Information & Security

C. Yucel et al.
vol. 50, no. 1 (2021): 49-65
https://doi.org/10.11610/isij.5024

Published since 1998 ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

MAIT: Malware Analysis and Intelligence Tool

***Cagatay Yucel* (✉), *Adam Lockett*,**
Ioannis Chalkias*, *Dimitrios Mallis*, *Vasilios Katos

Bournemouth University, Fern Barrow, Poole, Dorset,
BH12 5BB, United Kingdom

ABSTRACT:

Malware is the instrument that delivers the decisive blow in cyber-attacks. A first-time presented malware or an updated malware can remain undetected and stealth until the attackers achieve their objectives. Information about malware and its use needs to be shared with other entities that are protecting their infrastructure from the same or similar threats. Malware intelligence can be critical in a rapidly changing threat landscape, allowing entities to respond to incidents in a successful and timely manner. We introduce the *Malware Analysis and Intelligence Tool,* a tool that uses state-of-the-art malware analysers (static and dynamic), combined with open-source malware databases to provide a malware signature and an intelligence report that is collected from publicly available cyber threat intelligence sources. The tool can be used to obtain chronological data for a malicious file, related vulnerabilities, and towards providing attribution and techniques, tactics and procedures when used in attacks from Advanced Persistent Threat groups.

ARTICLE INFO:

RECEIVED: 25 JUNE 2021
REVISED: 10 SEP 2021
ONLINE: 16 SEP 2021

KEYWORDS:

malware analysis, cyber threat intelligence, malware intelligence, static analysis, dynamic analysis

Introduction

For the majority of cyber-attacks, malware is the component that delivers the decisive blow to the victim. Regardless of the increased capabilities of the attackers in delivering the attack or avoiding detection,[1] the malware plays an important role in the success (or failure) of the attack. A first time introduced mal-

✉ Corresponding Author: E-mail: cyucel@bournemouth.ac.uk

ware, or a new version of a known malware, can achieve its objectives while remaining undetected by the detection mechanisms that the infrastructure is using when trying to protect its premises from being hacked.

The impact of such incidents is rendering malware analysis a critical part of the defence against malware. A successful analysis of a malware sample can determine the structure, behaviour, ownership, its lineage and the relations with other malware. This can consequently estimate the potential impact of its operation inside a targeted infrastructure. The analysis of the structure of the malware does not require its execution and is able to check its properties (i.e. static analysis), a fact which allows new or unidentified versions of malware to remain undetected. On the other hand, the behavioural analysis of malware can identify specific system activities or software behaviours, which can raise alerts for malicious activity even for unknown malware (i.e., dynamic analysis).[2]

The results of malware scrutinising can become valuable assets in the adverse environment that constitutes the rapidly changing threat landscape. This information can and should be disseminated among teams and organisations that suffer from similar threats and attacks. The correct, accurate, and timely sharing of malware intelligence to other parties is imperative for identifying the relationships between malware and their authors. Its final aim objective is to attribute their use to personas or even Advanced Persistent Threat (APT) groups that use their uniquely designed malware to wage cyber campaigns against nations or major enterprises in order to gain a proactive stance against them.[3] The attribution of malware to specific APT groups is a task that faces a series of technical challenges like the architecture and the geography of the Internet, the lack of source code, the obfuscation techniques, the fake traces of authorship, spoofed IP addresses, or even legal issues related to the operation of the APT groups and their sponsors.[4, 5]

In this effort, Cyber Threat Intelligence (CTI) can benefit from tools and platforms that can extract and analyse information from malware so that the threat analysts can share it with others. The purpose of using such software is the accurate and timely sharing of information that is related to an incident and all the issues mentioned above. With our research, we introduce the Malware Analysis and Intelligence Tool (MAIT), a tool that is using state-of-the-art malware analysers (static and dynamic), combined with open source malware databases to provide a malware signature and an intelligence report that is collected from public sources. The tool is seamlessly integrated, in the form of a cyber ticket, with the Early Warning System (EWS) that is built for the European network of Cybersecurity centres and competence Hub for innovation and Operations (ECHO) as a platform for secured collaborative information sharing of cyber-relevant information.[6] With the use of MAIT, the user of the EWS can use any malware executable as input in order to retrieve a report containing information consisting of the following:

- Chronological data about the malicious file; i.e., first appearance, increase in time;
- Any weaponisation in any APT campaigns or cyber-attacks in general;

- Public information on cyber attribution;
- Related vulnerabilities and information.

The remainder of this paper is structured as follows. The next section presents the related work and background research on CTI on Malware analysis. Section 3 discusses the development and the structure of the tool. Section 4 includes the test results and evaluation of the study in the form of two demo cases. Finally, conclusions, tool's limitations, and future work are given in the last section.

Literature Review on Malware analysis and CTI

Cyber Threat Intelligence Collection on Malware Analysis

One of the first studies in the intersection of the malware analysis and cyber threat intelligence is the work that has been conducted by Miles et. al.[7] In this work, the malware analysis is directed towards identifying and extracting the artefacts for correlating malware samples. For this aim, function tracers, path tracers, and relationships between samples are deployed, and these artefacts are then utilised to create clusters out of samples.

A proactive approach is taken in the study of AZSecure Hacker Assets Portal,[8] compiling and analysing information from the CTI that is collected from DarkNet and open web sources. In this study, the blog entries and posts from these sources are sieved, classified based on their attachments and source code postings. The presentation of this tool contains the search, comparison and visualisations for source code and attachments provided with useful dashboards.

Cyber Knowledge Graphs (CKG) have been a very conventional tool when it comes to visualising the CTI data. In the work of Piplai et. al.,[9] the CKG of a malware sample is generated by collecting the performance and tracking metrics from the execution of the sample. This CKG is then fused with the information extracted from CTI reports and blogs. Although not exclusively on malware, another study that generates CKGs from open source cyber threat intelligence sources is SecurityKG,[10] where the html and pdf reports are parsed, and correlated on a graph or an SQL database.

A very similar tool is developed by Tan et. al.[11] This tool automates the extraction tools for various analyses grouped in the categories of hash and metadata extraction, anti-virus detection information, portable executable format (PE) specific information, host related Indicators of Compromise (IOCs), Network related IOCs and information about program semantics and functionality. These analyses are retrieved from CTI sources and the static analysis. The developed tool also has a modular/extendable back-end with containerisation.

Our work differentiates from the literature with the following novelties:

- The analysis of MAIT takes a comprehensive approach around malware, including static and dynamic analyses as well as memory dump analysis utilising Volatility framework;

- There is an overwhelming amount of information about the malware samples on online CTI sources. MAIT's analyses do not only include the bulk retrieval of the artefacts but also allows automated timeline generation, chronological analysis, APT attribution and MITRE ATT&CK [12] representation;
- The artefacts that are extracted from static, dynamic and memory analysis are queried for another iteration to retrieve relevant host and network-related queries. These queries include, but are not limited to, geolocation, passive DNS information, and historical whois information.

State of the Art Tools

At the time of writing, there are many automated solutions for analysing malware using dynamic and static analysis techniques. In this section, some of the most popular solutions that are either open-source or provide some free of cost features are outlined and compared with MAIT.

Any.Run

Any.Run [13] is a web-based hybrid sandbox that incorporates both automated and manual features for analysing malware. Any.Run provides a real-time view of processes running in the Virtual Machine (VM) that the malware has been uploaded to and executed in, as well as extracting IOCs such as HTTP requests and malware signatures.

While Any.Run provides an extensive report of the detected malicious behaviour, the network operations, and the IOCs, its community version does not collect much CTI information for the malware sample. This can be crucial to determining the TTPs of the uploaded malware sample and the threat actors it is related to. In addition to this, there is limited CTI for the potentially malicious URL and IP address IOCs extracted from the malware. This information can determine if an indicator is malicious and the types of threat it is associated with.

Cuckoo sandbox

Cuckoo Sandbox [14] is an open-source automated malware analysis system that can analyse malware in virtualised environments with various operating systems such as Windows, Linux, and Android. Additionally, Cuckoo Sandbox has many integration features, which include the Cuckoo API and plugins like Volatility. Once the analysis of the uploaded malware sample is complete, a report that provides information about malicious behaviour, network traffic, and signatures is generated.

One of the main advantages of Cuckoo Sandbox for malware analysis is that the automated analysis process is highly configurable to tailor for different user requirements, such as features to add new YARA rules to capture specific data when a malware sample is executed.

Although Cuckoo Sandbox supplies a detailed analysis report of behavioural and static information about a malware sample, it does not provide much CTI depth, such as associated APT campaigns, the timeline of the sample, and TTPs.

Also, while URLs can be submitted for analysis in Cuckoo, only behavioural analysis from executing the URL in a VM is done, and no CTI, like information about the URL's domain and malware associated with the URL, is collected.

VirusTotal

VirusTotal [15] is a web-based tool for scanning URLs and files with antiviruses, in addition to analysing the given file or URL in multiple automated sandboxes. The antivirus scanning provides a quick indication of whether a file is malicious. Analysis results, including behaviour and network operations, are also reported quickly.

Unlike Any.Run and Cuckoo Sandbox, VirusTotal provides actionable URL and IP address CTI including WHOIS records, SSL certificates, associated malware, and what the URL or IP address is used for e.g., command and control. While VirusTotal does provide some intelligence into what sort of threat a URL or IP address is associated with, it does not go into much depth. This is the same for malware samples submitted to VirusTotal. Although VirusTotal antivirus results determine the type of malware a sample is, it provides limited CTI into the Tactics, Techniques, and Procedures (TTPs) of the malware and the associated threat groups.

Comparison of current solutions with MAIT

While the previously mentioned solutions are great for providing automated dynamic and static analysis of malware and detailed reports, they all have limitations in the context of CTI depth for both malware samples and the extracted IOCs.

MAIT addresses these limitations by providing detailed CTI for both malware samples and their URL and IP address IOCs. This includes associated APT campaigns and threat actors, chronological intelligence about when the malware was first identified, TTPs of the malware sample and a MITRE ATT&CK mapping of TTPs identified for the malware. All of this information provides actionable CTI about the malicious actions and type of threat of a malware sample or IOC.

Development of the Tool

Requirements

For the tool to meet its objectives, it needs to adhere to the following functional requirements.

When designing and developing this software, there are also non-functional requirements that are taken into account. The following Table 2 contains these requirements.

Structure of the Tool

MAIT tool consists of the main components of Analysis, Reporting and CTI packages. The Analysis package contain Static and Dynamic Analysis modules. Static analysis provides useful metadata and information including strings, functions,

Table 1. Functional requirements of MAIT tool.

Identifier	Title	Description
FRQ-01	Static Analysis	The tool must be capable of extracting valuable metadata information while also determining several aspects from the binary form of a malicious file without executing it.
FRQ-02	Dynamic Analysis	MAIT must be connected with an open-source dynamic analysis tool. Cuckoo Sandbox is selected and integrated.
FRQ-03	CTI collection Analysis	The tool must be able to accomplish the aforementioned analyses within the capabilities of CTI collection.
FRQ-04	MITRE ATTCK Export	The tool should be able to export STIX v2.0 graph of STIX Domain Objects (SDOs) and STIX Relationships Objects (SROs) for the representation of CTI.
FRQ-05	EWS Export	The tool must be able to export reports to the EWS, including the information resulting from static and dynamic analyses, as well as those coming from the CTI collection processes.

Table 2. Non-functional requirements of MAIT tool.

Identifier	Title	Description
NFRQ-01	Tool Adaptation	The sandbox that will be hosting the dynamic analysis tasks should be as close to a real IT environment as possible.
NFRQ-02	Analysis Avoidance and Awareness	General analysis avoidance techniques of malware should be considered.
NFRQ-03	CTI Completeness	The generated CTI could be as complete as possible.
NFRQ-04	CTI Timeliness	The generated CTI could also be as timely/fresh as possible.

network and IP addresses, entropy and hash calculation (MD5 and SHA-256) of the malware, as well as extraction of textual features of the executable file. Moreover, the impfuzzy, officemeta, and pdfinfo libraries are included in this process for the calculation of Fuzzy hashes from import API of PE files, and the extraction of metadata from Microsoft Office documents and pdf files respectively.

Dynamic analysis, on the other hand, provides useful metadata and information after the execution of the suspect binary and monitoring of its behaviour in an isolated environment (i.e., Cuckoo sandbox). Network monitoring and C&C communication results are provided in this feature with the support of a network simulator. Disk and function-call usage monitoring results are also included in the analysis along with information about DLL library injects, memory dump and injection analysis and results. Finally, downloaded external malicious files and Packing/obfuscation/encryption information complement the dynamic analysis of MAIT.

Finally, the feature of CTI collection is built upon the data extracted from the other two features, static and dynamic analysis, and the identified attack vectors. In this module, the focus is on APT campaigns, cyber attribution, malware-related TTPs, and chronological data coming from antivirus (e.g., VirusTotal) that delineate the timeliness and the source of malware data. Furthermore, MAIT offers mapping of the retrieved TTPs to the MITRE ATT\&CK framework in layers. URL and IP intelligence is also provided by the tool, including information like related vulnerabilities, digital signatures, and downloaded external malicious files. The collected malware intelligence is structured in a form of a (non)-technical report and shared again in a complete and timely manner within the EWS CTI sharing and incident handling system.

Overall class diagram of the tool can be found in Figure 1.

- Dispatcher: This class is the main class to define and operate the workflow between packages. This is the class that also greets webhooks and receives the files that are uploaded for analysis.
- Analysis Package: This package contains the interfaces with the selected tools for sandboxing and analysis. The interfaces also contain the necessary scripts to administer the virtualisation tool for the sandboxing.
- Reporting Package: This package is responsible for creating a cyberticket inside the EWS environment with the collected and correlated results of the analyses.
- CTI Generator: This package contains 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 is handled and tackled within this package.

Integrated Technologies

In this section, the technologies and open-source solutions that are integrated into the workflow of this tool are given. These solutions are given in four subsections, malware analysis, cyber threat intelligence APIs, software support and the tools that are utilised for the representation of data.

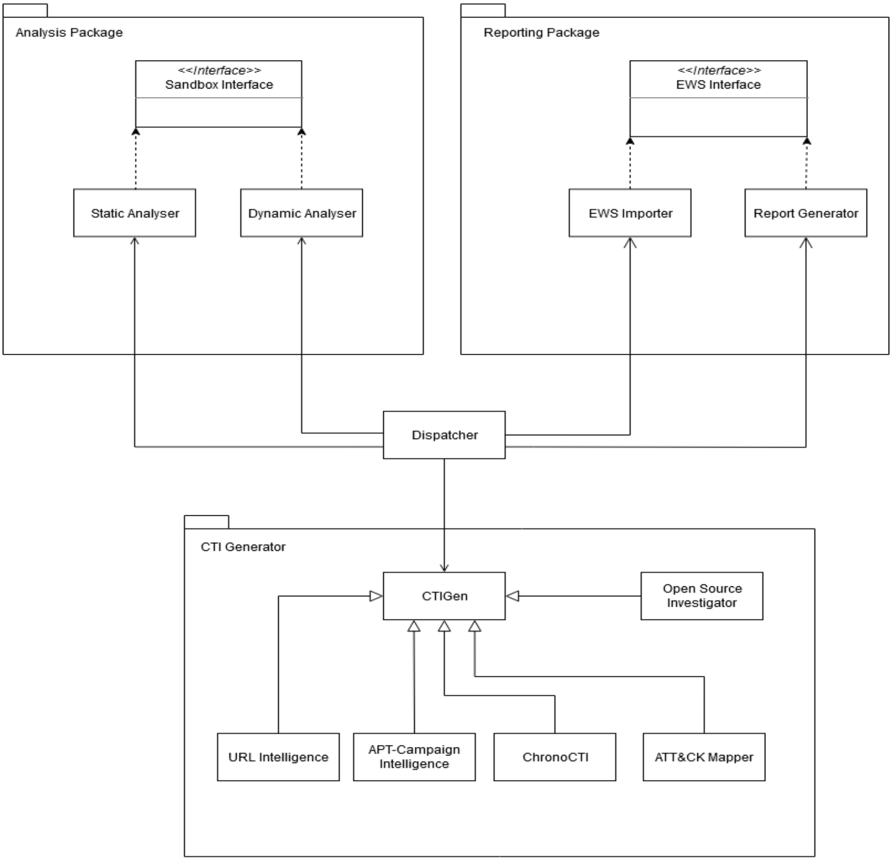

Figure 1: Class Diagram for the MAIT back-end .

Malware analysis

- Cuckoo Sandbox: The main malware analysis technology used for MAIT is Cuckoo Sandbox. The open-source tool's API has been used for uploading a malware sample to the sandbox and retrieving static and dynamic analysis results automatically. Cuckoo Sandbox has also been used to generate process memory dumps of the environment, so that domain and URL IOCs can be extracted. Furthermore, Cuckoo's Volatility plugin has been enabled to extract IP address IOCs from the sandbox's process memory.
- Radare2: Radare2 [16] a reverse engineering and disassembly tool, is used in MAIT to extract static analysis artefacts from the malware.
- Regular Expressions: Regular expressions that match URL, domain, and IP address signatures have been implemented to extract the specified types of IOC from text strings and process memory dumps.

Cyber Threat Intelligence APIs

- VirusTotal API: The VirusTotal API is used for collecting CTI about URL, domain, and IP address IOCs extracted from a malware sample, such as WHOIS information and SSL certificates. It is also used as part of the chronological intelligence of a malware sample by determining when a malware sample was analysed on VirusTotal.
- AlienVault Open Threat Exchange (OTX) Direct Connect API: The AlienVault OTX API[17] is used to retrieve related APT campaigns for the uploaded malware sample, in addition to related TTPs and when the malware samples hash was submitted to AlienVault. It is also used for retrieving related threat groups, threats, and ATT&CK IDs for extracted URL, domain, and IP address IOCs.
- Farsight Security DNSDB API: Farsight Security's DNSDB API[18] is used to retrieve passive DNS CTI for domain and IP address IOCs.
- Security Trails API: The Security Trails API[19] is an alternative source of current DNS records for an extracted domain IOC but does not provide related historical DNS records as part of the free version of the API.
- Threat Intelligence Platform API: The Threat Intelligence Platform API[20] is used to retrieve SSL certificate chain and configuration information for a domain IOC, to provide more context that may help indicate if a domain is malicious.
- IPWhois API: The IPWhois API[21] is used to retrieve geolocation information such as latitude and longitude for an IP address IOC.
- IP2Proxy API: The IP2Proxy API[22] checks whether an IP address IOC is a known proxy, VPN, or Tor exit node, which indicates whether an IP address's geolocation is accurate.
- GreyNoise API: The GreyNoise API[23] provides indications on whether an IP address is associated with malicious activity, as well as indicating if an IP address is benign, but may have been flagged as malicious due to seemingly malicious activity.
- Malware Bazaar API: The Malware Bazaar API[24] is used to retrieve associated APT campaigns for a malware sample in addition to determining when the malware sample was first identified and uploaded to the Malware Bazaar database.
- MITRE ATT&CK: The MITRE ATT&CK Navigator is used to determine the TTPs of a malware sample, which is calculated from the analysis results of the malware sample.

Software Support

- Python 3: Python 3 was chosen to implement the back-end features of MAIT, partially due to its extensive collection of libraries, in addition to other chosen technologies such as Cuckoo Sandbox being written in Python. The Python Flask library has been used to implement an API, to ensure efficient interaction with the web UI.

- HTML, CSS, and JavaScript: Both HTML and CSS have been used to define the structure and style of the MAIT web user interface (UI). Also, JavaScript has been used to retrieve and interact with the back-end functionality for MAIT.
- MongoDB: MongoDB is used to store the malware analysis reports.

Data representation

- JSON: To represent the analysis report in a structured and easily understandable format, dynamic analysis, CTI collection, and URL/IP intelligence results have been formatted in JSON.

Demonstration Cases

Demo Case 1: URL and IP Extraction

To evaluate the effectiveness of the URL and IP address IOC extraction capabilities of MAIT using static and dynamic analysis, domain, URL, and IP address IOCs were extracted from 4 different malware samples using MAIT and two other current solutions for automated malware analysis. Each IOC was then searched for in AlienVault OTX to prove verification, although this does not fully prove that a potential IOC is legitimate but provides an indication of the number of extracted IOCs that are known.

The following table shows details of each of the five malware samples used for the experiment, including their SHA-256 hash.

Table 3. Evaluated samples for URL and IP Extraction.

#	Name	Description	SHA-256 Hash
1	Remcos RAT	Remote access trojan for Windows computers	492823289d0cbc07c789546fda1d7bbee05327c29964f5738f70e82ae7c4f4ad
2	Netwire RAT	Remote access trojan for multiple platforms	364e721eeab968e3a203fbdd6e156d6884469471356f7ab19450142a0ea4cd67
3	Dridex	Trojan for stealing banking credentials	6b827f03d297775876210966a4f6fcd80fadeb4da4417be4d879489104478805
4	Lokibot	Information and credential stealer	51d2bd93ffe8e6856d5c99512b2eb5ed1aa8e1ea871f8c59512080a0329fcf7e
5	Formbook	Data stealer and form grabber trojan	0bb5c3d128d7c78eca860ad07e610fa54fd238907bb09ee21783e15d35874fb5

The following table shows the number of the domain, URL, and IP address IOCs extracted from MAIT, VirusTotal, and Any.Run. Duplicate IOCs extracted from both static and dynamic analysis for MAIT are not counted.

Table 4. URL and IP IoC comparisons.

	Evaluation of IOC extraction	Malware Sample				
		1	2	3	4	5
MAIT	Static IoCs	16	15	0	2	5
	Dynamic IoCs	3	15	18	1	5
	Total IoCs	17	15	18	3	5
	Verified IoCs	13	15	1	2	2
Any.Run	Total IoCs	10	2	0	2	1
	Verified IoCs	7	2	0	2	0
VirusTotal	Total IoCs	10	2	0	2	1
	Verified IoCs	7	2	0	2	0
VirusTotal	Total IoCs	11	2	4	2	38
	Verified IoCs	7	1	4	2	16

From the IOC extraction results, it is discernible that MAIT provides similar results for the number of IOCs extracted when compared to VirusTotal and Any.Run. For malware sample 3, no IOCs were extracted using MAIT. However, the static analysis entropy calculation results of the file sections indicated that many of the file sections were encrypted. This shows the importance of dynamic analysis, as 18 IOCs were extracted from dynamic analysis for malware sample 3 using MAIT

Demo Case 2: APT Attribution

The demonstration case for attribution of malware involves automated steps that are taken with the MAIT tool to identify the APT groups that are behind the development and dissemination of a sample set of malware. For this demonstration case, a cybersecurity incident reported for adversary group APT 29 is selected.[25] The reports for this incident include a dissemination of a set of malware samples through a phishing campaign. The samples are run by MAIT tool and reports the following attribution results in Table 5 for their relationships with APT group 29 in this section.

For these samples, MAIT automatically extracts the chronological intelligence defined by the detection and first seen dates from various AV engines available in the publicly available sources. The chronological intelligence results for these three samples are given in Table 6.

In addition to this chronological intelligence, TTPs that are extracted from CTI sources are compared. Extracted TTPs are demonstrated by their IDs in ATT&CK framework and presented in Table 7. This property of MAIT is integrated into

Table 5. APT 29 phishing campaign and related malware samples.

#	SHA-256 Hash	APT Attribution
1	ca66b671a75bbee69a4a4d3000b45d5dc7d3891c7ee5891272ccb2c5aed5746c	Nobelium/APT29
2	6e2069758228e8d69f8c0a82a88ca7433a0a71076c9b1cb0d4646ba8236edf23	Nobelium/APT29
3	749bf48a22ca161d86b6e36e71a6817b478a99d935cd721e8bf3dba716224c84	Nobelium/APT29

Table 6. Number of detections and the first seen dates on various AV engines, collected by MAIT.

First Seen Dates	Malware #1	Malware #2	Malware #3
18/03/2019	1	1	1
12/03/2021	0	2	0
16/04/2021	0	1	0
06/05/2021	1	1	1
26/05/2021	2	0	2
27/05/2021	1	1	1
28/05/2021	0	0	1
29/05/2021	0	1	0
01/06/2021	0	1	0
02/06/2021	1	2	1
03/06/2021	0	1	0
04/06/2021	0	3	0
05/06/2021	1	49	1
07/06/2021	1	0	1
09/06/2021	20	0	20
10/06/2021	35	0	35

the front-end interface. This comparison shows that other than a few techniques, the malware intelligence resulted in similar techniques and, along with the chronological intelligence, shows the attribution of the malware.

Table 7. TTPs collected by MAIT from CTI sources.

Malware #1	Malware #2	Malware #3
T1027 - Obfuscated Files or Information	T1204 - User Execution	T1204 - User Execution
T1036 - Masquerading	T1027 - Obfuscated Files or Information	T1027 - Obfuscated Files or Information
T1055 - Process Injection	T1036 - Masquerading	T1036 - Masquerading
T1055.001 - Dynamic-link Library Injection	T1055 - Process Injection	T1055 - Process Injection
T1070 - Indicator Removal on Host	T1055.001 - Dynamic-link Library Injection	T1055.001 - Dynamic-link Library Injection
T1071 - Application Layer Protocol	T1070 - Indicator Removal on Host	T1070 - Indicator Removal on Host
T1071.001 - Web Protocols	T1071 - Application Layer Protocol	T1071 - Application Layer Protocol
T1105 - Ingress Tool Transfer	T1071.001 - Web Protocols	T1071.001 - Web Protocols
T1140 - Deobfuscate/Decode Files or Information	T1105 - Ingress Tool Transfer	T1105 - Ingress Tool Transfer
T1195 - Supply Chain Compromise	T1140 - Deobfuscate/Decode Files or Information	T1140 - Deobfuscate/Decode Files or Information
T1199 - Trusted Relationship	T1195 - Supply Chain Compromise	T1195 - Supply Chain Compromise
T1204 - User Execution	T1199 - Trusted Relationship	T1199 - Trusted Relationship
T1204.001 - Malicious Link	T1204.001 - Malicious Link	T1204.001 - Malicious Link
T1547 - Boot or Logon Autostart Execution	T1547 - Boot or Logon Autostart Execution	T1547 - Boot or Logon Autostart Execution
T1566 - Phishing	T1566 – Phishing	T1566 - Phishing
T1566.002 - Spearphishing Link	T1566.002 - Spearphishing Link	T1566.002 - Spearphishing Link
T1573 - Encrypted Channel	T1573 - Encrypted Channel	T1573 - Encrypted Channel

T1574 - Hijack Execution Flow	T1574 - Hijack Execution Flow	T1574 - Hijack Execution Flow
T1598 - Phishing for Information	T1598 - Phishing for Information	T1598 - Phishing for Information
T1610 - Deploy Container	T1610 - Deploy Container	T1610 - Deploy Container

Conclusion & Future Work

In this paper, a demonstration of an automated malware analysis and intelligence collection tool is presented. To the best of our knowledge, MAIT differentiates from the state of the art by fusing the malware analysis with a novel way of enriching the analysis with a collection of publicly available CTI produced for the malware under investigation.

As an analysis and intelligence tool, MAIT contributes to this field by producing chronological information, TTP intelligence, taking an iterative step on intelligence collection and enriching the intelligence with URL and IP analysis. Another significant contribution of MAIT is that it tries to tackle the problem of APT attribution using specific analysis steps and use cases implemented in its development. These novel contributions of MAIT are presented and demonstrated with the demo cases in this paper.

However, there are some limitations to this tool, one being its' dependency on publicly available CTI sources. However, as a standalone tool, without being connected to any CTI sources, it constitutes a significant source of useful information composed from static and dynamic analysis. Another important limitation of this tool is the overwhelming amount of information it produced and the tool's effort of filtering these to compile an analysis accommodating analyst's needs. For evaluating and identifying the performance issues, future work for this research includes rigorous performance testing of the tool.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement no 830943 (ECHO).

References

[1] Nikolaos Pitropakis, Emmanouil Panaousis, Alkiviadis Giannakoulias, George Kalpakis, Rodrigo Diaz Rodriguez, and Panayiotis Sarigiannidis, "An Enhanced Cyber Attack Attribution Framework," *International Conference TrustBus 2018: Trust, Privacy and Security in Digital Business,* 27 July 2018, pp. 213-228, https://doi.org/10.1007/978-3-319-98385-1_15.

[2] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park, “Acomprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions,” *The Journal of Supercomputing* 75 (2019): 4543–4574, https://doi.org/10.1007/s11227-016-1850-4.

[3] Leandros Maglaras, Mohamed Amine Ferrag, Abdelouahid Derhab, Mithun Mukherjee, Helge Janicke, and Stylianos Rallis, “Threats, Countermeasures and Attribution of Cyber Attacks on Critical Infrastructures,” *EAI Endorsed Transactions on Security and Safety* 5, no. 16 (2018), https://doi.org/10.4108/eai.15-10-2018.155856.

[4] Coen Boot, “Applying Supervised Learning on Malware Authorship Attribution,” Master thesis, Radboud University, Institute for Computing and Information Sciences, 2019, https://cs.ru.nl/~aserban/theses/b_coen_boot.pdf.

[5] Thomas Rid and Ben Buchanan, “Attributing Cyber Attacks,” J*ournal of Strategic Studies* 38 no. 1-2 (2015): 4–37, https://doi.org/10.1080/01402390.2014.977382.

[6] “Project Summary,” European network of Cybersecurity centres and competence Hub for innovation and Operations (ECHO), 2021, https://echonetwork.eu/project-summary/.

[7] Craig Miles, Arun Lakhotia, Charles LeDoux, Aaron Newsom, and Vivek Notani, “VirusBattle: State-of-the-art malware analysis for better cyber threat intelligence,” *7th International Symposium on Resilient Control Systems (ISRCS), Denver, CO, USA,* 19-21 Aug. 2014, https://doi.org/10.1109/ISRCS.2014.6900103.

[8] Sagar Samtani, Kory Chinn, Cathy Larson, and Hsinchun Chen, “AZSecure Hacker Assets Portal: Cyber threat intelligence and malware analysis,” *IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA*, 28-30 Sept. 2016, pp. 19–24, https://doi.org/10.1109/ISI.2016.7745437.

[9] Aritran Piplai, Sudip Mittal, Mahmoud Abdelsalam, Maanak Gupta, Anupam Joshi, and Tim Finin, “Knowledge Enrichment by Fusing Representations for Malware Threat Intelligence and Behavior,” *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA*, 9-10 November, 2020, https://doi.org/10.1109/ISI49825.2020.9280512.

[10] Peng Gao, Xiaoyuan Liu, Edward Choi, Bhavna Soman, Chinmaya Mishra, Kate Farris, and Dawn Song, “A System for Automated Open-Source Threat Intelligence Gathering and Management,” *Proceedings of the SIGMOD/PODS’21: 2021 International Conference on Management of Data, China,* June 20-25, 2021, https://doi.org/10.1145/3448016.3452745.

[11] Haoxi Tan, Mahin Chandramohan, Cristina Cifuentes, Guangdong Bai, and Ryan Ko, “ColdPress: An Extensible Malware Analysis Platform for Threat Intelligence,” *arXiv:2103.07012* (2021).

[12] MITRE ATT&CK, 2021, https://attack.mitre.org/.

[13] Any.Run - Interactive Online Malware Sandbox, 2021, https://any.run/.

[14] Cuckoo Sandbox, 2021, https://cuckoosandbox.org/.

[15] VirusTotal, 2021, https://www.virustotal.com/.

[16] Radare2, 2021, https://rada.re/n/.

[17] AlienVault - OTX Direct Connect API, 2021, https://otx.alienvault.com/.

[18] Farsight Security - DNSDB Community Edition, 2021, https://www.farsight security.com/dnsdb-community-edition/.

[19] SecurityTrails API, 2021, https://securitytrails.com/.

[20] Threat Intelligence APIs Platform, 2021, https://threatintelligenceplatform.com/threat-intelligence-api.

[21] IPWhois - IP Geolocation API, 2021, https://ipwhois.io/.

[22] IP2Proxy™ Proxy Detection Web Service, 2021, https://www.ip2location.com/web-service/ip2proxy.

[23] GreyNoise - Community API, 2021, https://docs.greynoise.io/docs/using-the-greynoise-community-api.

[24] Malware Bazaar API, 2021, https://bazaar.abuse.ch/api/.

[25] Damien Cash, Josh Grunzweig, Matthew Meltzer, Sean Koessel, Steven Adair, and Thomas Lancaster, "Suspected APT29 Operation Launches Election Fraud Themed Phishing Campaigns," *Volexity, Blog*, 2021, https://www.volexity.com/blog/2021/05/27/suspected-apt29-operation-launches-election-fraud-themed-phishing-campaigns/.

About the Authors

Cagatay Yucel (PhD) is a Post-Doctoral Researcher and a Threat Intelligence Analyst at the Cyber Security Research Group and BU-CERT of Bournemouth University. Dr Yucel earned his PhD in 2019 with the thesis of "Imaging and Evaluating Memory Access for Malware". Dr Yucel is currently working on EU funded project, "European Network of Cybersecurity Centres and Competence Hub for Innovation and Operations" (ECHO), funded by the European Union's Horizon 2020 research and innovation programme. He has delivered lectures in computer forensics and cyber security workshops internationally on Reverse Engineering, Malware Analysis and Computer Forensics.

Adam Lockett (BSc) is a Postgraduate Student at Bournemouth University. He recently graduated from Bournemouth University with a degree in Forensic Computing and Security and is now studying for a Masters degree in Cyber Security and Human Factors. Adam has contributed to the development of MAIT (Malware Analysis Intelligence Tool) for his Bachelor's dissertation and has interests in malware analysis and penetration testing.

Ioannis Chalkias (MEng, MSc) is a Threat Intelligence Analyst and a Research Assistant at Bournemouth University. He has been involved in many Bournemouth University projects, one of them being the recently completed project funded by EUIPO on illegal IPTV. He is also teaching in forensic and cyber security labs. His fields of interest are digital forensics, cyber threat intelligence, Internet of Things, Information and Cyber Security.

Dimitris Mallis (MEng, MSc) is a Threat Intelligence Analyst and a Research Assistant at Bournemouth University. He has been involved in a number of Bournemouth University projects in the Cyber Security Research Group. His fields of interest are digital forensics, cyber threat intelligence, Internet of Things, Information and Cyber Security.

Vasilis Katos (PhD) is Professor of Cybersecurity at Bournemouth University. Prof. Katos is a member of the Cyber Security Research Group and a certified Computer Hacking Forensic Investigator (CHFI). He leads the BU-CERT (cert.bournemouth.ac.uk) and has worked in the industry as Information Security Consultant and served as an expert witness in Information Security for a criminal court in the UK and a misdemeanor court in Greece on a case related to illegal card sharing, by conducting OSINT and Dreambox forensics. Vasilis led the development of the technical aspects of the recently completed project funded by EUIPO on illegal IPTV. He has also been involved in delivering digital forensic workshops for LE staff. https://orcid.org/0000-0001-6132-3004