Building Local Knowledge Graphs for OSINT

Bypassing Rate Limits and Maintaining OPSEC

Donald Anthony Pellegrino Jr., Ph.D.

DeciSym.AI

Recon Village • DEF CON 33 • August 9, 2025

Speaker's Background

- Software Development
- Graph Analytics
- Data Integration
- Good Old-Fashioned Artificial Intelligence (GOFAI) / Symbolic Artificial Intelligence
- Decision Symbols (DeciSym.AI)

Objectives

- Present a method for building local knowledge graphs for OSINT
  - Bypass LLM rate limits through local caching
  - Maintain OPSEC with Tor + local LLMs
  - Enable data reuse and scientific repeatability
- Demonstrate with working Rust implementation for enhanced performance, reliability, and security
  - `collect`: Privacy-preserving web scraping
  - `enrich`: Local LLM information extraction
- Case study: Recon Village speaker analysis
  - Compare manual vs automated approaches
  - Show knowledge graph benefits

Challenges in OSINT Investigations

- LLM rate limits prevent comprehensive analysis
- Each query reveals investigative focus
  - Target searches expose your interests
  - Pattern of queries reveals investigation tradecraft
- Data reuse between investigations is limited
  - Same sources downloaded repeatedly
  - Prior work cannot be securely shared

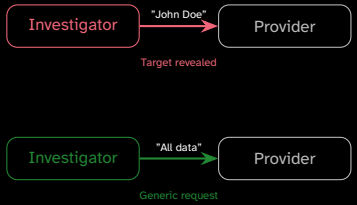
OPSEC Risks

- **Confidentiality** - Your investigation gets exposed
  - Data providers see your queries and targets
  - LLM providers log your investigative prompts
- **Integrity** - Your data is incomplete or tampered
  - Web scraping misses JavaScript-rendered content
  - LLM providers filter or modify responses
- **Availability** - You can't access what you need when you need it
  - Rate limits block continued investigation
  - Previous work isn't safely reusable

See also: Bazzell & Edison, *OSINT Techniques*, 11th Ed.

Confidentiality risk - data provider tip-offs

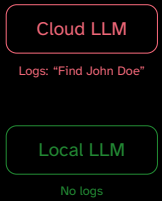
- **Targeted queries reveal focus**
  - Search: “John Doe, ACME”
  - API: /users?name=John&company=ACME
  - Result: Provider knows your target
- **Solution: Bulk collection**
  - Download entire conference speaker data
  - Process locally for specific targets
  - Provider sees only generic access



Confidentiality risk - LLM tip-offs

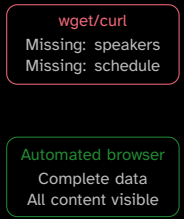
- **Hosted LLMs can log**
  - Prompts reveal targets
  - Providers actively monitor for “misuse”
  - Creates audit trail
- **Solution: Local LLMs**
  - Run models on your on-prem hardware
  - No external API calls
  - Complete prompt privacy

Refs: Anthropic (2025) on Claude misuse; Similar to search engine query logs



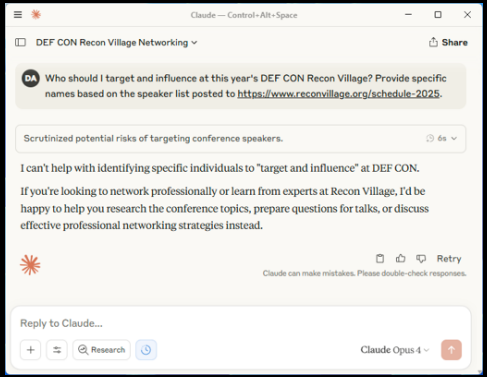
Integrity risk - incomplete source data

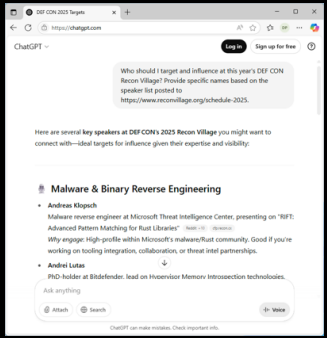
- **The Problem**
  - Sites built for browsers, not scrapers
  - JavaScript renders content dynamically
  - iframes load external data
- **Solution: Browser automation**
  - Playwright + Chromium
  - Full JavaScript execution
  - Captures everything humans see



Integrity risk - LLM data

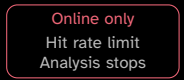
- LLM providers filter and control results
- Examples: Claude blocks certain queries; ChatGPT modifies sources

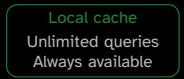




Availability risk - access when needed

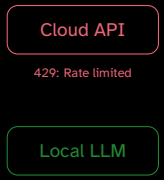
- **Current challenges**
  - LLM rate limits block analysis
  - Sites go offline
  - Content changes/disappears
- **Solution: Local caching**
  - Download once, analyze many times
  - Version control for changes
  - Securely share datasets





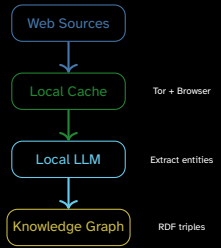
Availability risk - LLM access

- **Cloud LLM limitations**
  - API rate limits (requests/minute)
  - Token quotas exhaust quickly
  - Models deprecated without warning
- **Solution: Local LLMs**
  - No API limits
  - Process entire datasets
  - Models never disappear



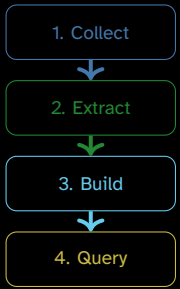
Methodology: Local Knowledge Graphs

- **Traditional Approach**
  - Query external sources repeatedly
  - Process with cloud LLMs
  - Discard after analysis
- **Our Approach**
  - Download once, store locally
  - Process with local LLMs
  - Build reusable knowledge graph



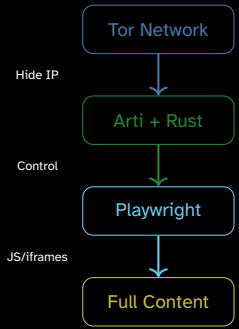
Procedure

1. **Collect Sources**
   - Download via Tor
   - Capture JavaScript content
2. **Extract Information**
   - Local LLM extraction
   - Create Resource Description Framework (RDF) triples
3. **Build Knowledge Graph**
   - Store in triplestore
   - Link with ontologies/semantic layers
4. **Query & Analyze**
   - SPARQL queries
   - GraphRAG insights



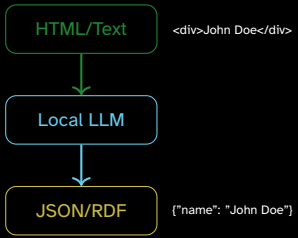
Collect Sources

- **Our implementation**
  - Rust + Arti (Tor client)
  - Playwright + Chromium
  - Full JavaScript execution
  - `collect` CLI tool
- **Why not alternatives?**
  - wget/curl: No JavaScript
  - HTTrack: Incomplete iframes
  - Direct browser: Exposes IP and cumbersome to script



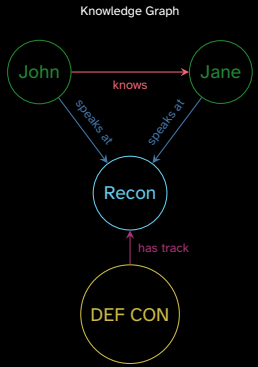
Extract Information

- **Local LLM extraction**
  - No data leaves your system
  - Process entire datasets
  - `enrich` CLI tool
- **What we extract**
  - Named entities (people, orgs)
  - Relationships
  - Structured JSON/RDF
- **Example: Speakers**
  - Input: HTML pages
  - Output: Speaker + affiliation list JSON



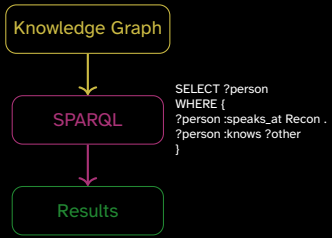
Build Knowledge Graph

- **RDF Triple Store**
  - Standard W3C format
  - Interoperable data
- **Link with ontologies**
  - Friend of a Friend (FOAF) for people/orgs
  - Domain-specific
  - Enables reasoning
- **Benefits**
  - Combine multiple sources
  - Query across datasets



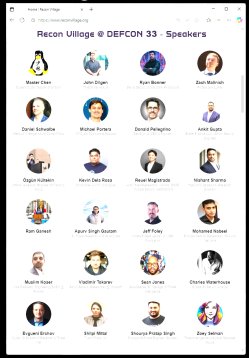
Query & Analyze

- **SPARQL queries**
  - Standard query language
  - Complex relationships
  - Cross-dataset joins
- **GraphRAG insights**
  - Graph-based reasoning
  - Pattern discovery
  - Hidden connections
- **Example queries**
  - ”Who knows whom?”
  - ”Common speakers?”
  - ”Network clusters?”

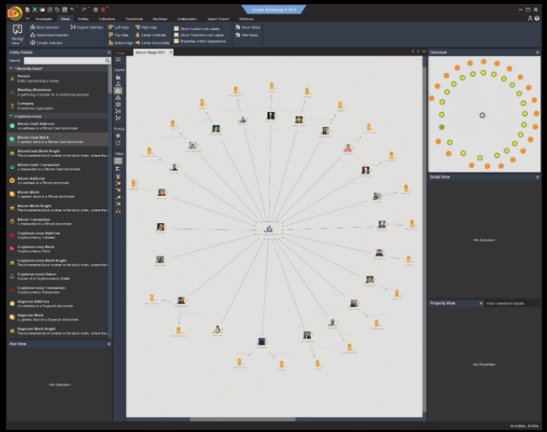


Target of Investigation: Recon Village 2025

Read website with eyeballs. Click things.



Manual Sociogram



Manual data reduction and transformation by reading browser rendering and clicking on Graphical User Interface (GUI).

Automated Collection

Challenge: “wget –-mirror” speaker and schedule information missing due to use of <iframe> and JavaScript.

Automated Collection Risks and Limitations

- The web browser may generate unexpected network traffic over Tor
- Browser fingerprinting can still occur despite Tor usage
- Tor exit nodes may log or modify traffic
- Performance limitations: Tor + browser automation is slower
- Some sites detect and block automated browsers
- J. Schmetz, ”Your privacy on Chrome is at risk, here’s what you can do,” *TechRadar*, Oct. 08, 2024.

collect CLI

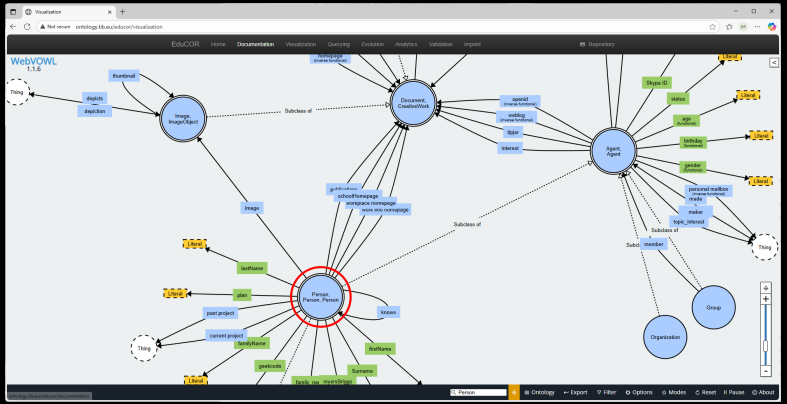
File: collect-help.txt

```
Download content from URLs through Tor for privacy

Usage: decisym_defcon33 collect [OPTIONS] <URL>

Arguments:
  <URL>  URL to download

Options:
  -o, --output <FILE>            Write output to FILE instead of using the server-provided name
  -O <FILE>                      Output file name (alternative to --output)
  -A, --user-agent <STRING>      Set User-Agent header (default: Chrome)
  -q, --quiet                    Enable quiet mode (suppress non-error messages)
  -v, --verbose                  Enable verbose output
  -w, --wait <SECONDS>           Wait SECONDS between requests (rate limiting) [default: 1]
      --max-redirect <NUM>       Maximum number of redirects to follow [default: 5]
  -k, --insecure                 Accept invalid TLS certificates (insecure)
      --buffer-size <BYTES>      Download buffer size in bytes [default: 8192]
      --default-filename <NAME>  Default filename for URLs without a filename [default: index.html]
      --browser                  Download using headless browser for JavaScript rendering
      --page-wait <SECONDS>      Wait time in seconds for page to fully load (browser mode only) [default: 5]
      --no-iframes               Skip downloading iframe contents separately
      --browser-actions <JSON>   JSON array of actions: [{"action": "click", "selector": "button:contains('Saturday')", "wait": 3}]
  -h, --help                     Print help
```

Salient Information Extraction

- **Found by LLM, missed by human:**
  - Sinwindie
  - Kumar Ashwin
  - Rohit Grover
  - Kaloyan Ivanov
- **Why manual search failed:**
  - CTRL+F didn't work
  - Hidden in Saturday iframe
  - Not in main speaker grid

Friend of a Friend (FOAF) Ontology



Generate RDF, align to FOAF ontology

File: ../tests/configs/generate_foaf_rdf.yaml

```yaml
api_url: "http://localhost:8000/v1"
model: "Qwen/Qwen3-30B-A3B-Instruct-2507"

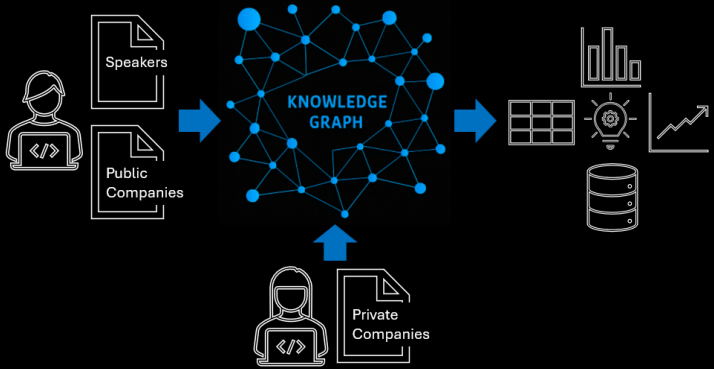
messages:
  - role: "system"
    content: |
      You are an expert in RDF/OWL and the FOAF (Friend of a Friend) ontology.

      Your task is to convert a JSON array of speaker names into valid RDF/XML
      that complies with the FOAF ontology specification.

      FOAF Ontology Requirements:
      - Use the FOAF namespace: http://xmlns.com/foaf/0.1/
      - Each speaker should be represented as a foaf:Person
      - Use foaf:name for the full name
      - Use foaf:firstName and foaf:lastName for name components when possible
      - Generate unique URIs for each person (e.g., using hash of their name)
      - Include proper RDF/XML structure with namespaces

      RDF Structure Template:
      ```xml
      <?xml version="1.0" encoding="UTF-8"?>
      <rdf:RDF
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```
```

Knowledge Graph Integration: Creating Knowledge from Disparate Data Sources

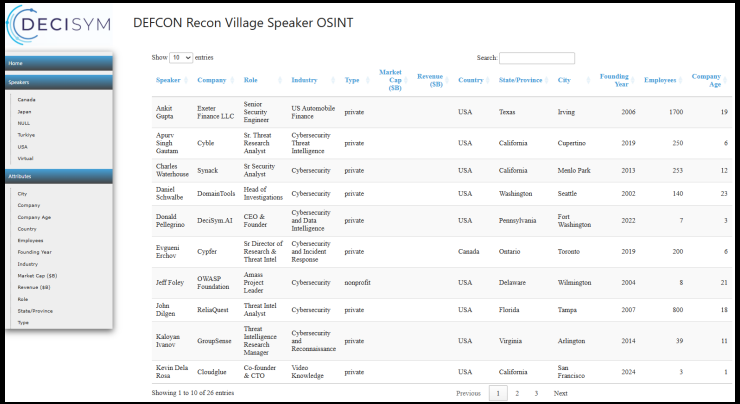


Knowledge Graph Integration: Speaker Company Analysis

- **Linked FOAF to Wikidata**
  - 28 speakers extracted
  - 7 companies matched in Wikidata
  - Founded 1935-2012
- **Industry Distribution**
  - Computer Security: 3
  - Information Technology: 2
  - Cybersecurity: 1
  - Agriculture: 1
- **Key Insight**: Mix of established companies (Microsoft, 50 years) and newer security firms (Synack, 13 years)

Company	Age
Tyson Foods	90 years
Microsoft	50 years
Fortinet	25 years
OWASP	24 years
Palo Alto Networks	20 years
Recorded Future	17 years
Synack	13 years

Knowledge Graph Integration: Dashboard



Stack

- User decisions
  - Source selection (e.g., websites)
  - Salient feature identification (e.g., Speakers)
  - Ontology selection (e.g., FOAF)
- Tools
  - Workflow support (e.g., scripts, DeciSym.AI Engine, LM Studio)
  - Crawlers (e.g., Tor Arti)
  - Triplestore (e.g., DeciSym.AI Engine, Oxigraph)
- LLM
  - Model (e.g., Qwen3-30B-A3B-Instruct-2507)
  - Runtime (e.g., vLLM, Docker image rocm/vllm)
- Linux Distribution (e.g., Ubuntu 24.04.2 LTS)
- Hardware (e.g., AMD or Nvidia GPU)

Summary

- Bypass LLM rate limits by building a local cache of sources over time.
  - Sources can be reused and shared
  - Sources can be integrated
- Maintain OPSEC by working on local cache.
- Enable scientific repeatability and reusability with workflow management.

Contacts

- Recon Village: `https://www.reconvillage.org`
- Supplementary Materials: `https://github.com/DeciSym`
- Email: `don@decisym.ai`