Accepted Manuscript

A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories

Umara Noor, Zahid Anwar, Asad Waqar Malik, Sharifullah Khan, Shahzad Saleem

PII: S0167-739X(18)30670-8
DOI: https://doi.org/10.1016/j.future.2019.01.022
Reference: FUTURE 4715

To appear in: *Future Generation Computer Systems*

Received date : 28 March 2018
Revised date : 27 November 2018
Accepted date : 13 January 2019

Please cite this article as: U. Noor, Z. Anwar, A.W. Malik et al., A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.01.022

A Machine Learning Framework for Investigating Data Breaches Based on Semantic Analysis of Adversary's Attack Patterns in Threat Intelligence Repositories

Umara Noor [a, c], Zahid Anwar [a, b], Asad Waqar Malik [a], Sharifullah Khan [a], Shahzad Saleem [a]

[a] *National University of Sciences and Technology (NUST), Islamabad, Pakistan*

[b] *Mathematics and Computer Science, Fontbonne University, St. Louis, MO USA*

[c] *Department of Computer Science and Software Engineering, Faculty of Basic and applied Sciences, International Islamic University, Islamabad, Pakistan*

Abstract

With the ever increasing cases of cyber data breaches, the manual process of sifting through tons of security logs to investigate cyber-attacks is error-prone and time-consuming. Signature-based deep search solutions only give accurate results if the threat artifacts are precisely provided. With the burgeoning variety of sophisticated cyber threats having common attack patterns and utilizing the same attack tools, a timely investigation is nearly impossible. There is a need to automate the threat analysis process by mapping adversary's Tactics, Techniques and Procedures (TTPs) to attack goals and detection mechanisms. In this paper, a novel machine learning based framework is proposed that identifies cyber threats based on observed attack patterns. The framework semantically relates threats and TTPs extracted from well-known threat sources with associated detection mechanisms to form a semantic network. This network is then used to determine threat occurrences by performing probabilistic relationships between threats and TTPs. The framework is trained using a TTP taxonomy dataset and the performance is evaluated with threat artifacts reported in threat reports. The framework efficiently identifies attacks with 92% accuracy and low false positives even in the case of lost and spurious TTPs. The average detection time of a data breach incident is 0.15 seconds for a network trained with 133 TTPs from 45 threat families.

Keywords: Cyber Threat Intelligence, Data Breach Investigation, Tactics Techniques and Procedures, Indicators of Compromise, Belief network, Latent Semantic Indexing

Email addresses: 13phdunoor@seecs.nust.edu.pk (Umara Noor [a, c]), zahid.anwar@seecs.nust.edu.pk (Zahid Anwar [a, b]), asad.malik@seecs.nust.edu.pk (Asad Waqar Malik [a]), sharifullah.khan@seecs.nust.edu.pk (Sharifullah Khan [a]), shahzad.saleem@seecs.nust.edu.pk (Shahzad Saleem [a])

1. Introduction

The pervasiveness of high-speed Internet connectivity has attracted a large majority of businesses to move their sensitive and confidential information and transactions to the clouds. This transition has made business operations flexible and easy for the customers at the cost of making them vulnerable to cyber data breach attacks. In 2016 alone, over 1.4 billion data records were breached in a total of 1792 incidents targeting financial, banking and retail outlets [1]. The situation is getting worse as in the first half of the year 2017, over 1.9 billion records were breached in 918 incidents [2] including the massive data breach of Equifax where 148 million consumers had their personal information accessed [3]. The detailed analysis of these cyber incidents reveals that the adversaries use common attack patterns to compromise their target. Due to this reason, the security community is paying more attention towards sharing and consuming Cyber Threat Incident Reports (CTIR) as a measure of proactive defense against data breach incidents.

The massive size of CTIR and the constant onslaught of new Advanced Persistent Threats (APT) make it nearly impossible for the security analyst to identify a relevant attack's behavioral signature. Currently, there are approximately 1 billion threat indicators publicly available on Hail-a-TAXII [4]. Similarly, IBM X-Force [5] reports thousands of malware on a weekly basis. The Verizon's Data Breach Investigations Report (DBIR) [6] reports millions of data breaches including stolen credit card credentials. Along with these reports, a network administrator has locally available log files [7] containing useful threat data, e.g., DNS [8], packet capture [9], email [10] and IP logs [11]. Among these different types of reports, Structured Threat Information Expression (STIX) [12] encoded CTIRs are more comprehensive because they record the details of cyber breaches as attack observables, indicators, TTPs, incidents, threat actors, campaigns, exploit targets and Course of Actions (COAs). To investigate data breach incidents, so far, a major focus of the security community has been on sharing and identifying indicators and observables. These are low-level threat artifacts comprising of IP addresses, domain names and file hashes. Unfortunately, they have a very short lifespan with respect to threat defense as they are susceptible to change having fewer chances of being reused. The attacker constantly changes IP addresses by buying new attack servers and domain names. While cyber data breach incidents are caused by such malware and APTs that share attack patterns as TTPs. The TTPs represent the attacker's actions for breaching an organization's network. As the indicators and observables change rapidly, similarly the goals and targets of attackers also change frequently. Ironically the attackers' TTPs remain the same and are re-used over and over again with little innovation. This problem was observed in the data breach incidents of some notable organizations like Sony [13], Target [14] and Home Depot [15] where millions of customers were affected. In these incidents, the adversaries' employed the same TTPs to breach into the network. The greatest amount of pain can be inflicted upon the attacker by identifying TTPs in the network and thereby achieving a timely investigation.

In this research work, we propose a novel and automated data breach investigation framework that exploits adversary's attack patterns, i.e., TTPs in CTIR. The framework generates a Threat-TTP-Detection (TTD) network by semantically correlating threat incidents among themselves to identify the existence of a TTP in a network. The TTD semantic network is further augmented to investigate the most probable threat incident family based on detected TTPs in a network, by adopting a machine learning based

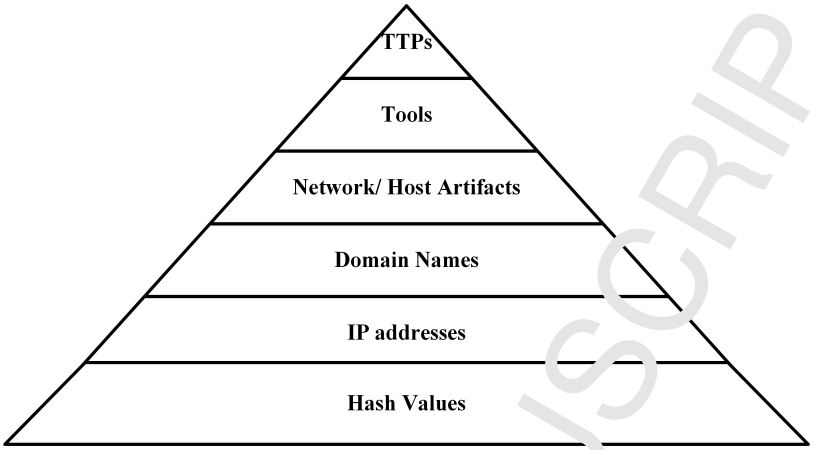

Figure 1: Pyramid of Pain Model [16]

probablistic analysis. The proposed framework reliably detects data breach incidents with an accuracy of 92% and low false positives even in the case of lost and spurious TTPs.

The organization of the paper follows this sequence: The problem statement is discussed in the second section. Based on the problem statement we describe our research methodology and contributions in section three. Section four discusses the related work. The proposed framework design is detailed in the fifth section. The working of this framework is explained in the sixth section with the help of a financial RAM scrapping malware family data breach case study. The effectiveness and efficiency of the framework are evaluated in the seventh section. Finally, in the eighth section, the research work is concluded along with future perspectives.

2. Problem Statement

Cyber data breach incidents are caused by such malware and APTs that share attack patterns as IOCs. There are six levels of IOCs as defined in the Pyramid of Pain model [16, 17] shown in figure 1. The IOCs present in the first three levels are atomic indicators comprising file hashes, IPs, domain names, network and host artifacts. Low-level IOCs, also termed *technical* threat intelligence, are consumed by technical resources such as firewall, Intrusion Detection System (IDS) and spam filters on email servers. In the upper part of the pyramid, there are more generalized threat indicators related to the behavioral attack signatures of threats such as exploit toolkits, malware and TTPs. The high-level IOCs, also termed as *tactical* threat intelligence is consumed by incident responders for investigation, hardening defenses by upgrading systems and policies. Interestingly the indicators in the lowest part of the pyramid can be easily identified and extracted from the threat reports due to their fixed format. Therefore, they can also be easily defended by applying intrusion detection and firewall rules. Currently, the emphasis of the security community is to employ low-level IOCs for sharing and investigating a data breach incident. Unfortunately, these low-level IOCs are only useful for a short time or

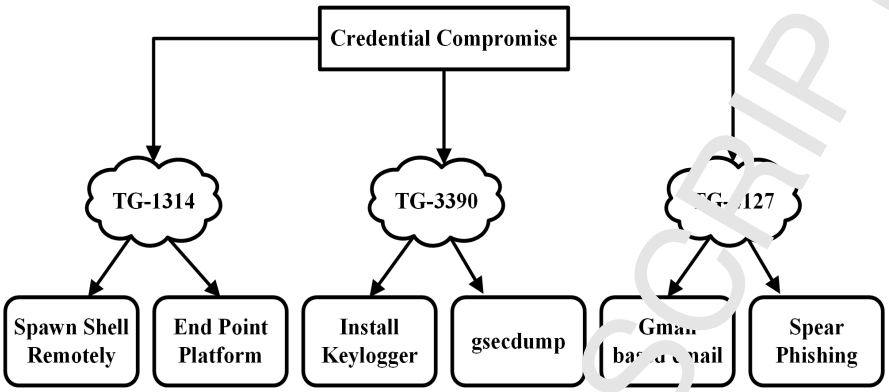

Figure 2: Credential Compromise TTPs for Three Threat Groups

for immediate use. The attacker constantly changes IP addresses by buying new attack servers and domain names. On the other hand high-level IOCs are for a long-term use and provide sophisticated defense against cyber threats. Thus by identifying high-level IOCs, i.e., TTPs the attacker is forced to change his attack patterns.

To illustrate the significance of TTPs in the investigation process, in figure 2 an example of three famous threat groups, i.e. TG-1314 [18], TG-3390 [19] and TG-4127 [20] is provided. All these groups compromise victims' credentials but using different attack TTPs. The threat group TG-1314 uses the victim's remote access tools by compromising credentials of the network's endpoint management platform. The threat group TG-3390 installs a keylogger and employs a publicly available credential dumper tool to get password hashes. The threat group TG-4127 compromises credentials via spear phishing. A well-known example related to TG-4127 is the Democratic National Committee's Gmail-based email account hack during the presidential campaign. Similarly, the goals and targets of these threat groups are also different. The goal of the threat group TG-1314 is to log into the victim's remote access facility while the threat group TG-3390 steals industrial intellectual property. The threat group TG-4127 targets government and military networks for espionage and cyber warfare. This shows that by identifying the presence of a particular threat group related TTP in a network provides a distinct advantage to the analyst in explicitly identifying the cyber threat family.

Our problem statement is thus as follows. Current techniques to diagnose a data breach incident use low level IOCs which are not useful due to their short life span. On the other hand if high-level IOCs, i.e., TTPs are used to investigate a data breach incident, they can allow more accurate detection. Therefore our problem is to identify TTPs based on the low level threat artifacts observed in a network with the help of appropriate machine learning algorithms.

3. Research Methodology and Contributions

To address the problem statement presented in the previous section, here we describe our research methodology from which we will derive our solution and accomplish our research challenges. To accurately investigate cyber threats using adversary's TTPs,

two main challenges need to be addressed. The *first* challenge is that to identify TTPs of an attack, the security analyst has to perform certain detection mechanisms that are frequently not specified in the CTIR. The *second* challenge is that TTPs are rarely referenced using standard identifiers in CTIR. They are mostly reported as unstructured, human understandable textual descriptions that make it difficult to correlate attack incidents of same threat group based on similar TTPs due to synonyms and polysemous words. Manual searching for correlated TTPs is a tedious, time consuming and error prone process which is nearly impossible due to the massive size of CTIR.

The *first* step of the methodology addresses the challenge of identifying the existence of TTPs. The detection mechanisms of our framework are obtained from the publicly available (ATT&CK) taxonomy provided by MITRE [21] that documents adversarial Tactics, Techniques and Common Knowledge. The framework generates a semantic network of Threats, TTPs and Detection menchanisms (TTD) by correlating threat incidents among themselves to identify the existence of a TTP in a network. The *second* step of the methodology addresses the challenge of referencing TTPs in CTIR. The TTPs in the CTIR are semantically mapped into the TTD using their appropriate labels in the ATT&CK taxonomy using Latent Semantic Indexing (LSI). LSI, not only groups documents using semantically similar words but also groups semantically similar words in a document to identify themes or topics of the documents. The TTD is further augmented to diagnose the most probable threat family based on detected TTPs in a network, by adopting a probablistic machine learning based analysis using belief networks between threats and TTPs. The advantage of this approach is that it outperforms sophisticated classification methods, treats all predictor attributes independently and is useful for very large datasets with missing data [22].

To evaluate the working of the proposed framework, we constructed a benchmark dataset from the ground truth data (threat, TTP and detection mechanisms) available on ATT&CK taxonomy [21]. Threat artifacts were compiled based on the threat incidents reported by multiple sources, i.e., IBM X-Force [5], Symantec [23], FireEye [24] and CrowdStrike [25]. These artifacts were populated into the prototype system and automated investigation was performed to determine the presence of attacks.

Our research has the following contributions:

1. This work is the first of its kind that presents a case for emphasizing the analytics of high-level adversarial TTPs. Previous work has primarily focused on the identification of low-level atomic indicators that are trivial for the attacker to change.
2. The results presented highlight the important relationship between security incidents, tactics and artifacts in a way that machines can identify these connections with certain probabilities. Henceforth this research paves the way for cyber security investigations with partial or incomplete information.
3. A well-known standard dataset of unstructured cyber threat incidents has been thoroughly studied and dissected to construct a comprehensive vocabulary of structured TTPs. The existing threat taxonomies are either too specific to a particular domain, e.g., web attacks or are too verbose and descriptive to be useful for machine learning.
4. Our research motivates the development and usage of a common vocabulary for TTPs in CTIR. The SIRS module allows security analysts to identify common TTPs in different documents even if they have different texts but have the same

meaning. At the same time, this also gives SIRS the ability to discover TTPs mentioned in one document but not the other. We illustrate this in the evaluation section by showing how the proposed system identifies missing TTPs in three documents (ATT&CK, IBM X-Force, Symantec) that concern the same threat (AXIOM Group). These TTPs can be derived or essentially imported from one dataset into the other thereby automatically "augmenting" the latter. This is hence an important contribution to our work.

5. The results prove that the cyber threat classes and their associated incidents utilize a finite and typically deterministic collection of TTPs, which characterize them. If this set of TTPs can be systematically documented then the threat groups can be automatically profiled and ultimately stopped. Machines can help investigate these "telltale" threat group signatures. This is not surprising as threat groups constitute real people who specialize in particular tactics and techniques which evolve over time.

4. Related Work

Cyber threat diagnosis based on learning patterns from CTIR is a relatively new domain. We did not find any directly related research work that can be compared to the proposed scheme. However, our research benefits from several existing research domains. First, we studied the different state of the art solutions that deal with cyber threats. Second, we studied research work where CTIR have been used to forecast cyber incidents, suggest investment in data security and operational risk management in financial sectors. Third, we studied how cyber and network security operations can be enhanced by extracting security concepts from unstructured sources and enriching intrusion detection knowledge base to detect attacks and vulnerabilities. Fourth, we studied the most state-of-the-art solutions for malware, APT and intrusion detection using machine learning techniques.

A machine learning based cyber threat detection model to identify the seven top security threats in cloud computing and their remediation is proposed in [26]. The presence and type of attack is determined by training the machine learning model with a limited set of artifacts that depict the activity pattern of the cloud. The artifacts are performance logs of CPU, storage media, network usage patterns of the hypervisor and the guest operating system. In [27], the authors propose an abstraction layer over the Internet to record events from different information systems, and correlate and share these among the partners in order to detect and monitor frauds. A limited set of cyber threats belonging to inter-domain port scans are considered. An automated framework to respond to cyber threats with appropriate response plans is proposed in [28]. It works by integrating and evaluating operational, financial and threat impact models. The data employed is network inventory, security policies, mitigation actions, reachability matrix and vulnerability inventory. The data is local to the network and based on pre-established security guidelines. The proposed framework does not employ CTI data for dynamic risk management. In [29], a fog-based storage approach is proposed for the integrity, availability and confidentiality of users data stored in clouds from a limited set of cyber threats such as malicious modification and data loss.

Publicly available CTIR repositories such as VERIS incident database, Hackmageddon, and the Web Hacking Incidents Database have been used by Liu et al [30]. The

purpose is to forecast the cyber security incidents based on the network state. This state is depicted with externally detected properties i.e mismanagement symptoms, DNS or BGP misconfigurations and a time series of malicious activity including scanning, phishing and spam. The VERIS incident database and Alexa Web Information Service AWIS is used by Sarabi et al [31] to devise a categorization metric. It suggests the investment a business should do on achieving targeted and effective resource allocation for security incidents dealing with critical data of that business. The incidents are categorized based on the type, the originating source, motive, and assets used related to that incident. Similarly, Vasily [32] used the VERIS incident database to investigate data breaches in financial sectors. To manage operational risk a belief network is formed. The results obtained from the model assist management in understanding the problem areas relevant to their businesses and make decisions for future betterment. In [33], the authors propose the use cases of cyber threat intelligence sharing using STIX to gain information about the disruptive consequences of cyber threats and data breaches on society. A threat analytics framework to contextualize the massive Cyber Threat Intelligence (CTI) is proposed in [34]. The CTI standards, network configurations, and Common Vulnerabilities Exposure (CVE) are represented using ontological formal specification The ontology developed is employed to semantically reason about the cyber threat relevance to a network their likelihood and the vulnerable and affected assets. The focus is on the network relevance towards a cyber threat instead of detecting or correlating existing threats with new cyber threats. In [35], the authors performed a comparative analysis of the STIX, IODEF, VERIS and X-ARF cyber security incident reporting formats. The study reveals that STIX is the most comprehensive and practicable cyber security incident reporting format. In [36], the authors describe threat intelligence use cases with STIX. The purpose is to improve the automated management of threats. The information from different threat intelligence providers and network elements is correlated. A comprehensive survey of the existing CTI standards in the context of cyber threat information sharing is given in [37].

Security concepts extracted from unstructured online web sources proved to be really helpful in enhancing traditional knowledge bases of intrusion detection systems. More et al. [38] proposes an intrusion detection system which extracts security concepts from text. These concepts are compared with monitoring sensors' logs with the help of a reasoner for generating security alerts. The security concepts are extracted from heterogeneous sources. The extracted threat is used to populate a security ontology borrowed from [39]. In a related research, Mulwad et al [40] present a framework for vulnerability extraction and cyber attack related information from web text and compare it with wikitology concepts [41]. A model proposed by Joshi et al [42] takes text, vulnerability descriptions, blog posts and security bulletins as input and automatically extracts entities and populates concepts in DBpedia [43]. In DBpedia, these concepts are matched and assigned corresponding class values. Similarly Bridges et al. [44] present a maximum entropy model for automatic labeling of security text. An approach for early detection of real-world exploits and vulnerabilities from Twitter is proposed by Sabottke et al [45]. It observes tweets of security vendors, hackers, and administrators. More vulnerabilities can be detected using Twitter security forums than actually present in the proof of concept public repositories. To extract useful content from news stories Ryan et al. [46] use a text mining approach. To measure relatedness between concepts collected from news stories Latent Semantic Analysis (LSA) is used. To semantically contextualize IoT data, Mário

et al. [47] proposes an approach for extracting semantic features from publicly accessible web services to facilitate M2M based communication. A cyber security ontology called CRATELO is introduced by Oltramari et al. [48]. It consists of three levels that use a simplified version of Dolce named Dolce spray as first level ontology. The proposed cyber security related ontology SECO as a middle level and domain level ontology OSCO for cyber operations. The research considers spatial and temporal properties of attacks and the attacker's host and network artifacts are not considered. In the related research the same authors [49] introduced the concept of trust management in CRATELO, e.g., any value delay which is out of an acceptable network delay range will be unreliable. Details of acceptable delay for different networks are not given. To investigate APTs via artifacts collected from cloud apps, Christian et al. [50] developed a taxonomy of security mechanism circumvention techniques by analyzing case studies of iOS cloud apps.

A Support Vector Machine (SVM) based signature free malware detection based on the selection of filter based malware feature is given in [51]. The features employed are Application Program Interface (API) call statistics. Similarly, a machine-learned model based malware detection system is provided in [52]. The malware features are obtained from both benign and malicious executables. The features considered are file hashes, malicious IP addresses and malicious external calls. The machine learning techniques used are SVM and decision trees. In [53], the authors propose a machine learning based malware detection engine. The malware features considered are static, behavioral and permission related. The static features are obtained from APK files. The behavioral features are obtained by executing the Apps in an emulator. The permissions are selected using information gain. The J48 (open source Java-based implementation of the decision tree) classifiers outperforms other techniques. In [54], a machine learning based approach is used to detect and predict APT. The features of APT considered are low-level IOCs, i.e., executable files, malicious file hashes domain names, IPs and SSL.

The aforementioned works introduced the groundwork for cyber security related information retrieval and set the stage for investigating cyber data breach incidents. However, the models and ontologies employed are very basic and do not cover all aspects of threats both low-level artifacts and TTPs. They only capture vulnerabilities exploited and their needs. The proposed threat investigation is based on semantic analysis of adversary's attack patterns by employing machine learning techniques for predicting the threat family class. In order to identify cyber threats, security analysts must perform adequate investigations to identify threats in the context of recent data breaches and exploits found in CTIR.

5. Data Breach Investigation Framework

A high-level architecture of the proposed framework is shown in figure 3. It can be broadly divided into three segments: *Semantic Indexer and Retrieval System (SIRS), TTD Semantic Network* and *Cyber Threat Prediction*. The input to the system is CTIR and ATT&CK documents. A CTIR corresponds to a single cyber threat or incident while an ATT&CK document may correspond to many detection mechanisms related to a TTP. The SIRS segment semantically indexes CTIR and ATT&CK documents, maintains a TTP dictionary extracted from threat documents and retrieves a ranked list of CTIR and ATT&CK documents for each TTP present in the dictionary. The list of ranked documents is combined to connect cyber threats, TTPs and detection mechanisms

based on a higher rank specified by the ranking function. A belief network is further trained between the TTPs and the cyber threat incident to predict cyber threats based on detected threat artifacts. The last segment, i.e., *Cyber Threat Prediction* takes as input the *TTD* and the detected threat artifacts to produce a reliable threat prediction. The functional details of each segment are given in the following subsections. The details of notations used are given in Table 1.

5.1. Semantic Indexer and Retrieval System

Table 1: Data Breach Investigation Framework Notations and Definitions

Notation	Definition
A_{li}	A ATT&CK documents set, associated with the TTP ttp_i
TTP_{ti}	A TTPs set, associated with a threat t_i
T_{si}	A threats set, associated with the TTP ttp_i
D_{di}	A set of detection mechanisms, associated with d_i
TIndex	Latent Semantic Indexed threat documents
AIndex	Latent Semantic Indexed ATT&CK documents
TRank	A list of ranked threat documents associated with ttp_i
ARank	A list of ranked ATT&CK documents associated with ttp_i
TTPD	A detected ttps set
$TTPD_{ti}$	A set of detected ttps due to threat t_i
$TTPU_{ti}$	A set of undetected TTPs due to threat t_i
p_i	A set of predicted threats based on *TTPD*
$S(t_i)$	Threat support function to measure the maximal support of TTPD towards t_i
P'	A new set of threat predictions, p_i based on *TTPD*
TTPM	A set of correlated but missing TTPs associated with threat in a prediction
TTPE	A subset of *TTPM* that are present
TTPN	A subset of *TTPM* that are not present

The structure of a general CTIR document is shown in figure 4. In the case of this example, it is an excerpt of a STIX encoded CTIR document reporting the Backoff malware belonging to the financial systems-breach threat family [55]. Each CTIR document is assigned a unique identification number using the *id* attribute in the *STIX_Package* element. The low-level IOCs are represented by the *Indicator* STIX element. The high-level IOCs are represented by the *TTPs* STIX element. Both kinds of IOCs have unique IDs to distinguish them. Each TTP is also assigned a unique identification number using the *id* attribute in the *TTP* element. The details of the TTPs are given in the *Description* sub-element of the *TTP* element. The STIX standard does not directly support the specification of detection mechanisms, which are typically not found in cyber threat reports. This is in order to enable security administrators to identify TTPs in their network, those found in the CTIR must be mapped to their associated detection mechanisms. The detection mechanisms are provided in the ATT&CK repository by MITRE [21]. In this repository, ten tactic categories are defined i.e *Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Execution, Collection,*

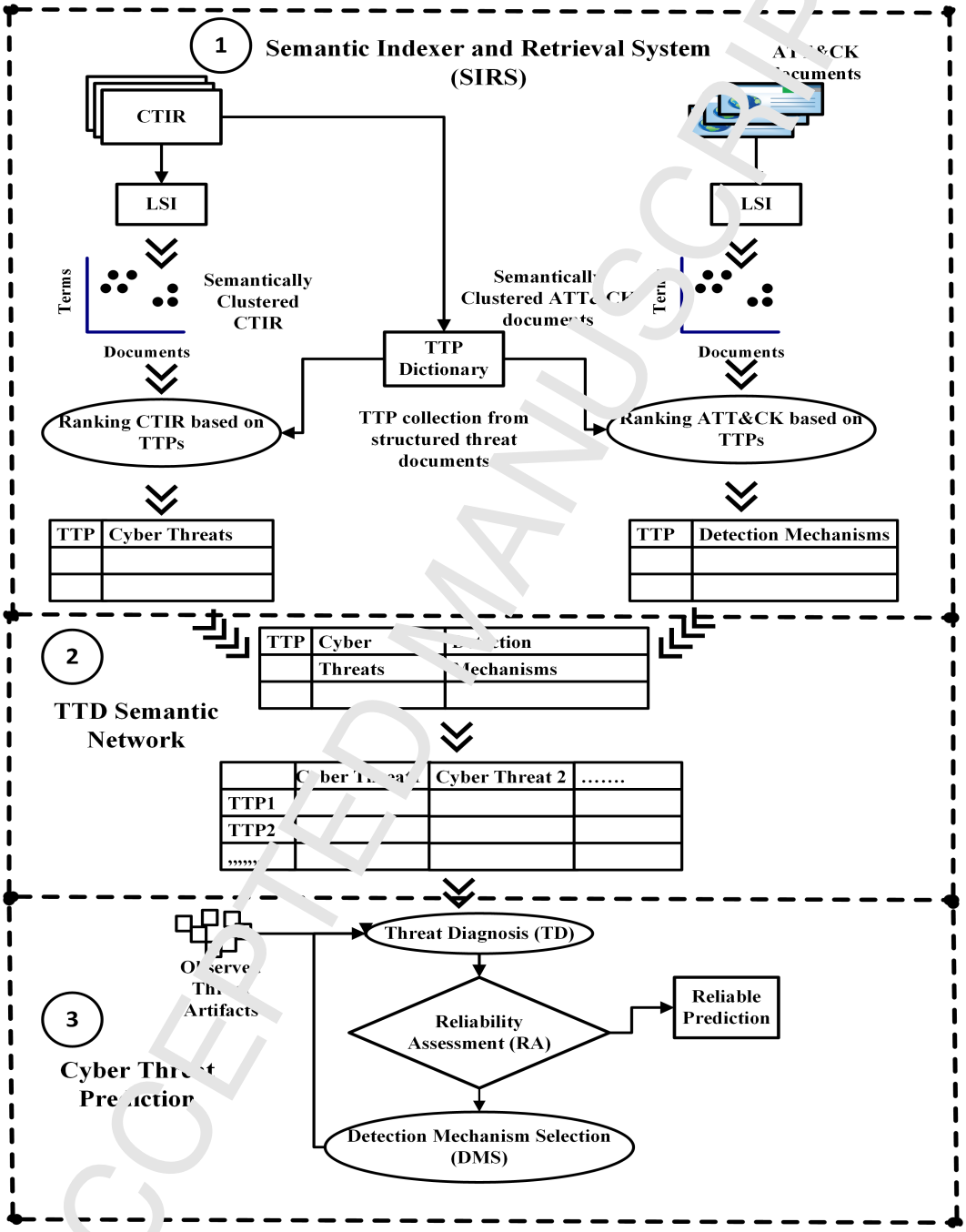

Figure 3: Data Breach Investigation Framework

Exfiltration and *Command and Control*. Each category specifies the adversary's techniques with threat examples and actions for detection and mitigation. An example of an ATT&CK document describing *Application Deployment Software* technique is shown in figure 5. Each technique has its own unique identification number. The technique's title is a generalized taxonomic label which is difficult to map to comprehensively described TTPs in the CTIR.

The SIRS algorithm is provided in listing 1 and it's detailed explanation is provided here. In addition, its working with respect to a specific case study of financial malware is provided in section 6.2.1. The SIRS segment constitutes a main function *SEM_INDEX_RETRIEVE* whose job is to retrieve ranked CTIR and ATT&CK documents for each TTP. The function takes as input a set of m CTIR documents $T{=}t_1, t_2, ..., t_m$ and a set of l ATT&CK documents $A{=}\ a_1, a_2, ..., a_l$. T represents individual attack instances, malware families or software tools. A set of k detection mechanisms $D{=}d_1, d_2, ..., d_k$ can be used to verify existence of a set of n TTPs $TTP{=}ttp_1, ttp_2, ..., ttp_n$. The first step, shown in line number 2 to 4 extracts TTPs from the CTIR and archives them in a dictionary. As mentioned in the introduction section, we have used STIX encoded CTIR documents. STIX uses the commonly used XML format to import, export and share data between different platforms. However, to analyze XML encoded data, these files can be converted into different flexible formats. A common practice is to import the XML encoded data into spreadsheets. To extract TTPs from the STIX encoded CTIR, a dictionary is created on line number 4 whereby the CTIR files are imported as separate rows into a spreadsheet and the columns constitute the *CTIR ID, TTP ID* and *TTP Description*. An excerpt of the data structure of the TTP dictionary is shown in figure 6.

Figure 4: An excerpt of STIX encoded CTIR document reporting Backoff financial malware [55]

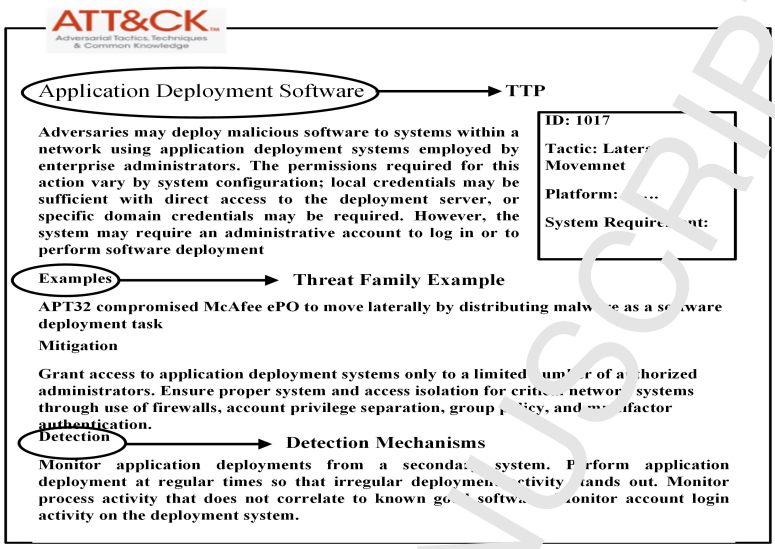

Figure 5: An ATT&CK MITRE document specifying adver y's TTP, threat example, mitigation and detection mechanisms

CTIR ID	TTP ID	TTP Description
Feed-4170e5e758e50d29934ade2e19cd0c02	TTP-b500e -4955 3-d8j2-8c3d4b304	UsesTOR hidden services
Feed-4170e5e758e50d29934ade2e19cd0c02	T -f9d6e e-a7d6-0094- 8c-535f8c 1d73	Encrypts every file it can find that isn't an EXE or DLL
Feed-4170e5e758e50d29 4ade2e19cd0c02	T -006e0a14-677b-dfed- 447-4 59298c6272	Downloads the encryption key from its CnC server

Figure 6: An Excerpt of TTP Dictionary

The second step shown on line number 6, is to semantically correlate the TTPs in the CTIR among themselves (line number 7) and with the TTPs in the ATT&CK documents (line number 8). In figure 7, the semantic relevance among TTPs in ATT&CK documents and CTIR is shown. On the left-hand side, there are five ATT&CK documents. The document's title is the label of the TTP. On the right hand side, there are three different threats of financial malware family, i.e., Backoff, Treasure Hunt and FastPOS. It can be seen that there are some common TTPs, e.g., *key logging functionality* and *Deliver keylog data* but due to different textual descriptions, it is hard for a machine to correlate them using a simple keyword matching technique. Similarly, It can be seen that the titles of relevant ATT&CK TTP documents represent the general labels or taxonomic classes of the TTPs, which is again hard to correlate with the textual TTP description in the CTIR. For this purpose, we use LSI to semantically map TTPs with each other. In figure 7, the TTPs are numbered on both sides. This numbering represents their semantic rele-

vance which is the desired outcome of using LSI. In the past, researchers have successfully used LSI for enhancing the semantic understanding of text documents in web search [56], recommendation systems [57], personality identification [58] and automatic assignment of multi-level security labels [59]. The traditional keyword-based lexical matching techniques retrieve information by matching query terms with the terms in the text corpora. However, they can be erroneous matches due to inaccurate concept matching caused by synonyms and polysemous words. Thus instead of using the simple keyword search, we index CTIR and ATT&CK documents using LSI. The TTPs are searched for semantically relevant concepts or topics in statistically derived conceptual indices (line number 11-24).

LSI assumes a latent pattern in document terms which is hidden due to variable ways to represent a concept. The dimensionality of the sparse (with more zero entries) term-document matrix (representing the frequency of terms in the documents) is reduced using Singular Value Decomposition (SVD) to figure out the hidden word patterns in the documents. The relevance between the TTP query and the indexed documents is calculated by taking the cosine of the two vectors. The documents are ranked by the distance to the TTP queries. A smaller angle leads to a large cosine value that means the document has high relevance to the given TTP query. A larger angle leads to a smaller cosine value that means the document is low in relevance to the given TTP query.

To connect TTPs in the dictionary among themselves, the TTP descriptions in column 3 of the dictionary are provided one by one as search terms in the CTIR corpora. If there is a match then a new row is created in the dictionary with the same TTP ID as the ID of the TTP that was matched. Similarly, to connect TTPs in the dictionary with the detection mechanisms, the TTP descriptions are provided as search terms in the ATT&CK corpora. If a match is found, the corresponding detection mechanism is fetched and added as a new column in the dictionary.

The Latent semantic indexer indexes CTIR and ATT&CK documents. The Semantic retrieval system retrieves a ranked list of CTIR and ATT&CK documents for each TTP present in the TTP dictionary. The ranked results against each TTP are merged and further provided to the second segment, i.e., TTD network where TTPs are semantically linked to their corresponding detection mechanisms.

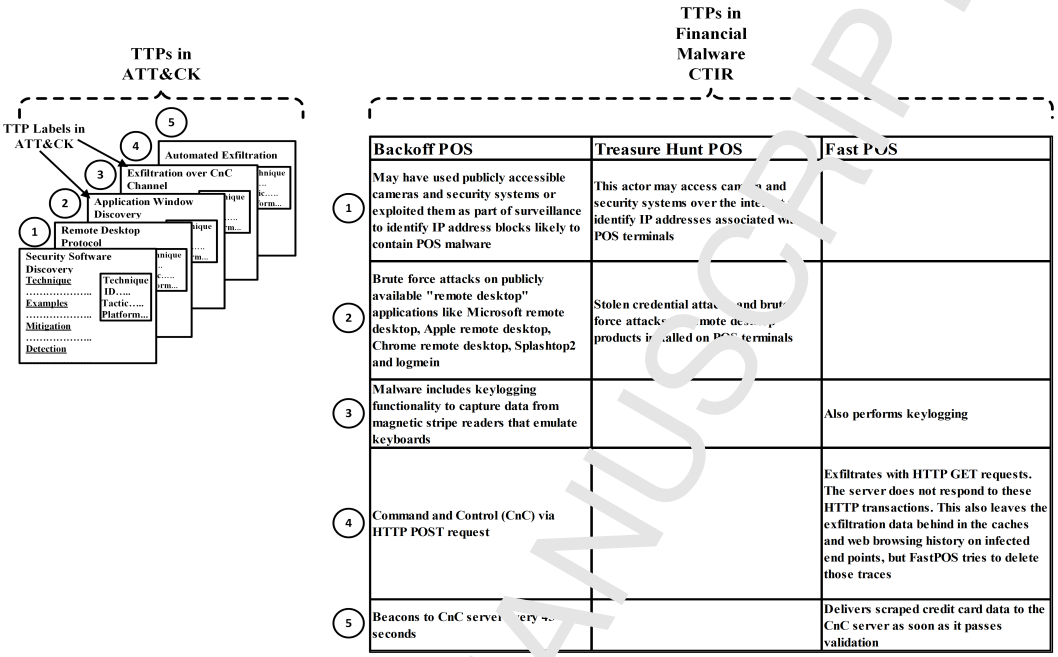

Figure 7: Semantic correlation among TTPs in CTIR and with TTPs of ATT&CK documents

Algorithm 1 Semantic Indexer and Retrieval System

Input: CTIR T, ATT&CK documents A

Output: An n-element array of Ranked CTIR and ATT&CK documents (TRank, ARank) for each TTP

function SEM_INDEX_RETRIEVAL(T, A)
  TTP_Dictionary= Extract_Relevant_Fields(T) ▷ Populate TTP_Dictionary from CTIR
  LSI_Indexer(T,)
  Semantic_Retrieval(*TTP_Dictionary, TIndex, AIndex*)
end function
function LSI_INDEXER(T, A)
  TIndex=LSI(T)
  AIndex= LSI(A)
end function
function SEMANTIC_RETRIEVAL(*TTP_Dictionary, TIndex, AIndex*)
  for $= 1$ to m **do** ▷ m = count(ttp_i) where $ttp_1, ttp_2, ..., ttp_m \in$ TTP_Dictionary
    Search ttp_i in $(TIndex, AIndex)$
    TL_i= Rank($TIndex$) for ttp_i ▷ Where Rank - LSI document ranking
    Select $tl_i \in TL_i > Threshold$
    for $j = 1$ to n **do** ▷ n = count(tl_j) where $tl_1, tl_2, ..., tl_n \in TL_i$
      TRank= add_Record(tl_j, ttp_i)
    end for
    AL_i= Rank($AIndex$) for ttp_i
    Select $al_i \in AL_i > Threshold$
    for $k = 1$ to r **do** ▷ r = count(al_k) where $al_1, al_2, ..., al_r \in AL_i$
      ARank= add_Record(al_k, ttp_i)
    end for
  end for
  return TRank, ARank
end function

5.2. TTD Semantic Network

The SIRS segment retrieves semantically relevant cyber threat reports and adversary's techniques documents. The next step is to link threats to their respective TTPs and detection mechanisms. A dependency table $P_{T\times S} = p(ttp_i/t_i)$ is built between the threat incidents and the TTPs. Thus TTD represents semantic relations of three independent concept sets, i.e., threat set, TTP set and detection mechanisms set. TTPs and threats are connected through links to show that the two are dependent. TTPs and detection mechanisms are also similarly connected to show dependencies between the TTPs and the detection mechanisms as described in the ATT&CK taxonomy.

Algorithm 2 Threat-TTP-Detection Semantic Network

Input: *TRank, ARank*
Output: TTD Semantic Network

function TTD_SEMNET(*TRank, ARank*)
  TTD_NETWORK(ttp_i, *TRank, ARank*)
  BELIEF_NETWORK(*TTD Network*)
end function
function TTD_NETWORK(ttp_i, *TRank, ARank*)
  for ($ttp_i \in TTP$) **do**
    add ($t_i \in TRank$)
    add ($d_i \in D \in ARank$)
  end for
  return TTD Network
end function
function BELIEF_NETWORK(*TTD Network*)
  for $t_i \in T$ **do**
    $\omega(ttp_i/t_i) = \frac{p(tt_i/t_i)}{\sum_{t_i \in T_{ttp_i}} p(ttp_i/t_i)}$
    $\mu(t_i/ttp_i) = \frac{(ttp_i/t_i)p(t_i)}{\sum_{t_i \in T_{ttp_i}} \omega(ttp_i/t_i)p(t_i)}$
  end for
  return TTD Semantic Network
end function

The stepwise details to build a TTD semantic network are given in algorithm 2. The working of the TTD is explained in section 6.2.2 with the financial malware case study. After initial mapping between threats, TTPs and detection mechanisms, a belief network is trained between threats and TTPs in order to predict threats based on the presence of certain artifacts. It is based on a threat support function $S(t_i)$ to measure the maximal support of the detected TTPs towards a threat occurrence. The algorithm assumes that all predictor TTPs are independent of each other. The historical artifacts signifying the existance of a cyber attack are gathered to calculate conditional probability between TTPs and threats, i.e., $p(ttp_i/t_i) \in (0, 1)$. These probabilities are based on historical data that forms a frequency table in the Threat-TTP mapping. The frequency table may need to be normalized to eliminate null values. This is done by adding 1 to all entries of the mapping table and thus eliminating entries with "0" or null value. The normalized table is used to calculate the normalized likelihood or normalized conditional

probability $\omega(ttp_i/t_i)$ to show the support of each threat t_i to their detected ttp_i shown in equation 1.

$$\omega(ttp_i/t_i) = \frac{p(ttp_i/t_i)}{\sum_{t_i \in T_{ttp_i}} p(ttp_i/t_i)} \tag{1}$$

Using the above normalized likelihood table, the normalized posterior probability $\mu(t_i/ttp_i)$ can be computed by using Naive Bayes in equation 2 [22]:

$$\mu(t_i/ttp_i) = \frac{\omega(ttp_i/t_i)p(t_i)}{\sum_{t_i \in T_{ttp_i}} \omega(ttp_i/t_i)p(t_i)} \tag{2}$$

The predicted class is the one that has the highest posterior probability. All the detected TTPs are considered for such support and for each attack for a different kind so in order to extract a threat prediction all such support values are combined to find a best candidate threat prediction set with maximum support value $S(t_i)$ shown in equation 3.

$$S(t_i) = \frac{\sum_{ttp_i \in TTPD_i} \mu(t_i/ttp_i)}{\sum_{ttp_i \in TTP_{t_i}} \mu(t_i/ttp_i)} \tag{3}$$

The output of the second segment is a TTP network which is provided as input to the third segment, i.e., cyber threat prediction.

5.3. Cyber Threat Prediction

The cyber threat prediction segment can be further divided into three functional modules: 1) Threat Investigation (TD), 2) Reliability Assessment (RA) and 3) Detection Mechanism Selection (DMS). The stepwise details of the cyber threat prediction segment are given in algorithm 3. The working of each functional module is explained in section 6.2.3 with POS malware case study.

The responsibility of the threat investigation module is to produce a predicted threat set P' given a set of detected TTPs i.e *TTPD*. P' may contain a set of attack predictions $(p_1, p_2, ..., p_n)$. p_i is a threat actor that characterizes the TTPs detected. Next, the reliability module determines if any of the predictions $p_i (p_i \in P')$ is reliable. The first reason for reliability assessment is that a prediction with low reliability can lead to an incorrect prediction and thus waste the time and resources of an organization. The second reason for assessing reliability is to determine the presence of poisoned or spurious symptoms, i.e., detected TTPs. The Reliability Assessment (RA) equation used to measure the reliability of prediction p_i is given in equation 4. The $TTPD_{ti}$ represents a set of all detected TTPs due to threat t_i. The TTP_{ti} represents a set of all TTPs, associated with threat t_i.

$$RA(P') = \frac{\sum_{t_i \in p_i} (TTPD_{ti}/TTP_{ti})}{p_i} \tag{4}$$

The threat investigation is deemed complete if a high reliability is determined. If the relevant TTPs related to the threat prediction p_i were not detected then, *TTPM*, the set of undetected TTPs that most reliably contribute to explaining p_i are considered by the Detection Mechanism Selection module. This outputs a set of existing TTPs, i.e., *TTPE* and non-existing TTPs *TTPN* based on the least cost detection mechanisms and the prediction p_i reliability value is recalculated. If the reliability value increases

Algorithm 3 Cyber Threat Prediction

Input: TTD Network, TTPD
Output: Reliable Prediction

function THREAT_PREDICTION(*TTD Network, TTPD*)
    THREAT_INVESTIGATION(ttp_i, *TTD Network, TTPD*)
    RELIABILITY_ASSESSMENT(P')
end function
function THREAT_INVESTIGATION(*TTD Network, TTPD*)
    $S(t_i) = \frac{\sum_{ttp_i \in TTPD_i} \mu(t_i/ttp_i)}{\sum_{ttp_i \in TTP_{t_i}} \mu(t_i/ttp_i)}$
    $P' = \text{Max}(St_i)$
    return P'
end function
function RELIABILITY_ASSESSMENT(P')
    $RA(P') = \frac{\sum_{t_i \in p_i} (TTPD/TTP)}{p_i}$
    return P'
end function
if $P' > Threshold$ **then**
    Reliable Prediction
else
    if $TTPMexists$ **then**
        Threat_Investigation$(TTD, TTPE)$
    else
        select low cost d_i
        Reliability_Assessment$(P' = TTPE, TTPN)$
    end if
end if

sufficiently then the prediction procedure terminates, otherwise the threat investigation module receives *TTPD*, *TTPU* for a new threat prediction.

6. Financial Malware Case Study

The working of the data breach investigation framework is elaborated with the case study of financial memory scrapping malware [60]. The malware steals identity and payment records by reading device memory at the retail checkout. The TTPs of a few malware instances extracted from multiple sources [24], [15], [3] are mapped to their appropriate cyber kill-chain stages to understand the data theft process. The TTD map is built by semantically mapping TTPs extracted from multiple sources to financial malware classes and their corresponding detection mechanisms as defined in the ATT&CK repository. The TTD network and the detected threat artifacts are given as input to the cyber threat prediction segment to investigate threat occurrence and the presence of more TTPs by applying appropriate detection mechanisms. A detailed working is explained in the following subsections.

Table 2: Financial Malware TTP mapping to Cyber Kill Chain Phases

Reconaissance	Weaponize & Deliver	Exploit	Install	Command and Control	Actions and Objectives
Phishing	Hardcoded CnC addresses	Handle Service Control Process Request	Use Enum Process	Use HTTP POST	Collect System Information
Drive by Download	Hardcoded Process Names	Use [illegible] 32Snapshot method	Custom Search Function	Use HTTP GET	Injects Code
Luring Attack	Hardcoded File Names	Use Blacklist	Use Regexes	Use FTP Server	Exfiltrates data as Hex digits
	BOT Function	Hooks API	Encodes data as Base64	Use SMTP	Manually data exfiltrated
	Socially Engineered File names		Encodes data as RC4	Use TOR	
	Pretends to be JAVA		Store Results in logs	Use InternetOpen URL	
	Code in VBScript		Performs Luhn Validation	Send PHP() subroutine	
	Code Obfuscation		Installs Watchdog	Grab Browser Form	
			RDP Brute Force		
			Source Code Updated		
			Exploit Auto Start Runkey		
			Copy files in %APPDATA%		
			Registers to bypass firewal		
			Has a Kill Switch		

6.1. Mapping Financial Malware to Cyber Kill Chain Phases

Different TTPs have been used over the ages to steal payment cards' data. One technique is to target victims by downloading keylogging tools on their systems. But unfortunately, keyloggers are not capable of retrieving data stored on cards' tracks. The attackers concentrated on victimizing corporations rather than individuals that employ POS terminals to handle transactions conducted through cards. The vulnerability targeted involves stealing cards' information temporarily held in the memory of point of sale terminals as clear text. Therefore, a specialized malware is developed for harvesting the data stored in the memory. This specialized malware is typically called a 'RAM scraper' and presently there are over 50 different families of RAM scrapers that exist for stealing payment card details in various ways. To accurately diagnose the family of RAM scraper, the security analyst has to perform certain investigative actions to discover the behavior and TTPs of the malware. Thus the TTPs are mapped to the phases of the cyber kill chain model to elaborate the theft incident. TTPs of eleven financial malware samples, i.e., Rdasrv, Alina, VSkimmer, Dexter, BlackPoS, Decebal, JackPOS, Soraya, Chewbacca, BrutPOS, and Backoff are collected and compared. The TTPs for each cyber kill chain phase are discussed below with a summary provided in Table 2.

In the *reconnaissance* phase, phishing techniques are often adapted to break into the victim's account in a network. Other techniques used are to discover the structure of the network and their exploitable vulnerabilities. The purpose of the adversary in this phase is to identify the victims and vulnerable resources of the target network.

The detection mechanisms related to this phase comprise all those proactive measures that should be taken to avoid attackers giving a chance of identifying network structures and selecting vulnerable targets. This includes user training, enforcing the principle of least privilege for all the users at every access and constantly assessing the network for risks associated with each vulnerability [61].

Once the intruder gets to know about his target, the malicious content is prepared based on the type of vulnerabilities present. This phase is known as *Weaponize & Deliver*. Once the malware is prepared it is sent to the target network bypassing the security controls. Malware is formulated as either single components or multiple components performing different functions. The processes to be searched in the RAM of the POS device are hardcoded in the binary. The file names used to store malware on the system are also sometimes hardcoded in the binary. The file names are carefully selected using social engineering techniques to avoid detection. In the same way, the Command and Control (CnC) addresses to transfer the stolen content are also hardcoded. Again the investigation doesn't start at this point. It starts when the malware is finally delivered to the target. So these two phases also depend on the proactive measures taken by the target as discussed in the previous step.

The next phase is the *exploit* phase in which the malware actually begins to exploit the vulnerabilities. The TTPs used in this phase register the POS malware as a service. The execution of the malware is controlled by the service control handler. To detect this TTP, the security analyst performs certain detection mechanisms. In the first step, changes to the service registry are monitored. Secondly, the frequency of modification to existing services is checked. Thirdly, the security analyst looks for any abnormal change to service binary path location which is not conventionally meant for that service and is also irrelevant to software updates. After successful registration, the POS malware starts

scanning the RAM. The malware uses the *CreateToolhelp32Snapshot* method for scanning the active RAM processes. A list of blacklisted processes can help in identifying tracks 1 and 2 card data. Some malware uses regular expressions to do this, but it is a slow process. An efficient way for scanning processes is "process enumeration" in which all the processes in the POS RAM are scanned to search track 1 and 2 data. The adversary uses the Windows Management Instrumentation (WMI) [62] tool for process enumeration. To detect process enumeration, the security analyst should monitor process executions. To obfuscate the malware's extracted content, the adversary uses *Base64* encoding [21].

The next phase is the *installation* phase in which the malware tries to be persistent. There are multiple steps followed to achieve persistence, e.g., malware code may be actively developed and regularly updated by installing latest code versions from the Internet during execution. It can be detected by carefully examining triggers from antivirus and other security tools. The malware may add a file path to auto start run key that boots it each time the system is turned on. This can be checked by detecting irrelevant changes to registry run keys and checking the start folder for changes and additions. Similarly, the malware may add itself to a firewall as an authorized application via registry keys or install a keylogger in the current working directory. This can be detected by monitoring irrelevant changes made to the accessibility utility binaries or binary paths. Keylogger installation can be tracked through changes made to DLLs [21].

The next phase is the *command and control* phase in which the malware establishes a covert connection with the CnC server to give the attacker a direct control over the target network. This connection may be established using HTTP POST/ GET requests, connecting to FTP servers, using emails through SMTP protocols and bots. To detect CnC communication the detection mechanisms used by the security analyst include analyzing network data for uncommon data flows via unseen processes, processes communicating over the network that do not require that function and irrelevant ports for application layer data. Similarly, the security analyst can monitor the processes for their file access patterns and network behavior. Also there can be unwanted network communications and network access without a user-driven request.

Action and objectives is the final cyber-kill chain phase in which the attacker steals payment card data, temporarily stores it on the disk and then exfiltrates that data. To collect system information, host enumeration techniques are used. To detect host enumeration, the security analyst investigates the execution of processes and detect the time when programs are doing system enumeration. The presence of keyloggers can be detected by observing modification in the DLLs. To detect code injection, the process registries should be analyzed for changes. Similarly, data encryption can be detected by finding high data entropy that is an indication of data encryption [21].

Thus we have seen that for each phase of the intrusion kill chain model there exist adversary patterns in the form of TTPs that can be mapped to their particular attack and detection mechanisms.

6.2. Threat Diagnosis for Financial Ram Scrapper Malware

The working of the data breach investigation framework is detailed here with the help of a case study of financial RAM scraping malware. The details of each segment are given in the following subsections.

6.2.1. SIRS for financial Malware

The process of threat diagnosis for data breach incidents caused by financial malware starts with building a TTD network by extracting TTPs from financial malware related CTIR and semantically indexing them using LSI. Once when all the TTPs are extracted, semantic relevance among every single TTP and all the financial malware attacks mentioned in CTIR are found by searching the TTP in the LSI indexed financial malware CTIR. In figure 8, the search result of financial malware TTP related to establishing command and control connection using email or SMTP protocol is shown. The keyword "use email or SMTP for exfiltration" is searched in semantically indexed CTIR documents and we get relevance probability for each. Based on the results it can be seen that the TTP is present in BlackPOS, Dexter, and Vskimmer. This means that the TTP is used by the adversary of that particular attack campaign. The results of such semantic relevance can be mapped in the form of a Threat-TTP frequency table as shown in Table 3 where the table entries specify the frequency of threat incident occurrence. Similarly, we indexed all the cyber attack techniques mentioned in ATT &CK framework using LSI. Each TTP is searched against the indexed documents to look for the relevant detection mechanisms. In figure 9, the search result of financial malware TTP related to establishing command and control connection using email or SMTP is shown. The keyword "use email or SMTP for exfiltration" was searched in semantically indexed ATT&CK documents and we get relevance probability for each.

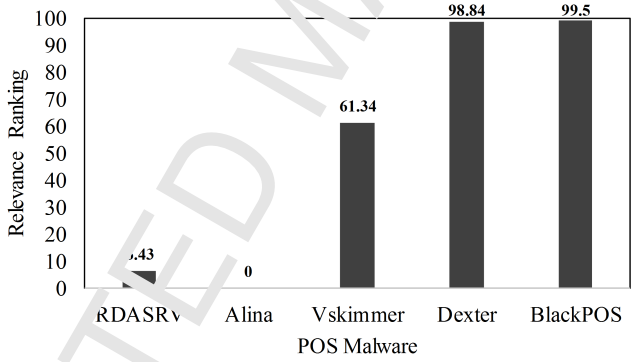

Figure 8: Search results for "use email or SMTP for exfiltration" TTP in LSI indexed CTIR

Table 3: Financial Malware Threat-TTP Frequency Table

TTP ID	Description	RDASRV	Alina	Vskimmer	Dexter	BlackPOS
TTP1	Credential Compromise	2	4	3	6	5
TTP2	Known POS System	2	1	1	1	1
TTP3	Hardcoded Filenames	1	5	3	2	1
TTP4	Changes registry as legitimate Firewall App	1	1	5		1
TTP5	Hooks WH_KEYBOARD_LL	1	1	1	5	1
TTP6	CreateToolHelp 31 Snapshot Method	1	2	2	5	1
TTP7	Use Email or SMTP	1	1	3	6	2

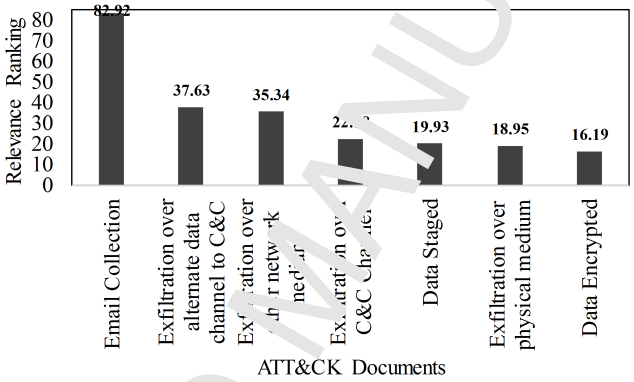

Figure 9: Search results for "use email or SMTP for exfiltration" TTP in LSI indexed ATT&CK documents

Based on the results it can be seen that most relevant detection mechanisms related to the mentioned TTP are present in ATT&CK document termed as "Email collection". The detection mechanisms from this document are enumerated as follows:

1. Check for processes utilizing network communication that should not require this function
2. Check for processes accessing network without user-driven events

6.2.2. TTP Network for Financial Malware

After retrieving semantically relevant financial malware related CTIR and ATT&CK documents, a TTP network is built shown in Table 4. The associated detection mechanism details are given in Table 5. Here five malware instances are considered, i.e., *RDASRV, Alina, Vskimmer, Dexter*, and *BlackPOS*. A belief network between threats and TTPs is formed by calculating posterior probabilities based on the incident frequency. The posterior and prior probabilities are shown in Table 6. For a predictor attribute, the financial malware class, TTP, that boasts the highest posterior probability is considered the prediction output. For instance, "Alina" has the highest occurrence probability for TTP3. The prior probabilities for the malware threat class, and predictor

attribute, i.e., TTP before seeing any real-world data are also given. It can be seen that "Dexter" has the highest prior probability which is also evident from historical data of financial malware incidents. Similarly, it can be seen that among all TTPs, TTP1 is the most prevalent TTP as its prior probability is high. The TTD network predicts the threat incident based on the detected threat symptoms provided in the form of host and network artifacts.

Table 4: Financial Malware Threat-TTP-Detection Semantic Network

Threat	TTP ID	Detection Mechanism
RDASRV, Alina, Vskimmer, Dexter	TTP1	d1
RDASRV	TTP2	d2, d3
Alina, Vskimmer, Dexter	TTP3	d4, d5, d6
Vskimmer	TTP4	d7, d8
Dexter	TTP5	d9
Alina, Vskimmer, Dexter	TTP6	d10
Vskimmer, Dexter, BlackPOS	TTP7	d11, d12

Table 5: Detection Mechanisms for Financial Malware TTP

Detection Mechanism	Description	Weight
d1	Look for suspicious account behavior	5
d2	Audit file system access logs for failed attempts	2
d3	Monitor process capturing the arguments	4
d4	Analyze file read, write and modify method for malicious activity	2
d5	Monitor anti-virus and IDS alerts	1
d6	Acquire IP and port	1
d7	Monitor changes in utility arguments and binaries	4
d8	Monitor command line invocations modifying registry	3
d9	Monitor registry for key stroke interception driver installation	3
d10	Monitor program for process enumeration	4
d11	Check for out of context network communication	5
d12	Check network access without user driven event	5

Table 6: Financial Malware Threat-TTP Posterior and Prior Probabilities

TTP ID	RDASPV	Alina	Vskimmer	Dexter	BlackPOS	Prior (TTP)
TTP1	0.1	0.2	0.15	0.3	0.25	0.26
TTP2	0.33	0.17	0.17	0.17	0.17	0.08
TTP3	0.08	0.42	0.25	0.12	0.08	0.15
TTP4	0.11	0.11	0.56	0.11	0.11	0.12
TTP5	0.14	0.14	0.14	0.43	0.14	0.09
TTP6	0.09	0.18	0.18	0.45	0.09	0.14
TTP7	0.08	0.08	0.23	0.46	0.15	0.17
Prior (Class)	0.12	0.19	0.23	0.31	0.15	

6.2.3. Financial Malware Threat Prediction

After generating the TTD network, the process of cyber threat investigation starts with threat prediction. This happens when the network administrator observes certain artifacts. The artifacts, that can be seen, are low-level indicators such as IPs, CnC servers, and ports or it can be unseen high-level indicators such as TTPs. These indicators are either obtained directly or through certain detection mechanisms. Suppose the network administrator observes a suspicious account behavior, changes in registry with the installation of a driver intercepting keystrokes and a process accessing the network without any user-driven event. Based on the derived TTD semantic network, the detection mechanisms selected are d1, d9, and d12. These detection mechanisms are mapped to TTP1, TTP5, and TTP7 which become our detected TTPs i.e., TTPD. Based on the TTPD, the belief network predicts the probability of occurrence for each malware instance as shown in figure 10. It can be seen that "BlackPOS" has the highest probability of occurrence. The second highest probability is that of "Dexter". Now we have to select a threat prediction set P'. A problem that arises concerns determining exactly which malware instances should be included in the prediction set P'. For this purpose, a threshold must be defined based on expert opinion e.g., suppose for this example we set threshold as above as 20%. Thus the prediction set comprises two threat instances, i.e., "BlackPOS and Dexter". Setting a high threshold leads to a more reliable threat prediction as a low threshold compromises the reliability of the threat prediction.

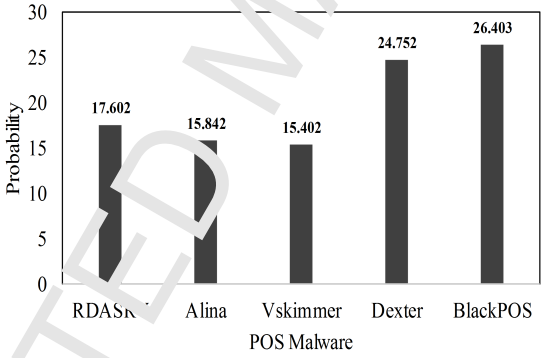

Figure 10 Probability Prediction for Financial Malware Instances

We assess the reliability of the prediction using equation 4 given in the design section. In this case, the total number of TTPs for the BlackPOS malware are "2" and that for Dexter are "5". The detected TTPs, i.e., TTPD for BlackPOS are "2" and that for Dexter are "3". There are two malware in the threat prediction set P'. After inserting the values in equation 4, we get equation 5

$$RA(P') = \frac{\sum_{t_i \in p_i}(2/2) + (3/5)}{2} \tag{5}$$

Thus the reliability of this prediction set becomes 80% which is quite high. This shows that the prediction is reliable. The reliability of threat prediction increases more if we increase the threshold for threat selection in the prediction set, e.g., we include

all those malware in the threat prediction set who has a prediction greater than 25%, then there will be only one malware selected, i.e., BlackPOS. The reliability for this prediction set is a 100% greater. On the other hand, if the threshold for selecting threats in the prediction set is decreased the reliability also decreases, e.g., we include all those malware in the prediction set that have a prediction greater than 10%, then there will be three malware selected, i.e., BlackPOS, Dexter, and RDASRV. The reliability of this prediction set is 70% less than the previous prediction set.

Suppose that the reliability assessment of the threat prediction is low. Then the undetected TTPs of the malware in the prediction set, i.e., TTPM must be considered. To check the existence of TTPM, the security administrator performs their associated detection mechanisms. This may be hard to accomplish if the number of TTPs in TTPM is large. Thus, there is a need to find the least cost detection mechanisms that can verify the undetected TTPs, i.e., TTPM of the threat prediction with a high reliability. One solution is to let the security administrator decide which detection mechanisms should be performed based on his knowledge, expertise, environment and the resources. In the capacity of this research work, we have assigned weights to actions based on the security expert opinion. The complexity of detection mechanism selection increases as the size of TTPM grows large. Therefore, detection mechanisms are assigned weights based on their complexity as shown in Table 5. The weights lie in the range [1-5]. The detection mechanisms with a weight closer to '0' are considered less costly than those with a weight near to '5'.

For example, one of the malware's TTP is to use RC4 to encode the contents for exfiltration. The detection mechanisms related to this TTP are:

1. Use samples to acquire the key and the algorithm and employ the same for decrypting network traffic for determining the communications signatures for malware.
2. Inspect SSL/ TLS to determine if encrypted traffic contains the presence of a C&C or in other words a Command and Control channel.

It is clear that detection mechanisms for getting the encryption algorithm and its key is a more difficult process as compared to inspecting SSL and TLS. Thus the weight of SSL/ TLS inspection is low as compared to finding the encryption key.

7. Evaluation and Results

The effectiveness and efficiency of all the segments of the data breach investigation framework were evaluated.

The effectiveness of SIRS was evaluated using the financial malware TTP dataset compiled in the case study using the parameters f-measure, recall and precision. The effectiveness of the TTD network was evaluated by constructing a benchmark training dataset from the threat, TTP and detection mechanisms data available in the ATT&CK taxonomy [21]. This benchmark data serves as a ground truth regarding cyber threats associated with different threat families for training our model by building a TTD network and provides a baseline for future comparisons. The TTD network is evaluated using a hold-out means validation method [63]. The evaluation parameters, in this case, are the accuracy of detection and the false discovery rate. The accuracy of the threat detection process was monitored for two kinds of situations. The first situation was when certain

threat artifacts were lost. The second situation was when certain spurious threat artifacts were added to poison the threat investigation process. The effectiveness of the cyber threat prediction segment was evaluated through the factors that impact the reliability of the prediction. These factors include varying levels of probability thresholds of the prediction set and the number of detected artifacts and types of TTPs (either disjoint or overlapping) controlled by attack type selection. Whereby having the same attack family provides more overlapping TTPs and using data about more recent attacks provides disjoint TTPs.

The efficiency was evaluated based on the time taken to detect an attack for different dataset sizes and threat records. The TTP coverage for the threat incidents of the ATT&CK dataset and the capability of our prototype system in data augmentation is discussed in the light of well-known threat incidents related to particular threat actors reported in multiple sources. Finally, a feature based comparative analysis of the threat investigation framework was performed with existing intrusion and attack detection techniques to elaborate on the benefits of the proposed approach to the overall security measurement and management infrastructure.

7.1. Benchmark Dataset

To construct a benchmark dataset, ground truth data related to cyber threat incidents, encompassing the details of TTPs and detection mechanisms, was required. For this purpose, we conducted a thorough survey of literature available on standardized cyber threat taxonomies. The three most well-known are the Open Threat Taxonomy [64], the Enisa Threat Taxonomy [65] and the ATT&CK Taxonomy [21]. Among these, we found that the Open Threat and Enisa did not serve our purpose because they simply list threats along with their descriptions but do not distinguish the TTPs and detection mechanisms. On the other hand, ATT&CK lists threat incidents and connects them to the TTPs and detection mechanisms, and is, therefore, the closest to the requirements of our training model.

Table 7: ATT&CK Taxonomy Dataset Statistics [21]

TTP Classes	Documents	Threat Actors	Documents	Software Tools	Documents
Persistence	28	Chinese	17	Malware	65
Privilage Escalation	1[illegible]	Russian	3	Backdoor	37
Defense Evasion	33	Iranian	2	Bot	3
Credential Access	8	Indian	2	Web Shells	3
Discovery	16	Portuguese	1	Toolkit, Bootkit	4
Lateral Movement	14	Unattributed	20	Rootkit	4
Execution	1[illegible]			Credential Dumper	4
Collection	11			Credential Harvester	1
Exfiltration	9			Remote Access Tools	9
Command and Control	17			System Utilities	16

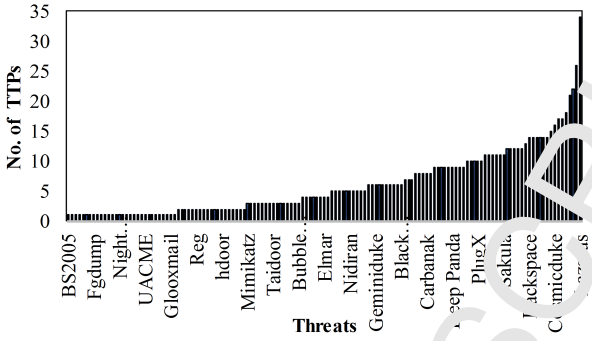

Figure 11: TTP distribution for Threat Incident

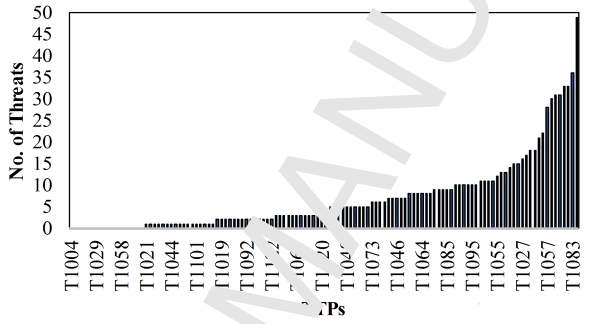

Figure 12: Threat incident distribution for ATT&CK TTPs

To establish the reliability of the ATT&CK data we investigated its sources. "ATT&CK" is created and maintained by MITRE Corporation [21]. It covers adversary's TTPs encountered in pre-attack as well as post-attack stages proposed by Lockheed Martin [66].

At the time of writing, there exist 133 adversary's technique documents classified under 10 tactic classes. A technique may belong to more than one class. We have compiled some key statistics for the interest of the reader in Table 7. Each technique document has a short description along with its mitigation, detection mechanism, and threat incident example. Threat incident examples are either threat actors or software tools used by them. At the time of writing, there are 45 threat actors and 123 software tools in the ATT&CK taxonomy. Figure 11 shows the TTP distribution of threat incidents. The minimum number of TTPs encountered in a threat incident is 1 and the maximum is 34. An average TTP count for each threat incident is 6. Similarly, figure 12 shows the other side of the picture, i.e., threat distribution for TTPs. The range of TTPs lies between 0 and 49. An average threat incident count for each TTP is 7.

The ATT&CK TTP taxonomy at any snapshot in time does not always encompass all TTPs of any given threat mainly because the website is constantly being updated as more information about a threat is uncovered. One can observe cases of threats where connections are obviously missing between threat incidents and TTPs. For instance, consider the case of threat incidents caused by a hacking group known as Axiom or

Hidden Lynx [67] extracted from ATT&CK. This group offers hacker hiring services and is known to have launched highly targeted campaigns against diverse business domains including multiple sectors and can run multiple campaigns simultaneously. It is observed that only three TTPs are listed for this group in the ATT&CK repository as shown in Table 8 while there are several more TTPs that can be related to this group obtained from other sources such as IBM X-Force STIX reports and Symantec blog reports. A comprehensive list of Axiom group's TTPs obtained through the CTRS segment of the threat diagnosis framework is shown in Table 8. This also illustrates that an added contribution of our framework is that it is capable of *data augmentation*. Meaning it can connect threats to TTPs from different data sources in the case of missing connections and provides a comprehensive TTP coverage for each threat incident.

Table 8: Axiom hacker group TTP Coverage by ATT&CK, IBM X-Force and Symantec

TTP	ATT&CK	IBM X-Force	Symantec
Zero-day exploits in Internet Explorer 10 to transfer malware		✓	✓
Remote Access Trojan		✓	✓
Credential Dumping	✓		✓
Accessibility using RDP	✓	✓	✓
Supply Chain Attacks		✓	✓
Strategic Web Compromise through waterhole attack		✓	✓
Deposits malware signed with stolen Bit9 keys		✓	✓
Payload Obfuscated using XOR encoding	✓	✓	✓
Direct injection of attack code into the Internet Explorer process without writing it to disk		✓	✓
The User-Agent header for CnC communication is "lynx"		✓	✓
CnC traffic as HTTP POST and non-HTTP protocol traffic		✓	✓
Sometimes uses HTTPS (port 443)		✓	✓
SQL Injection attack on an Internet-facing Web server		✓	✓
Oracle Java SE Runtime Environment Hotspot code execution		✓	✓

The ATT&CK taxonomy has been used as a baseline to model adversary's attack patterns in multiple research works. In [68], the ATT&CK taxonomy is used as a baseline to collect historical knowledge of the threat agent for automating the comparison of post-compromise actions of adversaries. Similarly, in [69], the ATT&CK taxonomy is used to design and implement an automated red teaming system by employing TTPs to drive the atomic actions of an intelligent decision-making system. To aggregate analysis of the enterprise's log data, [70] uses the ATT&CK taxonomy to model malicious threat scenarios based on adversary's attack patterns. It helps select useful audit log data from massive raw logs. In [71], a visual analytic tool that integrates the analytics of a cyber attack with the data required by the security analyst is proposed. The analytics of cyber attacks are taken from ATT&CK taxonomy which is used to establish an appropriate baseline against which to compare anomalies.

As such, ATT&CK serves as the best candidate to construct our required training dataset. We therefore, extracted all the relevant threats, TTPs and artifacts from the online ATT&CK and constructed a dataset, a small excerpt of which is shown in table 9. The highlighted data fields represent the presence of the TTP in a threat. The data

fields are normalized to remove zero entries. A TTD network is then built between threat incidents, TTPs and detection mechanisms.

Table 9: An excerpt of constructed ATT&CK MITRE benchmark dataset

TTP ID / **TTP Title**	T1001 Data Obfuscation	T1002 Data Compressed	T1003 Credential Dumping	T1004 Winlogon Helper DLL	T1005 Data from Local System	T1006 File System Logical Offset	**Cyber Threat Class**
	1	0	1	0	0	0	Axiom
	1	0	1	0	1	0	Patchwork
	1	1	0	0	0	0	Duqu
	1	0	0	0	0	0	S-Type
	1	0	1	0	0	0	Oldrea
	1	1	1	0	0	0	Prikorma
	0	1	1	0	0	0	Ke3chang
	0	1	1	0	1	0	APT2
	0	1	1	0	1	0	TG-3390
	0	1	0	0	0	0	Lazarus
	0	1	1	0	0	0	Fin6

7.2. Effectiveness of SIRS

The effectiveness of the SIRS module is evaluated with financial malware TTP dataset and ATT&CK Taxonomy. A TTP dictionary for 11 different kinds of malware is created using threat reports of [60], [24], [15] and [13]. The TTPs related to different phases of cyber kill chain are given in Table 2. These TTPs are stored in the TTP dictionary and are queried against the indexed malware and ATT&CK documents. The effectiveness of the SIRS is evaluated by measuring precision and recall for each TTP in the dictionary. Precision measures the number of correctly retrieved documents shown in equation 6. While recall measures the number of correct items collected by the retrieval system given in equation 7[72].

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{6}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{7}$$

The average of precision and recall for all TTPs is calculated. A weighted harmonic mean of both the parameters, measured using F-measure, is shown in equation 8[72].

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

The results related to the TTPs of each phase of the cyber kill chain are shown in Table 10. It can be seen that the overall effectiveness of the SIRS for financial malware documents is quite good with an average F-measure of 0.94.

Table 10: Effectiveness evaluation results of SIRS

	Precision		Recall		F-measure	
TTP Queries (Kill Chain Phases)	POS Malware	ATT&CK	POS Malware	ATT&CK	POS Malware	ATT&CK
Reconaissance	95%	22%	100%	33%	0.97	0.26
Weaponize and Deliver	94%	48%	87%	46%	0.90	0.47
Exploit	97%	50%	98%	27%	0.97	0.35
Install	93%	85%	96%	80%	0.94	0.82
Command and Control	95%	90%	90%	84%	0.92	0.87
Actions and Objectives	94%	82%	89%	78%	0.91	0.79

7.3. Effectiveness of TTD Network

To check the sensitivity of methods towards data, we evaluated the *TTD Semantic Network* using the hold-out cross validation method [63]. In this method, the given dataset is split into separate parts, i.e., a training set and a testing set. In our case, we are using the threat-TTP ground truth data provided by ATT&CK MITRE to train our cyber threat prediction model. For the test set, we examined a set of recent cyber threat incidents listed in ATT&CK MITRE and we determined their threat actors. We searched and downloaded more recent attacks of these threat actors from four different well-known cyber threat sources, i.e., IBM X-Force [5], Symantec [23], FireEye [24] and CrowdStrike [25]. These new threat incidents thereafter comprised our test set. These threat sources were chosen so that they differ to some extent in the level of abstraction, the vocabulary used and the level of specialization in the type of threat. Out of these, IBM X-Force supports STIX encoded CTIR, while for the rest the reports are encoded in STIX before importing them into the TTP dictionary.

We experimented with several machine learning algorithms for determining the best prediction accuracy. The results are shown in table 11. The results depict that the belief network outperforms the other techniques which is clearly suggestive that it is more suitable to our problem. It is easy to see why. When a data breach occurs, security analysts analyze them from multiple aspects. For recently occurred incidents the data is likely to contain conflicting or missing feature values as various vendors begin to study it. For data breaches that have been well studied the features are likely to be well established and complete. Our dataset has a mix of both new and well studied data breaches. Therefore, we required a classifier that can reliably predict a cyber threat family in the presence of erroneous or missing features. Furthermore, we observed that the belief network is less sensitive to irrelevant features (i.e., noisy data) that an adversary can intentionally insert to poison the cyber threat prediction. Finally, a belief network considers all the features independently and trains in linear time as compared to other iterative expensive approximation approaches. We evaluated the accuracy of the TTD network for each threat source separately. As a result, we observed a varying accuracy between the ranges of 53%-100% based on the cyber threat source chosen as the test subject as shown in table 11. The results demonstrate the sensitivity of the TTD semantic network to the amount of TTPs that exist in the test data, which is an essential requirement of our model to achieve accurate classification. It is to note

that our results are not a reflection of the data quality of these threat sources as they are based on a very selective CTIR test set and their use of the TTP labels defined in our ATT&CK training dataset. Not all cyber threat sources follow the same TTP vocabulary because of the reasons quoted above. The TTP statistics for cyber threats in the training and test datasets are shown in figure 13. It can be seen that since the IBM X-force test dataset has the highest number of TTPs, it also has the highest cyber threat prediction accuracy (100%). The Fireeye test dataset has the next highest number of TTPs, thus the prediction accuracy is slightly lower (95%), followed by CrowdStrike and Symantec that are at respectively 67% and 53%. Also it is worth noting that even when there is a misclassification due to missing TTPs the model picks the next relevant threat, e.g., in one case "Carbanak", was misclassified as "GCMAN" both of which target banks for financial fraud.

Table 11: Accuracy of the TTD Semantic Network for test data compiled from different threat sources

	IBM X-Force	Symantec	FireEye	CrowdStrike
Belief Network	100%	53%	95%	67%
Decision Tree	30%	18%	33%	33%
Random Forest	45%	35%	57%	33%
Deep Learning	20%	12%	14%	17%
SVM	60%	0%	85%	67%

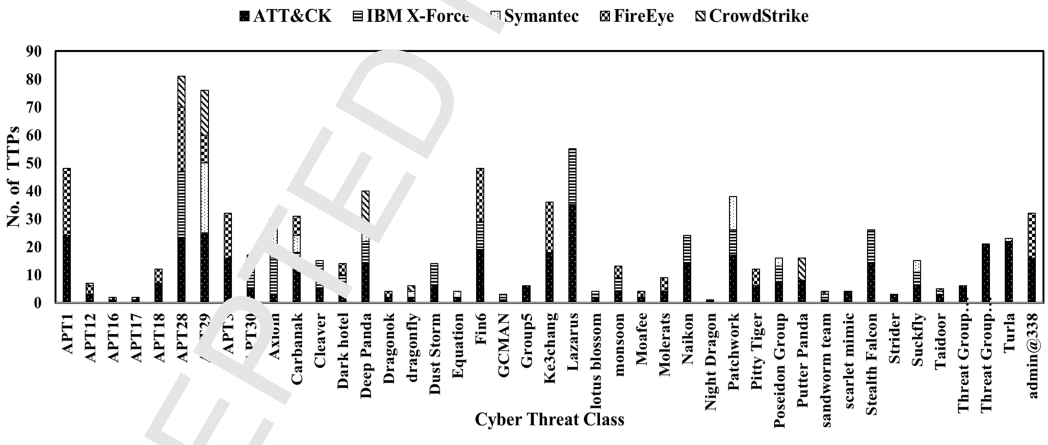

Figure 13: TTP statistics for the under consideration cyber threat sources

Further, we evaluated the accuracy of the *TTD Semantic Network* for two kinds of situations, i.e. lost TTP ratio and spurious TTP ratio. The results are shown in figure 14 and 15. In both situations, accuracy is compared with the three types of lost and spurious TTP ratios, i.e., random, overlap and disjoint. The random TTP ratio represents an unbiased approach of attack injection encountering both the best case and the worst case of threat artifact identification. While the overlap and disjoint TTP ratios are biased

based on attack injections representing the worst case and best case of threat artifacts identification respectively. The worst case is the one in which the attacker's aim is to misguide the security analyst in threat identification. The overlap in the lost TTP ratio represents that the disjoint TTPs of the threat are intentionally removed or disguised to increase the number of false positives in the prediction results. While the disjoint lost TTP ratio represents that the overlapped TTPs of the threat are ideally removed or disguised to increase the detection accuracy in the presence of a minimum number of detected threat artifacts. Similarly, the overlap in spurious TTP ratio represents that the overlapping TTPs of other similar threat incidents are intentionally added to increase the number of false positives in the prediction results. While the disjoint spurious TTP ratio represents that the disjoint TTPs of the threat incident are ideally added to increase the detection accuracy in the presence of a maximum number of spuriously detected threat artifacts.

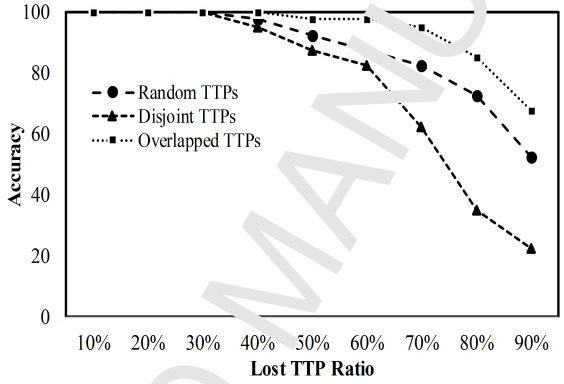

Figure 14: Accuracy Results for Lost TTP Scenario

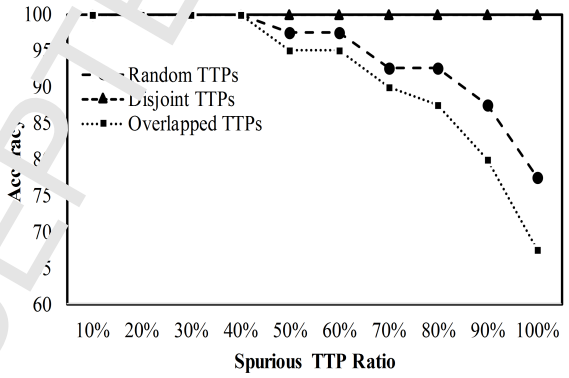

Figure 15: Accuracy Results for Spurious TTP Scenario

The results show that the overall accuracy for spurious TTPs is high as compared to lost TTPs. The average accuracy of a random unbiased situation in case of spurious

TTPs is 95% and in the case of lost TTPs is 87%. For an ideal situation where all the spurious TTPs are disjoint, the accuracy becomes 100%. While for the worst case when TTP overlap is high the average accuracy is 92%. Similarly, in the case of lost TTPs an ideal situation where the left over TTPs are disjoint the average accuracy becomes 94%. While for the worst case when the left over TTPs have high overlap, the average accuracy is nearly 76%.

Similarly, the ratio of false positive aka type I errors is measured by False Discovery Rate (FDR) for lost TTP and spurious TTP ratio [73]. FDR is defined in equation 9:

$$FDR = \frac{\sum FalsePositive}{\sum PositiveExperimentOutcome} \tag{9}$$

The results are shown in figure 16 and 17. In both situations, FDR is compared for the three types of lost and spurious TTP ratio, i.e., random, overlap and disjoint. The results show that the overall FDR of spurious TTPs is low as compared to lost TTPs. For an unbiased random situation, the average FDR for spurious TTPs is 6% while the average FDR for lost TTPs is 19%. For an ideal situation where all the spurious TTPs are disjoint, the FDR becomes 0. While for the worst case when TTP overlap is high the average FDR is 10%. Similarly, in the case of lost TTPs an ideal situation where the left over TTPs are disjoint the average FDR becomes 8%. While for the worst case when the left ones TTPs have high overlap, the average FDR is nearly 30%.

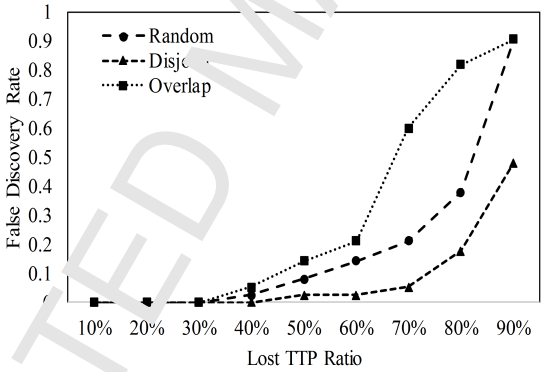

Figure 16: FDR Results for Lost TTP Scenario

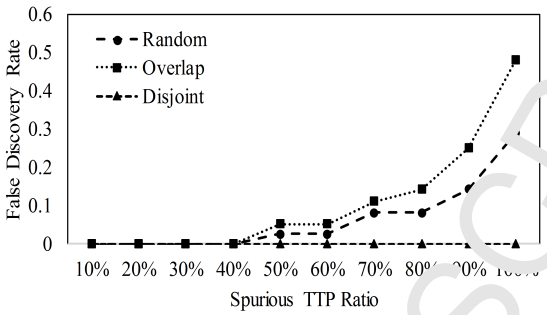

Figure 17: FDR Results for Spurious TTP Scenario

7.4. Effectiveness of Cyber Threat Prediction

The goal of the proposed threat investigation framework is to achieve a highly reliable threat prediction. During the experiments, there are certain factors observed that affect the prediction's reliability. These factors are the prediction selection probability threshold, the number of detected threat artifacts (TTPD) and the overlap between TTPs.

7.4.1. Threshold Impact on Prediction Reliability

Cyber threat investigation results in threat prediction by assigning an appropriate probability value to the threat incident. Among those threat incidents, a prediction set above a certain probability threshold needs to be selected. For this purpose, we analyzed the impact of increasing and decreasing probability threshold values over the prediction reliability. We evaluated prediction reliability using the probability thresholds: 25%, 50%, 75% and 100%. In figure 18, an example of one such scenario is given. It can be seen that according to the observed TTPs, Decebal malware has the highest probability, i.e., 10.93% that becomes 100% after normalizing the probability using equation 10.

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} * 100 \tag{10}$$

If the system has to select only one threat incident with the highest probability based on the observed TTPs, then the Decebal malware will be selected. However, in our case, the framework selects a set of threat incidents above a certain probability threshold. In figure 19, the prediction reliability for 10 different threat scenarios is shown. The series represents the threat scenarios. The results show that when we increase the threshold value the prediction reliability increases but there were certain scenarios where no attack classes were identified above 90%. Therefore, there was a need to select a threshold value that always generates a prediction set for all threat scenarios and also provides one with high reliability. Based on experimentation, this threshold value is selected as 75%.

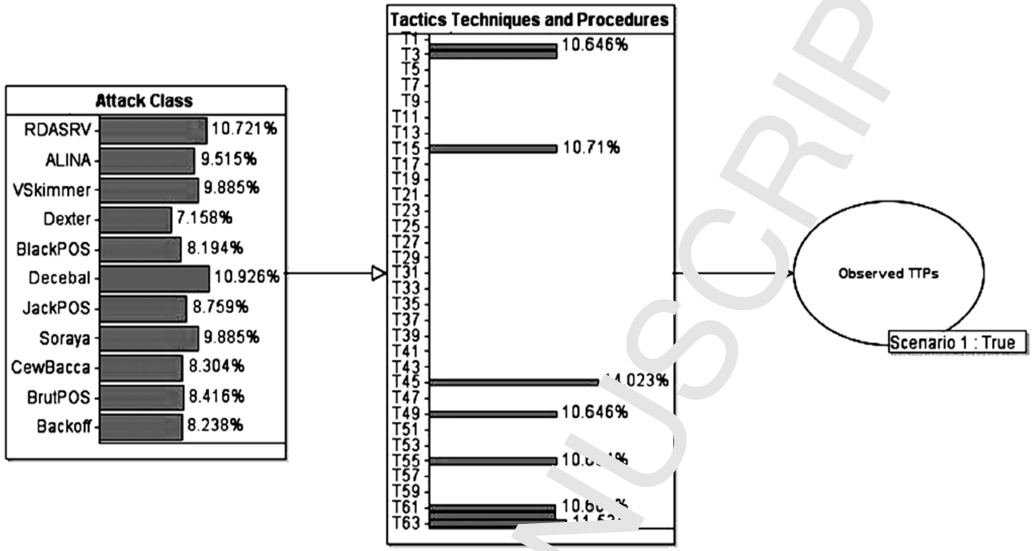

Figure 18: An example of cyber threat prediction

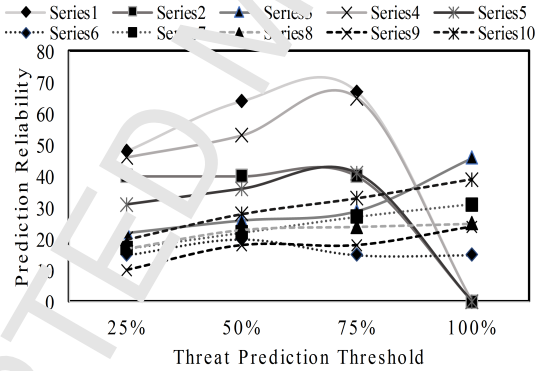

Figure 19: Impact of increasing and decreasing Probability threshold value over Prediction Reliability

7.4.2. TTPD Impact on Prediction Reliability

The number of detected TTPs, i.e., TTPD also has a great impact on prediction reliability. During experiments, it is observed that the prediction reliability is high if there are more detected TTPs related to the threat. In figure 20, the prediction reliability and TTPD ratio are shown. The graph shows that the reliability of the prediction is above 50% if there exists TTPD 50%.

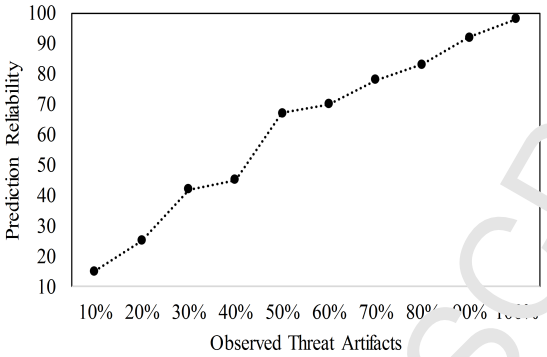

Figure 20: The impact of number of detected artifacts, i.e. TTPD over reliability

7.4.3. Disjoint and Overlapping TTPs Impact on Prediction Reliability

There are certain TTPs that are more common than others, e.g., the *data obfuscation* TTP is more frequently used than *routing command and control traffic over a non-standard custom port bypassing security tools* [74]. Hence, a significant overlap is usually observed across multiple threats for these common TTPs. We will refer to TTPs that do not occur across different threats in our threat prediction set as disjoint. This particular section details the effect of overlapping as well as disjoint TTPs on the threat prediction reliability. A subset of the dataset is divided into two classes, where one class has threat incidents having mostly disjoint TTPs and the other class has threat incidents having mostly overlapping TTPs. It is observed that the prediction reliability for threat incidents having disjoint TTPs is about 20% higher than the threat incidents having overlapping TTPs. The reason behind the fact is disjoint TTPs depict the distinguishing features of a threat incident that help a threat to be identified quite easily and with a high reliability as compared to the overlapping TTPs that share less distinguishing features. Figure 21 illustrates the results.

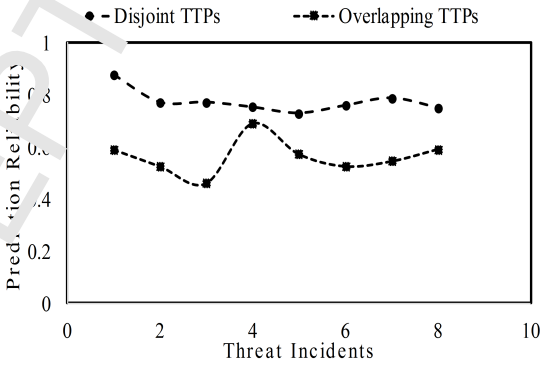

Figure 21: The impact of disjoint and Overlapping TTPs on Prediction Reliability

7.5. Efficiency of Data Breach Investigation Framework

The TTD semantic network is constructed offline or as a background process so we are not interested in the time taken in training. However, to study the impact of the dataset size on the time for threat detection, we performed experiments with different dataset sizes. We performed our experiments on a PC with Intel Core i5-4210U processors and 1.7 GHz and 2.4 GHz processing speed, 4 GB of RAM running a 64 bit Windows 10. The results can be seen in figure 22. We observed the threat detection time by partitioning the dataset into four equal parts. The observations are recorded for 25%, 50%, 75% and 100% instances of the dataset. Then for each dataset division, the number of observed threat artifacts were incrementally increased. The division of observed threat artifacts ranged from 10% to 100%. The results show that the size of the dataset does not have a significant impact on the threat detection time. The factor that matters most is the threat detection time provided the threat artifacts are fed into the threat investigation framework. The experimental results in figure 23 for threat prediction time with increasing number of threat artifacts show that our system can predict threat incidents for 6 detected TTPs in an average time of 0.04 seconds. The results do not show any significant difference in the detection time when the number of TTPD increases. The detection time for 34 TTPD is approximately 0.5 seconds and the average detection time is 0.15 seconds. Considering that data breach incidents remain undetected for months or even years [15] and attacks such as DDOS last for 9-12 hours [75], the running time of our system is quite practical. Moreover, we can claim that the threat detection time of our proposed data breach investigation framework is low as compared to existing threat detection mechanisms.

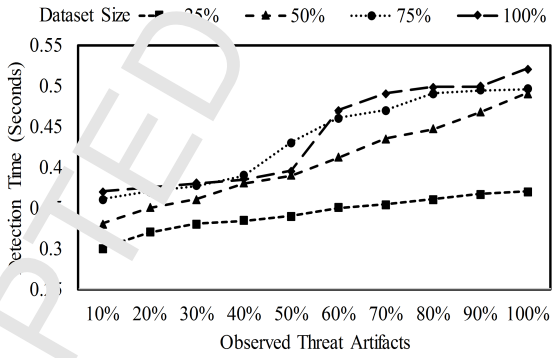

Figure 22: Impact of TTPD size on Threat Detection Time

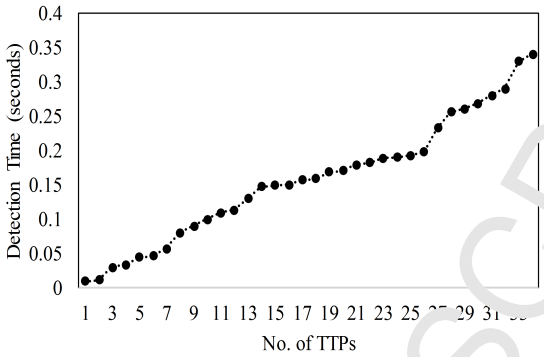

Figure 23: Impact of Dataset size on Threat Detection Time

7.6. Feature Based Comparative Analysis

The performance of the proposed data breach investigation framework cannot be directly compared with any existing threat detection mechanism due to its novelty. However, a comparative feature analysis was conducted. The existing threat detection approaches fall under (1) signature-based, (2) anomaly (behavior) based and (3) stateful protocol analysis [76]. In Table 12, a feature based detailed comparison of each approach is given.

Table 12: Feature based comparative analysis of threat diagnosis framework with existing threat detection techniques

Features	Signature	Anomaly	Stateful Protocol	Proposed Framework
Fixed Signatures	✓	✓	✓	
Dynamic Signatures		✓	✓	
Known Attack	✓	✓	✓	✓
Unforeseen Attack		✓		✓
Semantic Mapping				✓
Statistical Prediction		✓		✓

The signature-based threat detection approaches work with fixed format signatures that need to be regularly updated to cope with new threats. They can efficiently detect known attacks provided the same previously known signatures are used for future attacks. As discussed earlier, the signatures are susceptible to change and have a very short life, thus signature based approaches cannot detect unknown attacks or the same family of attacks using different signatures. On the converse, anomaly-based detection approaches are heuristic or rule-based and are capable of detecting new threats by classifying behaviors as normal or abnormal using machine learning and artificial intelligence techniques. They use a fixed format in the sense that the threat data sources are fixed patterned. Also, the false alarm rate is high due to weak normal and abnormal behavior profile of the attack traffic. The alarm system alerts an abnormal activity without providing further contextual information to guide the security analyst towards further threat investigations. The stateful protocol analysis is the specification based approach that identifies

abnormal sequences of commands in protocol functional trace. This technique falsely classifies threats that have the correct protocol execution sequence as benign. The existing threat detection approaches are not capable of mapping the detected network and host artifacts to the adversary's TTPs. The proposed data breach investigation framework works with both semantic and statistical procedures to investigate threat incidents. It does not follow any fixed pattern for threat detection rather it maps semantically the detected network and threat artifacts to the adversary's attack behavior quoted as TTPs in the structured CTIR and unstructured security documents. The mapping helps in identifying both known and unforeseen attacks. The framework either classifies the unforeseen attacks to an existing attack family or suggests further investigations to the security analyst to help discover the root cause of the underlying threat activity.

8. Conclusions

In this paper, a novel data breach investigation framework is presented that investigates cyber threat incidents using high-level adversary's TTPs from cyber threat intelligence documents. At the core of the framework lies the TTD semantic network that is based on the idea of semantically mapping low-level threat artifacts to high-level adversarial attack techniques and employs them to reason about an incident occurrence. The methodology and results presented highlight that security incidents can be mapped to tactics, that are further mapped to artifacts in a way that machines can identify these connections with certain probabilities. Henceforth this research paves the way for cyber security investigations with partial or incomplete information. The system extracts TTPs from CTIR and employs them to find semantic relevance (using LSI) between the threats and the detection mechanisms. The TTD semantic network is enriched by training a belief network that maps threats to TTPs to predict the most probable set of threat incidents based on the detected threat artifacts provided by the security analyst. In the case of an unforeseen threat, the system helps the security analyst investigate the threat artifacts against the most probable attack family by suggesting the most optimized and cost-effective detection mechanisms. This mapping of the threat artifacts to the TTPs was able to detect the threat incidents with high accuracy and low false positives in the case of lost and spurious detected threat artifacts. The TTD semantic network is extensible as more threat documents arrive they can be indexed and ranked using LSI. Similarly, the detection mechanisms can also be dynamically updated as the ATT&CK taxonomy updates. The evaluation results demonstrate that the detection time of the framework is quite low as compared to the considerable time it typically takes to investigate data breach incidents. In the future, we will integrate and automate the mitigations for the identified threat incidents.

References

[1] Gemalto releases findings of 2016 breach level index, https://www.gemalto.com/press/pages/gemalto-releases-findings-of-2016-breach-level-index.aspx, accessed: 2018-3-12.

[2] First half 2017 breach level index report: Identity theft and poor internal security practices take a toll, https://www.gemalto.com/press/pages/first-half-2017-breach-level-index-report-identity-theft-and-poor-internal-security-practices-take-a-toll.aspx, accessed: 2018-3-12.

[3] Equifax data breach affected 2.4 million more consumers, https://www.consumerreports.org/credit-bureaus/equifax-data-breach-was-bigger-than-previously-reported, accessed: 2018-3-17.
[4] HAIL A TAXII, http://hailataxii.com/ (2016).
[5] IBM X-Force, IBM X-Force: Deep security research expertise and global threat intelligence for enhanced security solutions, http://www-03.ibm.com/security/xforce/ (2017).
[6] Verizon data breach investigations reports, http://www.verizonenterprise.com/verizon-insights-lab/dbir/ (2016).
[7] K. Kent, M. Souppaya, Guide to computer security log management, Tech. rep., National Institute of Standards and Technology (2006).
[8] Active DNS Project, https://www.activednsproject.org/ (2016)
[9] Publicly available PCAP files, http://www.netresec.com/?page=PcapFiles (2017).
[10] Enron Email Dataset, https://www.cs.cmu.edu/~enron/ (2015).
[11] USC/ISI ANT Datasets, https://ant.isi.edu/datasets/all.html (2017).
[12] Structured Threat Information eXpression (STIX) A structured language for cyber threat intelligence., http://stixproject.github.io/ (2017).
[13] G. Sanchez, R. Carbone, Case Study: Critical Controls that Sony Should Have Implemented, https://www.sans.org/reading-room/whitepapers/casestudies/case-study-critical-controls-sony-implemented-36022 (2015).
[14] T. Radichel, Case Study: Critical Controls that Could Have Prevented Target Breach, https://www.sans.org/reading-room/whitepapers/casestudies/case-study-critical-controls-prevented-target-breach-35412 (2014).
[15] B. Hawkins, Case Study: The Home Depot Data Breach, https://www.sans.org/reading-room/whitepapers/casestudies/case-study-home-depot-data-breach-36367 (2015).
[16] D. Bianco, The Pyramid of Pain, http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html (2014).
[17] W. Tounsi, H. Rais, A survey on technical threat intelligence in the age of sophisticated cyber attacks, Computers & Security (2017).
[18] Threat group-1314, tg-1314, https://attack.mitre.org/wiki/Group/G0028 (2016).
[19] Threat group-3390, tg-3390, emissary panda, https://attack.mitre.org/wiki/Group/G0027 (2016).
[20] SecureWorks Counter Threat Unit Threat Intelligence, Threat group-4127 targets google accounts, https://www.secureworks.com/research/threat-group-4127-targets-google-accounts (2016).
[21] Adversarial Tactics, Techniques and Common Knowledge, https://attack.mitre.org/wiki/Main_Page (2016).
[22] H. Jeffreys, Scientific Inference, Third Edition, Cambridge University Press, England, 1974.
[23] Symantec, Symantec Software Company, https://www.symantec.com/ (2017).
[24] Fireeye: Cyber security & malware protection, https://www.fireeye.com/ (2016).
[25] Crowdstrike, https://www.crowdstrike.com/ (2016).
[26] M. T. Khorshed, A. S. Ali, S. A. Wasimi, A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing, Future Generation computer systems 28 (6) (2012) 833–851.
[27] G. Lodi, L. Aniello, G. A. Di Luna, R. Baldoni, An event-based platform for collaborative threats detection and monitoring, Information Systems 39 (2014) 175–195.
[28] G. Gonzalez-Granadillo, S. Dubus, A. Motzek, J. Garcia-Alfaro, E. Alvarez, M. Merialdo, S. Papillon, H. Debar, Dynamic risk management response system to handle cyber threats, Future Generation Computer Systems 83 (2018) 535–552.
[29] T. Wang, J. Zhou, M. Huang, M. Z. A. Bhuiyan, A. Liu, W. Xu, M. Xie, Fog-based storage technology to fight with cyber threat, Future Generation Computer Systems 83 (2018) 208–218.
[30] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, M. Liu, Cloudy with a chance of breach: Forecasting cyber security incidents, in: 24th USENIX Security Symposium, Washington, D.C., USENIX, 2015.
[31] A. Sarabi, P. Naghizadeh, Y. Liu, M. Liu, Prioritizing security spending: A quantitative analysis of risk distributions for different business profile, in: Annual Workshop on the Economics of Information Security, WEIS, 2015.
[32] A. Vasilyev, Bayesian Network Modeling for Analysis of Data Breach in a Bank, Master's thesis in Risk Management, http://hdl.handle.net/11250/183843 (2011).
[33] F. Jansen, A. Smulders, R. Kerkdijk, Cyber security information exchange to gain insight into the effects of cyber threats and incidents, e & i Elektrotechnik und Informationstechnik 132 (2) (2015)

106–112.
[34] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, B.-T. Chu, Data-driven analytics for cyber-threat intelligence and information sharing, Computers & Security 67 (2017) 35–58.
[35] F. Menges, G. Pernul, A comparative analysis of incident reporting formats, Computers & Security 73 (2018) 87–101.
[36] S. Appala, N. Cam-Winget, D. McGrew, J. Verma, An actionable threat intelligence system using a publish-subscribe communications model, in: Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, ACM, 2015, pp. 61–70.
[37] F. Skopik, G. Settanni, R. Fiedler, A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing, Computers & Security 60 (2016) 154–176.
[38] S. More, M. Matthews, A. Joshi, T. Finin, A knowledge-based approach to intrusion detection modeling, in: IEEE Symposium on Security and Privacy Workshops, San Francisco,CA, USA, IEEE, 2012, pp. 75–81.
[39] J. Undercoffer, A. Joshi, J. Pinkston, Modeling computer attacks: An ontology for intrusion detection, in: Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003, Pittsburgh, PA, USA, RAID, 2003, pp. 113–135.
[40] V. Mulwad, W. Li, A. Joshi, T. Finin, K. Viswanathan, Extracting information about security vulnerabilities from web text, in: Proceedings of the 2011 IEEE ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Workshops, WI-IAT 2011, Lyon, France, IEEE, 2011, pp. 257–260.
[41] Wikitology, http://ebiquity.umbc.edu/project/html/id/83/Wikitology#people (2016).
[42] A. Joshi, R. Lal, T. Finin, A. Joshi, Extracting cybersecurity related linked data from text, in: IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, IEEE, 2013, pp. 252–259.
[43] Dbpedia towards a public data infrastructure for a large, multilingual, semantic knowledge graph, wiki.dbpedia.org/ (2016).
[44] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, J. R. Goodall, Automatic labeling for entity extraction in cyber security, CoRR abs 1308.4941.
[45] C. Sabottke, O. Suciu, T. Dumitras, Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits, in: 24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, USENIX, 2015, pp. 1041–1056.
[46] Golden, R. Christian, Supporting Identity Risk Identification and Analysis Through Text Mining of News Stories., http://hdl.handle.net/2152/22743 (2013).
[47] M. Antunes, D. Gomes, R. L. Aguiar, Towards iot data classification through semantic features, Future Generation Computer Systems, Elsevier (2017).
[48] A. Oltramari, L. F. Cranor, R. J. Walls, P. McDaniel, Building an ontology of cyber security, in: Proceedings of the Ninth Conference on Semantic Technology for Intelligence, Defense, and Security, Fairfax VA, USA, CEUR, 2014, pp. 54–61.
[49] A. Oltramari, L. F. Cranor, R. J. Walls, P. McDaniel, Computational ontology of network operations, in: 34th IEEE Military Communications Conference, MILCOM 2015, Tampa, FL, USA, IEEE, 2015, pp. 318–323.
[50] C. J. D. Orazio, K.-K. R. Choo, Circumventing ios security mechanisms for apt forensic investigations: A security taxonomy for cloud apps, Future Generation Computer Systems, Elsevier 79 (2018) 247–261.
[51] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, J. Yearwood, Hybrids of support vector machine wrapper and filter based framework for malware detection, Future Generation Computer Systems 55 (2016) 376–390.
[52] Q. K. A. Mirza, I. Awan, M. Younas, Cloudintell: An intelligent malware detection system, Future Generation Computer Systems 86 (2018) 1042–1053.
[53] J. Abawajy, S. Huda, S. Sharmeen, M. M. Hassan, A. Almogren, Identifying cyber threats to mobile-iot applications in edge computing paradigm, Future Generation Computer Systems 89 (2018) 525–536.
[54] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, F. J. Aparicio-Navarro, Detection of advanced persistent threat using machine-learning correlation analysis, Future Generation Computer Systems 89 (2018) 349–359.
[55] D. Franklin, Backoff POS Malware, https://exchange.xforce.ibmcloud.com/collection/Backoff-POS-Malware-4170e5e758e50d29934ade2e19cd0c02 (2017).
[56] J. Wang, J. Peng, O. Liu, A classification approach for less popular webpages based on latent

semantic analysis and rough set model, Journal of Expert Systems with Applications Elsevier 42 (1) (2015) 642–648.

[57] L. Wang, S. Tasoulis, T. Roos, J. Kangasharju, Kvasir: Seamless integration of latent semantic analysis-based content provision into web browsing, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 251–254.

[58] P. J. Kwantes, N. Derbentseva, Q. Lam, O. Vartanian, H. H. Marmurek, Assessing the big five personality traits with latent semantic analysis, Journal of Personality and Individual Differences Elsevier 102 (2016) 229–233.

[59] D. Thorleuchter, D. Van den Poel, Improved multilevel security with latent semantic indexing, Journal of Expert Systems with Applications Elsevier 39 (18) (2012) 13462–13471.

[60] N. Huq, PoS RAM Scraper Malware Past, Present, and Future, http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-pos-ram-scraper-malware.pdf (2014).

[61] D. Sherry, How to prevent phishing attacks: User awareness and training, http://searchsecurity.techtarget.com/tip/How-to-prevent-phishing-attacks-User-awareness-and-training (2011).

[62] Microsoft, Windows management instrumentation, https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx (2017).

[63] J. Han, J. Pei, M. Kamber, Data mining: concepts and techniques, Elsevier, 2011.

[64] J. Tarala, K. Tarala, Open Threat Taxonomy version 1.1, http://www.auditscripts.com/free-resources/open-threat-taxonomy/ (2015).

[65] L. Marinos, ENISA Threat Taxonomy, https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information (2016).

[66] E. M. Hutchins, M. J. Cloppert, R. M. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, in: Sixth International Conference Information Warfare and Security (ICIW 11), ICIW, 2010, pp. 113–125.

[67] S. Doherty, J. Gegeny, B. Spasojevic, Hidden Lynx Professional Hackers for Hire, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapershidden_lynx.pdf (2013).

[68] S. Bromander, A. Jøsang, M. Eian, Semantic cyberthreat modelling, in: Semantic Technologies in Intelligence, Defense, and Security, 2016, pp. 74–78.

[69] A. Applebaum, D. Miller, B. Strom, C. Korban, R. Wolf, Intelligent, automated red team emulation, in: Proceedings of the 32nd Annual Conference on Computer Security Applications, ACM, 2016, pp. 363–373.

[70] I. Rose, N. Felts, A. George, E. Miller, I. Planck, Something is better than everything: A distributed approach to audit log anomaly detection, in: Cybersecurity Development (SecDev), IEEE, 2017, pp. 77–82.

[71] L. Franklin, M. Pirrung, L. Blaha, M. Dowling, M. Feng, Toward a visualization-supported workflow for cyber alert management using threat models and human-centered design, in: Symposium on Visualization for Cyber Security (VizSec), IEEE, 2017, pp. 1–8.

[72] B.-Y. Ricardo, Ribeiro-Neto, Berthier, Modern information retrieval, ACM press, New York, 1999.

[73] J. D. Storey, False discovery rate, in: International encyclopedia of statistical science, Springer, 2011, pp. 504–508.

[74] MITRE, Command and control, https://www.attack.mitre.org/wiki/Command_and_Control (2017).

[75] S. A. O'Brien, Widespread Cyberattack Takes Down Sites WorldWide, http://money.cnn.com/2016/10/21/technology/ddos-attack-popular-sites/ (2016).

[76] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, Intusion detection system: A comprehensive review, Journal of Network and Computer Applications Elsevier 36 (1) (2013) 16–24.

Biographical Sketch

Umara Noor received Gold Medal for her MS degree in Information Technology from Institute of Management Sciences, Pakistan in 2011. She is currently pursuing her PhD degree in Information Technology from National University of Sciences and Technology (NUST), Pakistan in the domain of machine learning approaches to automated analysis of cyber security threats under the research supervision of Dr. Zahid Anwar. She is also working as a faculty member in the Department of Computer Science and Software Engineering, International Islamic University (IIU), Pakistan. Her research interests include experimenting with machine learning techniques to enhance security systems and services.

Zahid Anwar received his Ph.D. and M.S. degrees in Computer Sciences in 2008 and 2005 respectively from the *University of Illinois at Urbana-Champaign*. Zahid has worked as a software engineer and researcher at *IBM*, *Intel*, *Motorola*, *National Center for Supercomputing Applications (NCSA), xFlow Research and CERN* on various projects related to information security, operating systems design and data analytics. Zahid holds post-doctorate experience from *Concordia University.* He has worked as a faculty member at the *National University of Sciences and Technology* and the *University of North Carolina at Charlotte.* He is currently an Assistant Professor at Fontbonne University.

Asad Waqar Malik is an Assistant Professor at NUST School of Electrical Engineering and Com- puter Science, Pakistan. He did his PhD from College of Electrical and Mechanical Engineering, NUST. He worked as a visiting scholar at Georgia Institute of Technology, and North Dakota State University, USA. He is the co-director of Center for Research in Modeling and Simulation (CRIMSON). His research interests include parallel and distributed simulation, cloud computing, Internet of Things (IoT) and large-scale networks.

Sharifullah Khan received his PhD in Computer Science from the University of Leeds, Leeds, UK in 2002. During 2005-2006, he completed his one-year PostDoc in the Universite Paul Sabatier (UPS), Toulouse, France, which was sponsored by CNRS France. Currently, he is serving as Associate Professor at NUST School of Electrical Engineering and Com- puter Science, Pakistan. He co-authored more than 60 papers, published in international journals and conferences. He leads the Data Engineering for Large Scale Applications (DELSA) research group at SEECS. Dr. Khan is conducting research activities in the areas of Data Science and Ontology Engineering, Information Retrieval.

Shahzad Saeem joined School of Electrical Engineering and Computer Sciences (SEECS-NUST) in 2015. He did his PhD. form Department of Computer and Systems Sciences at Stockholm University, Sweden. His research interests include Digital Forensics, Mobile Device Forensics, Forensics Tool Testing & Evaluation, NFC, TPM and Smart Cards. His current area of research focuses on preserving the integrity of digital evidence and protecting the basic

human rights during the process of digital forensics. The focus is to extend digital forensics on the abstract level to include preservation and protection as umbrella principles (?PasU).

Highlights

- A novel framework for cyber data breach investigation.
- Predicts cyber-threat with high accuracy.
- Employs high-level Tactics Techniques and Procedures for cyber threat investigation.
- A first of its kind framework that uses a comprehensive vocabulary for structured TTP analysis.