KRAKEN

Continuous Infrastructure Tracking For Adversary Intelligence Modeling

Technical Whitepaper | Malwarebox Research | Version 1.0
Author: Robin Dost
https://kraken.malwarebox.eu

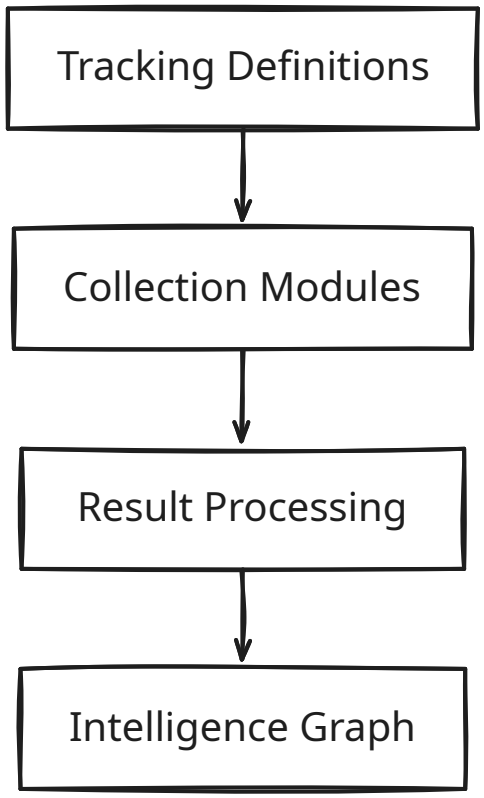
Abstract

Modern threat intelligence work depends on the continuous collection, processing and correlation of infrastructure signals. Most analysts work with a fragmented toolset: enrichment services, passive DNS databases, malware sandboxes and visualization platforms.
Kraken is an actor-centric threat intelligence platform that continuously tracks adversary infrastructure and models operational activity in a structured intelligence graph.
It was built to explore a different approach to threat research. Collection, analysis and correlation run in the same workflow.

By combining automated tracking modules, structured entity modeling and a graph-based representation of intelligence data, Kraken allows analysts to observe how adversary infrastructure evolves over time and how campaigns, malware and operational activity are related.

This paper describes the architecture, data model and workflow of the Kraken platform.

Conceptual Model



Introduction

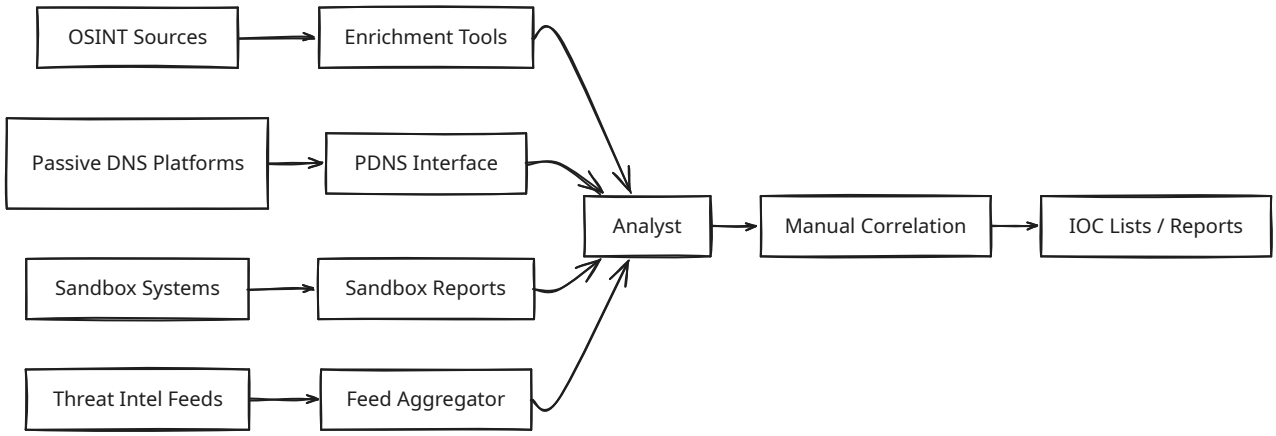
Threat intelligence operations today rely on a diverse ecosystem of tools and data sources. Threat intel work usually pulls data from many different places: passive DNS, sandbox results, OSINT sources, infrastructure monitoring and plain manual digging.

The problem is that these tools rarely work together. Data gets pulled in one place, enriched somewhere else, stored in another database and then inspected in a separate tool.

A lot of the time an analyst is wasted just moving data around. Exporting results, fixing formats and trying to correlate indicators that came from different systems.

Meanwhile adversary infrastructure keeps changing. Domains disappear, IPs move, hosting shifts and delivery infrastructure rotates quickly. Static intelligence reports capture only a snapshot and rarely reflect how campaigns evolve over time. As a result, analysts must continuously rediscover infrastructure patterns that could otherwise be tracked automatically.

Fragmented Intelligence Workflow



Kraken brings collection, processing, modeling and visualization into one platform. Kraken models infrastructure, campaigns and malware activity as a continuously evolving entity graph. Indicators are stored with their relationships, making infrastructure changes visible.

Design Philosophy

Kraken was built around a simple observation: most threat intelligence workflows are fragmented. Collection, enrichment, analysis and correlation are often handled by separate tools that do not share a consistent model of the data they produce.

Kraken takes a different approach. The platform models intelligence as an evolving network of entities and relationships. Infrastructure, malware, campaigns and threat actors are represented as interconnected elements within a continuously expanding intelligence graph.

Kraken does not store indicators as static datasets. Infrastructure signals are processed as part of a continuous pipeline. New signals are added to the intelligence graph where relationships between infrastructure, campaigns, malware and actors become visible over time.

Actor-Centric Intelligence Modeling

Traditional threat intelligence workflows generally revolve around indicators such as domains, IP addresses or file hashes. These indicators are useful, but they rarely reflect the operational structure behind an adversary.

Kraken therefore follows an actor-centric intelligence model.

It organizes intelligence around adversary activity. Infrastructure, malware, campaigns and operational artifacts are modeled as entities linked to the actors operating them.
This approach allows analysts to observe how different infrastructure nodes, malware samples and operational behaviors relate to the same threat actor over time.

As the intelligence graph evolves, Kraken gradually reconstructs the operational environment of an adversary. Infrastructure clusters, campaign overlaps and operational patterns emerge naturally as relationships between entities accumulate.

Kraken shifts the focus from indicators to actors. Indicators rotate constantly. Tracking actors is more stable.

Continuous Infrastructure Tracking

Adversary infrastructure is rarely static. Domains, hosting providers and command-and-control endpoints frequently rotate as campaigns evolve. Traditional workflows often capture infrastructure only at the moment it is discovered.

Kraken introduces tracking definitions that continuously monitor infrastructure and related entities. Collection modules periodically retrieve new signals which may reveal additional infrastructure nodes, campaign activity or operational changes.

Structured Intelligence Modeling

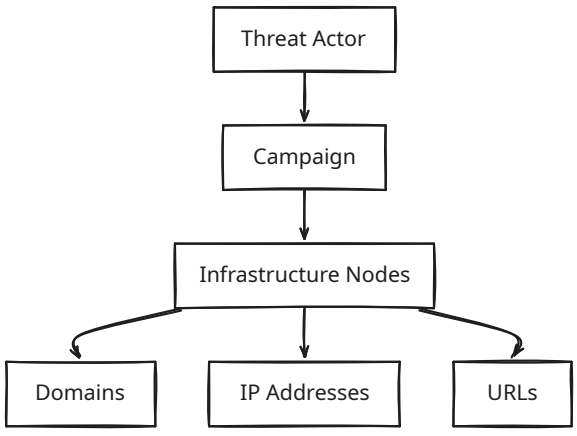
Raw signals alone **do not** constitute intelligence. Kraken takes observed signals and turns them into entities like domains, IPs, malware samples, campaigns and actors.

Relationships between these entities form the intelligence graph and show how indicators connect within adversary operations.

Modular Data Collection

Threat intelligence sources vary widely.
Kraken uses a modular architecture where collection logic is implemented as independent modules.

Modules may retrieve data from OSINT sources, passive DNS systems, blog platforms, sandbox environments or other intelligence services. New modules can easily be added without modifying the core of the platform.



Graph-Based Intelligence Representation

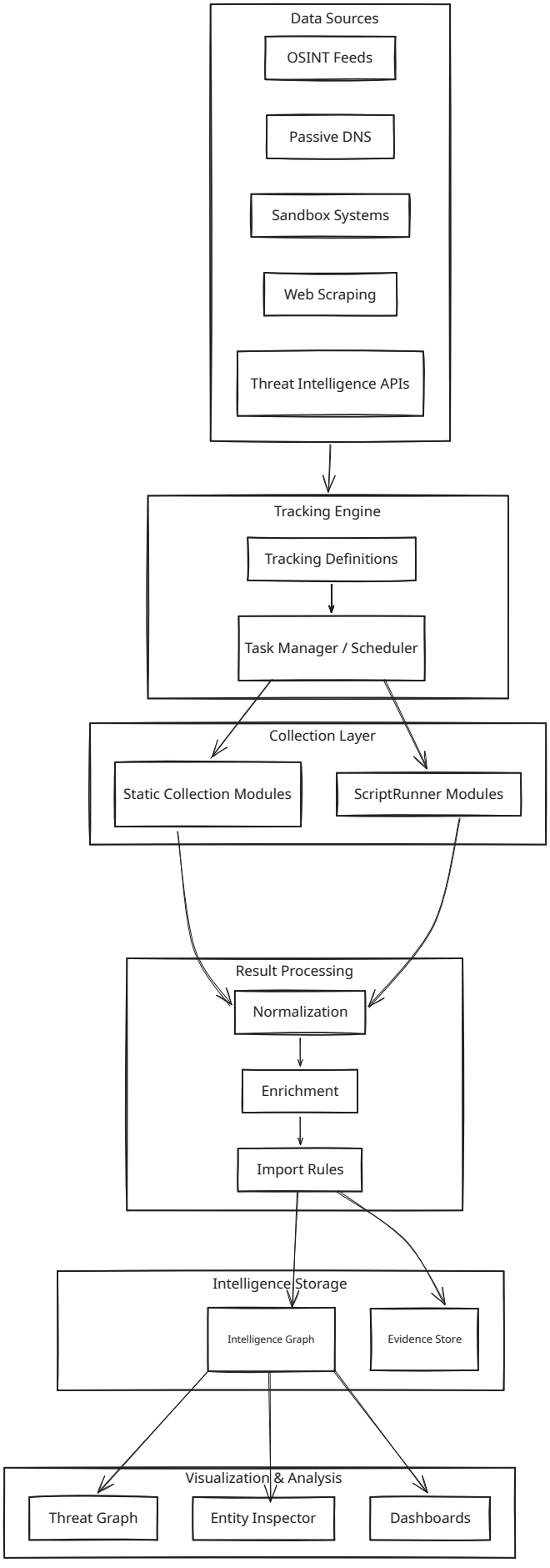
Kraken models intelligence as a graph of entities and relationships. Indicators are analyzed within their relational context rather than as flat lists. This

allows analysts to identify infrastructure clusters, campaign overlaps and operational patterns that would remain hidden in traditional datasets.

System Architecture

Kraken is structured as a pipeline that continuously transforms raw infrastructure signals into structured threat intelligence.

At the beginning of the pipeline are external data sources. These include OSINT feeds, passive DNS datasets, sandbox environments, web scraping targets and various threat intelligence APIs. Each of these sources may produce signals that indicate new infrastructure, campaign activity or malware distribution.



Tracking Definitions

Tracking definitions specify what should be monitored. They describe which entities or infrastructure elements are relevant and how frequently they should be checked. The task manager schedules and executes these tracking tasks, ensuring that collection runs continuously rather than as one-time enrichment operations.

Orchestration and Task Management

Continuous infrastructure tracking requires coordinated execution of multiple collection tasks.

Kraken uses an orchestration layer that schedules and runs tracking operations inside the platform.

The Task Manager evaluates tracking definitions, determines when tasks should be executed and dispatches the corresponding collection modules. A tracking task may execute several modules at the same time depending on the entity type and configuration.

This orchestration layer ensures that intelligence collection runs continuously rather than as isolated enrichment queries. Tasks may run at regular intervals, react to newly discovered infrastructure or follow predefined monitoring schedules.

By coordinating collection modules through a centralized scheduler, Kraken transforms intelligence gathering into an automated monitoring system capable of observing adversary infrastructure over extended periods of time

Modules

Collection itself is handled by modular components. Static collection modules implement predefined collectors for known intelligence sources while Script-Runner modules allow custom collection logic to be executed as isolated scripts. This design makes it possible to integrate new collection sources without changing the core system.

Results

Results generated by collection modules are processed by the result processing pipeline. Normalization helps to convert raw data into a structured internal format.
Enrichment can add additional context while import rules determine how extracted signals should be converted into entities and relationships.

Structured intelligence is stored in the intelligence graph, which represents connections between infrastructure nodes, campaigns, malware and threat actors. The associated evidence store preserves the original artifacts and signals from which these relationships were derived.

Analysts interact with this data through the analysis interfaces, including the threat graph visualization, the entity inspector and operational dashboards.

Together, these components form a system that continuously collects, processes and models adversary infrastructure.

Threat Entity Model

Threat intelligence data is often represented as flat collections of indicators. Domains, IP addresses and hashes are often enriched with metadata but rarely modeled as structured objects with clear relationships.

Kraken follows a different approach. Intelligence is represented as a graph of entities and relationships. Indicators are not stored as isolated records. Each entity represents part of adversary activity. Relationships describe how infrastructure, malware, campaigns and actors connect within an operation.

Infrastructure, malware samples, campaigns and threat actors exist in the same intelligence graph.

Entity Types

Entities represent observable elements within threat intelligence data.
Typical entity types include infrastructure components, malware artifacts and operational actors.

Examples of entity categories:

- Domain
- IP Address
- URL
- File / Hash
- Malware
- Campaign
- Threat Actor

Each entity type has identity attributes that identify the object in the graph.

Identity and Deduplication

A key aspect of the entity model is identity resolution.

When new signals are processed by the collection pipeline, Kraken attempts to determine whether the observed artifact already exists as an entity within the graph. If an entity with the same identity attributes already exists, the observation is linked to the existing entity instead of creating a duplicate.

This ensures that repeated observations strengthen existing intelligence rather than fragmenting the dataset into multiple independent indicators.

Relationships Between Entities

Entities usually don’t exist in isolation. Domains resolve to IP addresses, malware communicates with command and control infrastructure and campaigns operate specific infrastructure nodes.

Kraken represents these connections through explicit relationships between entities.

Examples include:

Domain → resolves_to → IP
URL → hosted_on → Domain
Malware → communicates_with → Infrastructure
Campaign → uses → Infrastructure
Threat Actor → operates → Campaign

These relationships form the structural backbone of the intelligence graph.

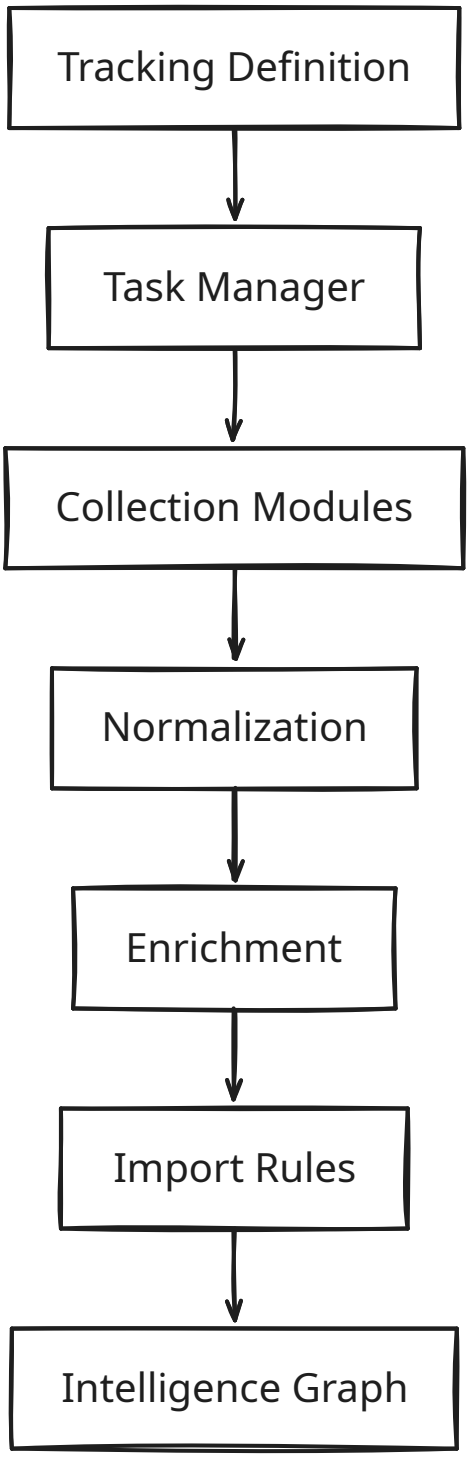
Evidence and Attribution

Every relationship in the intelligence graph is backed by evidence.

Evidence may originate from sandbox reports, passive DNS observations, OSINT sources or collection modules. By preserving the underlying evidence, Kraken allows analysts to trace how and why a relationship was established.

This approach provides transparency and supports validation of intelligence conclusions.

Tracking Pipeline



Infrastructure tracking in Kraken is organized as a continuous pipeline. Tracking operations run automatically based on monitoring definitions. A tracking definition defines what to monitor and how to run the monitoring.

Tracking definitions are important, because they specify what should be monitored and how the monitoring is performed. In practice this usually means selecting one or more entities and assigning collection modules that are capable of producing additional intelligence signals for those entities.

Once a definition is created, the task manager schedules tracking tasks at regular intervals. A task runs one or more collection modules to retrieve signals for the tracked entity.

These signals may include newly observed domains, additional IP addresses, updated infrastructure endpoints or other operational artifacts. The collection layer does not directly modify the intelligence graph. Instead, all results pass through the result processing pipeline where the data is normalized and evaluated by import rules.

Import rules determine how extracted signals should be interpreted. Some signals create new entities. Others extend existing infrastructure nodes or add new relationships.

Because tracking tasks are executed repeatedly, the intelligence graph gradually expands as new observations are collected. Infrastructure that was previously unknown may become visible after several tracking cycles, revealing campaign structure or infrastructure rotation patterns.

This turns intelligence collection into continuous monitoring instead of a one-time enrichment task. As a result, infrastructure changes can be observed over time and correlated with existing campaign or actor entities.

Import Rules and Signal Interpretation

Raw signals collected from external sources rarely represent intelligence in their original form. Infrastructure observations should be interpreted and mapped into structured entities before they become part of the intelligence graph.

Kraken performs this transformation using Import Rules.

Import Rules decide what happens with collected signals, a rule can create a new entity, update or link two entities if needed.

Example: passive DNS shows that a domain resolves to an IP

```
Domain → resolves_to → IP Address
```

More complex rules may extract multiple entities from a single signal or associate evidence with existing infrastructure nodes.

This rule based interpretation layer separates signal extraction from intelligence modeling. Collection modules focus exclusively on retrieving data while Import Rules define how that data becomes structured intelligence.

The core platform stays unchanged when new intelligence sources are added.

Module System & Script-Runner

Intelligence collection in Kraken is performed through modules.
Each module implements a specific method of retrieving or extracting signals from external sources.

This approach separates collection logic from the core platform.
Kraken executes independent modules that focus exclusively on data acquisition rather than embedding source-specific code directly into the system.

Modules typically interact with external intelligence sources such as passive DNS services, OSINT platforms, sandbox environments, blog platforms or messaging channels. A module receives input entities from the tracking pipeline and attempts to retrieve additional signals related to those entities.

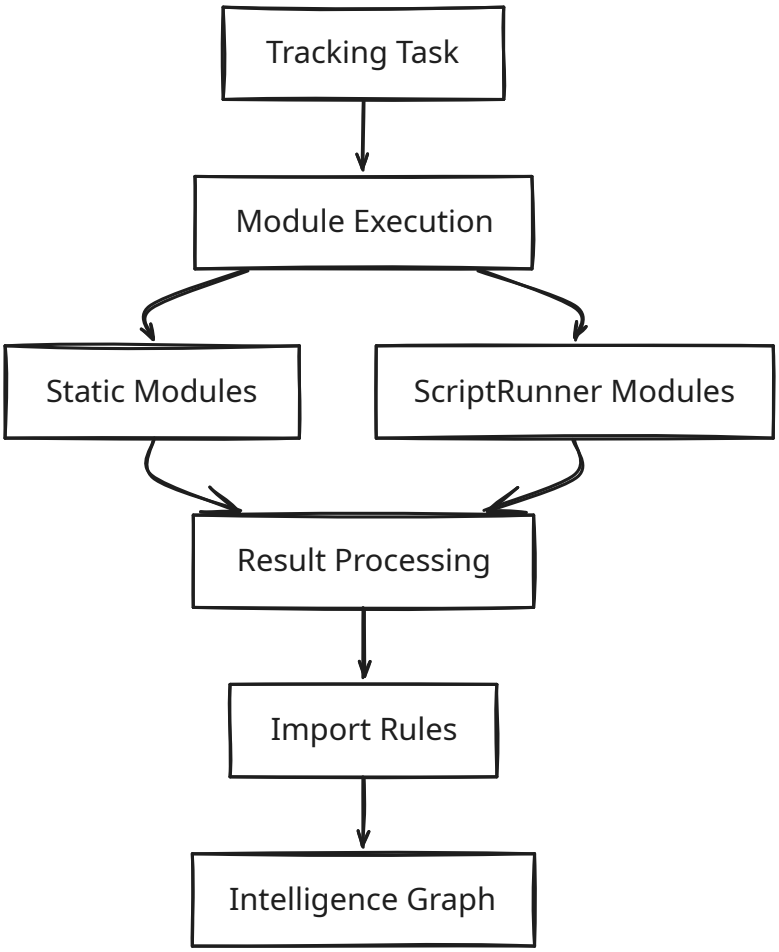
Two different module types are supported.

Static Collection Modules

Static modules are built-in collectors that ship with the platform. They typically interact with known sources or APIs and are maintained with the core system.

Examples include collectors for passive DNS systems, OSINT feeds, or infrastructure discovery workflows.

Static modules are useful for stable intelligence sources where the collection logic rarely changes.



Script-Runner Modules

Script-Runner modules allow custom collection logic to be executed as independent scripts. This mechanism was introduced to support rapid experimentation during threat research.

Analysts can write small collection scripts to implement specific intelligence workflows.
These scripts receive input entities and produce structured results that are passed back into the Kraken processing pipeline.

Script-Runner makes it possible to quickly integrate new intelligence sources, test collection ideas or adapt to changes in adversary infrastructure.

Isolated Execution Environment

Script-Runner modules are executed within isolated runtime environments.

Each script runs inside a containerized execution environment that separates custom collection logic from the core platform, therefore experimental modules and external scripts run isolated from the core system.

Containerized execution also helps analysts to define module dependencies independently. Scripts include additional libraries, tools or specialized environments required for specific intelligence sources.

Module Execution

When a tracking task is executed, the task manager invokes the modules assigned to the tracking definition. Each module receives the relevant input entities and attempts to extract additional signals.

Results generated by the module are returned to the result processing pipeline where they are normalized and evaluated by import rules.

Modules themselves do not directly modify the intelligence graph.
Their responsibility is limited to signal extraction.

Extending the Platform

The platform is modular. The core system stays unchanged when new modules are added. New intelligence sources can be integrated by implementing additional modules rather than modifying the tracking pipeline or storage layer.

This makes it possible to easily integrate new sources, research methods or analysis techniques over time.

Sandbox Integration

Malware analysis often produces infrastructure signals that are directly relevant for threat intelligence. Network connections, command and control endpoints, dropped files and domain lookups can reveal additional infrastructure used by an adversary. Kraken mainly tracks infrastructure.

Sandbox systems look at the other side: how malware behaves when it runs.

Future versions of Kraken will be able to connect to malware analysis systems. Sandbox analysis results can produce signals that are directly fed into the tracking pipeline. Observed domains, IP addresses or URLs extracted from sandbox executions may become new infrastructure entities or extend existing relationships within the intelligence graph.

This integration allows behavioral telemetry from malware samples to contribute to infrastructure discovery. In practice, sandbox observations may reveal command and control endpoints or staging servers that were not previously visible through OSINT collection.

With infrastructure data and sandbox results together, the graph can link malware to the infrastructure it uses.

Gamaredon Tracking Example

Infrastructure used by threat actors often changes rapidly.
Domains are rotated, hosting providers change and command and control endpoints are replaced once they become visible to defenders.

Gamaredon provides a useful example of this behavior.
The group frequently uses temporary infrastructure, public services and short-lived domains to distribute operational data or store command instructions.
Tracking such infrastructure manually is difficult because individual signals often appear unrelated at first. One domain or IP usually doesn’t exposes the full campaign infrastructure.

In Kraken, infrastructure tracking begins with a small number of known entities. These may originate from OSINT research, incident reports or previous infrastructure observations.

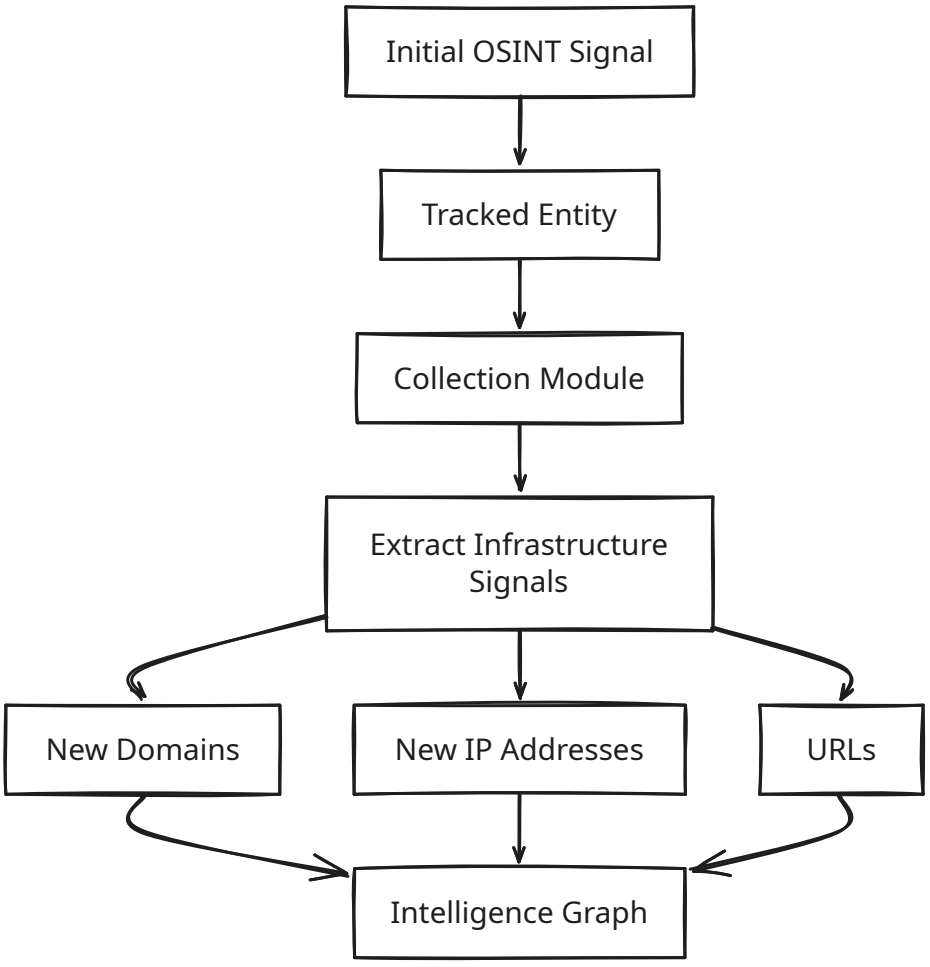
For example, a Telegram or blog platform account used as a dead-drop location may serve as the initial tracking entity. A tracking definition assigns a collection module capable of retrieving and parsing the content of that platform.

During each execution cycle the module retrieves the latest content and extracts potential infrastructure signals such as domains, IP addresses or URLs embedded in the posts. Signals enter the pipeline, are normalized and evaluated by import rules. New infrastructure nodes are added to the intelligence graph, while existing entities may gain new relationships or evidence.

Over time this process gradually expands the visible infrastructure.
New domains may resolve to previously unseen IP addresses.
Those IP addresses may host additional domains that belong to the same campaign.

One indicator can reveal a larger infrastructure cluster. Because tracking tasks run continuously, Kraken is able to observe infrastructure rotation patterns and changes in operational behavior. Domains that disappear may be replaced by new ones, while existing infrastructure may be reused across different campaign stages.

The resulting intelligence graph reflects the evolving infrastructure behind the tracked activity.



Operational Use Cases

Kraken is intended to support practical threat intelligence workflows where infrastructure signals must be collected, correlated and monitored over longer periods of time. The platform is designed to enable continuous observation of adversary infrastructure and related operational activity.

This approach works well in several common research scenarios.

Infrastructure Monitoring

Threat actors frequently rotate infrastructure such as domains, command-and-control endpoints and staging servers.
Once an indicator becomes known, it may quickly disappear or be replaced.

Tracking definitions keep known infrastructure entities under continuous monitoring. Collection modules periodically retrieve additional signals related to those entities, making it possible to observe how infrastructure evolves during an active campaign.

It is particularly useful when investigating infrastructure clusters reused across multiple operations.

Campaign Expansion

Initial intelligence reports often contain only a small number of indicators. Domains, IPs or URLs mentioned in a report often show only part of the infrastructure an adversary uses.

Collecting signals from these indicators repeatedly can expose more infrastructure. New domains may resolve to shared IP addresses, while newly discovered infrastructure may host additional campaign resources.

This process gradually expands the intelligence graph and can reveal campaign infrastructure that was not present in the original intelligence source.

Malware Infrastructure Discovery

Oftentimes Malware samples contain infrastructure references such as command-and-control endpoints or staging servers.
These artifacts can serve as starting points for further infrastructure tracking.

Once these signals enter the tracking pipeline, collection modules can discover additional infrastructure linked to the new endpoints.

Long-Term Infrastructure Observation

Some threat actors reuse infrastructure components over extended periods of time. Domains may disappear temporarily and later reappear under new campaigns or operational contexts.

Because Kraken stores historical observations, analysts can see how infrastructure evolves across multiple tracking cycles. This makes it possible to identify patterns such as infrastructure reuse, domain rotation strategies or hosting provider changes.

Future Development

Kraken is still evolving and several components of the platform are expected to expand as additional research workflows are integrated.

One area of development is the integration of malware analysis data. Infrastructure signals extracted during malware execution often reveal command and control endpoints, staging servers or domain generation patterns that are not visible through OSINT collection alone.

Future versions of Kraken will integrate with Malwarebox, a sandbox used to analyze malware and collect behavioral telemetry. Network connections, domain lookups and other observed infrastructure artifacts produced during sandbox execution can be fed directly into the tracking pipeline.

This allows malware behavior to contribute to infrastructure discovery. Signals observed during sandbox execution may reveal previously unknown domains or IP addresses that can then be tracked as infrastructure entities within the intelligence graph.

Another direction involves automated infrastructure clustering. Infrastructure nodes often appear in groups that share hosting providers, domain naming patterns or operational characteristics. Graph analysis techniques may help identify such clusters and highlight relationships between campaigns that are not immediately visible during manual analysis.

Additional work will likely focus on expanding the module ecosystem. Collection logic is implemented in modules. New intelligence sources can be added without modifying the core system. This makes it possible to incorporate new research techniques, data sources or extraction methods as they become relevant.

These developments extend the platform while preserving the central idea behind Kraken: continuous observation of adversary infrastructure and the ability to model operational activity as a growing intelligence graph.

Conclusion

Adversary infrastructure is easy to observe, but tracking is often short-lived and spread across multiple tools. Domains, IP addresses and hosting providers are typically observed only at the moment they are discovered, while the broader structure of the infrastructure behind a campaign remains difficult to reconstruct.

Kraken explores a different approach. Infrastructure, campaigns, malware and threat actors are modeled as connected entities within a continuously evolving intelligence graph, allowing indicators to be analyzed in the context of their relationships.

By combining automated tracking modules, structured entity modeling and a modular collection architecture, Kraken turns infrastructure analysis into an ongoing observation process rather than a one-time enrichment task.

This approach makes it possible to follow how adversary infrastructure changes over time, how campaigns expand and how operational patterns emerge from seemingly unrelated signals.

Kraken does not attempt to replace existing intelligence sources, but provides a framework for collecting, structuring and connecting those signals in a way that allows long-term infrastructure analysis to emerge naturally from continuous tracking.