

|  |  |
| --- | --- |
| **Title** | European network of Cybersecurity centres and competence Hub for innovation and Operations |
| **Acronym** | ECHO |
| **Number** | 830943 |
| **Type of instrument** | Research and Innovation Action |
| **Topic** | SU-ICT-03-2018 |
| **Starting date** | 01/02/2019 |
| **Duration** | 48 |
| **Website** | www.echonetwork.eu |

ECHO Malware Analysis Tool concept

|  |  |
| --- | --- |
| **Work package** | WP4 – ECHO Inter-sector Technology Roadmaps |
| **Lead author** | Cagatay Yucel (BU) |
| **Contributors** | Cagatay Yucel (BU)  Ioannis Chalkias (BU)  Dimitris Mallis (BU)  George Aivatoglou (CERTH)  Mike Anastasiadis (CERTH) |
| **Peer reviewers** |  |
| **Version** | V0.3 |
| **Due date** |  |
| **Submission date** |  |

Dissemination level

|  |  |
| --- | --- |
|  | PU: Public |
| X | CO: Confidential, only for members of the consortium (including the Commission) |
|  | EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC) |
|  | EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC) |
|  | EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC) |

#### Version history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Editor | Comments |
| 0.1 | 05/15/2020 | Antonis Voulgaridis (CERTH) | Initial document template |
| 0.2 | 05/27/2020 | George Aivatoglou (CERTH)  Mike Anastasiadis (CERTH) | -Perform an extensive state of the art analysis indicating the functions/assets that separate the proposed tool from the basic competitors on the market -> ***Section 2.5***  -Provide a high level design of the tool, explaining the basic functionality and architecture of the proposed tool ***Sections 2.1, 2.2 and 3.1*** |
| 0.2.1 | 27/08/2020 | Cagatay Yucel (BU)  Ioannis Chalkias (BU) | * Convert the prepared document for malware analysis tool to malware analysis intelligence tool. * Add high level description, concept, updated SOTA, literature review for malware intelligence. |
| 0.2.2 | 03/09/2020 | Cagatay Yucel (BU)  Ioannis Chalkias (BU)  Dimitris Mallis (BU) | * Requirements are added. |
| 0.3 | 23/10/202 | Cagatay Yucel (BU)  Ioannis Chalkias (BU)  Dimitris Mallis (BU) | Concept paper is finalised. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### List of contributors

The list of contributors to this deliverable are presented in the following table:

|  |  |
| --- | --- |
| Section | Author(s) |
| Parts from Introduction & Static and Dynamic Analysis | George Aivatoglou (CERTH) |
| Parts from Introduction & Static and Dynamic Analysis | Mike Anastasiadis (CERTH) |
| Rest of the Paper | Cagatay Yucel (BU) |
| Rest of the Paper | Ioannis Chalkias (BU) |
| Rest of the Paper | Dimitris Mallis (BU) |
|  |  |
|  |  |
|  |  |

#### Keywords

#### Disclaimer

This document contains information which is proprietary to the ECHO consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or parts, except with the prior written consent of the ECHO consortium.

#### Table of contents

[Version history 2](#_Toc40704677)

[List of contributors 2](#_Toc40704678)

[Keywords 2](#_Toc40704679)

[Disclaimer 2](#_Toc40704680)

[Table of contents 3](#_Toc40704681)

[List of figures 4](#_Toc40704682)

[List of tables 4](#_Toc40704683)

[1. Introduction 5](#_Toc40704684)

[1.1 Purpose and scope of the document 5](#_Toc40704685)

[1.2 Structure of the document 5](#_Toc40704686)

[1.3 Relation to other work in the project 5](#_Toc40704687)

[1.4 Applicable and reference documents 5](#_Toc40704688)

[1.5 Glossary of acronyms 5](#_Toc40704689)

[2. ECHO-MAIT Overview 6](#_Toc40704690)

[2.1 Concept 6](#_Toc40704691)

[2.2 Capabilities 6](#_Toc40704692)

[2.3 High Level Requirements 6](#_Toc40704693)

[2.3.1 Functional Requirements 6](#_Toc40704694)

[2.3.2 Non-Functional Requirements 6](#_Toc40704695)

[2.4 Components Requirements 7](#_Toc40704696)

[2.4.1 Component A 7](#_Toc40704697)

[Functional Requirements 7](#_Toc40704698)

[Non-Functional Requirements 7](#_Toc40704699)

[2.5 State of the art analysis 7](#_Toc40704700)

[3. ECHO-MAIT Design 7](#_Toc40704701)

[3.1 High-Level Structure View 7](#_Toc40704702)

[3.2 PROTOTYPE Structure View 7](#_Toc40704703)

[3.3 PROTOTYPE Data View 7](#_Toc40704704)

[3.4 PROTOTYPE Behavioural View 7](#_Toc40704705)

[3.5 PROTOTYPE Deployment View 7](#_Toc40704706)

[3.6 PROTOTYPE Technology View 7](#_Toc40704707)

[3.7 Graphical User Interface Mock-ups 8](#_Toc40704708)

[3.8 Architectural Design Decisions 8](#_Toc40704709)

[3.9 Architecture Validation 8](#_Toc40704710)

#### List of figures

**No table of figures entries found.**

#### List of tables

[Table 1: Applicable documents 5](#_Toc25831210)

[Table 2: Glossary of acronyms and abbreviations 5](#_Toc25831211)

[Table 3: PROTOTYPE Capabilities 6](#_Toc25831212)

[Table 4: Functional Requirements 6](#_Toc25831213)

[Table 5: Most Relevant Decisions 8](#_Toc25831214)

# Introduction

As part of the ECHO project, early Prototypes are one of four technologies under development.

## Purpose and scope of the document

This document captures the main technical aspects of the ECHO Malware Analysis Tool. The document can be used as a reference on the design and as a source for the development.

## Structure of the document

This document introduces the Malware Analysis Tool concept in section 2, followed by the related high level architecture design in section 3.

## Relation to other work in the project

Within Work package 4 this document can be used as source of discussions for Task 4.1, Task 4.2, Task 7.3 and also tasks related to WP2 and WP8.

## Applicable and reference documents

The following documents contain requirements applicable to the generation of this document:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Document Title | Document Reference | Version | Date |
| [PMP] | ECHO Project Manager Plan |  | 0.5 | 01/01/2019 |
| [QAPeer] | ECHO |  | 0.5 | 01/01/2019 |
| [QAReqs] | ECHO Requirements Review QA Checklist |  | 0.5 | 01/01/2019 |
| [QASchedule] | ECHO QA Schedule |  | 0.5 | 01/01/2019 |
| [GA] | Grant Agreement 830943 - ECHO |  | 1.0 | 02/04/2019 |
| [PH] | D1.1 Projet Handbook | ECHO\_D1.1\_v1.0 | 1.41 | 02/05/2019 |

Table 1: Applicable documents

## Glossary of acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| WP | Work Package |
|  |  |

Table 2: Glossary of acronyms and abbreviations

# ECHO Malware Analysis Intelligence Tool (MAIT) Overview

## Concept

The proposed proof-of –concept (TRL-4) tool will be an automated and behaviour-based malware analysis toolkit which is going to identify potential malicious executables files (.exe, .dll) and collect Cyber Threat Intelligence for the file by using online resources.

By utilising a BU-CERT instance of open source state-of-the-art malware static and dynamic analysers (such as cuckoo sandbox) and with the use of open source malware databases, this tool aims to provide a malware signature along with an intelligence report collected from public sources.

The contents of the report include (but are not limited to) the following;

* Chronological data about the malicious file, i.e. first appearance, increase in time
* Any weaponisation in any APT campaigns or cyberattacks in general
* Public information on cyber attribution
* Related vulnerabilities and information on relevance

The tool will seamlessly integrate with the EWS system to share this information as a cyberticket within the organisation and with member constituencies/organisations.

**Input**

Any malware executable file (.exe, .dll).

**Analysis**

Static analysis will provide useful metadata and information which can be used as an input in the final malware analysis report.

The reported information will include:

* Entropy calculation of the executable file.
* MD5, SHA1, SHA256 hash calculation of the malware.
* Extraction of textual features of the executable file.
* The impfuzzy library, calculate hashes from import API of PE files.
* Extraction of metadata from Microsoft office documents, pdf files using the libraries pdfinfo, officemeta.
* Cross-checking of OSINT with extracted binary hashes.

Dynamic analysis will provide useful metadata and information on malware execution and behaviour. The information collected can be used as input in the final malware analysis report.

The reported information will include:

* Cuckoo sandbox analysis results.
* Network monitoring and C&C communication results.
* Disk and function-call usage monitoring results.
* DLL library injects information.
* Memory injection analysis and inspection results.
* Packing/obfuscation/encryption information
* Downloaded external malicious files

**Collecting Cyber Threat Intelligence**

The constantly changing threat landscape and the evolving TTPs of the adversaries require a vigilant stance and increased information-sharing between entities that are affected by cyberattacks. The purpose of malware intelligence is to disseminate the information produced by malware analysis among security teams and organisations. This technical (or non-technical) information, when shared timely and appropriately can provide answers to the following issues[[1]](#footnote-2);

* The timeliness and the source of the malware data
* The contemporary state of the malware and the status of its use cybercriminals
* The expiration of the intelligence
* Appropriate incident response

The deployment and proper use of platforms and tools can extract and share malware related information that will create an identity for the malware, using attributes (e.g. Indicators of Compromise) and proceed into making automatic correlations with other malware or attack campaigns/incidents; according to taxonomies and frameworks like MITRE ATT&CK[[2]](#footnote-3) that ease the communication with other cooperating entities.

Applying malware intelligence and identifying the aforementioned interrelationships can aid in establishing shared behavior, authorship, and provenance, evolutionary artifacts and lineage, author profiles[[3]](#footnote-4), follow the trends in malware and possibly designate relationships between attack groups; leading to cyber attribution and deterrence[[4]](#footnote-5).

Based on extracted data (resulting from static and dynamic analysis) and identified attack vectors (resulting from dynamic analysis) the following malware intelligence can be extracted and reported.

* Crosschecked hashes in the [National Software Reference Library](https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl). database.
* Results from scanning the sample in antiviruses, ClamAVscan, Virustotal, Hybrid analysis, vtsearch, metadefender.
* MITRE | ATT&CK methodologies, mapped to malware functionality (table in last page)
* Chronological data about the malicious file, i.e. first appearance, increase in time
* Publicly available weaponisation in any APT campaigns
* Publicly available information on cyber attribution
* Related vulnerabilities and information on relevance
* Digital Signatures of the malicious executable

**Attack visibility-Out**

* Illustrates the entire attack (path) of a malware in the system in a process tree, enriched with contextual and threat intelligence data.
* Provides with an analytical report of the libraries and functions that a malware uses, the network activity of the malware and the files that the malware creates-deletes during its execution.

## Capabilities

The capabilities of the Malware Analysis Tool can be found on the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Functionality | Description | Note |
| Static Analysis |  | Perform static analysis in order to identify useful metadata of the executable files. | The tool will try to extract useful information for the .exe file without actually running it. Some features and metadata are:   * Entropy calculation of the executable file. * MD5, SHA1, SHA256 hash calculation of the malware. * Extraction of textual features of the executable file. * impfuzzy library, calculate hashes from import API of PE files. |
| Dynamic analysis |  | Identify the processes and behaviour of the malware by running it in a sandbox. | The tool will run the .exe file in a secure environment in order to:   * address the running procedures * monitor the network connections * identify the function-calls. |
| CTI Collection |  | This component will collect publicly available information on the executable file and store this information safely. | This tool will;   * Look for hashes in the [National Software Reference Library](https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl). database. * Scan the sample in antiviruses such as ClamAVscan, Virustotal, Hybrid analysis, vtsearch, metadefender. * Utilisie MITRE ATT&CK methodologies mapping to malware functionality * Find chronological data about the malicious file, i.e. first appearance, increase in time, malware lineage * Utilise known APT and Threat reports that the malware hash is found * Look for any other malware that the malware tries to download * Look for any weaponisation in any known APT campaigns * Use publicly available information on cyber attribution * Cross-check related vulnerabilities and information on relevance * Provide with digital signing of the malicious executable |
| Reporting |  | The collected information from static, dynamic analysis and the collected intelligence will be reported and integrated and prepared as ready to share to the EWS system. | The report will contain:   * A MITRE version of the collected intelligence * An EWS compatible json version of the intelligence * Collaboration with Trust& Quality Tool (VST)   Will be reported after the successful completion of the analysis. |

Table 3: PROTOTYPE Capabilities

## High Level Requirements

### Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Title | Description | Priority | Difficulty |
| ECHO-MAIT-FUNC-0010 | **Static analysis** | The tool MUST be capable of analysing a malicious file without running it and extract information from its binary form. | High | Low |
| ECHO-MAIT-FUNC-0020 | **Dynamic Analysis Network simulation** | The tool’s connection with the open source dynamic analysis tool MUST involve a network simulator such as INetSim to reveal network communications of the binary. | High | Medium |
| ECHO-MAIT-FUNC-0030 | **CTI Collection – Analysis** | The tool MUST be able to accomplish the aforementioned analyses within the capabilities. | High | Hard |
| ECHO-MAIT-FUNC-0040 | **MITRE Export** | The tool SHOULD be able to export STIX v2.0 | Medium | Medium |
| ECHO-MAIT-FUNC-0050 | **EWS Export** | The tool MUST be able to export reports to EWS. | High | Medium |
| ECHO-MAIT-FUNC-0060 | **Collaboration with TQM tool.** | The tool COULD communicate with TQM tool to assess the reports quality measures. | Low | Medium |

Table 4: Functional Requirements

### Non-Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Title | Description | Priority | Difficulty |
| ECHO-MAIT-NFUNC-0010 | **Tool adaptation** | The sandbox SHOULD to be as close to a real environment as possible | High | High |
| ECHO-MAIT-NFUNC-0020 | **Analysis Avoidance Awareness** | General analysis avoidance techniques of malware SHOULD be considered. | High | High |
| ECHO-MAIT-NFUNC-0030 | **CTI completeness** | The generated CTI COULD be as complete as possible. | Medium | Medium |
| ECHO-MAIT-NFUNC-0040 | **CTI timeliness** | The generated CTI COULD be as timely/fresh as possible. | Medium | Medium |
| ECHO-MAIT-NFUNC-0050 | **PDF Reports Output** | Automated output of the report SHOULD be produced. | Medium | Low |

## Components Requirements

### Component A

#### Functional Requirements

#### Non-Functional Requirements

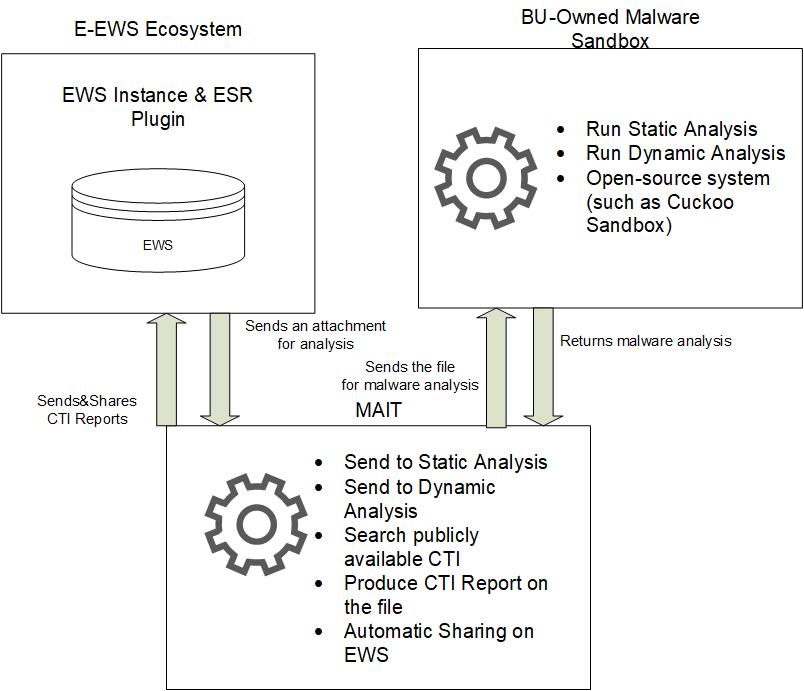
## State of the art analysis

A state-of-the-art analysis can be found on the table below. As the tool to be developed is aimed at providing an intelligence report on top of the malware analysis, there are limited competitors on the market. The features are selected based on the CTI sharing properties of the tools, as open source sandboxing solutions are expected to be used in this project.

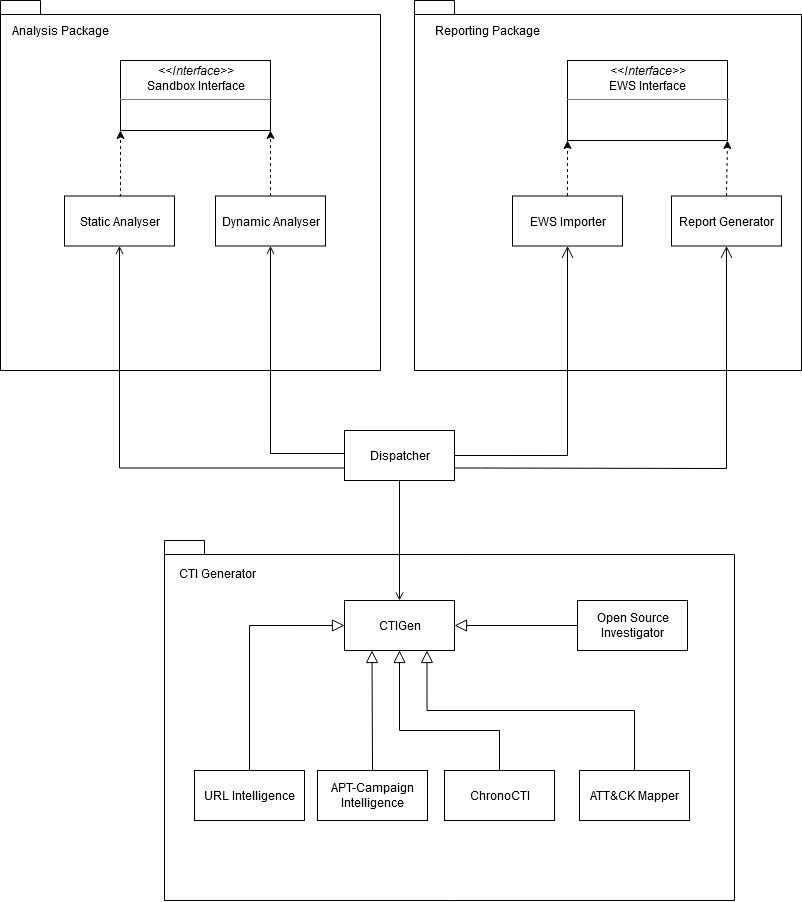
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Malware Intelligence Tool/ Capability | MISP | The-Hive | Google Rapid Response | VT Intelligence (Commercial) | Any.Run | OpenCTI |
| Database/Elastic Search | Yes | Yes | Yes | Yes | Yes | Yes |
| Correlations within attributes and indicators | Yes | Yes | No | Yes | No | Yes |
| GUI | Yes | Yes | Yes | Yes | Yes | Yes |
| Sharing Functionalities | Yes | Yes (with MISP) | No | Yes | Yes (with MISP) | Yes |
| Synchronisation w Feeds/sources | Yes | Yes( with Multiple instances of Cortex) | Yes | Yes | Yes | Yes |
| Pushing/Publishing CTI | Yes | Yes (with MISP) | Yes | Yes | Yes | Yes |
| Following a Taxonomy | Several Taxonomies are followed, provided functionality by tags | Yes | No (aimed at computer forensics practices) | Yes | Yes | Yes |
| STIX/MITRE support | Yes | Yes | No (aimed at computer forensics practices) | Yes | Yes | Yes |
| Alert Management | Yes (in the form of events) | Yes | No | No |  | Yes |
| Dashboarding | Yes | Yes | Yes | Yes | Yes | Yes |
| CTI Metric Management | Yes | Yes | No | No | No | No |

# ECHO Malware Analysis Tool Design

## High-Level Structure View



## PROTOTYPE Structure View

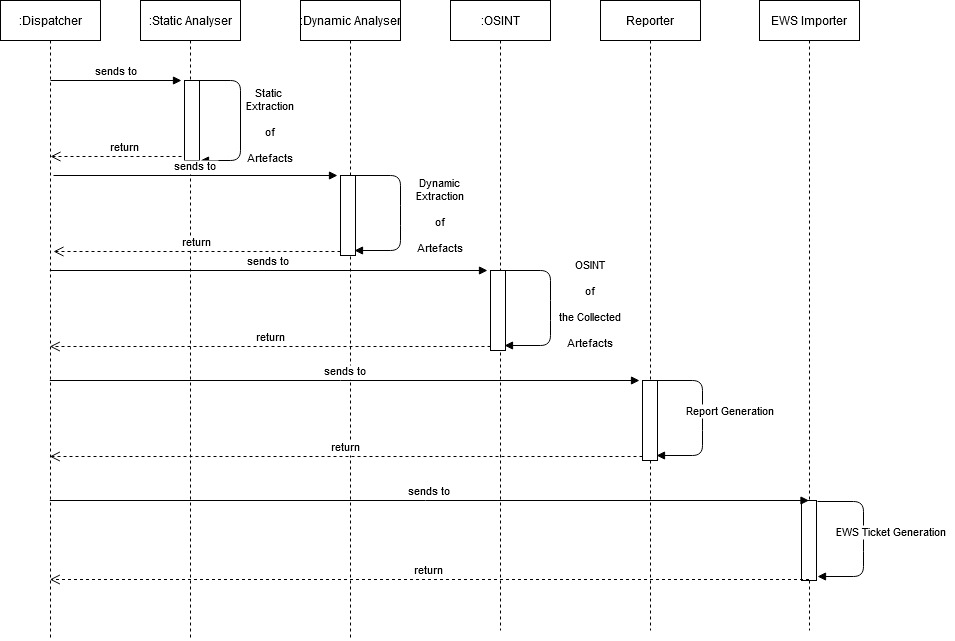


* **Dispatcher**: This class will be the main class to define and operate the workflow between packages. This will also be the class that greets web hooks and receives the files that are uploaded for analysis.
* **Analysis Package:** This package will contain the interfaces with the selected tools for sandboxing and analysis. The interfaces will also contain the necessary scripts to administer the virtualisation tool for the sandboxing.
* **Reporting Package:** This package will be responsible for creating a cyberticket inside the EWS environment with the collected and correlated results of the analyses. A PDF report will also be generated depending on the configuration of the plugin.
* **CTI Generator:** This package will contain the novelty of this tool; all the malware intelligence collections including URL intelligence, APT campaign and attribution challenges, chronological threat intelligence of the malicious file and the mapping of this file to the ATT&CK interface will be handled and tackled within this package.

## PROTOTYPE Data View

The MAIT tool will be working on malicious files and their artefacts found online. The application server and open-source sandbox instance will be tightly coupled. The data view and the information flow will be implemented depending on the sandbox tools and the generated CTI content. However, for the first version of this tool, a database implementation is not foreseen, therefore the data view is left empty.

## PROTOTYPE Behavioural View

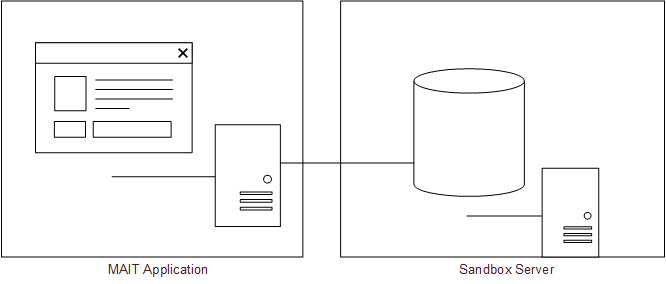


## PROTOTYPE Deployment View

The application will be deployed to a lab environment where the open source sandbox server will be within the same network perimeter. The MAIT application will receive the file to be examined from the EWS server (attachments and file uploads), and then the file is transferred to the sandbox server for static and dynamic analysis. The results and collected artefacts will be investigated on the application server for further CTI extraction.

The communication of the files from EWS server to the application will be established by the web hooks that will have been provided by the EWS API. The results and the created CTI ticket will be shared utilising SDK.

The planned TRL of this tool will be level 4 which is the proof-of-concept level development of the MAIT tool. Therefore, a one-to-one plugin integration is not planned, and the demonstrations will be done with the servers that are available to Bournemouth University Cyber Security Research Lab and the development server of the EWS.



## PROTOTYPE Technology View

The application will be developed using the following technologies:

* Python 3.8
* Flask App for Web hooks
* Cuckoo Sandbox for analysis
* Radare2 and Python Interface of Radare2 for static analysis
* Virtual box API for operating virtual machines
* Ubuntu 20.04 for host operating system.
* EWS development environment and web hooks will be tested utilising ngrok.io

## Graphical User Interface Mock-ups

The application will not have a specific GUI and will be following a sequential process for the analysis. This flow is given in the behavioural model of the tool and will be implemented within the dispatcher’s class. The output of the tool will be a downloadable PDF report and a cyberticket added into the EWS system. The report and the ticket will be generated by the Reporting package and will be accessible within the EWS dashboard.

## Architectural Design Decisions

|  |  |  |
| --- | --- | --- |
| **Design Decision** | **Rationale** | **Iteration** |
| EWS Integration for Reporting | Instead of generating a separate GUI for this plugin, the EWS cyberticket GUI will be utilised to increase the efficiency. This will also allow the user to further edit, annotate or manage the collected information. | 1 |
| No Data Storage | The plugin does not aim to store any of the collected information for the proof-of-concept version. After the successful development in this TRL level, the malware information will be stored for machine learning dataset generation. | 1 |

Table 5: Most Relevant Decisions

## Architecture Validation

With the completion of the aforementioned architecture, the validation will be done by a series of demonstration and presentation of the showcases for all the use cases. Test modules for Static and Dynamic analysis will be further described in the deliverable D4.6 Inter-sector prototype verification plan. However, the collection of CTI is highly dependent on the availability of the publicly available resources therefore only demonstrations are planned for the CTI generation tests.

1. https://intel471.com/Malware%20Intelligence%20-%20Mar%202019.pdf [↑](#footnote-ref-2)
2. https://www.misp-project.org/features.html [↑](#footnote-ref-3)
3. VirusBattle: State-of-the-art malware analysis for better cyber threat intelligence (Miles et al, 2014) [↑](#footnote-ref-4)
4. https://www.fireeye.com/blog/threat-research/2019/08/apt41-dual-espionage-and-cyber-crime-operation.html [↑](#footnote-ref-5)