Threat Miner - A Text Analysis Engine for Threat Identification Using Dark Web Data

Nathan Deguara
Birmingham City University
Birmingham, UK

Junaid Arshad
Birmingham City University
Birmingham, UK

Anum Paracha
Birmingham City University
Birmingham, UK

Muhammad Ajmal Azad
Birmingham City University
Birmingham, UK

***Abstract*—Cyber threats continue to grow with novel methods to attack computing systems, highlighting the need for sophisticated mechanisms and techniques to protect against such dynamic threats. Contemporary cyber defence mechanisms utilise a range of methods which rely on monitoring network or system-level events. However, with the growing use of the dark web by mal-actors to share exploits, breaches, and data leaks, the use of such information to strengthen defence mechanisms becomes an intriguing prospect. In this paper, we present our efforts to develop a text mining engine (Threat Miner) which analyses data from dark web forums and transforms it into actionable intelligence. Leveraging cutting-edge machine learning techniques and utilising a bespoke threat dictionary, Threat Miner extracts useful information from dark web forums into STIX form, enabling it to be used with threat intelligence platforms. We also present the results of a thorough evaluation of our scheme which was conducted with the CrimeBB dataset [1] to understand the feasibility of the approach as well as its effectiveness in strengthening defence capability against cyber threats.**

***Index Terms*—Dark web, Cyber Threats, Threat Intelligence, Sentiment Analysis, Cyber Attacks**

I. Introduction

The volume and complexity of cyber threats continue to grow with novel methods and techniques employed by mal-actors to bypass protection mechanisms. The cyber security industry is indeed an example of continuous warfare between mal-actors and those looking to defend against them. The rapid increase in cyber attacks and the need to improve and develop intelligent security mechanisms are highlighted in [2]. Recent studies indicate that in the first six months of 2021 there was a 125% increase in cyber-related incidents [3]. Many of these attacks have origins in the dark web such as buying, selling or sharing malware source code [4].

The Dark web is the most hidden layer of the internet, also informally known as the *world of cyber crimes* used mostly for illicit activities and cyber-attacks. It is a collection of websites that can only be accessed using a browser, such as *Tor*, which provides anonymity and random routing that makes it feasible for criminal usage. Instead, each device connects through a series of nodes and relays and each device will meet at a set node where all communication will take place, ensuring several layers of protection and almost impossible to trace IP addresses [5].

Contemporary cyber defence mechanisms utilise a range of methods which rely on event monitoring at the network or system-level events to achieve desired protection. Using open-source intelligence (OSINT) for preparing and countering these threats is a vital part of many organisations' cyber security strategies. However, with the massive dark web use by mal-actors to share exploits, breaches, and data leaks and using such information to strengthen defence mechanisms becomes an interesting prospect. Specifically, the dark web provides an excellent source of intelligence about cyber attacks as it has numerous hacker forums which cyber criminals use for discussing cyber attack techniques and sharing inter-related malware. Carrying out monitoring and analysis of information shared on the dark web is something already being done by several different organisations such as various law enforcement agencies and it often relies on a mix of technology and analysts [6]. Finding this information and turning it into actionable intelligence that can be used by organisations to prevent a cyber attack is the core goal of this study which could help the security hardening of organisations against current and emerging attacks.

This paper is focused on investigating the challenge of extracting actionable intelligence from dark web forums which can be used to strengthen cyber defence mechanisms. The study builds on our existing research into investigating dark web forums to identify emerging trends and patterns which can provide a unique insight into the evolution of cyber threats. Specifically, part of our previous work [7] is focused on the challenge of identifying influential hackers on the dark web social networks. Leveraging influencer and hacker analytics, this paper presents our efforts to develop a text mining engine (Threat Miner) which uses semantic analysis to accurately identify new cyber threats from the most influential hackers within the CrimeBB dataset. The identification of these threats is then transformed into actionable intelligence in a format that could be used by a threat intelligence platform. Examples of such threats could be in the form of a new hacking technique, new malware, or a targeted cyber attack against a particular organization. This will allow organisations to understand and prepare for the emerging cyber threats identified by the Threat Miner framework.

The major contributions of this research are:

1) A mechanism to analyse information from non-trivial sources (the dark web) to strengthen cyber defence mechanisms
2) Implementation of a semantic analysis scheme to iden-

tify emerging cyber threats shared and discussed on the dark web forums. Threat Miner enables transforming intelligence extracted from dark web forums data in STIX format which facilitates collaborative defence through sharing of such intelligence.

3) Evaluation of the Threat Miner system to assess the feasibility of the approach as well as its effectiveness in strengthening defence capability against cyber threats.

The rest of the paper is organised as follows. The next section introduces fundamental concepts with respect to cyber threat intelligence and semantic analysis as used within organisational security. Section 3 presents a critical review of existing work relevant to this paper followed by details of the dataset used in Section 4. Section 5 presents the architecture of the Threat Miner system followed by its implementation in section 6. Section 7 presents an evaluation of the Threat Miner system followed by a discussion of conclusions and future work in section 8.

II. Threat Intelligence and Semantic Analysis

In this section, we present fundamental concepts within cyber threat intelligence (rationale, platforms, and sharing formats) and semantic analysis to aid understanding of the proposed threat miner framework.

A. Cyber Threat Intelligence

A cyber attack attempts to affect an organization's use of cyberspace with the aim of disrupting, disabling, destroying or maliciously controlling the computing environment, stealing information or destroying data integrity [8]. Cyber attacks can take many forms such as hacking, malware, denial of service and phishing [9]. The number and complexity of cyber threats facing organizations are evolving continuously. Most organizations have become the target of cyberattacks. Some of these are automated attacks while others are specifically chosen [10]. Whether the attacks originated from automated systems, random hackers or are part of an attack by advanced persistent threats (APTs), it is ever more important to have a strong cyber security strategy in place to defend against these threats. One of the key defenses is to have a strong cyber security policy in place [9]. It is also important to have defence in depth with mechanisms such as intrusion detection/prevention systems (IDS/IPS), network segregation and, anti-virus [11].

Open-source threat intelligence (OSINT) forms an important part of cyber threat intelligence (CTI) [12]. The goal of CTI is to research the developments in cyber-attacks and analyze the latest trends [13]. OSINT is the collection of information using publicly available data either from social media or internet websites [13]. Collecting and analyzing open-source intelligence is a complex process and often employs the use of machine learning techniques to efficiently analyze, filter and transform data into useful information [14].

There are several open source tools that exist to facilitate the collection of open-source intelligence such as MISP (Malware Information Sharing Platform), Echosec, Talc's Intelligence by Cisco. A comparison of these platforms can be seen in table I.

An important aspect of CTI is the ability to share such information across organisational systems and third parties. Structured threat information expression (STIX) is one such open-source CTI format that aims to make threat intelligence consistent and easy to share. There are nine key constructs including the indicators of compromise, types of exploits and how to respond [15]. STIX is written using a series XML files and Python scripts. STIX is specialized to characterize malicious activity using attack patterns [15]. Some other CTI formats are CybDX, Malware Attribute Enumeration and Characterization (MAEC) [16] and Trusted automated exchange of intelligence information (TAXII) [17]. Some OSINT platforms may accept more than one type of CTI format in which case it is important to use the one that best describes the cyber threat intelligence that has been gathered and is contained within the reports generated [15].

B. Semantic Analysis

Semantic analysis is the process of using natural language processing and text analysis to understand the meaning of information [18], [19]. There are many different algorithms that are used for semantic analysis. One of the most common algorithms used is latent semantic analysis (LSA) which is highly accurate to depict the context of text [20]. Another commonly used algorithm is the inverse document frequency (TF-IDF) which is an extremely useful algorithm to be used for recommendation systems [21]. Some other algorithms, useful for semantic analysis are support vector machine (SVM) and Naive Bayes [22].

A relatively new set of algorithms being used for semantic analysis is Word2Vec which is used to learn word embedding from data. The Word2Vec semantic analysis algorithms are very scalable and work well on both large and small datasets [23]. Word2Vec proved to be more efficient comparing TF-IDF in text classification and semantic analysis [24]. This was an important consideration for this study as the posts in the dataset varied in length from just a few words to large paragraphs of text. The Word2Vec algorithms are pre-trained using data gathered online [25]. Despite this, the Word2Vec model must still be trained on the particular dataset that is being used for analysis to ensure that it has the correct understanding of words in context and to both tokenize the words and calculate similarity scores for the words in the dataset being used [25]. Whereas semantic analysis as a whole aim to understand the meaning of the text, sentiment analysis aims to determine if the text has a positive, negative or neutral sentiment towards something [26].

III. Related Works

The dark web has been increasingly used in cyber security research for multiple objectives. For instance, Akyazi et al. [27] studied the criminal services provided by the hacking community. Using the CrimeBB dataset, the authors investigated the supply and demand for cybercrime services to

TABLE I
COMPARISON OF OPEN-SOURCE THREAT INTELLIGENCE PLATFORMS

MISP	***Echosec***	***Talos Intelligence***
Regular updates	Real-time data	Weekly updates
Community-run	Managed by Echosec team	Managed by Cisco team
Highly automated	Very large database	Updates sent to Cisco devices
CTI format - custom JSON files but has dependency on STIX, CybOX and MAEC	CTI format - not stated	CTI format - not stated
Free to use	Paid service	Free to use

the related community. Further, Cabrero-Holgueras & Pastrana [28] focused on the challenge of disparate user identities across different dark web forums to aid criminal investigations.

Ghaith Husari et al. [29] developed an automated mining system ActionMiner that takes input from open-source cyber threat intelligence systems and provides threat actions in response. This system uses publicly available threat intelligence sources including Wikipedia pages, MITRE ATT&CK framework reports etc. Still, this system lacks the verification strategy to verify the text authentication before transforming it into actionable items. Another issue in this system is that it works properly with the English text rather than the domain text of cybersecurity and these issues are rectified in our research study.

Wenzhuo Yang and K wok-Yan Lam [30] proposed an efficient approach to analyze open-source CTI data and transform it to be used by the Security Operation Center (SOC). Again, the described approach follows the open-source CTI data and does not use and verification mechanism to filter out the spam data before converting data into actionable objects. Secondly, this approach follows the internet sources which mostly highlight the details of threats after incidents. To cope up with the issue and provide proactive intelligence, our system targets the CrimeBB dataset which is compiled from the dark web.

A Framework is designed by Max Landauer and others [31] that pointed log files as the input source for the CTI extraction and provides patterns that detect the attacks and intrusions. The purpose of this framework is to gather the detectable patterns and secure other systems from the same attacks. The major problem identified in this system is that the proposed framework targets the logs for attack detection which works after the determination of the attacks at least once. This framework is designed to work for multi-attack scenarios and not proactively secure systems.

Lorenzo Neil et al. [32], pointed out the vulnerabilities in the open-source projects and libraries and extract the threatening bugs and vulnerabilities to be stored in a knowledge graph that can be further utilised for threat intelligence purposes. The study conducted by Lorenzo and others is specific to open-source projects and libraries. Comparing it to our research, we have pointed out the CrimeBB dataset that is developed by continuous crawling of data from the dark web forums. Our study leverages proactive threat identification and does not specify any particular domain.

IV. DATASET DESCRIPTION

The dataset used for this study was the CrimeBB dataset by [1] and provided by Cambridge Cyber Crime Lab. This is a large dataset with several subsets that have been gathered from dark web hacker forums. In its raw form, the dataset is large containing data about a range of different forums from the dark web.

A. Data Pre-processing

As part of our previous work [7], we have used CrimeBB dataset to perform intelligent analytics to identify emerging trends within cyber attacks. One such effort has been to identify influential hackers within these forums with the hypothesis that identifying influential hackers can lead to credible threat intelligence. In particular, we utilised different techniques such as Feature Engineering, Social Network Analysis, Text Mining, Semantic Analysis, and K-means clustering to derive an *influencer score* for each user which represents the social stature of a user within the dark web forum.

Table II provides the details of the input dataset used in this research. The dataset comprises 2 CSV files that are segregated in relational format. The file Author.csv comprises of the details regarding the authors and the Content.csv gives posted content details with respect to those authors.

The ID columns were all numerical in value, were unique to each user and did not contain the actual usernames. The posts contained string data from the forums and were not filtered. The reputation (as part of the forum metrics) was an integer and the influencer score (calculated as part of data pre-processing) was a floating-point number. Key statistics for the reputation of the users in the forums used are presented in Table III.

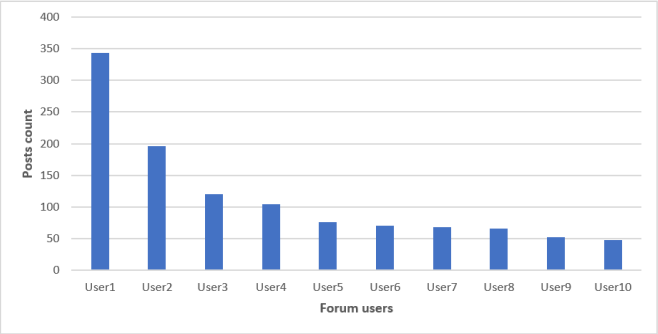

Fig. 1. Users with maximum posted contents

TABLE II
DATASET DESCRIPTION

Feature Name	*Data type*	*Related File*	*Description*
Id	Integer	Author.csv	Author Id generated by the INSPECT framework
Author Id	Integer	Author.csv	Author Id used in CrimeBB dataset
Reputation	Integer	Author.csv	Author's reputation provided by CrimeBB dataset
Influencing score	Float	Author.csv	Author's impact calculated by the INSPECT framework
Source	Integer	Content.csv	Author Id, who posts the content
Destination	Integer	Content.csv	User Id, whose post got the response
Post	String	Content.csv	Posted text by the respective author

TABLE III
KEY STATISTICS OF USER REPUTATION

Key Feature	*Value*
Lowest reputation	-1046
Highest reputation	3634
Average reputation	40
Most common reputation	1

In total there were 13,214 users and 17,719 posts in the dataset. An in-depth statistical analysis of the dataset is presented in Figure 1, 2 and 3. Figure 1 provides a detailed analysis of the users interaction and the activity statistics in the forum. For this research, we filtered users who posted the maximum content in the form of posts and replies to the posts in the social forum. The graphical interpretation presented in Fig 2 and 3 provide details regarding the popularity and impact of the users in the forum. The graphs represent the influence of the users calculated versus the reputation (individual ratings provided by co-users) provided by the CrimeBB dataset.

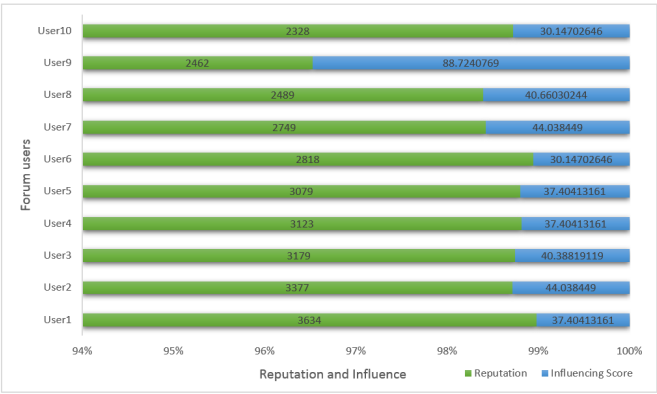

Fig. 2. Maximum user reputation and corresponding influence

B. Ethical Issues

The data-set is originally collected and cleansed by the CrimeBB [1]. They have already considered all the suitable measures to ensure the anonymity and security of users in the data set. In this research, we have not tried to de-anonymize the identity of the users.

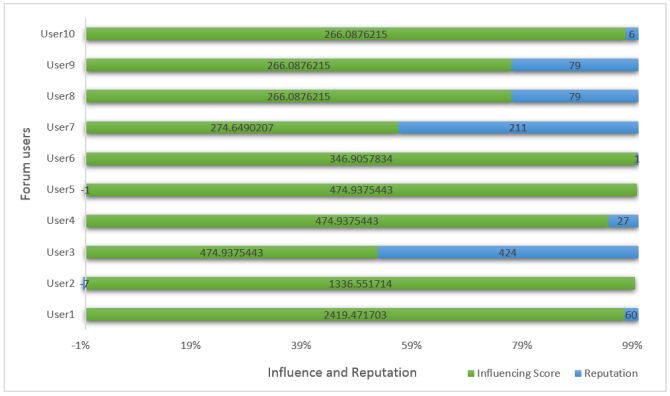

Fig. 3. Maximum user influence and corresponding reputation

V. THE THREAT MINER SYSTEM

In this section, we present a detailed discussion of the research challenge we have focused along with the description of our proposed solution - the Threat Miner system.

A. Problem Statement

Cyber warfare is a continuous struggle between mal-actors targeting computing systems and those defending against them. Cyber threats continue to grow with mal-actors utilising cutting-edge techniques and novel methods to achieve their objectives which highlights the need for stronger mechanisms to protect against such dynamic threats. Contemporary cyber defence mechanisms utilise a range of methods which rely on monitoring network and system-level events. However, with the growing use of the dark web by mal-actors to share exploits, breaches, and data leaks, the use of such information to strengthen defence mechanisms becomes an intriguing prospect.

A number of existing work are focused on development of proactive threat intelligence systems such as [33], [34], [35] and [36]. However, the missing piece (the problem-focused in this paper) is to explore the use of knowledge from the dark web to strengthen cyber defence mechanisms. The paper specifically attempts to address the core challenge i.e. transforming social interactions from the dark web into intelligence that can be shared across diverse sectors as well as being able to utilise this intelligence for detection and mitigation against emerging cyber threats.

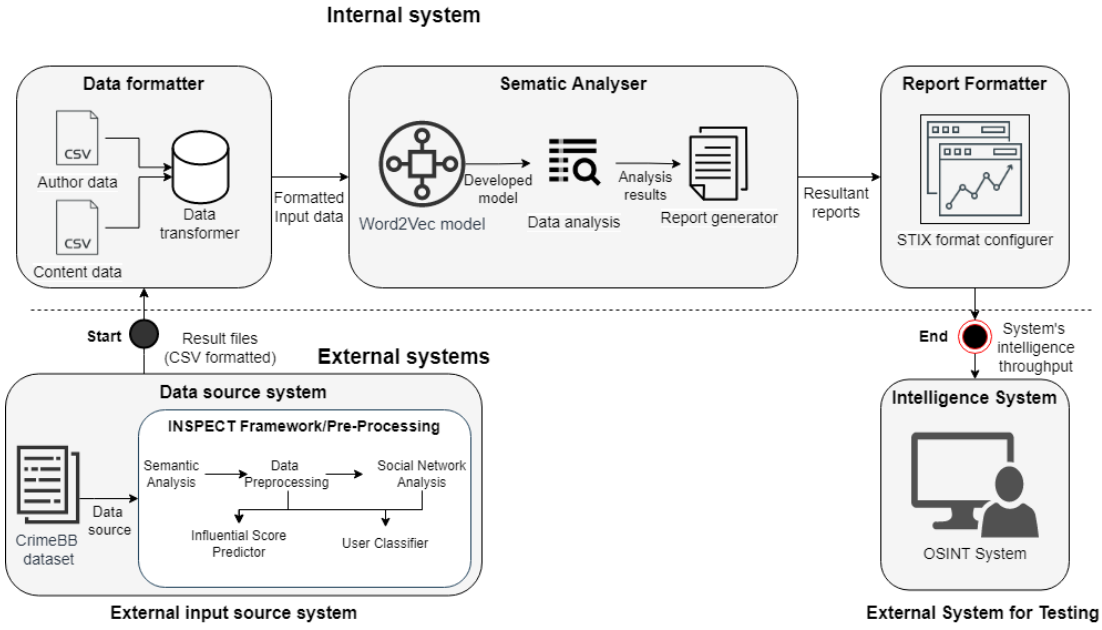

Fig. 4. Overview of threat miner architecture

B. System Design

Threat Miner is comprised of three modules that form the internal architecture of the system and are also linked with two external systems that represent the source of input dataset and testing of the Threat Miner. The architectural overview of this system is given in Fig 4 and further descriptions of each of the system modules are described in sequence as follows:

1) Data Formatter: Data formatter retrieves the dataset in the form of CSV files. Two CSV files are received as the input data source, which then is cleansed, merged and transformed into the required format as needed by the Threat Miner. This module also consists of the connecting point to the input source by an external system.

2) Semantic Analyser: The semantic Analyser is the core processing module of the Threat Miner system. This module is responsible for developing a semantic model, known as the Word2Vec model, by building its vocabulary and performing pre-processing on the data. The next step is to analyse data through the implementation of the Word2Vec model and generate reports as the output to that semantic analysis. This module generates threat intelligence reports by analysing input data with the help of the Word2Vec semantic model. Initially, two prototypes were designed for the semantic analysis including **Latent Semantic Analysis** and **Word2Vec** machine learning model. From the two prototypes, the Word2Vec model is more effective for this system because the output of the algorithm was significantly adaptable for the aims of this study than the output of the latent semantic analysis. Word2Vec model works on the search terms immediately and is adjustable to the large size of the dataset. For these reasons, Word2Vec was chosen as the algorithm used for this study.

3) Report Formatter: Threat Miner is responsible for providing CTI to the external systems in a readable format that can be easily integrated to the systems irrespective of their designs. Therefore, we have transformed the generated CTI from the semantic analysis module into the STIX format, which outputs results in the STIX formatted reports. *Report Formatter* module is designated to develop the CTI reports into the addressed STIX format.

4) External Systems: Two external systems are integrated with Threat Miner which is described as below:

1) Threat miner system utilises data about the activity of influencer hackers on the dark web forums. This includes author details including reputation and influencer value and the post contents by the authors along with the corresponding receiving author. The input data is in the form of CSV files which are handled and transformed by the threat miner.
2) The output from the Threat Miner system is expected to be consumed by an OSINT system to strengthen the overall security posture of organisational security.

The algorithm of the Threat Miner is given in algorithm 1.

VI. Implementation of Threat Miner

The implementation included two phases i.e. dataset preparation & setup and model generation and development. We describe these in more detail below.

A. Database Setup and Analysis

The data containing the posts included source, destination and the post contents. Sample data from the data file is shown in Table IV. The author data included the ID, author ID, reputation and an influencer score. The ID is the graph-generated ID and is the primary key used to identify users.

Algorithm 1 Technical interpretation - Threat miner

```
posts ← csv_to_dataframe(posts.csv)
profiles ← csv_to_dataframe(profiles.csv)
hackingKeywords ← ["sql","virus","compromise","malware",
"script","xss","phishing","spoofing","rat","encryption"]
cursor = 0
filteredPosts ← []
while posts ≠ null do
    if posts[cursor].getId() > 10 then
filteredPosts ← filteredPosts.addPost(posts[cursor])
    end if
end while
wordToVecModel ← Word2Vec(window = 10, min_count = 2)
keywordsListFromPosts ← wordToVecModel.develop_vocab(
filteredPosts, progress_per = 1000)
searchWords ← []
for i in hackingKeywords do
    for j in keywordsListFromPosts do
        similarityMeasure ← similarity_analyser_model(w1 = i, w2 =
j)
        if ctiCheck >= 0.9 then
            searchWords ← searchWords.append(i)
            searchWords ← searchWords.append(j)
        end if
    end for
    ctiFound ← []
    for searchWord in searchWords do
        for post in filteredPosts do
            if post.contains(searchWord) then
                ctiFound ← ctiFound.add(ctiSearch)
            end if
        end for
    end for
    cursor = 0
    while ctiFound ≠ null do
        sp ← STIXPackage()
        sr ← Report()
        sr.header ← Header()
        sr.header.description ← "CTIReport"
        ind ← Indicator()
        ind.title ← ctiFound[cursor].getPost()
        ttpTitle ← ctiFound[cursor].getId()
        activity ← TTP(title = ttpTitle)
        sp ← sp.add_indicator(ind)
        sp ← sp.add_ttp(activity)
        sp1 ← to_string(sp.to_xml())
        report ← "CTI%d.xml"%i
        file ← open("Reports/%s"%report,"w", encoding = "utf = 8")
        file ← file.write(sp1)
        file ← file.close()
    end while
```

The influencer score is calculated as part of our existing work utilising user activity (posts, replies, likes, etc) and is envisaged to represent the social stature of a user within the dark web forum. The author CSV file was saved into a variable called hackers after filtering data with a user reputation greater than 10. This created the list of influential hackers whose posts would be analysed. The value of 10 was chosen as it provided the optimal number of results. Results for different values can be seen in table V.

As can be seen in the table V, using a reputation of greater than 10 provided 7515 posts to be analyzed by the system. Through testing of the system later in this study, it was found that using a reputation greater than 50 did not provide enough results to effectively train the machine learning model on the dataset being used. Using a reputation value of 0 provided a large number of data to train the model but added a significant number of posts and users which were not considered influential and added *noise* to the data. It should be noted that it was found there were 3903 users with a reputation less than 0 and 9705 users with a reputation less than or equal to 10. This makes up 73.4% of users on the forum. There are 3509 users with a reputation greater than 10. Using a reputation of greater than 10 means only the top 26.6% of users' posts are included for analysis by the system.

B. Model Development

The next step involved getting posts from all the influential hackers by renaming the source header in the content file to ID. This allowed the two data frames to be combined together using the merge function in Pandas library. The content and the *iHackers* data frames were merged based on the ID column which allows the information in both data frames for that record to be written to a new data frame called *iContent* representing list of all the posts written by influential hackers.

In order to create the Word2Vec model, the class was defined and the data was pre-processed. The pre-processing utility also converted all the letters in the words to lowercase. Then the Word2Vec model itself was created by calling the Word2Vec algorithm from the Gensim library. The configuration describes the number of windows used, minimum count and the analyzing ability of the model. The parameters set for the model were the window size and minimum count. The window parameter was set to 10 as research [37] has shown that small values such as 1 or 2 often had little to no effect on the performance of the algorithm. Further, changes to performance were often found at window sizes of 10, 20, 30 and so forth. As the posts were often short, it was seen as appropriate to set the window size to 10. The minimum count parameter was set to 2 which tells the algorithm the shortest word to include when training because anything less than 2 letters is probably not significant. Then the model was called and used the build vocabulary function from the Gensim library which was given a list of tokenised posts from the iContent variable that had been pre-processed in the first part of this class. This trained the Word2Vec model on data specific to the dataset being used for this study. This is in addition to all pre-training that is included with the Word2Vec model as mentioned in section V-B. The Word2Vec model was now ready to be used for analysis on the data.

The next class to be defined was the "Analyser" class using Word2Vec model for the posts analysis from the influential users stored in the iContent data frame. First, two list variables were created; *word1List* and *Word2List*. word1List contained a series of technical terms related to hacking. These terms would form the search words and the Word2Vec model would aim to find words similar to these terms. There were ten search words i.e. SQL, virus, compromise, malware, script, XSS, phishing, spoofing, rat, and encryption. These were found to be some of the most popular hacking terms [38].

The second list variable was called word2List and contained the text that the Word2Vec model had learned from the posts in the dataset. Then an empty list called "searchWords" was created. This list would contain all the words that were found to be similar to the hacking terms in word1list and would form

TABLE IV
SAMPLE POSTS DATA

Source	*Destination*	*Post*
367	0	Lmao this is definitely the first i've seen before that actually includes a password, if you find out what database they got it from lemme know. I'm changing all my passwords just in-case.
1607	0	Definitely sounds like database dumps from a forum. Obviously dating way back, they've just sent the email to the email shown on file with the password shown. I've never had anything like this before, so it could be 'new' and they're hopeful of tricking people. Since a lot of people do fall for this shit."
1923	0	"I know several people that have also received this. Follows what appears to be a generic template.ḿ wondering if anyone has been able to engage in a twoway conversation with this actor. Their weakness will be their desire to financially profit, so Iḿ sure someone could reverse social engineer them into making a mistake by acting the fearful victim.
2285	0	Sent him a couple cents :D***IMG***[https://i.imgur.com/sAYQtyd.png]***IMG******LINK*** (Click to View)[javascript:void(0);]***LINK******IMG*** [https://thumbs.gfycat.com/SpryIncredibleBengaltiger-size$_restricted.gif$] $***IMG***$
2358	0	@Omni Just saw an article on the news about this a couple of days ago, seems like this is a new Nigerian prince type of scam.***LINK***http://www.ladbible.com/news/technology-...n-20180726[http://www.ladbible.com/news/technology-news-new-online-scan-claims-to-have-videos-of-you-watching-porn-20180726]***LINK***
3139	0	Do you remember which main sites you used this pass for and if they have had a data breach?
3676	0	***CITING***[https://hackforums.net/showthread.php? pid=57336769pid57336769]***CITING***¿ disable images in email¿ ???¿ profitseriously, do people actually fall for this?
5537	0	***CITING***[https://hackforums.net/showthread.php? pid=57336769pid57336769]***CITING***I just looked through all my emails... nothing of that nature.are smart , I have never seen this type of ransom before. Very clever

TABLE V
REPUTATION VALUES

Reputation Value	*Posts Returned*
0	13861
10	7515
50	4451
100	1034
500	278
1000	127
2000	17

the final list of words to be searched for. Then the comparison iterations between both the word lists were carried out.

Then duplicate words are removed by the *Analyser* class from searchWords list followed by identifying occurrences of posts comprised of searchWords. These identified posts along with user ID are saved in the list called ctiFound. Then ctiFound transformed into data frame known as ctilist which removed the duplicate posts and contained all the posts that were deemed to contain cyber threat intelligence, along with the posts author user ID.

The final class of the system was the "Report Generator" class that generates cyber threat intelligence reports in STIX format using STIX Python library by the OASIS Cyber Threat Intelligence Technical Committee. The sample report format is given below in fig 5.

VII. EXPERIMENTATION AND ANALYSIS

The Threat Miner system was evaluated based on the quality of the reports produced and whether the cyber threat intelligence it extracted was useful and effective. For instance, if the report did not provide useful information, it was considered a false positive as it had been incorrectly classed as *threat intelligence* by the system. However, such analysis and classification is a complex task requiring extensive knowledge of traits that form a threat intelligence report. For instance, this could include details from CVE database in case of a known threat and the task becomes even more complex for an unknown threat. Due to the complexity of such tasks, we render this for future research and for the purpose of this study, we use manual expert analysis to determine the quality of the reports, they had to be manually reviewed. The Threat Miner system generated a large amount of threat intelligence reports and it was impossible to conduct manual analysis for all of them in an effective manner. Therefore, a total of 150 reports were randomly chosen, 50 reports for each type, for manual expert review however we intend to extend our evaluation in future to address potential limitations of this process.

A. Experimentation outcomes

The results gathered from the experiments are presented in table VI, table VII and table VIII. In order to analyse performance of the Threat Miner approach, we classified threat reports (indicators of compromise (IoC)) into *good*, *bad*, and *vague* reports and evaluate the ratio between them. A report is classified as *good* if it represents a cyber threat which can also be linked to a known CVE. A *bad* report on the other hand had very little relevance to cyber threats and therefore rendered not useful. Further, our analysis also identified reports which are relevant to a cyber threat and can be linked to a known CVE however are incomplete in terms of the information they contain. We term these are *vague*.

The results were gathered for three different values of similarity score. The metrics gathered for each value of the similarity score were the total number of reports generated, the number of good reports, the number of bad reports and the number of vague reports. The graphical interpretation of results is given in fig VI, which provides the similarity scores of the reports generated.

```
b'<stix:STIX_Package \n\txmlns:stix="http://stix.mitre.org/stix-1"\n\txmlns:ttp="http://stix.mitre.org/TTP-1"\n
\txmlns:indicator="http://stix.mitre.org/Indicator-2"\n\txmlns:CTI Report="http://CRIReport.com"\n
\txmlns:xlink="http://www.w3.org/1999/xlink"\n\txmlns:ds="http://www.w3.org/2000/09/xmldsig#"\n
\txmlns:xs="http://www.w3.org/2001/XMLSchema"\n\txmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"\n\t
id="CTI Report:Package-b764f28d-e407-4bb3-9d72-a23c39747b34" version="1.2">\n
<stix:Indicators>\n
<stix:Indicator id="CTI Report:indicator-f67e46d9-f219-4862-8a40-3640b4ffe438" timestamp="2021-12-20T13:17:22.052139+00:00"
xsi:type=\'indicator:IndicatorType\'>\n
<indicator:Title> "So hay.\\n\\nI am using prorat to rat someone/some people lol.\\n\\nanyway I got a java thingy found it online.
 How do I test it if its working? \\nI don\'t really have any other pcs to test it on.\\n\\nand I doubled clicked on my own rat
 but it wont show up on my list. So I don\'t really know if its working or not.\\n\\nCan anyone give me some ideas on how can I
 test my rat? :P\\n\\n\\nthanks for your time and help!"</indicator:Title>\n
</stix:Indicator>\n
</stix:Indicators>\n
<stix:TTPs>\n
<stix:TTP id="CTI Report:ttp-a0710700-0a09-4c31-a6ad-a33f1134f750" timestamp="2021-12-20T13:17:22.052139+00:00"
xsi:type=\'ttp:TTPType\'>\n
<ttp:Title>85.0</ttp:Title>\n
</stix:TTP>\n
</stix:TTPs>\n
</stix:STIX_Package>\n'
```


Fig. 5. Example STIX report generated by Threat Miner

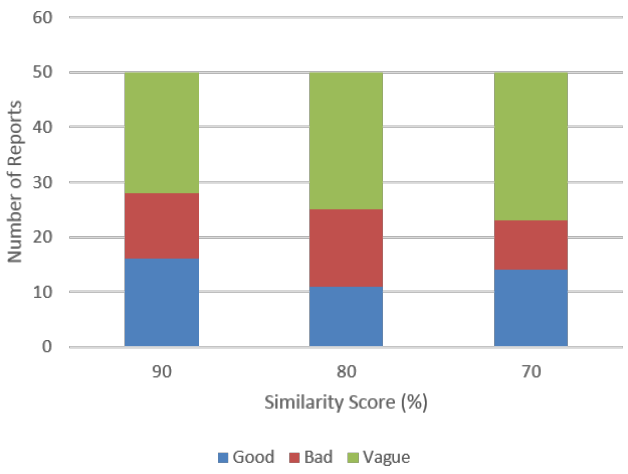

Fig. 6. Analysis of good, bad and vague intelligence reports

The number of reports generated for each value of similarity score can be seen in table VI.

TABLE VI
ANALYSIS OF SIMILARITY SCORE FOR DIFFERENT CLASSES

Similarity Score Value (%)	*Total Number of Reports*
90	257
80	2810
70	3629

A breakdown of the results showing the number of good, bad and vague reports can be seen in table VII and table VIII. It should be noted that this analysis was carried out on a subset of 50 reports.

Some samples of good reports can be seen in fig 7. Here, good reports are significantly providing information regarding the malicious attempts as described below.

1) The post at index 1568 discusses spoofing MAC addresses to attempt to bypass Windows Defender
2) The post at index 2074 talks about bypassing the login screen. This appears to be exploiting a known vulnerability CVE-2004-2339

Index 1568

"in macchanger you can randomize your mac so what I will probably do is spoof a mac that makes it look like my computer is from the same manufacturer of those users on the net. or I can just listen in on the ARP's from another mac and forge packets back to the router to get my keystream.\n\nthere was a conversation yesterday about once you have enumerated the network how would you go about crippling Windows Defender. I am looking for any ideas on how to bypass or cripple any of these dependencies preferably wuaueng.dll because that is the primary definition file or engine. any ideas? \n\n* wuapi.dll\n * wuaueng.dll\n * wucltui.dll\n * wups.dll\n * wuweb.dll\n * atl.dll\n * softpub.dll\n * wintrust.dll\n * initpki.dll\n * mssip32.dll\n\nI thought about crippling the updater then going to work on the engine in hexworkshop. since the first 15 - 25 lines are the opening of the file or dependency I think i will leave them alone and just fill a few sections with 0 data."

Index 2074

"On XP you can just boot in safe mode and login into main admin account from there and do what ever. That's how I got into my dad's computer now he won't let me touch after he found out lol. But for vista and 7 you are going to have to either use ophcrack or this other bootable thing I can't remember the name but it bypasses the login altogether supposedly

Fig. 7. Examples of good reports

In a similar manner, some samples of bad and vague reports are shown in fig 8 and fig 9.

Index 5460

"Why can't you just login with Filezilla or something and download it yourself?"

Index 6008

'Is he an admin user on the computer?'

Fig. 8. Examples of bad reports

Here, the reports, marked as bad reports 8 does not provide any useful technical details. As given below:

1) The post at index 5460 talks about logins, downloads and Filezilla but not in a way related to a cyber attack
2) The post at index 6008 talks about admin users on the computer but there is no obvious signs this is related to a cyber attack

Likewise, the vague reports are the type of resultant reports which are providing details regarding cyber crimes but are incomplete as shown below:

1) The post at index 3039 talks about a DDoS attack but states no target

TABLE VII
RESULTS BREAKDOWN

Similarity Score Value (%)	*Number of Good Reports*	*Number of Bad Reports*	*Number of Vague Reports*	*Ratio (Good/Bad/Vague)*
90	16	12	22	8:6:11
80	11	14	25	11:14:25
70	14	9	27	14:9:27
Average	**13.6**	**11.6**	**18.0**	**13.6:11.6:18.0**

TABLE VIII
RESULTS BREAKDOWN AS PERCENTAGE

Similarity Score Value (%)	*Good Reports (%)*	*Bad Reports (%)*	*Vague Reports (%)*
90	32.0	24.0	44.0
80	22.0	28.0	50.0
70	28.0	18.0	54.0
Average	**27.3**	**23.3**	**49.3**

2) The post at index 851 mentions compromising the network but the exact method is not defined in this post although it sounds like another post in the threat might do

Index 3039

'I don\'t think it\'s possible. You\'ll just have to wait and hope for luck.\nGet his login IP from the "Last Online" thing. DDoS him multiple times a day

Index 851

'As sirru5h mentioned i would use that method among others to compromise the network. The more information you know about their network the better you can compromise their machines.'

Fig. 9. Examples of vague reports

The analysis done when determining the number of good, bad and vague reports was based on 50 randomly selected reports. This was done due to the large number of reports generated for each value of similarity score. The ratio was calculated using these 50 reports and aims to provide a representation of the entire dataset.

B. Analysis of results

The results show the expected outcome, comprises of a higher number of reports with a lower value of the similarity score as per the hypothesis. More results being generated with lower values of similarity score because more words will be included in the words list used by the Word2Vec model. One interesting point depicted from results, given in table VI, is the significant jump in reports generated between similarity scores of 90% and 80%. There is an increase of 2553 reports yet the difference between similarity scores of 80% and 70% is only 819. The sharp increase between 90% and 80% could be explained as it is possible the majority of the words present in the dataset have a similarity of greater than 80% and there are less words that have a similarity score of less than 80%. Furthermore, since most posts are quite short and only contain a few sentences, it could be possible that most words are given a higher value similarity score than would be the case in a longer passage of text which becomes a limitation of the Word2Vec model and could be solved by increasing the amount of data used to train the model.

Analysing results further as shown in table VIII and VI, provides some more interesting trends like the largest number of good reports were generated when using a similarity score of greater than 90%. Another interesting fact is that the similarity score of greater than 70% produces more good results than a similarity score of greater than 80%. This can be explained by the method used to analyse the results. Since only 50 results were analysed and they were chosen at random this meant that some of the good reports would have been missed and there is a chance that more of the bad and vague reports were picked up. This combined with the fact that a similarity score of greater than 70% produced more reports than a similarity score of greater than 80% meaning there were more reports to randomly choose from. This is. a major limitation of the method of analysis used and if a full analysis of all the reports generated had been carried out it is expected that a similarity score of greater than 80% would have a higher percentage of good reports than a similarity score of greater than 70%.

Another interesting trend in the results is the high proportion of vague results. On average 49.3% of the reports generated were classed as being vague. A similarity score of greater than 90% produced the lowest percentage of vague reports with 44% of reports being classed as vague. Across all three similarity scores tested, the majority of reports generated were classed as being vague.

VIII. CONCLUSION AND FUTURE WORK

This study is focused on the challenge of strengthening cyber defence system through the use of information exchange between mal-actors over the dark web. Specifically, an AI-based system was created to automatically generate cyber threat intelligence reports (IoCs) based on analysis of data from dark web hacker forums. The CTI reports generated from this system vary in their usefulness, but the initial results are promising. With further work, the results could be improved

further to make the system more resilient to noise in the data and to minimise false-positive results. One of the key areas that could be improved is a more in-depth analysis of the reports generated. The current approach relied on manual analysis of the CTI reports which limits the number of reports that can be analysed. By analysing more of the reports, a better and more accurate representation of the data could be gathered. Another interesting area for future research could be to do more analysis on the users i.e. to examine how users interact with each other to identify a list of threats with high amounts of CTI generated. This could prove an interesting source of information and allow more cyber threat intelligence to be gathered and with more context.

IX. Acknowledgment

The authors would like to pay gratitude to the Cambridge Cyber Crime Center, UK for the provision and accessibility to the CrimeBB dataset, which has been used to conduct this research.

References

[1] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," pp. 1845–1854, April 2018.

[2] R. K. Goutam, "Importance of cyber security," *International Journal of Computer Applications*, February 2015.

[3] F. Cyber investigations and Response, "Triple digit increase in cyberattacks: What's next?" *Accentures blogs*, August 2021.

[4] "What is deep and dark web?" *Kaspersky blogs*, 2020, https://www.kaspersky.com/resource-center/threats/deep-web.

[5] M. Hatta, "Deep web, dark web, dark net: A taxonomy of "hidden" internet," *Annals of Business Administrative Science*, October 2020.

[6] P. R. Team, "Dark web monitoring: The good, the bad, and the ugly," 2019, https://www.digitalshadows.com/blog-and-research/dark-web-monitoring-the-good-the-bad-and-the-ugly/.

[7] A. A. Paracha, J. Arshad, and M. M. Khan, "S. u. s. you're sus! - identifying influencer hackers on dark web social networks," *Submitted to Computer and Electrical Engineering journal*, 2022.

[8] R. M. B. P. D. Gallagher, "Nist special publication 800-30 revision 1 - guide for conducting risk assessments," *MST Special Publication*, no. 95, September 2012.

[9] K. K. R. Choo, "The cyber threat landscape: Challenges and future research directions," *Computers and Security*, no. 719-731, 2011.

[10] B. Runciman, "Cybersecurity report 2020," *Itnow*, no. 28-29, 2020.

[11] M. O'Leary, "Cyber operations building, defending, and attacking modern computer networks," 2019.

[12] C. Martins and I. Medeiros, "Generating quality threat intelligence leveraging osint and a cyber threat unified taxonomy," *ACM Trans. Priv. Secur.*, vol. 25, no. 3, may 2022. [Online]. Available: https://doi.org/10.1145/3530977

[13] A. Tundis, S. Ruppert, and M. Mijhlhauser, "On the automated assessment of open-source cyber threat intelligence sources," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, no. 453-467, 2020.

[14] I. Li, X. Zhou, and A. Xue, "Open source threat intelligence discovery based on topic detection," *Proceedings - International Conference on Computer Communications and Networks*.

[15] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression," 2014.

[16] "Maec 5.0 specification," *MITRE Corporation*, 2017.

[17] "Oasis cyber threat intelligence technical committee," *STIX Version 2.1 OASIS Standard*, 2021, https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.pdf.

[18] T. Omitola, S. A. Rios, and J. G. Breslin, "Social semantic web mining," *Synthesis Lectures on the Semantic Web: Theory and Technology*, 2015.

[19] T. Landauer, "Latent semantic analysis : Theory, method and application," *CSCL '02: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, no. 742-743, 2002.

[20] C. Liu and Y. M. Wang, "On the connections between explicit semantic analysis and latent semantic analysis," *ACM International Conference Proceeding Series*, no. 1804-1808, 2012.

[21] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries. Springer Berlin Heidelberg*, no. 305-338, 2016.

[22] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal. Faculty of Engineering, Ain Shams University*, no. 1093-1113, 2014.

[23] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus," *Data Science and Engineering*, no. 157–175, 2019.

[24] M. Arora, V. Mittal, and P. Aggarwal, "Enactment of tf-idf and word2vec on text categorization," in *Proceedings of 3rd International Conference on Computing Informatics and Networks*. Springer, 2021, pp. 199–209.

[25] A. Perambai, "Theory behind word embeddings in word2vec," 2020.

[26] S. Gupta, "Sentiment analysis: Concept, analysis and applications," 2018.

[27] U. Akyazi, M. van Eeten, and C. H. Ganan, "Measuring cybercrime as a service (caas) offerings in a cybercrime forum," 2021.

[28] J. Cabrero-Holgueras and S. Pastrana, "A methodology for large-scale identification of related accounts in underground forums," September 2021.

[29] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, "Using entropy and mutual information to extract threat actions from cyber threat intelligence," *IEEE International Conference on Intelligence and Security Informatics (ISI)*, December 2018.

[30] W. Yang and K.-Y. Lam, "Automated cyber threat intelligence reports classification for early warning of cyber attacks in next generation soc," *Information and Communications Security*, no. 145–164, 2020.

[31] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, and A. Rauber, "A framework for cyber threat intelligence extraction from raw log data," *IEEE International Conference on Big Data (Big Data)*, 2019.

[32] L. Neil, S. Mittal, and A. Joshi, "Mining threat intelligence about open-source projects and libraries from code repository issues and bug reports," *International Conference on Intelligence and Security Informatics (ISI)*, December 2018.

[33] M. Schafer, M. Fuchs, M. Strohmeier, M. Engel, M. Liechti, and V. Lenders, "Black-widow: Monitoring the dark web for cyber security information," *International Conference on Cyber Conflict*, May 2019.

[34] I. Deliu, C. Leichter, and K. Franke, "Collecting cyber threat intelligence from hacker forums via a two-stage, hybrid process using support vector machines and latent dirichlet allocation," *IEEE Xplore*, December 2018.

[35] J. Robertson, A. Diab, E. Marin, E. Nunes, V. Paliath, J. Shakarian, and P. Shakarian, "Darknet mining and game theory for enhanced cyber threat intelligence," *The Cyber Defence Review*, vol. 1, no. 2, pp. 95–122, 2016.

[36] S. Samtani, R. Chinn, H. Chen, and J. F. N. Jr., "Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence," *Journal of Management Information Systems*, vol. 34, no. 4, January 2018.

[37] G. Di Gennaro, A. Buonanno, and F. A. Palmieri, "Considerations about learning word2vec," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 12 320–12 335, 2021.

[38] B. Chauhan, "Must-know hacking terminologies to safeguard your online business from hackers," 2021, https://www.getastra.com/blog/knowledge-base/hacking-terminologies/.