Harmonizing People, Processes, and Technology for Robust Cybersecurity

D3.1 Enhanced Threat Intelligence System v1

Document Summary Information

Grant Agreement No	101158555	**Acronym**	CYberSynchrony
Full Title	Harmonizing People, Processes, and Technology for Robust Cybersecurity		
Start Date	01/06/2024	**Duration**	36 months
Project URL	https://www.cybersynchrony.eu/		
Deliverable	D3.1 Enhanced Threat Intelligence System v1		
Work Package	WP3 Comprehensive Threat Intelligence and Customized Risk Evaluation (CYBERRA) and Cybersecurity Resilience and Incident Response Enhancement (CYRESCUE)		
Contractual due date	31/05/2025	**Actual submission date**	03/06/2025
Type	Other	**Dissemination Level**	Public
Lead Beneficiary	BYTE		
Responsible Author	George Karavokiros (BYTE)		
Contributions from	A.Purificato (MER), A.Psychas (MAG), E.Michos (IANUS)		

Funded by the European Union

ECCC

Funded by the European Union under Grant Agreement no. 101158555 and supported by the European Cybersecurity Competence Centre (ECCC). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the granting authority can be held responsible for them.

Revision history

Version	Issue Date	Changes	Contributor(s)
0.1	14.03.2025	Initial deliverable structure	George Karavokiros (BYTE)
0.2	02.05.2025	Technical contribution – Kitsune	Andrea Purificato (MER)
0.3	20.04.2025	Technical contribution – MISP	Evangelos Michos (IANUS)
0.4	23.04.2025	Added UFT section and updated architecture section	Alexandros Psychas(MAG)
0.5	06.05.2025	Final draft ready for internal review	George Karavokiros (BYTE)
0.6	09.05.2025	Peer review	Vangelis Malamas (FP)
0.7	13.05.2025	Peer review	Evangelos Michos (IANUS)
0.8	21.05.2025	Final draft addressing comments from internal review	Alexandros Psychas (MAG)
0.9	03.06.2025	Final quality check	Panayiotis Klitou (EBOS)
1.0	03.06/3035	Final version ready for submission	Violeta Vasileva (MAG)

Disclaimer

The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the CYberSynchrony consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© CYberSynchrony Consortium This Deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

[1] According to CYberSynchrony's Quality Assurance Process for reports:

- 3 months before Deliverable's Due Date: Table of Contents ready to be reviewed by WP leader/ Technical Manager/ Quality Manager
- 2 months before Deliverable's Due Date: 50% should be complete. Review by the Quality Manager (& Technical Manager for software).
- 1 month before Deliverable's Due Date:100% should be complete. Review by the Quality Manager and Peer Reviewers.
- M: Sent to PC for Submission to the EC after addressing all comments by Quality Manager and Peer Reviewers

Table of Contents

Executive Summary 7
1 Introduction 8
1.1 Mapping CYberSynchrony Outputs 8
1.2 Linkage to other project outputs 9
1.3 Deliverable Structure 9
2 Position in CYberSynchrony Architecture 11
3 MISP (IANUS) 13
3.1 Component Description 13
3.2 Architecture 14
3.3 Technologies Used 15
4 Kitsune Search (MER) 17
4.1 Component Description 17
4.2 Architecture 18
4.3 Technologies Used 18
5 Unalterable Forensic Techniques adaptation(UFT) (BYTE/MAG) 19
5.1 Component Description 19
5.2 Architecture 19
5.3 Technologies Used 20
6 Conclusions 22
7 References 23
Annex I: *Preliminary PoC and Prototypes (MISP)* 24
Annex II: *Preliminary PoC and Prototypes (Kitsune Search)* 29
YAML File (Kitsune Monitoring, Search) 29
Mock Prototype / Demo 29
Authentication/Authorization API 29
Search API (Underground) 30
Search API (Data Leak) 33
Search API (Infostealer) 35
Feeds (IOCs) 37

List of Figures

Figure 1: CYberSynchrony Architecture – Highlighting Components Relevant to D3.1 11
Figure 2: MISP internal Architecture 13
Figure 3: Kitsune Search placement inside CyberSynchrony 18
Figure 4: UFT integrity flow architecture. 20
Figure 5: MISP Home page 24
Figure 6: Application Logs 25
Figure 7: Audit logs 25
Figure 8: Diagram illustrating the API login process 30
Figure 9: Diagram illustrating the API refresh authorization process 30

© CYberSynchrony, 2024

Figure 10: Representation of “underground search” functionality with response 31
Figure 11: Representation of “Leak search” functionality with response 34
Figure 12: Representation of “botnet (infostealer) search” functionality with response 35

List of Tables

Table 1: Adherence to CYberSynchrony’s GA 8
Table 2: Links to other Project Outputs/Work 9
Table 3: MISP alignment with CYberSynchrony objectives. 14
Table 4: MISP technology stack 16
Table 5: Kitsune Search alignment with CYberSynchrony objectives. 17
Table 6: Kitsune Monitoring technology stack 18
Table 7: UFT alignment with CYberSynchrony objectives. 19
Table 8: UFT technology stack. 21
Table 9: Kitsune Search - API Underground list 31
Table 10: Kitsune Search - API Data Leak list 34
Table 11: Kitsune Search - API Botnet Infostealer list 35
Table 12: Kitsune Search - API Feeds list 37

© CYberSynchrony, 2024

Acronyms and abbreviations

Abbreviation / Term	Description
AGPL	Affero General Public License
API	Application Programming Interface
APT	Advanced Persistent Threat
ATT	Adversarial Tactics, Techniques
CIRCL	Computer Incident Response Center Luxembourg
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
DNS	Domain Name System
DPM	Data Preparation Module
GDPR	General Data Protection Regulation
HEUR	Heuristic
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
LAMP	Linux, Apache, MySQL, PHP (web stack)
LDAP	Lightweight Directory Access Protocol
LEMP	Linux, Nginx (Engine-X), MySQL, PHP (web stack)
LLC	Limited Liability Company
MFA	Multi-Factor Authentication
MISP	Malware Information Sharing Platform
MITRE	MITRE Corporation
PHP	Hypertext Preprocessor
RAW	Raw data format
RBAC	Role-Based Access Control
REST	Representational State Transfer
SAML	Security Assertion Markup Language

© CYberSynchrony, 2024

SDK	Software Development Kit
SHA	Secure Hash Algorithm
SOC	Security Operations Center
STIX	Structured Threat Information Expression
TLP	Traffic Light Protocol
UFT	Unalterable Forensic Techniques
WHOIS	Domain Registry Lookup Protocol
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

© CYberSynchrony, 2024

Executive Summary

This deliverable, D3.1 "Enhanced Threat Intelligence System v1," introduces the initial design, deployment, and alignment of three key components developed within Work Package 3 (WP3) of the CYberSynchrony project: the Malware Information Sharing Platform (**MISP**), the **Kitsune Search** tool, and the **Unalterable Forensic Techniques (UFT)** adaptation.

The work presented focuses on enhancing the platform's cyber threat intelligence (CTI) capabilities, enabling asset-specific vulnerability assessment, and ensuring the integrity and immutability of collected security data. **MISP** enables structured and collaborative threat intelligence sharing, enriching the internal security posture with external threat knowledge. **Kitsune Search** provides a focused, API-driven capability to retrieve threat intelligence related to specific assets, facilitating proactive risk management. The **UFT process** guarantees the forensic soundness of security data by registering cryptographic hashes on a permissioned distributed ledger (Hyperledger Fabric [5]), ensuring that collected evidence remains trustworthy throughout its lifecycle.

All three components are closely aligned with CYberSynchrony's project objectives, particularly in strengthening trust, enhancing proactive defence mechanisms, improving infrastructure resilience, and enabling privacy-preserving collaboration across stakeholders. Since this deliverable is categorized as an **"Other"** type deliverable, preliminary demonstrators and proof-of-concept implementations of the developed components are included in the annexes. These demonstrations provide initial insights into the functionality, integration readiness, and operational value of each tool within the CYberSynchrony ecosystem.

This first version focuses on developing the core functionalities and ensuring that each tool can operate independently within the broader architecture. In the next iteration of the deliverable the primary focus will shift toward achieving seamless **integration and interoperability** of these components within the unified CYberSynchrony platform. The ultimate goal is to embed these capabilities into existing cybersecurity ecosystems without disrupting established monitoring or incident response processes, thereby providing additional intelligence, integrity assurance, and operational resilience.

© CYberSynchrony, 2024

1 Introduction

1.1 Mapping CYberSynchrony Outputs

This section presents the CYberSynchrony's Grant Agreement (GA) commitments, as extracted from the formal deliverable and task description, in respect to their outputs and work to be performed.

Table 1: Adherence to CYberSynchrony's GA

Component no	Component title	Respective Document Chapter(s)	Justification
DELIVERABLE			
D3.1	D3.1 Enhanced Threat Intelligence System v1	Chapters 1–6, Annex I–II	This deliverable presents the initial design and implementation of components supporting comprehensive threat intelligence, aligned with WP3 objectives. It introduces tools such as MISP, Kitsune Search, and Unalterable Forensic Techniques (UFT), showcasing their integration and technical roles.
TASKS			
T3.1	T3.1 – Cyber-Threat Intelligence Platforms Deployment and Fine-Tuning	Chapter 3, Annex I	Describes the deployment and fine-tuning of MISP as a structured and sharable threat intelligence platform. It highlights MISP's integration with other tools to ingest and disseminate cyber threat information efficiently
T3.2	T3.2 – Adopting Unalterable Forensic Techniques	Chapter 5,	Details the architecture and design of UFT processes, which ensure forensic soundness by protecting the integrity of cyber evidence before it enters the analysis pipeline
T3.3	T3.3 – Crafting Personalized Risk Assessment Tools	Chapter 4, Annex II	Presents Kitsune Search, a system enabling asset-aware threat intelligence that tailors risk correlation to specific digital assets, enhancing the precision of threat assessments

© CYberSynchrony, 2024

1.2 Linkage to other project outputs

The development of the components described in this deliverable is inherently connected to broader architectural and data assurance efforts within the CYberSynchrony project. The following table outlines the key project outputs and tasks that influence or are influenced by the design and integration mainly of the UFT.

Table 2: Links to other Project Outputs/Work

WP Number	Task Number	Deliverable Number related	Content
WP2	T2.3, T2.4	D2.1	The architectural principles and system requirements defined in T2.4 guided the integration approach for MISP, UFT, and Kitsune Search. These components were designed to align with the modular, interoperable architecture established in D2.1.
WP4	T4.1	D4.1	The Unalterable Forensic Techniques (UFT) described in D3.1 are closely tied to the IDS systems in T4.1. These systems are expected to integrate with UFT's pipeline to ensure data fed into the intrusion detection process maintains its integrity, thereby supporting reliable analysis and evidence defensibility.
WP5	T5.1	D5.1	The distributed ledger technologies developed under T5.1 are fundamental to the UFT solution. They enable secure hashing and traceability of security log data, ensuring immutable records for forensic use by IDS components.
WP8	T8.1	D8.1	The design and development of threat intelligence and unalterable forensic components in D3.1 were guided by the need for seamless integration into the CYberSynchrony platform. The work performed under WP3 took into consideration the technical and operational requirements of the overall integration effort, ensuring that the developed tools align with the platform's use case-driven validation approach defined in WP8.

1.3 Deliverable Structure

This deliverable is structured into six core chapters and two annexes, detailing the design, functionality, and preliminary implementation of the components contributing to the CYberSynchrony Enhanced Threat Intelligence System.

Chapter 2 Position in CYberSynchrony Architecture: Presents how the MISP platform, Kitsune Search service, and Unalterable Forensic Techniques (UFT) are positioned within the updated CYberSynchrony modular architecture, highlighting their roles and interactions.

Chapter 3 MISP: Describes the deployment, configuration, and integration of the MISP platform as a centralized cyber threat intelligence aggregation and sharing tool.

Chapter 4 Kitsune Search: Details the external service providing asset-specific threat intelligence search capabilities, supporting proactive risk management.

© CYberSynchrony, 2024

Chapter 5 Unalterable Forensic Techniques adaptation: Explains the UFT process, focusing on the development of the Data Preparation Module and the use of distributed ledger technology to guarantee the integrity of collected forensic data.

Chapter 6 Conclusions: Summarizes the main findings and contributions of the deliverable, and outlines future development steps toward component integration and interoperability within the CYberSynchrony platform.

© CYberSynchrony, 2024

2 Position in CYberSynchrony Architecture

This section presents the positioning and interaction of the components described in this deliverable within the overall CYberSynchrony architecture. The architectural view shown below updates and expands the initial design from Deliverable D2.1, reflecting the project's evolving integration and deployment plans.

The figure highlights the relevant components associated with D3.1: **MISP**, **Kitsune Search**, and the processes related to the **Unalterable Forensic Techniques (UFT)**. MISP and Kitsune Search are marked in **red** as direct components, while the UFT-related processes and modules are marked in **yellow**, indicating their cross-cutting role in ensuring data integrity across the platform.

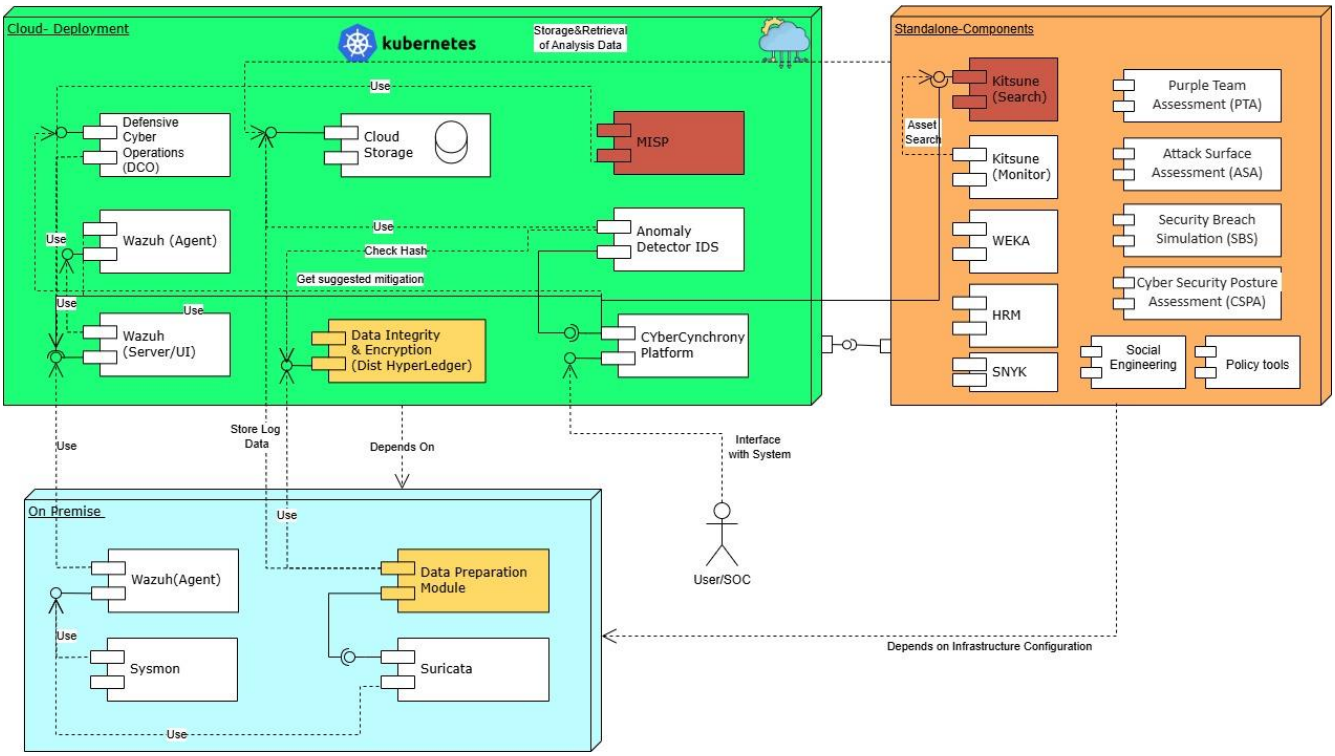


Figure 1: CYberSynchrony Architecture – Highlighting Components Relevant to D3.1

MISP (Malware Information Sharing Platform & Threat Sharing) plays a key role as CYberSynchrony's core threat intelligence aggregation and dissemination platform. Positioned within the cloud deployment layer, MISP is connected to two main subsystems:

- **Wazuh Server/UI**: MISP interfaces with the Wazuh server to enhance internal threat correlation by cross-referencing detected alerts with external cyber threat intelligence feeds, offering richer context and more accurate incident triage.
- **Cloud Storage**: MISP also interacts with the Cloud Storage module, enabling the secure storage and retrieval of structured threat intelligence (CTI) objects. This integration supports both real-time threat analysis and historical correlation needs.

MISP acts as a dynamic intelligence broker, enriching internal monitoring outputs with external situational awareness, and contributing to proactive cybersecurity defence.

Kitsune Search operates as an **external standalone component**, offering powerful search capabilities across multiple cyber threat intelligence sources. Rather than ingesting data continuously, Kitsune Search provides an **API-based access service**, allowing the CYberSynchrony platform to perform targeted searches regarding:

© CYberSynchrony, 2024

- Attempted vulnerabilities,
- Exploit attempts,
- Data leaks,
- Underground or dark web activity linked to important internal assets.

By querying Kitsune's API when needed, CYberSynchrony extends its threat intelligence coverage without overburdening internal data ingestion pipelines, ensuring efficient, focused intelligence gathering.

The **Unalterable Forensic Techniques** approach ensures that cybersecurity monitoring and incident response are underpinned by immutable, trustworthy evidence. UFT is implemented through a combination of:

- **Data Preparation Module**: Deployed on-premises, this module collects log batches produced by the local agents (e.g., Wazuh Agent, Sysmon, Suricata). It generates cryptographic hashes for each batch, ensuring evidence authenticity.
- **Data Integrity & Encryption Module (Distributed Ledger - HyperLedger)**: The generated hashes are stored securely on a distributed ledger, providing tamper-proof, verifiable proof of data integrity.

After hash registration, the original log batches are transmitted to Cloud Storage for future processing and analysis.
This process guarantees **end-to-end evidence integrity** across the platform lifecycle — from initial log collection to final forensic analysis or threat intelligence enrichment.

© CYberSynchrony, 2024

3 MISP (IANUS)

3.1 Component Description

MISP is an open-source threat intelligence platform, originally developed by CIRCL (Computer Incident Response Center Luxembourg) and maintained by the MISP Project, co-funded by the European Union and the Luxembourg House of Cybersecurity. In the context of CYberSynchrony, IANUS is responsible for its deployment, integration, and technical customization under T3.1. MISP serves as the core CTI platform within the CYberSynchrony project. Its primary purpose is to aggregate, analyse, correlate, and share threat intelligence in a structured, machine-readable format, facilitating early warning, situational awareness, and coordinated defence across critical sectors (e.g. healthcare, public administration, academia). MISP enhances cybersecurity resilience by enabling real-time exchange of Indicators of Compromise (IoCs), attack patterns, and contextual intelligence between stakeholders, helping organizations detect, assess, and respond to threats before they materialize into incidents [1].

Within WP3, MISP contributes directly to the collection and dissemination of cyber threat indicators. It acts as the foundational tool for WP3's objective to "*enhance the ability of Member States (MS) to assess risk in real-time based on actionable threat data*." MISP can provide structured intelligence objects, such as:

- MISP Events, Objects
- Attributes
- Galaxies (e.g., threat actor clusters)
- Taxonomies (standardized labels and classifications)

In practical terms, MISP supports GDPR/NIS2 compliance through access control (RBAC, OAuth/SAML) and audit logging. This ensures threat data is shared securely and ethically. Its open-source nature, combined with mature documentation and community support, makes MISP a cost-effective and extensible solution that aligns with the project's emphasis on sustainable, scalable technology deployments. The platform has already seen wide adoption by CSIRTs and governmental cybersecurity teams across Europe, further solidifying its role as a state-of-the-art, trusted standard for threat intelligence management. It acts not only as a passive intelligence aggregator but also as a proactive threat predictor [1].

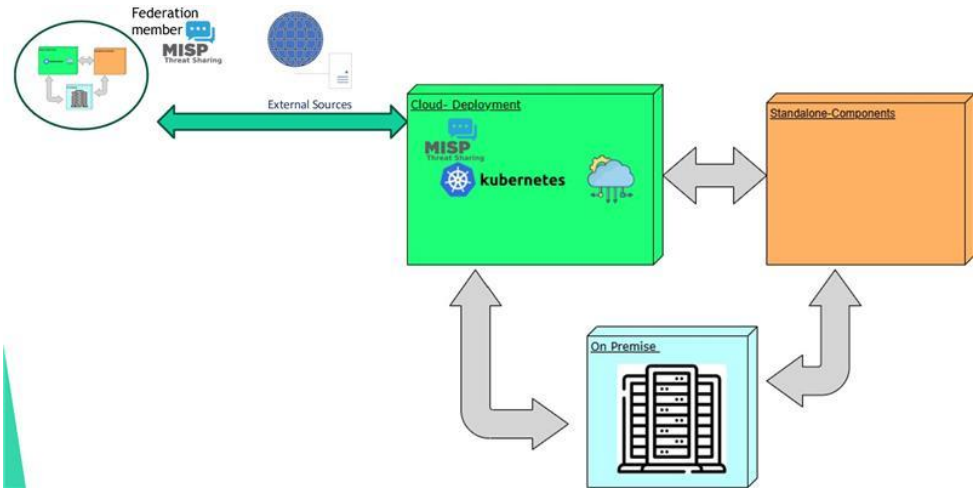


Figure 2: MISP internal Architecture

Internally, MISP will operate as a **cloud-based threat intelligence hub**, receiving external threat data and synchronizing with other MISP federation members. It integrates with cloud-deployed components and standalone tools by exposing enriched IoCs and contextual metadata via REST APIs, supporting cross-border sharing and real-time security enhancement. To add more to this, MISP will be exploited for **exchange of threat intelligence** using its **federated services**. In more detail, for cross-border cases, the MISP federation enables **bidirectional communication** between national or sector-specific MISP instances - each capable of both sending

© CYberSynchrony, 2024

and receiving threat intelligence data. This setup allows organizations across MSs (*e.g. hospitals in a specific country*) to share homogeneous, high-value intelligence relevant to their sector, such as targeted phishing campaigns or ransomware patterns. The **added value** lies in the relevance and immediacy of the shared IoCs, which reflect sector-specific threats. The automation of this structured exchange enhances the overall situational awareness, early warning, and coordinated defence. **Ultimately, MISP federation operationalizes the core principle of CYberSynchrony: real-time threat intelligence sharing across trusted borders**.

Regarding **MISP's** deployment inside the updated CYberSynchrony architecture, data collection begins on-premises, where **Suricata** and **Wazuh Agent** monitor network and endpoint activity for anomalies and threats. These signals, together with telemetry from **Sysmon**, are structured and enriched by the **Data Preparation Module** and then transmitted to the cloud. Instead of feeding directly into the **Wazuh Server**, the validated and pre-processed logs are first stored in **Cloud Storage**. From there, a cloud-based **Wazuh Agent** retrieves the data and forwards it to the **Wazuh Server/UI** for centralized correlation, threat analysis, and incident visualization. In parallel, **MISP** correlates this refined input with external threat intelligence feeds, enhancing alert context and enabling actionable insights. All outputs are unified through the **CYberSynchrony Platform UI**, where **SOC analysts** can investigate incidents, correlate evidence, and respond in real time.

Table 3: MISP alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1-Enhance real-time cyber threat assessment capabilities	MISP directly supports this objective by aggregating and IoCs in near real-time. Through automated feeds, REST API integrations, and correlation engines, MISP enables the rapid identification of emerging threats and supports incident responders in reacting swiftly to them. Its interoperability with SIEMs and IDS/IPS systems like Wazuh and Suricata further enhances real-time threat assessment.
Obj 2-Offer specialized risk scenario analysis tailored for Member States	While MISP itself does not generate full risk scenarios, it provides highly structured, tagged, and contextualized threat data (including threat actors, malware families, TTPs) that can be used by other components (e.g., risk scoring engines in WP3 or WP4) to generate these analyses. Its federation capability also allows tailoring threat intel for specific MS contexts.
Obj 4 - Provide insights into potential cyber risks faced by entities	MISP enriches raw threat indicators with contextual metadata (e.g., TLP, threat level, attack vectors), taxonomies, and historical correlation. These insights support analysts and security operators in understanding the scope and nature of cyber risks relevant to their sector or geography.

3.2 Architecture

As already mentioned, MISP is not a proprietary software of IANUS, but a rather open-source solution, which IANUS has implemented with custom solutions in previous EU-funded projects. The source code for MISP can be found in the official GitHub repository (see [2]). MISP is offered under the AGPL (Affero General Public License) version 3 license (see [3])

The internal architecture of MISP is designed around modularity, performance, and interoperability, making it highly adaptable for cybersecurity environments. MISP is composed of several key subsystems that interact seamlessly to support the creation, enrichment, analysis and sharing of structured cyber threat intelligence. At its core, MISP uses a PHP-based backend running on a LAMP/LEMP stack, with MariaDB/MySQL as the central database storing events, attributes, taxonomies, users, and organizational data. The frontend is built with

© CYberSynchrony, 2024

JavaScript and jQuery, offering a dynamic and intuitive user interface. All operations are exposed via a RESTful API, enabling automation and integration with third-party tools, including SIEMs, IDS/IPS, and external CTI platforms.

The architectural flow begins with data ingestion, where users or automated feeds ingest IoCs, threat actor profiles, and contextual metadata into the platform. Each contribution becomes a MISP Event, which can be further enriched with MISP Objects (structured representations like malware samples, phishing attempts, or threat campaigns) and MISP Attributes (key-value pairs such as IP addresses, file hashes, or domains). The platform utilizes Galaxies (collections of related entities, e.g., APT groups) and Taxonomies (standardized classification tags like TLP or threat level) to semantically organize data. Background tasks such as correlation, enrichment, or feed synchronization are managed via Redis-based workers, allowing the system to scale and respond quickly to new intelligence input.

A key architectural strength of MISP lies in its modular plugin system called misp-modules, which enables third-party extensions for enrichment (e.g., DNS, WHOIS, GeoIP lookups), export formats (e.g., Suricata rules, STIX, CSV), and importers (e.g., from VirusTotal or other CTI platforms). These modules communicate over an internal API layer and are called dynamically as needed—either through the user interface or via automated tasks triggered through the API. In federated environments, MISP supports synchronization between instances, allowing distributed deployment models where multiple CSIRTs or Member States can share subsets of intelligence based on trust levels and data classification.

From a cybersecurity needs perspective, this architecture offers both strategic flexibility and operational robustness. It allows organizations to build collaborative defense ecosystems, share real-time intelligence without vendor lock-in, and automate the triage of alerts via integration with detection systems like Suricata and Wazuh. Additionally, the use of open standards (e.g., STIX 2.1, OpenIOC, CVE, MITRE ATT&CK) ensures that MISP can interoperate within the broader cybersecurity toolchain, addressing needs for situational awareness, incident response, and proactive threat hunting.

MISP's architecture is considered state-of-the-art for several reasons: it is highly extensible, supported by a large community of developers and researchers, and co-funded by the EU, making it not only technically mature but also aligned with regulatory and strategic priorities across Europe. Unlike closed, proprietary CTI platforms, MISP empowers users to customize their deployment to local needs, ensuring both scalability in cloud-native environments (via Docker, Kubernetes) and deep integration in legacy or hybrid infrastructures. This flexibility, combined with its rich feature set and strong governance model, positions MISP as a leading solution for structured, actionable threat intelligence management. MISP is already adopted by CSIRTs across Europe as a central hub for structured threat intelligence collaboration.

3.3 Technologies Used

The core backend is developed in PHP, while the frontend uses JavaScript and jQuery to offer a responsive and interactive user experience. The platform follows an API-first design, exposing a powerful RESTful interface that enables seamless automation and integration with external cybersecurity tools such as SIEMs (e.g., Splunk, Wazuh), IDS/IPS (e.g., Suricata), and threat intelligence platforms (e.g., ThreatConnect, Anomali).

The platform's database is managed by MariaDB/MySQL, optimized for storing structured threat intelligence objects like MISP Events, Attributes, and Tags. For background task processing, MISP uses Redis as a message queue, supporting asynchronous jobs such as feed synchronization, correlation, and data enrichment. MISP also relies on the PyMISP Python library, which provides a client interface to the MISP API, allowing users to script interactions and automate operations. To further extend its capabilities, MISP integrates with misp-modules, a microservices framework written in Python, which supports enrichment services, format converters, and third-party data ingestion.

MISP supports Docker-based deployment and is also compatible with Kubernetes and bare-metal installations, making it flexible across cloud-native and traditional infrastructures. It supports multiple authentication

© CYberSynchrony, 2024

protocols, including LDAP, OAuth, and SAML, and adheres to security best practices with support for multi-factor authentication (MFA), audit logging, and RBAC (Role-Based Access Control).

Table 4: MISP technology stack.

Technology	Purpose
PHP	Used for backend development and core logic of the MISP platform.
JavaScript, jQuery	Power the frontend for interactive and dynamic user interfaces.
MariaDB / MySQL	Serve as the relational database backend for storing threat data and metadata.
Redis	Handles background job queuing and asynchronous task execution.
PyMISP (Python)	A Python client library for accessing and automating the MISP REST API.
misp-modules (Python)	Provide enrichment, import/export, and data transformation functionality.
STIX 2.0/2.1, JSON, XML, CSV	Supported data formats for threat intelligence interoperability and sharing.
Docker, Kubernetes	Used for containerized deployment and orchestration in cloud-native environments.
Apache / Nginx	Act as the web server for hosting the MISP user interface and API.
LDAP, SAML, OAuth	Enable secure authentication and access control mechanisms.
GnuPG	Ensures secure email communication and encryption of sensitive notifications.

© CYberSynchrony, 2024

4 Kitsune Search (MER)

4.1 Component Description

Kitsune Search is a cybersecurity tool designed to gather and analyse threat intelligence through advanced search functionalities.

It allows users to search for specific information about Threat Actors and the Threat Landscape, providing details related to Indicators of Compromise (IOCs). Additionally, it retrieves data leak exposure information associated with particular domains or keywords, helping organizations identify potential security risks.

Kitsune enables queries across threat actor mentions, data leaks, botnet infections (infostealer), and dark web activity, offering real-time insights on cyber exposures.

The innovation lies in the use of the tool for proactive risk detection and a timely response to emerging incidents, minimizing the impact of potential cyber threats.

Table 5: Kitsune Search alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1-Enhance real-time cyberthreat assessment capabilities.	Kitsune Search enhances real-time cyberthreat assessment by enabling fast and targeted searches for Threat Actors, Indicators of Compromise (IOCs), data leaks, and BOTNET infections. It retrieves critical intelligence on compromised domains, credentials, and dark web mentions, allowing security teams to quickly assess emerging threats. By providing immediate access to underground cyber activities, Kitsune Search helps organizations proactively identify risks and strengthen their cybersecurity posture with actionable intelligence.
Obj 2-Offer specialized risk scenario analysis tailored for MS.	By aggregating and analyzing threat intelligence data across a Member State, a more comprehensive and up-to-date picture of the specific cyber threats and vulnerabilities prevalent in that MS can be developed.
Obj 4-Provide insights into potential cyber risks faced by entities.	Kitsune Search accomplishes the objective of providing insights into potential cyber risks faced by entities by enabling advanced searches for critical threat intelligence data. It retrieves information on Threat Actors, Indicators of Compromise (IOCs), data leak exposures, and BOTNET infections, helping organizations identify vulnerabilities linked to their domains or email addresses. By scanning for mentions of specific keywords in the dark web and darknet, it uncovers hidden threats and potential attack vectors.

© CYberSynchrony, 2024

4.2 Architecture

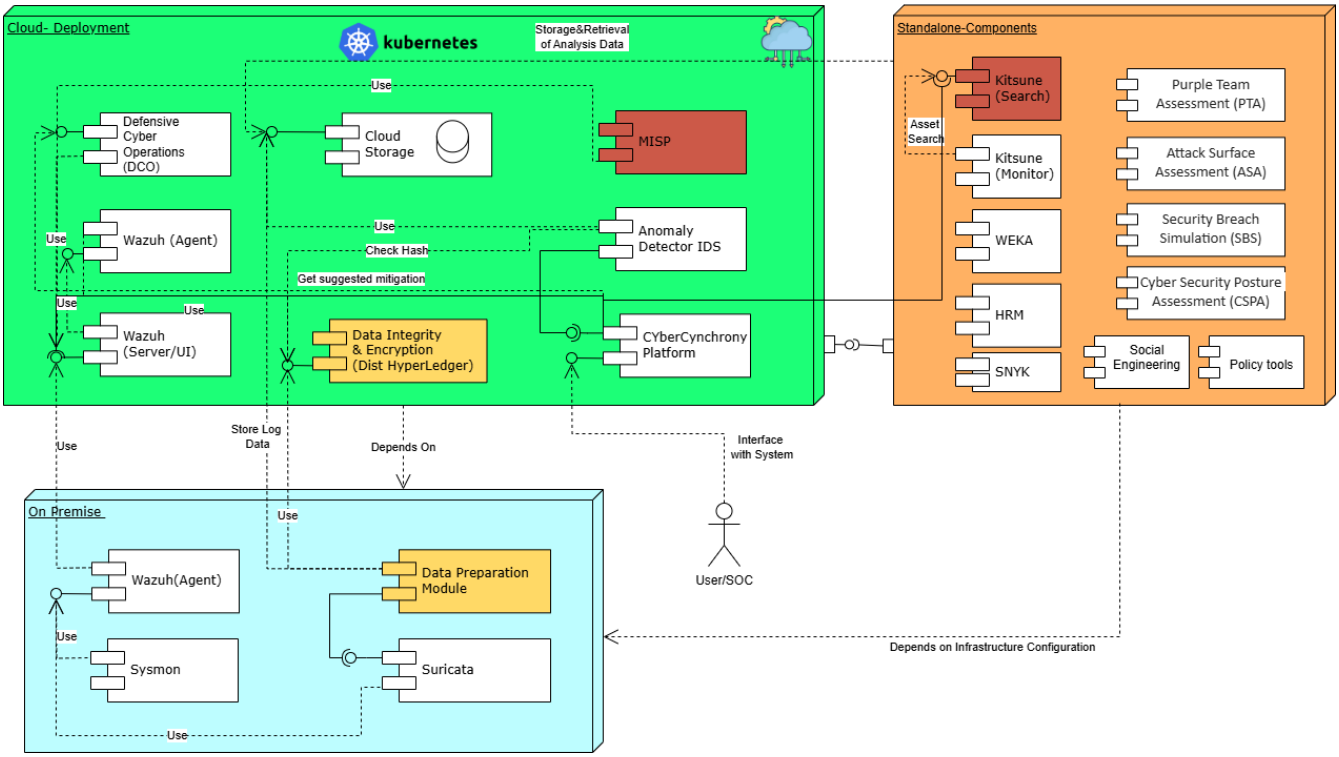


Figure 3: Kitsune Search placement inside CyberSynchrony

Kitsune Search functions as an autonomous external module, thereby providing the CYberSynchrony platform with advanced search capabilities across a broad spectrum of cyber threat intelligence sources.

In contradistinction to conventional methodologies, which necessitate uninterrupted data ingestion, Kitsune Search provides an API-based interface, thereby empowering CYberSynchrony platform to execute meticulously targeted queries as required; this approach facilitates the platform's capacity to investigate issues such as vulnerability exploitation, data leaks, and suspicious activity on the dark web or underground forums involving key internal assets.

This approach is designed to ensure the continued efficacy and precision of intelligence gathering operations.

4.3 Technologies Used

The service is classified as external components. Kitsune Search provides a **REST API** that exchanges data in **JSON format,** it does not provide a user interface.

The tool is primarily focused on searching for underground mentions, botnet exposures, and data leaks, enhancing cybersecurity threat intelligence.

Table 6: Kitsune Monitoring technology stack.

Technology	Purpose
REST	RESTful APIs use standard HTTP methods for CRUD (Create, Read, Update, Delete) operations

© CYberSynchrony, 2024

5 Unalterable Forensic Techniques adaptation(UFT) (BYTE/MAG)

5.1 Component Description

The Unalterable Forensic Techniques (UFT) adaptation in CYberSynchrony is designed to guarantee the integrity and immutability of security logs throughout their entire lifecycle. It ensures that collected forensic evidence remains trustworthy, verifiable, and admissible for further analysis or legal proceedings. The core component enabling UFT is the Data Preparation Module (DPM). This Python-based script acts as a gateway between the on-premises cybersecurity monitoring tools (e.g., Suricata) and the cloud storage environment. The DPM processes log batches by:

- Computing a SHA-256 cryptographic hash for each batch,
- Storing the hash onto the project's permissioned blockchain (Hyperledger Fabric),
- Uploading the original log batch securely to the cloud storage.

This guarantees that every log batch has a verifiable, tamper-evident proof anchored in the distributed ledger. Any entity retrieving logs in the future can recompute the hash and query the blockchain to ensure the data has not been altered.

The UFT aligns directly with CYberSynchrony's focus on secure, privacy-preserving cyber defence mechanisms, providing a chain-of-custody model for forensic data, crucial for incident handling, post-event analysis, and compliance with cybersecurity regulations.

Table 7: UFT alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 3 - Strengthen the integrity of digital evidence during threat assessments.	UFT ensures that every monitored event and collected log can be cryptographically validated and hashed for integrity and authenticity.

5.2 Architecture

The architecture of the UFT adaptation centres around the Data Preparation Module interacting with the local monitoring stack, the distributed ledger, and the cloud storage as depicted in Figure 4.

© CYberSynchrony, 2024

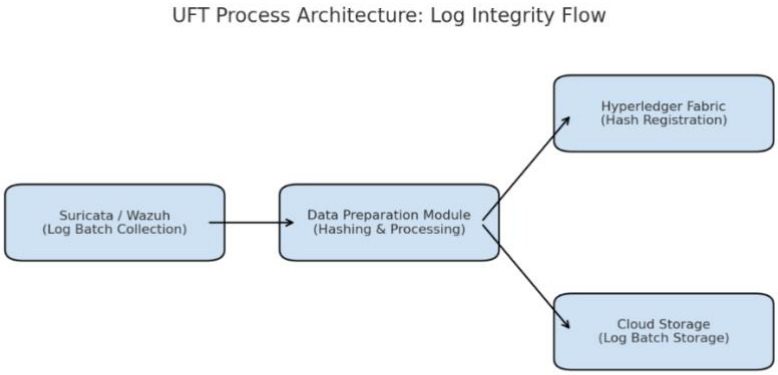


Figure 4: UFT integrity flow architecture.

Data Flow Steps:

1. **Log Collection**: Suricata (and/or Wazuh) agents produce batched logs of security events.
2. **Hash Computation**: The DPM retrieves these batches and computes a **SHA-256** hash for each file.
3. **Hash Registration**: Using secure APIs, the DPM submits the hash to the **Hyperledger Fabric** ledger as a transaction, creating a permanent, timestamped, tamper-evident record.
4. **Secure Upload**: The DPM uploads the full log batch to the Cloud Storage for retention and future analysis.
5. **Verification Workflow**: Later, when accessing a batch, the system recomputes the SHA-256 hash and queries the blockchain to validate the authenticity.

Functionalities and Cybersecurity Value:

- **Evidence Preservation**: Prevents accidental or malicious tampering of forensic evidence.
- **Traceability**: Ensures every collected event has an immutable audit trail.
- **Compliance Support**: Facilitates adherence to legal and regulatory requirements regarding incident handling and forensic evidence.

Unlike traditional log archiving, UFT combines **distributed ledger technology** with **real-time batch processing**, providing cryptographic evidence of data integrity without relying on trusted third parties. The modular and lightweight design (Python script + Hyperledger Fabric backend) makes it deployable in a wide range of infrastructures while maintaining strong forensic guarantees.

5.3 Technologies Used

he DPM was implemented using robust, industry-standard technologies optimized for cybersecurity monitoring and blockchain-based proof recording.

© CYberSynchrony, 2024

Table 8: UFT technology stack.

Technology	Purpose
Python	Main programming language for the Data Preparation Module
Requests (Python Library)	Secure API communication with blockchain network (Hyperledger Fabric)
hashlib (Python Standard Library)	Computation of SHA-256 hashes for log batches
Hyperledger Fabric SDK for Python (hfc/fabric-sdk-py)	Interaction with the distributed ledger for hash registration
Suricata Integration (custom script/API)	Automated retrieval of collected log batches

© CYberSynchrony, 2024

6 Conclusions

This deliverable, D3.1 "Enhanced Threat Intelligence System v1," presented the initial development and integration of three critical cybersecurity components within the CYberSynchrony platform: MISP, Kitsune Search, and the UFT adaptation. Each of these components addresses specific cybersecurity challenges and directly contributes to fulfilling CYberSynchrony's strategic objectives.

The MISP platform enables the structured aggregation, enrichment, and sharing of CTI, enhancing the project's real-time threat assessment capabilities and supporting proactive defence across diverse organizational environments. Kitsune Search extends this threat landscape understanding by offering a focused, asset-specific search service, enabling the retrieval of intelligence related to vulnerabilities, underground activities, and exposed credentials. Meanwhile, the UFT approach introduces an innovative chain-of-custody process, leveraging blockchain technologies to ensure the immutability and verifiability of forensic evidence across the platform's security monitoring infrastructure.

Together, these developments significantly strengthen CYberSynchrony's ability to deliver trustworthy, actionable, and privacy-respecting cybersecurity services, while ensuring traceability and integrity across all data and processes.

The next phases of the component's development effort will focus on the seamless integration and interoperability of these components within a unified CYberSynchrony platform. Particular attention will be given to ensuring that this integration does not disrupt existing cybersecurity monitoring and incident response processes within end-user environments. Instead, CYberSynchrony aims to provide an added value layer of intelligence, validation, and risk evaluation that can operate alongside and enhance existing systems without introducing operational friction.

© CYberSynchrony, 2024

7 References

[1] MISP Project. (2024). User guide of MISP intelligence sharing platform (Version 2.4). Computer Incident Response Center Luxembourg (CIRCL). https://www.circl.lu/doc/misp/
[2] https://github.com/MISP/MISP
[3] https://github.com/MISP/MISP?tab=AGPL-3.0-1-ov-file#readme
[4] Author, Title, Date, URL or page address
[5] https://www.lfdecentralizedtrust.org/projects/fabric

© CYberSynchrony, 2024

Annex I: *Preliminary PoC and Prototypes (MISP)*

A default instance of MISP has been successfully deployed within the IANUS local infrastructure as part of T3.1. This preliminary setup includes the standard MISP package installed on a virtual machine running Ubuntu Server, configured with MariaDB, Apache2, and the MISP web application stack. The platform is currently hosted on a local VM inside the IANUS lab environment, accessible via a secure internal network interface. This deployment serves as a baseline proof-of-concept for testing MISP's core threat intelligence functionalities, including manual event creation, indicator tagging, taxonomy use, and REST API testing. The source code for MISP can be found in the official GitHub repository (see [2]).

The MISP interface is fully operational and accessible through a web browser, enabling IANUS analysts to simulate key workflows such as creating new events, adding IoCs, and enriching data using built-in Galaxy clusters and Taxonomies. The prototype is unmodified at this stage, presenting the out-of-the-box user experience with full access to tools like object templates, sightings, correlation mechanisms, and data export features. Analysts can also test integrations using the available REST API and PyMISP client. This foundational deployment is intended for demonstration, testing, and familiarization purposes, and will later be extended to support federation with other MISP instances, automated data ingestion from Suricata and Wazuh, and integration into the wider CYberSynchrony threat intelligence ecosystem.

In the context and scenarios of the project, MISP operates as the central threat intelligence platform that ingests structured data (IoCs, threat actor profiles, metadata) from both external feeds and internal telemetry sources (e.g., Wazuh, Suricata). The scenario assumes a federated environment where MISP instances across Member States or sectors (e.g., public hospitals, universities) exchange relevant threat intelligence. This data is categorized, tagged, and enriched using taxonomies, galaxies, and correlation logic, allowing other components in the architecture to consume high-value CTI via REST APIs. This enables real-time alert enrichment, risk assessment, and synchronized response across cross-border cybersecurity entities.

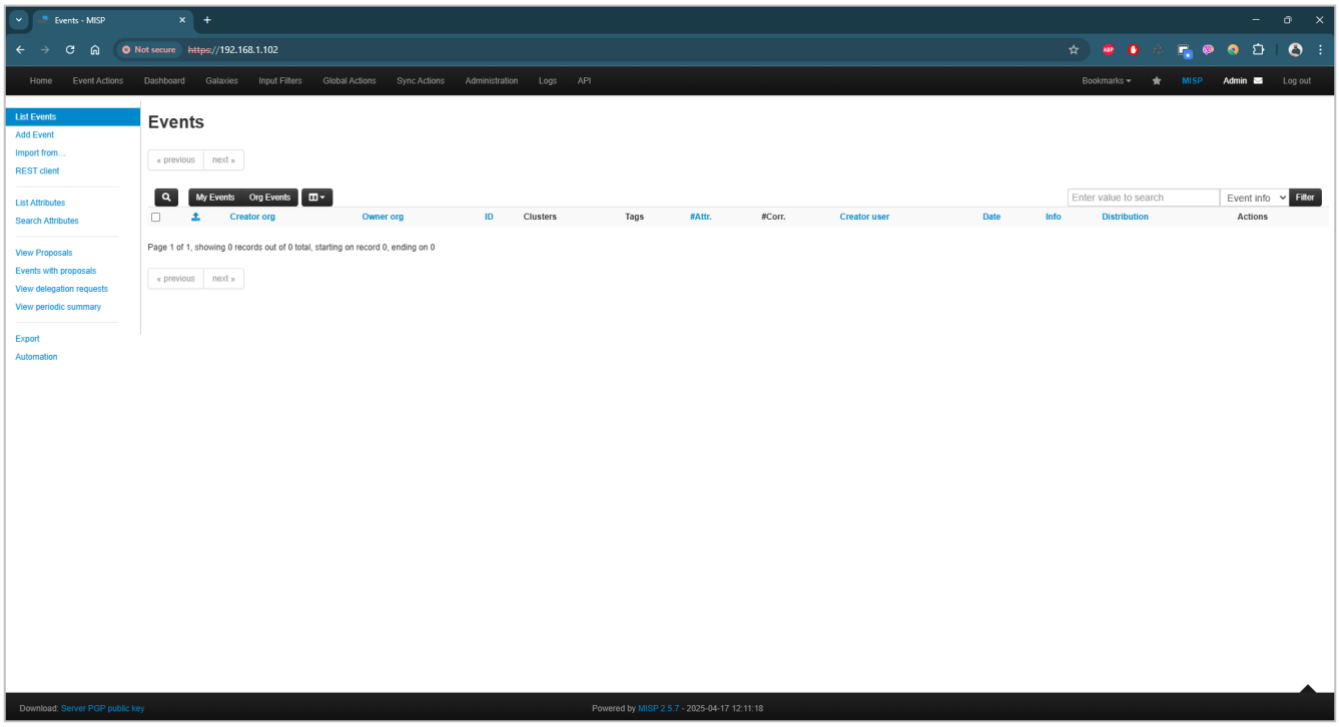


Figure 5: MISP Home page

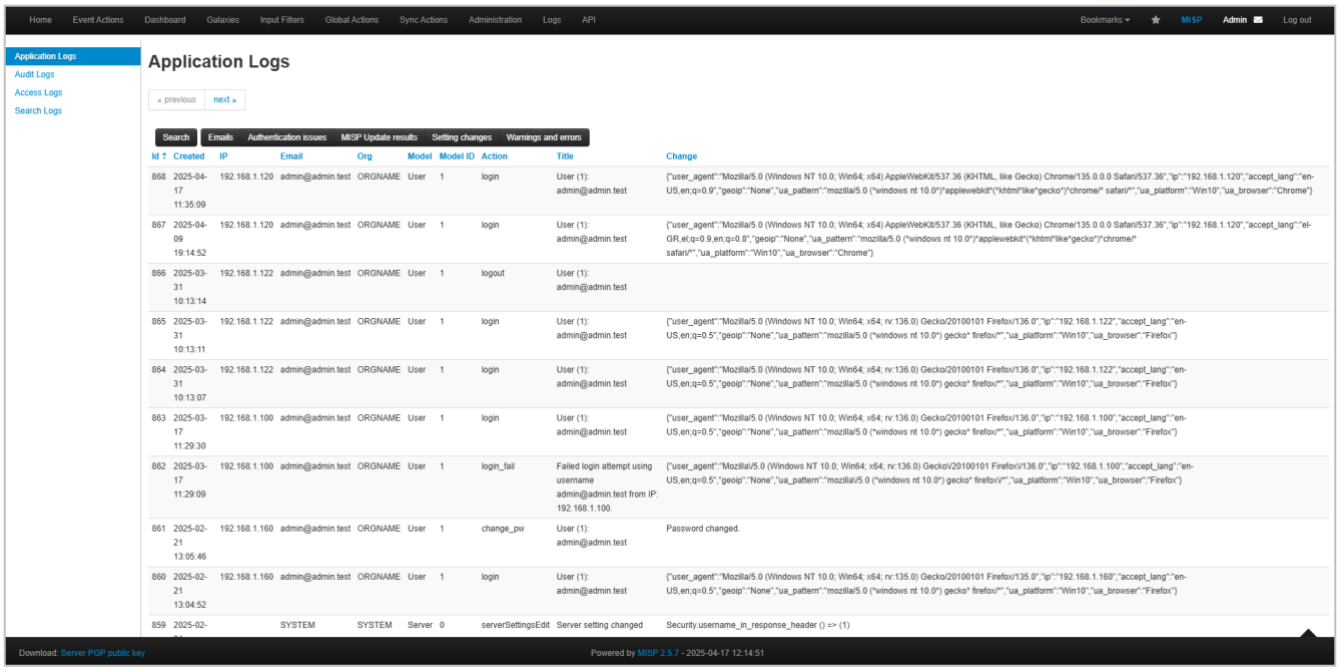


Figure 6: Application Logs

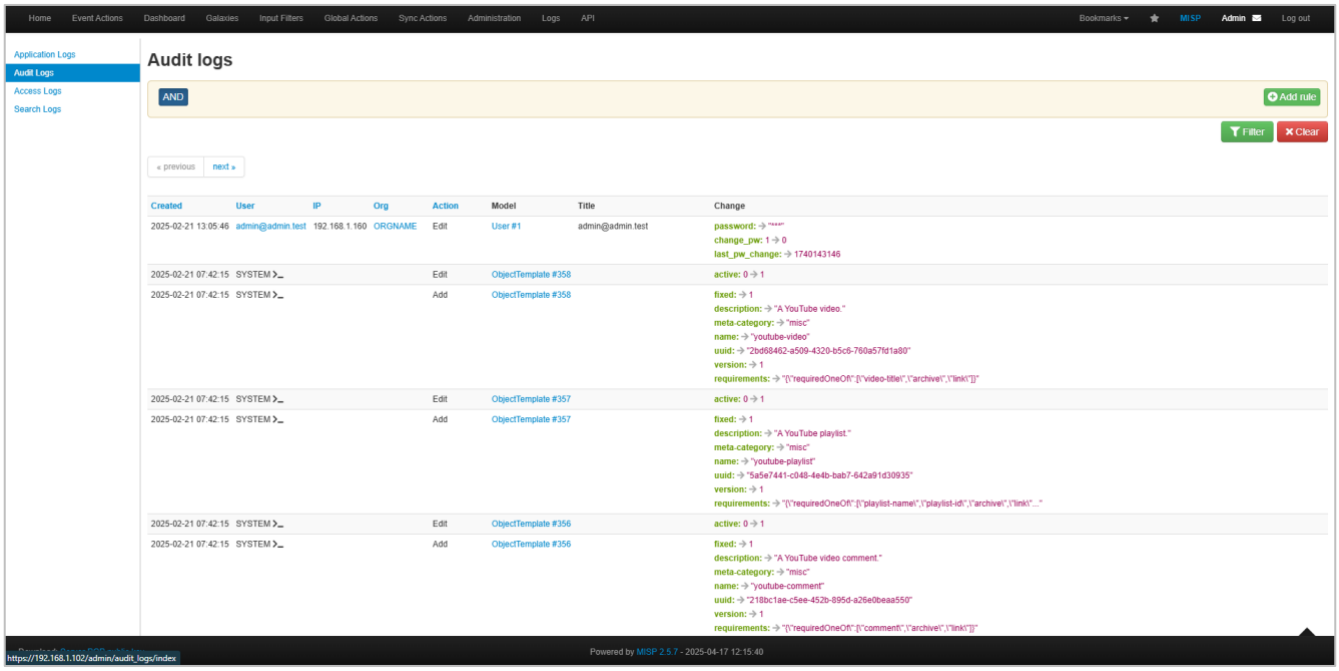


Figure 7: Audit logs

API Underground

Endpoint	Method	Description
`/events`	GET	Retrieve a list of all events with metadata.

/events/view/{id}	GET	Fetch detailed information about a specific event by its ID.
/events/add	POST	Create a new event with associated attributes and metadata.
/events/edit/{id}	POST	Modify an existing event's details.
/events/delete/{id}	POST	Remove an event from the MISP instance.
/attributes/restSearch	POST	Search for attributes using filters like value, type, category, tags, or time range.
/events/restSearch	POST	Search for events based on various criteria, including tags and timestamps.
/events/stix/download	GET/POST	Export events in STIX format for interoperability with other CTI platforms.
/events/nids/suricata/download	GET	Export IDS rules in Suricata format based on event attributes.
/events/hids/md5/download	GET	Export MD5 hashes for Host-based Intrusion Detection Systems.
/attributes/add	POST	Add a new attribute to an existing event.
/attributes/edit/{id}	POST	Update details of a specific attribute.
/attributes/delete/{id}	POST	Delete an attribute from an event.
/sightings/add	POST	Submit a sighting to indicate observation of an attribute in the wild.
/feeds/index	GET	List all configured feeds available for synchronization.
/feeds/fetchFromFeed/{id}	POST	Manually fetch data from a specific feed by its ID.
/tags/index	GET	Retrieve all tags defined in the MISP instance.

© CYberSynchrony, 2024

`/galaxies/index`	GET	List all galaxy clusters (e.g., threat actors, malware families) available.
`/warninglists/index`	GET	Retrieve all warning lists used for attribute validation and filtering.

For example, to search for attributes related to a specific IP address, you can use the following POST request:

```
curl -X POST -H "Authorization: YOUR_API_KEY" -H "Accept: application/json" -H "Content-Type: application/json" \
https://<misp_url>/attributes/restSearch \
-d '{"value": "192.168.1.1"}'
```

This request will return all attributes matching the IP address 192.168.1.1, facilitating integration with detection systems like Suricata or Wazuh.

In another cause, we might use the Attribute search of MISP. To this end, if we hit the API endpoint **POST /attributes/restSearch,** with the value of "1.54.137.170" in the request payload body, the response could look like this


```
{
 "response": {
  "Attribute": [
   {
    "id": "39634",
    "event_id": "148",
    "object_id": "0",
    "object_relation": null,
    "category": "Network activity",
    "type": "ip-src",
    "to_ids": true,
    "uuid": "571f4d47-9940-4638-91d4-4f92950d210f",
    "timestamp": "1461669191",
    "distribution": "5",
    "sharing_group_id": "0",
    "comment": "spambot (originating SMTP connection)",
    "deleted": false,
    "disable_correlation": false,
    "first_seen": null,
    "last_seen": null,
```

```
      "value": "1.54.137.170",
      "Event": {
       "org_id": "1",
       "distribution": "3",
       "id": "148",
       "info": "Malspam (2016-04-26) - Dridex botnet 122",
       "orgc_id": "7",
       "uuid": "571f489a-26e4-403c-948f-4aff950d210f"
      }
     }
   ]
  }
}
```


This response indicates that the IP address 1.54.137.170 was observed as a source IP in a spam campaign associated with the Dridex botnet. The attribute is categorized under "Network activity" and is linked to event ID 148. Such detailed information facilitates automated threat intelligence processing and integration with detection systems. In CYberSynchrony, enriched MISP data like the Dridex-linked IP enables real-time threat detection updates in Wazuh/Suricata and risk scoring in HRM/CSPA. It also supports cross-border sharing and evidence integrity via DLT, ensuring collaborative, actionable defense.

More information on the above API requests can be found in the official guide: https://misp.gitbooks.io/misp-book/content/automation/.

© CYberSynchrony, 2024

Annex II: *Preliminary PoC and Prototypes (Kitsune Search)*

Kitsune Search is designed as an autonomous standalone module that empowers the CYberSynchrony platform with advanced, on-demand search capabilities across diverse cyber threat intelligence sources.

Unlike systems that require persistent data ingestion, Kitsune Search operates via a RESTful API, enabling targeted, real-time queries for:

- Data leaks and credential exposures
- Mentions of internal assets in underground and dark web forums
- Botnet and infostealer activity related to specific entities

This approach supports efficient, focused intelligence gathering and proactive risk detection, minimizing operational overhead and ensuring timely response to emerging threats.

Each of these RESTful API calls leverages standard HTTP(S) methods and returns data in a machine-readable JSON format, facilitating easy integration with SIEM, SOAR, or custom security tooling within the CYberSynchrony ecosystem.

Each type of RESTful API provided by Kitsune Search, including authentication, underground search, data leak search, infostealer (botnet) search, and threat intelligence feeds, is described in detail in the following sections of this annex, with an explanation of their HTTP(S) endpoints, request methods, and operational context

YAML File (Kitsune Monitoring, Search)

The attached YAML file contains the API structure for utilizing the Kitsune Monitoring (alert) and Kitsune Search services. It provides the necessary endpoints, parameters, and request formats required to interact with these services, enabling seamless integration and efficient data monitoring and searching.

Mock Prototype / Demo

A mock test environment with fictitious data is provided, allowing users to interact and work on API integration.

While this environment is exposed and accessible via the internet, access requires specific IP addresses to be allowed through the external firewall to ensure proper connectivity.

Authentication/Authorization API

The login process uses a **Bearer Token** authentication method.

To authenticate, the user must send a request to the authentication endpoint with valid credentials. Upon successful authentication, the system returns an access token, which must be included in the *Authorization header* of subsequent API requests in the format: **Authorization: Bearer <token>**.

© CYberSynchrony, 2024

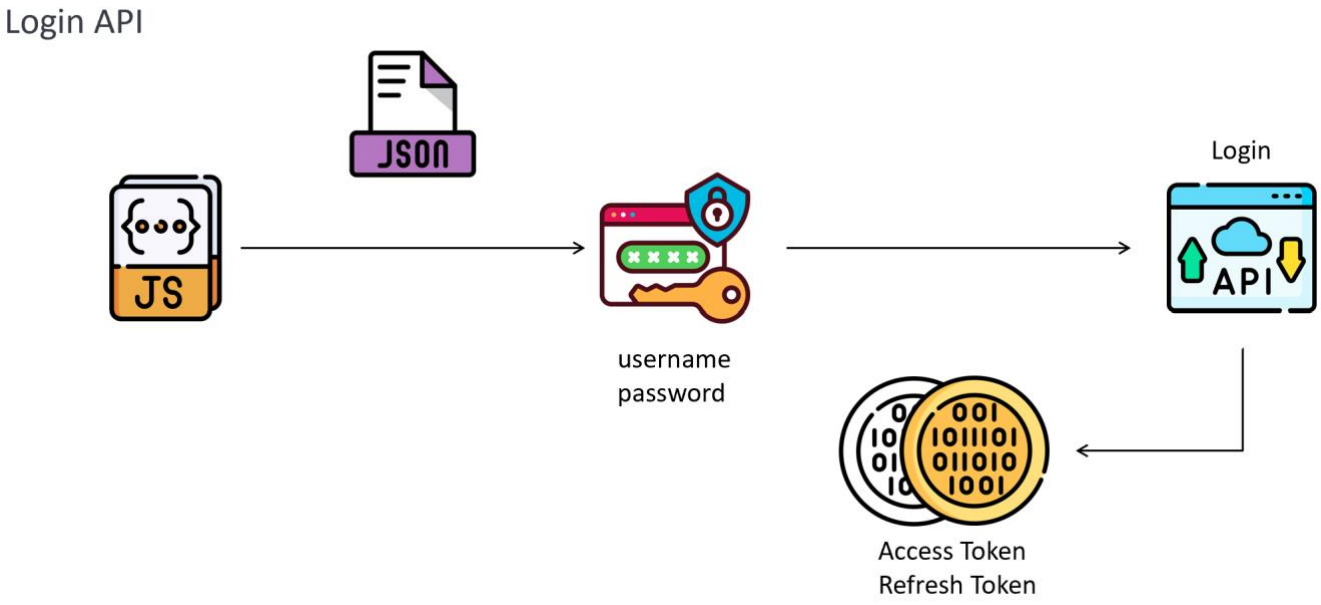


Figure 8: Diagram illustrating the API login process

This token grants access to the protected resources for a limited period (**20 minutes**), after which it may need to be refreshed or renewed.

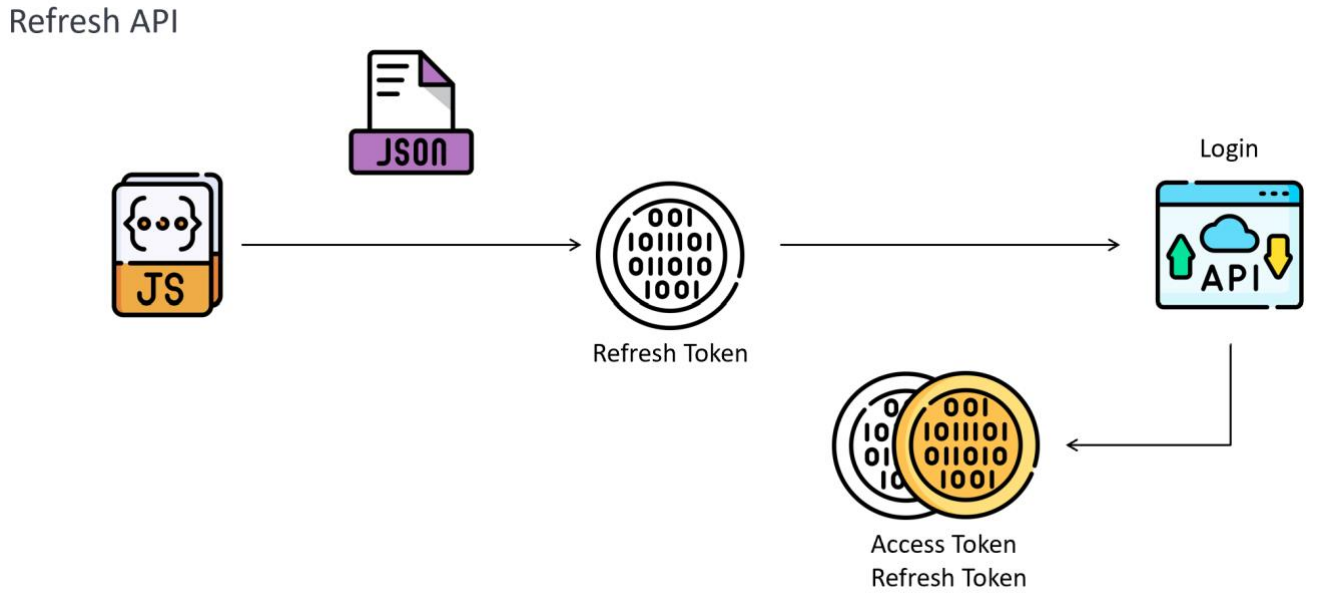


Figure 9: Diagram illustrating the API refresh authorization process

This mechanism ensures that only authorized users can interact with sensitive monitoring resources, reducing the risk of unauthorized access

Search API (Underground)

This image represents the workflow and response structure for the underground search API.

It reports a risk metric for each evidence as “hackishness”, that represents a confidence score quantifying how likely content relates to malicious activity.

© CYberSynchrony, 2024

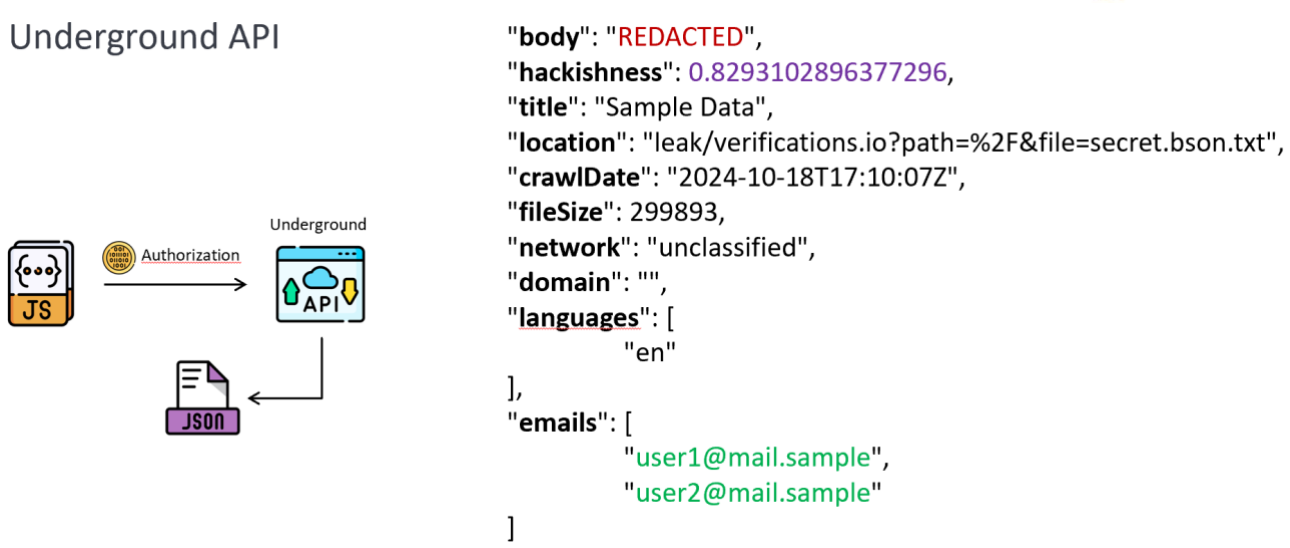


Figure 10: Representation of "underground search" functionality with response

Table 9: Kitsune Search - API Underground list

Method	Endpoint	Description
POST	/api/do/v2/lookup	Generic Search
POST	/api/do/entity/bank	Entity Bank Bin
POST	/api/do/entity/card	Entity Credit Card
POST	/api/do/entity/crypto	Entity Cryptocurrency Address
POST	/api/do/entity/email-address	Entity Email Address
POST	/api/do/entity/email-domain	Entity Email Domain
POST	/api/do/entity/ip	Entity IP Address

This set of APIs provides various search and lookup functionalities for different types of entities.

The POST method at **/api/do/v2/lookup** performs a generic search across available data.

The other POST endpoints are specialized for specific entity types: **/api/do/entity/bank** looks up bank BIN information, **/api/do/entity/card** handles credit card data, and **/api/do/entity/crypto** searches for cryptocurrency addresses.

© CYberSynchrony, 2024

Additionally, **/api/do/entity/email-address** and **/api/do/entity/email-domain** are used to query email addresses and domains, respectively. Lastly, **/api/do/entity/ip** retrieves data related to a specific IP address.

RAW Request/Response sample

RAW Request

```
POST /api/do/v2/lookup?offset=0&sort=h HTTP/2
Host: <endpoint-domain>
Content-Length: 17
Accept: application/json, text/plain, */*
Content-Type: application/json
Authorization: Bearer 4uthb34r3r
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

{"search":"test"}
```

RAW Response

```
HTTP/2 200 OK
<response headers>

{ "results": [
    {
        "id": "4f00f6dcf7ecc55d602833aec0b60cf5dd7e1ea2",
        "body": "Fraudnation\n(Type: group  Members: 3,847  ID: -100",
        "hackishness": 1,
        "title": "",
        "uri": "https://t.me/fraudn/2614451",
        "url": "https://t.me/fraudn/2614451",
        "location": "https://t.me/fraudn/2614451",
        "crawlDate": "2024-08-21T12:43:19Z",
        "fileSize": 37107,
        "network": "telegram",
        "domain": "t.me",
        "languages": [
            "en"
        ],
```

© CYberSynchrony, 2024

```
        "emails": null,
        "groups": null
    },
    {
        "id": "7e490896914d73de7a84cec135af4f8b6f0e56dc",
       "body": "LoliPoxn Main\nLogin\nBuy\nGallery\nVideo\nTags\nContact",
        "hackishness": 0.14975014329622746,
        "title": "Underage Sisters Video | LoliPorn",
        "uri": "http://xxxcpczy47jkaz7lxnid6fbs64lzt5qurrcm5tdprwf",
        "url": "http://xxxcpczy47jkaz7lxnid6fbs64lzt5qurrcm5tdprwf",
        "location": "http://xxxcpczy47jkaz7lxnid6fbs64lzt5qurrcm5tdprwf",
        "crawlDate": "2024-08-21T12:43:12Z",
        "fileSize": 53252,
        "network": "onion.v3",
        "domain": "xxxcpczy47jkaz7lxnid6fbs64lzt5qurrcm5tdprwf3qqfagh",
        "languages": [
            "en"
        ],
        "emails": null,
        "groups": null
    }
],
"message": null,
"total" :1245
}
```


Search API (Data Leak)

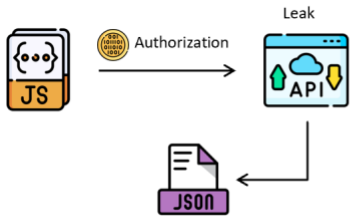
The Data Leak Search API enables users to discover and investigate live data leaks through both POST and GET HTTP methods.

The /api/i/3/lookup/live POST endpoint allows users to search for ongoing leaks, while the GET endpoints (/api/i/3/lookup/live/{key} and /api/i/view/{bucket}/{systemid}) provide detailed access to specific leak records or data within defined storage buckets.

These calls deliver actionable intelligence on compromised data, supporting rapid incident triage and exposure assessment for affected assets or credentials

© CYberSynchrony, 2024

Leak API



"**email**": "michele.test@gmail.com",
"**type**": "",
"**crawlDate**": "2024-12-29T21:43:36Z",
"**location**": "https://craxpro.io/threads/dazn-premium-accounts.1557296/",
"**fragment**": "...michele.test@gmail.com:Password.1...",
"**network**": "unclassified",
"**leak**": "sample.com"

Figure 11: Representation of "Leak search" functionality with response

Table 10: Kitsune Search - API Data Leak list

Method	Endpoint	Description
POST	/api/i/3/lookup/live	Search Leaks Live
GET	/api/i/3/lookup/live/{key}	Fetch Leaks Live
GET	/api/i/view/{bucket}/{systemid}	View

The table outlines three API methods related to accessing and managing live data leaks.

The POST **/api/i/3/lookup/live** endpoint is used to search for live leaks, allowing users to query current data.

The GET **/api/i/3/lookup/live/{key}** endpoint retrieves specific leak data based on a unique key.

Lastly, the GET **/api/i/view/{bucket}/{systemid}** endpoint enables users to view detailed information within a specified data bucket and system ID.

RAW Request/Response sample

RAW Request

```
GET /api/cs/alerts?limit=10&offset=0 HTTP/2
Host: <endpoint-domain>
Accept: application/json, text/plain, */*
Authorization: Bearer b34r3rt0k3n
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

© CYberSynchrony, 2024

RAW Response
HTTP/2 200 OK <response headers>

Search API (Infostealer)

The Infostealer Search API offers a set of endpoints for querying and retrieving information related to botnet and infostealer activity.

A POST request to /api/i/bots initiates a search for bot-related data, while GET requests to /api/i/bots/{indexfile} and /api/i/bots/{indexfile}/download return detailed hierarchical structures or allow for the download of entire data trees for offline analysis.

These API calls are essential for tracking infostealer campaigns, mapping botnet infrastructure, and correlating compromised endpoints within an organization's environment

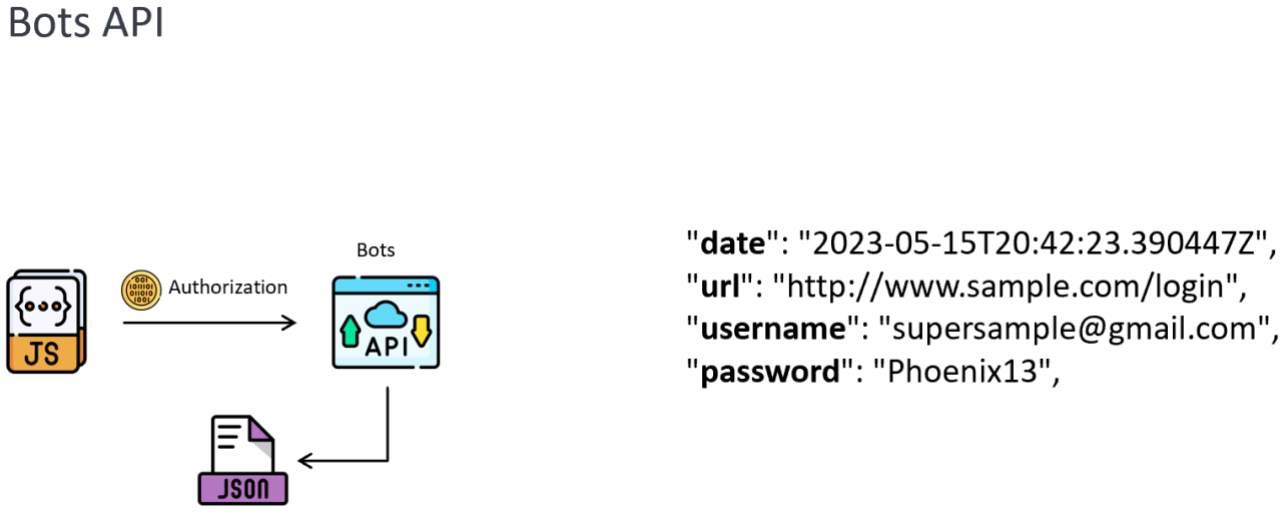


Figure 12: Representation of "botnet (infostealer) search" functionality with response

Table 11: Kitsune Search - API Botnet Infostealer list

Method	Endpoint	Description
POST	/api/i/bots	Search Bots
GET	/api/i/bots/{indexfile}	Tree
GET	/api/i/bots/{indexfile}/download	Tree Download

These search APIs provide various functions for interacting with bot-related data.

The **POST** method at the **/api/i/bots** endpoint allows for searching bots within the system.

© CYberSynchrony, 2024

The **GET** method at **/api/i/bots/{indexfile}** returns a tree structure of the data related to a specific index file.

Finally, the **GET** method at **/api/i/bots/{indexfile}/download** enables downloading the corresponding tree structure, which is useful for offline analysis or archiving.

RAW Request/Response sample

RAW Request

```
POST /api/i/bots HTTP/2
Host: <endpoint-domain>
Accept: application/json, text/plain, */*
Authorization: Bearer b34r3rt0k3n
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

{
  "query": "test",
}
```

RAW Response

```
HTTP/2 200 OK
<response headers>

{
    "results": [
        {
            "date": "2024-08-15T15:09:41.257094Z",
            "url": "httpsrosoft.com/oauth2/v2.0/authorize",
            "username": "lucia.samasd.it",
            "password": "asd",
            "indexfile":
"4311aba25583e4413fe386852cba1460728331c80e5571a3f7af8fd2471c9963f65c40f4
8afb9adc0ef1c6431029b1abee5d8d68446d0ea669b2844498bbf78c",
            "representative": {
                "name": "lucia.sampasdt"
            }
        },
        {
            "date": "2024-08-15T15:09:41.257094Z",
```

© CYberSynchrony, 2024

```
            "url": "https://netasaas.b2c2/v2.0/authorize",
            "username": "robasd",
            "password": "asd",
            "indexfile":
"4311aba25583e4413fe386852cba1460728331c80e5571a3f7af8fd2471c9963f65c40f4
8afb9adc0ef1c6431029b1abee5d8d68446d0ea669b2844498bbf78c",
            "representative": {
                "name": "roberto.fasd"
            }
        }
    ]
}
```


Feeds (IOCs)

The Feeds API consists of multiple GET endpoints that provide access to real-time threat intelligence feeds, such as /api/feeds/ransomware_url, /api/feeds/apt_hash, /api/feeds/apt_ip, /api/feeds/apt_url, and /api/feeds/botnet_cnc_url.

Each endpoint delivers a specific category of indicators of compromise (IOCs), including ransomware URLs, APT-related hashes, IPs, and URLs, as well as botnet command-and-control addresses and other malicious indicators.

These feeds are designed for seamless integration into security monitoring and enrichment workflows, enabling automated threat detection and proactive defense.

Table 12: Kitsune Search - API Feeds list

Method	Endpoint	Description
GET	/api/feeds/ransomware_url	Ransomware URL
GET	/api/feeds/apt_hash	APT Hash
GET	/api/feeds/apt_ip	APT IP
GET	/api/feeds/apt_url	APT URL
GET	/api/feeds/botnet_cnc_url	Botnet CnC URL

© CYberSynchrony, 2024

GET	/api/feeds/ip_reputation	IP Reputation
GET	/api/feeds/malicious_hash	Malicious Hash
GET	/api/feeds/malicious_url	Malicious URL
GET	/api/feeds/phishing_url	Phishing URL

This collection of API endpoints provides access to various cybersecurity threat intelligence feeds. Each GET method retrieves a specific type of threat data.

For example, **/api/feeds/ransomware_url** returns URLs associated with ransomware, while **/api/feeds/apt_hash**, **apt_ip**, and **apt_url** provide hashes, IPs, and URLs linked to Advanced Persistent Threats (APTs).

The **/api/feeds/botnet_cnc_url** endpoint lists known Botnet Command and Control (CnC) URLs.

Additional endpoints deliver data on IP reputation, malicious hashes, malicious URLs, and phishing URLs.

RAW Request/Response sample

RAW Request

```
GET /api/feeds/ransomware_url?search=&offset=0&limit=10&ordering=-last_seen HTTP/2
Host: <endpoint-domain>
Accept: application/json, text/plain, */*
Authorization: Bearer b34r3rt0k3n
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

RAW Response

```
HTTP/2 200 OK
<response headers>

{
    "count": 2,
    "next": "http://fake/api/feeds/crimeware_url?search=&offset=-10&limit=10&ordering=",
```

© CYberSynchrony, 2024

```
    "previous":
"http://fake/api/feeds/crimeware_url?search=&offset=10&limit=10&ordering=
",
    "results": [
        {
            "id": 117302972,
            "mask": "zp.kosmohubeq.space",
            "type": 2,
            "first_seen": "2024-06-22T13:33:00Z",
            "last_seen": "2024-08-21T08:16:00Z",
            "popularity": 5,
            "geo": "ru, by, kz, ua, uz, tm, am, md, kg, tj",
            "IP": "172.67.143.137, 104.21.47.3, 45.84.0.26, 1.8.75.2,",
            "files": [
                {
                    "MD5": "82143033173CBEEE7F559002FB8AB8C5",
                    "SHA1": "E03AEDB8B9770F899A29F1939636DB43825E95CF",
                    "SHA256":
"4AE110BB89DDCC45BB2C4E980794195EE5EB85B5261799CAEDEF7334F0F57CC4",
                    "threat": "Trojan-Ransom.Win32.Medusa.aj"
                }
            ],
            "whois": {
                "domain": "kosmohubeq.space",
                "created": "23.05.2024",
                "updated": "12.06.2024",
                "expires": "23.05.2025",
                "org": "Privacy service provided by Withheld for Privacy
ehf",
                "country": "Iceland",
                "email": "please query the rdds service of the registrar
of record identified in this output for information o",
                "registrar_name": "NameCheap, Inc.",
                "NS": "pat.ns.cloudflare.com, terin.ns.cloudflare.com",
                "NS_ips":        "108.162.192.139,        108.162.193.236,
172.64.32.139, 172.64.33.236, 173.245.58.139, 173.245.59.236"
            }
        },
```

```
        {
            "id": 108014137,
            "mask": "bradford-marine.com",
            "type": 1,
            "first_seen": "2024-05-21T13:03:00Z",
            "last_seen": "2024-08-21T07:07:00Z",
            "popularity": 2,
            "geo": "ru",
            "files": [
                {
                    "MD5": "86116AA19D3CA753B252AF42BE007FD5",
                    "threat": "HEUR:Trojan.Win32.Generic"
                },
                {
                    "MD5": "BB42B1918E08CEC385B6FAEA3EABC2A5",
                    "SHA1": "ACE79DB9609A9F3614DA4C527C672F73A7D4E138",
                    "SHA256":
"626DCE586D9FB4782467DC6C9A331C09E7D4577E04E3DEA3108022E206C0ADB7",
                    "threat": "HEUR:Trojan.Win32.Generic"
                },
                {
                    "MD5": "0F5606A79F69FFB88C5F99070C0E9E53",
                    "SHA1": "1E50FF9CB27B2BF226C04938C18FB1DAC946560B",
                    "SHA256":
"832D8F12843EA1591A40557345C4CED1B54D8DF5AE9B1D4DB7118240E852C0D2",
                    "threat": "HEUR:Packed.Win32.Mentiger.gen"
                },
                {
                    "MD5": "977CDD6422A21D3B0FD1E697C84FB559",
                    "SHA1": "403B814781CE79067A46B82396055A651221CAB8",
                    "SHA256":
"E9BFCABA4F87E0C12713900BD5797C2682A4BEFC58425AAF8F589D92A3ED69E8",
                    "threat": "HEUR:Trojan.Win32.Generic"
                },
                {
                    "MD5": "23CD9EDBBF2220D31F923185A47437ED",
                    "threat": "Trojan-Ransom.Win32.Cryptodef.ablf"
```

```
                },
                {
                    "MD5": "534A87F85A91B0389613A0D881BEF67D",
                    "SHA1": "6CD76B51E3D5326DC95FD3C60BE15C4E7056945A",
                    "SHA256":
"B65D53A0046109359D33F9C1B9BC4DFB71C0946E84C016517ECCEAC97A99ECD6",
                    "threat": "Trojan-Ransom.Win32.Cryptodef.ablf"
                },
                {
                    "MD5": "E2B022A5AB75894F4D1909846BF44ED8",
                    "SHA1": "943678D32D66D927F274CBF6F418CABD199332F6",
                    "SHA256":
"98C969873CCA529E03C2A28AEC5D7C646BBE589EF76479E39585253F3915BA4C",
                    "threat": "Trojan-Ransom.Win32.Cryptodef.ablf"
                },
                {
                    "MD5": "226458D7F3E558980CD2A06E5AC14617",
                    "threat": "Trojan.Win32.Yakes.nskv"
                },
                {
                    "MD5": "DEBA64A54CA8FFB3122F6AC70D9A38F9",
                    "threat": "Trojan.Win32.Yakes.nskv"
                },
                {
                    "MD5": "99466ED790747ED50CB8678DD91E1A1D",
                    "SHA1": "0A572941C7341C5019CC3865A667E81FE77499A2",
                    "SHA256":
"2E597CA8ED73D8B1C5DD3B928832B9F44A063A529619F95ABF478213492A6961",
                    "threat": "Trojan.Win32.Yakes.nskv"
                }
            ],
            "whois": {
                "domain": "bradford-marine.com",
                "created": "10.07.1998",
                "updated": "29.12.2023",
                "expires": "09.07.2025",
                "name": "BRADFORD MARINE",
```

```
                "org": "BRADFORD MARINE",
                "country": "United States",
                "city": "FORT LAUDERDALE",
                "email": "mitch.flinn@bradford-marine.com",
                "registrar_name": "Network Solutions, LLC",
                "registrar_email": "domain.operations@web.com",
                "NS": "ns41.worldnic.com, ns42.worldnic.com",
                "NS_ips": "162.159.26.111, 162.159.27.50",
                "MX":  "bradford-marine.com.1.0001.arsmtp.com,  bradford-
marine.com.2.0001.arsmtp.com",
                "MX_ips": "8.19.118.216, 8.31.233.240"
            }
        }
    ]
}
```