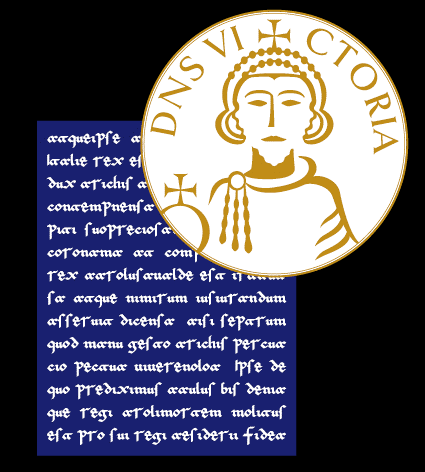
**UNIVERSITÀ DEGLI STUDI DEL SANNIO**

Dipartimento di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

**Progetto di Sicurezza delle Reti e dei Sistemi Software**

**PIATTAFORMA OPENCTI**

Vitagliano Pietro Mat.: 399000461

Petraccaro Angelo Mat.: 399000428

Rapuano Antonio Mat.: 399000429

A.A. 2020/2021

Indice

[OpenCTI 5](#_Toc91238179)

[Scopo 6](#_Toc91238180)

[Architettura 7](#_Toc91238181)

[Installazione e configurazione di OpenCTI 8](#_Toc91238182)

[API clients 15](#_Toc91238183)

[Client in Python 15](#_Toc91238184)

[Client in Go 15](#_Toc91238185)

[Connettori OpenCTI 16](#_Toc91238186)

[Data import 16](#_Toc91238187)

[AlienVault 16](#_Toc91238188)

[AM!TT 17](#_Toc91238189)

[CAPE Sandbox 17](#_Toc91238190)

[CrowdStrike 18](#_Toc91238191)

[Cryptolaemus 18](#_Toc91238192)

[Cuckoo Sandbox 19](#_Toc91238193)

[CVE Database 19](#_Toc91238194)

[Cyber Threat Coalition (COVID-19) 19](#_Toc91238195)

[Cybercrime Tracker 20](#_Toc91238196)

[Files restore 20](#_Toc91238197)

[FireEye 21](#_Toc91238198)

[Kaspersky 21](#_Toc91238199)

[LastInfoSec 22](#_Toc91238200)

[Malpedia 22](#_Toc91238201)

[MISP 23](#_Toc91238202)

[MITRE ATT&CK 23](#_Toc91238203)

[OpenCTI datasets 24](#_Toc91238204)

[RiskIQ 24](#_Toc91238205)

[Sekoia 25](#_Toc91238206)

[TAXII2 25](#_Toc91238207)

[TheHive 26](#_Toc91238208)

[ThreatMatch 27](#_Toc91238209)

[URLhause by Abuse.ch 27](#_Toc91238210)

[Valhalla 28](#_Toc91238211)

[Virustotal Livehunt 28](#_Toc91238212)

[VX Vault 28](#_Toc91238213)

[Yeti 28](#_Toc91238214)

[Stream consumer 29](#_Toc91238215)

[Backup Files 29](#_Toc91238216)

[Elastic 29](#_Toc91238217)

[History 30](#_Toc91238218)

[QRadar 30](#_Toc91238219)

[Splunk 30](#_Toc91238220)

[Tanium 31](#_Toc91238221)

[TenziràThreatBus 31](#_Toc91238222)

[Internal Enhrichment 32](#_Toc91238223)

[AbuseIPDB 32](#_Toc91238224)

[CAPE Sandbox 32](#_Toc91238225)

[Gatewatcher Lastinfosec 33](#_Toc91238226)

[GreyNoise 33](#_Toc91238227)

[Hatching Triage Sandbox 34](#_Toc91238228)

[Hybrid Analysis 34](#_Toc91238229)

[Hygiene 35](#_Toc91238230)

[Import External Reference 36](#_Toc91238231)

[Intezer Sandbox 36](#_Toc91238232)

[IPinfo 37](#_Toc91238233)

[Ivre 37](#_Toc91238234)

[Malbeacon 38](#_Toc91238235)

[Shodan 38](#_Toc91238236)

[Unpac-me 39](#_Toc91238237)

[VirusTotal 39](#_Toc91238238)

[File import 40](#_Toc91238239)

[ImportFileStix 40](#_Toc91238240)

[ImportReport (PDF) 41](#_Toc91238241)

[File export 42](#_Toc91238242)

[ExportFileCSV 42](#_Toc91238243)

[ExportFileSTIX 42](#_Toc91238244)

[ExportFileTXT 42](#_Toc91238245)

[Third party modules & plug-in 43](#_Toc91238246)

[Cortex 43](#_Toc91238247)

[Maltego 44](#_Toc91238248)

[Connettori custom 45](#_Toc91238249)

[Anyrun 45](#_Toc91238250)

[Trendmicro 45](#_Toc91238251)

[CONFIGURAZIONE DEI CONNETTORI 46](#_Toc91238252)

[Configurazione Manuale 46](#_Toc91238253)

[Configurazione Tramite Container Docker 47](#_Toc91238254)

[Configurazione Tramite Container Docker VS Manuale 48](#_Toc91238255)

[Workers 49](#_Toc91238256)

[Configurazione di MISP ed abilitazione dei feed 51](#_Toc91238257)

[Configurazione, sviluppo e testing del connettore Anyrun 56](#_Toc91238258)

[Configurazione, sviluppo e testing del connettore Trend Micro 60](#_Toc91238259)

[Export dei dati 66](#_Toc91238260)

# OpenCTI

OpenCTI è una piattaforma open source che consente a qualsiasi utente o organizzazione di gestire le proprie informazioni sulle minacce informatiche. È stato creato per strutturare, archiviare e visualizzare informazioni tecniche e non tecniche sulle minacce informatiche.

I dati sono strutturati utilizzando uno schema di conoscenza basato sugli standard STIX2.

Le varie entità sono organizzate gerarchicamente e possono essere abstract entity, cioè entità base caratterizzate da un insieme di attributi e sub-entity, ossia entità che estendono quelle di tipo abstract, ereditandone gli attributi.

La piattaforma è stata progettata come una moderna applicazione web per cui, alla sua apertura la prima schermata che compare è la dashboard, la quale raggruppa tutti I dati principali.

OpenCTI include l'API GraphQL e può essere integrata con altri tools e/o applicazioni come MISP, TheHive, etc.



Dashbord di OpenCTI

# Scopo

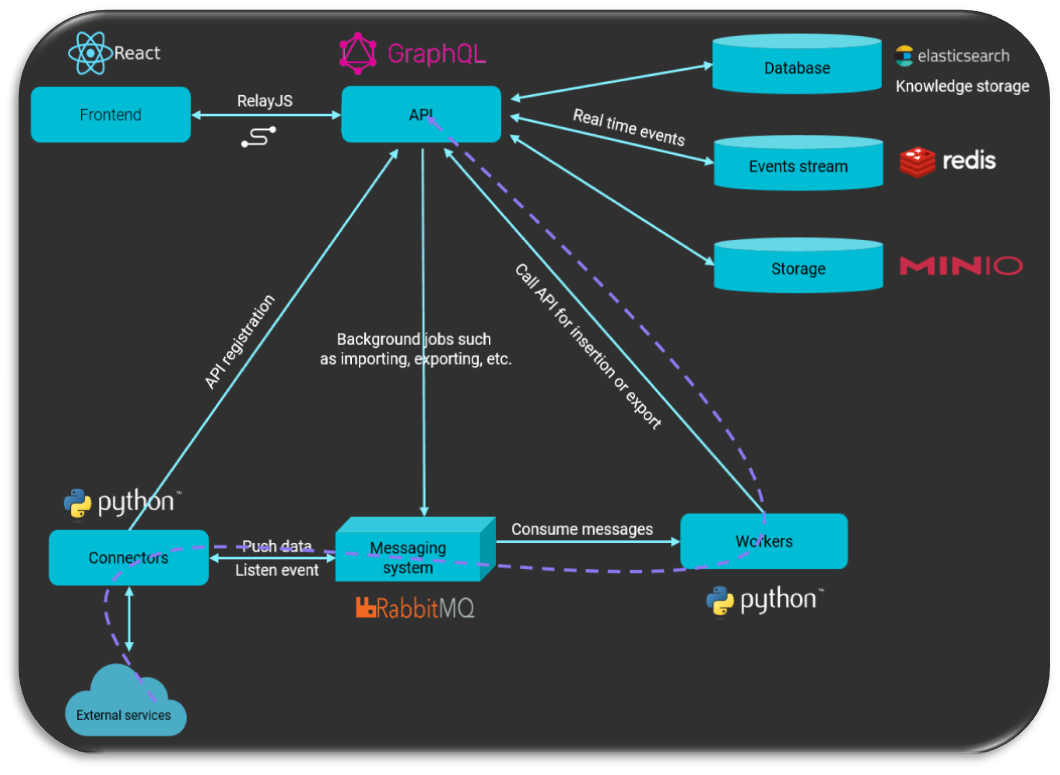
Questa piattaforma nasce con lo scopo di consentire agli utenti di archiviare e visualizzare informazioni tecniche (come TTP e osservabili) e informazioni non tecniche (come attribuzione suggerita, vittimologia, settore di attività e localizzazione), permettere di collegare ogni informazione alla sua fonte (un report, un evento MISP, etc) e fornire funzionalità come date di prima e ultima visualizzazione, collegamenti tra le informazioni, etc. OpenCTI sfrutta, tramite connettori dedicati, il framework MITRE ATT&CK, ossia una base di informazioni e conoscenze, accessibile gratuitamente da chiunque, di tattiche e tecniche ostili basate su osservazioni del mondo reale. MITRE ATT&CK viene utilizzato come base per lo sviluppo di modelli di cyber threats validi sia nel settore privato che in quello pubblico e governativo per migliorare i servizi di sicurezza informatica.

L'utente può anche scegliere di implementare i propri set di dati. Una volta che i dati sono stati integrati all'interno di OpenCTI, è possibile elaborarli per ricavare nuove relazioni e informazioni al fine di facilitare la comprensione e la rappresentazione dei dati stessi. OpenCTI consente non solo di importare i dati, ma anche di esportarli in diversi formati (CSV, STIX2 bundles, etc).

# Architettura

Per il suo funzionamento, la piattaforma OpenCTI si basa su vari componenti.

* **API GraphQL**: è la parte centrale della piattaforma OpenCTI, consentendo ai client di interagire con il database e il sistema di messaggistica, implementato con il broker RabbitMQ. È costruito in NodeJS e implementa il linguaggio di query GraphQL.
* **Database**: serve per archiviare le informazioni e le conoscenze acquisite. È implementato con ElasticSearch.
* **Workers**: sono processi Python autonomi che consumano messaggi dal broker per eseguire query di scrittura asincrona. È possible avviare tutti I workers di cui si necessita per incrementare le scritture parallele e quindi il throughput. Ciò, ovviamente, è possibile finché non si raggiungerà il punto di saturazione rappresentato dal throughput del database.
* **Connettori**: sono processi Python di terze parti che possono svolgere 5 ruoli differenti:
  1. **EXTERNAL\_IMPORT**: estraggono i dati da fonti remote, li convertono in STIX2 e li importano su OpenCTI.
  2. **INTERNAL\_IMPORT\_FILE**: estraggono i dati dai file caricati su OpenCTI tramite la GUI o l’API.
  3. **INTERNAL\_ENRICHMENT**: rimangono in attesa di eventi relativi a nuove entità o a richieste degli utenti, per poi estrarre i dati allo scopo di arricchirli
  4. **INTERNAL\_EXPORT\_FILE**: esportano i dati da OpenCTI, sulla base delle varie entità definite e delle loro relazioni.
  5. **STREAM CONSUMER**: consumano i dati della piattaforma.

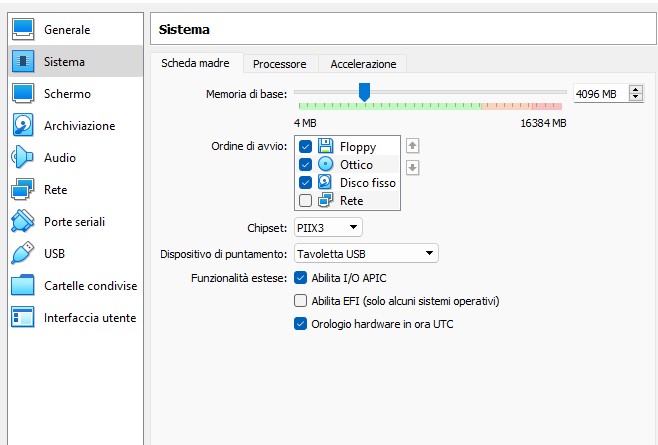


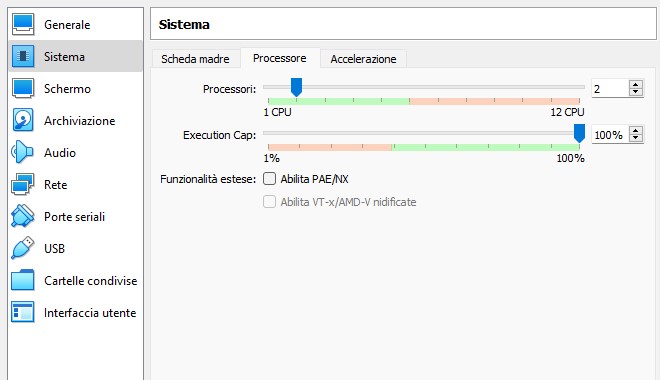
# Installazione e configurazione di OpenCTI

OpenCTI può essere installata manualmente, su macchine con Ubuntu come sistema operativo, oppure attraverso procedure automatizzate che richiedono l’uso di una Virtual Machine o di Docker. Per semplicità si è deciso di optare per un’installazione automatizzata basata su una macchina virtuale, con Virtual Box come HyperVisor di tipo 2.

Anzitutto occorre effettuare il download del file OVA «opencti-release-5.0.3.ova» reperibile al link: <https://drive.google.com/drive/folders/1bvB6RmdQNHMW_3h-88KbAit9GRZlL5Bj>. Questo file potrà, poi, essere importato in Vitual Box per generare automaticamente una macchina virtuale preconfigurata.

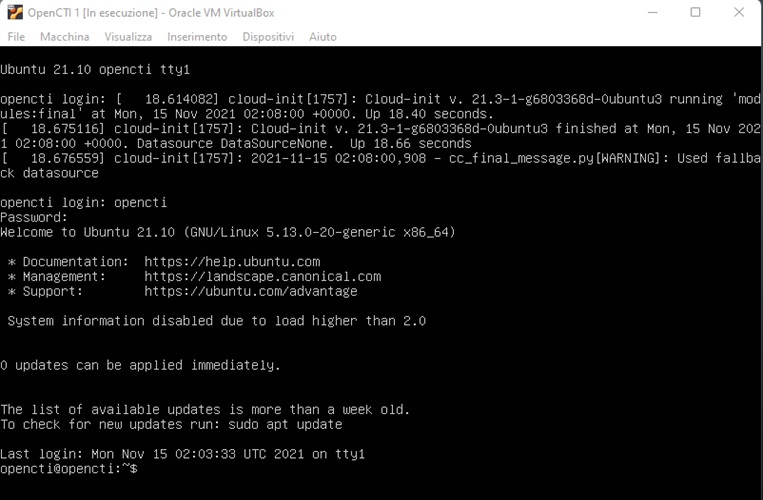
Inoltre, siccome la VM è preconfigurata con 1 core per la CPU e 16 GB di memoria RAM, al fine di garantire prestazioni migliori ed evitare di concedere più memoria di quella necessaria, le risorse assegnate alla macchina sono state modificate in 2 core e 4 GB di RAM.





Una volta avviata la VM è necessario effettuare l’accesso usando le credenziali di login:

* Username: opentcti
* Password: opencti

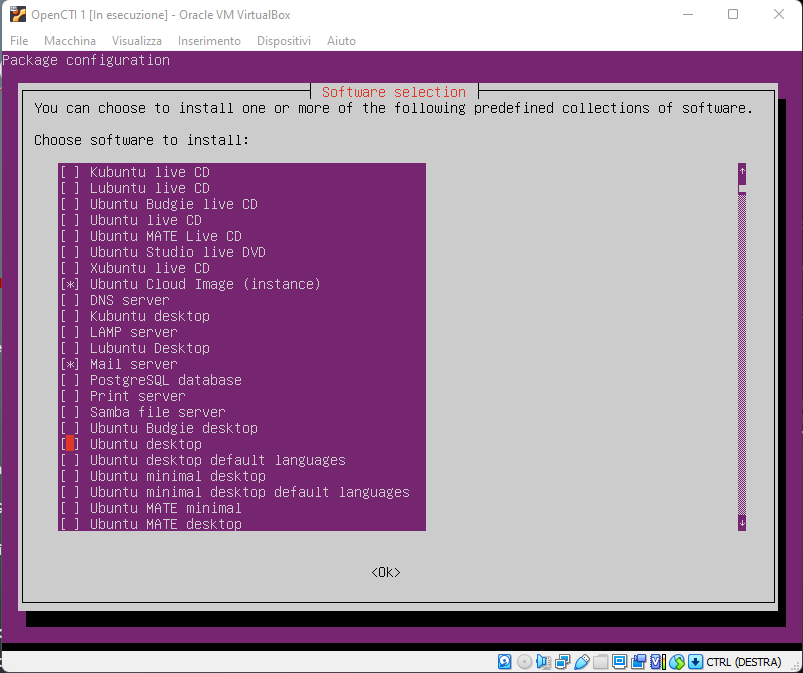


In questa fase, la VM è sprovvista sia di un browser che di un tool per reperire il proprio indirizzo IP, entrambi necessari per accedere alla dashboard di OpenCTI dalla VM stessa. Dunque, per permettere ciò, il successivo step della configurazione consisterà nella loro installazione.

Per installare i componeneti mancanti è sufficiente eseguire i seguenti comandi:

1. *sudo apt-get upgrade*
2. *sudo apt-get update*
3. *sudo apt install net-tools* (consente di utilizzare il comando ifconfig, necessario per ricavare il proprio indirizzo IP)
4. *sudo apt install tasksel* (necessario per installare una GUI con cui sarà possibile utilizzare un browser, a sua volta necessario per visualizzare la dashboard di OpenCTI)
5. *sudo apt install slim* (display manager per poter visualizzare la GUI)
6. *sudo tasksel*

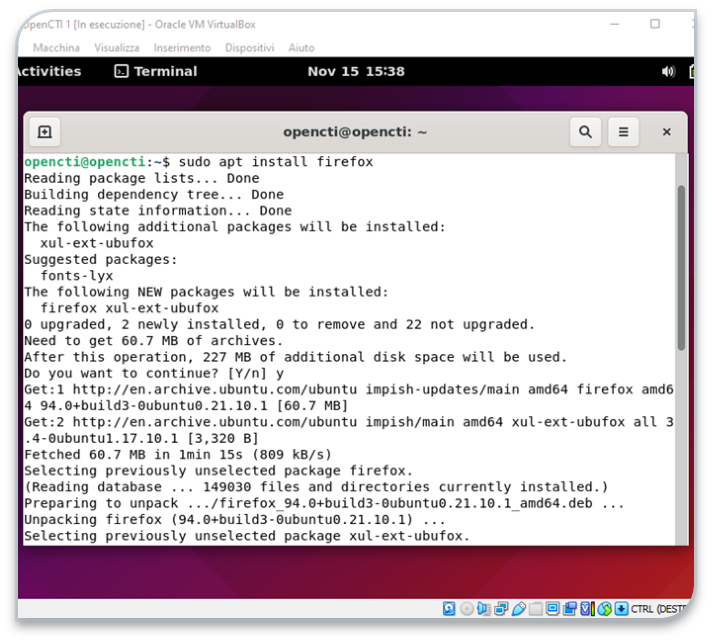
Eseguendo l’ultimo comando comparirà un’interfaccia tramite la quale sarà possibile scegliere quale GUI installare. Per semplicità è stata selezionata l’interfaccia più essenziale disponibile, ossia Ubuntu Desktop.



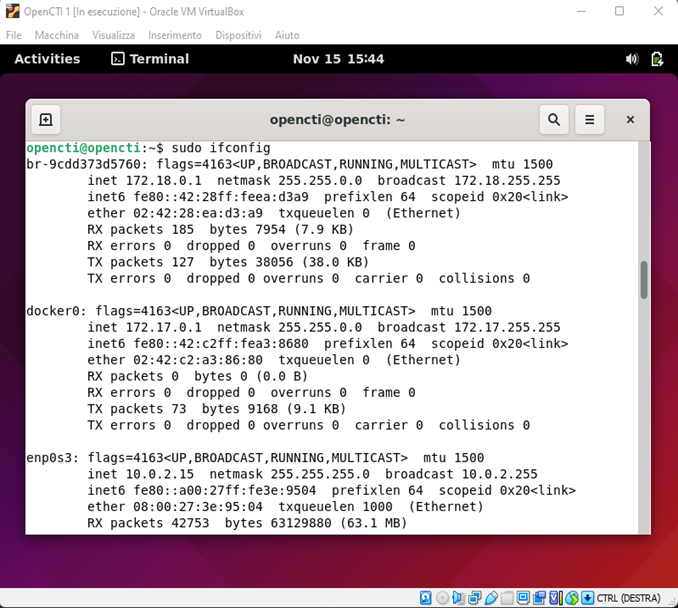
Al termine dell’installazione bisognerà riavviare la VM.

Da questo momento in poi, ogni volta che si avvierà la VM, verrà effettuato l’accesso ad Ubuntu automaticamente tramite la GUI e non più dalla shell di comando.

Una volta avviata la VM da GUI sarà necessario installare un browser per visualizzare la dashboard di OpenCTI. Per convenzione si è deciso di installare Firefox, per cui è sufficiente aprire la shell ed eseguire il comando *sudo apt install firefox*.



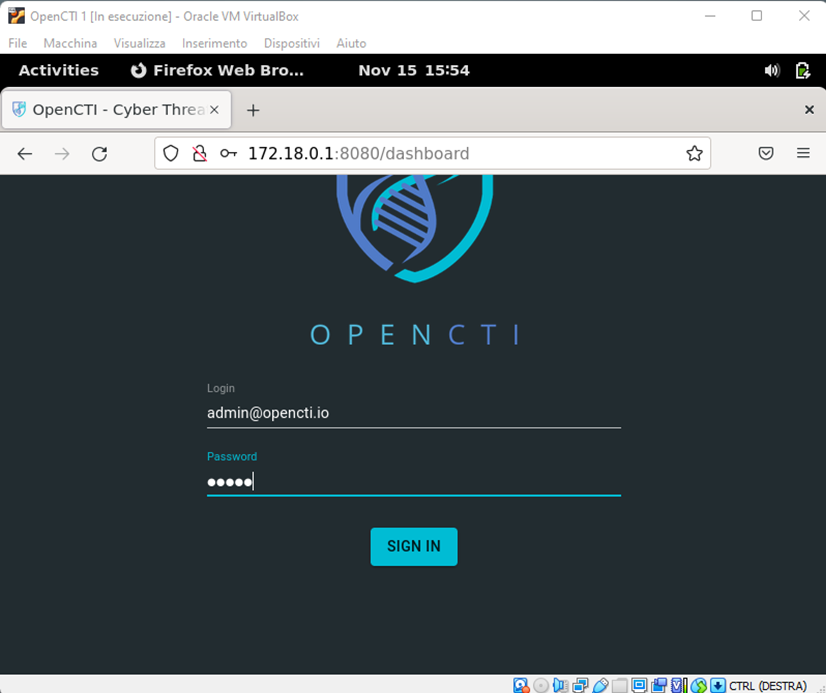
A questo punto bisogna recuperare il proprio indirizzo IP, il che può essere fatto tramite il comando *sudo ifconfig.* L’IP da utilizzare sarà il primo restituito dal comando, ossia 172.18.0.1.

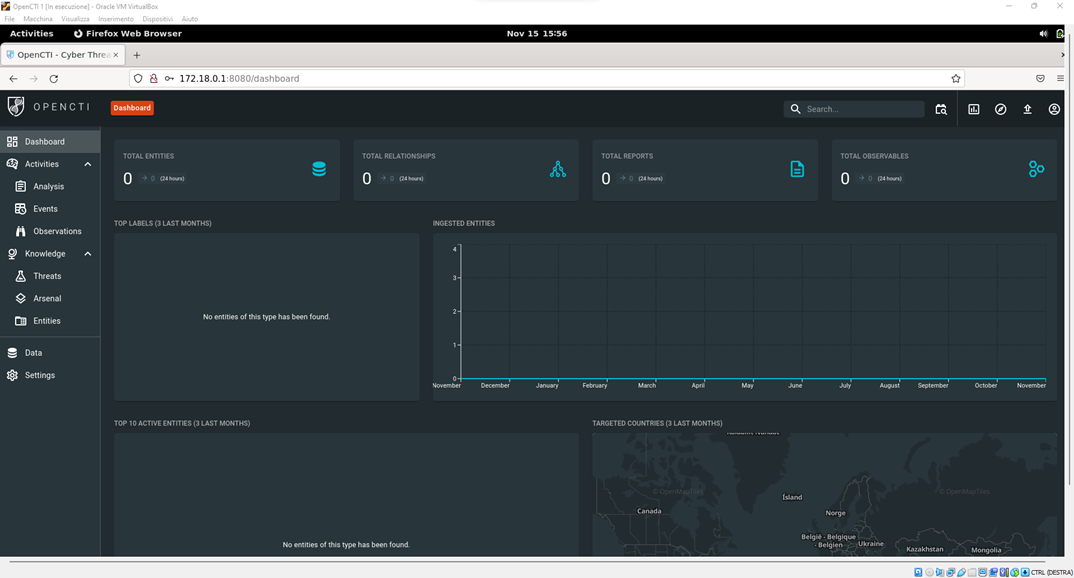


Dopodiché, occorre eseguire il comando *sudo -p 22* [*opencti@172.18.0.1*](mailto:opencti@172.18.0.1)per impostare questo IP come affidabile.

Infine, si deve aprire il browser e contattare il seguente URL: http://172.18.0.1:8080 (questa operazione può essere effettuata solo dopo il termine dell’avvio della piattaforma, cosa che richiede mediamente 5 minuti da quando si accede alla VM). Si verrà reindirizzati alla schermata di login di OpenCTI, dove si potranno utilizzare le seguenti credenziali per accedere alla dashboard:

* Username: admin@opencti.io
* Password: admin

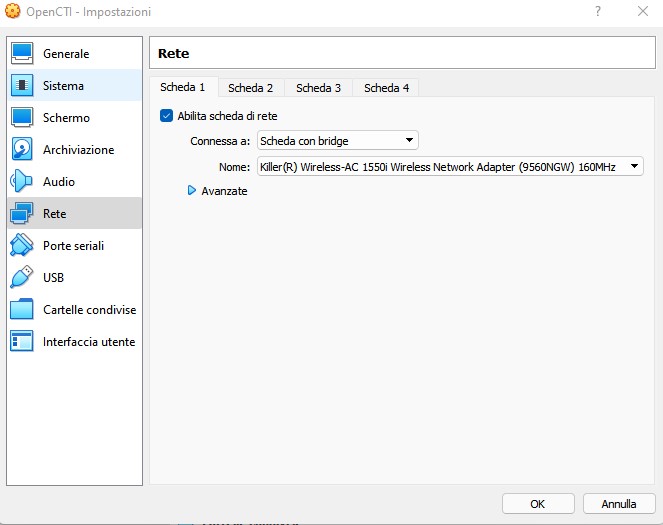




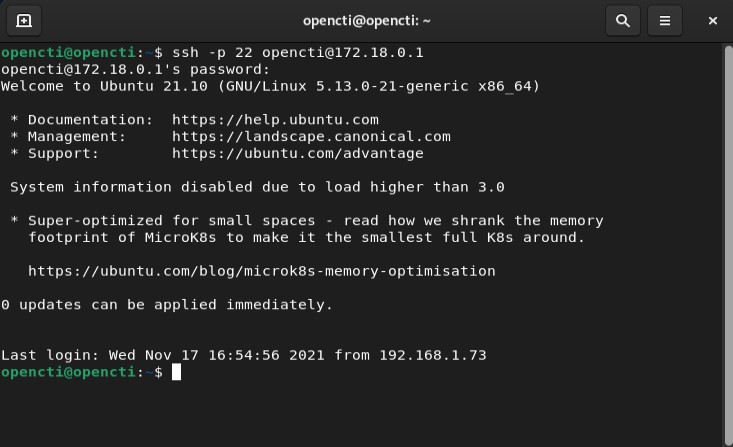
A questo punto la VM è completamente configurata ed è possibile utilizzarla anche per accedere, tramite essa, alla dashboard di OpenCTI.

In realtà, è possibile accedere alla dashboard anche tramite browser di altre machine, come l’host della VM o uno smartphone. Per un’ottimizzazione delle risorse, oltre ad aver seguito i passaggi illustrati precedentemente, sono state effettuate anche delle ulteriori operazioni, al fine di raggiungere questo risultato.

Anzitutto, in Virtual Box, la connessione della VM è stata opportunamente modificata da *NAT* a *Scheda con Bridge.*

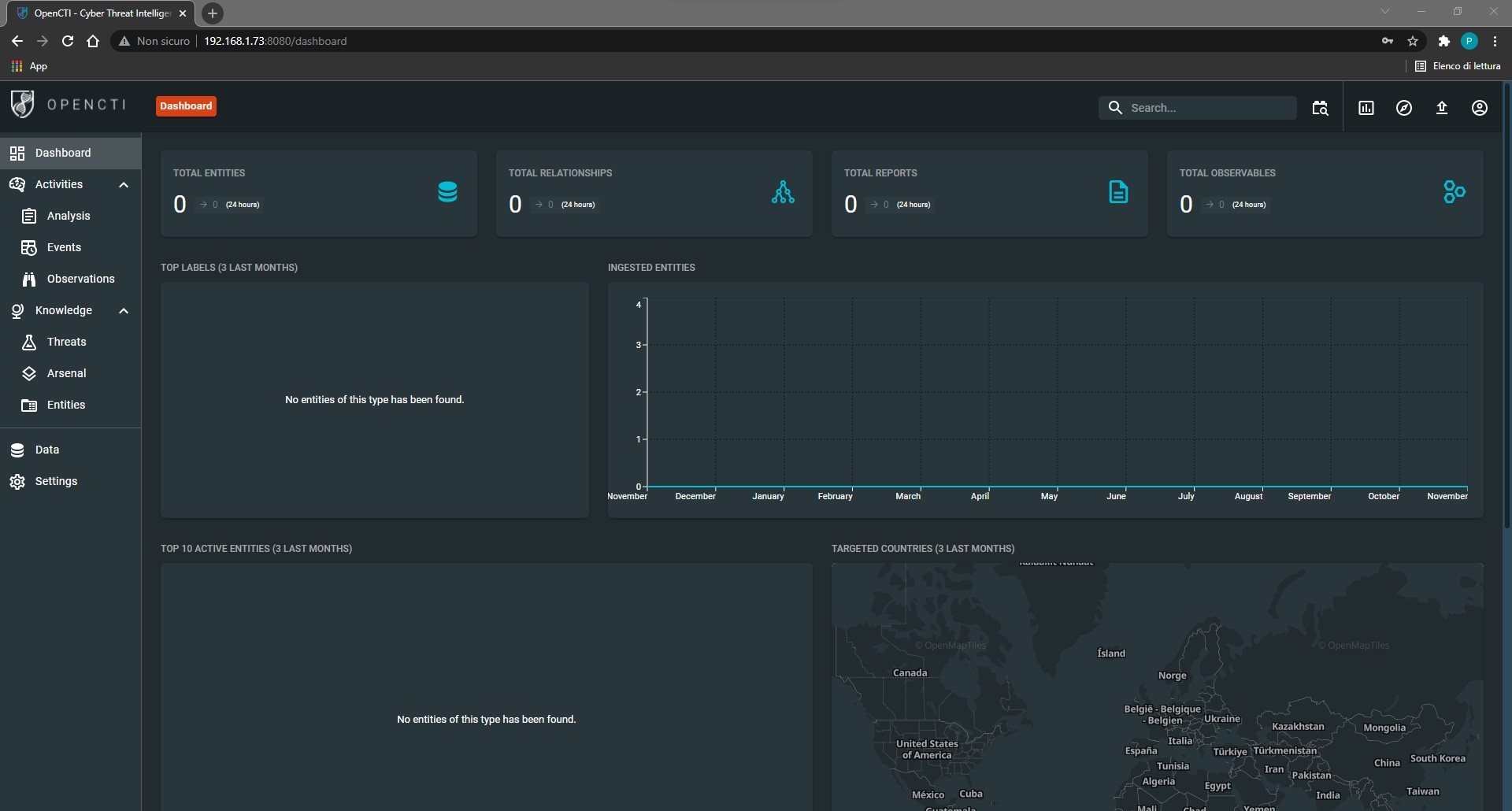
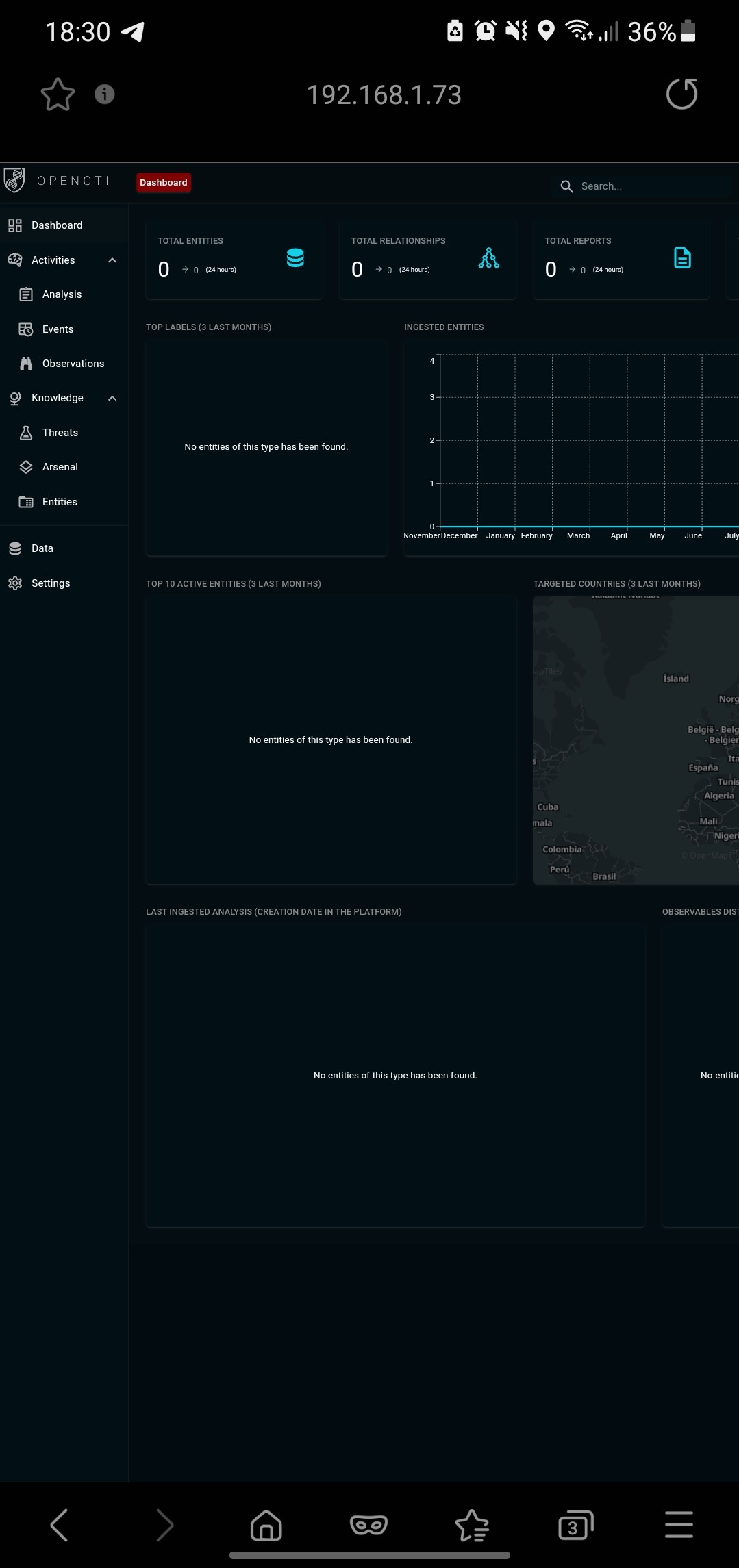


Poi, dalla shell di comando, è stato usato nuovamente il comando *sudo -p 22* [*opencti@172.18.0.1*](mailto:opencti@172.18.0.1)*,* questa volta con lo scopo di recuperare l’indirizzo IP da usare per accedere alla dashboard da un’altra macchina.



Arrivati a questo punto, sarà possibile accedere alla dashboard di OpenCTI anche tramite altre macchine, ricordando, ovviamente, di usare il nuovo IP per comporre l’URL da contattare via browser.

Dashborad di OpenCTI acceduta, rispettivamente, da uno smartphone e dall’Host della VM



# API clients

Per quanto riguarda i clients di OpenCTI, è possibile scriverli in 2 diversi linguaggi di programmazione:

* Python
* Go

## Client in Python

Con Python, l’interazione con OpenCTI avviene tramite la libreria pycti.

* Prerequisiti: Python >= 3
* Installazione: è richiesta l’installazione dell’ultima versione della libreria pycti tramite il comando *pip3 install pycti*

****

Esempio di inizializzazione di un client e creazione di un indicatore

È possibile trovare ulteriore documentazione, per la libreria pycti, al seguente link: [opencti-client-for-python.readthedocs.io](https://opencti-client-for-python.readthedocs.io/en/latest/pycti/pycti.html).

## Client in Go

Nella documentazione di OpenCTI non è riportata alcuna informazione relativa alla creazione e all’utilizzo di un client scritto in Go.

# Connettori OpenCTI

In questa sezione saranno descritti alcuni connettori (già sviluppati o ancora in fase di sviluppo) che è possibile collegare alla piattaforma e che consentono, a loro volta, di integrare altri tools o altre applicazioni.

Per la maggioranza dei connettori è disponibile sia il codice sorgente (che è essenzialmente uno script scritto in Python) sia un’immagine docker, che consente di effettuare il deploying in modo veloce.

Esistono due modi per integrare un connettore con OpenCTI:

* avviando il processo Python direttamente dopo aver fornito la configurazione corretta nel file config.yml
* attraverso un container docker

## Data import

### AlienVault

Può essere utilizzato per importare la conoscenza dalla piattaforma Alien Labs Open Threat Exchange (OTX).

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | È richiesta l'adesione alla community di intelligence sulle minacce OTX ([AlienVault - Open Threat Exchange](https://otx.alienvault.com/))  Librerie:   * pycti==5.0.3 * pydantic==1.8.2 * OTXv2==1.5.12 | [connectors/external-import/alienvault at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/alienvault) |

### AM!TT

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * python-dateutil==2.8.1 | [connectors/external-import/amitt at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/amitt) |

### CAPE Sandbox

Questo è un connettore per sincronizzare l'analisi sandbox CAPE come report e IOC OpenCTI.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | * OpenCTI Platform >= 5.1.0 * pycti==5.0.3 * certifi==2021.5.30 * chardet==4.0.0 * datefinder==0.7.1 * idna==2.10 * python-dateutil==2.8.1 * pytz==2021.1 * regex==2021.8.28 * simplejson==3.17.5 * six==1.16.0 * stix2-patterns==1.3.2 * urllib3==1.26.6 | [https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/cape](#_top) |

### CrowdStrike

Il connettore OpenCTI CrowdStrike può essere utilizzato per importare conoscenza dalla piattaforma CrowdStrike Falcon. Il connettore sfrutta le API Intel per ottenere informazioni sull'intelligence di CrowdStrike, inclusi dati su attori, indicatori, report e regole YARA.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | Richiede l'abbonamento alla piattaforma CrowdStrike Falcon. Inoltre, sono richieste le credenziali RabbitMQ (vengono fornite direttamente dell’API).  Librerie:   * pycti==5.0.3 * lxml==4.6.3   [https://github.com/certeu/crowdstrike-client#egg=crowdstrike-client](https://github.com/certeu/crowdstrike-client%23egg=crowdstrike-client%20) | [connectors/external-import/crowdstrike at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/crowdstrike) |

### Cryptolaemus

I membri di Cryptolaemus condividono informazioni in modo tale che tutti possano importare gli IOC nei loro prodotti di sicurezza informatica e proteggerli da possibili infezioni da Emotet, o aiutare con rilevamenti precoci prima che il malware possa causare danni ingenti.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * python-dateutil==2.8.1 * feedparser==6.0.2 * beautifulsoup4==4.9.3 * lxml==4.6.3 | [connectors/external-import/cryptolaemus at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/cryptolaemus) |

### Cuckoo Sandbox

Cuckoo Sandbox è il principale sistema di analisi malware automatizzato open source. Puoi lanciare qualsiasi file sospetto in un ambiente sandbox e in pochi minuti Cuckoo fornirà un rapporto dettagliato che delinea il comportamento del file. È un software gratuito che automatizza l'attività di analisi di qualsiasi file dannoso in Windows, macOS, Linux e Android. Il repository non è al momento raggiungibile.

### CVE Database

L'NVD è l'archivio del governo degli Stati Uniti di dati di gestione delle vulnerabilità. L'NVD include database di riferimenti a elenchi di controllo di sicurezza, difetti software relativi alla sicurezza, configurazioni errate, nomi di prodotti e metriche di impatto.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * certifi==2020.6.20 | [connectors/external-import/cve at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/cve) |

### Cyber Threat Coalition (COVID-19)

Il connettore OpenCTI Cyber Threat Coalition importa gli indicatori pubblicati dal team Cyber Threat Coalition. Cyber Threat Coalition è un team di esperti che raccolgono e condividono informazioni sulle minacce informatiche legate alla pandemia durante il periodo di crisi del COVID-19

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * feedparser==6.0.2 | [connectors/external-import/cyber-threat-coalition at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/cyber-threat-coalition) |

### Cybercrime Tracker

Il connettore utilizza il feed RSS del tracker in: http://cybercrime-tracker.net/rss.xml.

Genera un indicatore per ogni voce, indicando il relativo malware.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * feedparser==6.0.2 * pygrok==1.0.0 | [connectors/external-import/cybercrime-tracker at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/cybercrime-tracker) |

### Files restore

Questo connettore consente alle organizzazioni di ripristinare i propri dati OpenCTI da una cartella specifica.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti==5.0.3 * una directory accessibile dallo script Python con accesso in scrittura | [https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/restore-files](#_top) |

### FireEye

Questo connettore si collega all'API FireEye Intel V3 e raccoglie tutti i dati a partire da una determinata data.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| Non è detto che la versione free sia integrabile. | OPEN la versione free. | Librerie:   * pycti==5.0.3 | [connectors/external-import/fireeye at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/fireeye) |

### Kaspersky

Il connettore OpenCTI Kaspersky può essere utilizzato per importare la conoscenza da Kaspersky Threat Intelligence Portal. Il connettore sfrutta l'API del portale di Kaspersky Threat Intelligence per recuperare le informazioni pubblicate sul portale di Kaspersky Threat Intelligence, inclusi PDF di report, IoC e regole YARA.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | Licenza per utilizzare i servizi di Kaspersky Threat Intelligence Portal  Librerie:   * pycti==5.0.3 * pydantic==1.8.2 * lxml==4.6.3 | [connectors/external-import/kaspersky at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/kaspersky) |

### LastInfoSec

Il connettore OpenCTI LastInfoSec utilizza l'API /v2/stix21/getlasthour. Questo feed di minacce di LastInfosec contiene rapporti STIXv2.1 aggiornati ogni ora con URL, hash, indicatori di domini.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | Necessaria una chiave API (<https://info.gatewatcher.com/en/lp/opencti>).  Librerie:   * pycti==5.0.3 * urllib3==1.26.5 | [connectors/external-import/lastinfosec at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/lastinfosec) |

### Malpedia

Questo connettore importa la conoscenza dalla libreria di Malpedia. Il connettore aggiunge dati per i seguenti tipi di indicatore/osservabile OpenCTI:

* Yara
* file-sha256

Si prega di notare che Malpedia è gestito come un gruppo dove si accede solo su invito. Di conseguenza ogni registrazione sarà sottoposta a verifica.

L’accesso alle sole informazioni pubbliche può essere effettuato settando TLP:WHITE. Solo in questo caso non sarà richiesto un account.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API OPEN per informazioni pubbliche. | Necessaria una chiave API (<https://malpedia.caad.fkie.fraunhofer.de/settings>).  Librerie:   * pycti==5.0.3 * pydantic==1.8.2 | [connectors/external-import/malpedia at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/malpedia) |

### MISP

Questo connettore importa la conoscenza dalla libreria di Malpedia. Il connettore aggiunge dati per i seguenti tipi di indicatore/osservabile OpenCTI:

* Yara
* file-sha256

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Credenziali RabbitMQ (vengono fornite direttamente dell’API).  Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * pymisp * python-dateutil==2.8.1 * stix2==2.1.0 | [connectors/external-import/misp at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/misp) |

### MITRE ATT&CK

MITRE ATT&CK è una base di conoscenza accessibile a livello globale di tattiche e tecniche avversarie basate su osservazioni del mondo reale. La base di conoscenza ATT&CK viene utilizzata come base per lo sviluppo di modelli e metodologie di minaccia specifici nel settore privato, nel governo e nella comunità di prodotti e servizi di sicurezza informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * certifi==2020.12.5 | [connectors/external-import/mitre at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/mitre) |

### OpenCTI datasets

Il seguente repository viene utilizzato per archiviare i campioni di dati OpenCTI utilizzati dai connettori OpenCTI per sincronizzare i cataloghi di settori, regioni e paesi.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 | [connectors/external-import/opencti at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/opencti) |

### RiskIQ

Il connettore OpenCTI RiskIQ può essere utilizzato per importare la conoscenza dall'API RiskIQ.

Il connettore importa gli articoli di RiskIQ. Un articolo viene archiviato come un report STIX, contenente più indicatori.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN piano base. | Probabilmente è richiesta registrazione.  Librerie:   * pycti==5.0.3 * stix2==2.1.0 * urllib3==1.26.6 | [connectors/external-import/riskiq at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/riskiq) |

### Sekoia

SEKOIA.IO integra i flussi di intelligence degli aggressori e le capacità di automazione per identificare, comprendere e neutralizzare gli attacchi più rapidamente.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | Probabilmente è richiesta registrazione.  Librerie:   * pycti==5.0.3 * python-dateutil==2.8.1 | [connectors/external-import/sekoia at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/sekoia) |

### TAXII2

Importa i bundle STIX2.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Probabilmente è richiesta registrazione.  Librerie:   * pycti==5.0.3 * antlr4-python3-runtime * certifi * chardet * datefinder * idna * pika * python-dateutil * python-magic * pytz * PyYAML * regex * requests * simplejson * six * sseclient * stix2 * stix2-patterns * taxii2-client * urllib3 | [connectors/external-import/taxii2 at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/taxii2) |

### TheHive

È una piattaforma di risposta agli incidenti di sicurezza scalabile, open source e gratuita. Fornisce informazioni relative ai security incidents.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Probabilmente è richiesta registrazione.  Librerie:   * pycti==5.0.3 * thehive4py | [connectors/external-import/thehive at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/thehive) |

### ThreatMatch

Garantisce la consegna di rapporti periodici sulle minacce, rapporti e ricerche personalizzate sui clienti, inclusi collegamenti ad avvisi, incidenti, profili e scenari. I rapporti sulle minacce vengono forniti come parte dell'abbonamento a ThreatMatch.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | Vengono richieste le credenziali RabbitMQ (vengono fornite direttamente dell’API).  Probabilmente è richiesta registrazione.  Librerie:   * pycti==5.0.3 | [connectors/external-import/threatmatch at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/threatmatch) |

### URLhause by Abuse.ch

URLhaus è un progetto di abuse.ch con l'obiettivo di condividere URL dannosi che vengono utilizzati per la distribuzione di malware.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN | Librerie:   * pycti==5.0.3 * urllib3==1.26.5 * certifi==2020.6.20 | [connectors/external-import/urlhaus at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/urlhaus) |

### Valhalla

Questo connettore importa la conoscenza dall'API Valhalla. Il connettore aggiunge dati per i seguenti tipi di indicatore/osservabile Valhalla:

* modello indicatore stix2: yara

Il set di regole demo (noto anche come signature-base) contiene regole gratuite con licenza CC-BY-NC. L'accesso completo al database delle regole richiede un abbonamento attivo. Gli abbonamenti possono essere richiesti da <https://www.nextron-systems.com/valhalla/>

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN PIANO BASE | Abbonamento attivo per versione full.  Librerie:   * pycti==5.0.3 * pydantic==1.8.2 * valhallaAPI==0.3.0 | [connectors/external-import/valhalla at master · OpenCTI-Platform/connectors · GitHub](https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/valhalla) |

### Virustotal Livehunt

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | OPEN VERSIONE DEMO | Librerie:   * pycti==5.1.2 * vt-py==0.8.0 | <https://github.com/OpenCTI-Platform/connectors/tree/master/external-import/virustotal-livehunt-notifications> |

### VX Vault

Repository non raggiungibile.

### Yeti

Repository non disponibile.

## Stream consumer

### Backup Files

Questo connettore consente alle organizzazioni di prendere i propri dati OpenCTI e scriverli in una cartella specifica (di backup).

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * una directory accessibile dallo script Python con accesso in scrittura. * pycti==5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/backup-files> |

### Elastic

Questo connettore consente alle organizzazioni di alimentare la propria piattaforma elastica utilizzando la conoscenza di OpenCTI. Ha due modalità: ecs e stix. Questo connettore utilizza il flusso di eventi OpenCTI, quindi prende la conoscenza (le info) in tempo reale e aggiorna i documenti degli indicatori in formato ECS.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * Elastic platform >= 7.14.0 * elasticsearch==7.13.1 * Python3.9.x | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/elastic> |

### History

Questo connettore utilizza il flusso OpenCTI e scrive la cronologia delle entità. I connettori della cronologia memorizzano la configurazione attiva più recente di uno stato e dei relativi sottostati. Una volta creato, un oggetto viene associato a una configurazione per uno stato attivo, a partire dalla configurazione iniziale, e poi evolve man mano che il diagramma di stato risponde ai messaggi.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti==5.0.3 * elasticsearch==7.13.1 | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/history> |

### QRadar

Questo connettore è in grado di stabilire una correlazione tra tutte le varie informazioni e di raggruppare gli eventi correlati in singoli avvisi al fine di accelerare l'analisi e la correzione degli incidenti. Il repository GitHub non è disponibile.

### Splunk

È un connettore OpenCTI che importa eventi da Splunk Enterprise.

Questo connettore prende tutti gli eventi STIX2 memorizzati in un indice e li importa in OpenCTI.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | * OpenCTI Platform >= 5.0.0 * pycti==5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/splunk> |

### Tanium

Questo connettore consente alle organizzazioni di alimentare uno Splunk KV Store utilizzando la conoscenza di OpenCTI. Dopo aver avviato il connettore, si dovrebbe essere in grado di vedere una nuova fonte di Intelligence all'interno della piattaforma Tanium:

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | * OpenCTI Platform >= 5.0.0 * Tanium Threat Response >= 3.X.X * pycti == 5.0.3 * ioc\_writer == 0.3.3 * stix2-slider==3.0.0 | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/splunk> |

### TenziràThreatBus

Questo connettore consente la comunicazione tra OpenCTI e Threat Bus, il livello di diffusione dell'intelligence sulle minacce per gli strumenti di sicurezza open source. Utilizzando questo connettore, si può integrare profondamente OpenCTI threat intelligence con strumenti di rilevamento e database, come VAST, Zeek o CIF-3. Come funziona:

Il connettore funge da imbuto bidirezionale per indicatori e avvistamenti (Sightings) STIX-2. Utilizza il flusso di eventi OpenCTI (SSE) per elaborare gli aggiornamenti degli indicatori in tempo quasi reale e inoltra gli indicatori STIX-2 a Threat Bus tramite ZeroMQ. Allo stesso tempo, il connettore si iscrive al Threat Bus Topic per STIX-2 Sightings. Ogni volta che vengono pubblicati nuovi avvistamenti sul bus, il connettore li inoltra a OpenCTI tramite chiamate API.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | * un file di configurazione o determinate variabili di ambiente per l'avvio. * pycti == 5.0.3 * threatbus == 2021.6.24 * pyzmq==22.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/stream/threatbus> |

## Internal Enhrichment

### AbuseIPDB

È un database di indirizzi IP dannosi, coinvolti in attività dannose come spamming, tentativi di hacking, attacchi DDoS, ecc.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | 1.000 controlli IP al giorno | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/abuseipdb> |

### CAPE Sandbox

CAPE Sandbox è un software Open Source per automatizzare l'analisi dei file sospetti. Per fare ciò si avvale di componenti personalizzati che monitorano il comportamento dei processi dannosi durante l'esecuzione in un ambiente isolato.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO | * OpenCTI Platform >= 5.1.0 * pycti == 5.0.3 * pyzipper==0.3.5 | [https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/cape-sandbox](#_top) |

### Gatewatcher Lastinfosec

Il feed delle minacce di LastInfoSec è un feed di dati che semplifica il rilevamento delle minacce all'interno del sistema informativo. Contiene evidenze compromesse arricchite al fine di ridurre il tempo di analisi delle minacce una volta rilevate. LastInfoSec è stato creato per facilitare il rilevamento delle intrusioni nella tua azienda o organizzazione attraverso un feed di Threat Intelligence basato sull'intelligenza artificiale e la nostra esperienza nella risposta agli incidenti.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | Bisogna richiederla:  https://info.gatewatcher.com/en/lp/opencti | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 | [https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/lastinfosec](#_top) |

### GreyNoise

È un sistema che raccoglie, analizza ed etichetta la scansione Internet omnidirezionale e l'attività di attacco. Lo scopo di questo connettore è rispondere a questa domanda: "Tutti gli altri vedono queste cose o sono solo io che le vedo?" In altre parole: "Si tratta solo di un normale rumore di fondo di Internet o la macchina sta effettivamente mirando e attaccando ME in modo specifico?". Il connettore GreyNoise è un processo Python autonomo che deve avere accesso alla piattaforma OpenCTI e RabbitMQ. Le credenziali RabbitMQ e i parametri di connessione sono forniti direttamente dall'API, come configurato nelle impostazioni della piattaforma. L'abilitazione di questo connettore può essere eseguita avviando il processo Python direttamente dopo aver fornito la configurazione corretta nel file config.yml o all'interno di un Docker con l'immagine opencti/connector-greynoise:latest.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN, versione demo | * OpenCTI Platform >= 5.0.0 * pycountry==20.7.3 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/greynoise> |

### Hatching Triage Sandbox

Una sandbox di malware. Triage è la nuova e rivoluzionaria soluzione sandbox di malware di Hatching. Sfrutta un'architettura unica, sviluppata pensando alla scalabilità fin dall'inizio. Il triage può scalare fino a 500.000 analisi al giorno, un numero senza precedenti per un servizio di sandboxing.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO, con versione demo | * OpenCTI Platform >= 5.1.0 * hatching-triage==0.1.4 * git+git://github.com/OpenCTI-Platform/client-python | [https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/hatching-triage-sandbox](#_top) |

### Hybrid Analysis

Questo è un servizio di analisi malware, gratuito per la comunità, che rileva e analizza le minacce sconosciute utilizzando un'esclusiva tecnologia di analisi ibrida.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/hybrid-analysis-sandbox> |

### Hygiene

Questo è un connettore di arricchimento interno che utilizza i seguenti progetti esterni per cercare valori osservabili nel database che potresti voler eliminare perché è noto che portano ad alse-positivi quando vengono utilizzati per il rilevamento. [misp-warninglists](https://github.com/MISP/misp-warninglists)

Il connettore funziona per i seguenti tipi osservabili OpenCTI:

* + Indirizzo IPv4
  + Indirizzo IPv6
  + Nome del dominio
  + StixFile
  + Artefatto

L'abilitazione di questo connettore può essere eseguita avviando il processo Python direttamente dopo aver fornito la configurazione corretta nel file config.yml o all'interno di un Docker con l'immagine opencti/connector-hygiene:latest.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | Limitazione giornaliera sul numero di query. | * OpenCTI Platform >= 5.0.0 * <https://github.com/MISP/PyMISPWarningLists> | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/hygiene> |

### Import External Reference

Questo connettore consente alle organizzazioni di importare riferimenti esterni come file PDF o file MarkDown.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 * weasyptint == 53.2 * html2text == 2020.1.16 * pdfminer.six == 20201018 * pdfkit == 0.6.1 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/import-external-reference> |

### Intezer Sandbox

Il più grande database di minacce genetiche per tenere traccia delle varianti e degli attori delle minacce più recenti.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API | Requisiti | Link al repository GitHub |
| SI | API A PAGAMENTO, con versione free limitata | * OpenCTI Platform >= 5.1.0 * requests==2.26.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/intezer-sandbox> |

### IPinfo

Con questo connettore si possono individuare le posizioni degli utenti, personalizzare le loro esperienze, prevenire le frodi, garantire la conformità e molto altro ancora. Veloce, preciso e scelto da oltre 300.000 aziende e sviluppatori dal 2013

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | L'utilizzo gratuito della nostra API è limitato a 50.000 richieste API al mese. | * OpenCTI Platform >= 5.0.0 * pycountry==20.7.3 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/ipinfo> |

### Ivre

IVRE è un framework di ricognizione di rete open source, che semplifica la creazione di alternative completamente controllate e self-hosted a servizi come Shodan, ZoomEye, Censys (scansioni di rete estese), Greynoise (monitoraggio degli scanner) e/o PassiveDNS. Il connettore crea relazioni tra osservabili e aggiunge nuovi osservabili.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.1.0 * pycti == 5.0.3 * <https://github.com/ivre/ivre> * PyYAML | [https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/ivre](#_top) |

### Malbeacon

Questo è un connettore di arricchimento che utilizza l'API Malbeacon per aggiungere informazioni sulle reti di origine degli aggressori. MalBeacon impianta beacon tramite il traffico di check-in dei bot malware. Gli avversari che conducono campagne nefaste in natura amministrando questi pannelli malware C2 ora possono essere monitorati. MalBeacon è uno strumento per i buoni che fornisce informazioni aggiuntive sull'attribuzione degli attacchi.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pydantic==1.7.3 * pycti == 5.0.3 * urllib3==1.26.5 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/malbeacon> |

### Shodan

È un motore di ricerca che permette di trovare i dispositivi collegati alla rete Internet amministrabili da remoto. Uno strumento che aiuta a capire quant'è importante mettere in sicurezza la rete locale. Entro 5 minuti dall'utilizzo di Shodan Monitor vedrai cosa hai attualmente connesso a Internet all'interno della tua rete e sarai configurato con notifiche in tempo reale quando si presenta qualcosa di imprevisto.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| NO | OPEN | * OpenCTI Platform >= 5.0.0 * shodan==1.25.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/shodan> |

### Unpac-me

UNPACME è un servizio di spacchettamento automatico di malware. Questi vengono analizzati utilizzando una serie di processi di disimballaggio personalizzati gestiti da OpenAnalysis. Questi processi estraggono tutti i payload crittografati o compressi e restituiscono un set univoco di payload all'utente. In breve, UNPACME automatizza il primo passaggio nel processo di analisi del malware. È attualmente operativo come beta pubblica.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | Versione Beta OPEN | * OpenCTI Platform >= 5.1.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/unpac-me> |

### VirusTotal

Permette l'analisi gratuita di files e/o URLs per scovarne virus o malwares all'interno. Utilizza più di 70 software di [antivirus](https://it.wikipedia.org/wiki/Antivirus) tra cui Kaspersky, Avira, BitDefender, AVG, ESET, G-Data, Comodo, Malwarebytes e McAfee. Il 7 settembre 2012 è stato annunciato l'acquisto di VirusTotal da parte di [Google](https://it.wikipedia.org/wiki/Google_(azienda)). È possibile cercare, nel set di dati di VirusTotal, campioni di malware, URL, domini e indirizzi IP in base a proprietà binarie, verdetti di rilevamento antivirus, funzionalità statiche, modelli di comportamento come la comunicazione con host o indirizzi IP specifici, metadati di invio e molte altre nozioni. È possibile individuare i file simili a quello sospettato (oggetto di studio) ed i campioni corrispondenti ai criteri di ricerca possono essere scaricati per ulteriori studi.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | L'API pubblica è limitata a 500 richieste al giorno e una velocità di 4 richieste al minuto.  L'API pubblica non deve essere utilizzata in prodotti o servizi commerciali. | * OpenCTI Platform >= 5.0.0 * stix2==2.1.0 * plyara~=2.1.1 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-enrichment/virustotal> |

## File import

### ImportFileStix

Questo connettore è in grado di importare file JSON/XML contenenti dati STIX1 o STIX2.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * maec==4.1.0.17 * stix2-elevator==4.0.1 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-import-file/import-file-stix> |

### ImportReport (PDF)

Questo connettore consente alle organizzazioni di fornire informazioni dal report a OpenCTI.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 4.5.1 * L'estensione del file caricato nella piattaforma deve essere .json affinché il connettore possa gestirla. * urllib3==1.26.5 * pycti == 5.0.3 * beautifulsoup4==4.9.3 * pdfminer.six==20201018 * stix==1.2.0.11 * pydantic==1.8.2 * ioc-finder==6.0.1 * dateparser==1.0.0 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-export-file/export-file-stix> |

## File export

### ExportFileCSV

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-export-file/export-file-csv> |

### ExportFileSTIX

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * pycti == 5.0.3 | <https://github.com/OpenCTI-Platform/connectors/tree/master/internal-export-file/export-file-stix> |

### ExportFileTXT

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.1.0 * pycti == 5.0.3 | [https://github.com/OpenCTI-Platform/connectors/tree/master/internal-export-file/export-file-txt](#_top) |

## Third party modules & plug-in

### Cortex

Cortex cerca di risolvere un problema comune: come analizzare gli osservabili che sono stati raccolti, interrogando un singolo strumento anziché diversi?

Cortex, un software open source e gratuito, è stato creato da TheHive Project proprio per questo scopo. Gli osservabili, come indirizzi IP ed e-mail, URL, nomi di dominio, file o hash, possono essere analizzati uno per uno o in modalità bulk utilizzando un'interfaccia Web. Gli analisti possono anche automatizzare queste operazioni grazie all'API Cortex REST.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * OpenCTI Platform >= 5.0.0 * six>=1.14.0 * pycti * cortexutils | <https://github.com/TheHive-Project/Cortex-Analyzers/tree/master/analyzers/OpenCTI> |

### Maltego

Maltego è un tool sviluppato dalla società Paterva, che offre la possibilità OSINT (Open Source INTelligence) cioè di raccogliere informazioni tramite la consultazione di dati pubblicamente accessibili e raggrupparle, in formato grafico. Tramite questo strumento si è in grado di raccogliere informazioni da:

* Siti web
* Comunicazioni web (social network, wiki, blog, …)
* Dati pubblici (conferenze stampa, rapporti dei governi, dati demografici, …)
* Osservazioni dirette (dati geolocalizzati, conversazioni radio, foto satellitari, …)
* Professionisti ed accademici (simposi, conferenze, pubblicazioni scientifiche, associazioni professionali, …)

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN  per usi non commerciali | * OpenCTI Platform >= 5.0.0 * Python >= 3.6 * maltego-trx==1.3.8 * stix2==2.1.0 * six>=1.4.0 * markdown==3.3.4 * pycti>=5.0.1 * maltego-stix2>=2.1.1 | <https://github.com/amr-cossi/opencti-maltego> |

## Connettori custom

### Anyrun

È una malware sandbox online, per la ricerca dinamica e statica della maggior parte dei tipi di malware. Può analizzare diversi tipi di contenuto come file java, documenti office, file PDF, script, e-mail ed archivi compressi in formati zip, rar, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * pycti == 5.1.3 * web-socket.client=0.56.0 | Non presente |

### Trendmicro

TrendMicro è stato sviluppato come connettore di external import. Nello specifico TrendMicro importa report e osservabili.

|  |  |  |  |
| --- | --- | --- | --- |
| Integrabile | API pubblica | Requisiti | Link al repository GitHub |
| SI | OPEN | * pycti == 5.0.3 * feedrss == 6.0.8 | Non presente |

# CONFIGURAZIONE DEI CONNETTORI

I connettori possono essere configurati ed eseguiti in due modi distinti:

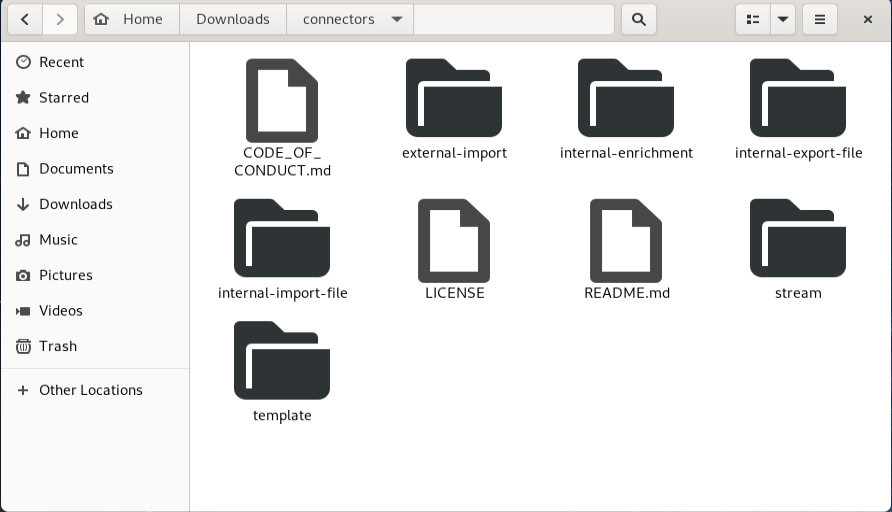
* Manualmente
* Tramite container docker

## Configurazione Manuale

Dopo l’avvio della VM, per la configurazione manuale occorre installare delle dipendenze coi seguenti comandi:

1. NodeJS: *apt install nodejs npm python3 python3-pip*
2. Redis-Server: *apt install redis-server*
3. RabbitMQ-Server: *apt install rabbitmq-server*
4. RabbitMQ Management Plugin: *rabbitmq-plugins enable rabbitmq\_management*

Successivamente bisogna scaricare il repository GitHub con tutti i connettori, ripartiti per categorie, tramite il comando *git clone* [*https://github.com/OpenCTI-Platform/connectors*](https://github.com/OpenCTI-Platform/connectors)*.*



Directory dei connettori dopo il comando git clone

A questo punto, per ogni connettore è necessario seguire i seguenti passaggi:

1. Apertura della directory con i file requirements.txt, config.yml.sample ed eseguibile python
2. Da terminale, esecuzione del comando *pip3 install -r requirements.txt* per scaricare le dipendenze dell’eseguibile python
3. Configurazione del file config.yml.sample, con l’eventuale inserimento dell’api key per abilitare il connettore
4. Creazione del file .yml con il comando *cp config.yml.sample config.yml*
5. Esecuzione del comando *python3 <nome dell’eseguibile>.py* per avviare il connettore e mandarlo in stato di running

## Configurazione Tramite Container Docker

Per la gestione dei connettori tramite container docker, invece, si può utilizzare Portainer.io, una piattaforma dalla quale è possibile configurare tutti i connettori all’interno di un unico file, denominato docker-compose.yml, accessibile dalla piattaforma via browser.

In questo caso, dopo l’avvio della VM, bisogna connettersi all’url https://{IP\_OPENCTI}:9443 e accedere al suddetto file .yml.

Immagine che contiene testo

Descrizione generata automaticamente

File docker-compose.yml di Portainer.io

## Configurazione Tramite Container Docker VS Manuale

Anche se, teoricamente, le due configurazioni dovrebbero essere equivalenti, quella manuale presenta alcuni svantaggi:

* Occorre scaricare numerose dipendenze e prerequisiti
* Per eseguire ogni connettore è necessario un terminale da cui lanciare il comando di avvio. Di conseguenza, occorrerebbe avviarli in modalità detatched, oppure scrivere uno script python che si occupi ogni volta di avviarli tutti
* Alcuni connettori, se eseguiti manualmente, non funzionano in modo propriamente corretto

Di conseguenza, la configurazione tramite docker risulta essere l’approccio migliore, nonché quello scelto.

Immagine che contiene testo

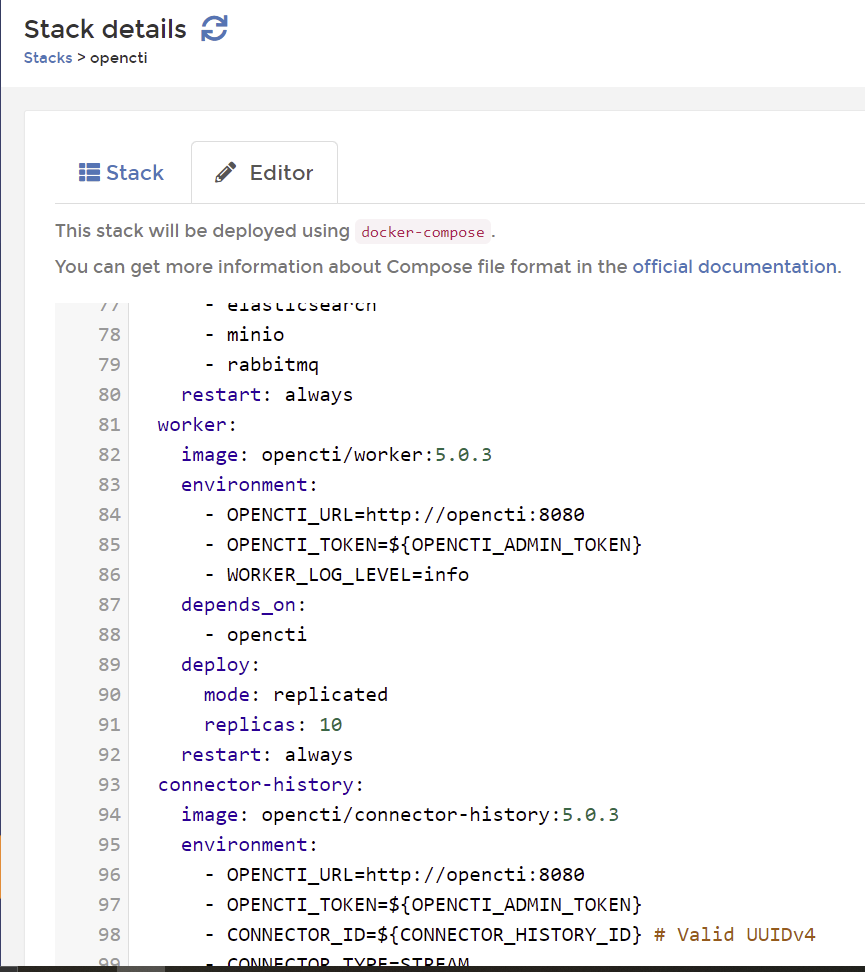
Descrizione generata automaticamente

Sezione “connettori” di OpenCTI Dopo la configurazione

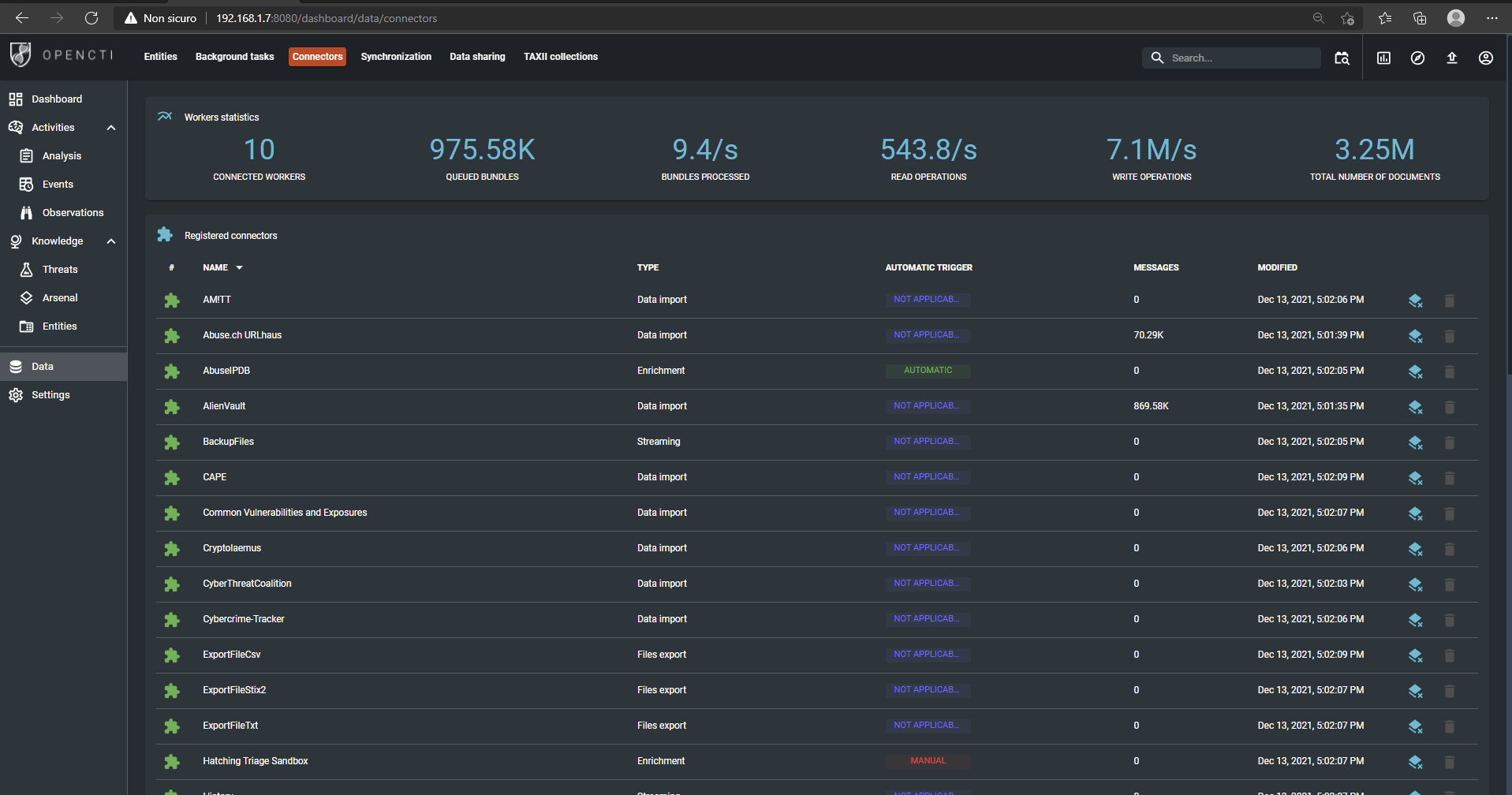
## Workers

I workers sono processi python autonomi che effettuano delle query di scrittura asincrona. Per aumentare il volume di dati importati dai connettori sarebbe opportuno aumentare il numero di workers. In questo modo si hanno più processi che possono occuparsi del download dei dati dai connettori e del relativo salvataggio nella base di dati di OpenCTI.

La modifica del numero di workers può essere effettuata direttamente dallo stack modificando il campo replicas.



Dopo la modifica e dopo aver eseguito l’update dello stack, il numero di workers sarà modificato anche nella dashboard.

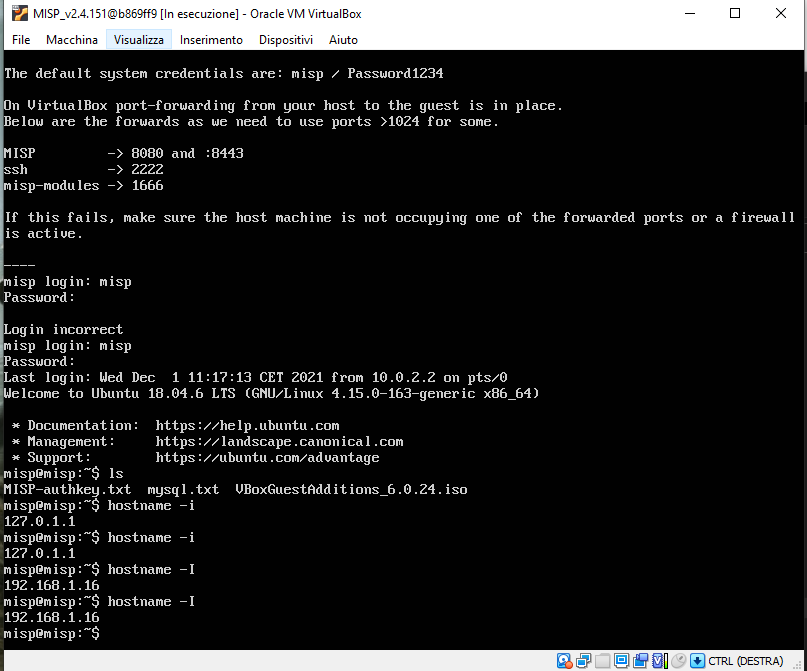


## Configurazione di MISP ed abilitazione dei feed

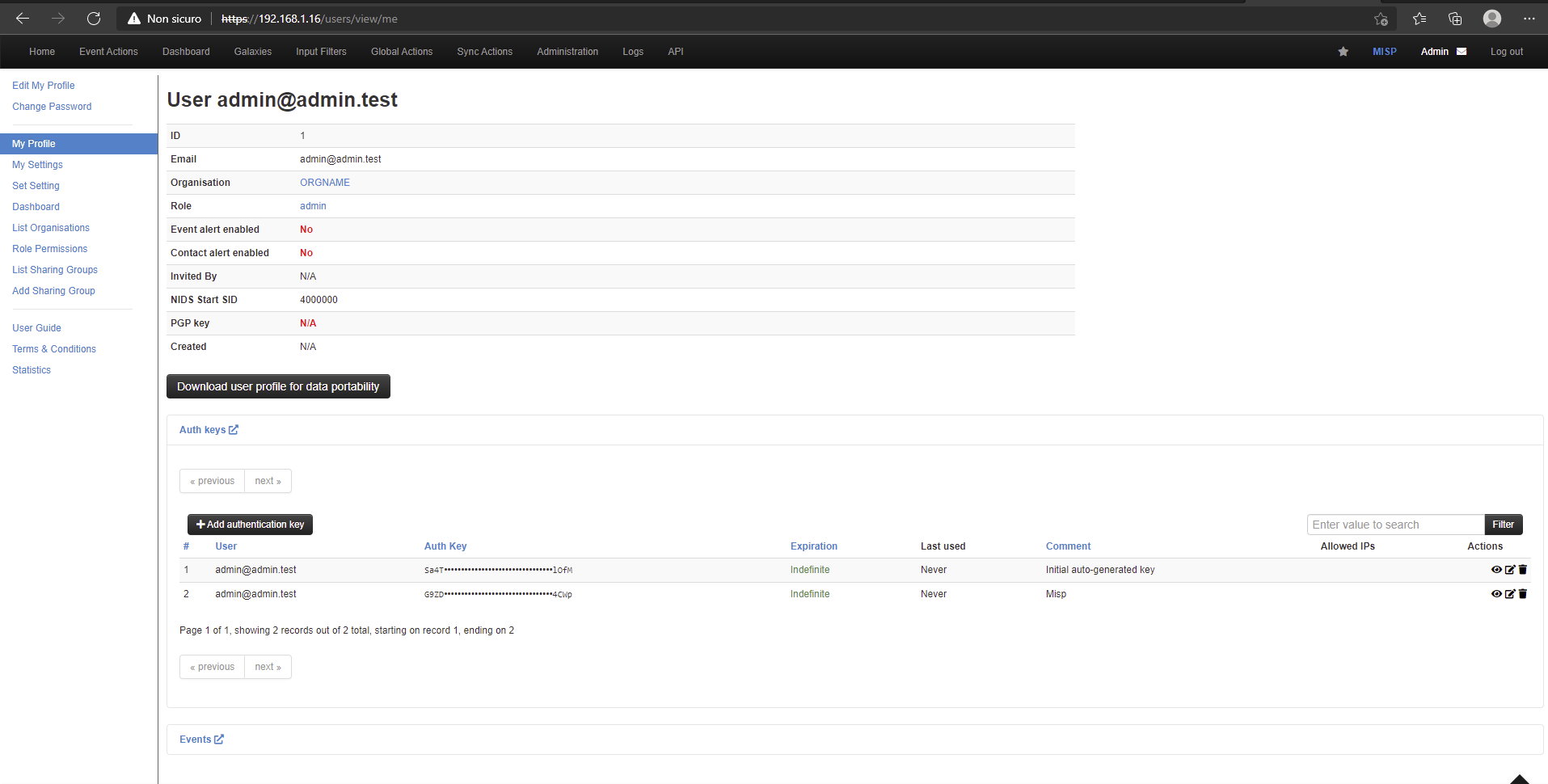
MISP è una piattaforma di intelligence che consente la condivisione, l'archiviazione e la correlazione di indicatori di compromissione, intelligence sulle minacce, informazioni sulle frodi finanziarie, informazioni sulle vulnerabilità o persino informazioni sull'antiterrorismo.

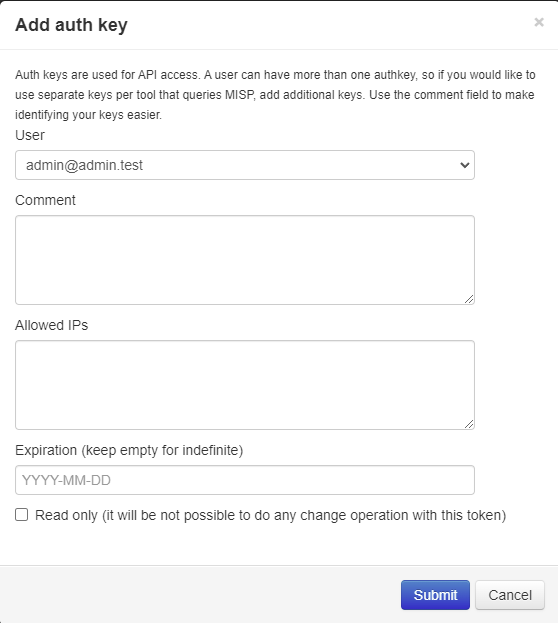
Per abilitare il connettore MISP bisogna:

* Scaricare e importare la macchina virtuale di MISP (<https://vm.misp-project.org/> ).
* Avviare la macchina virtuale e loggarsi (admin: misp e password: misp)



* Prelevare l’IP della macchina
* Connettersi al servizio sulla porta 80 (https://192.168.1.16)
* Autenticarsi: al primo avvio l’username sarà [admin@admin.test](mailto:admin@admin.test) e la password admin. Successivamente sarà possibile effettuare il cambio password.
* Generare un’api key: Selezionare la sezione Global Actions e poi cliccare su My Profile. In figura viene mostrata la sezione che contiene le informazioni relative all’utente e la lista delle api key create.

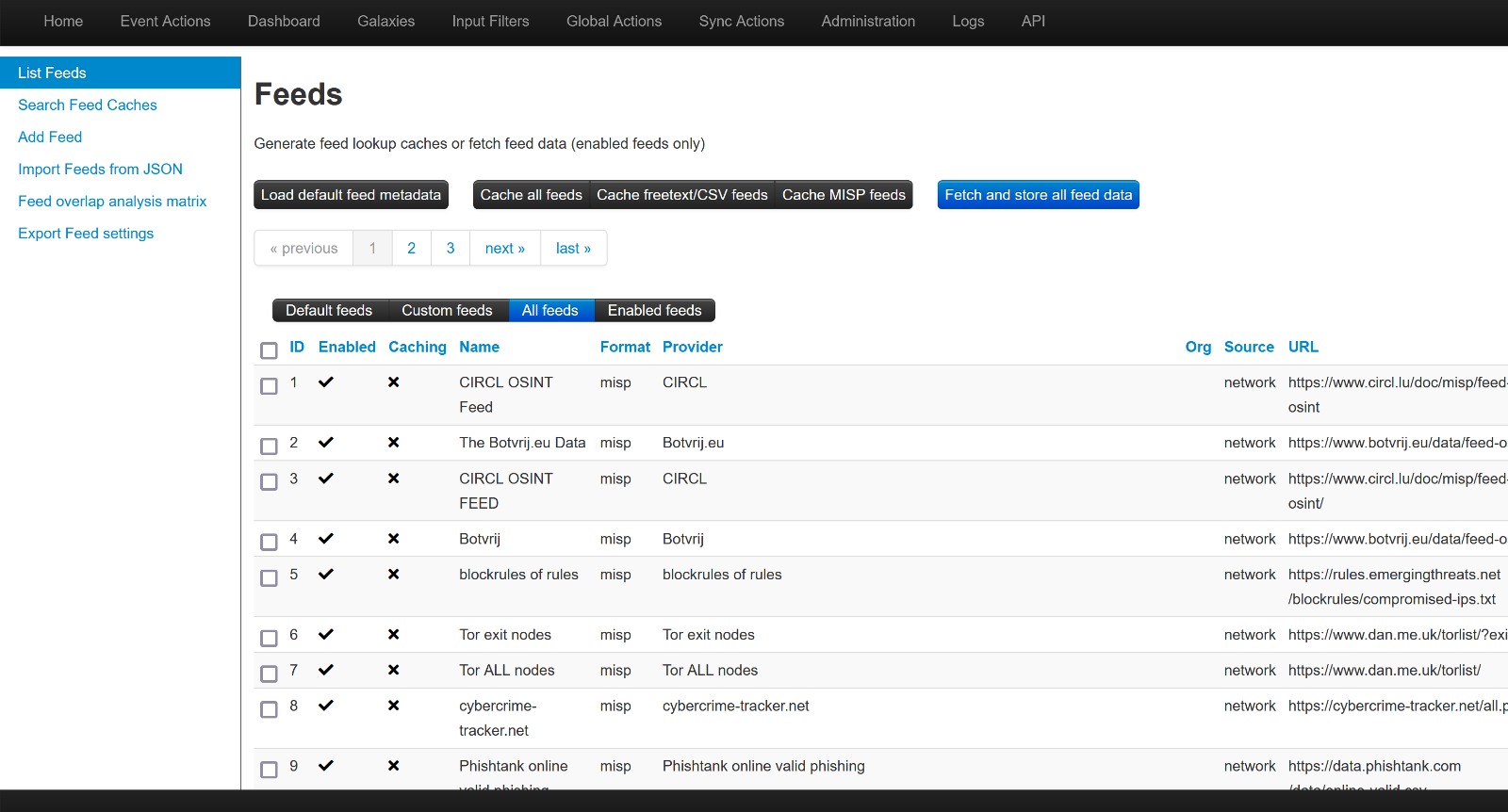




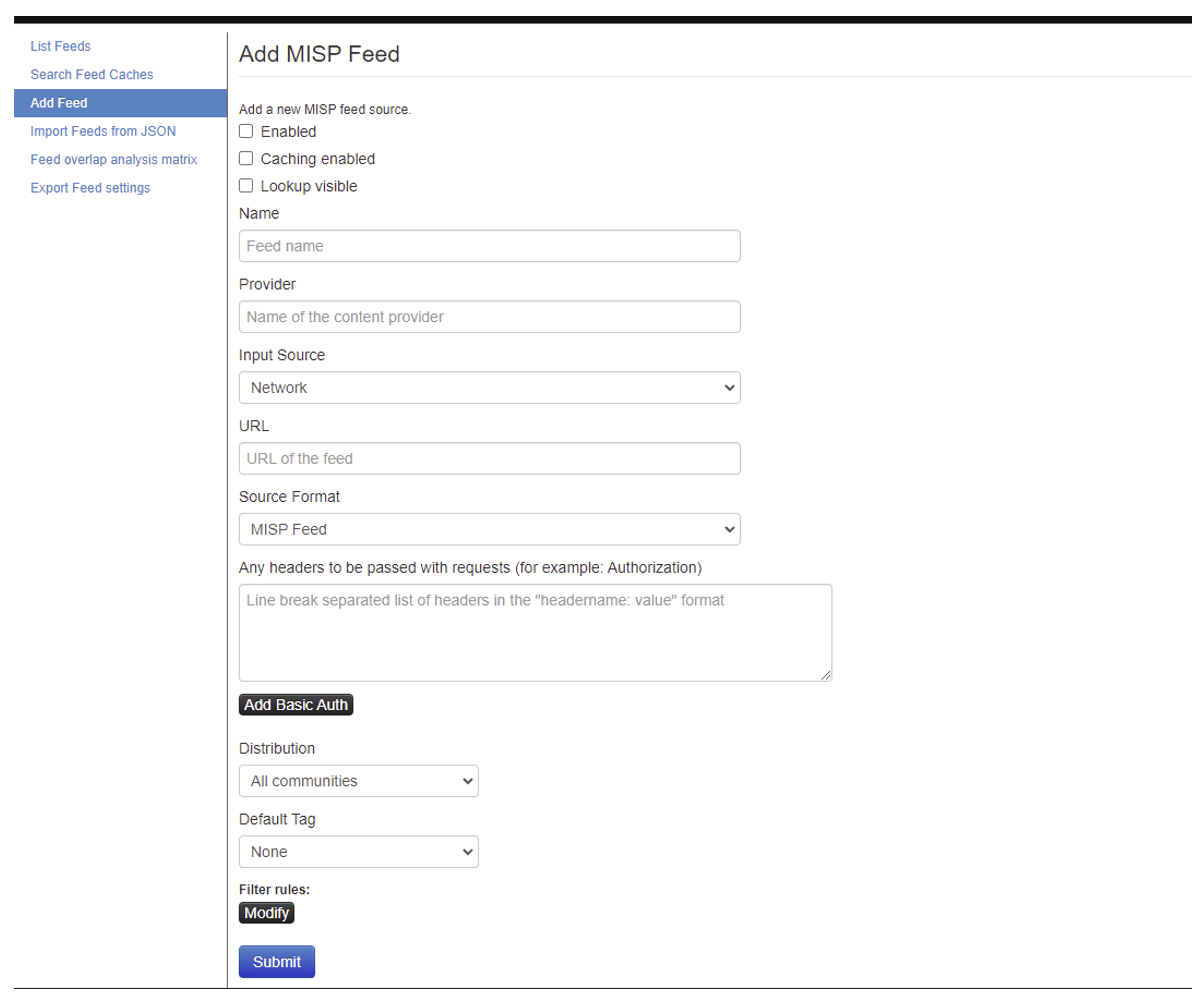
Definita la policy per l’api key è possibile confermarne la creazione cliccando su submit.

* Aggiungere feed: i feed sono documenti (txt, JSON, csv) che contengono informazioni relative a minacce, indicatori di compromissione, etc... Per aggiungere i feed bisogna loggarsi sulla piattaforma MISP, cliccare nella sezione Sync Action e poi su list feed.

L’interfaccia utente mostrerà tutti i feed caricati.



Bisogna poi cliccare su “Add Feed”e specificare almeno il nome, il provider, la sorgente, l’url e la sorgente dei dati.



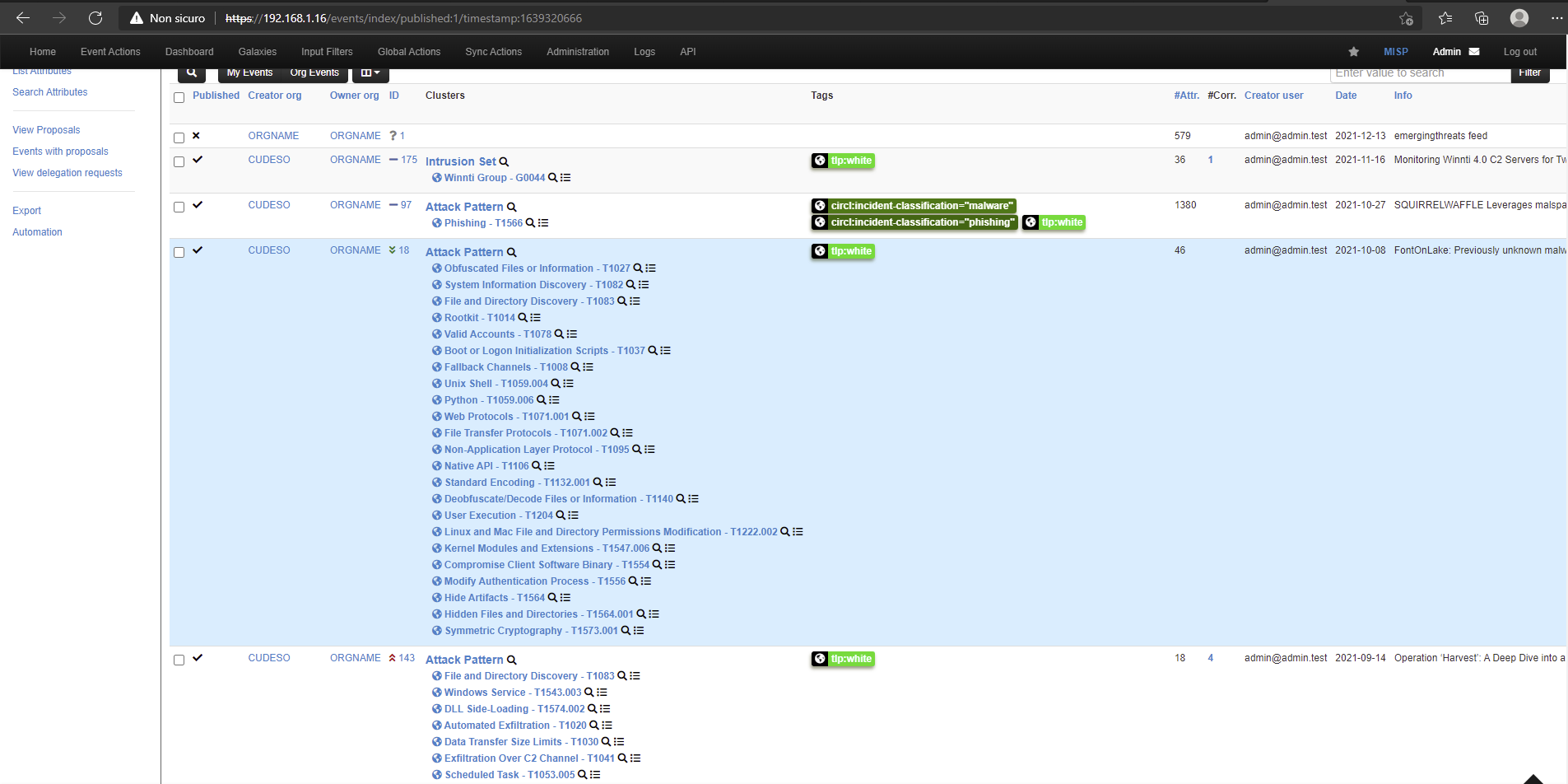
Successivamente sono stati sviluppati dagli scenari di test per testare il funzionamento della piattaforma MISP e per testare l’interazione tra MISP e OpenCTI.

Nel seguente scenario di test sono stati aggiungi dei feed alla piattaforma MISP, e poi, è stato verificato se effettivamente le informazioni venivano importate e mostrate nella dashboard.

Feed aggiunti:

* CIRCL OSINT Feed
* The Botvrij.eu Data
* Tor exit nodes
* Home.nuug.no
* emergingngthreats

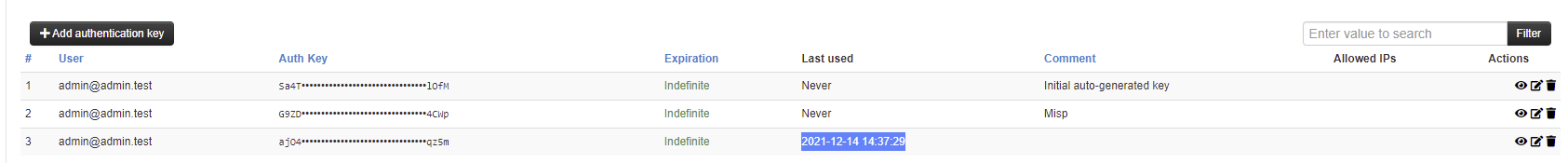
Dashboard:



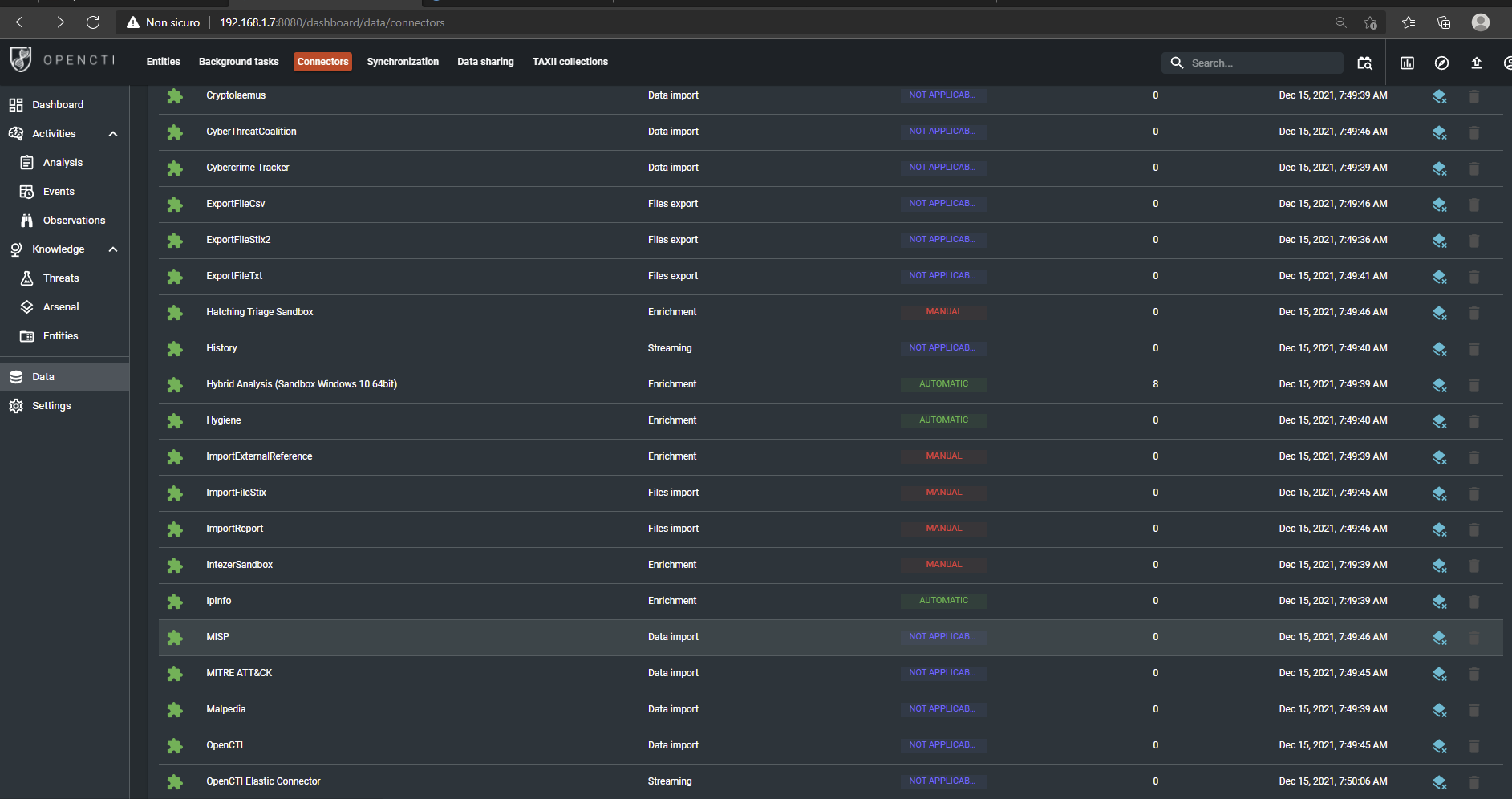
Come si può osservare dalla precedente figura, le informazioni vengono correttamente integrate e mostrate nella dashboard di MISP.

Il seguente scenario di test prevede la verifica dell’interazione tra OPENCTI e MISP. Nello specifico è stato verificato se nella dashboard di OPENCTI venivano correttamente importante le informazioni proveniente dalla piattaforma MISP.

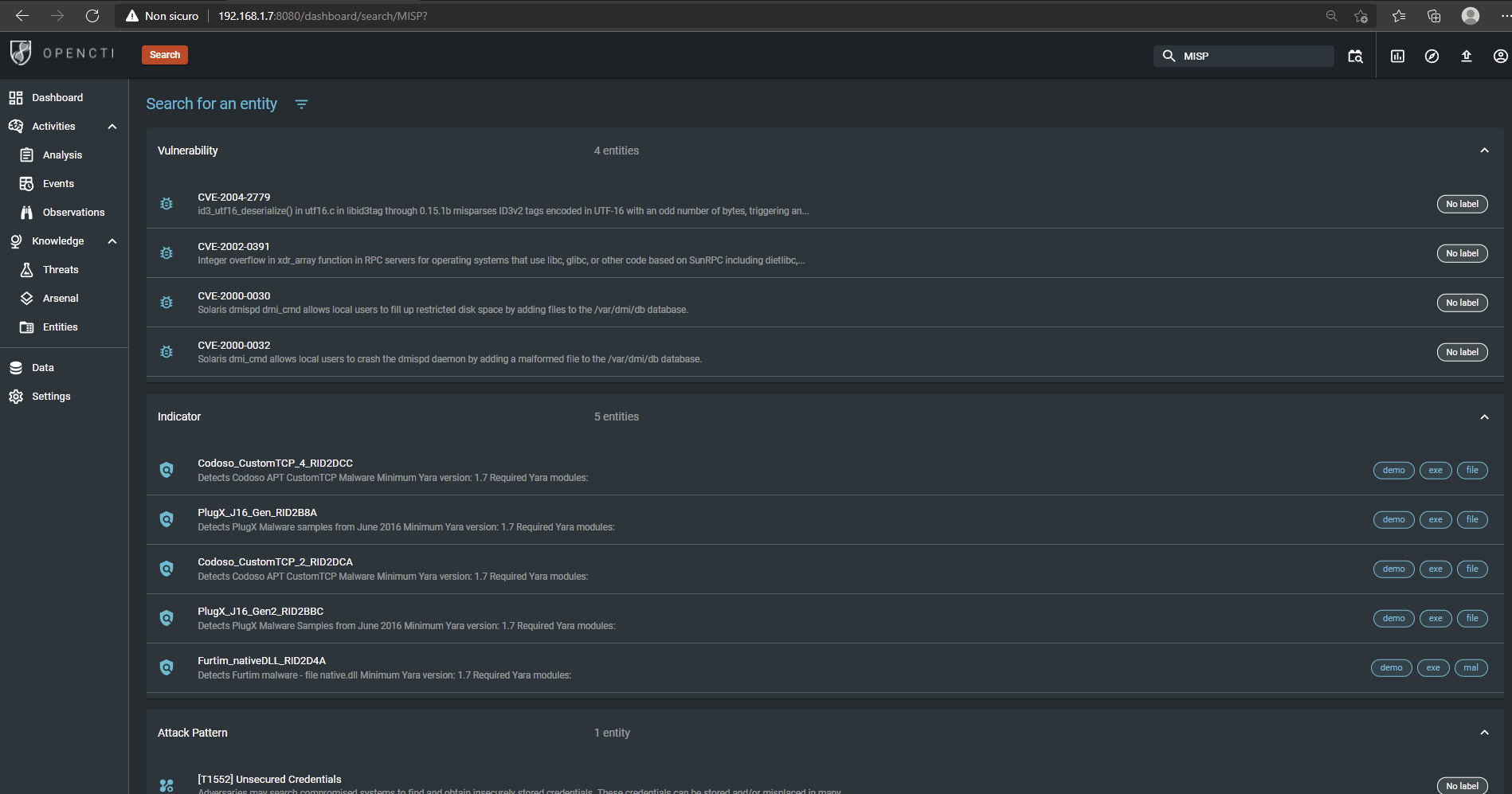
Come si può osservare in figura l’api key generata per consentire l’interazione tra MISP e OpenCTI è stata utilizzata.



Inoltre, il connettore è stato correttamente importato nella lista dei connettori di OpenCTI.



Le informazioni importate dal connettore sono le seguenti:



## Configurazione, sviluppo e testing del connettore Anyrun

Anyrun è stato sviluppato come connettore di Enrichment. Nello specifico, Anyrun va ad arricchire la descrizione delle famiglie di malware presenti nella piattaforma OpenCTI.

Siccome non sono presenti delle Api online che consento di ottenere le informazioni di anyrun è stato integrato il client di Pypl([anyrun · PyPI](https://pypi.org/project/anyrun/)).

Il client apre una websocket verso anyrun:



Nel metodo init\_connection vengono specificate le tipologie di informazioni che è possibile ottenere:

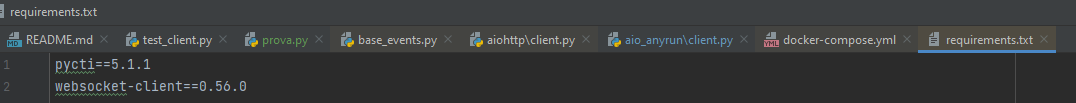


Compreso il funzionamento del client si è passati alla fase di configurazione del connettore e l’individuazione dei requisiti necessari all’installazione.

Il seguente file docker-compose.yml contiene la configurazione del connettore:



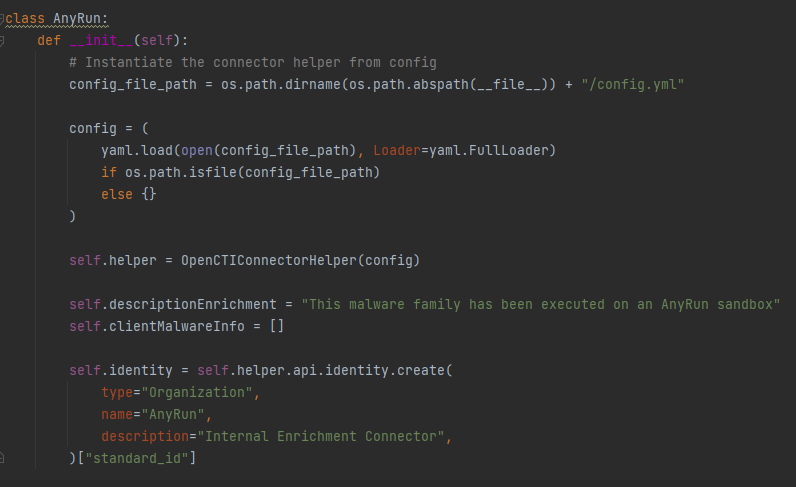
I requisiti di installazione sono contenuti nel file requirements.txt:



La seconda fase è stata lo sviluppo del connettore.

I passaggi salienti dello sviluppo sono:

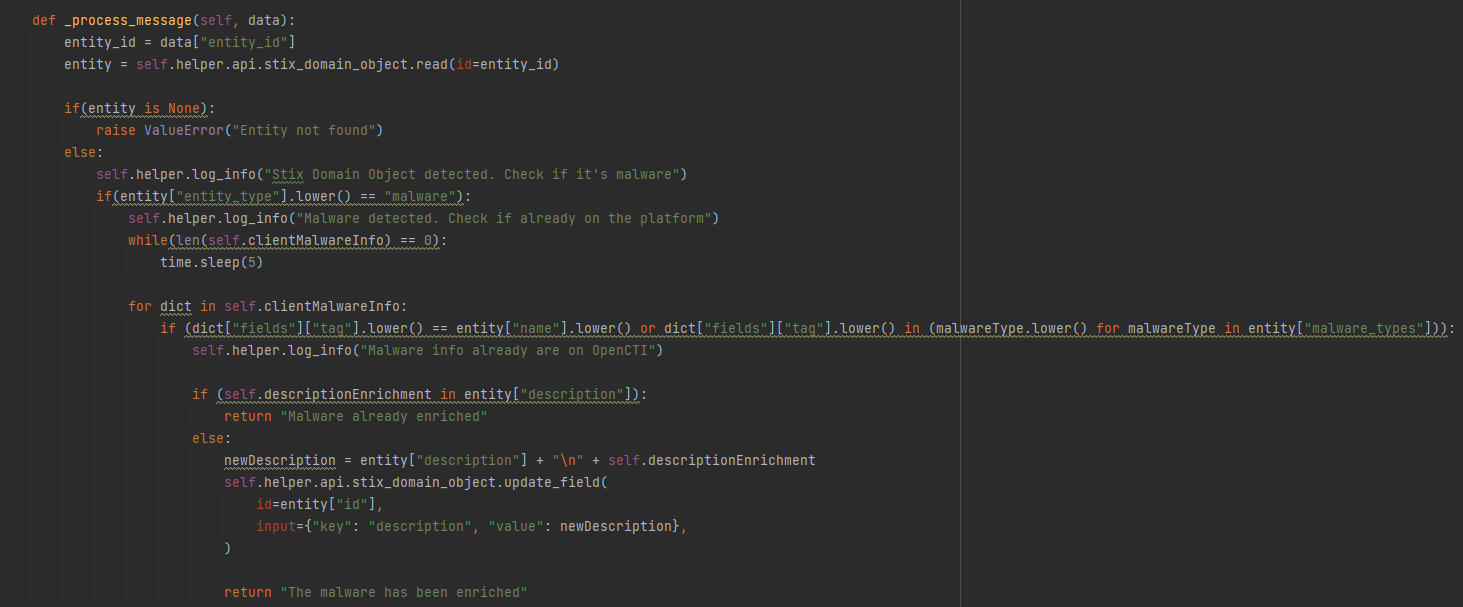
* Caricamento della configurazione del connettore (contenuta nel file yml):



* + Avvio del client:



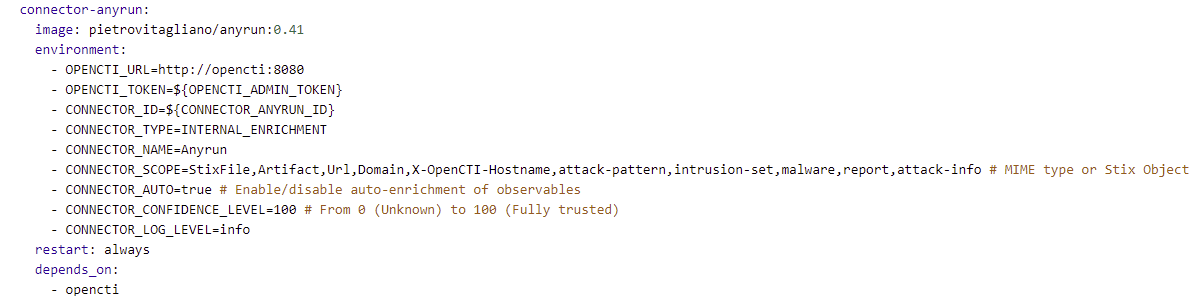
* + Ricerca della famiglia dei malware e arricchimento: l’arricchimento avviene invocando il metodo \_process\_message, ogni volta che nuovi dati arrivano su OpenCTI (programmazione a eventi)



* Creazione dell’immagine Docker:

docker image build -t pietrovitagliano/anyrun:0.41

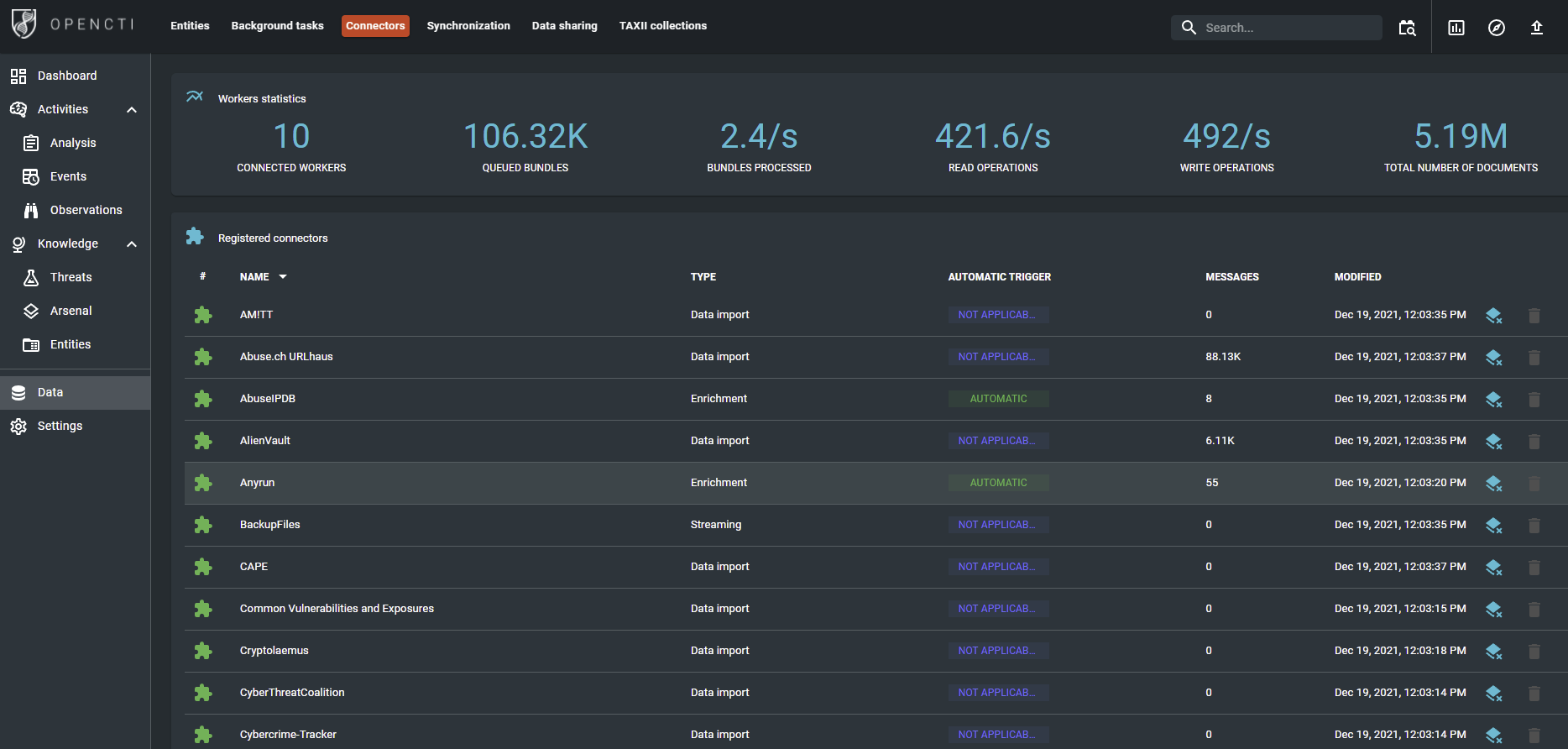
* Aggiornamento dello stack



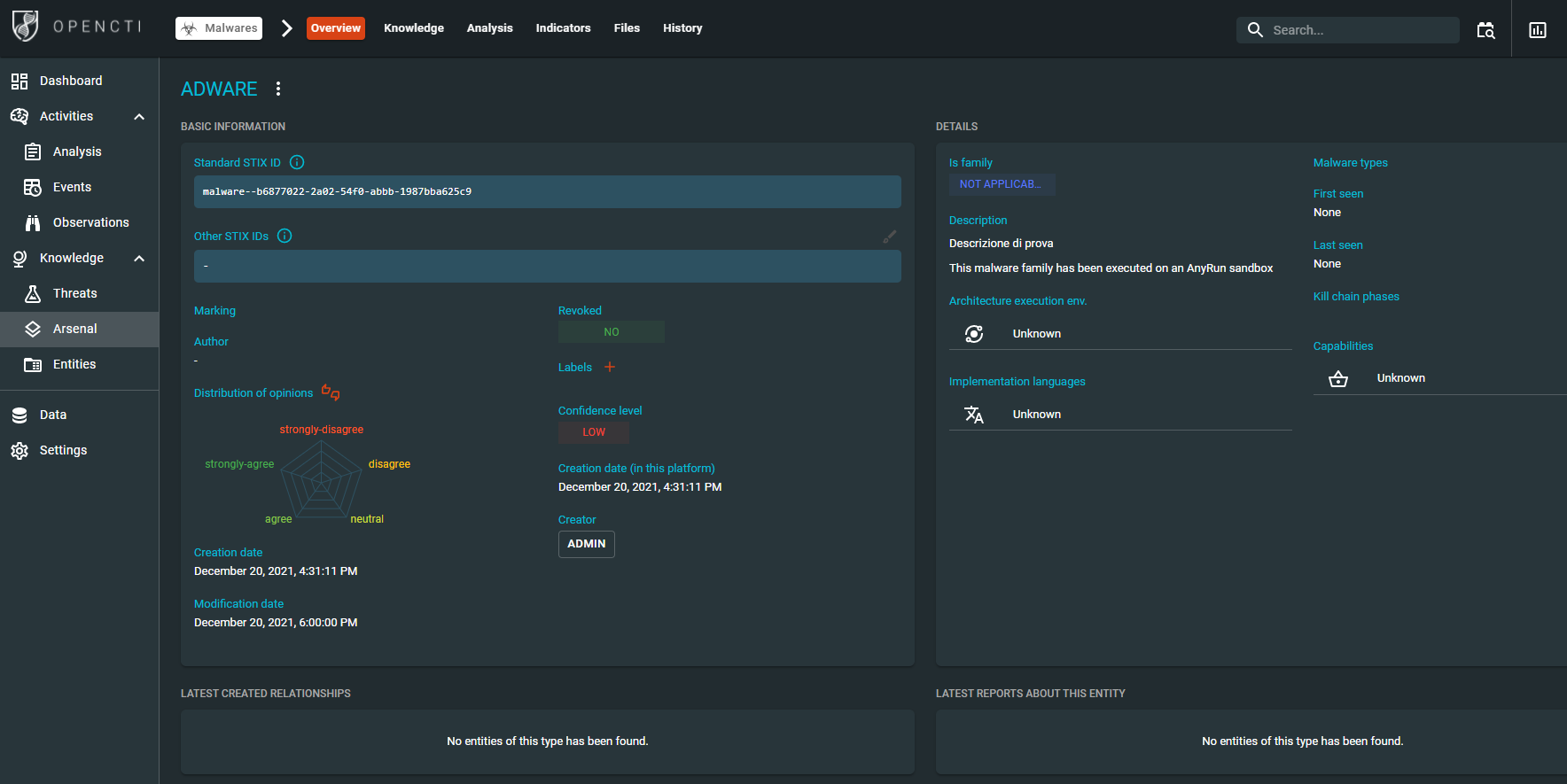
La terza fase è stata il testing del connettore Anyrun.

Nel seguente scenario di test si verifica che:

* Anyrun sia stato correttamente importato nella lista dei connettori



* Anyrun abbia arricchito correttamente le informazioni: come possiamo osservare in figura, anyrun va ad arricchire la descrizione specificando che per quella famiglia di malware è stato eseguito un task nel sandbox di anyrun.

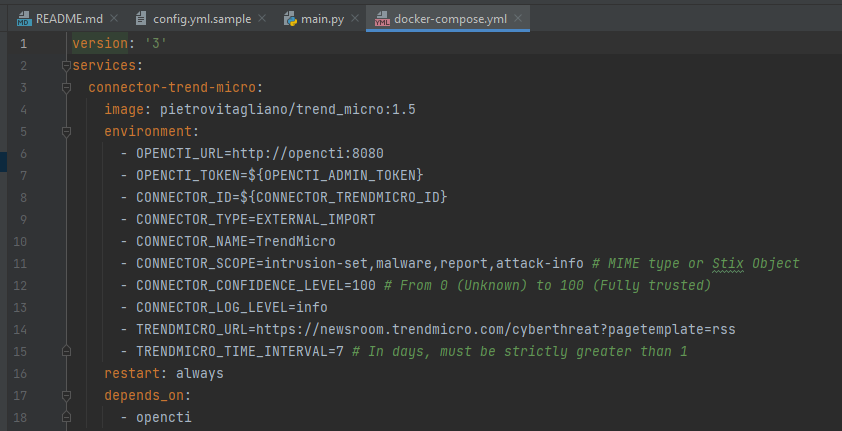


Non è stato possibile arricchire con ulteriori informazioni perché è necessario acquistare una licenza. In conclusione, si può affermare che il test ha avuto esito positivo.

## Configurazione, sviluppo e testing del connettore Trend Micro

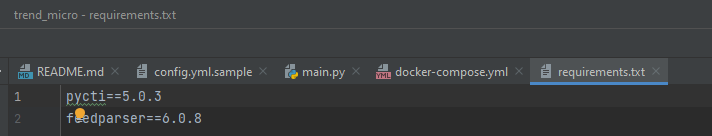
Trend Micro è stato sviluppato come connettore di external import. Nello specifico Trend Micro importa report e osservabili. La prima fase è stata la configurazione del connettore e l’individuazione dei requisiti necessari all’installazione del connettore.

Il seguente file docker-compose.yml contiene la configurazione del connettore:



Come si può osservare in figura, i feed RSS vengono forniti dal seguente url: <https://newsroom.trendmicro.com/cyberthreat?pagetemplate=rss>

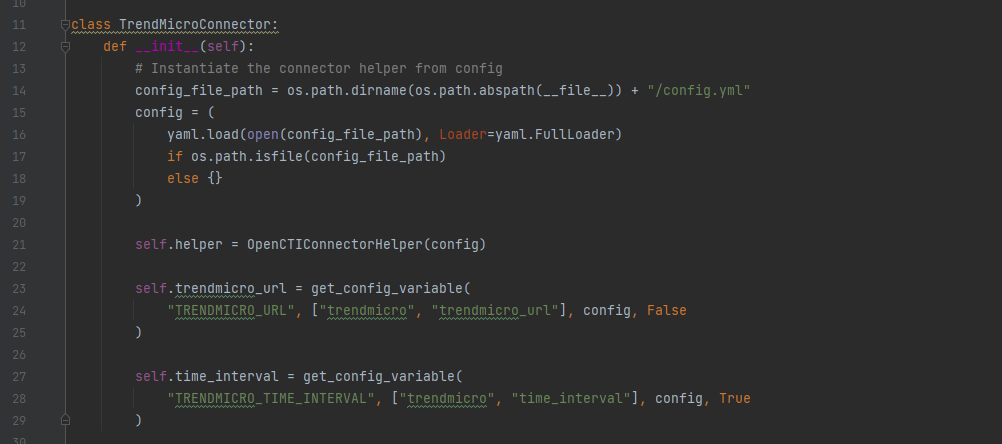
I requisiti di installazione sono contenuti nel file requirements.txt:



La seconda fase è stata lo sviluppo del connettore.

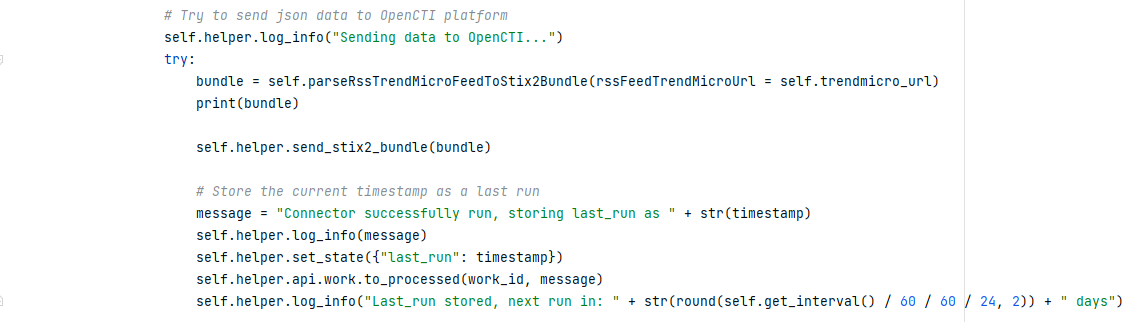
I passaggi salienti dello sviluppo sono:

* Caricamento della configurazione del connettore (contenuta nel file .yml) e connessione all’url di Trend Micro



* Sottomissione del bundle a OpenCTI: il bundle creato racchiude oggetti creati attraverso AttackPattern. Creato il bundle viene inviato a OpenCTI utilizzando la funzione send\_stix2\_bundle().





* Creazione dell’immagine Docker:

docker image build -t pietrovitagliano/trend\_micro:1.52

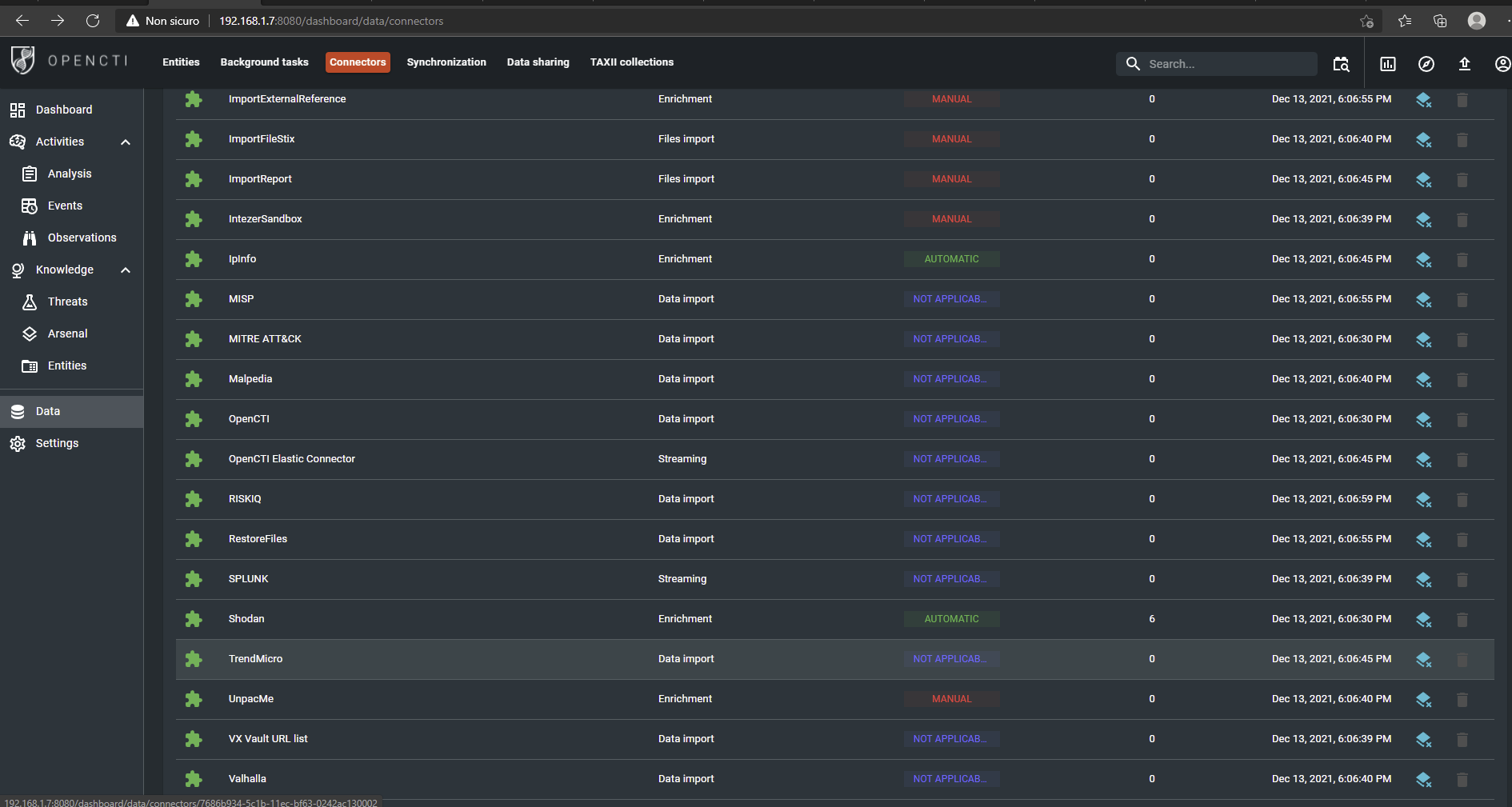
* Aggiornamento dello stack



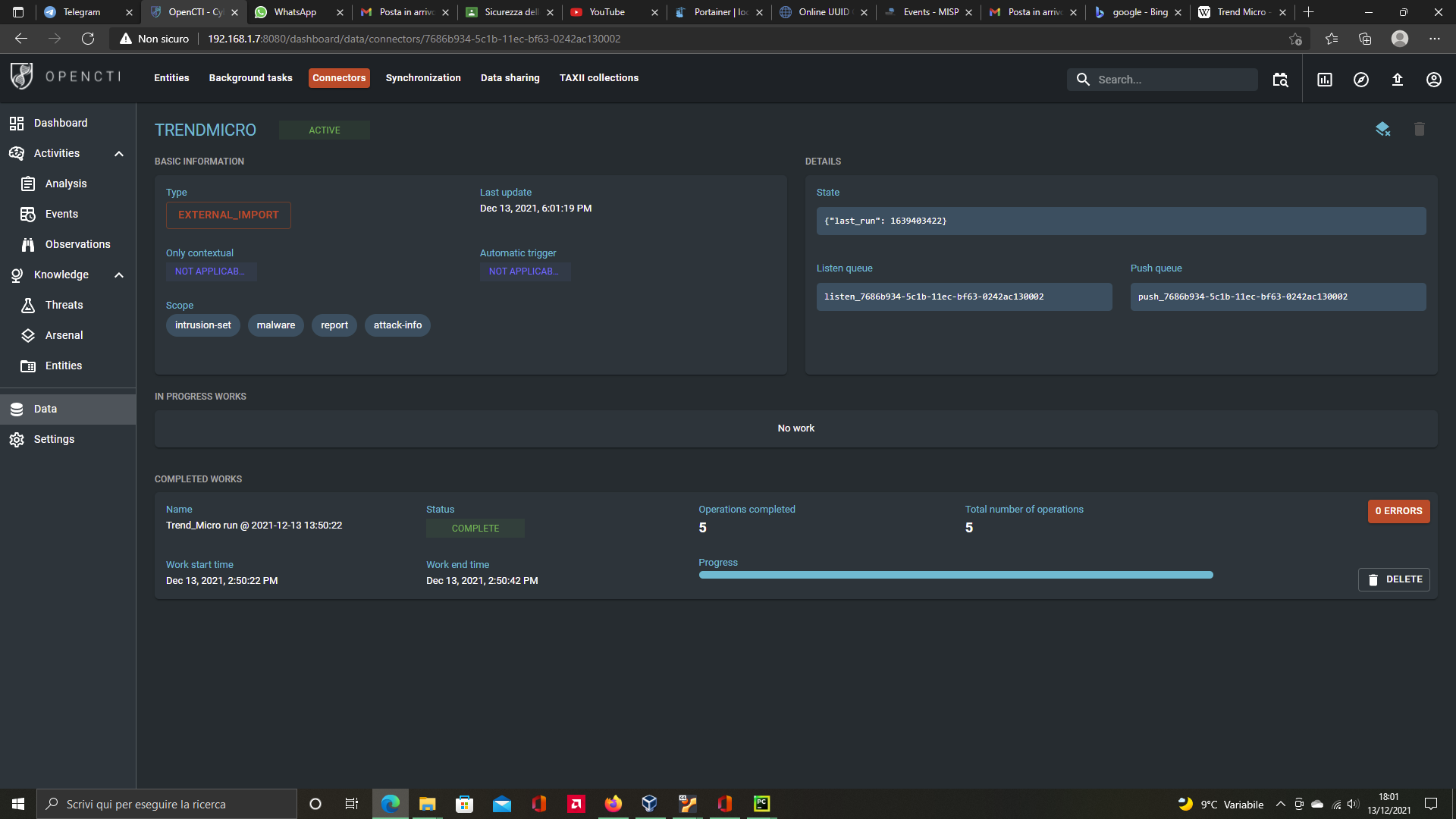
La terza fase è stata il testing del connettore Trend Micro.

Nel seguente scenario di test si verifica che:

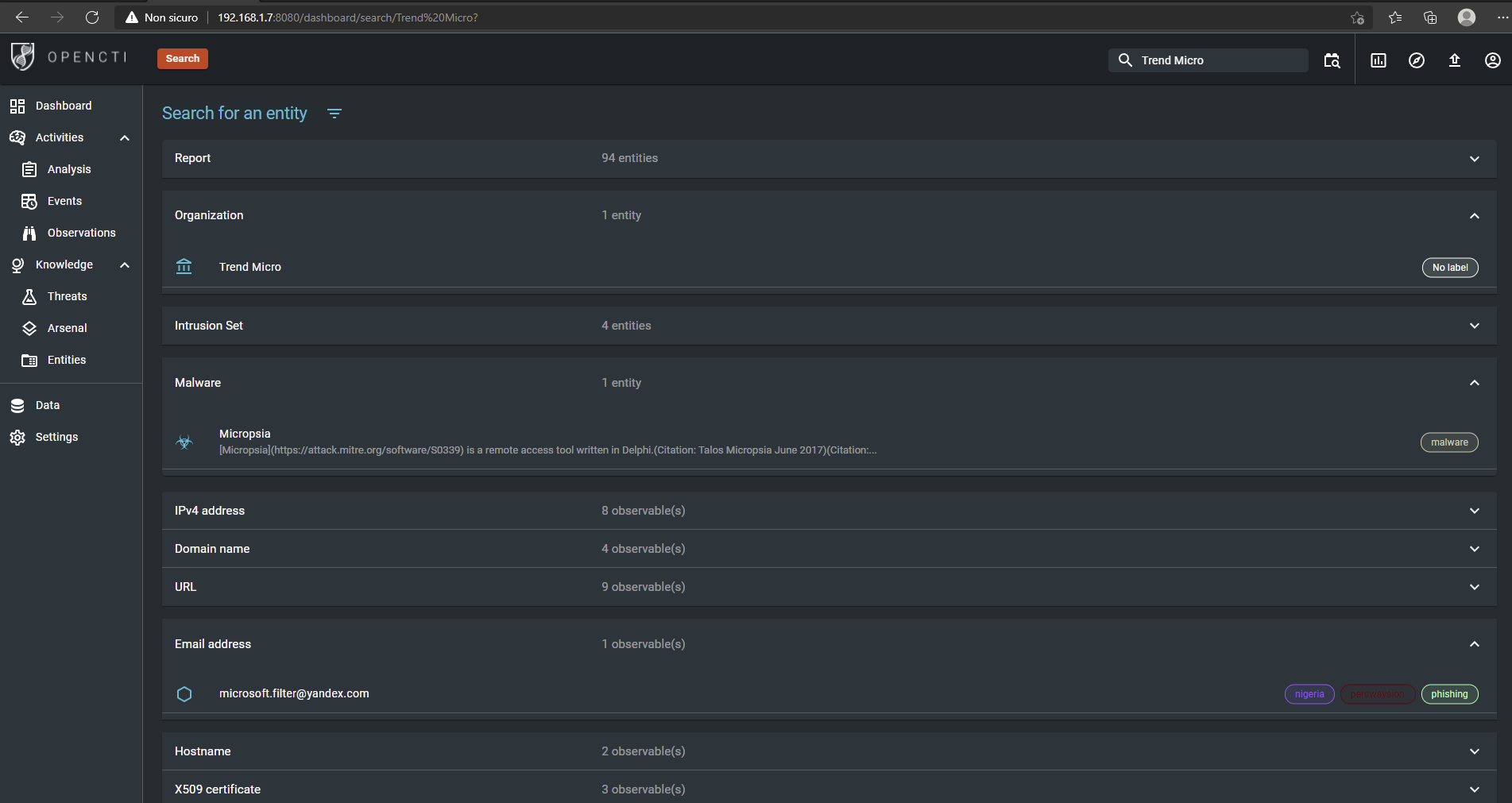
* Trend Micro sia stato correttamente importato nella lista dei connettori



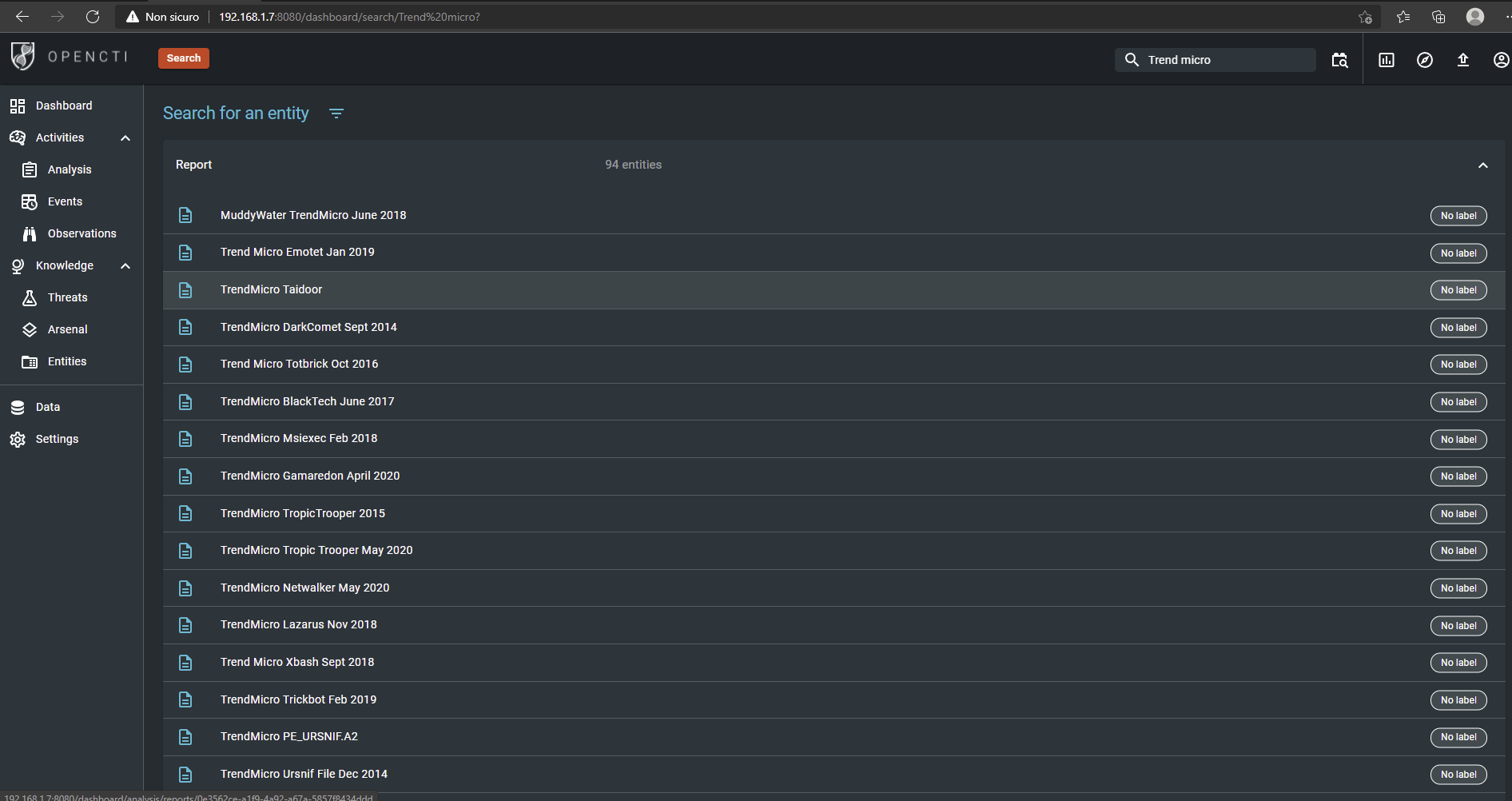
* Trend Micro abbia eseguito dei task



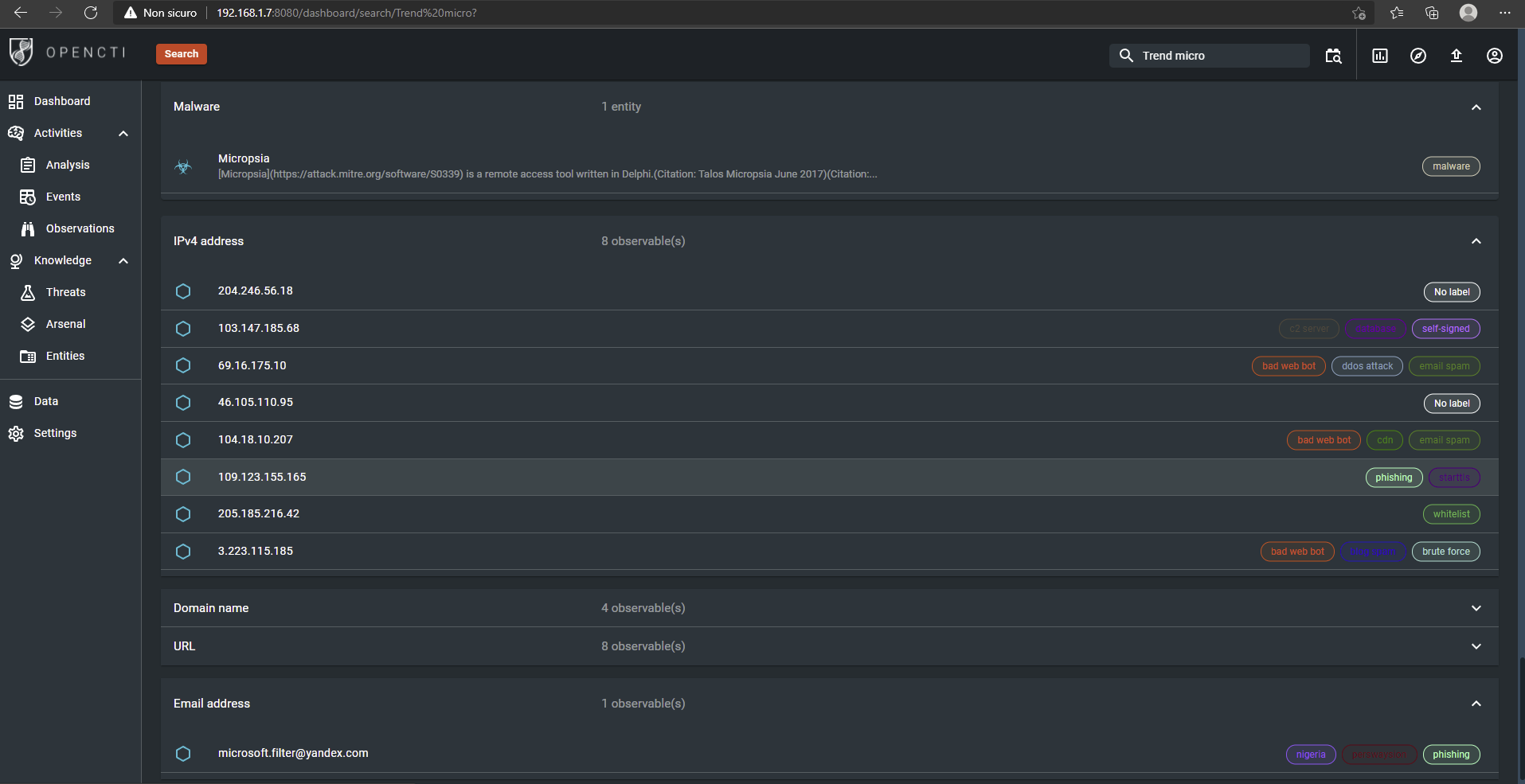
* Trend Micro abbia importato correttamente i dati. Come si può osservare in figura Trend Micro ha importato report, entità e osservabili.



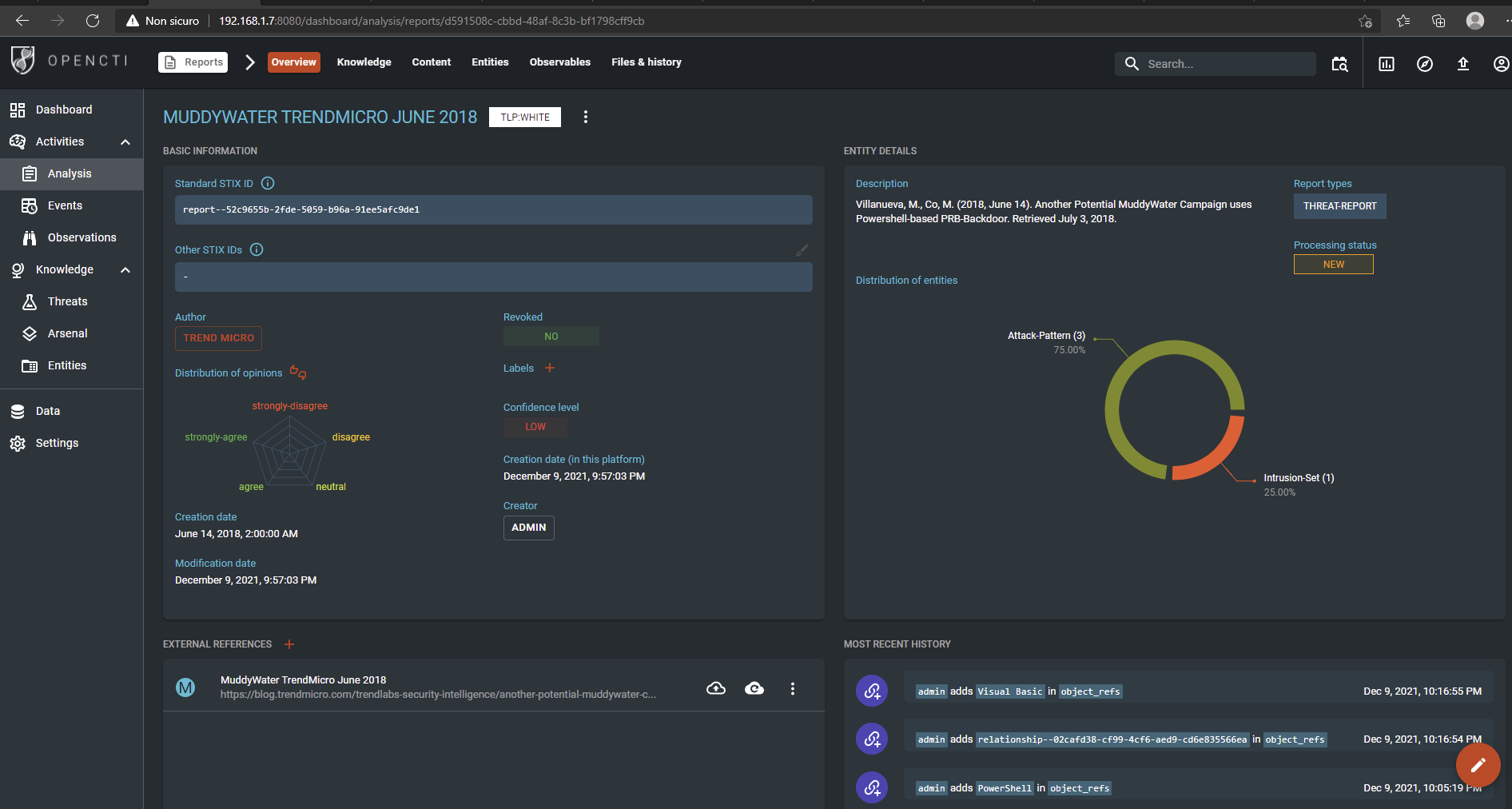
Lista dei reports:



Osservabili relativi ad IP e URL:



Struttura di un report Trend Micro:



In conclusione, si può affermare che il test ha avuto esito positivo.

## Export dei dati

È possibile effettuare l’export delle informazioni (osservabili, minacce, eventi, etc.…) nei seguenti formati:

* Txt
* Json
* Csv

Cliccando in alto a destra e selezionando il formato desiderato è possibile procedere con l’export.

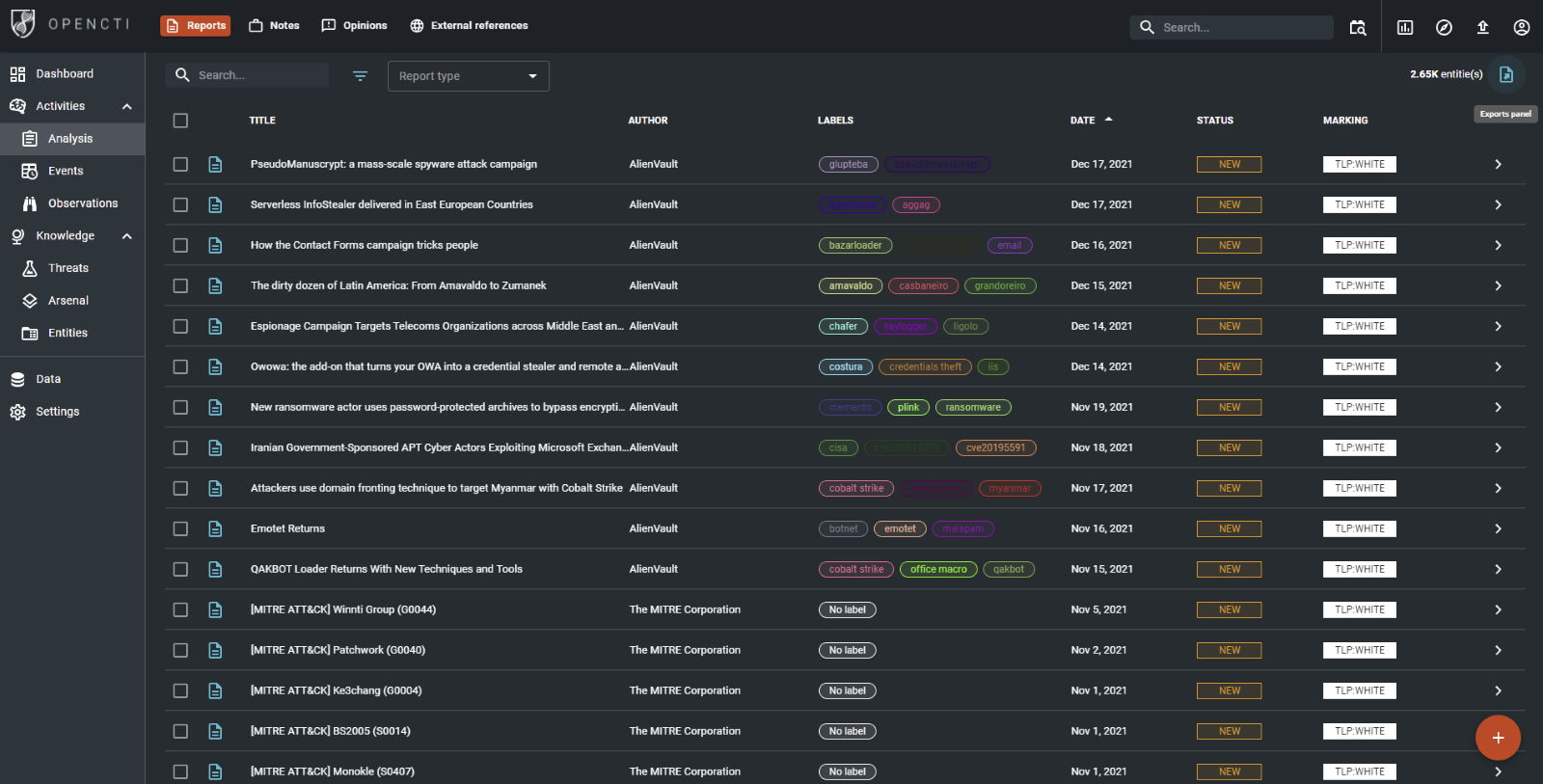


Immagine che contiene testo, monitor, nero, elettronico

Descrizione generata automaticamente