



# Sécurisation de l'Enterprise-D

Un projet complet de supervision, monitoring et analyse de sécurité pour protéger l'infrastructure du vaisseau Enterprise-D. Cette présentation détaille les missions accomplies par l'équipe : **Capron Dylan, Domenico Mandolino, Laurent Fauveau et Safia Meradji.**



# Architecture du Projet



## Supervision & Monitoring

Installation de Zabbix pour surveiller CPU, RAM et disque.  
Configuration de Prometheus avec collecte via Node Exporter et création de tableaux de bord Grafana connectés.



## Alertes & Logs

Configuration des triggers Zabbix et installation de l'ELK Stack (Elasticsearch, Logstash, Kibana) avec dashboards et alertes Watcher pour automatiser la détection d'incidents.



## Analyse de Sécurité

Utilisation de Wireshark pour capturer et analyser les paquets réseau, recherche d'événements suspects dans Kibana et préparation de rapports techniques sur les menaces identifiées.



## Réponse & Coordination

Création de playbooks de sécurité avec procédures d'incident, installation de TheHive pour la gestion des incidents et intégration de MISP pour le partage d'loCs.

# Mission 01 : Zabbix - Surveillance Complète

## Installation Serveur

Déploiement de Zabbix Server 7.0 sur Debian 12 avec interface web, base de données MariaDB et agent intégré. Configuration de la connexion à la base de données et démarrage des services.

## Composants Installés

- Zabbix Server avec frontend PHP
- Base de données MariaDB configurée
- Agent Zabbix pour auto-surveillance
- Interface web accessible sur port

80

## Agents Déployés

Installation des agents Zabbix sur les systèmes Linux (Debian) et Windows pour collecter les métriques système. Configuration des templates OS appropriés pour chaque plateforme.

## Métriques Surveillées

- Utilisation CPU et charge système
- Mémoire disponible et utilisée
- Espace disque et I/O
- Processus et services actifs



# Visualisations et Alertes Zabbix



01

## Création de Graphiques

Configuration de graphiques personnalisés combinant CPU, RAM et disque pour une vue globale des ressources. Ajout de widgets dans les tableaux de bord pour un monitoring en temps réel.

02

## Configuration des Triggers

Mise en place de déclencheurs pour CPU >80%, RAM >90% et disque <10% libre. Définition des niveaux de严重性 (Warning, High) selon l'impact.

03

## Actions et Notifications

Configuration du type de média Gmail avec authentification par mot de passe d'application. Création d'actions automatiques pour envoyer des emails lors des alertes critiques.

04

## Tests et Validation

Exécution de stress tests avec stress-ng pour valider les alertes. Vérification de la réception des notifications email et du bon fonctionnement du système d'alerte.

# Mission 02-03 : Prometheus et Grafana



## Prometheus

Installation de Prometheus 2.53.0 avec configuration de scrape jobs pour collecter les métriques.  
Déploiement de Node Exporter sur Linux et Windows Exporter sur Windows pour exposer les métriques système.

## Exporters

Node Exporter (port 9100) pour les systèmes Linux et Windows Exporter (port 9182) pour Windows.  
Configuration des targets dans prometheus.yml pour la collecte automatique des données.

## Grafana

Installation de Grafana avec connexion à Prometheus comme source de données.  
Import de dashboards préconfigurés : Node Exporter Full (ID 1860) pour Linux et Windows Node (ID 21697)

**9090**

**Port Prometheus**

Interface web et API

**9100**

**Node Exporter**

Métriques Linux

**9182**

**Windows Exporter**

Métriques Windows

**3000**

**Port Grafana**

Dashboards visuels

# Mission 04-05 : ELK Stack Complet

## Architecture ELK Stack

Pipeline complet de collecte, traitement, stockage et visualisation des logs

## Indexation et Stockage

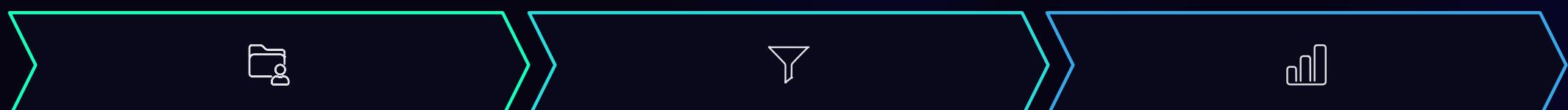
Elasticsearch pour une indexation robuste et une recherche performante

## Ingestion et Traitement

Logstash pour la collecte et l'enrichissement des données en continu

## Visualisation et Alertes

Kibana pour les dashboards, la visualisation et le monitoring



### Elasticsearch

Moteur de recherche et d'analyse distribué version 8.14.3 configuré en mode single-node.

### Logstash

Pipeline de traitement avec entrée syslog UDP 5140, filtres Grok pour RFC5424 et RFC3164, normalisation des timestamps. Sortie vers index quotidien syslog-YYYY.MM.DD dans Elasticsearch.

### Kibana

Interface de visualisation connectée via Service Account Token. Création de data views, dashboards avec volume de logs, top programs, top hosts et derniers événements en temps réel.

Déploiement complet via Docker Compose avec volumes persistants pour Elasticsearch. Configuration des clés d'encryption pour les alertes Kibana et mise en place de règles Elasticsearch query pour détecter les événements ALERT dans les logs syslog.

# Alertes et Détection ELK

## Configuration des Règles

Mise en place de règles Kibana de type "Elasticsearch query" pour surveiller les logs syslog. Détection automatique des messages contenant "ALERT" ou provenant du programme "CAPTAIN".

## Paramètres de la Règle

- Index surveillé : syslog-\*
- Requête KQL : message:"\*ALERT\*"
- Fréquence : toutes les 1 minute
- Fenêtre temporelle : 5 minutes
- Action : écriture dans index alerts-syslog



## Définition

Établir les concepts et les objectifs fondamentaux du projet.

## Paramétrage

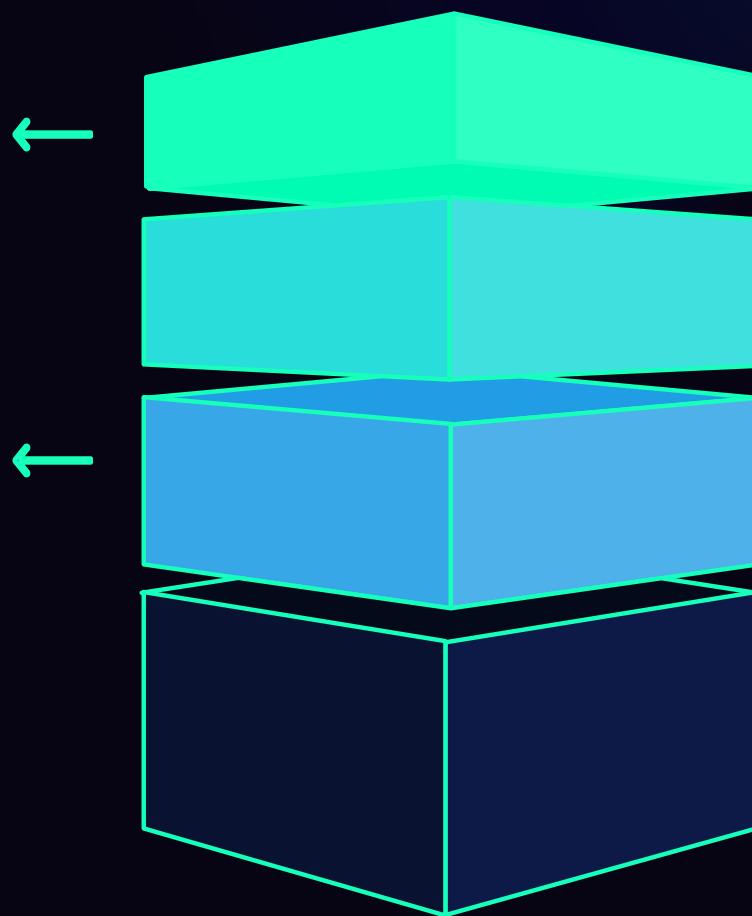
Configurer les systèmes et ajuster les paramètres initiaux.

## Validation

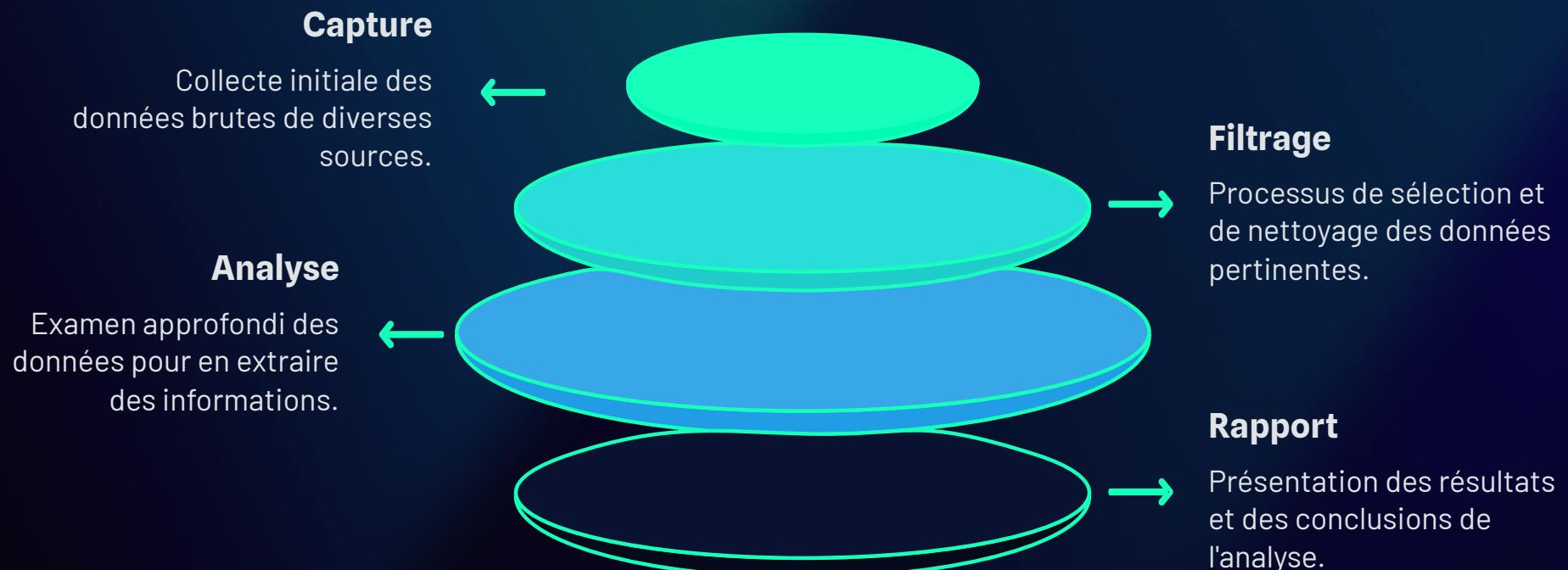
Vérifier la conformité et assurer le bon fonctionnement.

## Maintenance

Assurer le support continu et les mises à jour nécessaires.



# Mission 06-07: Analyse Réseau avec Wireshark



**Installation**  
Déploiement de Wireshark et TShark sur la VM  
Analyse. Configuration des capacités  
**cap\_net\_raw** et **cap\_net\_admin** pour permettre  
la capture sans privilège root.

**Scan Nmap**  
Capture du trafic TCP pendant un scan Nmap  
SYN depuis le Client. Analyse des paquets SYN  
sans ACK pour identifier les tentatives de  
reconnaissance réseau.



**Captures ICMP**  
Tests initiaux avec captures de paquets ping  
entre Client Linux et VM Analyse. Validation du  
fonctionnement avec filtres BPF et  
enregistrement en format pcapng.

**Brute-Force SSH**  
Enregistrement d'une attaque Hydra sur le  
compte testuser. Corrélation des timestamps  
entre captures réseau et logs /var/log/auth.log  
pour prouver l'incident.

La corrélation temporelle entre les paquets SSH capturés et les entrées "Failed password" dans auth.log constitue une preuve robuste d'une attaque par brute-force depuis 192.168.10.112 vers 192.168.10.110.

# Scénarios d'Attaque DéTECTés



## Scan de Ports Nmap

Détection d'un scan SYN sur les ports 22, 80, 443, 9200 et 3306. Les paquets TCP avec flag SYN sans ACK révèlent une tentative de reconnaissance des services actifs sur la VM Analyse.



## Brute-Force SSH

Attaque Hydra avec 912 tentatives de mots de passe sur le compte testuser. Capture de 2547+ trames SSHv2 montrant les échanges Key Exchange Init répétés et les échecs d'authentification.



## Corrélation Logs

Alignement des horodatages entre Wireshark (15:06:47.309) et auth.log (15:06:47 Failed password). Écart <1s prouvant que les paquets capturés correspondent aux tentatives d'authentification échouées.



**Signaux  
Visibles**

**Alertes & Logs**

**Comportement  
s Anormaux**

**Techniques  
d'Attaque**

**Motivations &  
Objectifs**

# Mission 08 : Playbooks et Réponse à Incident



## Détection

Identification de l'incident via alertes ELK, logs anormaux dans Zabbix, ou captures Wireshark.

Surveillance des multiples échecs de connexion SSH dans auth.log et des pics de trafic réseau.



## Confinement

Isolation immédiate de la machine impactée : désactivation de l'interface réseau, blocage IP source via iptables, coupure des accès SSH/RDP. Limitation de la propagation de la menace.



## Éradication

Suppression de la menace identifiée : désinfection malware, application de patches de sécurité, suppression des comptes compromis (testuser), révocation des clés SSH malveillantes.

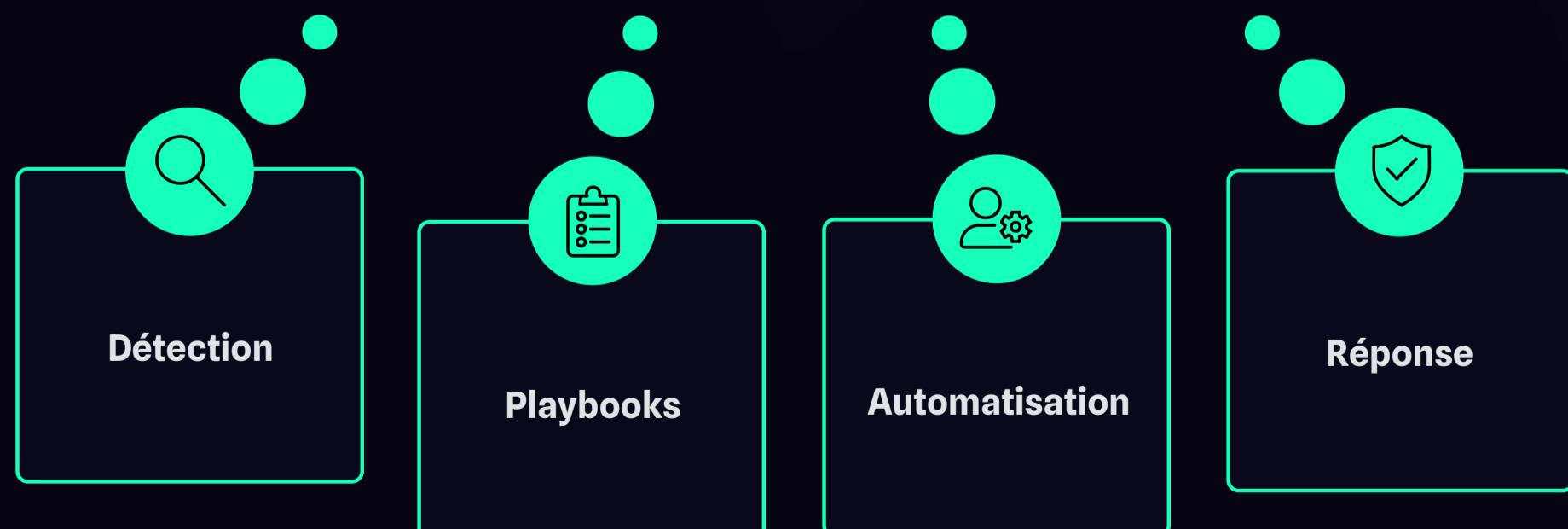


## Récupération

Remise en production sécurisée : restauration depuis sauvegarde, redéploiement VM si nécessaire, relance des services avec monitoring renforcé via Zabbix et Prometheus.

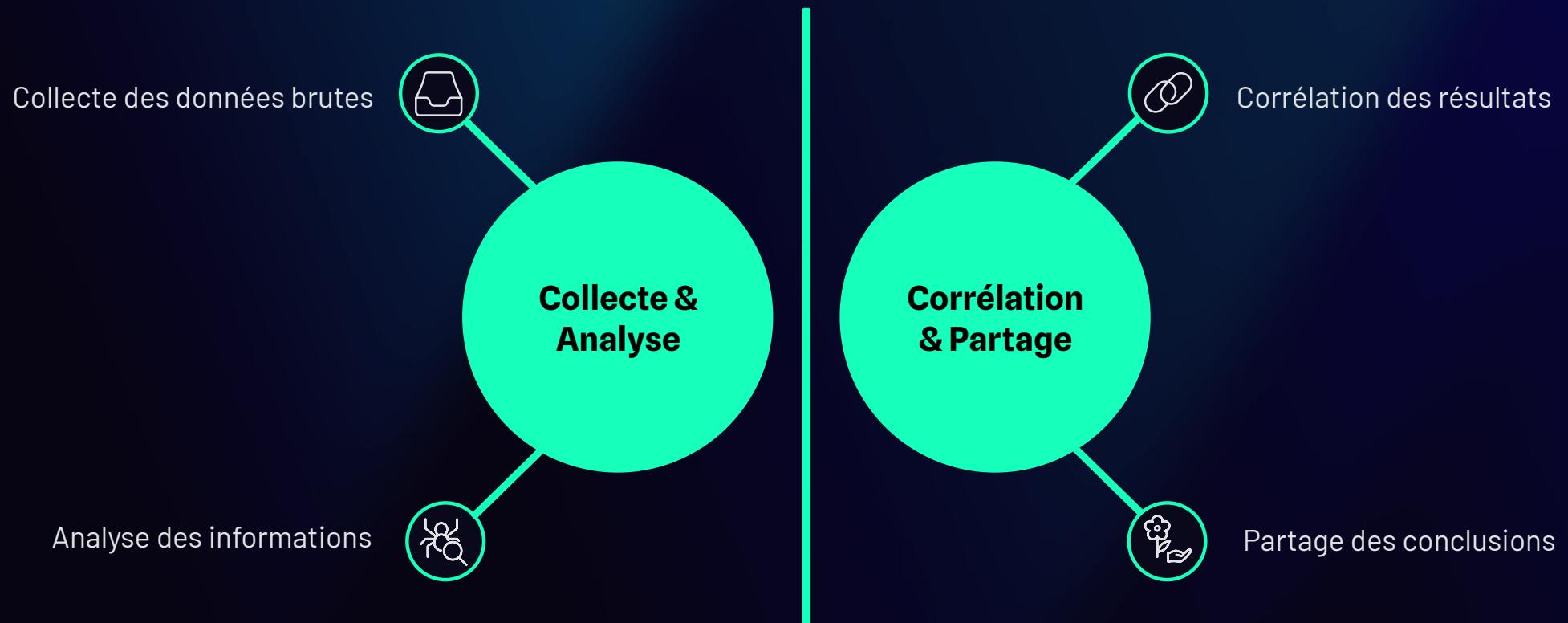
Les playbooks définissent des procédures claires pour chaque type d'incident : intrusion réseau (brute-force SSH), infection malware sur client Linux, et sabotage de fichiers système sur Windows. L'automatisation via Ansible permet des audits réguliers et une réponse rapide.

## Cycle de Sécurité



# Mission 09 : TheHive et MISP - Gestion d'Incidents et Threat Intelligence

Cette mission établit une infrastructure robuste pour la gestion des incidents de sécurité et l'intégration de renseignements sur les menaces, afin de renforcer les capacités de détection et de réponse de l'équipe SOC.



## TheHive

Plateforme collaborative de gestion des incidents de sécurité. Elle permet la création de cas, l'ajout d'observables (IP, domaines, hachages de fichiers), la gestion des tâches et une collaboration fluide entre les analystes.

- **Gestion des cas** : Organisation structurée des incidents.
- **Observables** : Extraction et analyse des éléments pertinents.
- **Réponse automatisée** : Intégration avec des outils tiers via Cortex.

## MISP

Plateforme open-source de Threat Intelligence qui facilite le partage d'Indicateurs de Compromission (IoCs) et de renseignements sur les menaces de manière structurée et automatisée.

- **Partage d'IoCs** : Échange d'informations avec la communauté.
- **Flux TI** : Abonnement à des flux de renseignements sur les menaces.
- **Corrélation** : Enrichissement des alertes avec des données de menaces connues.

L'intégration bidirectionnelle entre TheHive et MISP automatise la corrélation des observables avec les bases de données d'IoCs, permettant d'enrichir chaque cas d'incident avec des informations de Threat Intelligence pertinentes en temps réel.