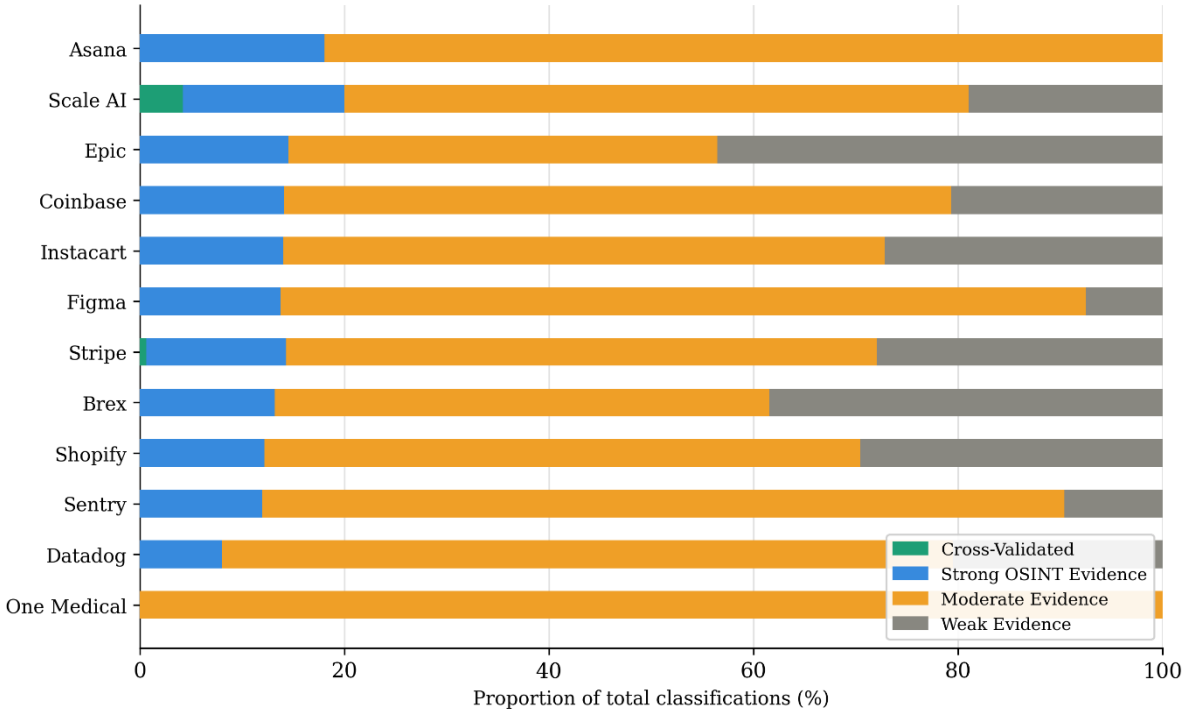

Figure 5.5 Method 3 evidence-tier distribution by organization.

5.7 Integration Method Comparison

Table 5.9 provides a summary comparison of all three integration methods, enabling direct evaluation of their respective characteristics in terms of RQ2.

Table 5.9 Comparison of the three integration methods across key analytical dimensions. Total output counts differ because each method produces a different unit of analysis and should not be compared as equivalent raw counts.

Dimension	Method 1: Union	Method 2: Source-Strength Weighted	Method 3: Evidence-Tier
Unit of analysis	Unique company-technology pairs	Weighted source-technology records	Evidence-tier classifications
Total output count	1,297	1,330	1,331
Output type	Flat technology list	Ranked technology list	Tiered technology classification
Output characteristic	Maximum recall	Prioritizes multi-source signals	Separates technologies by evidence quality
Multi-source technologies	Not differentiated	173 identified	169 Strong OSINT + 5 Cross-Validated
One Medical multi-source	Not differentiated	0	0
Primary analytical utility	High-recall integrated baseline	Ranking by empirical source strength	Rule-based evidence quality classification

Table 5.9 shows that all three integration methods expand technology coverage beyond the tool-only baseline, each producing a substantially richer set of detected technologies than active scanning alone. The methods differ primarily in how they organize and present that expanded coverage.

Method 1 provides the broadest technology inventory by including all detected signals without quality differentiation, making it suitable as a high-recall integrated baseline and for use cases where completeness is prioritized over evidence quality. Method 2 provides a ranked output in which technologies supported by stronger and more diverse evidence sources rank more prominently, making it suitable for use cases requiring prioritization of the most reliably corroborated detections. Method 3 provides an explicitly classified output in which each technology's evidence quality is transparently documented through rule-based tier assignment, making it suitable for use cases requiring interpretable and defensible evidence characterization without reliance on numeric scoring.

The three methods are best understood as complementary analytical approaches that serve different aspects of technology fingerprinting rather than as competing approaches where one definitively outperforms the others.

5.8 Summary Answer to RQ2

The results presented in Sections 5.6 and 5.7 provide a direct answer to RQ2. All three integration methodologies effectively combine human-generated OSINT with tool-based reconnaissance, but they do so through different output structures. Method 1 establishes a high-recall integrated baseline, with integration coverage gains ranging from 53.85% to 96.35% across the twelve organizations. Method 2 identifies 173 multi-source technologies across the dataset and provides a ranked output that prioritizes signals supported by the strongest empirical source-strength evidence, with the highest-scoring technologies achieving corroboration across three or four independent source types. Method 3 classifies the integrated technology inventory into evidence tiers through entirely rule-based criteria, revealing that only five technologies across the full dataset achieve cross-source validation between tools and OSINT, a finding that directly reinforces the structural complementarity established in RQ1 rather than contradicting it. Together, the three methods demonstrate that structured integration can improve technology fingerprinting by expanding coverage, prioritizing stronger signals, and making evidence quality more interpretable than tool-only reconnaissance alone.

Chapter 6 Discussion

This chapter interprets the empirical results presented in Chapter 5 in relation to the two research questions, evaluates the methodological design choices made in Chapters 3 and 4, acknowledges limitations and validity considerations, and discusses the practical implications of the findings for cybersecurity practice. No new data or results are introduced in this chapter.

6.1 Interpretation of RQ1 Findings

The central quantitative finding of this study is that the average Jaccard similarity between tool-detected and OSINT-detected technology sets across the twelve studied organizations is 0.005, and is exactly zero for ten of the twelve. This figure requires interpretation beyond its surface reading. A Jaccard similarity of zero does not indicate that either source class is failing or producing erroneous signals. It indicates that the two classes are observing fundamentally different layers of the organizational technology stack. Active scanning tools produce signals from the perimeter layer, where externally visible services, network ports, HTTP headers, and server software declarations are observable. Human-generated OSINT sources, by contrast, produce signals from the application and development layers, where the technologies that power internal systems, development pipelines, and software products are disclosed through job postings, employee profiles, and public repositories.

The 94.1% average OSINT coverage gain reported in Chapter 5 reflects this structural separation directly. It does not mean that OSINT is more reliable or more accurate than active scanning. It means that OSINT reaches a category of technology intelligence that tools cannot structurally access, and that this category constitutes the overwhelming majority of the publicly inferable technology surface for the organizations studied. The finding that tool-based reconnaissance alone captured an average of 5.9% of the combined publicly detectable technology inventory suggests that security assessments relying exclusively on active scanning are working with a substantially incomplete picture of organizational technology exposure.

The category-level breakdown from Chapter 5 adds interpretive depth to this finding. Languages at 34.66% and frameworks at 17.74% of OSINT-only detections represent the application logic layer of organizational technology stacks, precisely the layer that active scanning cannot reach because it produces no externally observable network signals. The presence of DevOps and CI/CD tooling within OSINT-only detections further confirms that internal development infrastructure, which is never exposed at the network perimeter, is systematically disclosed through human-generated public channels.

6.2 Interpretation of RQ2 Findings

The three integration methods evaluated in Chapter 5 each address RQ2 from a different analytical angle, and the results demonstrate that no single method is universally superior. Each serves a distinct and legitimate purpose within the broader goal of enhancing technology fingerprinting through OSINT integration.

Method 1 demonstrates that the simple act of combining tool and OSINT signals without any weighting or filtering produces substantially richer technology inventories than tools alone,

with integration coverage gains ranging from 53.85% to 96.35% across the twelve organizations. This finding establishes that the coverage improvement from OSINT integration does not depend on sophisticated fusion machinery. Even the simplest possible integration approach produces a meaningful result. Method 1 is therefore most appropriate for use cases where maximum recall is the priority, such as initial reconnaissance phases where completeness matters more than evidence quality.

The union-based model produces the largest integrated fingerprint but is not considered the strongest integration approach precisely because size is not equivalent to quality. Method 1 includes every signal regardless of how weakly it is evidenced, meaning a single unverified LinkedIn self-report and a GitHub dependency file confirmed by three sources are treated identically. In contexts where practitioners need to act on fingerprinting results, such as vulnerability prioritization or threat modeling, an undifferentiated list of all detected technologies is less actionable than a ranked or tiered output that communicates evidence quality. This is why Method 2 and Method 3 were designed as progressively more selective approaches. Method 3 in particular does not produce the largest fingerprint but produces the most interpretable one, because every detection is accompanied by a transparent evidence classification that communicates how confidently it was identified and from how many independent sources

Method 2 demonstrates that empirically derived source-strength weighting produces a differentiated ranking of technologies in which the most strongly corroborated detections appear prominently. The finding that only Scale AI achieved four-source corroboration, while most multi-source technologies achieved three-source corroboration, reflects the structural separation established by RQ1: because tools and OSINT rarely detect the same technologies, the maximum possible weighted score requires a rare alignment across all source types. Method 2 is most appropriate for use cases where a practitioner needs to prioritize which detected technologies warrant immediate attention.

Method 3 demonstrates that rule-based evidence-tier classification provides a transparent and auditable structure for organizing the integrated technology inventory without any numeric scoring. The near-zero Cross-Validated count (5 out of 1,331 total classifications) is not a limitation of Method 3 but a direct structural consequence of the RQ1 finding, and is discussed further in Section 6.3. Method 3 is most appropriate for use cases where the quality and interpretability of evidence classification matters, such as formal security assessments or reporting contexts where every detection must be defensible.

6.3 Structural Complementarity of Tools and Human-Generated OSINT

The central theoretical contribution of this thesis extends beyond its individual empirical findings. The consistent near-zero Jaccard similarity across ten of twelve organizations, the near-zero Cross-Validated count in Method 3, and the 94.1% average OSINT coverage gain collectively constitute evidence for a structural thesis: tools and human-generated OSINT are not competing approaches to technology fingerprinting that produce overlapping results of differing quality. They are observationally distinct instruments that measure different properties of the same organizational technology landscape.

This structural complementarity has implications for how technology fingerprinting should be conceptualized in cybersecurity practice. The prevailing assumption underlying active scanning tool use is that a sufficiently comprehensive tool suite can produce a reasonably complete technology inventory. The results of this study challenge that assumption empirically. The technologies that organizations use internally, the languages their engineers write in, the frameworks their applications run on, the CI/CD tools that deploy their software, and the cloud services their infrastructure depends on are largely invisible to perimeter scanning precisely because they are internal. They are, however, systematically and voluntarily disclosed through the routine activities of hiring, employee professional networking, and open-source development.

This finding connects to the theoretical framework established in Chapter 2. Makrushin [3] identified human-generated OSINT as the least systematically exploited component of attack surface analysis. Browne et al. [2] noted that most OSINT research in cybersecurity focuses on technical signals rather than human-generated disclosures. The present study provides empirical evidence that this gap in the literature corresponds to a real and measurable gap in practical reconnaissance capability, and that structured integration of human-generated OSINT with active scanning can substantially close it.

6.4 Evaluation of Design Choices

Four key design decisions made in Chapter 3 merit evaluation in light of the results obtained in Chapter 5: the noise-category filter applied to tool signals, the three-method integration design, the purposive sampling strategy, and the thirteen-category technology taxonomy.

The decision to apply a noise-category filter to the tool signal set in the RQ1 comparison pipeline, excluding security header, subdomain pattern, and TLS metadata signals, was validated by the results. These categories do not represent technology stack components in the sense addressed by the research questions. Their inclusion would have inflated the tool-only baseline with signals that are not meaningfully comparable to the application-layer and development-layer signals produced by OSINT sources, producing a misleading picture of tool-OSINT overlap. The filter was the correct methodological choice and its effect, the difference between the filtered count of 32 and the unfiltered count of 110 in Chapter 5, is appropriately documented and explained.

The decision to design three integration methods of increasing sophistication rather than a single integration approach proved analytically productive. The results reveal that the three methods produce meaningfully different outputs that are useful in different contexts, validating the multi-method design. A single-method evaluation would have answered RQ2 less completely by providing only one perspective on how integration can be structured.

The decision to use purposive sampling of twelve organizations spanning diverse industry verticals was appropriate for the exploratory framing of the study. The consistent pattern of near-zero Jaccard similarity across organizations as different as a healthcare provider (One Medical), a financial technology company (Brex), and an enterprise software company (Asana) suggests that the structural complementarity finding is not an artifact of any particular industry or organizational type. The consistency of the pattern across the sample strengthens

the generalizability argument within the scope of technology-active organizations with public engineering presence.

The thirteen-category technology taxonomy, however, requires honest evaluation. The Other category constitutes 26.60% of OSINT-only detections, making it the second largest category in the dataset. This result indicates that the taxonomy, while useful for structured comparison across the named categories, was not fully expressive enough to capture the diversity of technology signals present in human-generated OSINT. A broader or more granular taxonomy would likely have produced more informative category-level analysis. This is identified as a limitation in Section 6.6 and as a direction for future work in Chapter 7.

6.5 Anomalies and Unexpected Findings

Four findings in Chapter 5 warrant specific discussion as anomalies or results that require explanation beyond their quantitative presentation.

The first is the consistent behavior of One Medical across all three integration methods. One Medical exhibits the lowest OSINT coverage gain at 70.00%, zero Strong OSINT Evidence classifications in Method 3, zero multi-source technologies in Method 2, and the smallest OSINT-detectable technology inventory of any organization in the sample at seven technologies. This pattern is consistent across every analytical dimension of the study, confirming that the finding reflects a genuine property of One Medical's public technology disclosure posture rather than an artifact of any particular method. One Medical's limited public engineering footprint, which may reflect a more conservative public disclosure posture relative to the technology-native organizations in the sample, results in fewer publicly observable technology signals across all three OSINT source types.

The second anomaly is Scale AI's position as the only organization with non-trivial Jaccard similarity (0.0556) and the only organization with four Cross-Validated technologies. This reflects Scale AI's unusually active public technical presence: a large GitHub repository portfolio, extensive technical job posting activity, and a high density of employee LinkedIn profiles with specific technology disclosures. Scale AI also has the largest tool-detected footprint in the sample at five technologies, providing more surface for cross-source validation. Even so, 67 of its 72 combined technologies remain OSINT-only detections, confirming that the structural complementarity finding holds even in the most favorable case for overlap.

The third anomaly requiring discussion is the ‘Other’ category at 26.60% of OSINT-only detections. This is the second largest category in the dataset and signals a genuine limitation of the thirteen-category taxonomy. The large ‘Other’ category suggests that the taxonomy was useful for structured comparison but not fully expressive enough to capture the diversity of technology signals present in human-generated OSINT. Future work should refine or expand the taxonomy to reduce the proportion of signals classified as Other, which would improve the analytical resolution of category-level findings.

The fourth finding worth discussing is the near-zero Cross-Validated count in Method 3, with only five technologies out of 1,331 classifications achieving Cross-Validated status. As discussed in Sections 6.1 and 6.3, this is not a methodological anomaly but a direct consequence of the structural complementarity of tools and OSINT established by RQ1. It is, however, worth noting that the five Cross-Validated technologies that did achieve this status,

primarily cloud platform technologies such as AWS and Google Cloud, represent technologies that are both perimeter-visible through tool scanning and extensively disclosed through all three OSINT source types. This suggests that only the most ubiquitous and universally adopted technologies are likely to achieve cross-source validation between tools and OSINT in this analytical framework.

6.6 Validity and Limitations

6.6.1 Validity Considerations

Several methodological choices require explicit validity justification given that this study departs from conventional ground-truth-based evaluation designs.

The use of a reference inventory constructed from consolidated public evidence, rather than an internal company-verified technology inventory, is the most significant validity consideration. The reference inventory was constructed from the union of all publicly discernible technology signals across all source types, filtered to exclude noise categories. While this construction cannot guarantee completeness relative to an organization's actual technology stack, it provides a consistent and principled comparison baseline for measuring the relative coverage of tool-only versus OSINT-enhanced fingerprinting. The study's conclusions about the coverage gain from OSINT integration are valid within this frame: they describe how much OSINT adds to what is publicly inferable, which is precisely the scope of the research questions as stated.

The near-zero Jaccard similarity finding is valid as a structural observation because it derives from the same dataset and normalization pipeline applied consistently across both source types. The finding does not depend on the completeness of the reference inventory. Whether or not the reference inventory captures the full organizational technology stack, the near-zero overlaps between what tools detect and what OSINT detects is a direct observation of the two signal sets, not an inference about any unobserved ground truth.

The source-strength values used in Method 2 are defensible because they are computed from the data itself rather than assigned by expert judgment. The formula, source strength as the arithmetic mean of agreement rate and category coverage, uses no manually chosen parameters and produces values that directly reflect observed source behavior across the full dataset. The source-strength weighting in Method 2 is therefore grounded in empirical observation rather than assumption.

Method 3's evidence-tier classification is defensible because every classification decision is determined by observable, rule-based criteria applied to the source composition of each technology detection. The near-zero Cross-Validated count is not a failure of the model but a finding that reflects the structural reality documented by RQ1. No arbitrary thresholds, bonus values, or manually assigned scores appear anywhere in the Method 3 classification logic, which was specifically designed to address concerns about numeric parameter defensibility.

6.6.2 Limitations

The study has six limitations that should be considered when interpreting its findings.

First, the reference inventory is constructed from publicly consolidated evidence rather than internal company-verified truth. Technologies used internally but not disclosed through any public channel are invisible to this study by design. The coverage gains reported reflect gains within the publicly observable domain, not gains relative to the full organizational technology stack.

Second, LinkedIn data collection was performed through structured manual extraction from PDF-exported profiles rather than automated collection. This introduces potential sampling bias in the LinkedIn signal set, as the profiles selected for extraction may not be fully representative of the technical workforce at each organization. The smaller LinkedIn signal volume relative to GitHub and job postings partially reflects this collection constraint.

Third, the thirteen-category taxonomy is an incomplete representation of the full diversity of technology signals in human-generated OSINT, as evidenced by the 'Other' category constituting 26.60% of OSINT-only detections. Category-level findings should be interpreted with awareness that a substantial proportion of signals are not captured by the named categories.

Fourth, the study was conducted across a purposively selected sample of twelve organizations. While the consistency of the structural complementarity finding across diverse organizational types strengthens its credibility, the sample is not statistically representative of any broader population of organizations, and the findings should not be generalized beyond technology-active organizations with meaningful public engineering presence.

Fifth, OSINT signals are temporally sensitive. Job postings are removed when positions are filled, LinkedIn profiles are updated as employees change roles, and GitHub repositories evolve continuously. The reference inventory and OSINT signal sets reflect a snapshot collected at a specific point in time, and the coverage characteristics observed in this study may differ if collection were repeated at a later date.

Sixth, the active scanning tool suite used in this study, comprising WhatWeb, Nmap, Subfinder, and curl header inspection, represents one configuration of tool-based reconnaissance among many possible configurations. Different tool selections or scan configurations might produce modestly different tool-only baselines, though the structural limitation of tool-based reconnaissance to perimeter-visible signals would remain regardless of tool choice.

A related concern specific to LinkedIn data is the possibility that employees list technologies from previous positions rather than their current role. LinkedIn skill endorsements and role descriptions may reflect a career history of technology exposure rather than the technologies actively used at the studied organization. This study partially mitigates this by extracting signals from experience sections associated specifically with the target organization's name, and by treating LinkedIn signals as weak evidence that requires corroboration from other sources before contributing to higher evidence tiers. However, the possibility of historical skill contamination cannot be fully eliminated without access to verified employment records.

The active scanning tool suite used in this study was limited to open-source and freely accessible tools, specifically WhatWeb, Nmap, Subfinder, and curl header inspection, to ensure methodological reproducibility. Commercial platforms such as Shodan and BuiltWith were

excluded as described in Section 3.3.1. It is possible that commercial tools, particularly Shodan with its continuously updated passive scan database, would produce a larger tool-only baseline than the one observed in this study. However, the structural argument that tools are limited to perimeter-visible signals would remain unchanged regardless of the tool suite used. Commercial tools observe the same externally accessible signals as open-source tools, and the application-layer and internal technologies revealed through human-generated OSINT would remain inaccessible to any scanning-based approach. Future work should empirically test whether the inclusion of commercial tools meaningfully reduces the coverage gap.

6.7 Ethical Considerations

All data collected in this study was obtained from publicly accessible sources through non-intrusive methods, as documented in Chapters 3 and 4. No authentication bypass, access-controlled probing, or exploitation of any kind was performed. The ethical considerations discussed here relate to the implications of the research findings rather than to the data collection process itself.

The most significant ethical dimension of this research is the dual-use nature of OSINT-enhanced technology fingerprinting as a capability. The methodology developed in this thesis can be applied proactively by security teams to assess their own organizations' inadvertent technology disclosures, and it can equally be applied by adversaries to gather intelligence about target organizations. This study does not introduce a novel attack capability; the OSINT sources it draws on are publicly accessible to any internet user. It does, however, demonstrate that structured aggregation of these sources produces substantially richer technology intelligence than either source yields individually, which has implications for how organizations think about their public information disclosure posture.

The aggregation of employee LinkedIn profile data raises individual privacy considerations even when profiles are publicly accessible. This study addresses this concern by using LinkedIn signal data only in aggregated form at the organizational level, anonymizing all employee-level identifiers in the collection pipeline, and reporting no individual employee data in any results. The LinkedIn collection methodology is designed to extract organizational-level technology intelligence rather than individual employee intelligence.

The findings of this study suggest that organizations may be disclosing substantially more about their internal technology stacks than they intend through routine activities such as hiring, employee networking, and open-source contribution. This finding has security awareness implications: organizations that wish to reduce their publicly inferable technology surface should consider the cumulative disclosure effect of job posting content, employee skill listings, and public repository metadata, not only their perimeter-facing network exposure [7, 8].

6.8 Practical Implications

The findings of this study have direct implications for three practitioner communities in cybersecurity.

For security professionals conducting attack surface assessments and penetration testing, the results demonstrate that tool-only technology fingerprinting produces a substantially

incomplete inventory of the technologies an organization uses and potentially exposes. A reconnaissance workflow that incorporates structured collection from job postings, LinkedIn profiles, and GitHub repositories alongside active scanning produces a substantially larger technology inventory than the tool-only baseline, as reflected by the 94.1% average OSINT coverage gain reported in Chapter 5. Practitioners should treat human-generated OSINT as a standard component of technology fingerprinting workflows rather than an optional supplement.

For blue team defenders and security architects, the findings highlight a category of organizational risk that is structurally different from conventional network exposure. An organization's perimeter defenses may be strong while its publicly inferable technology surface discloses significant intelligence about internal systems through job postings, employee profiles, and public repositories. Several defensive mechanisms follow directly from the study findings. First, organizations should audit their job posting content to avoid unnecessarily specific technology disclosures in public role descriptions, particularly for internal tooling and proprietary platform names that provide no recruiting advantage but do increase OSINT-based exposure. Second, organizations should establish GitHub repository disclosure policies that distinguish between public-facing projects and internal infrastructure configurations, preventing CI/CD pipeline definitions and infrastructure manifests from being inadvertently published. Third, security awareness programs should educate technical employees about the cumulative disclosure risk of detailed LinkedIn skill listings, particularly for technologies used in sensitive internal systems. Fourth, organizations can apply the methodology developed in this thesis proactively by running the OSINT collection pipeline against their own public presence periodically, effectively auditing their own inadvertent technology disclosures before adversaries do.

For threat intelligence analysts, the source contribution findings from Chapter 5 indicate that GitHub is the single largest contributor to OSINT-only technology detections at 45.83% of source-level contributions, followed by LinkedIn at 36.50% and job postings at 17.67%. This ordering suggests that public repository analysis should be prioritized in technology intelligence workflows. Job posting analysis, despite its lower volume contribution, provides the most strongly corroborated signals in this dataset, as reflected by its highest source-strength value of 0.6467. The most effective intelligence strategy combines all three source types to maximize both volume and corroboration.

6.9 Summary of Discussion

This chapter has interpreted the empirical results of the study across four dimensions: the meaning of the findings for the two research questions, the broader theoretical contribution of the structural complementarity finding, the validity and limitations of the methodological choices, and the practical implications for cybersecurity practice.

The near-zero average Jaccard similarity of 0.005 and the 94.1% average OSINT coverage gain together provide a consistent and interpretable answer to RQ1: human-generated OSINT reveals a substantially larger set of publicly inferable technology detections than active scanning tools produce, and the two signal classes are structurally complementary rather than redundant. The three integration methods evaluated in response to RQ2 each provide a valid and useful approach to combining these signal classes, differing in output structure and analytical purpose rather than in fundamental effectiveness. The low Cross-Validated count in

Method 3 is the most striking single result of the study and serves not as evidence of methodological failure but as the strongest available confirmation of the structural complementarity finding that constitutes the central theoretical contribution of this thesis.

Chapter 7 Conclusion and Future Work

7.1 Summary of the Study

This thesis investigated the extent to which human-generated Open-Source Intelligence can enhance technology fingerprinting for attack surface analysis beyond what active scanning tools alone can achieve. The study was motivated by a structural limitation inherent to all active scanning tools: they can only detect technologies that produce externally observable signals at the network perimeter, leaving the application layer, internal development infrastructure, and backend technology stack systematically undetected. To address this limitation, a research pipeline was designed and implemented to collect, normalize, and integrate technology signals from three human-generated OSINT sources, specifically job postings, LinkedIn employee profiles, and GitHub public repositories, alongside a tool-only baseline produced by WhatWeb, Nmap, Subfinder, and curl header inspection. The pipeline was applied across a purposively selected sample of twelve organizations spanning diverse industry verticals, and three integration methods of increasing analytical sophistication were designed and evaluated against an empirically constructed reference inventory.

7.2 Summary of Key Findings

RQ1: Human-generated OSINT sources reveal a substantially larger set of publicly inferable technology detections than active scanning tools produce, and the two signal classes are largely non-overlapping. The average Jaccard similarity between tool-detected and OSINT-detected technology sets across the twelve studied organizations is 0.005, and is exactly zero for ten of the twelve. OSINT contributed 857 organization-level detections absent from the tool-only baseline, representing an average OSINT coverage gain of 94.1%. Active scanning alone captured an average of 5.9% of the combined publicly detectable technology inventory. The dominant OSINT-only categories are programming languages at 34.66% and application frameworks at 17.74%, confirming that OSINT reaches the application and development layers that tools structurally cannot observe. These findings confirm that human-generated OSINT and active scanning are structurally complementary rather than redundant, and that technology fingerprinting conducted through tools alone produces a substantially incomplete picture of organizational technology exposure.

RQ2: All three integration methods effectively combined human-generated OSINT with tool-based signals to produce technology fingerprints substantially richer than the tool-only baseline, each serving a distinct analytical purpose. Method 1 maximized recall through union aggregation, producing integration coverage gains ranging from 53.85% to 96.35% across the twelve organizations. Method 2 provided empirical prioritization through source-strength weighting, identifying 173 multi-source technologies and ranking detections by the cumulative strength of their cross-source corroboration. Method 3 provided interpretable rule-based evidence classification without arbitrary numeric scoring, distributing 1,331 technology classifications across four evidence tiers and confirming through the near-zero Cross-Validated count that tool and OSINT signals operate at structurally different layers of the organizational technology stack. Together, the three methods demonstrate that structured integration of human-generated OSINT with active scanning consistently produces more complete and more analytically useful technology fingerprints than either signal class alone.

7.3 Contributions of the Research

This thesis makes three contributions to the fields of cybersecurity reconnaissance and OSINT-based security analysis.

The first contribution is an empirical quantification of the technology coverage gap between tool-only active scanning and human-generated OSINT across a diverse sample of twelve real organizations. The near-zero average Jaccard similarity of 0.005 and the 94.1% average OSINT coverage gain provide a concrete, data-driven measurement of a gap that prior literature had identified theoretically but not empirically quantified in this specific tool-versus-human-generated-OSINT setting.

The second contribution is the design and comparative evaluation of three integration methodologies for combining tool-based and human-generated OSINT technology signals. The evaluation demonstrates that each method serves a legitimate and distinct analytical purpose, and that the evidence-tier model provides a fully transparent, rule-based approach to technology fingerprint organization that requires no manually assigned numeric parameters and is directly defensible against empirical challenge.

The third contribution is a replicable data collection, normalization, and integration pipeline for human-generated OSINT in the technology fingerprinting domain. The pipeline's collection modules for job postings, LinkedIn profiles, and GitHub repositories, its normalization schema, its empirical source profile computation, and its three integration implementations are documented with sufficient specificity to enable replication and extension by future researchers.

7.4 Future Work

Five directions for future research are identified based on the findings and limitations of this study.

Taxonomy expansion: The Other category constituted 26.60% of OSINT-only detections in this study, indicating that the thirteen-category taxonomy, while useful for structured comparison, is not fully expressive enough to capture the diversity of technology signals present in human-generated OSINT. Future work should develop a more granular and comprehensive taxonomy through systematic analysis of the signal types currently classified as Other, potentially drawing on established technology classification frameworks from software engineering and IT asset management domains.

Scalable and ethical LinkedIn collection: The LinkedIn collection methodology in this study relied on manual PDF export due to platform access constraints, which limits collection scale and introduces potential sampling bias. Future work should explore ethical and compliant approaches to larger-scale LinkedIn signal collection, including the use of officially sanctioned data partnerships, user-consented data exports, or alternative professional networking platforms with more accessible data interfaces. Access to organizational HR or employee directory data: This study relied on publicly accessible LinkedIn profiles as its primary employee-level technology signal source. If a security team or researcher had access to internal HR databases or authenticated employee directory systems, the volume and accuracy of employee-level technology signals would likely increase substantially. Future work should

investigate how the coverage gap and source contribution findings change when authenticated employee data is available, and whether access to such data would meaningfully alter the structural complementarity observed between tools and OSINT in this study

Larger and more diverse organizational sample: This study was conducted across twelve purposively selected organizations and is explicitly framed as exploratory rather than statistically representative. Future work should extend the methodology to a larger and more systematically sampled organizational population to assess the generalizability of the structural complementarity finding and to quantify variation in OSINT coverage gain across industry verticals, organizational sizes, and geographic regions.

Longitudinal and real-time OSINT monitoring: The OSINT signals collected in this study represent a point-in-time snapshot. Job postings are removed when positions are filled, LinkedIn profiles change as employees move roles, and GitHub repositories evolve continuously. Future work should explore longitudinal collection designs that track changes in OSINT-detectable technology signals over time, enabling the study of temporal dynamics in organizational technology disclosure and providing a foundation for real-time technology intelligence monitoring pipelines.

Practitioner-validated reference inventories: The reference inventory used in this study was constructed from consolidated public evidence rather than internal company-verified technology documentation. Future work should investigate the feasibility of incorporating practitioner-validated or organization-verified reference inventories into the evaluation framework, either through research partnerships with participating organizations or through the use of publicly available technology disclosure reports. Such inventories would provide a stronger evaluation baseline and enable more precise measurement of the coverage gap between publicly inferable and internally known technology stacks.

7.5 Closing Remarks

This thesis set out to ask whether human-generated OSINT could serve as a meaningful complement to active scanning in technology fingerprinting, and the findings provide a clear affirmative answer. The structural complementarity of the two signal classes, confirmed consistently across twelve diverse organizations and three analytical methods, points to a fundamental gap in conventional reconnaissance practice: the technologies that organizations actually use are largely disclosed through the routine activities of hiring, employee networking, and open-source development, and they are largely invisible to the perimeter-level scanning tools that security practitioners currently rely on for technology fingerprinting. Closing this gap requires not a better scanner but a different kind of intelligence, one that looks at what organizations say about themselves rather than only at what their networks reveal. This thesis provides an empirical foundation and a replicable methodology for doing exactly that.

References

[1] M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, "A comprehensive survey of recent internet measurement techniques for cyber security," *Computers and Security*, vol. 128, p. 103123, May 2023.

[2] T. O. Browne, M. Abedin, and M. J. M. Chowdhury, "A systematic review on research utilising artificial intelligence for open source intelligence (OSINT) applications," *International Journal of Information Security*, 2024.

[3] D. Makrushin, "Attack surface analysis and monitoring using open-source intelligence," Kaspersky, 2023. [Online]. Available: https://securelist.com

[4] S. Shafee, A. Bessani, and P. M. Ferreira, "Evaluation of LLM chatbots for OSINT-based cyber threat awareness," arXiv:2401.15127, 2024. [Online]. Available: https://arxiv.org/abs/2401.15127

[5] A. Mahboubi, M. Bahrami, M. Shukur, and H. Ibrahim, "Evolving techniques in cyber threat hunting: A systematic review," *Journal of Network and Computer Applications*, vol. 232, p. 103988, 2024.

[6] M. Herranen, "Analysis of e-skills found in job postings," B.S. thesis, Laurea University of Applied Sciences, Espoo, Finland, 2022.

[7] M. Duffy, X. Pan, and S. Wilson, "Information reconnaissance by accumulating public information data sources," *Open Access Library Journal*, vol. 11, no. 3, 2024.

[8] Z. Avrahami, M. Zwilling, and C. Hajaj, "Leveraging OSINT for advanced proactive cybersecurity: Strategies and solutions," IEEE Access, vol. 13, pp. 154229–154250, 2025.

[9] S. R. B. M. Kassim, S. Li, and B. Arief, "The use of public data and free tools in national CSIRTs' operational practices: A systematic literature review," arXiv:2306.07988, 2023. [Online]. Available: https://arxiv.org/abs/2306.07988

[10] M. S. Rahman, "The art of open source intelligence (OSINT): Addressing cybercrime, opportunities, and challenges," SSRN, May 2025. doi: 10.2139/ssrn.5281845.

[11] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

[12] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, Jan. 1997.

[13] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[14] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[15] H. B. Mitchell, *Data Fusion: Concepts and Ideas*, 2nd ed. Berlin, Germany: Springer, 2012.

[16] M. Bazzell, *Open Source Intelligence Techniques*, 9th ed. Self-published, 2022.

[17] European Union Agency for Cybersecurity (ENISA), "Threat intelligence and incident coordination," ENISA, Athens, Greece, Tech. Rep., 2023. [Online]. Available: https://www.enisa.europa.eu

[18] MITRE Corporation, "MITRE ATT&CK Framework," 2024. [Online]. Available: https://attack.mitre.org

[19] G. F. Lyon, *Nmap Network Scanning*. Nmap Project, 2009. [Online]. Available: https://nmap.org/book/

[20] A. MacFarlane, "WhatWeb: Next generation web scanner," 2024. [Online]. Available: https://www.morningstarsecurity.com/research/whatweb

[21] BuiltWith Pty Ltd, "BuiltWith technology profiler," 2024. [Online]. Available: https://builtwith.com

[22] GitHub, Inc., "GitHub REST API documentation," 2024. [Online]. Available: https://docs.github.com/en/rest

[23] LinkedIn Corporation, "LinkedIn public profiles," 2024. [Online]. Available: https://www.linkedin.com

[24] Greenhouse Software, Inc., "Greenhouse recruiting API documentation," 2024. [Online]. Available: https://developers.greenhouse.io

[25] P. K. Manadhata, D. K. Kaynar, and J. M. Wing, "A formal model for a system's attack surface," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-07-153, 2007. [Online]. Available: https://www.cs.cmu.edu/~wing/publications/ManadhataKaynarWing07.pdf

[26] ProjectDiscovery, "Subfinder: A subdomain discovery tool," 2024. [Online]. Available: https://github.com/projectdiscovery/subfinder

Appendix A: Data Collection and Normalization Reference

A.1: Target Organization Data Collection Reference

The twelve organizations selected for this study are documented in Table A.1. Public source availability was assessed prior to collection based on the presence of active job postings, public GitHub repositories, and LinkedIn profiles for technical employees.

Table A 1 Target Organization Data Collection Reference

Organization	Industry Vertical	Size	Careers Source	GitHub Handle	Source Availability
Stripe	Financial Technology	Large	stripe.com/jobs	github.com/stripe	High across all three OSINT sources
Shopify	E-Commerce	Large	shopify.com/careers	github.com/Shopify	High across all three OSINT sources
Datadog	Developer Infrastructure	Large	datadoghq.com/jobs	github.com/DataDog	High across all three OSINT sources
Sentry	Developer Infrastructure	Mid-size	sentry.io/careers	github.com/getsentry	High across all three OSINT sources
Coinbase	Financial Technology	Large	coinbase.com/careers	github.com/coinbase	High across all three OSINT sources
Figma	Design Tooling	Mid-size	figma.com/careers	github.com/figma	High across all three OSINT sources

Organization	Industry Vertical	Size	Careers Source	GitHub Handle	Source Availability
Instacart	E-Commerce / Logistics	Large	instacart.com/careers	github.com/instacart	Moderate — limited GitHub activity
Brex	Financial Technology	Mid-size	brex.com/careers	github.com/brexhq	High across all three OSINT sources
Scale AI	Artificial Intelligence	Mid-size	scale.com/careers	github.com/scaleapi	Moderate — selective GitHub presence
Asana	Enterprise Software	Mid-size	asana.com/jobs	github.com/Asana	Moderate across OSINT sources
Epic Systems	Healthcare Technology	Large	epic.com/careers	Limited public presence	Low — conservative public disclosure
One Medical	Healthcare Technology	Mid-size	onemedical.com/careers	Limited public presence	Low — minimal OSINT signals

A.2 Technology Category Taxonomy

All technology signals collected from tools and OSINT sources were normalized and mapped to one of thirteen categories prior to integration. Table A.2 defines each category with representative technologies.

Table A 2 Technology Category Taxonomy with Definitions

Category	Definition	Example Technologies
Language	Programming and scripting languages used in development	Python, Go, TypeScript, Java, Ruby, Rust

Category	Definition	Example Technologies
Framework	Application, web, and API development frameworks	React, Django, Spring Boot, FastAPI, Rails
Database	Relational and non-relational data storage systems	PostgreSQL, MongoDB, Redis, DynamoDB, Snowflake
Cloud	Cloud infrastructure platforms and managed services	AWS, GCP, Azure, Kubernetes, Heroku
CI/CD	Continuous integration and deployment tooling	GitHub Actions, CircleCI, ArgoCD, Jenkins, Buildkite
DevOps	Infrastructure automation and orchestration tooling	Terraform, Docker, Ansible, Helm, Pulumi
Infrastructure	Network, server, and edge infrastructure components	Nginx, HAProxy, Cloudflare, Envoy
Security	Security tooling, scanning, and compliance platforms	Snyk, Vault, SonarQube, Semgrep, Wiz
Data and Analytics	Data processing, warehousing, and analytics platforms	Apache Spark, Snowflake, Kafka, Airflow, dbt
API and Integration	API gateways, messaging, and third-party integrations	GraphQL, gRPC, Kafka, Twilio, Stripe API
Monitoring and Observability	System monitoring, alerting, and tracing tools	Datadog, Prometheus, Grafana, Sentry, PagerDuty
Identity and Authentication	Identity management and authentication services	Okta, Auth0, LDAP, SAML, SSO
Other	Technology signals not assignable to named categories	Varies

A.3 Normalization Dictionary - Representative Mappings

Technology signals collected from heterogeneous sources frequently appear under variant names. A standardized name dictionary was applied during the normalization phase to map all observed variants to a single canonical form. Table A.3 presents representative mappings across common variant types.

Table A 3 Representative Technology Name Normalization Mappings

Raw Signal Variant	Canonical Name	Category
js, JavaScript, javascript	JavaScript	Language
nodejs, node.js, Node, Node JS	Node.js	Framework
ts, TypeScript, typescript	TypeScript	Language
postgres, postgresql, Postgres, pg	PostgreSQL	Database
mongo, MongoDB, mongodb	MongoDB	Database
k8s, Kubernetes, kubernetes	Kubernetes	Cloud
AWS, Amazon Web Services, amazon-aws	AWS	Cloud
GCP, Google Cloud, google-cloud-platform	Google Cloud	Cloud
gh-actions, GitHub Actions, github-actions	GitHub Actions	CI/CD
tf, terraform, Terraform	Terraform	DevOps
docker, Docker, dockerfile	Docker	DevOps
py, python, Python 3, Python 3.9	Python	Language
react, ReactJS, React.js, react.js	React	Framework
go, golang, Go Language	Go	Language
redis, Redis, redis-cache	Redis	Database
elastic, elasticsearch, ElasticSearch	Elasticsearch	Database

Version-specific references such as Python 3.9 and Java 11 were normalized to their base technology name to enable cross-source comparison. All mappings were applied consistently across all source types and all twelve organizations.

A.4 Sample Raw-to-Normalized Signals

This section illustrates how raw technology signals collected from each source type were processed through the normalization pipeline to produce structured entries in the master signals file.

Job Posting Example:

Raw extracted text from a Datadog engineering role posting: *"We are looking for engineers with experience in Go, Python, Kubernetes, and AWS. Familiarity with Terraform and Datadog observability tooling is a plus."*

Extracted signals after parsing and normalization:

Tech	Category	Source	Organization
Go	Language	jobs	Datadog
Python	Language	jobs	Datadog
Kubernetes	Cloud	jobs	Datadog
AWS	Cloud	jobs	Datadog
Terraform	DevOps	jobs	Datadog

GitHub Repository Example:

Raw dependency signals extracted from a package.json file in a public Stripe repository: dependencies listed as react, typescript, jest, webpack.

Normalized signals:

Tech	Category	Source	Organization
React	Framework	github	Stripe
TypeScript	Language	github	Stripe
Jest	CI/CD	github	Stripe
Webpack	Framework	github	Stripe

LinkedIn Profile Example:

Raw skill endorsements from a technical employee profile at Coinbase: Python, AWS, Kubernetes, dbt, Snowflake, PostgreSQL.

Normalized signals:

Tech	Category	Source	Organization
Python	Language	linkedin	Coinbase

Tech	Category	Source	Organization
AWS	Cloud	linkedin	Coinbase
Kubernetes	Cloud	linkedin	Coinbase
dbt	Data and Analytics	linkedin	Coinbase
Snowflake	Database	linkedin	Coinbase
PostgreSQL	Database	linkedin	Coinbase

Tool-Detected Example:

Raw WhatWeb output for a Shopify web property: detected signals including nginx, Ruby on Rails HTTP header signature, Cloudflare CDN.

Normalized signals:

Tech	Category	Source	Organization
Nginx	Infrastructure	tools	Shopify
Ruby on Rails	Framework	tools	Shopify
Cloudflare	Infrastructure	tools	Shopify

Appendix B: Detailed Empirical Results

B.1 Full RQ1 Organization-Level Comparison Results

Table B.1 presents the complete per-organization RQ1 comparison results. Tool completeness and OSINT coverage gain do not sum to 100% because overlapping technologies are excluded from both measures. All tool counts reflect the filtered signal set used in the RQ1 comparison pipeline, which excludes security header, subdomain, and TLS metadata signals.

Table B 1 Complete RQ1 Tool vs OSINT Comparison by Organization

Organization	Tool Techs	OSINT Techs	Combined	Common	Tool-Only	OSINT-Only	Tool Completeness	OSINT Coverage Gain	Jaccard
Asana	3	45	48	0	3	45	6.25%	93.75%	0.0000
Brex	1	73	74	0	1	73	1.35%	98.65%	0.0000
Coinbase	2	77	79	0	2	77	2.53%	97.47%	0.0000
Datadog	3	131	134	0	3	131	2.24%	97.76%	0.0000
Epic Systems	3	46	49	0	3	46	6.12%	93.88%	0.0000
Figma	2	52	54	0	2	52	3.70%	96.30%	0.0000
Instacart	3	70	73	0	3	70	4.11%	95.89%	0.0000
One Medical	3	7	10	0	3	7	30.00%	70.00%	0.0000
Scale AI	5	71	72	4	1	67	6.94%	93.06%	0.0556
Sentry	3	105	108	0	3	105	2.78%	97.22%	0.0000
Shopify	2	79	81	0	2	79	2.47%	97.53%	0.0000
Stripe	2	106	107	1	1	105	1.87%	98.13%	0.0093
Average							**5.90%**	**94.10%**	**0.0050**
Total	**32**	**862**	**890**	**5**	**27**	**857**			

B.2 OSINT Source-Level Contribution

Table B.2 presents the source-level contribution of each OSINT source type to OSINT-only technology detections. Because a technology detected by multiple OSINT sources contributes to each source's count independently, the total source-level contributions of 1,126 exceeds the 857 unique organization-level OSINT-only detections.

Table B 2 OSINT Source Contribution to OSINT-Only Detections

Source	Source-Level Contributions	Percentage of 1,126	Unique OSINT-Only Share
GitHub	516	45.83%	Largest volume contributor
LinkedIn	411	36.50%	Broadest employee-level coverage
Job Postings	199	17.67%	Highest corroboration rate
Total	**1,126**	**100%**	

B.3 Source Profile Computation Details

Table B.3 presents the complete source profile values computed from the full normalized dataset prior to integration. Source strength is the arithmetic mean of agreement rate and category coverage and is used only as a weighting factor in Method 2.

Table B 3 Complete Source Profile Computation Results

Source	Total Signals	Agreed Signals	Agreement Rate	Categories Covered	Category Coverage	Source Strength
Job Postings	203	157	0.7734	7 of 13	0.5385	0.6560
LinkedIn	414	154	0.3720	4 of 13	0.3077	0.3398
GitHub	850	166	0.1953	5 of 13	0.3846	0.2899
Tools	110	7	0.0636	6 of 13	0.4615	0.2536

Source strength formula: Source Strength = (Agreement Rate + Category Coverage) / 2

Agreed signals are those detected by at least one other independent source for the same organization.

B.4 Integration Method Results by Organization

Table B.4 presents a unified per-organization comparison across all three integration methods. Each method's output count reflects its own unit of analysis and the totals should not be interpreted as directly interchangeable across methods.

Table B 4 Integration Method Comparison by Organization

Organization	M1: Tool-Only	M1: Integrated	M1: Coverage Gain	M2: Total Records	M2: Multi-Source	M2: Avg Score	M3: Total	M3: CV	M3: Strong	M3: Moderate	M3: Weak
Asana	9	61	85.25%	61	11	0.4196	61	0	11	50	0
Brex	4	87	95.40%	87	12	0.4109	91	0	12	44	35
Coinbase	6	117	94.87%	117	17	0.3900	121	0	17	79	25
Datadog	7	192	96.35%	192	16	0.3251	200	0	16	143	41
Epic Systems	13	61	78.69%	61	9	0.3919	62	0	9	26	27
Figma	10	77	87.01%	77	11	0.3529	80	0	11	63	6
Instacart	10	113	91.15%	113	16	0.3773	114	0	16	67	31
One Medical	6	13	53.85%	13	0	0.1790	13	0	0	13	0
Scale AI	14	93	84.95%	93	19	0.4288	95	4	15	58	18
Sentry	9	212	95.75%	212	26	0.3171	218	0	26	171	21
Shopify	11	113	90.27%	113	14	0.3507	115	0	14	67	34

Organization	M1: Tool-Only	M1: Integrated	M1: Coverage Gain	M2: Total Records	M2: Multi-Source	M2: Avg Score	M3: Total	M3: CV	M3: Strong	M3: Moderate	M3: Weak
Stripe	11	158	93.04%	158	23	0.3793	161	1	22	93	45
Total	**110**	**1,297**		**1,330**	**173**		**1,331**	**5**	**169**	**874**	**283**

M1 = Method 1 Union-Based Integration. M2 = Method 2 Source-Strength Weighted Aggregation. M3 = Method 3 Evidence-Tier Classification. CV = Cross-Validated. M1 coverage gain = new OSINT technologies / total integrated count. Method totals reflect different units of analysis and are not directly interchangeable.

B.5 Full Method 3 Evidence-Tier Distribution

Table B.5 presents the complete Method 3 evidence-tier classification results including CV ratio and Strong OSINT ratio per organization, sorted by Strong OSINT ratio to illustrate variation across the sample.

Table B 5 Method 3 Evidence-Tier Distribution with Ratios

Organization	Total	Cross-Validated	Strong OSINT	Moderate	Weak	CV Ratio	Strong Ratio
Asana	61	0	11	50	0	0.000	0.180
Scale AI	95	4	15	58	18	0.042	0.158
Epic Systems	62	0	9	26	27	0.000	0.145
Coinbase	121	0	17	79	25	0.000	0.140
Instacart	114	0	16	67	31	0.000	0.140
Stripe	161	1	22	93	45	0.006	0.137
Figma	80	0	11	63	6	0.000	0.138
Brex	91	0	12	44	35	0.000	0.132
Shopify	115	0	14	67	34	0.000	0.122

Organization	Total	Cross-Validated	Strong OSINT	Moderate	Weak	CV Ratio	Strong Ratio
Sentry	218	0	26	171	21	0.000	0.119
Datadog	200	0	16	143	41	0.000	0.080
One Medical	13	0	0	13	0	0.000	0.000
Total	**1,331**	**5**	**169**	**874**	**283**		

CV Ratio = Cross-Validated count / Total. Strong Ratio = Strong OSINT Evidence count / Total. Organizations sorted by Strong OSINT ratio descending. Only Scale AI and Stripe achieved Cross-Validated status, with 4 and 1 technology respectively.