WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

"Enhancing Cybersecurity through Advanced Digital Forensics for Early Attack Detection"

Ajithkumar E[1], Dr. Chitra[2]

[1]MSc computer science Rathinam college of arts and science Coimbatore , ajithranj001@gmail.com
[2] Senior Faculty Department of Computer Science Rathinam College of Art and Science, India

ABSTRACT :

Conventional methods for detecting attacks rely on pre-established databases containing known signatures of tools and malicious activities from past cyber-attacks. Some more advanced techniques employ machine learning to identify abnormal behavior. However, the increasing number of successful attacks and the evolving tactics of attackers highlight the inadequacy of these methods. This paper presents a novel approach for early detection of cyber-attacks based on digital forensics, known as Forensics. Forensics integrates ontological reasoning with the MITRE ATT&CK framework, the Cyber Kill Chain model, and continuous acquisition of digital artifacts from monitored computer systems. By employing rule-based reasoning on the Forensics cyber-attack detection ontology, Forensics analyzes collected digital artifacts to uncover signs of adversarial techniques. These techniques are then linked to tactics, which are further associated with phases of the Cyber Kill Chain model, enabling the identification of ongoing cyber-attacks. The effectiveness of this approach is illustrated through a demonstration involving an email phishing attack scenario.

Keywords: Cyber-Attack Detection, Cyber Kill Chain, Cybersecurity, Digital Artifacts, MITRE ATT&CK, Ontology, Rule-based Reasoning

INTRODUCTION :

Cyber-attack mitigation stands as a pivotal element in organizational strategy, essential for the attainment of business objectives. Rather than merely a best practice, it is a business-driven imperative. Detection, one of the quintessential cybersecurity functions, as delineated by the Cybersecurity Framework of the National Institute of Standards and Technology (NIST), encompasses pre-attack threat assessment, real-time attack interception, and post-attack compromise analysis. Detection methodologies encompass statistics-based anomaly detection, pattern-based analysis, rule-based detection, state-based algorithms, and heuristic-driven approaches, each offering unique mechanisms to discern and respond to cyber threats.

Despite the array of detection methods, the early identification of ongoing cyber-attacks remains a pressing challenge. Mandiant's threat report underscores this challenge, revealing that a significant portion of security incidents go undetected by organizations, with adversaries often remaining undetected within compromised systems for extended periods.

To address these limitations, MITRE introduced the Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework, which delineates attacker techniques and tactics. While many detection approaches leverage ATT&CK to identify techniques, they often fall short in leveraging these insights for effective cyber-attack detection.

Digital artifacts, comprising both volatile and non-volatile data, offer rich insights into system events and user activities, providing a fertile ground for enhanced cyber-attack detection. The proactive application of digital forensics practices further enhances detection capabilities, ensuring the integrity of digital artifacts while minimizing response times during cyber incidents.

Introducing Forensics, a novel approach to early cyber-attack detection, which integrates the Cyber Kill Chain (CKC) model and MITRE ATT&CK with digital artifacts and forensic practices. Forensics maps CKC phases to ATT&CK techniques, leveraging digital artifacts to identify operating techniques and reconstruct ongoing cyber-attacks in real-time. Its rule-based reasoning, grounded in a structured ontology, streamlines detection logic, facilitating the identification of cyber threats.

Key contributions of Forensics include:

1) Mapping CKC phases to ATT&CK techniques, overcoming the CKC model's limitation.
2) Leveraging digital artifacts for enhanced detection accuracy and forensic readiness.

3) Reconnaissance and detection of ongoing cyber-attacks, minimizing response times and costs.
4) Correlating ATT&CK techniques with digital artifacts for comprehensive attack detection.
5) Introducing the notion of "Combinations Of Sequences of CKC Phases (COSPs)" to describe and detect cyber-attacks deviating from the CKC model.

However, Forensics' efficacy is bounded by the CKC model's limitations and the completeness of ATT&CK's techniques.
The paper unfolds by providing background information, detailing the Forensics approach, describing its ontology and rule-based reasoning, demonstrating its application, reviewing related work, and concluding with future directions.

II. BACKGROUND :

This Section presents the concepts necessary for presenting the subsequent Sections and the proposed approach.

A. CYBER KILL CHAIN

The Cyber Kill Chain (CKC), a model conceptualized by Lockheed Martin, serves as an intelligence-driven framework aimed at both protection and detection in cybersecurity endeavors. It delineates the sequential phases through which adversaries navigate to execute an attack and fulfill their nefarious objectives.

The CKC comprises the following key phases:

1) Reconnaissance (R): This initial phase encompasses activities aimed at identifying and selecting potential targets. Examples include gathering information from publicly available sources and conducting port scanning.
2) Weaponization (W): Here, attackers craft seemingly legitimate files (e.g., docx, xls, doc) embedded with malware within their own infrastructure, rendering it indistinguishable to defensive security measures.
3) Delivery (D): The delivery phase involves the dissemination of the aforementioned malicious files to the intended target. Common delivery methods include email attachments and USB devices.
4) Exploitation (E): Upon delivery, the payload embedded within the legitimate-looking file exploits system vulnerabilities, allowing the malware to execute within the target environment.
5) Installation (I): In this phase, the malware payload establishes a permanent foothold within the compromised system, ensuring automatic execution at predefined intervals or events such as system reboots.
6) Command and Control (C2): The payload establishes covert communication channels with the attacker, typically facilitated through mechanisms like DNS queries, enabling the exchange of information and commands.
7) Actions on Objective (A): The final phase sees attackers achieving their goals, which may include data exfiltration or lateral movement within the compromised network to access other systems.

The CKC framework provides valuable insights into the adversary's tactics and enables organizations to adopt proactive defense strategies aligned with each phase of the attack lifecycle. By understanding and mitigating threats at each stage, organizations can bolster their cybersecurity posture and thwart potential breaches effectively.

B. MITRE ATT&CK

MITRE ATT&CK serves as an accessible repository of knowledge encompassing the tactics employed by adversaries to fulfill their objectives. This resource, readily available to the public, serves both offensive and defensive purposes, facilitating activities such as penetration testing and cyber-attack detection. Comprising tactics, techniques, and procedures, MITRE ATT&CK provides a granular insight into adversarial strategies.

Tactics represent the short-term objectives pursued by adversaries during an attack. For instance, the "Initial Access" tactic denotes the initial establishment of a foothold within a network or system. MITRE furnishes comprehensive descriptions for each tactic, elucidating its specific goals.

Techniques, on the other hand, denote the actions undertaken by adversaries to fulfill a particular tactic. For instance, the "Phishing" technique may be employed to achieve the "Initial Access" tactic by delivering malware to the target. MITRE further dissects techniques into more specific sub-techniques to provide a nuanced understanding. Techniques may span multiple tactics, reflecting their versatility in achieving diverse objectives. For instance, the "Scheduled Task/Job" technique contributes to both "Persistence" and "Execution" tactics, enabling malware installation and automated execution at specified intervals.

Procedures encapsulate the practical implementation of techniques, delineating how a technique is executed in practice. For instance, one procedure of the "Spear phishing Attachment" technique entails an adversary sending an email containing a malicious Microsoft Office .doc attachment. MITRE supplements each technique with real-world examples, offering detailed insights into how techniques have been utilized in reported cases.

Through its comprehensive taxonomy, MITRE ATT&CK equips cybersecurity professionals with invaluable insights into adversary tactics, techniques,

and procedures, enabling proactive defense strategies and informed decision-making in the face of evolving cyber threats.

C. DIGITAL FORENSICS

Digital Forensics (DF) represents the application of computer science principles to facilitate the investigation, analysis, and potential prosecution of incidents involving digital data. This spectrum spans from minor policy violations to serious cyber-crimes and felonies. The culmination of a digital forensics endeavor typically yields a detailed report containing evidentiary data pertinent to the incident at hand. These evidentiary data serve as pivotal components in addressing fundamental questions regarding the incident, including the identities involved, the nature of the event, its location, timing, and modus operandi. Essentially, evidentiary data serve as the tangible remnants of an incident, be it a cyber-attack or other digital transgressions. Referred to interchangeably as evidentiary artifacts or evidentiary digital artifacts, these data encompass a broad array of digital entities altered or generated by human action, software processes, or device operations. Examples encompass diverse digital entities such as emails, email attachments, registry keys, word documents, and IP addresses, among others. Moreover, digital artifacts extend to include volatile data streams like running processes. The fidelity and integrity of digital artifacts retrieved from a system hold paramount importance in the investigative process. Preserving their integrity during acquisition ensures their reliability and authenticity, bolstering their value as irrefutable evidence in ongoing investigations.

III. THE PROPOSED APPROACH :

A particular cyber-attack might consist of various combinations of phases since some phases of the CKC model may not be used. For example, there might be a case of delivering malware where there is no need to install it in the targeted system; so, the Installation phase is skipped. One example of such malware is UIWIX [16]. In this vein, Forensics reconstructs an ongoing cyber-attack by detecting one of the following Combinations Of Sequences of CKC Phases (COSPs): 1) Delivery, and Exploitation – COSP(DE): An attacker gains access within a system via delivering malware that exploits a vulnerability. 2) Delivery, Exploitation, and Installation – COSP(DEI): In addition to DE, the malware obtains persistence in the compromised system. 3) Delivery, Exploitation, and C2 – COSP(DEC): In addition to DE, the malware communicates with the attacker for command-and-control purposes. 4) Delivery, Exploitation, Installation, and C2 – COSP(DEIC): In addition to DEC, the malware installs itself in the compromised system Forensics does not consider Reconnaissance and Weaponization since they are preparation phases [17]. According to MITRE, these phases are actions that attackers take before executing an attack or before trying to access a network [5]. MITRE considers the Reconnaissance and Weaponization (MITRE calls it as “Resource Development) phases in the PRE-ATT&CK framework which is focused on recognizing pre-attack actions for prevention purposes [5]. The Actions on Objective phase is also excluded since the detection should occur before the attackers achieve their objective in this phase.

CONCEPTS :

The main concepts of Forensics are: COSP, phases, techniques, tactics, and digital artifacts. The UML diagram in Figure 1 presents the relationships among the forenamed Forensics concepts. More specifically, a digital artifact can be associated with other digital artifacts with the hasRelatedDigitalArtifact relationship. For instance, a file (e.g., docx document) is associated with the process that opened it (e.g., Microsoft Word), or an email message is associated with its attachments. The hasRelatedDigitalArtifact relationship can be specialized to convey more precise semantics as well. For example, a specialized hasAttachedFile relationship can be created when specifying the relationship between an email message and its attached file. A technique can be associated with one or many digital artifacts with the hasTrace relationship. Indeed, the operation of a technique can leave one or more traces in a system. These traces are realized as digital artifacts. To identify them, the description of the technique as provided in the MITRE ATT&CK knowledge base is examined. For instance, the description of the technique “Spearphishing Attachment” mentions that the technique is accomplished via an email message that contains an attached file. This means that the digital artifacts, which are the traces of this technique, include an email message and an attached file. A tactic can be associated with one and only one technique with the hasTechnique relationship because the execution of a particular technique can achieve only one specific tactic at a time. A phase can be associated with a tactic with the mapsTo relationship. Indeed, the Delivery, Exploitation, Installation, and C2 phases are mapped to the Initial Access, Execution, Persistence, Command, and Control tactics, respectively, since they serve the same purpose. Forensics uses these tactics only since its detection logic is based on the CKC model with respect to COSPs. Consequently, a COSP as used in Forensics can be associated with two to four phases with the hasPhase relationship. It should be noted that even though a tactic can be achieved with more than one technique, Forensics creates and relates a new tactic instance for every recognized technique. In this way, a self-contained thread from a COSP to Digital Artifacts can be established such that only relevant traces (i.e., digital artifacts) and phases can be analyzed and decided to whether they constitute a COSP.

METHODOLOGY :

Forensics utilizes the relationships among the concepts described in subsection 3.1 to detect ongoing cyber-attacks. To do so, Forensics follows a proposed multi-step methodology, which is explained below and depicted in Figure 2: Step 1: Preparation steps: The following steps 1.1 to 1.3 are repeated until all digital artifacts are examined. This process is depicted with the “for all digital artifacts” loop in Figure 2. The examination of all digital artifacts ensures the recognition of all the techniques operated within the monitored system. Note that the digital artifacts are acquired from the monitored system and provided beforehand. It is out of the scope of this study to recommend a specific application tool to acquire the digital artifacts from the monitored system.

Step 1.1: Technique Recognition. The operation of a technique is recognized based on the digital artifacts that it creates in the monitored system during

its operation. As a result, an instance of the technique is created. The hasTrace relationship is used to associate the new Technique instance with the digital artifacts used to recognize the technique

Step 1.2: Tactic Association. This step associates a new Tactic instance to the Technique instance from Step 1 using the hasTechnique relationship. Because a technique may be used in several tactics, multiple Tactic instances and relationships may be created.

Step 1.3: Phase Mapping. The tactics identified in Step 2 are mapped to new Phase instances. The mapsTo relationship is used to associate each new Phase instance with a Tactic instance of Step 1.2. • Step 2: Ongoing Cyber-Attack Detection. Steps 1.1 to 1.3 result in chains of recognized instances (CORI). Each CORI consists of a chain of one Phase instance, one Tactic instance, one Technique instance, and a set of Digital Artifact instances linked according to the discovery in Step 1. So, the traces of a Technique instance are the traces of a Phase instance. As detailed in the next subsection 3.3, in this Step 2, Forensics utilizes CORIs to form a COSP. In essence, each formed COSP is the reconstruction and detection of an ongoing cyber-attack.

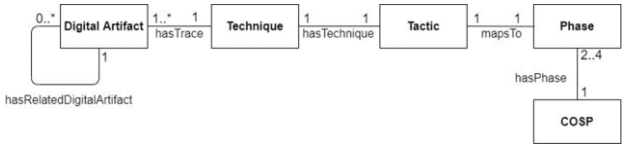

FIGURE 1. UML diagram of Forensics concepts.

ONGOING CYBER-ATTACK DETECTION :

In Step 2 of the methodology outlined in subsection 3.2, Figure 3 offers an analytical perspective. Forensics harnesses CORIs to fashion a COSP, a pivotal step in reconstructing and identifying ongoing cyber-attacks. For a COSP to be valid, it must adhere to three fundamental conditions:

1) The sequence and arrangement of Phase instances should align with one of the predefined COSPs delineated in subsection 3.1. For example, a COSP(DE) must encompass Delivery and Exploitation Phase instances, with Delivery preceding Exploitation as per the CKC model.

2) Every pair of adjacent Phase instances must share correlations; that is, their traces should exhibit attributes with either commonalities or temporal relationships. Attributes like names, full paths, and timestamps serve as indicators of correlation. For instance, if the attachment downloaded during a Delivery Phase instance shares the same full path as the file opened by a process during an Exploitation Phase instance, they are considered correlated. Timestamp attributes are deemed temporally-related if they fall within a defined timeframe.

3) Each pair of adjacent Phase instances must be consecutive in their occurrence. In other words, the correlated traces of a Phase instance should be chronologically newer than those of its preceding Phase instance. For instance, if the attachment was downloaded at time t1 and the file was opened by a process at time t2, the Delivery and Exploitation Phase instances are consecutive if the time difference (t2 - t1) is greater than or equal to zero.

IV. IMPLEMENTATION :

Forensics lends itself to various implementation methods, including program coding, machine learning, and rule-based logic. In this study, Forensics was instantiated through the instantiation of a proposed ontology and rule-based logic. The Forensics ontology delineates the core

principles of Forensics alongside their attributes and interconnections. This ontology was crafted using the Web Ontology Language (OWL) [18]. The methodology of Forensics, as expounded in subsection 3.3 for ongoing cyber-attack detection, is enacted through rule-based logic. These declarative regulations are scripted using the Semantic Web Rule Language (SWRL) [19]. The integration of rule-based logic is facilitated by the Drools rule engine [20]. The Protégé [21] ontology development environment served as the platform for implementation and experimentation.

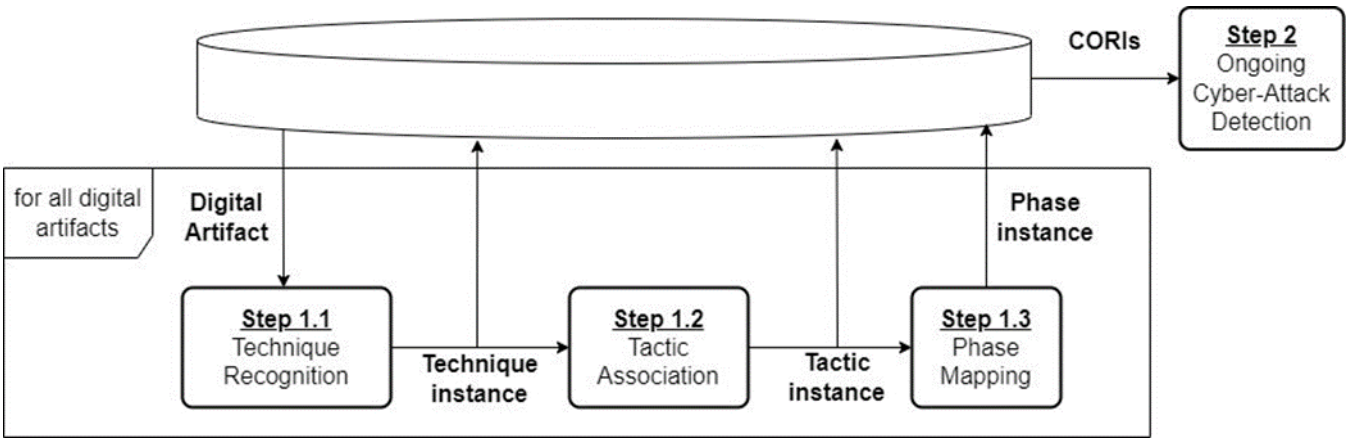

FIGURE 2. The proposed step-by-step methodology

1) Classes

The foundational classes within the proposed ontology are delineated below, each serving as a distinct entity in the schema:

Phase Class: This class encompasses subclasses such as Delivery, Exploitation, Installation, and CommandAndControl, each representing distinct phases within the cyber-attack lifecycle.

Tactic Class : Within this class, subclasses like MA_InitialAccess, MA_Execution, MA_Persistence, and MA_CommandAndControl delineate various tactics employed during cyber-attacks.

Technique Class: Subclasses within this category encapsulate diverse techniques associated with the aforementioned tactic subclasses, providing a granular understanding of attack methodologies.

Artifact Class: Representing a spectrum of digital artifacts such as WindowsTask, File, and EmailMessage, these subclasses are rooted in the Unified Cyber Ontology (UCO) 0.7.0 Release. UCO serves as a community-driven ontology for cyber security, encompassing representations of digital artifacts alongside their inherent data properties.

COSP Class: This class signifies the reconstruction of ongoing cyber-attacks based on identified COSPs. Each COSP instance, as an individual within the COSP class, is intricately linked with corresponding CORIs that collectively formulate the COSP structure.

2) Data Properties

Attributes, or data properties, establish a connection between an instance of a class and a specific value, which could be a string, integer, or another data type. Each attribute is defined by its domain and range. The domain specifies which classes are eligible to possess the attribute, while the range determines the allowed data types or values for the attribute.

In the context of Artifact subclasses, data properties are meticulously defined. These properties are bound to a specific Artifact subclass, with their ranges contingent upon the data type they describe, such as string or integer.

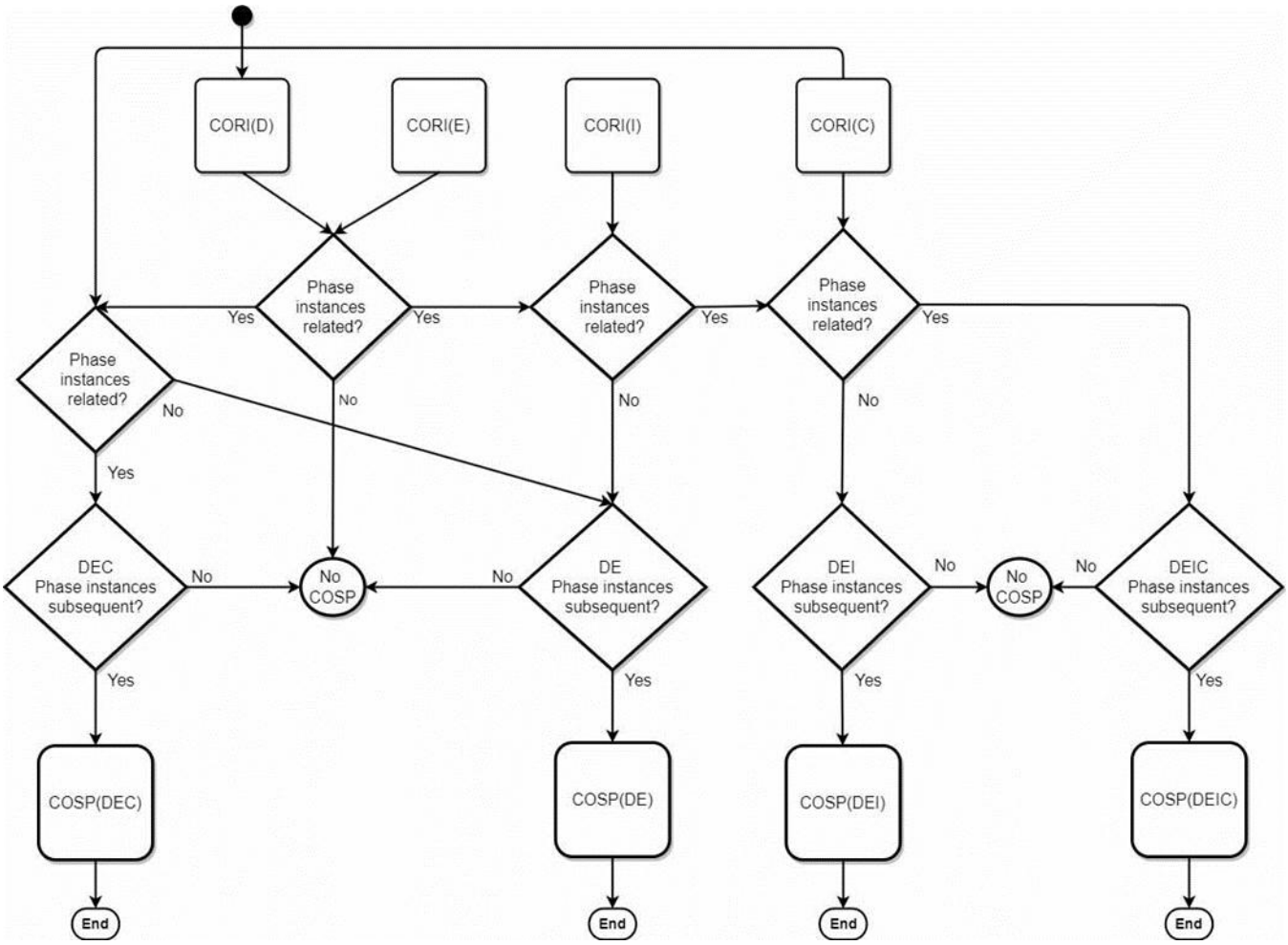

FIGURE 3. Step 2 of the methodology: Ongoing Cyber-Attack Detection

While many data properties within Artifact subclasses are aligned with the Unified Cyber Ontology (UCO), additional properties were introduced to address gaps identified in UCO. For instance, attributes like hasFullPath, hasFileName, and hasExtension were established to supplement missing elements in UCO. These new properties were devised based on the intrinsic attributes of digital artifacts.

Of notable importance are timestamps, including createdTime, modifiedTime, accessedTime, and deletedTime, which denote crucial temporal aspects of Artifact instances. These timestamps play a pivotal role in determining the correlation and subsequent nature of Phase instances within a COSP.

3) Object Properties

Object properties facilitate the establishment of connections between two entities, both of which are instances of classes within the ontology. One entity serves as the domain value of the object property, while the other entity acts as the range value. These object properties, also referred to as predicates, encapsulate relationships between subjects and objects within the ontology.

The ensuing object properties delineate relationships between COSP, Phase, Tactic, Technique, and Artifact classes, with these relationships being inherited by their respective subclasses. They serve as the fundamental constructs for defining associations and interactions between various entities within the ontology, enabling the representation of intricate relationships and dependencies among different elements.

- The property named hasPhase links entities from the COSP domain to entities in the Phase domain. Object constraints were established to ensure that each COSP entity is connected to a minimum of two and a maximum of four Phase entities.

- The relationship denoted as mapsTo connects entities from the Phase domain to entities in the Tactic domain. A strict cardinality restriction of 1 guarantees that each Phase entity is linked to precisely one Tactic entity.
- Under the hasTechnique property, entities from the Tactic domain are linked to entities in the Technique domain. A precise cardinality restriction of 1 mandates that each Tactic entity is associated with only one Technique entity.
- The relationship named hasTrace connects entities from the Technique domain to entities in the Artifact domain. A minimum cardinality restriction of 1 ensures that each Technique entity is linked with at least one Artifact entity.

Furthermore, object properties categorized as subproperties of hasRelatedDigitalArtifact were established to illustrate specific connections among the Artifact subclasses elaborated in subsection 3.1. The subsequent steps outline the process of defining these properties for all techniques:

1) Choosing a technique from MITRE ATT&CK, which should be categorized under one of the Initial Access, Execution, Persistence, or Command and Control tactics. For example, the technique "Spearphishing Attachment" from the Initial Access tactic is chosen.

2) Creating Artifact subclasses. The method involves scrutinizing the description of the chosen technique to identify the digital traces it leaves behind in a system. These traces manifest as digital artifacts, represented by individuals of Artifact subclasses. For instance, the description of the "Spearphishing Attachment" technique indicates that it is executed through an email message containing an attached file. Consequently, the digital artifacts associated with this technique include an email message and an attached file, represented respectively by the EmailMessage and File subclasses of the Artifact class.

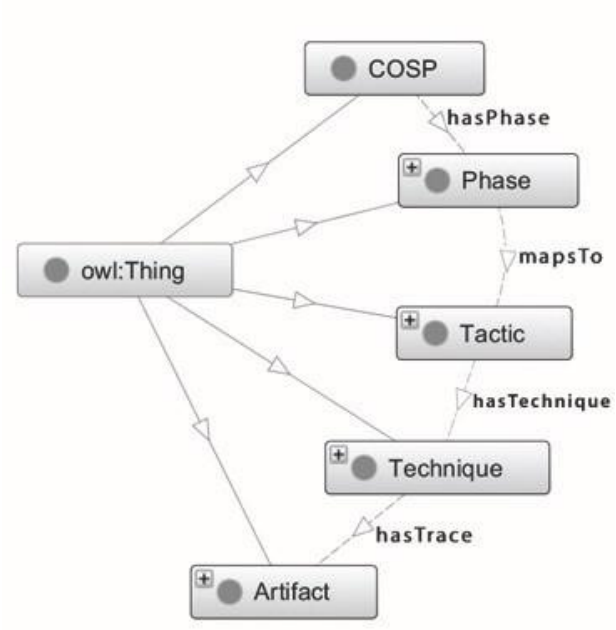

FIGURE 4. OntoGraf rendering of the proposed ontology.

3) Developing object properties. Guided by the description of the selected technique, the connections between the identified Artifact subclasses from the previous step are determined. These connections serve as the basis for defining object properties. In the aforementioned example, the email message might feature an attached file. Thus, the object property hasAttachedFile is defined, with its domain being the EmailMessage subclass and its range being the File subclass.

Figure 4 illustrates the five principal classes within the proposed ontology along with their interrelations. To streamline the presentation, the subclasses and their inherited relationships are not included.

Ensuring consistency is an integral part of ontology development to verify its validity. This process detects and resolves any logical inconsistencies or undesirable deductions across the classes, properties, and their associated domains and ranges. Consistency checking was carried out using the Pellet OWL reasoner plugin for Protege[23].

4) Declarative Rules

Rules were formulated to identify CORIs based on artifact assertions and establish COSPs indicative of ongoing cyber-attacks. While all individuals, except for those belonging to Artifact subclasses, are generated by rules, individuals of Artifact subclasses and their properties are asserted from digital artifacts acquired from the monitored system. However, recommending an application tool for this purpose is beyond the scope of this discussion.

The rules are an embodiment of the Forensics methodology expounded in Section 3. Each rule comprises a condition (i.e., antecedent) and an action (i.e., consequent). Upon satisfaction of the condition, the action is executed, resulting in the creation of a new individual and the assertion of properties. For instance, rules implementing Step 1.2 of the Forensics methodology match a Technique subclass individual in the condition part and generate a Tactic subclass individual while asserting a hasTechnique relationship to the Technique subclass individual in the action part. These rules are classified into four groups:

1) Rules for identifying techniques: Each rule identifies a specific technique from particular digital artifacts, as outlined in Step 1.1 of subsection 3.2. Upon detecting the requisite Artifact subclass individuals and their relationships, a new Technique subclass individual is asserted along with the has Trace object property to the Artifact subclass individuals activated by the rule.
2) Rules for associating tactics: Individual rules are defined for each technique, where the presence of a Technique subclass individual triggers the instantiation of the corresponding Tactic subclass individual through the hasTechnique object property.
3) Rules for mapping phases: A rule is established for each phase type, verifying the existence of a specific Tactic subclass individual based on the mapping between phases and tactics. Subsequently, an individual of the relevant Phase subclass is asserted, along with a mapsTo relationship to the Tactic individual.
4) Rules for detecting ongoing cyber-attacks: Aligned with Step 2 of the Forensics methodology in subsection 3.2, multiple rules are devised for each COSP. The condition part verifies the fulfillment of the three conditions outlined in subsection 3.3, while the action part asserts the corresponding COSP individual along with the hasPhase relationship to the appropriate Phase subclass individuals.

Since the number of rules required for comprehensive coverage is substantial, initiating the detection of ongoing cyber-attacks employing commonly used techniques is recommended. This approach, advocated by MITRE, focuses on detecting frequently employed techniques first. Figure 5 illustrates charts depicting the frequency of techniques utilized in reported incidents. By targeting commonly used techniques, the number of rules needed for detection can be significantly reduced, thereby enhancing efficiency in ongoing cyber-attack detection efforts.

V. EXAMPLE DETECTION :

In this section, we showcase the implementation of Forensics using an email phishing attack scenario as an example. The demonstration illustrates how the Forensics ontology and rule-based reasoning framework can effectively detect email phishing attacks. Email phishing, a prevalent type of social engineering attack, involves the fraudulent crafting and dissemination of emails to deceive recipients into unwittingly facilitating the attacker's objectives. These objectives often include downloading malicious attachments or accessing compromised URLs [26], [27]. The selection of an email phishing attack for demonstration purposes is motivated by its widespread occurrence in business environments, where it ranks among the primary vectors for malware distribution, either through direct attachments or indirect URL links [28], [29].

The email phishing attack scenario spans across the Delivery, Exploitation, and Installation phases, which are correlated with the Initial Access, Execution, and Persistence tactics, respectively, as outlined in subsection 3.2. Each tactic corresponds to a specific technique, resulting in the utilization of three distinct techniques to execute the attack phases. The chosen techniques, selected based on threat intelligence reports and insights from Figure 5, are as follows:

1) Delivery phase technique: According to MITRE ATT&CK, the "Spearphishing Attachment" technique is widely employed in email phishing attacks due to the prevalence of malicious attachments [30], [31], [32]. Hence, an infected Microsoft Office .doc file is chosen as the malicious attachment for this phase.
2) Exploitation phase technique: Drawing from insights presented in Figure 5, the "Malicious File" technique (ID: T1204.002) is selected, primarily associated with achieving the Execution tactic and mapped to the Exploitation phase.
3) Installation phase technique: In this example, the "Scheduled Task" technique (ID: T1053.005) is chosen to represent the Persistence tactic, mapped to the Installation phas

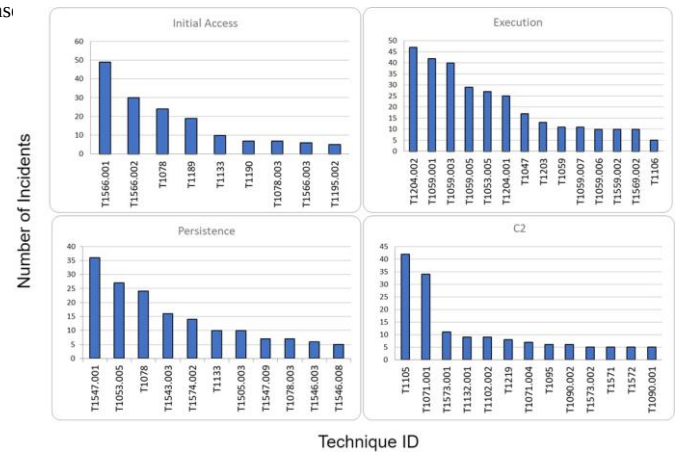

FIGURE 5. Charts of techniques used in incidents reported to MITRE ATT&CK.

TABLE 1. Artifact subclasses along with their axioms used in the application example.

Artifact subclass	Axioms
EmailMessage	It is not necessary to include any axiom for the example.
AttachedFile	hasDownloadedTimestamp max 1 xsd:dateTime
File	hasFullPath max 1 xsd:string
Process	hasStartedTimestamp exactly 1 xsd:dateTime
WindowsTask	hasCreatedTimestamp exactly 1 xsd:dateTime

Based on the aforementioned techniques, the email phishing attack scenario unfolds as follows:

1) Delivery phase: The attacker dispatches an email containing a malicious .doc file attachment to the target, who subsequently opens and downloads the attachment.
2) Exploitation phase: Upon opening the file, a word.exe process executes, launching the attached file.
3) Installation phase: Following the file's opening, a scheduled task is promptly generated.

The implementation of Forensics, as outlined in Section 4, involved the creation of Artifact subclasses along with their corresponding axioms, which are depicted in Table 1. Each axiom adheres to the OWL Manchester Syntax [34] and comprises the data property name, restriction type, and restriction filter. The data property names are deliberately chosen to reflect the attributes of the digital artifacts they represent. For instance, the "hasFullPath" property signifies the fullpath attribute of a file. Restriction types specify the permissible number of values for each data property, such as "max 1" indicating that a File entity can possess a maximum of one fullpath attribute at any given time. The restriction filter denotes the datatype of the data property, with examples including string and dateTime values. To streamline the presentation, only essential details relevant to illustrating the example are provided.

TABLE 2. Object properties between Artifact subclasses used in the example

Object Property Name	Domain Value	Range Value
hasAttachedFile	EmailMessage	AttachedFile
opensFile	Process	File

Table 2 delineates the object properties between Artifact subclasses. Each object property name succinctly conveys the nature of the relationship between two Artifact subclasses. One subclass serves as the domain value of the object property, while the other serves as its range value. For instance, an Email Message Artifact subclass may be linked to an AttachedFile Artifact subclass via the "hasAttachedFile" relationship, denoted as an object property.

As discussed in subsection 4.2, the detection rules are organized into four sets, with this example focusing on illustrating a COSP(DEI)-related rule. While COSP(DEIC)-related rules can be encoded similarly, they are not explicitly illustrated here. In our example scenario, the presence of specific Artifact subclass individuals representing the traces of the email phishing attack is necessary in each phase.

- A single Email Message entity and a corresponding Attached File entity are required. Both entities must have all properties specified in the axioms from Table 1 asserted. Furthermore, the EmailMessage entity should be linked to the AttachedFile entity through the assertion of the hasAttachedFile object property.
- An individual Process entity and a corresponding File entity are necessary. Both entities must have all properties outlined in the axioms from Table 1 asserted. Additionally, the Process entity must be associated with the File entity through the assertion of the opensFile object property.
- A sole WindowsTask entity must be present, with all properties specified in the axioms from Table 1 asserted.

TABLE 3. Rules for recognizing techniques.

Technique	Rule

Spearphishing Attachment	EmailMessage(?em) ∧ AttachedFile(?file) ∧ hasAttachedFile(?em, ?file) ∧ swrlx:makeOWLThing(?new, ?em) → Spearphishing_Attachment(?new) ∧ hasTrace(?new, ?em)
Malicious File	Process(?pr) ∧ File(?file) ∧ opensFile(?pr, ?file) ∧ swrlx:makeOWLThing(?new, ?pr) → Malicious_File(?new) ∧ hasTrace(?new, ?pr)
Scheduled Task	WindowsTask(?wtask)∧ swrlx:makeOWLThing(?new, ?wtask) → Scheduled_Task(?new) ∧ hasTrace(?new, ?wtask)

In addition to individual facts, rule-based reasoning in Forensics involves the application of declarative rules as elaborated in subsection 4.2. The defined rules encompass the following categories:

1) Rules for identifying techniques: Each rule verifies the existence of Artifact subclass individuals representing the traces of specific techniques. Furthermore, the rule examines the relationships among these Artifact subclass individuals. If both conditions are satisfied, the rule generates an individual of the generic OWL Thing class. Subsequently, the action segment of the rule assigns the new individual to the appropriate Technique subclass and asserts the hasTrace relationship with the Artifact subclass individuals. The rules for identifying the three techniques employed in the email phishing attack example are outlined in Table 3. For example, the rule associated with the "Spearphishing Attachment" technique endeavors to ascertain the presence of an EmailMessage Artifact subclass individual linked to an AttachedFile Artifact subclass individual via the hasAttachedFile object property. Upon verification, the condition segment of the

TABLE 4. Rules for associating tactics.

Tactic	Rule
Initial Access	Spearphishing_Attachment(?spa) ∧ swrlx:makeOWLThing(?new, ?spa) → MA_InitialAccess(?new) ∧ hasTechnique(?new, ?spa)
Execution	Malicious_File(?mf) ∧ swrlx:makeOWLThing(?new, ?mf) → MA_Execution(?new) ∧ hasTechnique(?new, ?mf)
Persistence	Scheduled_Task(?stasktechnique) ∧ swrlx:makeOWLThing(?new, ?staskt) → MA_Persistence(?new) ∧ hasTechnique(?new, ?stasktechnique)

TABLE 5. Rules for mapping phases.

CKC Phase	Rule
Delivery	MA_InitialAccess(?ia) ∧ swrlx:makeOWLThing(?new, ?ia) → Delivery(?new) ∧ mapsTo(?new, ?ia)
Exploitation	MA_Execution(?exe) ∧ swrlx:makeOWLThing(?new, ?exe) → Exploitation(?new) ∧ mapsTo(?new, ?exe)
Installation	MA_Persistence(?pers) ∧ swrlx:makeOWLThing(?new, ?pers) → Installation(?new) ∧ mapsTo(?new, ?pers)

rule instantiates an individual of the generic OWL Thing class. Subsequently, the action segment of the rule assigns the aforementioned new individual to the Spearphishing_Attachment class and establishes the hasTrace relationship with the EmailMessage Artifact subclass individual. The AttachedFile Artifact subclass individual does not necessarily need to be linked directly to the Spearphishing_Attachment individual, as it is already connected to the EmailMessage Artifact subclass individual via the hasAttachedFile object property, facilitating its retrieval.

2) Protocols for linking tactics. Each guideline is triggered when a Technique subclass individual is potentially affiliated with a distinct Tactic subclass individual. Upon satisfying this criterion, the guideline generates an instance of the generic OWL Thing class. Subsequently, the procedural segment of the guideline designates the newly created instance to the relevant Tactic subclass and affirms the has Technique connection to the Technique subclass individual. The details of these protocols are presented in Table 4. To illustrate, the guideline pertaining to the Initial Access tactic

is set in motion when a Spearphishing_Attachment individual is identified. Subsequently, the guideline's conditional component generates a novel instance of the OWL Thing class. Ultimately, the operational part of the guideline assigns the aforementioned instance to the MA_InitialAccess class and confirms the has Technique relationship with the Spear phishing _Attachment individual.

3) Procedures for aligning phases. Each procedure becomes active upon identification of a Tactic subclass individual that corresponds to a specific Phase subclass individual. Upon meeting this criterion, the procedure generates an individual within the generic OWL Thing class. Subsequently, the procedural segment of the rule allocates the new individual to the appropriate Phase subclass and affirms the maps To relation to the corresponding Tactic subclass individual. Table 5 delineates the procedures for aligning Phase to Tactic subclass individuals. To illustrate, the procedure associated with the Delivery phase initiates upon the presence of an MA_InitialAccess individual. Subsequently, the procedure generates a new individual within the OWL Thing class. Finally, the procedural segment assigns this new entity to the Delivery category and confirms the maps To relationship with the MA InitialAccess individual.

The final rule required in this instance aims to identify and establish a COSP(DEI), signifying an ongoing cyber-attack. This rule must ascertain the correlation and subsequent nature of the Phase individuals. In our case, this entails examining the DE and EI pairs. Two Phase subclass individuals are deemed correlated if their traces exhibit attributes with shared values. Table 6 delineates the DEI rule into nine segments, which collectively form the overarching COSP(DEI) rule as "Part1 ∧ Part2 ∧ Part3 ∧ Part4 ∧ Part5 ∧ Part6 ∧ Part7 ∧ Part8 → Part9".

Parts 1, 2, and 5 correspond to a CORI match, specifically identifying a Phase subclass individual, progressing to the mapped Tactic subclass individual, and identifying the corresponding Technique individual. Subsequently, they match the Artifact subclass individuals, which represent the traces of the Technique individual and consequently, the Phase subclass individual. Finally, they match the properties of the Artifact subclass individuals, which are then utilized in Parts 3 and 6 to determine if they share common or temporally-related values. When Parts 3 and 6 are validated, the Delivery, Exploitation, and Installation individuals are deemed correlated.

class Thing for each detected Delivery individual. Finally, Part 9 serves as the action segment, assigning the newly named individual to the COSP class and affirming the hasPhase relation with the Delivery, Exploitation, and Installation individuals.

Figure 6 illustrates the outcome of all declarative rules executed during the example detection. The COSP individual, depicted at the top of Figure 6 as COSP_1_DEI, incorporates three Phase individuals, each mapped to a Tactic individual, and subsequently, a Technique individual. In the final rows of boxes, the Artifact subclass individuals are portrayed, representing the traces of the Phase individuals and consequently, the detected ongoing cyber-attack. While the OntoGraf tool, on which Figure 6 is based, does not explicitly indicate that File1 and File2 are identical, their detection of the spearphishing attack suggests that they share the same properties and represent the same digital artifact. Notably, the sequence of Phase individuals adheres to chronological order from left to right, denoting their subsequent nature based on the timestamps of their traces.

Performance evaluations were conducted on the Forensics implementation based on the example detection. The Drools rule-engine demonstrated linear time in detecting the email phishing attack, as depicted in Figure 7. Despite processing over 200,000 individuals and 16 rules successfully within 13 minutes, the evaluation was performed on a mid-level personal computer equipped with 48GB RAM and an Intel Core i7-10850H Processor. This timeframe allows for early detection even in scenarios where digital artifacts are frequently restored, considering that cyber-attacks typically go unnoticed for at least 24 days.

TABLE 6. Declarative rule for detecting the email phishing attack of the application example.

Part	Rule in the Semantic Web Rule Language
1	Delivery(?del) ∧ MA_InitialAccess(?mainit) ∧ mapsTo(?del, ?mainit) ∧ Spearphishing_Attachment(?spa) ∧ hasTechnique(?mainit, ?spa) ∧ EmailMessage(?email) ∧ hasTrace(?spa, ?email) ∧ AttachedFile(?afile) ∧ hasAttachedFile(?email, ?afile) ∧ hasDownloadedTimestamp(?afile, ?downtime) ∧ hasFullPath(?afile, ?afilefullpath)
2	Exploitation(?exploit) ∧ MA_Execution(?exec)∧ mapsTo(?exploit, ?exec) ∧ Malicious_File(?malfile) ∧ hasTechnique(?exec, ?malfile) ∧ Process(?process) ∧ hasTrace(?malfile, ?process) ∧ File(?openedfile) ∧ opensFile(?process, ?openedfile) ∧ hasStartedTimestamp(?process, ?prtime) ∧ hasFullPath(?openedfile, ?openedfullpath)
3	swrlb:equal(?afilefullpath, ?openedfullpath)
4	temporal:before(?downtime, ?prtime)
5	Installation(?instal) ∧ MA_Persistence(?persist) ∧ mapsTo(?instal, ?persist) ∧ Scheduled_Task(?stasktechnique) ∧ hasTechnique(?persist, ?stasktechnique) ∧ WindowsTask(?wtask) ∧ hasTrace(?stasktechnique, ?wtask) ∧ hasCreatedTimestamp(?wtask, ?wtasktime)

6	temporal:duration(?duration, ?wtasktime, ?prtime, "Minutes") ∧ swrlb:lessThanOrEqual(?duration, 1)
7	temporal:before(?prtime, ?wtasktime)
8	swrlx:makeOWLThing(?new, ?del)
9	→ COSP(?new) ∧ hasPhase(?new, ?del) ∧ hasPhase(?new, ?exploit) ∧ hasPhase(?new, ?instal)

VI. RELATED WORK :

Existing literature related to Forensics comprises rule-based approaches that leverage MITRE ATT&CK or the CKC model for detection purposes, with most of these works listed in MITRE's directory [6]. Notably, Sigma and CAR predominantly employ rules to detect techniques operating within a system. However, unlike Forensics, they do not emphasize detecting sequences of techniques indicative of cyber-attacks. Forensics, in contrast, identifies technique operations within a system and leverages them for

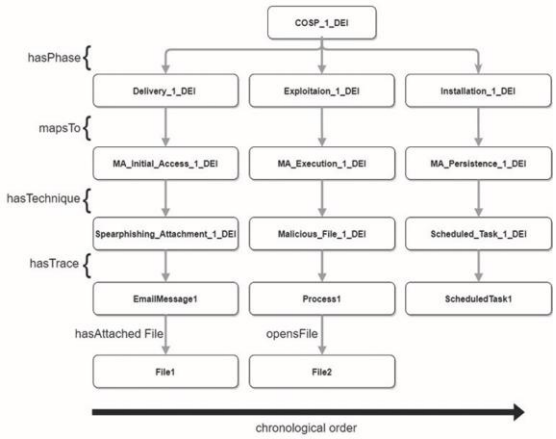

FIGURE 6. Visualization of a Spearphishing attack detection based on OntoGraf [35].

detecting ongoing cyber-attacks based on the CKC model. Notably, CAR does not account for techniques under the Initial Access tactic, crucial for detecting initial footholds within a system. Forensics, on the other hand, initiates detection from this tactic, crucial for early detection aligning with the NIST CyberSecurity Framework's Detect function.

Approaches in [6] primarily employ non-declarative rules, particularly CAR's rules, which are in pseudocode to illustrate their implementation. In contrast, Forensics adopts OWL and SWRL rules, executable and checkable for inconsistencies. SWRL rules can be shared among organizations, fostering collaborative development of detection capabilities concisely and unambiguously. Efforts based on Sigma, such as Atomic Threat Coverage [37], examine log files in decision logic, while Forensics

Criteria	Sigma	CAR	[38]	[39]	Forensics
Individual technique detection	Yes	Yes	No	No	Yes
Sequence of techniques detection	No	No	No	No	Yes
Cyber-attack reconstruction	No	No	Yes	Yes	Yes
Cyber-attack detection	No	No	No	No	Yes
Concise, unambiguous rules	No	No	Yes	Yes	Yes
Digital artifacts consideration	No	No	Yes	Yes	Yes
CKC model consideration	No	No	Yes	Yes	Yes
MITRE ATT&CK consideration	Yes	Yes	Yes	Yes	Yes

TABLE 7. Comparative analysis of Forensics and related work.

scrutinizes digital artifacts, encompassing both non-volatile and volatile data, for more comprehensive detection.

Another relevant work to Forensics is [38], proposing a reasoning process for incident analysis and composing them into campaigns. Authors propose a logical system applicable to the CKC model, specifying pre- and post-conditions for each CKC phase. However, this work does not focus on cyber-detection, lacks consideration of skipped CKC phases in cyber-attacks, and provides high-level recommendations compared to Forensics. Unlike Forensics, it does not utilize MITRE ATT&CK to specify occurring techniques. Forensics identifies traces of technique operations, mapping them to tactics and CKC phases for detailed incident analysis, as elaborated in Sections 3.2 and 3.3 through CORIs.

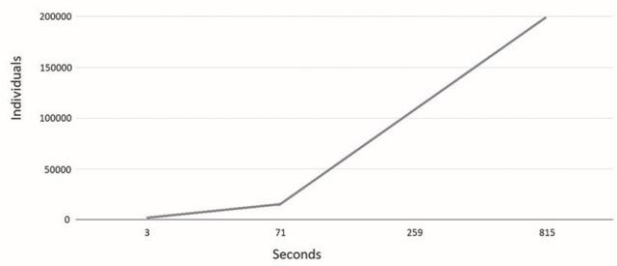

FIGURE 7. Forensics run time via the Drools rule engine and with 16 declarative rules

The latest endeavor concerning Forensics is described in [39], where the authors introduce a methodology leveraging the CKC model and MITRE ATT&CK to discern forensics data, amalgamate them, and correlate them to reconstruct the stages of a cyber-attack. Unlike Forensics, which focuses on detecting ongoing cyber-attacks, this work concentrates on cyber-attack investigation, requiring both the cyber-attack and its traces to have been previously detected. In [39], these traces are referred to as forensics data. In contrast, Forensics automatically identifies an ongoing cyber-attack.

The methodology proposed in [39] entails manually mapping the already detected forensics data to the MITRE ATT&CK techniques using a proof by contradiction approach. This involves comparing the forensics data against the descriptions of MITRE ATT&CK techniques to determine whether they align with the traces of the technique. Conversely, Forensics is an automated cyber-attack detection approach employing rules to identify the operation of each technique by analyzing digital artifacts. As detailed in Section 4.2, each rule is formulated once during the development of Forensics based on the descriptions of MITRE ATT&CK techniques.

While [39] identifies MITRE ATT&CK techniques, it proceeds to map MITRE ATT&CK tactics to the CKC model, albeit utilizing a different mapping strategy from Forensics, as elaborated in Section 3. For instance, [39] associates the "initial access" tactic with the "weaponization" CKC phase. Furthermore, [39] does not account for potential combinations of CKC phases, a feature inherent to Forensics with its COSPs.

Table 7 offers a comparative analysis of the related work presented in this section against Forensics based on seven criteria.

CONCLUSION :

This paper introduces Forensics, a novel approach to cyber-attack detection grounded in digital forensics principles. Forensics leverages the MITRE ATT&CK knowledge base, Lockheed Martin's Cyber Kill Chain (CKC) intelligence model, and digital artifacts obtained from monitored systems. These digital artifacts are acquired using appropriate sensors following established digital forensics protocols to maintain their integrity. Forensics scrutinizes these artifacts to identify MITRE ATT&CK techniques based on the discernible traces left by each technique's specific procedures. Subsequently, recognized techniques are linked with their corresponding MITRE ATT&CK tactics, which are then mapped to related CKC phases.

The implementation of Forensics relies on an ontology and rules represented in the Web Ontology Language (OWL) and the Semantic Web Rule Language (SWRL), respectively, constituting a rule-based detection approach. The ontology facilitates the representation of digital artifacts in an interchangeable and machine-processable format, while the rules analyze the artifact-related data to detect ongoing cyber-attacks. The technologies underpinning MITRE ATT&CK, CKC, OWL, SWRL, and associated rule-based reasoners are open-source, encouraging widespread adoption of Forensics.

Future research endeavors will focus on enhancing Forensics' computational efficiency to expedite the detection of ongoing cyber-attacks. Explorations into big data technologies, such as Hadoop big data clusters, may aid in this optimization process. Additionally, efforts will be directed towards evaluating Forensics against cyber-attacks simulated using MITRE Caldera [40]. Furthermore, the potential integration of machine learning (ML) algorithms will be explored. For instance, the Forensics ontology may serve to define similarity measures for ML models, facilitating the identification of similarities between digital artifacts. ML algorithms will also be investigated for automatically generating rules applicable in ontology-based reasoning to extend the capabilities of Forensics.

Declaration of Competing Interest: The authors affirm that they have no financial conflicts of interest.

Acknowledgments: This work has received partial support from a research grant awarded by the Ministry of Digital Governance of Greece to the Research Center of the Athens University of Economics & Business, Greece (2021-22).

REFERENCES :

[1] National Institute of Standards and Technology, "Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1," Tech. Rep. NIST CSWP 04162018, National Institute of Standards and Technology, Gaithersburg, MD, Apr. 2018.
[2] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, vol. 1. Academic Pub. International, 2011.
[3] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, pp. 16–24, Jan. 2013.
[4] MANDIANT, "M-trends 2021: Insights into Today's Top Cyber Trends and Attacks." [Online]. Available: https://www.fireeye.com/currentthreats/annual-threat-report/mtrends.html. Accessed on Sep. 05 2021.
[5] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and Philosophy," Tech. Rep. 10AOH08A-JC, The Mitre Corporation, McLean, VA, Mar. 2020.
[6] EU ATT&CK community, "Directory of ATT&CK Open Source Tools." [Online]. Available: https://www.attack-community.org/directory/. Accessed on Mar. 08 2022.
[7] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to integrating forensic techniques into incident response," Tech. Rep. NIST SP 800-86, National Institute of Standards and Technology, Gaithersburg, MD, Aug. 2006.
[8] S. Alharbi, J. Weber-Jahnke, and I. Traore, "The Proactive and Reactive Digital Forensics Investigation Process: A Systematic Literature Review," in Information Security and Assurance (T.-h. Kim, H. Adeli, R. J. Robles, and M. Balitanas, eds.), vol. 200 of Communications in Computer and Information Science, pp. 87–100, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
[9] P. G. Bradford and N. Hu, "A layered approach to insider threat detection and proactive forensics," in Proceedings of the Twenty-First Annual Computer Security Applications Conference (Technology Blitz), Citeseer, 2005.
[10] A. Orebaugh, "Proactive Forensics," Journal of Digital Forensic Practice, vol. 1, pp. 37–41, Mar. 2006.
[11] J. Sachowski, Implementing Digital Forensic Readiness: From reactive to proactive process. CRC Press, 2021.
[12] B. D. Bryant and H. Saiedian, "A novel kill-chain framework for remote security log analysis with SIEM software," Computers & Security, vol. 67, pp. 198–210, June 2017.
[13] The MITRE Corporation, "MITRE ATT&CK." [Online]. Available: https://attack.mitre.org/. Accessed on Sep. 29 2021.
[14] V. S. Harichandran, D. Walnycky, I. Baggili, and F. Breitinger, "CuFA: A more formal definition for digital forensic artifacts," Digital Investigation, vol. 18, pp. S125–S137, Aug. 2016.
[15] A. Dimitriadis, Leveraging digital forensics and information sharing into prevention, incident response, and investigation of cyber threats. PhD dissertation, Department of Applied Informatics, University of Macedonia, 2022.
[16] B. L. Krishna, "Comparative Study of Fileless Ransomware," International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 4, pp. 608–616, Apr. 2020.
[17] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, and J. Disso, "Cyber-Attack Modeling Analysis Techniques: An Overview," in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), (Vienna, Austria), pp. 69–76, IEEE, Aug. 2016.
[18] W3C, "OWL 2 Web Ontology Language Document Overview (Second Edition)." [Online]. Available: https://www.w3.org/TR/2012/REC-owl2-overview-20121211. Accessed on Nov. 29 2021.
[19] W3C, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." [Online]. Available: https://www.w3.org/Submission/SWRL/. Accessed on Nov. 29 2021.
[20] "Drools - Business Rules Management System (Java, Open Source)." [Online]. Available: https://www.drools.org/. Accessed on Nov. 29 2021.
[21] S. C. for Biomedical Informatics Research at the Stanford University School of Medicine, "Protégé." [Online]. Available: https://protege.stanford.edu/. Accessed on Nov. 1 2021.
[22] U. Community, "Unified Cyber Ontology (UCO) A foundation for standardized information representation across the cyber security domain/ecosystem." [Online]. Available: https://unifiedcyberontology.org/. Accessed on Sep. 9 2021.
[23] Clark and P. LLC, "Pellet - Semantic Web Standards." [Online]. Available: https://www.w3.org/2001/sw/wiki/Pellet. Accessed on Sep. 10 2021.
[24] B. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding Cyber Threats with ATT&CK-Based Analytics," MITRE TECHNICAL REPORT MTR170202, The Mitre Corporation, Annapolis Junction, MD, June 2017.

[25] MITRE, "Working with ATT&CK | MITRE ATT&CK in Excel." [Online]. Available: https://attack.mitre.org/resources/working-with-attack/. Accessed on Oct. 31 2021.

[26] S. R. Martin, J. J. Lee, and B. L. Parmar, "Social distance, trust and getting "hooked": A phishing expedition," Organizational Behavior and Human Decision Processes, vol. 166, pp. 39–48, Sept. 2019.

[27] T. Halevi, N. Memon, and O. Nov, "Spear-Phishing in the Wild: A RealWorld Study of Personality, Phishing Self-Efficacy and Vulnerability to Spear-Phishing Attacks," SSRN Electronic Journal, Jan. 2015.

[28] Cisco, "2018 Annual Cybersecurity Report," tech. rep., Cisco, Feb. 2018.

[29] BlackBerry, "2021 Threat Report," tech. rep., BlackBerrry, 2021.

[30] ESET, "Threat Report Q3 2020," tech. rep., ESET, 2020.

[31] Trustwave, "2021 Email Threat Report," tech. rep., Trustwave, 2021.

[32] C. Beek, M. Cashman, J. Fokker, M. Gaffney, S. Grobman, T. Hux, N. Minihane, L. Munson, C. Palm, T. Polzer, T. Roccia, R. Samani, and C. Schmugar, "McAfee Labs Threats Report, June 2021," tech. rep., McAffee, June 2021.

[33] Verizon, "2019 Data Breach Investigations Report," tech. rep., Verizon, May 2019.

[34] W3C, "OWL 2 Web Ontology Language Manchester Syntax (Second Edition)." [Online]. Available: https://www.w3.org/TR/owl2-manchestersyntax/. Accessed on Dec. 28 2021.

[35] S. Falconer, "OntoGraf." [Online]. Available: https://protegewiki.stanford.edu/wiki/OntoGraf. Accessed on Mar. 10 2022.

[36] Mandiant, "Today's top cyber trends; attacks insights: M-trends 2021."

[37] "Atomic Threat Coverage." [Online]. Available: https://github.com/atcproject/atomic-threat-coverage. Accessed on Mar. 11 2022.

[38] J. M. Spring and D. Pym, "Towards scientific incident response," in International Conference on Decision and Game Theory for Security, pp. 398–417, Springer, 2018.

[39] C. Liu, A. Singhal, and D. Wijesekera, "Forensic analysis of advanced persistent threat attacks in cloud environments," in IFIP International Conference on Digital Forensics, pp. 161–180, Springer, 2020.

[40] "CALDERA." [Online]. Available: https://caldera.mitre.org/. Accessed on Mar. 16 2022.